LilyPond

El tipografiador de música

Referencia de la notación

El equipo de desarrolladores de LilyPond

Este manual ofrece una referencia para toda la notación musical que se puede producir con LilyPond versión 2.26.0. Da por sentado que el lector está familiarizado con el material que hay en Sección "Manual de aprendizaje" en *Manual de Aprendizaje*.

Para mayor información sobre la forma en que este manual se relaciona con el resto de la documentación, o para leer este manual en otros formatos, consulte Sección "Manuales" en *Información general*.

Si le falta algún manual, encontrará toda la documentación en `https://lilypond.org/`.

Copyright © 1998–2026 por los autores.

La traducción de la siguiente nota de copyright se ofrece como cortesía para las personas de habla no inglesa, pero únicamente la nota en inglés tiene validez legal.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

> Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, versión 1.1 o cualquier versión posterior publicada por la Free Software Foundation; sin ninguna de las secciones invariantes. Se incluye una copia de esta licencia dentro de la sección titulada "Licencia de Documentación Libre de GNU".
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

Para la versión de LilyPond 2.26.0

Índice General

Notación musical

1 Alturas ... 3

1.1 Escritura de notas ... 3
1.1.1 Escritura de octava absoluta ... 3
1.1.2 Escritura de octava relativa ... 4
1.1.3 Alteraciones accidentales ... 8
1.1.4 Alturas aproximadas ... 10
1.1.5 Nombres de las notas en otros idiomas ... 10
1.2 Modificación de varias notas a la vez ... 12
1.2.1 Comprobación de octava ... 13
1.2.2 Transposición ... 14
1.2.3 Inversión ... 17
1.2.4 Retrogradación ... 17
1.2.5 Transformaciones modales ... 18
1.3 Imprimir las alturas ... 20
1.3.1 Clave ... 20
1.3.2 Armadura de la tonalidad ... 25
1.3.3 Corchetes de octava ... 27
1.3.4 Transposición de los instrumentos ... 30
1.3.5 Alteraciones accidentales automáticas ... 32
1.3.6 Glifos de alteración alternativos ... 40
1.3.7 Tesitura ... 41
1.4 Cabeza de las notas ... 44
1.4.1 Cabezas de nota especiales ... 44
1.4.2 Cabezas de notas de Notación Fácil ... 46
1.4.3 Cabezas de notas con formas diversas ... 48
1.4.4 Improvisación ... 52

2 Duraciones ... 53

2.1 Escritura de las duraciones (valores rítmicos) ... 53
2.1.1 Duración de las notas ... 53
2.1.2 Grupos especiales ... 56
2.1.3 Escalar las duraciones ... 62
2.1.4 Ligaduras de unión ... 63
2.2 Escritura de silencios ... 68
2.2.1 Silencios ... 68
2.2.2 Silencios invisibles ... 70
2.2.3 Silencios de compás completo ... 72
2.2.4 Cesuras ... 76
2.3 Impresión de las duraciones ... 79
2.3.1 Indicación de compás ... 79
2.3.2 Indicaciones metronómicas ... 86
2.3.3 Anacrusas ... 90
2.3.4 Música sin compasear ... 91
2.3.5 Notación polimétrica ... 93
2.3.6 División automática de las notas ... 97
2.3.7 Mostrar los ritmos de la melodía ... 98

2.4 Barras ... 101
2.4.1 Barras automáticas ... 101
2.4.2 Establecer el comportamiento de las barras automáticas ... 104
2.4.3 Barras manuales ... 114
2.4.4 Barras progresivas ... 118
2.4.5 Barras cruzadas ... 119
2.5 Compases ... 120
2.5.1 Barras de compás ... 120
2.5.2 Automatic bar lines ... 130
2.5.3 Numeración de los compases ... 132
2.5.4 Comprobación de compás y de número de compás ... 141
2.5.5 Llamadas de ensayo ... 142
2.5.6 Cuentas de compases ... 144
2.5.7 Divisores de sección ... 147
2.6 Asuntos rítmicos especiales ... 148
2.6.1 Notas de adorno ... 148
2.6.2 Alinear con una cadenza ... 153
2.6.3 Gestión del tiempo ... 155

3 Expresiones ... 156
3.1 Expresiones adosadas a las notas ... 156
3.1.1 Articulaciones y ornamentos ... 156
3.1.2 Matices dinámicos ... 160
3.1.3 Indicaciones dinámicas nuevas ... 169
3.2 Expresiones como curvas ... 171
3.2.1 Ligaduras de expresión ... 171
3.2.2 Ligaduras de fraseo ... 175
3.2.3 Marcas de respiración ... 177
3.2.4 Caídas y elevaciones ... 178
3.3 Expresiones como líneas ... 178
3.3.1 Glissando ... 178
3.3.2 Arpegio ... 183
3.3.3 Trinos ... 187

4 Repeticiones ... 190
4.1 Repeticiones largas ... 190
4.1.1 Repeticiones explícitas ... 191
4.1.2 Repeticiones sencillas ... 191
4.1.3 Finales alternativos (primera y segunda vez) ... 192
4.1.4 Otras variantes dentro de las secciones repetidas ... 194
4.1.5 Repeticiones Al fine ... 196
4.1.6 Estructura de las repeticiones de Segno ... 197
4.1.7 Apariencia de las repeticiones de Segno ... 201
4.1.8 Marcas de repetición manual ... 207
4.2 Repeticiones cortas ... 210
4.2.1 Repeticiones de compás o parte de ellos ... 211
4.2.2 Repeticiones de trémolo ... 214

5 Notas simultáneas ... 218
5.1 Una voz única ... 218
5.1.1 Notas en acorde ... 218
5.1.2 Repetición de acordes ... 220
5.1.3 Expresiones simultáneas ... 222
5.1.4 Racimos (clusters) ... 223
5.2 Varias voces ... 224
5.2.1 Polifonía en un solo pentagrama ... 224
5.2.2 Estilos de voz ... 229
5.2.3 Resolución de las colisiones ... 229
5.2.4 Combinación de silencios ... 235
5.2.5 Combinación automática de las partes ... 235
5.2.6 Escribir música en paralelo ... 241

6 Notación de los pentagramas ... 244
6.1 Impresión de los pentagramas ... 244
6.1.1 Crear instancias de pentagramas nuevos ... 244
6.1.2 Agrupar pentagramas ... 246
6.1.3 Grupos de pentagramas anidados ... 250
6.1.4 Separación de sistemas ... 251
6.2 Modificación de pentagramas sueltos ... 252
6.2.1 El símbolo del pentagrama ... 252
6.2.2 Pentagramas de Ossia ... 256
6.2.3 Ocultar pentagramas ... 260
6.3 Escritura de las particellas ... 265
6.3.1 Nombres de instrumentos ... 265
6.3.2 Citar otras voces ... 268
6.3.3 Formateo de las notas guía ... 272
6.3.4 Compresión de los compases vacíos ... 277

7 Anotaciones editoriales ... 280
7.1 Dentro del pentagrama ... 280
7.1.1 Seleccionar el tamaño de la tipografía para la notación ... 280
7.1.2 Indicaciones de digitación ... 284
7.1.3 Deslizamientos de dedo ... 287
7.1.4 Notas ocultas ... 291
7.1.5 Colorear objetos ... 292
7.1.6 Resaltados en el pentagrama ... 294
7.1.7 Corchetes para material opcional ... 297
7.1.8 Paréntesis ... 298
7.1.9 Plicas ... 300
7.2 Fuera del pentagrama ... 301
7.2.1 Nombre de las notas ... 301
7.2.2 Globos de ayuda ... 303
7.2.3 Líneas de rejilla ... 304
7.2.4 Corchetes de análisis ... 306

8 Texto ... 311
8.1 Escritura del texto ... 311
8.1.1 Panorámica de los objetos de texto ... 311

8.1.2 Inscripciones textuales ... 314
8.1.3 Objetos extensos de texto ... 315
8.1.4 Etiquetas de sección ... 316
8.1.5 Marcas de texto ... 317
8.1.6 Texto separado ... 321
8.2 Formatear el texto ... 323
8.2.1 Introducción al marcado de texto ... 323
8.2.2 Seleccionar la tipografía y su tamaño ... 327
8.2.3 Alineación de texto ... 330
8.2.4 Notación gráfica dentro de elementos de marcado ... 336
8.2.5 Notación musical dentro de elementos de marcado ... 338
8.3 Fuentes tipográficas ... 341
8.3.1 Tipos de fuente no soportados ... 341
8.3.2 Búsqueda de fuentes ... 342
8.3.3 Familias de fuentes ... 342
8.3.4 Funcionalidades de fuente ... 344
8.4 Cambiar las fuentes tipográficas ... 345

Notación especializada

9 Música vocal ... 349

9.1 Notación común para música vocal ... 349
9.1.1 Referencias para música vocal ... 349
9.1.2 Introducir la letra ... 350
9.1.3 Alineación de la letra a una melodía ... 351
9.1.4 Duración automática de las sílabas ... 353
9.1.5 Duración manual de las sílabas ... 356
9.1.6 Varias sílabas sobre una nota ... 357
9.1.7 Varias notas sobre una sílaba ... 358
9.1.8 Líneas de extensión y guiones ... 361
9.1.9 Cambios de vocal graduales ... 362
9.2 Técnicas específicas para la letra ... 362
9.2.1 Trabajar con letra y variables ... 362
9.2.2 Posicionamiento vertical de la letra ... 364
9.2.3 Colocación horizontal de las sílabas ... 368
9.2.4 Letra y repeticiones ... 370
9.2.5 Letras en divisi ... 379
9.2.6 Polifonía con letras compartidas ... 380
9.3 Versos ... 382
9.3.1 Añadir números de verso ... 382
9.3.2 Añadir expresiones dinámicas a los versos ... 383
9.3.3 Añadir el nombre de los cantantes a los versos ... 383
9.3.4 Versos con ritmos distintos ... 384
9.3.5 Imprimir los versos al final ... 387
9.3.6 Imprimir los versos al final en varias columnas ... 388
9.4 Canciones ... 390
9.4.1 Referencias para canciones ... 390
9.4.2 Hojas guía de acordes ... 390
9.5 Música coral ... 391
9.5.1 Referencias para música coral ... 391

9.5.2 Disposiciones de la partitura para música coral ... 392
9.6 Ópera y musicales ... 394
9.6.1 Referencias para ópera y musicales ... 395
9.6.2 Nombres de los personajes ... 395
9.6.3 Guías musicales ... 397
9.6.4 Música hablada ... 401
9.6.5 Diálogos encima de la música ... 401
9.7 Cánticos salmos e himnos ... 403
9.7.1 Referencias para cánticos y salmos ... 403
9.7.2 Preparar un cántico ... 403
9.7.3 Puntuación de un salmo ... 408
9.7.4 Barras de frase en himnos ... 411
9.7.5 Compases parciales en melodías de himno ... 412
9.8 Música vocal antigua ... 414

10 Teclados y otros instrumentos de varios pentagramas ... 415
10.1 Notación común para instrumentos de teclado ... 415
10.1.1 Referencias para teclados ... 416
10.1.2 Cambiar de pentagrama manualmente ... 417
10.1.3 Cambiar de pentagrama automáticamente ... 419
10.1.4 Líneas de cambio de pentagrama ... 421
10.2 Piano ... 424
10.2.1 Pedales de piano ... 424
10.3 Órgano ... 425
10.3.1 Indicaciones de pedal de órgano ... 425
10.4 Acordeón ... 426
10.4.1 Símbolos de Discanto ... 426
10.5 Arpa ... 427
10.5.1 Referencias para notación de arpa ... 427
10.5.2 Pedales de arpa ... 428

11 Instrumentos de cuerda sin trastes ... 429
11.1 Notación común para cuerdas sin trastes ... 429
11.1.1 Referencias para cuerdas sin trastes ... 429
11.1.2 Indicaciones de arco ... 430
11.1.3 Armónicos ... 430
11.1.4 Pizzicato Snap o de Bartók ... 432

12 Instrumentos de cuerda con trastes ... 433
12.1 Notación común para cuerdas con trastes ... 433
12.1.1 Referencias para cuerdas con trastes ... 433
12.1.2 Indicación de los números de cuerda ... 434
12.1.3 Tablaturas predeterminadas ... 436
12.1.4 Tablaturas personalizadas ... 455
12.1.5 Marcas de diagramas de trastes ... 459
12.1.6 Diagramas predefinidos de trastes ... 469
12.1.7 Diagramas de traste automáticos ... 480
12.1.8 Digitaciones de la mano derecha ... 482
12.2 Guitarra ... 484
12.2.1 Indicar la posición y la cejilla ... 484

12.2.2 Indicar armónicos y notas tapadas ... 485
12.2.3 Indicación de acordes de potencia o de quinta vacía ... 486
12.3 Banjo ... 487
12.3.1 Tablaturas de banjo ... 487
12.4 Laúd ... 488
12.4.1 Tablaturas de laúd ... 488

13 Percusión ... 490
13.1 Notación común para percusión ... 490
13.1.1 Referencias para percusión ... 490
13.1.2 Notación básica de percusión ... 490
13.1.3 Redobles ... 491
13.1.4 Percusión afinada ... 492
13.1.5 Pautas de percusión ... 492
13.1.6 Pautas de percusión personalizadas ... 495
13.1.7 Notas fantasma ... 497

14 Instrumentos de viento ... 498
14.1 Notación común para instrumentos de viento ... 498
14.1.1 Referencias para instrumentos de viento ... 498
14.1.2 Digitaciones ... 499
14.2 Gaita ... 501
14.2.1 Definiciones para la gaita ... 501
14.2.2 Ejemplo de música de gaita ... 502
14.3 Instrumentos de viento madera ... 503
14.3.1 Diagramas de posiciones para viento madera ... 503

15 Notación de acordes ... 510
15.1 Modo de acordes ... 510
15.1.1 Panorámica del modo de acordes ... 510
15.1.2 Acordes más usuales ... 511
15.1.3 Acordes extendidos y alterados ... 512
15.1.4 Inversiones de acorde y realizaciones especiales ... 515
15.2 Imprimir los acordes ... 515
15.2.1 Impresión de los nombres de acorde ... 515
15.2.2 Nombres de acorde personalizados ... 518
15.2.3 Rejillas de acordes ... 524
15.3 Bajo cifrado ... 528
15.3.1 Introducción al bajo cifrado ... 528
15.3.2 Escritura del bajo cifrado ... 529
15.3.3 Imprimir el bajo cifrado ... 533

16 Música contemporánea ... 537
16.1 Altura y armonía en la música contemporánea ... 537
16.1.1 Referencias para alturas y armonía en música contemporánea ... 537
16.1.2 Notación microtonal ... 537
16.1.3 Armonía y armaduras de tonalidad contemporáneas ... 537
16.2 Enfoques contemporáneos del ritmo ... 537
16.2.1 Referencias para enfoque contemporáneos del ritmo ... 537
16.2.2 Grupos de valoración especial en música contemporánea ... 537

16.2.3 Compases contemporáneos ... 537
16.2.4 Notación polimétrica extendida ... 537
16.2.5 Barrado en música contemporánea ... 537
16.2.6 Líneas divisorias en música contemporánea ... 537
16.3 Notación gráfica ... 537
16.4 Técnicas de partitura contemporáneas ... 539
16.5 Nuevas técnicas instrumentales ... 539
16.6 Lecturas adicionales y partituras de interés ... 539
16.6.1 Libros y artículos sobre notación musical contemporánea ... 539
16.6.2 Partituras y ejemplos musicales ... 539

17 Notación antigua ... 540
17.1 Panorámica de los estilos contemplados ... 541
17.2 Notación antigua: funcionalidades comunes ... 542
17.2.1 Contextos predefinidos ... 542
17.2.2 Ligaduras ... 542
17.2.3 Custos ... 543
17.3 Tipografiar música mensural ... 544
17.3.1 Contextos de la música mensural ... 544
17.3.2 Claves de la música mensural ... 545
17.3.3 Indicaciones de compás de la música mensural ... 546
17.3.4 Cabezas de nota de la música mensural ... 547
17.3.5 Corchetes de la música mensural ... 548
17.3.6 Silencios de la música mensural ... 548
17.3.7 Alteraciones y armaduras de la música mensural ... 549
17.3.8 Alteraciones de anotación (*musica ficta*) ... 549
17.3.9 Ligaduras mensurales blancas ... 550
17.4 Tipografiado del canto gregoriano ... 552
17.4.1 Contextos del canto gregoriano ... 552
17.4.2 Claves de canto gregoriano ... 553
17.4.3 Alteraciones y armaduras de canto gregoriano ... 553
17.4.4 Divisiones ... 554
17.4.5 Articulaciones del canto gregoriano ... 554
17.4.6 Puntos de aumentación (*morae*) ... 555
17.4.7 Ligaduras de neumas cuadrados gregorianos ... 556
17.5 Tipografiado del canto kievano en notación cuadrada ... 562
17.5.1 Contextos del canto kievano ... 562
17.5.2 Claves del canto kievano ... 563
17.5.3 Notas del canto kievano ... 563
17.5.4 Alteraciones accidentales del canto kievano ... 564
17.5.5 Líneas divisorias del canto kievano ... 564
17.5.6 Melismas del canto kievano ... 564
17.6 Trabajar con música antigua: escenarios y soluciones ... 565
17.6.1 Incipits ... 566
17.6.2 Disposición tipo «Mensurstriche» ... 566
17.6.3 Transcripción de canto gregoriano ... 567
17.6.4 Antigua y moderna desde la misma fuente ... 570

18 Músicas del mundo ... 573
18.1 Notación común para músicas no occidentales ... 573
18.1.1 Extensión de la notación y los sistemas de afinación ... 573

18.2 Música árabe ... 573
18.2.1 Referencias para música árabe ... 573
18.2.2 Nombres de nota árabes ... 574
18.2.3 Armaduras de tonalidad árabes ... 575
18.2.4 Indicaciones de compás árabes ... 578
18.2.5 Ejemplo de música árabe ... 578
18.2.6 Lecturas adicionales sobre música árabe ... 579
18.3 Música clásica de Turquía ... 580
18.3.1 Referencias para música clásica de Turquía ... 580
18.3.2 Nombres de las notas en turco ... 580
18.3.3 Armaduras de tonalidad de la música turca ... 580
18.3.4 Lecturas adicionales sobre música turca ... 581
18.4 Música clásica persa ... 582
18.4.1 Notación de la música persa ... 582
18.4.2 Afinaciones persas ... 582
18.4.3 Armaduras de tonalidad persas ... 583
18.4.4 Lecturas adicionales sobre música persa ... 584

Entrada y salida generales

19 Modos de entrada ... 587

20 Estructura del código de entrada ... 590
20.1 Estructura de una partitura ... 590
20.2 Varias partituras en un libro ... 591
20.3 Varios archivos de salida a partir de uno de entrada ... 593
20.4 Nombres de los archivos de salida ... 593
20.5 Estructura del archivo ... 594

21 Títulos y encabezamientos ... 598
21.1 Crear títulos, encabezamientos y pies de página ... 598
21.1.1 Explicación de los títulos ... 598
21.1.2 Presentación predeterminada de los títulos de partes de libro y partitura ... 601
21.1.3 Disposición predeterminada de las cabeceras y pies de página ... 605
21.2 Títulos, encabezamientos y pies de página personalizados ... 606
21.2.1 Formateo personalizado del texto de los bloques de título ... 606
21.2.2 Personalización de los títulos ... 606
21.2.3 Disposición personalizada de cabeceras y pies de página ... 610
21.3 Crear metadatos en los archivos de salida ... 611
21.4 Crear notas al pie ... 612
21.4.1 Notas al pie dentro de expresiones musicales ... 612
21.4.2 Notas al pie en texto independiente ... 618
21.5 Creación de notas interiores ... 621
21.6 Referencia a números de página ... 622
21.7 Tabla de contenidos ... 623

22 Trabajar sobre los archivos de entrada ... 627
22.1 Inclusión de archivos de LilyPond ... 627
22.2 Distintas ediciones a partir de una misma fuente ... 628
22.2.1 Uso de las variables ... 628
22.2.2 Uso de etiquetas ... 630
22.2.3 Uso de ajustes globales ... 641
22.3 Uso de las funciones musicales ... 641
22.3.1 Sintaxis de las funciones de sustitución ... 641
22.3.2 Ejemplos de funciones de sustitución ... 642
22.3.3 Cómo evitar la compartición de expresiones musicales ... 644
22.3.4 Funciones de sustitución y escritura de octava relativa ... 645
22.4 Caracteres especiales ... 647
22.4.1 Codificación del texto ... 647
22.4.2 Unicode ... 647
22.4.3 Alias de ASCII ... 648

23 Controlar la salida ... 650
23.1 Extracción de fragmentos de música ... 650
23.2 Saltar la música corregida ... 650
23.3 Formatos de salida alternativos ... 651
23.3.1 Salida de SVG ... 651
23.4 Empotrar archivos dentro de la salida PDF ... 652
23.5 Sustituir la tipografía de la notación ... 652

24 Creación de salida MIDI ... 655
24.1 Notación contemplada por el MIDI ... 655
24.2 Notación no contemplada en el MIDI ... 656
24.3 El bloque MIDI ... 656
24.4 Control de las dinámicas del MIDI ... 657
24.4.1 Indicaciones de matiz dinámico en el MIDI ... 657
24.4.2 Establecimiento del volumen MIDI ... 658
24.4.3 Establecimiento de las propiedades del bloque MIDI ... 660
24.5 Uso de los instrumentos MIDI ... 661
24.6 Uso de las repeticiones con el MIDI ... 662
24.7 Asignación de canales MIDI ... 663
24.8 Propiedades de contexto para efectos MIDI ... 665
24.9 Enriquecimiento de la salida MIDI ... 666
24.9.1 El script `articulate` ... 666
24.9.2 El script `swing` ... 667

25 Extraer información musical ... 669
25.1 Mostrar la notación de LilyPond ... 669
25.2 Impresión de las expresiones musicales de Scheme ... 669
25.3 Guardar los eventos musicales en un archivo ... 669

Problemas de espaciado

26 Disposición de la página ... 673

26.1 El bloque \paper ... 673
26.2 Tamaño del papel y escalado automático ... 674
26.2.1 Fijar el tamaño del papel ... 674
26.2.2 Escalado automático al tamaño del papel ... 675
26.3 Variables de espaciado de \paper verticales fijas ... 676
26.4 Variables de espaciado de \paper verticales flexibles ... 677
26.4.1 Estructura de las listas-A de espaciado vertical flexible ... 677
26.4.2 Lista de variables de espaciado de \paper verticales flexibles ... 678
26.5 Variables de espaciado de \paper horizontales ... 679
26.5.1 Variables de \paper para la anchura y los márgenes ... 679
26.5.2 Variables de \paper para el modo de doble cara ... 680
26.5.3 Variables de \paper para desplazamientos y sangrados ... 681
26.6 Otras variables de \paper ... 681
26.6.1 Variables de \paper para los saltos de línea ... 681
26.6.2 Variables de \paper para los saltos de página ... 682
26.6.3 Variables de \paper para la numeración de las páginas ... 683
26.6.4 Variables de \paper relativas a los encabezamientos y marcados ... 684
26.6.5 Variables de \paper para la depuración ... 685

27 Disposición de la partitura ... 686

27.1 El bloque \layout ... 686
27.2 Establecer el tamaño del pentagrama ... 688

28 Saltos ... 692

28.1 Saltos de línea ... 692
28.2 Saltos de página ... 696
28.2.1 Saltos de página manuales ... 696
28.2.2 Saltos de página óptimos ... 697
28.2.3 Saltos de página mínimos ... 697
28.2.4 Saltos de página del tipo página única ... 698
28.2.5 Saltos de página de línea única ... 698
28.2.6 Saltos de página del tipo línea única con altura automática ... 698
28.2.7 Paso de página óptimo ... 698

29 Espaciado vertical ... 701

29.1 Espaciado vertical flexible dentro de los sistemas ... 701
29.1.1 Propiedades de espaciado dentro de los sistemas ... 701
29.1.2 Espaciado de pautas no agrupadas ... 704
29.1.3 Espaciado de pautas agrupadas ... 706
29.1.4 Espaciado de las líneas que no son pautas ... 707
29.2 Posicionamiento explícito de los pentagramas y los sistemas ... 708
29.3 Evitar las colisiones verticales ... 715

30 Espaciado horizontal ... 718
30.1 Panorámica del espaciado horizontal ... 718
30.2 Sección de espaciado nueva ... 719
30.3 Cambiar el espaciado horizontal globalmente ... 720
30.3.1 Estiramiento uniforme de los grupos especiales ... 721
30.3.2 Espaciado estricto de las notas ... 722
30.4 Ajuste del espaciado horizontal para objetos de disposición específicos ... 722
30.4.1 Panorámica de los trucajes de espaciado horizontal específicos de cada objeto ... 722
30.4.2 Espaciado entre elementos no musicales adyacentes ... 723
30.4.3 Espaciado entre columnas adyacentes ... 725
30.5 Longitud de la línea ... 727
30.6 Notación proporcional ... 728

31 Encajar la música en menos páginas ... 734
31.1 Mostrar el espaciado ... 734
31.2 Cambiar el espaciado ... 735

Cambiar los valores por omisión

32 Ajuste fino de la salida ... 741

33 Contextos de interpretación ... 742
33.1 Explicación de los contextos ... 742
33.1.1 Definiciones de salida - estructura de los contextos ... 742
33.1.2 `Score`: el contexto maestro ... 742
33.1.3 Contextos del nivel superior: contenedores de pentagramas ... 743
33.1.4 Contextos de nivel intermedio. Pentagramas ... 743
33.1.5 Contextos del nivel más bajo. Voces ... 744
33.2 Crear y referenciar contextos ... 746
33.3 Mantener vivos los contextos ... 749
33.4 Modificar los complementos (plug-ins) de contexto ... 752
33.5 Cambiar los valores por omisión de los contextos ... 753
33.5.1 Cambiar todos los contextos del mismo tipo ... 753
33.5.2 Cambiar solamente un contexto determinado ... 756
33.5.3 Orden de precedencia ... 758
33.6 Definir contextos nuevos ... 758
33.7 Orden de disposición de los contextos ... 762

34 Explicación del Manual de referencia de funcionamiento interno ... 764
34.1 Navegar por la referencia del programa ... 764
34.2 Interfaces de la presentación ... 765
34.3 Determinar la propiedad del grob ... 766

35 Modificar las propiedades ... 768

35.1 Panorámica de la modificación de las propiedades ... 768
35.2 Las instrucciones \set y \unset ... 768
35.3 Las instrucciones \override y \revert ... 770
35.4 La instrucción \once ... 771
35.5 \set frente a \override ... 772
35.6 Las instrucciones \tweak y \single ... 773
35.7 La instrucción \offset ... 775
35.8 Modificación de las listas-A ... 780

36 Conceptos y propiedades útiles ... 783

36.1 Dirección y posición ... 783
36.1.1 Indicadores de dirección de las articulaciones ... 783
36.1.2 La propiedad de dirección ... 784
36.2 Distancias y medidas ... 784
36.3 Dimensiones ... 785
36.4 Objetos de extensión (spanners) ... 786
36.4.1 Modificar objetos de extensión fragmentados ... 786
36.4.2 Establecimiento de longitudes mínimas para los objetos de extensión ... 788
36.4.3 Control de los puntos extremos de un objeto de extensión ... 791
36.5 Estilos de línea ... 791
36.6 Líneas de extensión ... 792
36.7 Visibilidad de los objetos ... 794
36.7.1 Eliminar el sello ... 794
36.7.2 Hacer transparentes los objetos ... 795
36.7.3 Pintar los objetos de blanco ... 795
36.7.4 Uso de break-visibility (visibilidad en el salto) ... 796
36.7.5 Consideraciones especiales ... 798
36.8 Rotación de objetos ... 801
36.8.1 Rotación de objetos de presentación ... 801
36.8.2 Rotación de elementos de marcado ... 801
36.9 Alineación de objetos ... 801
36.9.1 Establecer directamente X-offset y Y-offset ... 802
36.9.2 Uso del side-position-interface ... 803
36.9.3 Uso del self-alignment-interface ... 803
36.9.4 Uso del break-alignable-interface ... 804
36.10 Modificación de los sellos ... 807
36.11 Modificación de las formas ... 808
36.11.1 Modificación de ligaduras de unión y de expresión ... 808

Apéndices

A Instrucciones de marcado ... 815

A.1 Instrucciones de marcado de texto ... 815
A.1.1 Font markup ... 815
A.1.2 Markup for text alignment ... 827
A.1.3 Graphical markup ... 845
A.1.4 Markup for music and musical symbols ... 856
A.1.5 Conditional markup ... 868

A.1.6 Instrument-specific markup ... 869
A.1.7 Accordion registers ... 874
A.1.8 Other markup commands ... 879
A.2 Instrucciones de lista de marcado de texto ... 891

B Tablas del manual sobre notación ... 895
B.1 Cuadro de nombres de acordes ... 895
B.2 Modificadores de acorde más usuales ... 895
B.3 Afinaciones predefinidas de instrumentos de cuerda ... 898
B.4 Diagramas predefinidos de posiciones de trastes ... 899
B.4.1 Diagramas para guitarra ... 899
B.4.2 Diagramas para ukelele ... 901
B.4.3 Diagramas para mandolina ... 902
B.5 Tamaños de página predefinidos ... 904
B.6 Instrumentos MIDI ... 906
B.7 Lista de colores ... 907
B.8 La tipografía Emmentaler ... 912
B.8.1 Tablas de glifos modernos ... 912
B.8.2 Tablas de glifos antiguos ... 921
B.9 Estilos de cabezas de nota ... 927
B.10 Conjuntos de glifos de alteraciones accidentales ... 927
B.11 Estilos de clave ... 928
B.11.1 Claves estándar ... 928
B.11.2 Clave de pentagramas de percusión ... 929
B.11.3 Claves de pautas de tablatura ... 929
B.11.4 Claves de la música antigua ... 930
B.12 Lista de caracteres especiales ... 932
B.13 Lista de articulaciones ... 934
B.13.1 Indicaciones de articulación ... 934
B.13.2 Indicaciones de adornos ... 934
B.13.3 Indicaciones de calderones ... 935
B.13.4 Indicaciones de instrumentos específicos ... 935
B.13.5 Indicaciones de repetición ... 936
B.13.6 Indicaciones antiguas ... 936
B.14 Lista de signos de respiración ... 936
B.15 Notas de percusión ... 937
B.16 Lista de líneas divisorias ... 939
B.17 Valores por omisión para `outside-staff-priority` ... 943
B.18 Valores por omisión para `script-priority` ... 944
B.19 Objetos gráficos (grobs) y sus símbolos de break-align ... 945
B.20 Glosario técnico ... 946
B.21 Funciones musicales disponibles ... 949
B.22 Identificadores de modificación de contextos ... 968
B.23 Variables del papel ... 969
B.24 Convenciones de nomenclatura ... 978
B.25 Predicados de tipo predefinidos ... 978
B.25.1 R5RS primary predicates ... 978
B.25.2 R5RS secondary predicates ... 979
B.25.3 Guile predicates ... 979
B.25.4 LilyPond scheme predicates ... 979
B.25.5 LilyPond exported predicates ... 980

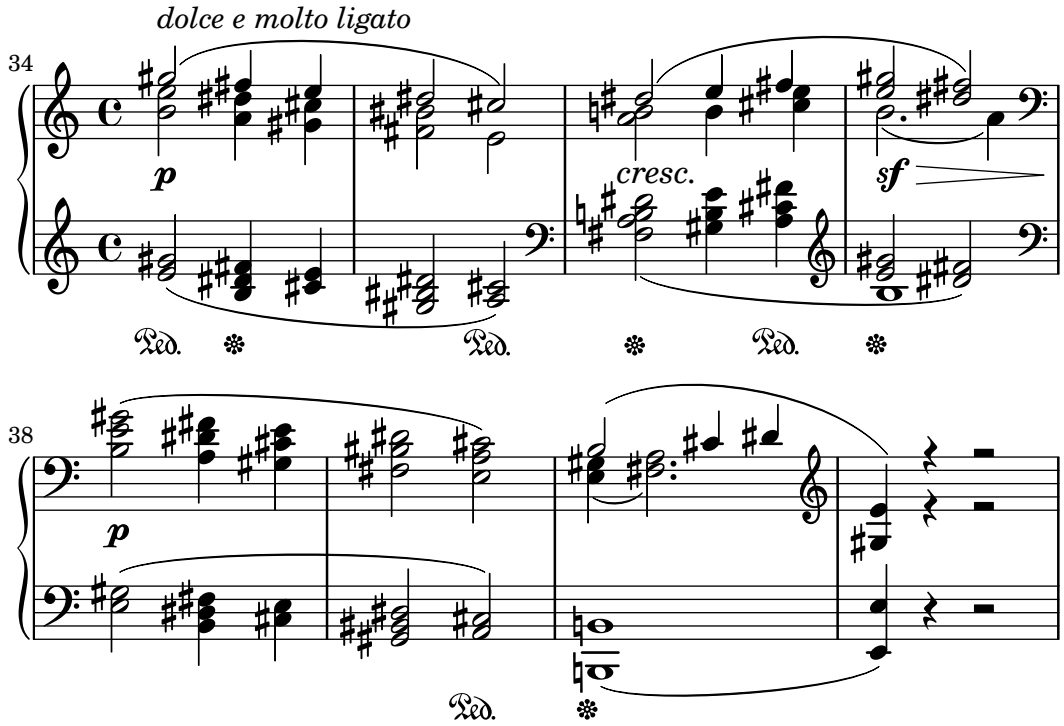
C Hoja de referencia rápida 982

D GNU Free Documentation License 985

E Índice de instrucciones y conceptos 992

Notación musical

1 Alturas



En esta sección se discute cómo especificar la altura de las notas. Este proceso se compone de tres fases: entrada, modificación y salida.

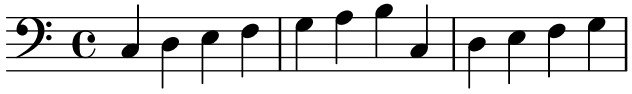
1.1 Escritura de notas

En esta sección se describe la manera de introducir la altura de las notas. Existen dos formas distintas de colocar las notas en su octava correspondiente: el modo absoluto y el relativo. En casi todas las ocasiones, será más práctico el modo relativo.

1.1.1 Escritura de octava absoluta

El nombre de una nota se especifica usando las letras minúsculas de la `a` a la `g`. Las notas cuyos nombres van desde `c` hasta `b` se imprimen en la octava inferior al Do central.

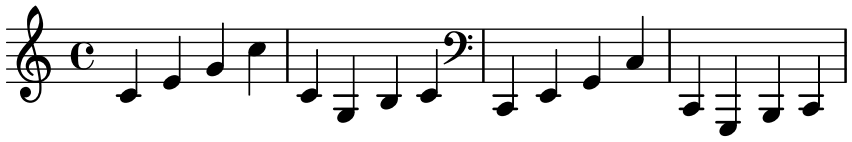
```
{
  \clef bass
  c4 d e f
  g4 a b c
  d4 e f g
}
```



Se pueden especificar otras octavas mediante una comilla simplee (`'`) o una coma (`,`) . Cada `'` eleva la altura en una octava; cada `,` baja la altura una octava.

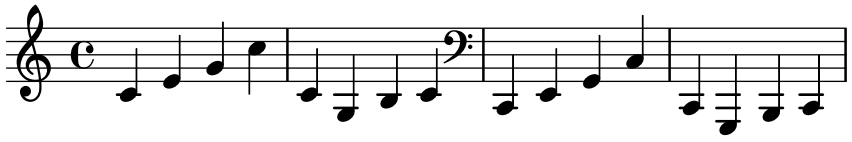
```
{
  \clef treble
  c'4 e' g' c''
  c'4 g b c'
```

```
  \clef bass
  c,4 e, g, c
  c,4 g,, b,, c,
}
```



Las marcas de octava comunes se pueden escribir una sola vez sobre una nota de referencia si se emplea `\fixed` antes de la música. Las notas dentro de `\fixed` solo necesitan las marcas de apóstrofo `'` o de coma `,` cuando están por encima o por debajo de la octava de la nota de referencia.

```
{
  \fixed c' {
    \clef treble
    c4 e g c'
    c4 g, b, c
  }
  \clef bass
  \fixed c, {
    c4 e g c'
    c4 g, b, c
  }
}
```



La altura de las notas de la expresión musical que sigue a `\fixed` no reultan afectadas por un `\relative` circundante, que se estudia a continuación.

Véase también

Glosario musical: Sección “Pitch names” en *Glosario Musical*.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

1.1.2 Escritura de octava relativa

La entrada de octava absoluta requere que se especifique la octava para todas y cada una de las notas. En contraste con ello, el modo de entrada de octava relativa especifica cada octava en relación a la nota anterior: si se cambia la octava de una nota ello afectará a todas las notas siguientes.

El modo relativo de notas se debe introducir de forma explícita usando la instrucción `\relative`:

`\relative` *`altura_inicial expresión_musical`*

En el modo relativo, se supone que cada nota se encuentra lo más cerca posible de la nota anterior. Esto significa que la octava de una nota que está dentro de *`expresión_musical`* se calcula como sigue:

- Si no se usa ninguna marca de cambio de octava en una nota, su octava se calcula de forma que el intervalo que forme con la nota anterior sea menor de una quinta. Este intervalo se determina sin considerar las alteraciones.

- Se puede añadir una marca de cambio de octava `'` o `,` para elevar o bajar la altura, respectivamente, en una octava más en relación con la altura calculada sin esta marca.
- Se pueden usar varias marcas de cambio de octava. Por ejemplo, `''` y `,,` alteran la altura en dos octavas.
- La altura de la primera nota es relativa a *altura_inicial*. *altura_inicial* se especifica en modo de octava absoluta. ¿Qué opciones tienen sentido?

  `c` (Do), en cualquier octava
  : La identificación del Do central con `c'` es algo bastante básico, por lo que tiende a ser fácil encontrar octavas de `c`. Si nuestra música comienza con `gis` (un Sol sostenido) por encima de `c'''`, tendríamos que escribir algo como `\relative { gis''' ... }`

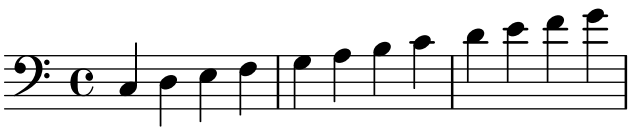
  una nota que está una o más octavas de la primera nota de dentro
  : Escribir `\relative { gis''' ... }` hace que sea fácil determinar la altura absoluta de la primera nota de dentro.

  ninguna altura de inicio explícita
  : La forma `\relative { gis''' ... }` sirve como una versión más compacta de la opción anterior: la primera nota de dentro se escribe ella misma en altura absoluta (esto resulta ser equivalente a elegir `f` (Fa) como la altura de referencia).

  La documentación suele utilizar esta última opción.

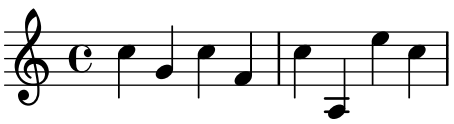
Aquí podemos ver el modo relativo en acción:

```
\relative {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```



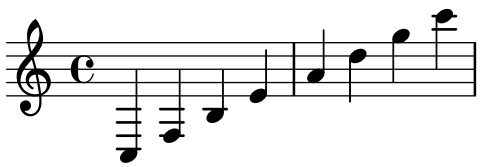
Las marcas de cambio de octava se utilizan para intervalos mayores de la cuarta:

```
\relative {
  c'' g c f,
  c' a, e'' c
}
```



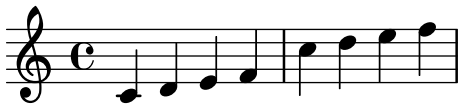
Una serie de notas sin ninguna marca de octava puede, a pesar de todo, abarcar intervalos muy grandes:

```
\relative {
  c f b e
  a d g c
}
```



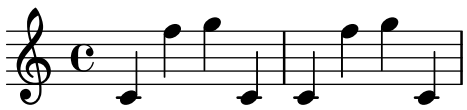
Cuando hay unos bloques \relative anidados dentro de otros, el bloque \relative más interno comienza con su propia nota de referencia independientemente del \relative exterior.

```
\relative {
  c' d e f
  \relative {
    c'' d e f
  }
}
```



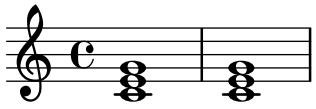
Para usar el modo absoluto dentro de \relative, escriba la música absoluta dentro de \fixed c { ... } y las alturas absolutas no afectarán a las octavas de la música relativa:

```
\relative {
  c'4 \fixed c { f'' g'' } c |
  c4 \fixed c'' { f g } c
}
```



\relative no tiene efecto sobre los bloques \chordmode.

```
\new Staff {
  \relative c''' {
    \chordmode { c1 }
  }
  \chordmode { c1 }
}
```

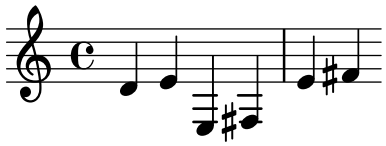


\relative no se permite dentro de los bloques \chordmode.

La música que está dentro de un bloque \transpose es absoluta, a no ser que se incluya una instrucción \relative.

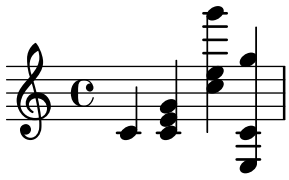
```
\relative {
  d' e
  \transpose f g {
    d e
    \relative {
      d' e
    }
  }
```

```
}
```



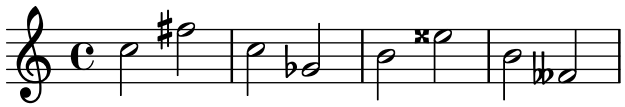
Si el elemento anterior es un acorde, la primera nota del acorde se utiliza para determinar la primera nota del siguiente acorde. Dentro de los acordes, la siguiente nota siempre está en relación a la anterior.

```
\relative {
  c'
  <c e g>
  <c' e g'>
  <c, e, g''>
}
```



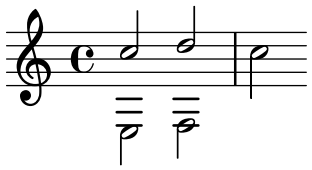
Como se explicó más arriba, la octava de las notas se calcula solamente a partir de sus nombres, sin tener en cuenta alteración alguna. Por tanto, un Mi doble sostenido después de un Si se escribirá más agudo, mientras que un Fa doble sostenido se escribirá más grave. En otras palabras, se considera a la cuarta doble aumentada un intervalo menor que la quinta doble disminuida, independientemente del número de semitonos de cada uno de ellos.

```
\relative {
  c''2 fis
  c2 ges
  b2 eisis
  b2 feses
}
```



En situaciones complejas puede ser útil recuperar una determinada altura, independientemente de lo que hubiera ocurrido con anterioridad. Esto se puede hacer utilizando `\resetRelativeOctave`:

```
\relative {
  <<
    { c''2 d }
    \\
    { e,,2 f }
  >>
  \resetRelativeOctave c''
  c2
}
```



Véase también

Glosario musical: Sección “fifth” en *Glosario Musical*, Sección “interval” en *Glosario Musical*, Sección “Pitch names” en *Glosario Musical*.

Referencia de la notación: Sección 1.2.1 [Comprobación de octava], página 13.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

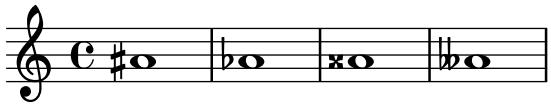
Referencia de funcionamiento interno: Sección “RelativeOctaveMusic” en *Referencia de Funcionamiento Interno*.

1.1.3 Alteraciones accidentales

> **Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas especifican las alturas; la armadura y la clave determinan de qué forma se presentan estas alturas. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

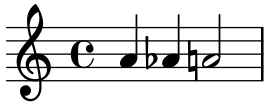
Se escribe una nota *sostenida* añadiendo `is` al nombre de la nota, y un *bemol* añadiendo `es`. Como es de esperar, un *dobles sostenido* y un *doble bemol* se obtiene añadiendo `isis` o `eses`. Esta sintaxis deriva de los nombres de las notas en neerlandés. Para utilizar otros nombres para las alteraciones, consulte Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10.

```
\relative c'' { ais1 aes aisis aeses }
```



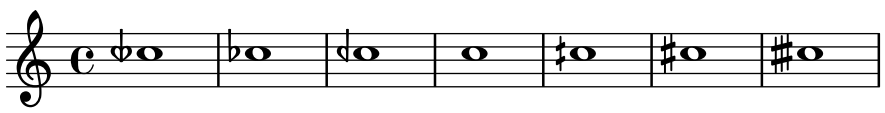
Una nota natural se introduce como el nombre de la nota, sin más; no se necesita ningún sufijo. Se impirime un símbolo de becuadro si es necesario para cancelar el efecto de una alteración o armadura anterior.

```
\relative c'' { a4 aes a2 }
```



Se pueden escribir medios bemoles y los medios sostenidos; a continuación presentamos una serie de DOs cada vez más agudos:

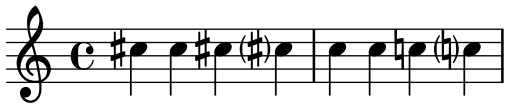
```
\relative c'' { ceseh1 ces ceh c cih cis cisih }
```



Normalmente las alteraciones accidentales se imprimen automáticamente, pero también puede imprimirlas manualmente. Un alteración recordatoria se puede forzar añadiendo un signo de

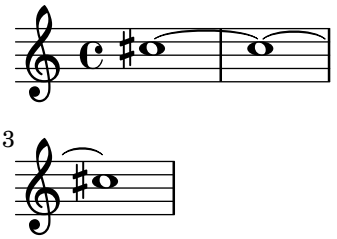
admiración ! después de la altura de la nota. Se puede obtener una alteración de precaución (o sea, una alteración entre paréntesis) añadiendo el signo de interrogación ? después del nombre de la nota.

```
\relative c'' { cis cis cis! cis? c c c! c? }
```



Las alteraciones sobre notas unidas por ligadura solo se imprimen al comienzo de un sistema:

```
\relative c'' {
  cis1~ 1~
  \break
  cis
}
```

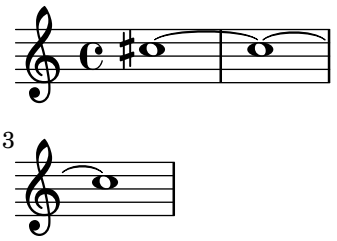


Fragmentos de código seleccionados

Ocultar las alteraciones sobre notas ligadas al principio de un sistema nuevo

Aquí se muestra la manera de ocultar las alteraciones de las notas ligadas al comienzo de un sistema nuevo.

```
\relative c'' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```

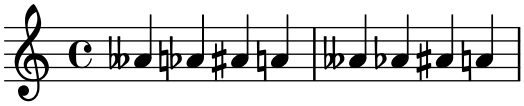


Evitar que se añadan becuadros adicionales automáticamente

Según las reglas tradicionales de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar un doble sostenido o un doble bemol anterior en la misma nota. Para modificar este comportamiento a la práctica actual, establezca el valor de la propiedad `extraNatural` a `#f` (falso) dentro del contexto de `Staff`.

```
\relative c'' {
  aeses4 aes ais a
```

```
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Véase también

Glosario musical: Sección “sharp” en *Glosario Musical*, Sección “flat” en *Glosario Musical*, Sección “double sharp” en *Glosario Musical*, Sección “double flat” en *Glosario Musical*, Sección “Pitch names” en *Glosario Musical*, Sección “quarter tone” en *Glosario Musical*.

Manual de aprendizaje: Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.3.5 [Alteraciones accidentales automáticas], página 32, Sección 17.3.8 [Alteraciones de anotación (*musica ficta*)], página 549, Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental_engraver” en *Referencia de Funcionamiento Interno*, Sección “Accidental” en *Referencia de Funcionamiento Interno*, Sección “AccidentalCautionary” en *Referencia de Funcionamiento Interno*, Sección “accidental-interface” en *Referencia de Funcionamiento Interno*.

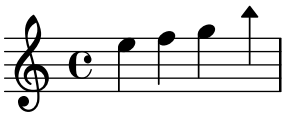
Advertencias y problemas conocidos

No existen estándares reconocidos ampliamente para denotar los bemoles de tres cuartos, de manera que los símbolos de LilyPond no se ajustan a ningún estándar.

1.1.4 Alturas aproximadas

Para escribir una nota cuando su altura no se conoce con exactitud, tal y como la nota más aguda que puede emitir un cantante, use la instrucción `\approximatePitch` con una altura que indica la altura aproximada. LilyPond imprime la nota a la altura dada con una cabeza de forma triangular y sin líneas adicionales.

```
\relative c'' {
  e4 f g \approximatePitch c
}
```



Véase también

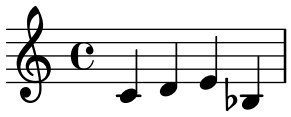
Referencia de funcionamiento interno: Sección “ApproximatePitchNoteHead” en *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” en *Referencia de Funcionamiento Interno*.

1.1.5 Nombres de las notas en otros idiomas

Existen conjuntos predefinidos de nombres de notas y sus alteraciones para algunos idiomas aparte del inglés. La selección del idioma de los nombres de las notas se suele hacer al principio del archivo; el ejemlo siguiente está escrito utilizando los nombres italianos de las notas:

```
\language "italiano"
```

```
\relative {
  do' re mi sib
}
```



Los idiomas disponibles y los nombres de las notas que definen son los siguientes:

Idioma	**Nombre de las notas**
nederlands	c d e f g a bes b
català o catalan	do re mi fa sol la sib si
deutsch	c d e f g a b h
english	c d e f g a bf/b-flat b
español o espanol	do re mi fa sol la sib si
français	do ré/re mi fa sol la sib si
italiano	do re mi fa sol la sib si
norsk	c d e f g a b h
português o portugues	do re mi fa sol la sib si
suomi	c d e f g a b h
svenska	c d e f g a b h
vlaams	do re mi fa sol la sib si

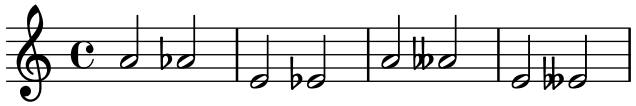
Además de los nombres de las notas, los sufijos de las alteraciones pueden también variar dependiendo del idioma:

Idioma	**sostenido**	**bemol**	**doble sostenido**	**doble bemol**
nederlands	is	es	isis	eses
català o catalan	d/s	b	dd/ss	bb
deutsch	is	es	isis	eses
english	s/-sharp	f/-flat	ss/x/-sharpsharp	ff/-flatflat
español o espanol	s	b	ss/x	bb
français	d	b	dd/x	bb
italiano	d	b	dd	bb
norsk	iss/is	ess/es	ississ/isis	essess/eses
português o portugues	s	b	ss	bb
suomi	is	es	isis	eses
svenska	iss	ess	ississ	essess
vlaams	k	b	kk	bb

En neerlandés, noruego y finlandés, aes se contrae como as, pero en neerlandés y en noruego las dos formas se aceptan en LilyPond. Exactamente lo mismo vale para es y ees, aeses y ases, y finalmente eeses y eses.

En alemán y en finlandés, LilyPond provee además la forma más frecuente asas para ases.

```
\relative c'' { a2 as e es a ases e eses }
```

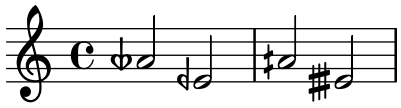


Algunas músicas utilizan microtonos cuyas alteraciones son fracciones de un sostenido o bemol 'normales'. La tabla siguiente relaciona los sufijos de los nombres de las notas para las alteraciones de un cuarto de tono; aquí, los prefijos *semi-* y *sesqui-* respectivamente significan 'medio' y 'uno y medio'.

Idioma	**semi-sostenido**	**semi-bemol**	**sesqui-sostenido**	**sesqui-bemol**
nederlands	ih	eh	isih	eseh
català o catalan	qd/qs	qb	tqd/tqs	tqb
deutsch	ih	eh	isih	eseh
english	qs	qf	tqs	tqf
español o espanol	cs	cb	tcs	tcb
français	sd	sb	dsd	bsb
italiano	sd	sb	dsd	bsb
norsk	ih	eh	issih/isih	esseh/eseh
português o portugues	sqt	bqt	stqt	btqt
suomi	ih	eh	isih	eseh
svenska	ih	eh	issih	esseh
vlaams	hk	hb	khk	bhb

En alemán existen contracciones similares del nombre para los microtonos como con las notas normales que se describen arriba.

```
\language "deutsch"

\relative c'' { asah2 eh aih eisih }
```



Casi todos los idiomas presentados aquí están asociados comúnmente con la música clásica occidental, también conocida como *Período de la práctica común*. Sin embargo, también están contempladas las alturas y los sistemas de afinación alternativos: véase Sección 18.1 [Notación común para músicas no occidentales], página 573.

Véase también

Glosario musical: Sección "Pitch names" en *Glosario Musical*, Sección "Common Practice Period" en *Glosario Musical*.

Referencia de la notación: Sección 18.1 [Notación común para músicas no occidentales], página 573.

Archivos instalados: `scm/define-note-names.scm`.

Fragmentos de código: Sección "Alturas" en *Fragmentos de código*.

1.2 Modificación de varias notas a la vez

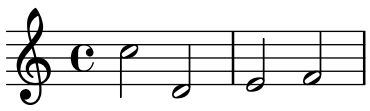
Esta sección trata de la manera de modificar las alturas.

1.2.1 Comprobación de octava

En el modo relativo, es fácil olvidar una marca de cambio de octava. Las comprobaciones de octava hacen más fácil encontrar estos errores, mediante la presentación de una advertencia y corrigiendo la octava si la nota se encuentra en una octava distinta de lo esperado.

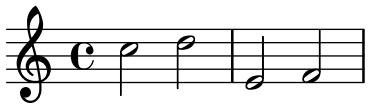
Para comprobar la octava de una nota, especifique la octava absoluta después del símbolo =. Este ejemplo genera un mensaje de advertencia (y corrige la altura) porque la segunda nota es la octava absoluta `d''` en lugar de `d'` como indica la corrección de octava.

```
\relative {
  c''2 d='
  e2 f
}
```



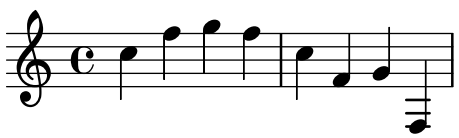
La octava de las notas se puede comprobar también con la instrucción `\octaveCheck` *`altura_de_control`*. *`altura_de_control`* se especifica en modo absoluto. Esto comprueba que el intervalo entre la nota anterior y la *`altura_de_control`* se encuentra dentro de una cuarta (es decir, el cálculo normal para el modo relativo). Si esta comprobación fracasa, se imprime un mensaje de advertencia. Aunque la nota previa no se modifica, las notas posteriores están en relación al valor corregido.

```
\relative {
  c''2 d
  \octaveCheck c'
  e2 f
}
```



Compare los dos compases siguientes. La primera y tercera comprobaciones de `\octaveCheck` fracasan, pero la segunda es correcta.

```
\relative {
  c''4 f g f

  c4
  \octaveCheck c'
  f
  \octaveCheck c'
  g
  \octaveCheck c'
  f
}
```



Véase también

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RelativeOctaveCheck” en *Referencia de Funcionamiento Interno*.

1.2.2 Transposición

Una expresión musical se puede transportar mediante `\transpose`. La sintaxis es

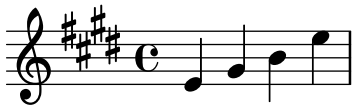
`\transpose` *`nota_origen nota_destino expresión_musical`*

Esto significa que la *`expresión_musical`* se transporta el intervalo que hay entre las notas *`nota_origen`* y *`nota_destino`*: cualquier nota con la altura de *`nota_origen`* se cambia por *`nota_destino`* y cualquier otra nota se transporta el mismo intervalo. Las dos notas se introducen en modo absoluto.

> **Nota:** La música que está dentro de un bloque `\transpose` es absoluta, a no ser que se incluya una instrucción `\relative` dentro del bloque.

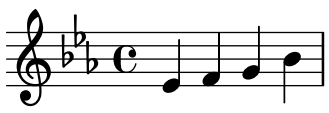
Por ejemplo, tomemos una pieza escrita en la tonalidad de Re mayor. Se puede transportar hacia arriba a Mi mayor; observe que también la armadura de la tonalidad se transporta automáticamente.

```
\transpose d e {
  \relative {
    \key d \major
    d'4 fis a d
  }
}
```



Si una particella escrita en Do (*afinación de concierto* normal) se debe tocar con un clarinete en La (para el que un La se escribe como un Do, y que suena una tercera menor por debajo de lo que está escrito), la particella correspondiente se produce mediante:

```
\transpose a c' {
  \relative {
    \key c \major
    c'4 d e g
  }
}
```

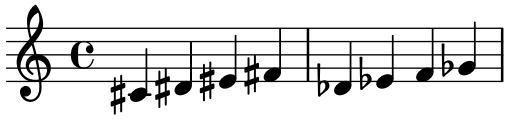


Observe que especificamos `\key c \major` de forma explícita. Si no especificamos ninguna tonalidad, las notas se transportan pero no se imprime la armadura.

`\transpose` distingue entre notas enarmónicas: tanto `\transpose c cis` como `\transpose c des` transportan un semitono hacia arriba. La primera versión imprime sostenidos y las notas no se mueven de su lugar en la escala, en cambio la segunda imprime bemoles de la nota siguiente.

```
music = \relative { c' d e f }
\new Staff {
```

```
  \transpose c cis { \music }
  \transpose c des { \music }
}
```



\transpose también se puede usar para introducir notas escritas para un instrumento transpositor. Los ejemplos anteriores muestran cómo escribir alturas en Do (o en *afinación de concierto*) y tipografiarlas para un instrumento transpositor, pero también es posible el caso contrario si, por ejemplo, tenemos un conjunto de partes instrumentales y quiere hacer un guión en Do para el director. Por ejemplo, al introducir música para trompeta en Si bemol que comienza por un Mi en la partitura (Re de concierto), se puede escribir:

```
musicaEnSiBemol = { e4 ... }
\transpose c bes, \musicaEnSiBemol
```

Para imprimir esta música en Fa (por ejemplo, al arreglarla para trompa) puede envolver la música existente con otro \transpose:

```
musicaEnSiBemol = { e4 ... }
\transpose f c' { \transpose c bes, \musicaEnSiBemol }
```

Para ver más información sobre instrumentos transpositores, consulte Sección 1.3.4 [Transposición de los instrumentos], página 30.

Fragmentos de código seleccionados

Transposing pitches with minimum accidentals ("smart" transpose)

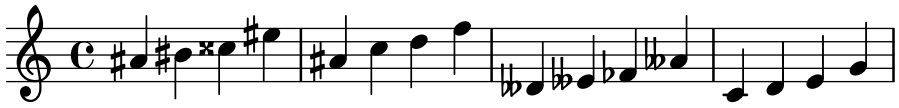
This example uses some Scheme code to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In this case, the following rules apply:

- double accidentals should be removed
- b sharp → c
- e sharp → f
- c flat → b
- f flat → e

In this manner, the most natural enharmonic notes are chosen.

```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         ;; `ly:pitch-alteration` returns quarter tone steps.
         (a (* 4 (ly:pitch-alteration p)))
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1)
            (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1)
            (or (eqv? n 0) (eqv? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
```

```
       ((> a 2)
        (set! a (- a 4))
        (set! n (+ n 1)))
       ((< a -2)
        (set! a (+ a 4))
        (set! n (- n 1))))
      (when (< n 0)
        (set! o (- o 1))
        (set! n (+ n 7)))
      (when (> n 6)
        (set! o (+ o 1))
        (set! n (- n 7)))
      (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (when (pair? es)
       (ly:music-set-property! music 'elements
                               (map naturalize es)))
     (when (ly:music? e)
       (ly:music-set-property! music 'element
                               (naturalize e)))
     (when (ly:pitch? p)
       (set! p (naturalize-pitch p))
       (ly:music-set-property! music 'pitch p))
     music))

naturalizeMusic =
#(define-music-function (m) (ly:music?)
   (naturalize m))

music = \relative c' { c4 d e g }

\new Staff {
  \transpose c ais { \music }
  \naturalizeMusic \transpose c ais { \music }
  \transpose c deses { \music }
  \naturalizeMusic \transpose c deses { \music }
}
```



Véase también

Referencia de la notación: Sección 1.3.4 [Transposición de los instrumentos], página 30, Sección 1.2.3 [Inversión], página 17, Sección 1.2.5 [Transformaciones modales], página 18, Sección 1.1.2 [Escritura de octava relativa], página 4, Sección 1.2.4 [Retrogradación], página 17.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TransposedMusic” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` ni `\relative` dentro de su argumento. Para usar el modo relativo dentro de música transportada, se debe colocar otro `\relative` dentro de `\transpose`.

No se imprimen alteraciones accidentales triples cuando se usa `\transpose`. En lugar de ello, se usará una nota ‘equivalente enarmónicamente’ (p.ej., Re bemol en lugar de Mi triple bemol).

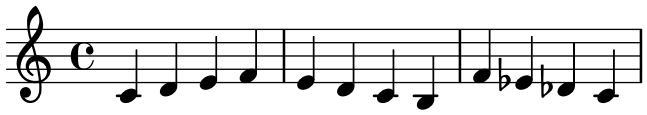
1.2.3 Inversión

Una expresión musical se puede invertir y transportar en una sola operación con:

```
\inversion nota-pivote nota-destino expresión_musical
```

La *expresión_musical* se invierte intervalo a intervalo alrededor de *nota-pivote*, y después se transporta de manera que *nota-pivote* se hace corresponder con *nota-destino*.

```
music = \relative { c' d e f }
\new Staff {
  \music
  \inversion d' d' \music
  \inversion d' ees' \music
}
```



> **Nota:** Los motivos a invertir deberían venir expresados en modo absoluto o convertirse previamente a la forma absoluta encerrándolos en un bloque `\relative`.

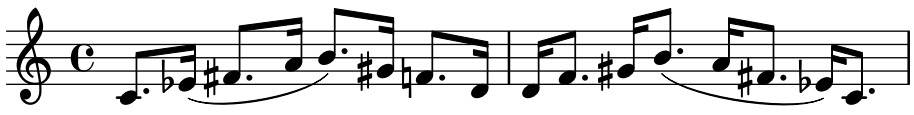
Véase también

Referencia de la notación: Sección 1.2.5 [Transformaciones modales], página 18, Sección 1.2.4 [Retrogradación], página 17, Sección 1.2.2 [Transposición], página 14.

1.2.4 Retrogradación

Se puede revertir una expresión musical para producir su retrogradación:

```
music = \relative { c'8. ees16( fis8. a16 b8.) gis16 f8. d16 }

\new Staff {
  \music
  \retrograde \music
}
```



Advertencias y problemas conocidos

`\retrograde` es una herramienta más bien sencilla. Puesto que muchos eventos se invierten como en un espejo en lugar de intercambiarse, los trucajes y los modificadores direccionales tienen que añadirse en los elementos de cierre correspondientes: `^(` debe terminar con `^)`, todos y cada uno de los `\<` o `\cresc` han de terminar en `\!` o en `\endcr`, y todo `\>` o `\decr` debe terminar en `\enddecr`. Las instrucciones o sobreescrituras que cambien propiedades y que tengan un efecto duradero, probablemente produzcan sorpresas.

Véase también

Referencia de la notación: Sección 1.2.3 [Inversión], página 17, Sección 1.2.5 [Transformaciones modales], página 18, Sección 1.2.2 [Transposición], página 14.

1.2.5 Transformaciones modales

En una composición musical que está basada en una escala, con frecuencia se transforman los motivos de diversas formas. Un motivo se puede *transportar* para que comience en distintos lugares de la escala o puede *invertirse* alrededor de una nota pivote de la escala. También se puede revertir para producir una *retrogradación*, véase Sección 1.2.4 [Retrogradación], página 17.

> **Nota:** Cualquier nota que no entre en la escada dada, se dejará sin transformar.

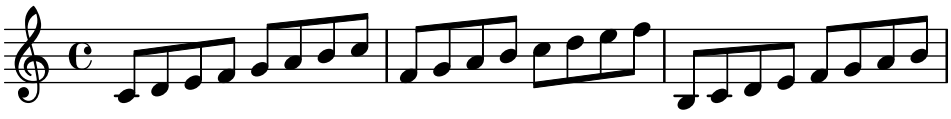
Transposición modal

Se puede transportar un motivo dentro de una escala dada con:

```
\modalTranspose nota-origen nota-destino escala motivo
```

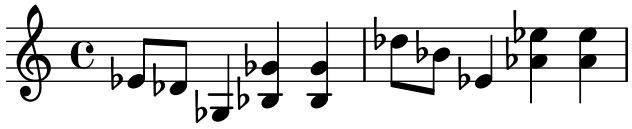
Las notas del *motivo* se desplazan dentro de la *escala* el número de grados de la escala dados por el intervalo entre *nota-destino* y *nota-origen*:

```
diatonicScale = \relative { c' d e f g a b }
motif = \relative { c'8 d e f g a b c }

\new Staff {
  \motif
  \modalTranspose c f \diatonicScale \motif
  \modalTranspose c b, \diatonicScale \motif
}
```



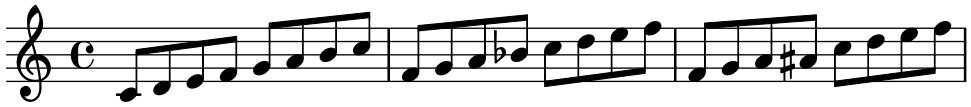
Se pueden especificar escalas ascendentes de cualquier longitud y con cualesquiera intervalos:

```
pentatonicScale = \relative { ges aes bes des ees }
motif = \relative { ees'8 des ges,4 <ges' bes,> <ges bes,> }

\new Staff {
  \motif
  \modalTranspose ges ees' \pentatonicScale \motif
}
```



Cuando se utiliza con una escala cromática, `\modalTranspose` tiene un efecto similar a `\transpose`, pero con la posibilidad de especificar los nombres de las notas que se quieren usar:

```
chromaticScale = \relative { c' cis d dis e f fis g gis a ais b }
motif = \relative { c'8 d e f g a b c }

\new Staff {
  \motif
  \transpose c f \motif
  \modalTranspose c f \chromaticScale \motif
}
```



Inversión modal

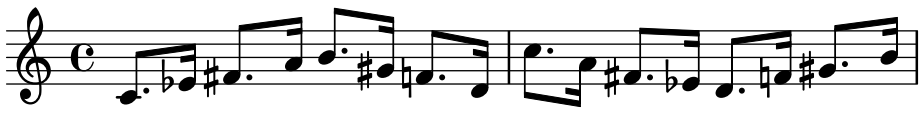
Se puede invertir un motivo dentro de una escala dada alrededor de una nota pivote dada y transportada al mismo tiempo en una única operación, con:

`\modalInversion` *nota-pivote nota-destino escala motivo*

Las notas del *motivo* se colocan al mismo número de grados de distancia de la escala a partir de la *nota-pivote* dentro de la *escala*, pero en la dirección opuestoa, y el resultado se desplaza después dentro de la *escala* el número de grados de la escala dados por el intervalo entre la *nota-destino* y la *nota-pivote*.

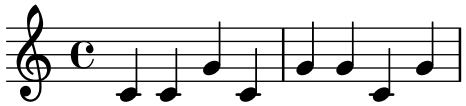
Así pues, para invertir sencillamente alrededor de una nota de la escala, utilice el mismo valor para *nota-pivote* y *nota-destino*:

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motif
  \modalInversion fis' fis' \octatonicScale \motif
}
```



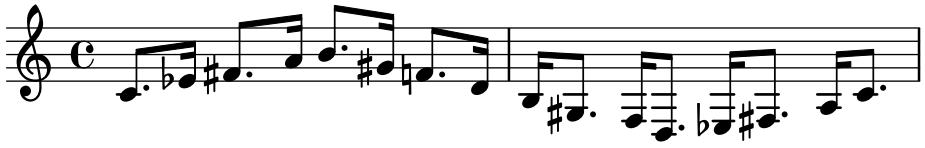
Para invertir alrededor de una nota pivote entre dos notas de la escala, invierta alrededor de una de las notas y después transporte en un grado de la escala. Las dos notas especificadas se pueden interpretar como que horquillan entre ellas a la nota pivote:

```
scale = \relative { c' g' }
motive = \relative { c' c g' c, }

\new Staff {
  \motive
  \modalInversion c' g' \scale \motive
}
```



La operación combinada de inversión y retrogradación produce la inversión retrógrada:

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motif
  \retrograde \modalInversion c' c' \octatonicScale \motif
}
```



Véase también

Referencia de la notación: Sección 1.2.3 [Inversión], página 17, Sección 1.2.4 [Retrogradación], página 17, Sección 1.2.2 [Transposición], página 14.

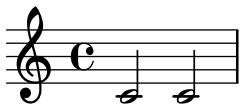
1.3 Imprimir las alturas

Esta sección trata de cómo alterar la presentación de la altura de las notas.

1.3.1 Clave

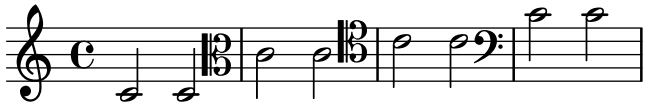
Sin ninugna instrucción explícita, la clave predeterminada en LilyPond es la clave “treble” (o clave de *Sol*).

```
c'2 c'
```



Sin embargo, se puede cambiar la clave usando la instrucción `\clef` y el nombre de la clave correspondiente. En los siguientes ejemplos se muestra la posición del *Do central* en distintas claves.

```
\clef treble
c'2 c'
\clef alto
c'2 c'
\clef tenor
c'2 c'
\clef bass
c'2 c'
```



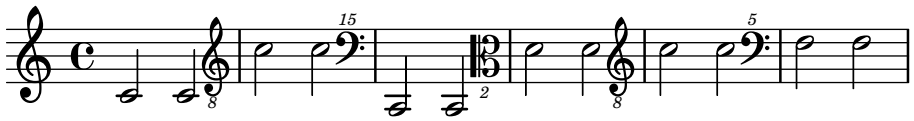
Para ver el repertorio completo de los posibles nombes para las cllaves, consulte Sección B.11 [Estilos de clave], página 928.

Las claves especiales como las utilizadas en la música *antigua*, se describen en Sección 17.3.2 [Claves de la música mensural], página 545, y en Sección 17.4.2 [Claves de canto gregoriano], página 553. La música que requiere claves de tablatura se estudia en Sección 12.1.3 [Tablaturas predeterminadas], página 436, y en Sección 12.1.4 [Tablaturas personalizadas], página 455.

Para emplear claves en las notas guía, véanse las instrucciones `\cueClef` y `\cueDuringWithClef` en Sección 6.3.3 [Formateo de las notas guía], página 272.

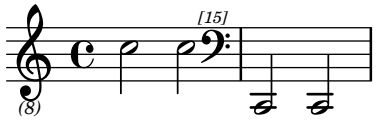
Al añadir `_8` o `^8` al nombre de la clave, la clave se transpone una octava hacia abajo o hacia arriba, respectivamente, y `_15` y `^15` la transpone dos octavas. Si es necesario se pueden usar otros números enteros. El argumento *nombre_de_clave* se debe encerrar entre comillas si contiene caracteres no alfabéticos:

```
\clef treble
c'2 c'
\clef "treble_8"
c'2 c'
\clef "bass^15"
c'2 c'
\clef "alto_2"
c'2 c'
\clef "G_8"
c'2 c'
\clef "F^5"
c'2 c'
```



Se puede obtener una indicación de octavación opcional encerrando el argumento numérico entre paréntesis o corchetes rectos:

```
\clef "treble_(8)"
c'2 c'
\clef "bass^[15]"
c'2 c'
```

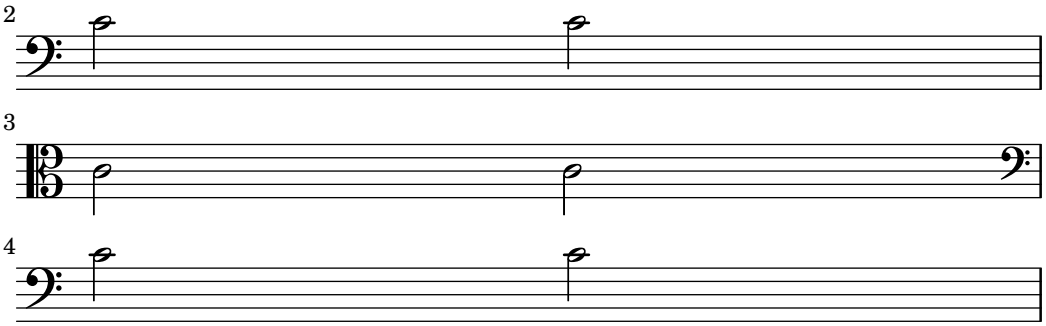


Las alturas se imprimen como si el argumento numérico no tuviera los paréntesis o corchetes.

De forma predeterminada, un cambio de clave que se produce en un salto de línea provoca que el símbolo de clave nuevo se imprima al final de la línea anterior, como una clave 'de cortesía', así como al principio del siguiente. Esta clave de cortesía se puede suprimir.

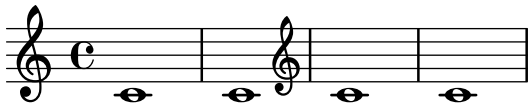
```
\clef treble c'2 c' \break
\clef bass c'2 c' \break
\set Staff.explicitClefVisibility = #end-of-line-invisible
\clef alto c'2 c' \break
\unset Staff.explicitClefVisibility
\clef bass c'2 c'
```





De forma predeterminada, una clave que se ha impreso previamente no se vuelve a imprimir si se emplea de nuevo la misma instrucción `\clef`, y se ignora. La instrucción `\set Staff.forceClef = ##t` modifica este comportamiento.

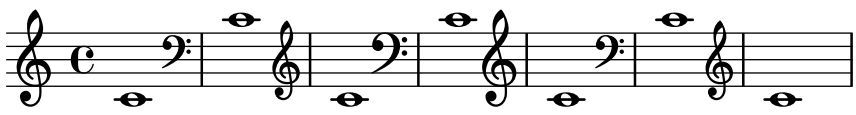
```
\clef treble
c'1
\clef treble
c'1
\set Staff.forceClef = ##t
c'1
\clef treble
c'1
```



Para ser más exactos, no es la propia instrucción `\clef` la que imprime una clave. Más bien, fija o cambia una propiedad del grabador de clave `Clef_engraver`, que a continuación decide por sí mismo si imprimir o no una clave en el pentagrama actual. La propiedad `forceClef` sobreescribe esta decisión localmente para reimprimir la clave una vez más.

Cuando hay un cambio de clave manual, el glifo de la clave modificada es pás pequeño de lo normal. Se puede sobreescribir este comportamiento.

```
\clef "treble"
c'1
\clef "bass"
c'1
\clef "treble"
c'1
\override Staff.Clef.full-size-change = ##t
\clef "bass"
c'1
\clef "treble"
c'1
\revert Staff.Clef.full-size-change
\clef "bass"
c'1
\clef "treble"
c'1
```



Fragmentos de código seleccionados

Trucaje de las propiedades de clave

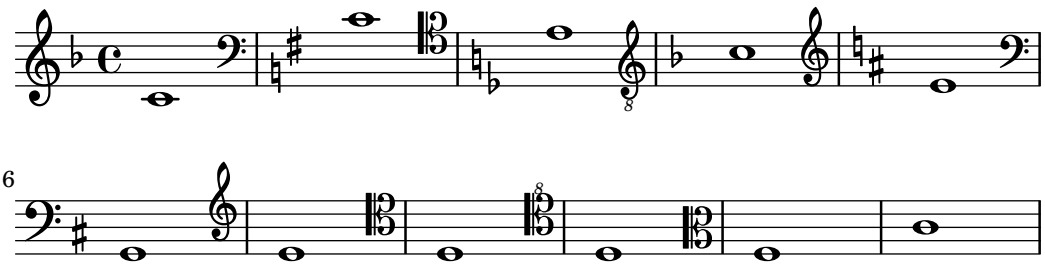
La modificación del glifo de la clave, su posición o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama. Para conseguir armaduras de tonalidad sobre las líneas del pentagrama adecuadas, también debe especificarse `middleCClefPosition`, con valores positivos o negativos que mueven el "Do central" hacia arriba o hacia abajo, respectivamente, en relación con la línea central del pentagrama (usualmente la tercera).

Por ejemplo, la instrucción `\clef "treble_8"` equivale a un ajuste de las propiedades de contexto `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave sobre el pentagrama), `middleCPosition` y `clefTransposition`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.

```
{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
  \set Staff.clefTransposition = 0
  \set Staff.middleCPosition = -4
  \set Staff.middleCClefPosition = -4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
```

```
  % and clefs.
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  c'1
  \set Staff.clefGlyph = "clefs.G"
  c'1
  \set Staff.clefGlyph = "clefs.C"
  c'1
  \set Staff.clefTransposition = 7
  c'1
  \set Staff.clefTransposition = 0
  \set Staff.clefPosition = 0
  c'1

  % Return to the normal clef.
  \set Staff.middleCPosition = 0
  c'1
}
```



Véase también

Referencia de la notación: Sección 17.3.2 [Claves de la música mensural], página 545, Sección 17.4.2 [Claves de canto gregoriano], página 553, Sección 12.1.3 [Tablaturas predeterminadas], página 436, Sección 12.1.4 [Tablaturas personalizadas], página 455, Sección 6.3.3 [Formateo de las notas guía], página 272.

Archivos instalados: `scm/parser-clef.scm`.

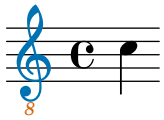
Fragmentos de código: Sección "Alturas" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Clef_engraver" en *Referencia de Funcionamiento Interno*, Sección "Clef" en *Referencia de Funcionamiento Interno*, Sección "ClefModifier" en *Referencia de Funcionamiento Interno*, Sección "clef-interface" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los números de octavación adjuntos a las claves se tratan como grobs distintos. Así pues, cualquier `\override` (sobreescritura) efectuada al objeto *Clef* deberá aplicarse también al grob *ClefModifier* como un `\override` diferente.

```
\new Staff \with {
  \override Clef.color = #(universal-color 'blue)
  \override ClefModifier.color = #(universal-color 'vermillion)
}

\clef "treble_8" c'4
```



1.3.2 Armadura de la tonalidad

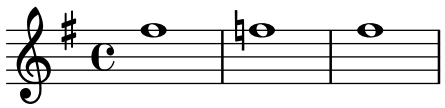
> **Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas son el código de entrada en bruto; la armadura y la clave determinan de qué forma se presenta este código en bruto. Una nota sin alteración como `c` significa 'Do natural', si que le afecten la armadura ni la clave. Para ver más información, consulte Sección "Alturas y armaduras" en *Manual de Aprendizaje*.

La armadura indica la tonalidad en que se toca una pieza. Está denotada por un conjunto de alteraciones (bemoles o sostenidos) al comienzo del pentagrama. El establecimiento o modificación de la armadura se hace con la instrucción `\key`:

```
\key nota modo
```

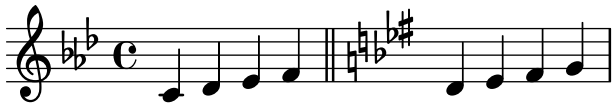
Aquí, *modo* debe ser `\major` o `\minor` para obtener la tonalidad *nota* mayor o *nota* menor, respectivamente. También puede usar los nombres estándar de modo (también conocidos como *modos eclesiásticos*): `\ionian` (jónico), `\dorian` (dórico), `\phrygian` (frigio), `\lydian` (lidio), `\mixolydian` (mixolidio), `\aeolian` (eolio) y `\locrian` (locrio).

```
\relative {
  \key g \major
  fis''1
  f
  fis
}
```



Se pueden definir modos adicionales, escribiendo una lista con la alteración que lleva cada nota de la escala cuando el modo comienza en Do.

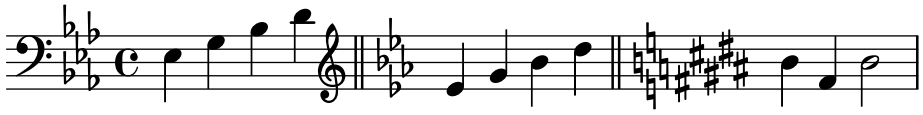
```
freygish = #`((0 . ,NATURAL) (1 . ,FLAT) (2 . ,NATURAL)
    (3 . ,NATURAL) (4 . ,NATURAL) (5 . ,FLAT) (6 . ,FLAT))

\relative {
  \key c \freygish c'4 des e f
  \bar "||" \key d \freygish d es fis g
}
```



Las alteraciones de la armadura de tonalidad se pueden imprimir en distinta octava que en sus posiciones tradicionales, o en más de una octava, usando las propiedades `flat-positions` y `sharp-positions` de `KeySignature`. Las entradas en estas propiedades especifican el rango de posiciones de pentagrama en que se imprimirán las alteraciones. Si en una entrada se especifica

una sola posición, las alteraciones se colocan dentro de la octava que termina en dicha posición del pentagrama.

```
\override Staff.KeySignature.flat-positions = #'((-5 . 5))
\override Staff.KeyCancellation.flat-positions = #'((-5 . 5))
\clef bass \key es \major es g bes d'
\clef treble \bar "||" \key es \major es' g' bes' d''

\override Staff.KeySignature.sharp-positions = #'(2)
\bar "||" \key b \major b' fis' b'2
```

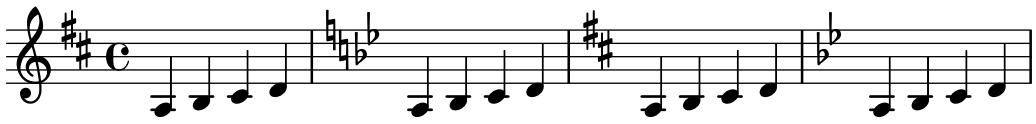


Fragmentos de código seleccionados

Evitar que se impriman becuadros cuando cambia la armadura

Cuando cambia la armadura de la tonalidad, se imprimen becuadros automáticamente para cancelar las alteraciones de las armaduras anteriores. Esto se puede evitar estableciendo la propiedad `printKeyCancellation` al valor "#f" dentro del contexto `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keyAlterations` dentro del contexto `Staff`.

Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

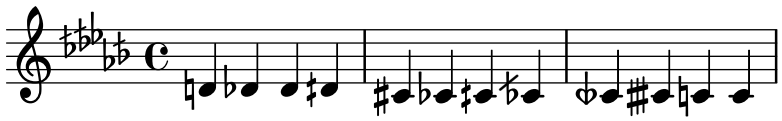
```
\set Staff.keyAlterations =
  #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)
```

donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es `SHARP`, `FLAT` o `DOUBLE-SHARP`, etc., precedido de una coma.

De forma alternativa, podemos usar el formato más conciso (*paso . alteración*) para cada elemento de la lista si se usa la misma alteración en todas las octavas.

Para escalas microtonales en las que un "sostenido" no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Véase también

Glosario musical: Sección "church mode" en *Glosario Musical*, Sección "scordatura" en *Glosario Musical*.

Manual de aprendizaje: Sección "Alturas y armaduras" en *Manual de Aprendizaje*.

Fragmentos de código: Sección "Alturas" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "KeyChangeEvent" en *Referencia de Funcionamiento Interno*, Sección "Key_engraver" en *Referencia de Funcionamiento Interno*, Sección "Key_performer" en *Referencia de Funcionamiento Interno*, Sección "KeyCancellation" en *Referencia de Funcionamiento Interno*, Sección "KeySignature" en *Referencia de Funcionamiento Interno*, Sección "key-signature-interface" en *Referencia de Funcionamiento Interno*.

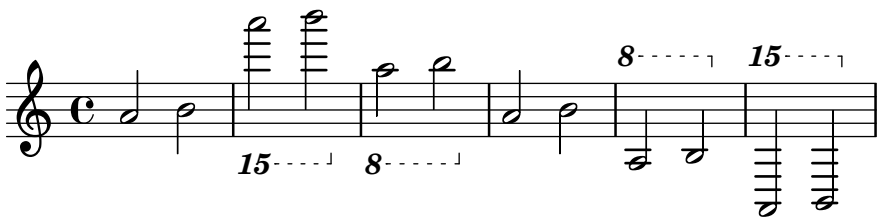
1.3.3 Corchetes de octava

Los *corchetes de Ottava* bajan o suben las notas mostradas en una o más octavas, sin alterar la altura del sonido. El propósito de la octavación es reducir el uso de líneas adicionales, especialmente cuando se producen intervalos amplios en pasajes breves. La instrucción `\ottava` lleva un número entero positivo para indicar que se toda más agudo de lo que está impreso, o un número negativo para tocar más bajo. El efecto de los corchetes de ottava dura hasta la entrada siguiente de una octavación distinta, y se puede dar por finalizada usando `\ottava 0`.

En el ejemplo siguiente, todas las notas suenan a la misma altura:

```
\relative c'' {
  a2 b
  \ottava -2
  a2 b
  \ottava -1
  a2 b
  \ottava 0
  a2 b
  \ottava 1
  a2 b
  \ottava 2
```

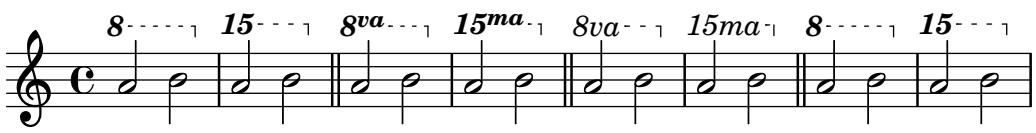
```
  a2 b
}
```



De forma predeterminada, solo se imprime un número al comienzo del corchete. Este ajuste se puede modificar para que incluya un ordinal abreviado, ya sea como superíndice o en letras de tipo normal; la tipografía inicial en negrita de estos caracteres se puede también alterar, como se explica en Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327.

El ejemplo siguiente muestra diversas opciones, así como la forma de volver al comportamiento predeterminado actualmente:

```
\relative c'' {
  \ottava 1
  a'2 b
  \ottava 2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-ordinals
  \ottava 1
  a,2 b
  \ottava 2
  a'2 b
  \bar "||"
  \override Staff.OttavaBracket.font-series = #'normal
  \set Staff.ottavationMarkups = #ottavation-simple-ordinals
  \ottava 1
  a,2 b
  \ottava 2
  a'2 b
  \bar "||"
  \revert Staff.OttavaBracket.font-series
  \set Staff.ottavationMarkups = #ottavation-numbers
  \ottava 1
  a,2 b
  \ottava 2
  a'2 b
}
```



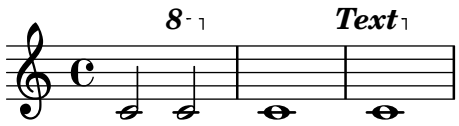
Fragmentos de código seleccionados

Cambiar el texto de las indicaciones de octava alta y baja

Internamente, `\ottava` fija las propiedades `ottavation` (por ejemplo para `8va` o `8vb`) y `middleCPosition`. Para sobreescribir el texto del corchete, fije `ottavation` después de invocar a `\ottava`.

Un texto breve es especialmente útil cuando se usa una indicación corta de octava.

```
{
  c'2
  \ottava 1
  \set Staff.ottavation = "8"
  c''2
  \ottava 0
  c'1
  \ottava 1
  \set Staff.ottavation = "Text"
  c''1
}
```

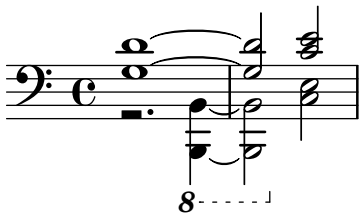


Añadir una indicación de octava alta a una sola voz

Si tiene más de una voz en el mismo pentagrama, el cambio de octavación de una voz transportará la posición de las notas en todas las voces mientras dure el corchete de octava. Si la octavación se quiere aplicar a una voz solamente, hay que trasladar el grabador `Ottava_spanner_engraver` al contexto `Voice`.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
    {
      r2.
      \ottava -1
      <b,,, b,,>4 ~ |
      q2
      \ottava 0
      <c e>2
    }
```

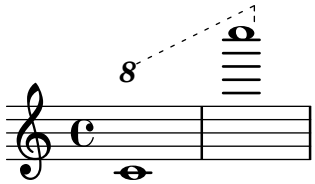
```
  >>
}
```



Modificación de la inclinación de la línea de octava

Es posible cambiar la inclinación de la línea de octava alta o baja.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5.0) ; Change the number here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
     #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c'''1
}
```



Véase también

Glosario musical: Sección “octavation” en *Glosario Musical*.

Referencia de la notación: Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ottava_spanner_engraver” en *Referencia de Funcionamiento Interno*, Sección “OttavaBracket” en *Referencia de Funcionamiento Interno*, Sección “ottava-bracket-interface” en *Referencia de Funcionamiento Interno*.

1.3.4 Transposición de los instrumentos

Al tipografiar partituras donde participan instrumentos transpositores, ciertas partes se pueden tipografiar en un tono distinto del *tono de concierto*. En estos casos, se debe especificar la tonalidad del *instrumento transpositor*; de otro modo, la salida MIDI y las notas guía en otras

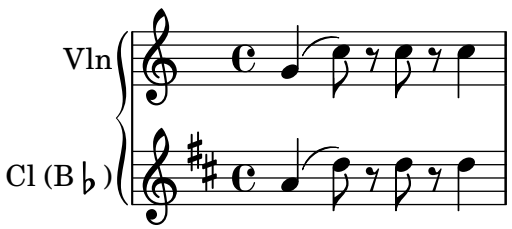
partes producirían alturas incorrectas. Para ver más información sobre partes citadas como guía, consulte Sección 6.3.2 [Citar otras voces], página 268.

`\transposition` *pitch*

El tono usado para `\transposition` debe corresponderse con el sonido real que se oye cuando el instrumento transpositor interpreta un Do central `c'` escrito en el pentagrama. Esta nota se escribe en altura absoluta, por tanto un instrumento que produce un sonido real un tono más agudo que la música impresa (un instrumento en Re) debe usar `\transposition d'`. La instrucción `\transposition` se debe usar *solamente* si las notas *no* se van a escribir en afinación de concierto.

A continuación pueden verse algunas notas para violín y para clarinete en Si bemol, donde las partes se han introducido usando las notas y la armadura tal y como aparecen en la partitura del director. Lo que tocan los dos instrumentos está sonando al unísono.

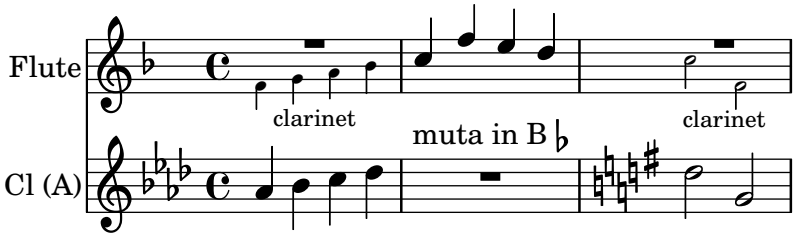
```
\new GrandStaff <<
  \new Staff = "violin" \with {
    instrumentName = "Vln"
    midiInstrument = "violin"
  }
  \relative c'' {
    % not strictly necessary, but a good reminder
    \transposition c'
    \key c \major
    g4( c8) r c r c4
  }
  \new Staff = "clarinet" \with {
    instrumentName = \markup { Cl (B\flat) }
    midiInstrument = "clarinet"
  }
  \relative c'' {
    \transposition bes
    \key d \major
    a4( d8) r d r d4
  }
>>
```



La `\transposition` se puede cambiar durante la pieza. Por ejemplo, un clarinetista puede cambiar del clarinete en La al clarinete en Si bemol.

```
flute = \relative c'' {
  \key f \major
  \cueDuring "clarinet" #DOWN {
    R1 _\markup\tiny "clarinet"
    c4 f e d
    R1 _\markup\tiny "clarinet"
  }
}
```

```
clarinet = \relative c'' {
  \key aes \major
  \transposition a
  aes4 bes c des
  R1^\markup { muta in B\flat }
  \key g \major
  \transposition bes
  d2 g,
}
\addQuote "clarinet" \clarinet
<<
  \new Staff \with { instrumentName = "Flute" }
    \flute
  \new Staff \with { instrumentName = "Cl (A)" }
    \clarinet
>>
```



Véase también

Glosario musical: Sección “concert pitch” en *Glosario Musical*, Sección “transposing instrument” en *Glosario Musical*.

Referencia de la notación: Sección 6.3.2 [Citar otras voces], página 268, Sección 1.2.2 [Transposición], página 14.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

1.3.5 Alteraciones accidentales automáticas

Existen muchas convenciones distintas sobre la forma de tipografiar las alteraciones. LilyPond proporciona una función para especificar qué estilo de alteraciones usar. Esta función se invoca como sigue:

```
\new Staff <<
  \accidentalStyle voice
  { ... }
>>
```

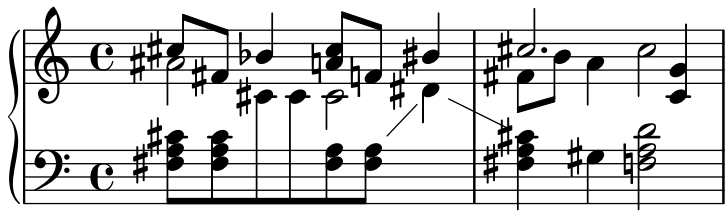
El estilo de alteraciones se aplica normalmente al `Staff` en curso (con la excepción de los estilos `choral`, `piano` y `piano-cautionary`, que se explican más adelante). Opcionalmente, la función puede tomar un segundo argumento que determina en qué ámbito se debe cambiar el estilo. Por ejemplo, para usar el mismo estilo en todos los pentagramas del `StaffGroup` en curso, use

```
\accidentalStyle StaffGroup.voice
```

Están contemplados los siguientes estilos de alteración. Para dar una muestra de cada uno de los estilos, utilizamos el ejemplo siguiente:

```
musicA = {
  <<
    \relative {
```

```
      cis''8 fis, bes4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
    \\
    \relative {
      ais'2 cis, |
      fis8 b a4 cis2 |
    }
  >>
}

musicB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative {
      <fis a cis>8[ <fis a cis>
      \change Staff = up
      cis' cis
      \change Staff = down
      <fis, a> <fis a>]
      \showStaffSwitch
      \change Staff = up
      dis'4 |
      \change Staff = down
      <fis, a cis>4 gis <f a d>2 |
    }
  }
}

\new PianoStaff {
  <<
    \new Staff = "up" {
      \accidentalStyle default
      \musicA
    }
    \new Staff = "down" {
      \accidentalStyle default
      \musicB
    }
  >>
}
```



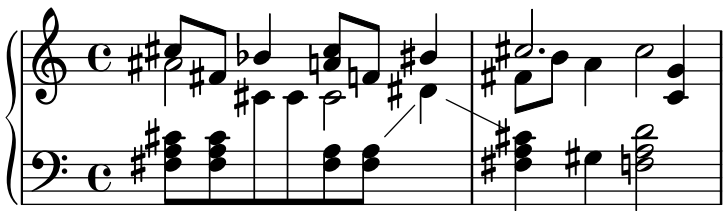
Observe que las últimas líneas de este ejemplo se pueden sustituir por las siguientes, siempre y cuando queramos usar el mismo estilo en los dos pentagramas.

```
\new PianoStaff {
```

```
    <<
      \new Staff = "up" {
        %%% cambie la línea siguiente como desee:
        \accidentalStyle Score.default
        \musicA
      }
      \new Staff = "down" {
        \musicB
      }
    >>
  }
```

`default (predeterminado)`

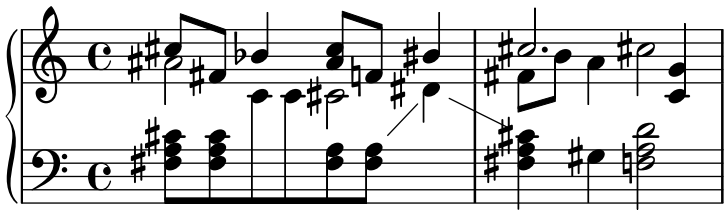
Es el comportamiento de composición tipográfica predeterminado. Corresponde a la práctica común del s. XVIII: las alteraciones accidentales se recuerdan hasta el final del compás en el que aparecen y solo en la misma octava. Así, en el ejemplo siguiente, no se imprimen becuadros antes del Si natural en el segundo compás ni en el último Do:



`voice (voz)`

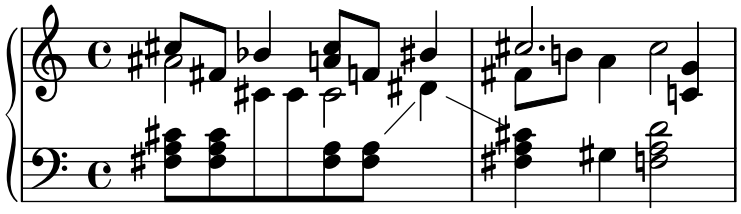
El comportamiento normal es recordar las alteraciones accidentales al nivel de `Staff`. En este estilo, sin embargo, se tipografían las alteraciones individualmente para cada voz. Aparte de esto, la regla es similar a `default`.

Como resultado, las alteraciones de una voz no se cancelan en las otras voces, lo que con frecuencia lleva a un resultado no deseado: en el ejemplo siguiente, es difícil determinar si el segundo La se debe tocar natural o sostenido. Por tanto, la opción `voice` se debe usar solo si las voces se van a leer individualmente por músicos distintos. Si el pentagrama va a utilizarse por parte de un solo músico (p.ej., un director, o en una partitura de piano), entonces se deben usar en su lugar los estilos `modern` o `modern-cautionary`.



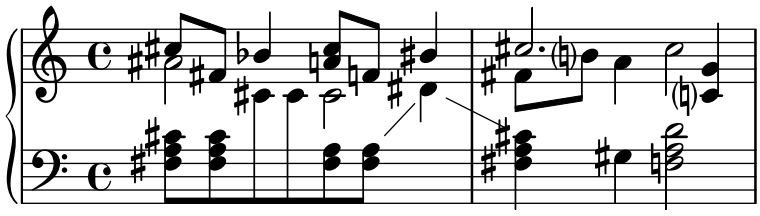
`modern (moderno)`

Esta regla corresponde a la práctica común del s. XX. Omite algunos becuadros adicionales, que tradicionalmente se imprimían precediendo a un sostenido que sigue a un doble sostenido, o a un bemol que sigue a un doble bemol. La regla `modern` imprime las mismas alteraciones que el estilo `default`, con dos adiciones que sirven para evitar la ambigüedad: después de alteraciones temporales se imprimen indicaciones de cancelación también en el compás siguiente (para notas en la misma octava) y, en el mismo compás, para notas en octavas distintas. De aquí los becuadros antes del Si natural y del Do en el segundo compás del pentagrama superior:



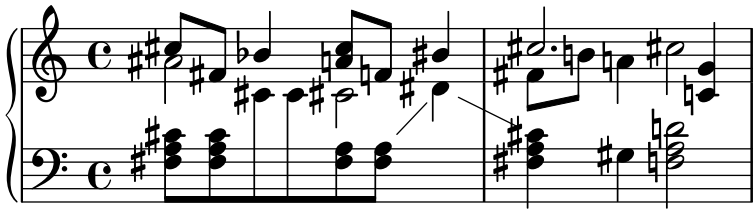
`modern-cautionary` (`moderno de precaución`)

Esta regla es similar a `modern`, pero las alteraciones 'añadidas' se imprimen como alteraciones de precaución (entre paréntesis). También se pueden imprimir en un tamaño distinto sobreescribiendo la propiedad `font-size` del objeto `AccidentalCautionary`.



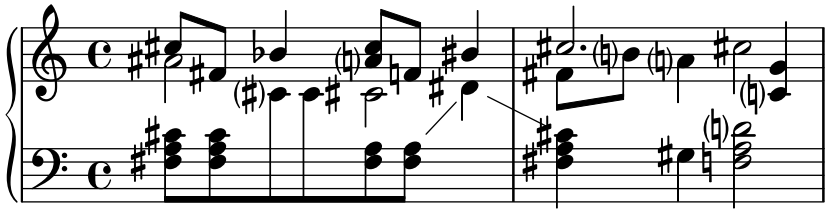
`modern-voice` (`moderno, para voces`)

Esta regla se usa para que puedan leer las alteraciones en varias voces, tanto músicos que tocan una voz como músicos que tocan todas las voces. Se imprimen las alteraciones para cada voz, pero *se cancelan* entre voces dentro del mismo `Staff`. Por tanto, el La en el último compás se cancela porque la cancelación anterior estaba en una voz distinta, y el Re en el pentagrama inferior se cancela a causa de la alteración en otra voz en el compás previo:



`modern-voice-cautionary` (`moderno, voz, de precaución`)

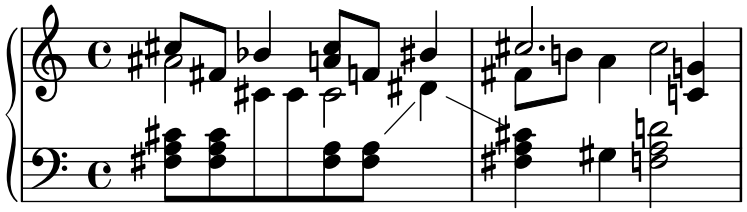
Esta regla es la misma que `modern-voice`, pero con las alteraciones añadidas (las que el estilo `voice` no imprime) compuestas como de precaución. Incluso aunque todas las alteraciones impresas por el estilo `default` *son* impresas con esta regla, algunas de ellas se tipografían como de precaución.



`piano`

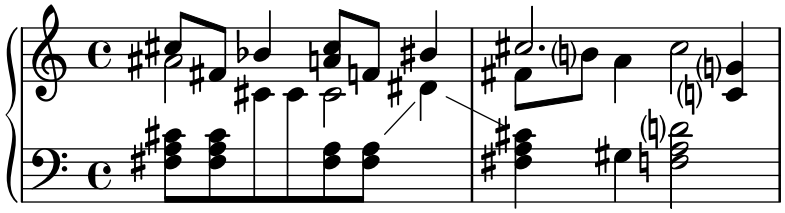
Esta regla refleja la práctica del s.XX para la notación de piano. Su comportamiento es muy similar al estilo `modern`, pero aquí las alteraciones también se cancelan entre distintos pentagramas del mismo grupo `GrandStaff` o `PianoStaff`, de ahí todas las cancelaciones de las últimas notas.

Este estilo de alteración se aplica al grupo `GrandStaff` o `PianoStaff` en curso, a no ser que se encuentre caracterizado con un segundo argumento.



`piano-cautionary` (`piano`, de precaución)

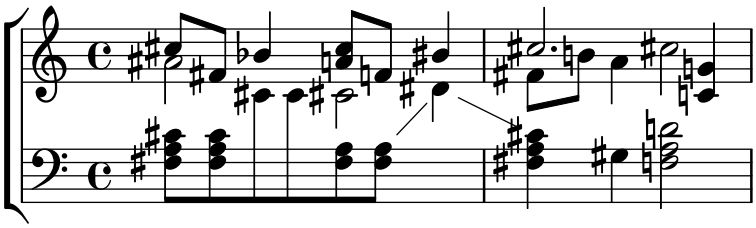
Igual que `piano` pero con las alteraciones añadidas compuestas como de precaución.



`choral`

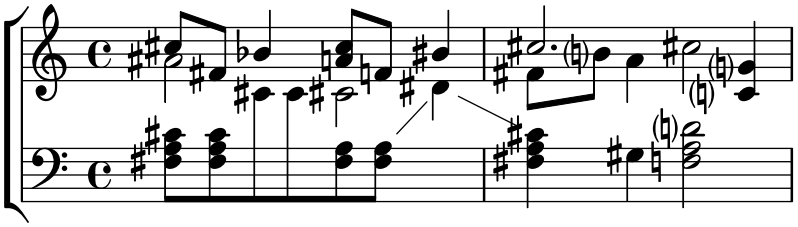
Esta regla es una combinación de los estilos `modern-voice` y `piano`. Muestra todas las alteraciones accidentales requeridas para los cantantes que siguen solo su propia voz, así como las alteraciones adicionales necesarias para los lectores que siguen simultáneamente todas las voces de un `ChoirStaff` completo.

Este estilo de alteraciones se aplica al contexto `ChoirStaff` actual, a no ser que se encuentre caracterizado con un segundo argumento.



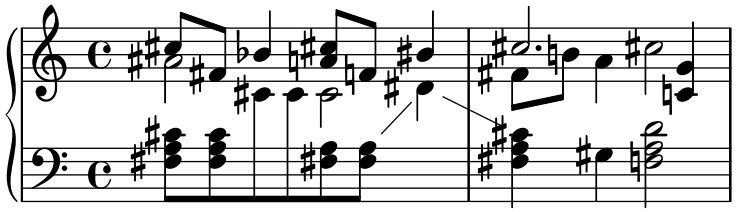
`choral-cautionary`

Es lo mismo que `choral` pero con las alteraciones adicionales impresas como de precaución (o cortesía).

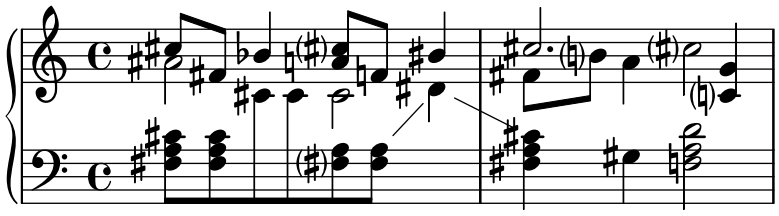


`neo-modern`

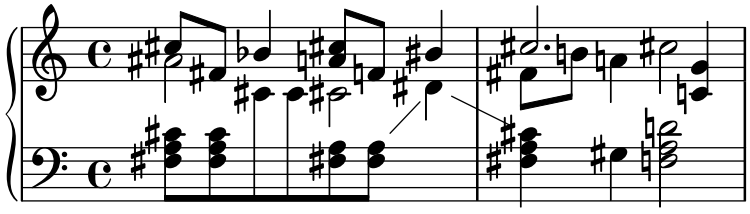
Esta regla reproduce una práctica común en la música contemporánea: las alteraciones accidentales se imprimen como en `modern`, pero se vuelven a imprimir si aparece la misma nota otra vez en el mismo compás (excepto si la nota se repite inmediatamente).



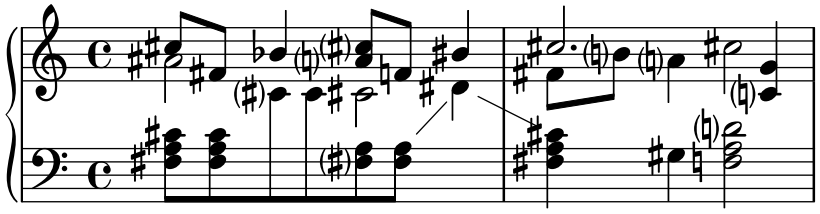
`neo-modern-cautionary`
: Esta regla es similar a `neo-modern`, pero las alteraciones 'adicionales' se imprimen como alteraciones de precaución (con paréntesis). También se pueden imprimir en un tamaño distinto sobreescribiendo la propiedad `font-size` del objeto `AccidentalCautionary`.



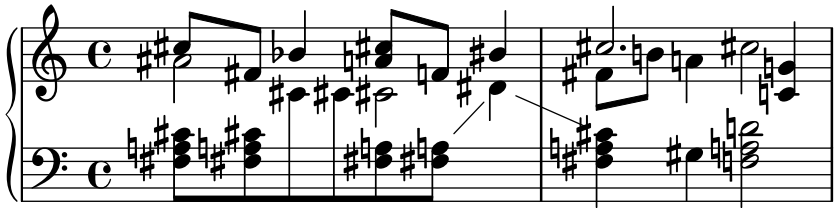
`neo-modern-voice`
: Esta regla se usa para alteraciones accidentales sobre varias voces que se han de leer por parte de músicos que tocan una voz, así como por músicos que tocan todas las voces. Las alteraciones se imprimen para cada voz como con `neo-modern`, pero se cancelan para otras voces que están en el mismo pentagrama `Staff`.



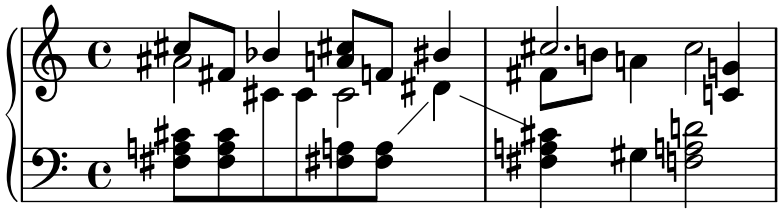
`neo-modern-voice-cautionary`
: Esta regla es similar a `neo-modern-voice`, pero las alteraciones adicionales se imprimen como alteraciones de precaución.



`dodecaphonic`
: Esta regla refleja una práctica introducida por los compositores de principios del s.XX, en un intento de abolir la jerarquía entre notas naturales y alteradas. Con este estilo, *todas* las notas llevan alteración, incluso becuadros.

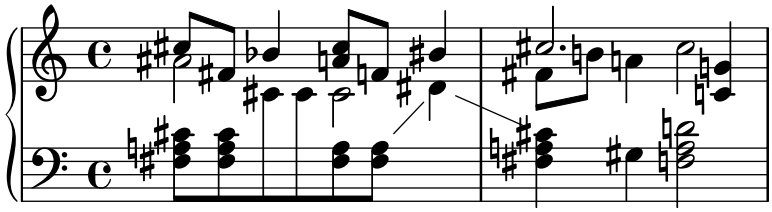


`dodecaphonic-no-repeat`
: Como con el estilo de alteraciones dodecafónico *todas* las notas llevan una alteración de forma predeterminada, pero las alteriaciones se suprimen cuando hay notas repetidas inmediatamente en el mismo pentagrama.



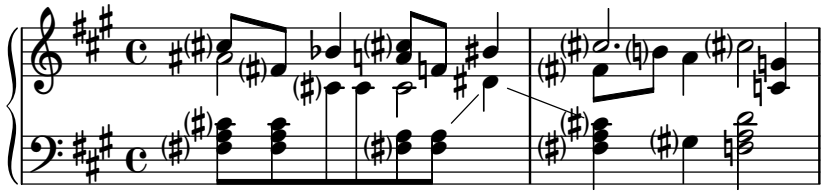
`dodecaphonic-first`

Similar al estilo de alteraciones dodecafónico, *todas* las notas llevan una alteración, pero solo la primera vez que se encuentran en el compás. Las alteraciones se recuerdan solamente para la octava actual pero a través de las voces.



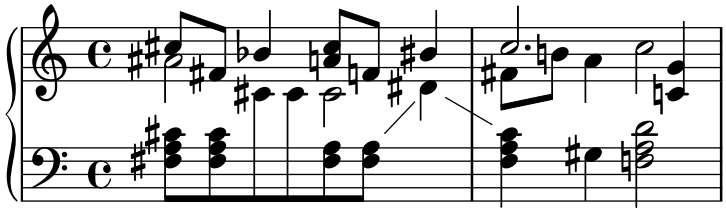
`teaching (enseñanza)`

Esta regla está pensada para estudiantes, y hace más sencillo crear hojas de escalas con alteraciones de precaución creadas automáticamente. Las alteraciones se imprimen como en el estilo `modern`, pero se añaden alteraciones de precaución para todas las notas sostenidas o bemoles especificadas por la armadura, excepto si la nota se repite inmediatamente.



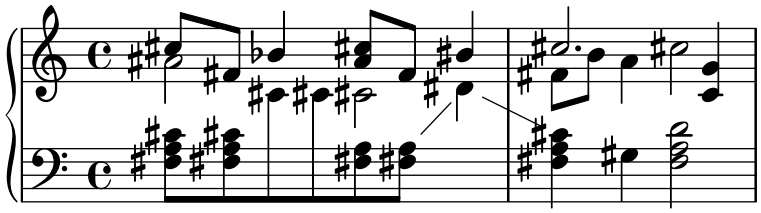
`no-reset (no restablecer)`

Es el mismo que `default` pero con alteraciones que duran 'para siempre' y no solo dentro del mismo compás:



`forget (olvidar)`

Es lo opuesto a `no-reset`: Las alteraciones no se recuerdan en absoluto: de aquí que todas las alteraciones se compongan tipográficamente en relación a la armadura de la tonalidad, sin que estén afectadas por lo que viene antes.



Véase también

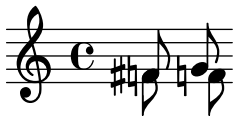
Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental” en *Referencia de Funcionamiento Interno*, Sección “Accidental_engraver” en *Referencia de Funcionamiento Interno*, Sección “GrandStaff” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” en *Referencia de Funcionamiento Interno*, Sección “AccidentalPlacement” en *Referencia de Funcionamiento Interno*, Sección “accidental-suggestion-interface” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

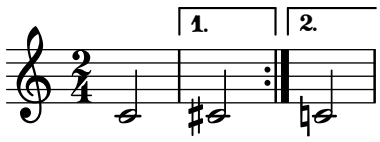
Las notas simultáneas no se tienen en cuenta para la determinación automática de las alteraciones accidentales; solo se tienen en cuenta las notas anteriores y la armadura de la tonalidad. Puede ser necesario forzar las alteraciones accidentales con ‘`!`’ o ‘`?`’ cuando la misma nota, con el mismo nombre, ocurre simultáneamente con distintas alteraciones, como en `<f! fis!>`.

```
\relative c' <<
  { fis8 g } \\
  { f! f }
>>
```



Una solución más sofisticada aparece en LSR snippet 1172 (`https://lsr.di.unimi.it/LSR/Item?id=1172`).

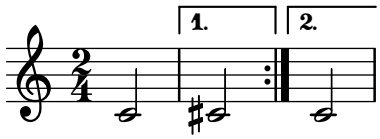
En los finales alternativos de primera y segunda vez, la cancelación de precaución de las alteraciones debería basarse en el compás previo *que se ha tocado*, pero está basado en el compás previo *que se ha impreso*. En el ejemplo siguiente, el Do natural de la casilla de segunda vez no necesita becuadro:



Se puede usar el siguiente rodeo del problema: definir una función que cambie localmente el estilo de alteraciones a `forget`:

```
forget = #(define-music-function (music) (ly:music?) #{
  \accidentalStyle forget
  #music
  \accidentalStyle modern
#})
{
  \accidentalStyle modern
  \time 2/4
  \repeat volta 2 {
    c'2
  }
  \alternative {
     \volta 1 { cis' }
```

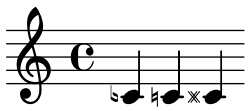
```
      \volta 2 { \forget c' }
  }
}
```



1.3.6 Glifos de alteración alternativos

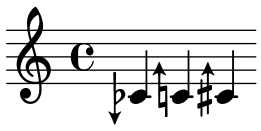
Los sistemas de notación de la música antigua y no occidentales tienen sus propios sistemas de alteraciones accidentales. Los glifos se controla a través de la propiedad `alterationGlyphs` del contexto de `Staff` y otros contextos de tipo similar. Los valores predefinidos para esta propiedad están listados en Sección B.10 [Conjuntos de glifos de alteraciones accidentales], página 927.

```
\layout {
  \context {
    \Staff
    alterationGlyphs = #alteration-vaticana-glyph-name-alist
  }
}

{ ces' c' cis' }
```



La propiedad también se puede establecer a una lista asociativa personalizada que asocia cada alteración a un nombre de glifo. Las alteraciones se dan como fracciones de tono. La lista de todos los glifos está en [Accidental glyphs], página 914.

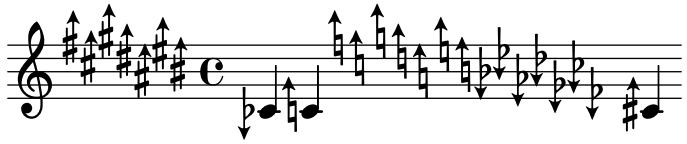
```
\layout {
  \context {
    \Staff
    alterationGlyphs =
      #'((-1/2 . "accidentals.flat.arrowdown")
         (0 . "accidentals.natural.arrowup")
         (1/2 . "accidentals.sharp.arrowup"))
  }
}

{ ces' c' cis' }
```



La propiedad `padding-pairs` de los objetos `KeySignature` y `KeyCancellation` es una lista asociativa que asocia pares de glifos al relleno de separación que se debe añadir entre estos glifos en las armaduras de tonalidad.

```
\layout {
  \context {
    \Staff
```

```
    alterationGlyphs =
      #'((-1/2 . "accidentals.flat.arrowdown")
         (0 . "accidentals.natural.arrowup")
         (1/2 . "accidentals.sharp.arrowup"))
    \override KeySignature.padding-pairs =
      #'((("accidentals.sharp.arrowup" . "accidentals.sharp.arrowup")
            . 0.25)
         (("accidentals.flat.arrowdown" . "accidentals.flat.arrowdown")
            . 0.3))
    \override KeyCancellation.padding-pairs =
      #'((("accidentals.natural.arrowup" . "accidentals.natural.arrowup")
            . 0.7))
  }
}

{
  \key cis \major
  ces' c'
  \key ces \major
  cis'
}
```



Véase también

Referencia de la notación: Sección B.10 [Conjuntos de glifos de alteraciones accidentales], página 927, [Accidental glyphs], página 914.

Referencia de funcionamiento interno: Sección “accidental-switch-interface” en *Referencia de Funcionamiento Interno*, Sección “Alteration_glyph_engraver” en *Referencia de Funcionamiento Interno*, Sección “key-signature-interface” en *Referencia de Funcionamiento Interno*.

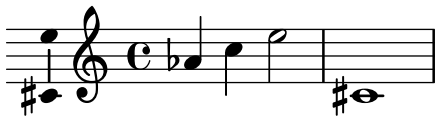
1.3.7 Tesitura

El término *ambitus* o ámbito, denota el rango de notas que abarca una voz dada en una parte musical. También puede denotar el margen de notas que es capaz de tocar un determinado instrumento musical. Los ámbitos se imprimen en las partes vocales de tal manera que los intérpretes puedan determinar con facilidad si cumplen con sus propias posibilidades.

Los ámbitos se presentan al comienzo de la pieza junto a la clave inicial. El rango se especifica gráficamente mediante dos cabezas de nota que representan a las notas inferior y superior. Solo se imprimen alteraciones si no forman parte de la armadura de la tonalidad.

```
\layout {
  \context {
    \Voice
    \consists Ambitus_engraver
  }
}

\relative {
  aes' c e2
```

```
  cis,1
}
```

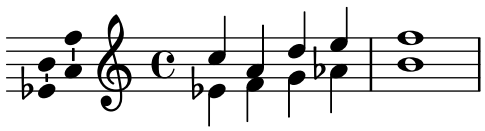


Fragmentos de código seleccionados

Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```

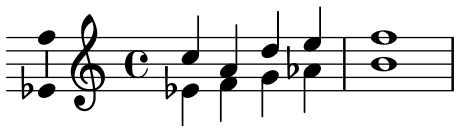


Ámbitos con varias voces

La adición del grabador `Ambitus_engraver` al contexto de `Staff` crea un solo ámbito por pentagrama, incluso en el caso de pentagramas con varias voces.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
```

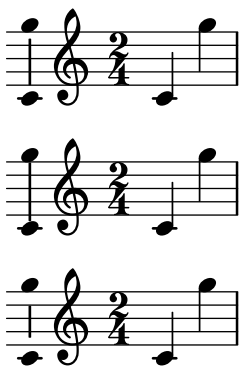
```
  }
>>
```

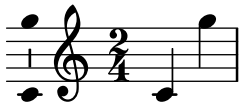


Modificación de la separación en las indicaciones de tesitura

Es posible ajustar la separación predeterminada entre las notas de la indicación de tesitura y la línea que las une.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1.5
  c'4 g''
}
```



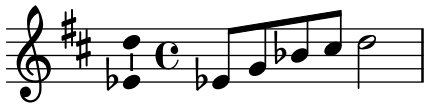


Indicación de tesitura después de la armadura

De manera predeterminada, las indicaciones de ámbito o tesitura se sitúan a la izquierda de la clave. La función `\ambitusAfter` permite cambiar esta colocación. La sintaxis es `\ambitusAfter` *grob-interface* (consulte Interfaces de los objetos gráficos (`https://lilypond.org/doc/v2.24/Documentation/internals/graphical-object-interfaces`) para ver una lista de los valores posibles para *grob-interface*.)

Un caso de utilización bastante común es imprimir la indicación de texitura entre la armadura de la clave y la indicación de compás.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Véase también

Glosario musical: Sección “ambitus” en *Glosario Musical*.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ambitus_engraver” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “Ambitus” en *Referencia de Funcionamiento Interno*, Sección “AmbitusAccidental” en *Referencia de Funcionamiento Interno*, Sección “AmbitusLine” en *Referencia de Funcionamiento Interno*, Sección “AmbitusNoteHead” en *Referencia de Funcionamiento Interno*, Sección “ambitus-interface” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No se efectúa un tratamiento de las posibles colisiones en caso de varias indicaciones de ámbito simultáneas en distintas voces.

1.4 Cabeza de las notas

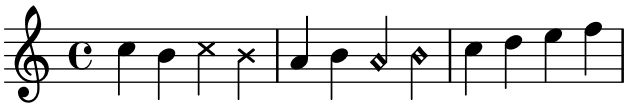
Esta sección propone formas de alterar las cabezas de las figuras.

1.4.1 Cabezas de nota especiales

Se puede modificar la apariencia de la cabeza de las notas:

```
\relative c'' {
  c4 b
  \override NoteHead.style = #'cross
  c4 b
  \revert NoteHead.style
  a b
```

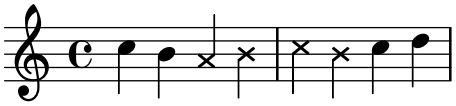
```
  \override NoteHead.style = #'harmonic
  a b
  \revert NoteHead.style
  c4 d e f
}
```



Para ver todos los estilos de cabeza de las notas, consulte Sección B.9 [Estilos de cabezas de nota], página 927.

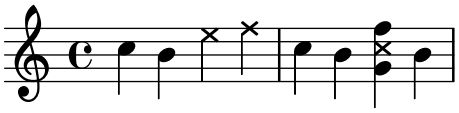
El estilo `cross` (aspas) se usa para representar una amplia variedad de intenciones musicales. Las siguientes instrucciones genéricas predefinidas modifican la forma de la cabeza de las figuras tanto en el contexto de pentagrama normal como en el de tablatura, y se pueden usar para representar cualquier significado musical:

```
\relative {
  c''4 b
  \xNotesOn
   a b c4 b
  \xNotesOff
  c4 d
}
```



La forma de función musical de esta instrucción predefinida se puede usar dentro y fuera de los acordes para generar cabezas de nota en aspa, tanto en el contexto de pentagrama normal como en el de tablatura:

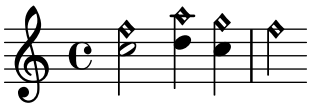
```
\relative {
  c''4 b
  \xNote { e f }
  c b < g \xNote c f > b
}
```



Como sinónimos de `\xNote`, `\xNotesOn` y `\xNotesOff`, se pueden usar `\deadNote`, `\deadNotesOn` y `\deadNotesOff`. El término *dead note* (nota muerta) se utiliza corrientemente por parte de los guitarristas.

También existe una abreviatura similar para las formas en rombo:

```
\relative c'' {
  <c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic> f\harmonic
}
```



Instrucciones predefinidas

`\harmonic`, `\xNotesOn`, `\xNotesOff`, `\xNote`.

Véase también

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

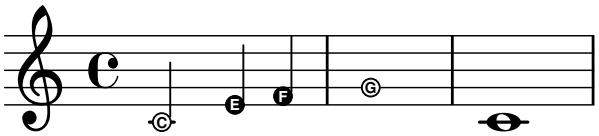
Referencia de la notación: Sección B.9 [Estilos de cabezas de nota], página 927, Sección 5.1.1 [Notas en acorde], página 218, Sección 12.2.2 [Indicar armónicos y notas tapadas], página 485.

Referencia de funcionamiento interno: Sección “note-event” en *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” en *Referencia de Funcionamiento Interno*, Sección “Ledger_line_engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteHead” en *Referencia de Funcionamiento Interno*, Sección “LedgerLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “note-head-interface” en *Referencia de Funcionamiento Interno*, Sección “ledger-line-spanner-interface” en *Referencia de Funcionamiento Interno*.

1.4.2 Cabezas de notas de Notación Fácil

Las notas con cabeza de ‘notación facilitada’ tienen el nombre de la nota (en inglés) dentro de la cabeza. Se usan en la música para principiantes. Para que las letras sean legibles, se deben imprimir en un tamaño grande de fuente tipográfica. Para imprimir con una fuente más grande, véase Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

```
#(set-global-staff-size 26)
\relative c' {
  \easyHeadsOn
  c2 e4 f
  g1
  \easyHeadsOff
  c,1
}
```



Instrucciones predefinidas

`\easyHeadsOn`, `\easyHeadsOff`.

Fragmentos de código seleccionados

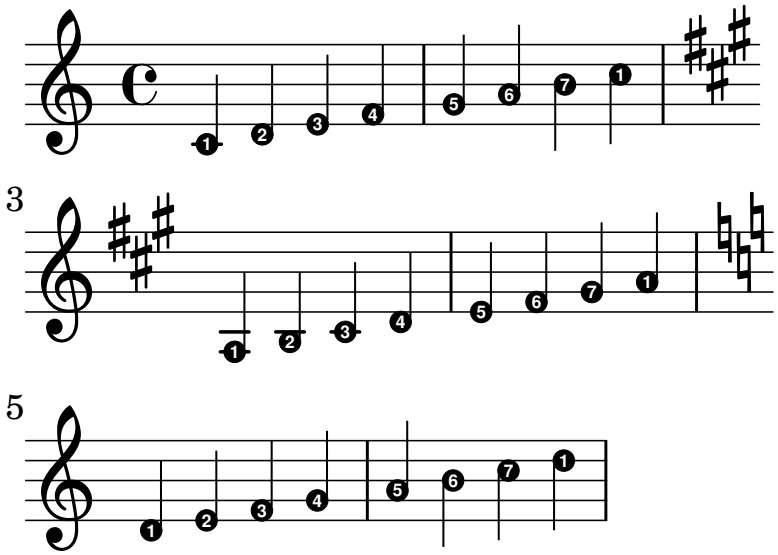
Números como notas de notación fácil

Las cabezas de nota de notación fácil utilizan la propiedad `note-names` del objeto `NoteHead` para determinar lo que aparece dentro de la cabeza. Mediante la sobreescritura de esta propiedad, es posible imprimir números que representen el grado de la escala.

Se puede crear un grabador simple que haga esto para la cabeza de cada nota que ve.

```
#(define Ez_numbers_engraver
   (make-engraver
    (acknowledgers
     ((note-head-interface engraver grob source-engraver)
      (let* ((context (ly:translator-context engraver))
             (tonic-pitch (ly:context-property context 'tonic))
             (tonic-name (ly:pitch-notename tonic-pitch))
```

```
              (grob-pitch
               (ly:event-property (event-cause grob) 'pitch))
              (grob-name (ly:pitch-notename grob-pitch))
              (delta (modulo (- grob-name tonic-name) 7))
              (note-names
               (make-vector 7 (number->string (1+ delta)))))
            (ly:grob-set-property! grob 'note-names note-names))))))

#(set-global-staff-size 30)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}
```



Véase también

Referencia de la notación: Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

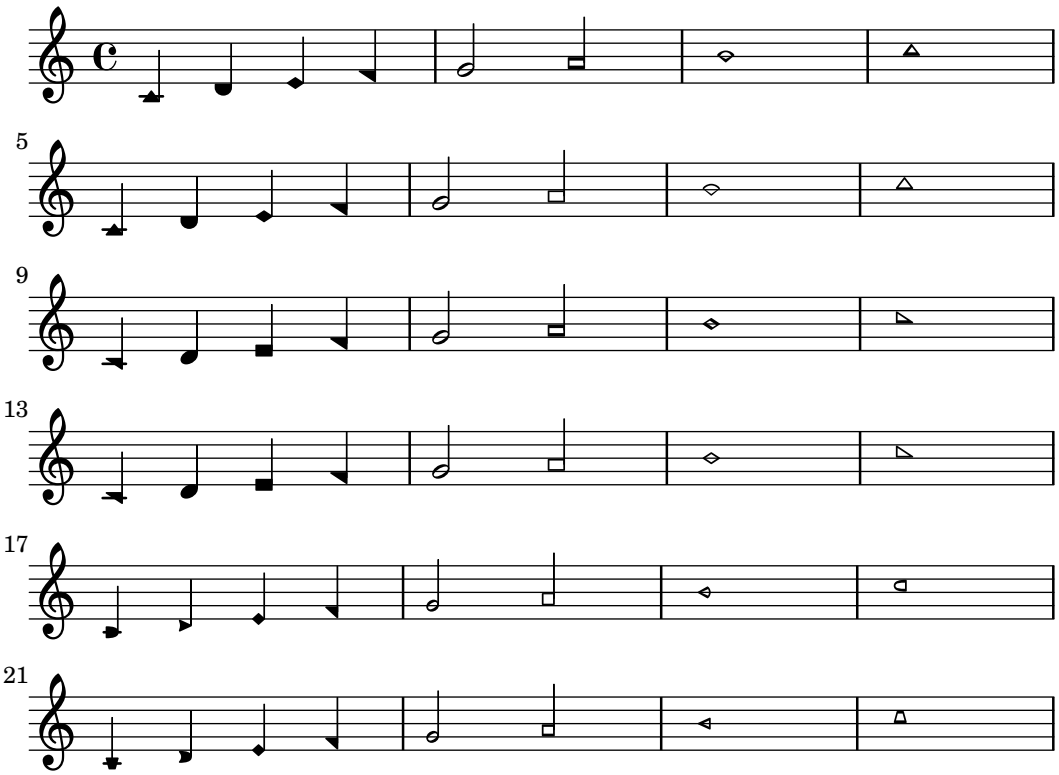
Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “note-event” en *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteHead” en *Referencia de Funcionamiento Interno*, Sección “note-head-interface” en *Referencia de Funcionamiento Interno*.

1.4.3 Cabezas de notas con formas diversas

En la notación de cabezas con forma, la forma de la cabeza corresponde a la función armónica de una nota dentro de la escala. Esta notación se hizo popular en los libros americanos de canciones durante el s.XIX. Las cabezas de nota con formas se pueden producir en los estilos ‘Sacred Harp’, ‘Southern Harmony’, Funk (Harmonia Sacra), Walker y Aiken ‘(Christian Harmony)’:

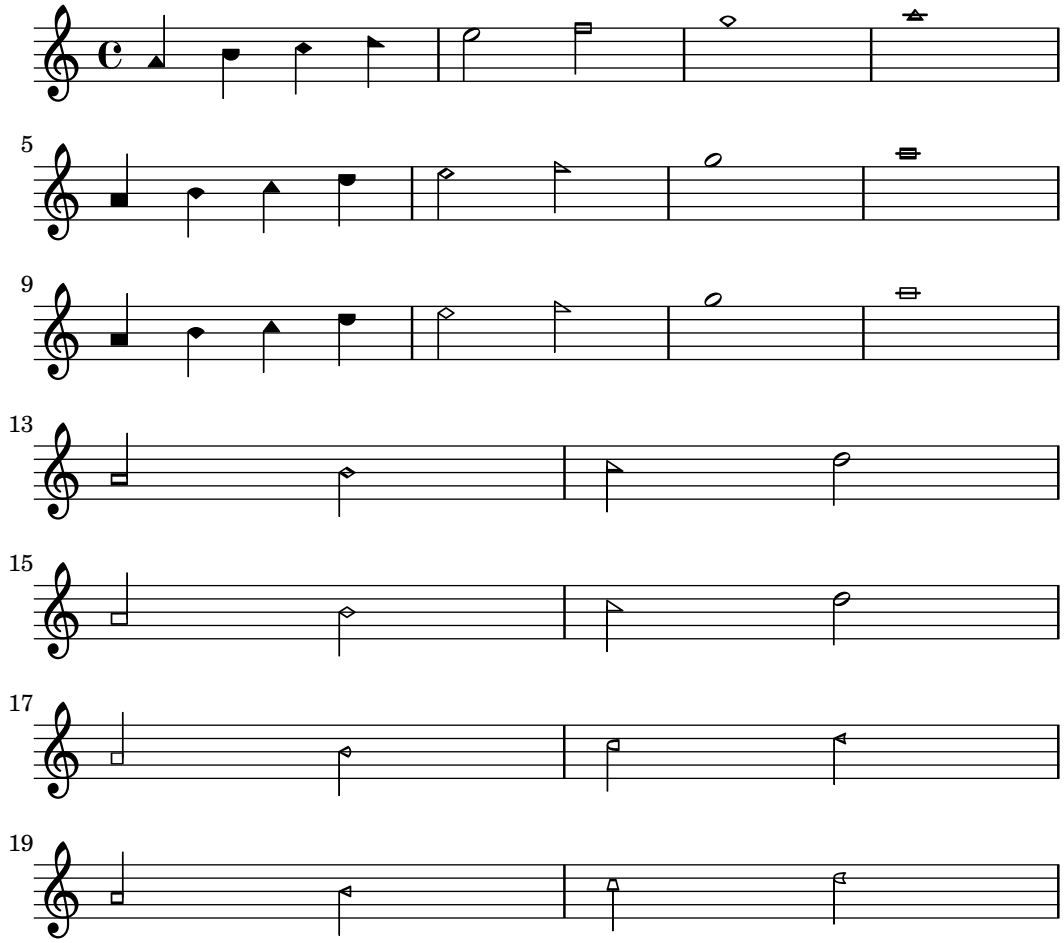
```
\relative c'' {
  \aikenHeads
  c, d e f g2 a b1 c \break
  \aikenThinHeads
  c,4 d e f g2 a b1 c \break
  \sacredHarpHeads
  c,4 d e f g2 a b1 c \break
  \southernHarmonyHeads
  c,4 d e f g2 a b1 c \break
  \funkHeads
  c,4 d e f g2 a b1 c \break
  \walkerHeads
  c,4 d e f g2 a b1 c \break
}
```



Las formas se determinan en función del grado de la escala, donde la tónica está determinada por la instrucción \key Cuando se escribe en un tono menor, la nota de la escala se puede determinar a partir del relativo mayor:

```
\relative c'' {
```

```
  \key a \minor
  \aikenHeads
  a b c d e2 f g1 a \break
  \aikenHeadsMinor
  a,4 b c d e2 f g1 a \break
  \aikenThinHeadsMinor
  a,4 b c d e2 f g1 a \break
  \sacredHarpHeadsMinor
  a,2 b c d \break
  \southernHarmonyHeadsMinor
  a2 b c d \break
  \funkHeadsMinor
  a2 b c d \break
  \walkerHeadsMinor
  a2 b c d \break
}
```



Instrucciones predefinidas

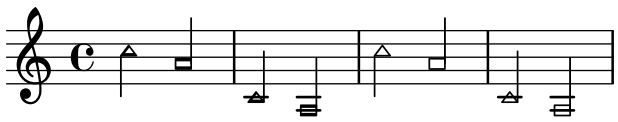
\aikenHeads, \aikenHeadsMinor, \aikenThinHeads, \aikenThinHeadsMinor, \funkHeads, \funkHeadsMinor, \sacredHarpHeads, \sacredHarpHeadsMinor, \southernHarmonyHeads, \southernHarmonyHeadsMinor, \walkerHeads, \walkerHeadsMinor.

Fragmentos de código seleccionados

Cabezas de nota tipo Aiken, variante delgada

Las notas blancas de tipo Aiken se vuelven algo difíciles de leer con tamaños de pentagrama pequeños, especialmente con líneas adicionales. Al perder el espacio vacío del interior pueden parecer más bien figuras negras.

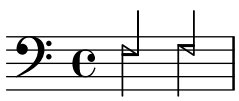
```
\score {
  {
    \aikenHeads
    c''2 a' c' a

    % Switch to thin-variant noteheads
    \set shapeNoteStyles = ##(doThin reThin miThin
                              faThin sol laThin tiThin)
    c'' a' c' a
  }
}
```



Direction of merged 'fa' shape note heads

Using property `NoteCollision.fa-merge-direction`, the direction of "fa" shape note heads ("fa", "faThin", etc.) can be controlled independently of the stem direction if two voices with the same pitch and different stem directions are merged. If this property is not set, the "down" glyph variant is used.

```
{
  \clef bass

  << { \aikenHeads
       f2
       \override Staff.NoteCollision.fa-merge-direction = #UP
       f2 }
  \\ { \aikenHeads
       f2
       f2 }
  >>
}
```



Aplicar estilos de cabeza según la nota de la escala

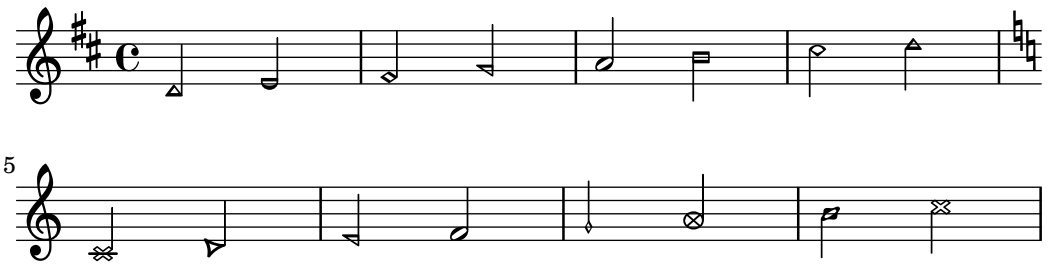
La propiedad `shapeNoteStyles` se puede usar para definir varios estilos de cabezas de nota para cada grado de la escala (según esté establecido por la armadura o por la propiedad `tonic`).

Esta propiedad requiere un conjunto de símbolos, que pueden ser puramente arbitrarios (se permiten expresiones geométricas como `triangle`, triángulo, `cross`, aspas, y `xcircle`, círculo con aspas) o basados en una antigua tradición americana de grabado (ciertos nombres de nota latinos trambién se permiten).

Dicho esto, para imitar antiguos cancioneros americanos, existen varios estilos predefinidos de cabezas de nota disponibles a través de instrucciones de abreviatura como `\aikenHeads` o `\sacredHarpHeads`.

Este ejemplo muestra distintas formas de obtener cabezas de notas con forma, y muestra la capacidad de transportar una melodía sin perder la correspondencia entre las funciones armónicas y los estilos de cabezas de nota.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                              #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                              mensural xcircle diamond)
    \fragment
  }
}
```



Para ver todos los estilos de formas de cabeza de nota, consulte Sección B.9 [Estilos de cabezas de nota], página 927.

Véase también

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de la notación: Sección B.9 [Estilos de cabezas de nota], página 927.

Referencia de funcionamiento interno: Sección “note-event” en *Referencia de Funcionamiento Interno*, Sección “Note_heads_engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteHead” en *Referencia de Funcionamiento Interno*, Sección “note-head-interface” en *Referencia de Funcionamiento Interno*.

1.4.4 Improvisación

La improvisación se denota a veces mediante cabezas de nota en forma de barra inclinada, donde el ejecutante puede elegir cualquier nota pero con el ritmo especificado. Estas cabezas de nota se crean así:

```
\new Voice \with {
  \consists Pitch_squash_engraver
} \relative {
  e''8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  2 ~ 8 f4 f8 ~
  2
  \improvisationOff
  a16( bes) a8 g e
}
```



Instrucciones predefinidas

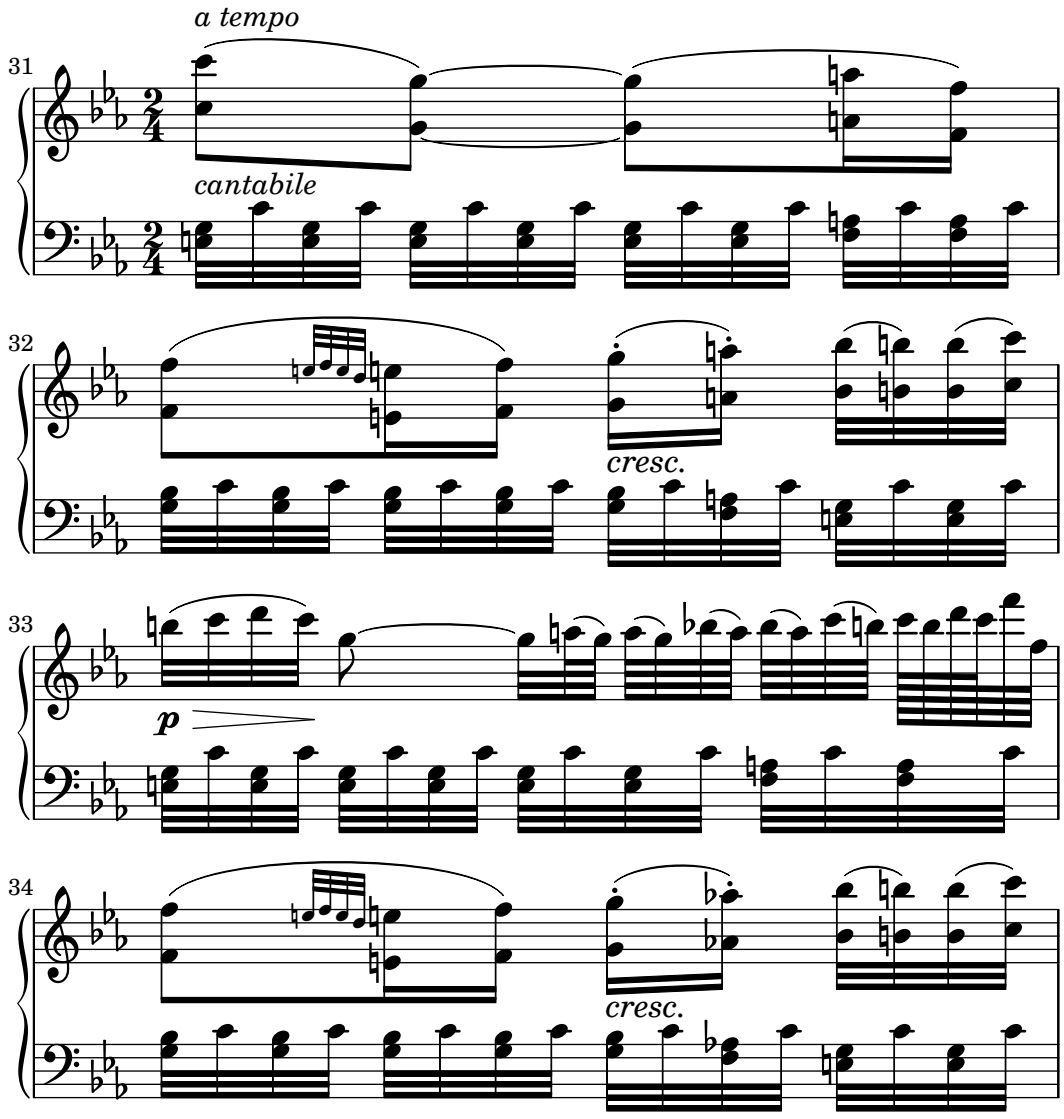
`\improvisationOn`, `\improvisationOff`.

Véase también

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Pitch_squash_engraver” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” en *Referencia de Funcionamiento Interno*.

2 Duraciones



Esta sección trata de los ritmos, los silencios, las duraciones, las barras y los compases.

2.1 Escritura de las duraciones (valores rítmicos)

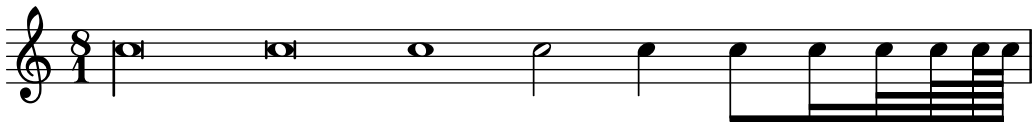
2.1.1 Duración de las notas

Las duraciones de las notas se introducen usando números y puntos. El número que se introduce está basado en el valor inverso de la duración de la nota respecto a la redonda. Por ejemplo, una negra (con un cuarto de la duración de la redonda) se designa usando el valor numérico de `4`, una blanca con un `2`, una corchea con un `8` y así sucesivamente. Se pueden escribir notas tan breves como `1024` pero cualquier duración más breve, aunque es posible, solo se puede introducir como notas unidas por una barra. Véase también Sección 2.4 [Barras], página 101.

Para notas mayores de la redonda se deben usar las instrucciones `\longa` (que es una breve doble) y `\breve`. Se puede escribir una nota con la duración de una cuádruple breve mediante `\maxima`, pero esto está contemplado solamente dentro de la notación musical antigua. Para ver más detalles, consulte Capítulo 17 [Notación antigua], página 540.

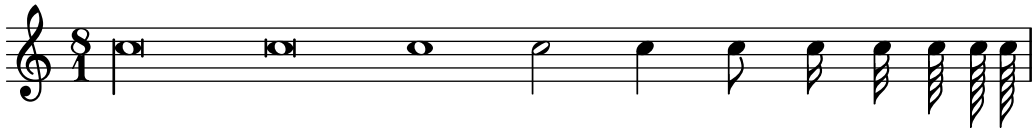
```
\relative {
```

```
  \time 8/1
  c''\longa c\breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
```



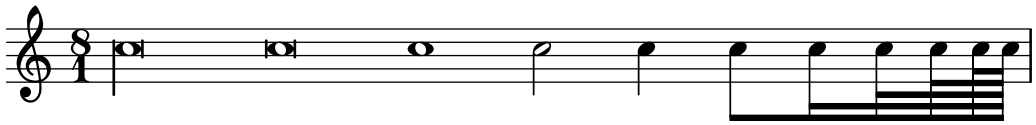
Aquí se pueden ver las mismas duraciones con el barrado automático desactivado.

```
\relative {
  \time 8/1
  \autoBeamOff
  c''\longa c\breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
```



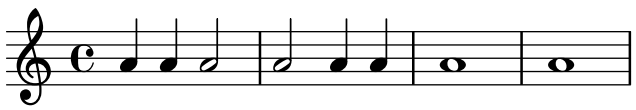
Las duraciones aisladas (aquellas que no especifican ninguna altura) en una secuencia musical toman la altura de la nota o acorde anterior.

```
\relative {
  \time 8/1
  c'' \longa \breve 1 2
  4 8 16 32 64 128 128
}
```



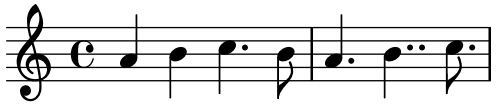
Las alturas aisladas (notas sin ninguna duración especificada) que aparecen dentro de una secuencia musical, obtienen su duración de la nota o acorde anterior. Si no hay ninguna duración anterior, entonces el valor predeterminado que se utiliza para la nota siempre es 4, una negra.

```
\relative { a' a a2 a a4 a a1 a }
```



Para obtener duraciones de notas con puntillo, escriba un punto (.) después del número. Las notas con doble puntillo se especifican escribiendo dos puntos, y así sucesivamente.

```
\relative { a'4 b c4. b8 a4. b4.. c8. }
```



Para evitar la superposición con las líneas del pentagrama, los puntillos se desplazan normalmente hacia arriba. Sin embargo, en situaciones de polifonía, se pueden colocar manualmente por encima o por debajo según sea necesario. Véase Sección 36.1 [Dirección y posición], página 783.

Ciertas duraciones no se pueden representar solamente mediante números y puntillos; puede ser necesario emplear dos o más notas ligadas entre sí. Véase Sección 2.1.4 [Ligaduras de unión], página 63.

Para especificat duraciones que puedan alinear las notas con las sílabas de la letra, véase Capítulo 9 [Música vocal], página 349.

Las notas también se pueden espaciar proporcionalmente a su duración, véase Sección 30.6 [Notación proporcional], página 728.

Instrucciones predefinidas

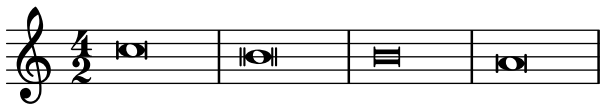
`\autoBeamOn`, `\autoBeamOff`, `\dotsUp`, `\dotsDown`, `\dotsNeutral`.

Fragmentos de código seleccionados

Formas alternativas de la figura breve

Las figuras de breve también están disponibles con dos líneas verticales a los lados de la cabeza en lugar de una sola línea y la forma en estilo barroco.

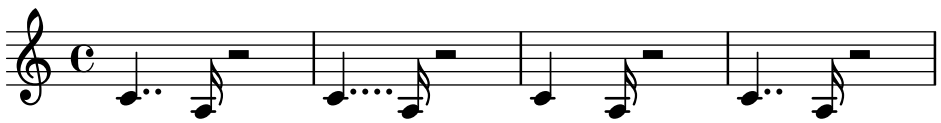
```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```



Modificar el número de puntillos de una nota

La cantidad de puntillos de una nota se puede sobreescribir mediante el establecimiento de la propiedad `dot-count` del objeto gráfico `Dots`.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
  \override Dots.dot-count = 0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```



Véase también

Glosario musical: Sección “breve” en *Glosario Musical*, Sección “longa” en *Glosario Musical*, Sección “maxima” en *Glosario Musical*, Sección “note value” en *Glosario Musical*, Sección “Duration names notes and rests” en *Glosario Musical*.

Referencia de la notación: Sección 2.4 [Barras], página 101, Sección 2.1.4 [Ligaduras de unión], página 63, Sección 7.1.9 [Plicas], página 300, Sección 2.1 [Escritura de las duraciones (valores rítmicos)], página 53, Sección 2.2 [Escritura de silencios], página 68, Capítulo 9 [Música vocal], página 349, Capítulo 17 [Notación antigua], página 540, Sección 30.6 [Notación proporcional], página 728.

Fragmentos de código: Sección "Duraciones" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Dots" en *Referencia de Funcionamiento Interno*, Sección "DotColumn" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existe un límite fundamental para las duraciones de los silencios (tanto para el más largo como para el más corto), pero el número de glifos es limitado: solo se pueden imprimir desde el silencio de 1/1024 de redonda (sin nombre en español) hasta el de `\maxima` (8 redondas).

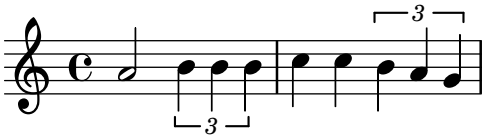
2.1.2 Grupos especiales

Los grupos especiales se obtienen a partir de una expresión musical con la instrucción `\tuplet`, multiplicando la rapidez de la expresión musical por una fracción:

```
\tuplet fracción { música }
```

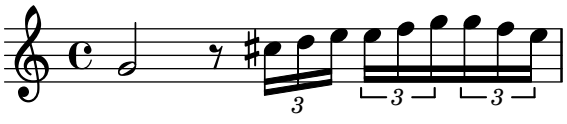
El numerador de la fracción se imprime encima o debajo de las notas, opcionalmente con un corchete. El grupo especial más común es el tresillo, en el que 3 notas tienen el valor que normalmente tienen 2:

```
\relative {
  a'2 \tuplet 3/2 { b4 4 4 }
  c4 c \tuplet 3/2 { b4 a g }
}
```



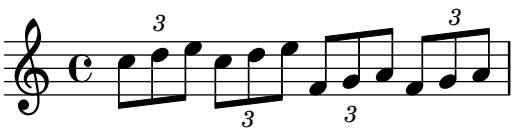
Si se están escribiendo pasajes con muchos grupos de valoración especial, resulta fastidioso tener que escribir una instrucción `\tuplet` distinta para cada grupo. Es posible especificar la duración de un conjunto de grupos directamente antes de la música para que así se agrupen automáticamente:

```
\relative {
  g'2 r8 \tuplet 3/2 8 { cis16 d e e f g g f e }
}
```



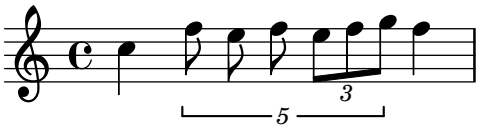
Se pueden colocar manualmente los corchetes de tresillo encima o debajo de la pauta:

```
\relative {
  \tupletUp \tuplet 3/2 { c''8 d e }
  \tupletNeutral \tuplet 3/2 { c8 d e }
  \tupletDown \tuplet 3/2 { f,8 g a }
  \tupletNeutral \tuplet 3/2 { f8 g a }
}
```



Los grupos pueden anidarse unos dentro de otros:

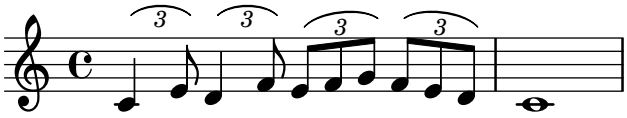
```
\relative {
  \autoBeamOff
  c''4 \tuplet 5/4 { f8 e f \tuplet 3/2 { e[ f g] } } f4
}
```



La modificación de los grupos especiales anidados que comienzan en el mismo instante musical se debe hacer con `\tweak`; véase Sección 35.6 [Las instrucciones `\tweak` y `\single`], página 773.

Se puede sustituir el corchete recto de un grupo de valoración especial con una ligadura curva, a la manera de muchas ediciones antiguas:

```
\relative {
  \tuplet 3/2 4 {
    \override TupletBracket.tuplet-slur = ##t
    c'4 e8 d4 f8
    \override TupletBracket.bracket-visibility = ##t
    e f g f e d
  } c1
}
```



De forma predeterminada, solo se imprime el corchete si no están unidas por una barra todas las notas que comprende el grupo; en ciertos casos (por ejemplo con las ligaduras, como en el ejemplo de arriba) podría ser preferible modificar este comportamiento, a través de la propiedad `bracket-visibility` tal y como se detalla en uno de los siguientes fragmentos de código.

De manera más general se pueden mostrar u ocultar los objetos `TupletBracket` o `TupletNumber`, o ambos, como se explica en Sección 36.7 [Visibilidad de los objetos], página 794; sin embargo, una forma más flexible de modificar la duración de las notas sin imprimir un grupo especial (ni un corchete) se explica también en Sección 2.1.3 [Escalar las duraciones], página 62.

Instrucciones predefinidas

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

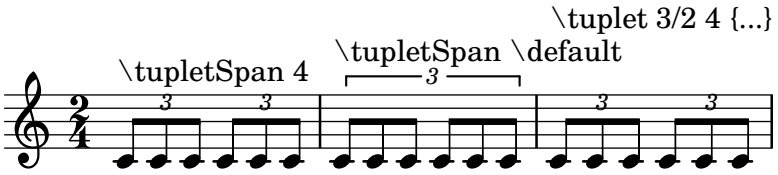
Fragmentos de código seleccionados

Escribir varios tresillos con una sola instrucción `\tuplet`

La propiedad `tupletSpannerDuration` establece cuánto tiempo debe durar cada uno de los tresillos contenidos dentro de los corchetes después de la instrucción `\tuplet`. Muchos tresillos consecutivos se pueden entonces escribir dentro de una sola instrucción `\tuplet`, ahorrando algo de tecleo.

Hay formas de fijar `tupletSpannerDuration` aparte de usar una instrucción `\set`. La instrucción `\tupletSpan` lo fija a una duración dada, o la borra cuando se especifica en su lugar una duración `\default`. Otra forma es usar un argumento opcional con `\tuplet`.

```
\relative c' {
  \time 2/4
  \tupletSpan 4
  \tuplet 3/2 { c8^"\\tupletSpan 4" c c c c c }
  \tupletSpan \default
  \tuplet 3/2 { c8^"\\tupletSpan \\default" c c c c c }
  \tuplet 3/2 4 { c8^"\\tuplet 3/2 4 {...}" c c c c c }
}
```

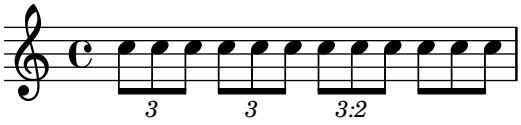


Cambiar el número del grupo especial

De forma predeterminada solo se imprime sobre el corchete de grupo el numerador del grupo especial, o sea, el numerador del argumento de la instrucción `\tuplet`.

De forma alternativa, se puede imprimr un quebrado en la forma *numerador*:*denominador* del número del grupo, o eliminar el número.

```
\relative c'' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
}
```

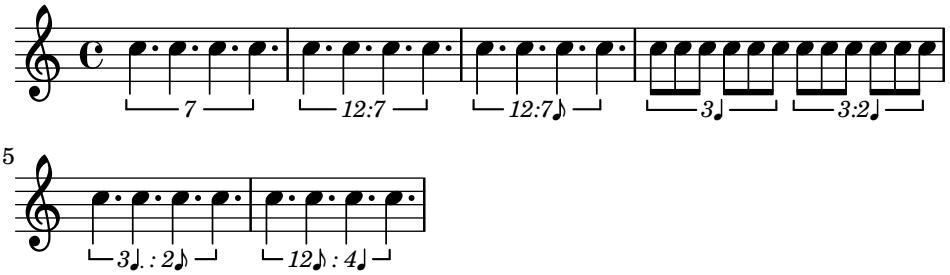


Números de agrupación especial distintos a los predeterminados

LilyPond también proporciona funciones de formato para imprimir números de grupo especial diferentes a la propia fracción, así como para añadir una figura al número o a la fracción de la agrupación.

```
\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
```

```
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7)
      (ly:make-duration 3 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text
      (ly:make-duration 2 0))
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text
      (ly:make-duration 2 0))
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::fraction-with-notes
      (ly:make-duration 2 1) (ly:make-duration 3 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-fraction-with-notes 12
      (ly:make-duration 3 0) 4 (ly:make-duration 2 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
}
```



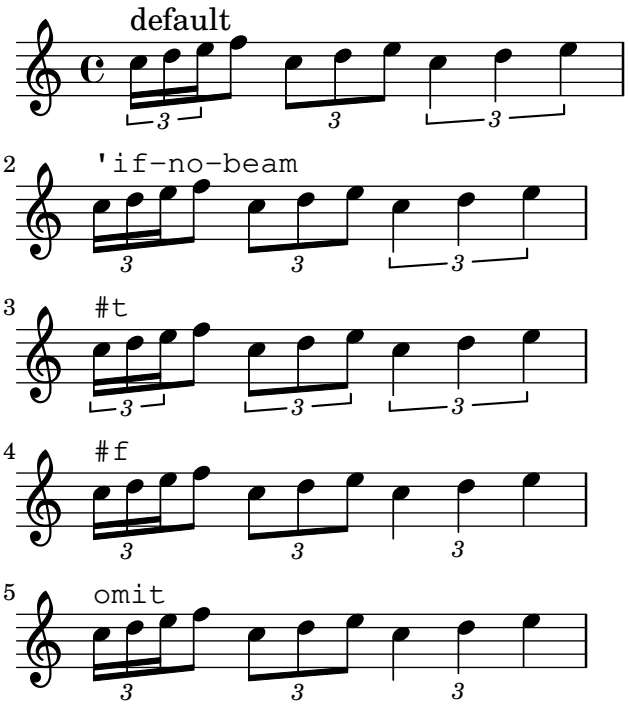
Controlar la visibilidad de los corchetes de grupo especial

El comportamiento predeterminado de la visibilidad de los corchetes de grupo de valoración especial es imprimir el corchete a no ser que haya una barra de la misma longitud que el grupo especial.

Para controlar la visibilidad de los corchetes de grupo, establezca la propiedad `bracket-visibility` a `#t` (imprimir el corchete siempre), `#if-no-beam` (imprimir el corchete solamente si no hay barra), o `#f` (no imprimir nunca el corchete). Este último equivale de hecho a omitir el objeto `TupletBracket` de la salida impresa.

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
```

```
  \relative c' {
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    \textMark \markup \typewriter "'if-no-beam" \music
    \override TupletBracket.bracket-visibility = ##t
    \textMark \markup \typewriter "#t" \music
    \override TupletBracket.bracket-visibility = ##f
    \textMark \markup \typewriter "#f" \music
    \omit TupletBracket
    \textMark \markup \typewriter "omit" \music
  }
}
```

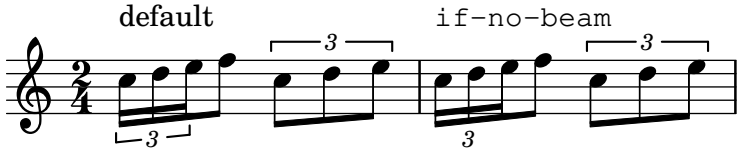


Imrpesión de corchetes de tresillo en el lado de la cabeza de la nota

Cualquiera que sea la opción elegida para el control de la visibilidad del corchete de los tresillos, mostrará u ocultará el corchete independientemenre de su ubicación (en el lado de la plica o en el lado de la cabeza). Sin embargo, cuando se coloca el corchete del lado de la cabeza, algunos autores recomiendan imprimir siempre el corchete de tresillo. Se puede usar la opción `visible-over-note-heads` para obtener esto.

```
music = \relative c'' {
  \tupletNeutral \tuplet 3/2 { c16[ d e } f8]
  \tupletUp \tuplet 3/2 { c8 d e }
}

\new Voice {
  \relative c' {
    \override TextScript.staff-padding = #2.5

    \time 2/4
    \override TupletBracket.visible-over-note-heads = ##t
```

```
    \override Score.TextMark.non-musical = ##f
    <>^\markup "default" \music
    \override TupletBracket.bracket-visibility = #'if-no-beam
    <>^\markup \typewriter "if-no-beam" \music
  }
}
```

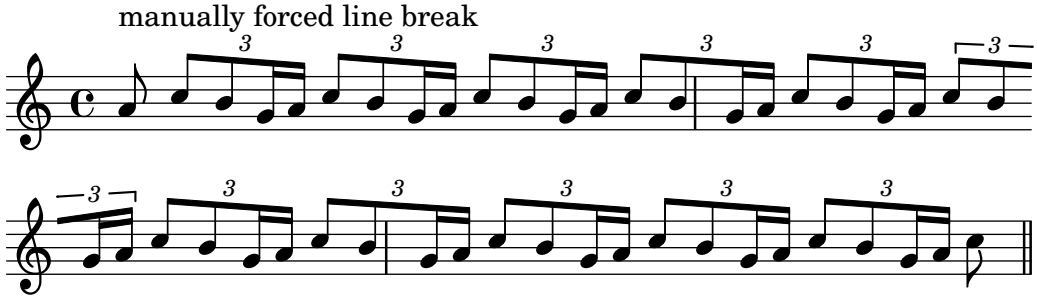


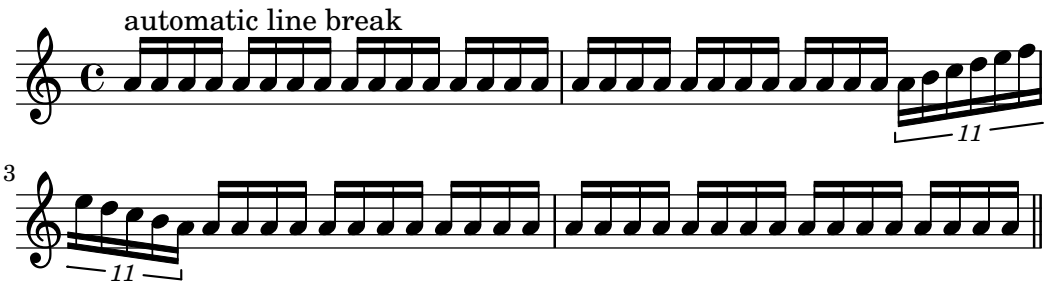
Permitir saltos de línea dentro de grupos especiales con barra

Estos ejemplos artificiales muestran cómo se pueden permitir tanto los saltos de línea manuales como los automáticos dentro de un grupo de valoración especial unido por una barra que no se puede dividir en una duración de manera exacta.

Esta funcionalidad solo funciona con grupos de valoración especial barrados manualmente.

```
\layout {
  \context {
    \Voice
    % Permit automatic line breaks within tuplets.
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks.
    \override Beam.breakable = ##t
  }
}

\relative c'' {
  <>^"manually forced line break"
  a8
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  \tuplet 3/2 { c8[ b \break g16 a] }
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  c8 \bar "||"
}

\relative c'' {
  <>^"automatic line break"
  \repeat unfold 28 a16
  \tuplet 11/8 { a16[ b c d e f e d c b a] }
  \repeat unfold 28 a16 \bar "||"
}
```





Véase también

Glosario musical: Sección “triplet” en *Glosario Musical*, Sección “tuplet” en *Glosario Musical*, Sección “polymetric” en *Glosario Musical*.

Manual de aprendizaje: Sección “Métodos de trucaje” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783, Sección 36.7 [Visibilidad de los objetos], página 794, Sección 2.6.3 [Gestión del tiempo], página 155, Sección 2.1.3 [Escalar las duraciones], página 62, Sección 35.6 [Las instrucciones `\tweak` y `\single`], página 773, Sección 2.3.5 [Notación polimétrica], página 93.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TupletBracket” en *Referencia de Funcionamiento Interno*, Sección “TupletNumber” en *Referencia de Funcionamiento Interno*, Sección “TimeScaledMusic” en *Referencia de Funcionamiento Interno*.

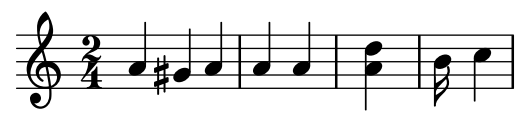
2.1.3 Escalar las duraciones

La duración de las figuras, silencios o acordes se puede multiplicar por un factor `N/M` añadiendo `*N/M` (o `*N` si `M` es 1) a la duración. También se pueden añadir factores usando expresiones de Scheme que se evalúan a un número o a una duración musical como `*#(ly:music-length` *`music`*`)`. Esto es muy útil para escalar una duración de ‘`1`’ y que una nota o silencio multicompás se estire hasta una longitud derivada de una variable musical.

La adición de un factor no afectará a la apariencia de las notas o silencios que se producen, pero la duración alterada se usará para calcular la posición dentro del compás y para establecer la duración en la salida MIDI. Los factores de multiplicación se pueden combinar en la forma `*L*M/N`. Los factores son parte de la duración: si no especificamos una duración para las notas siguientes, la duración por omisión que se toma de la nota anterior incluirá cualquier factor de escala que se haya aplicado.

En el siguiente ejemplo las tres primeras notas duran exactamente dos partes, pero no se imprime ningún corchete de tresillo.

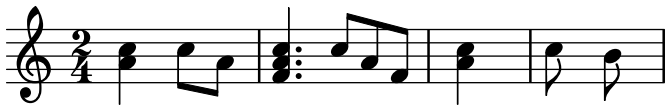
```
\relative {
  \time 2/4
  % Alter durations to triplets
  a'4*2/3 gis a
  % Normal durations
  a4 a
  % Double the duration of chord
  <a d>4*2
  % Duration of quarter, appears like sixteenth
  b16*4 c4
}
```



La duración de los silencios espaciadores también se puede modificar mediante un multiplicador. Esto es útil para saltar muchos compases, como por ejemplo `s1*23`.

De la misma forma, se pueden comprimir por una fracción trozos de música más largos, como si cada nota, acorde o silencio tuviera la fracción como multiplicador. Esto dejará intacta la apariencia de la *música*, pero la duración interna de las notas se multiplicará por el factor de escala dado, habitualmente *numerador/denominador*. He aquí un ejemplo que muestra cómo se puede comprimir y expandir la música:

```
\relative {
  \time 2/4
  % Normal durations
  <c'' a>4 c8 a
  % Scale music by *2/3
  \scaleDurations 2/3 {
    <c a f>4. c8 a f
  }
  % Scale music by *2
  \scaleDurations 2 {
    <c' a>4 c8 b
  }
}
```



Una aplicación de esta instrucción se encuentra en la notación polimétrica, véase Sección 2.3.5 [Notación polimétrica], página 93.

Véase también

Referencia de la notación: Sección 2.1.2 [Grupos especiales], página 56, Sección 2.2.2 [Silencios invisibles], página 70, Sección 2.3.5 [Notación polimétrica], página 93.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Advertencias y problemas conocidos

El cálculo de la posición dentro de un compás debe tener en cuenta todos los factores de escalado que se han aplicado a las notas dentro de ese compás, y cualquier acarreo fraccionado desde los compases anteriores. Este cálculo se efectúa utilizando números racionales. Si un numerador o denominador intermedio en dicho proceso de cálculo sobrepasa la cantidad de 2^30, la ejecución y el tipografiado se detendrán en ese punto sin indicar ningún error.

2.1.4 Ligaduras de unión

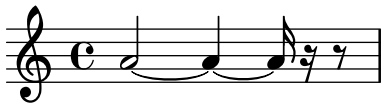
Una ligadura de unión conecta dos notas adyacentes de la misma altura. La ligadura en efecto extiende la longitud de una nota.

Las ligaduras que no se unen a nada se conocen como la articulacion *laissez vibrer* (dejar vibrar); véase [Laissez vibrer], página 65, para la instrucción `\laissezVibrer`. Las ligaduras que conectan ‘nada’ a notas (como a veces es necesario en la casilla de segunda vez en una repetición), se pueden introducir con la instrucción `\repeatTie`; véase [Repeat tie], página 64.

Nota: No deben confundirse las ligaduras de unión con las *ligaduras de expresión*, que indican articulación, ni con las *ligaduras de fraseo*, que indican el fraseo musical. Una ligadura de unión es tan solo una manera de extender la duración de una nota, algo parecido a lo que hace el puntillo.

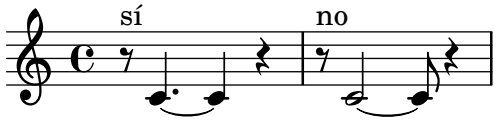
Se introduce una ligadura de unión escribiendo el símbolo de la tilde curva (‘~’) después de la primera de cada pareja de notas que se quieren unir. Esto indica que la nota se une a la siguiente, que debe tener la misma altura. Observe que las ligaduras de unión pueden hacer uso de la interpretación ‘última altura explícita’ para las duraciones aisladas:

```
{ a'2~ 4~ 16 r r8 }
```



Se usan ligaduras de unión bien cuando la nota atraviesa la barra de compás o bien cuando no se pueden usar puntillos para denotar el ritmo. También se deben usar ligaduras cuando las notas atraviesan subdivisiones del compás de mayor duración:

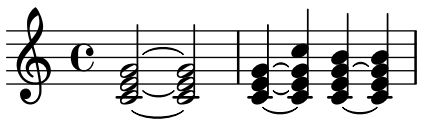
```
\relative {
  r8^"sí" c'4.~ 4 r4 |
  r8^"no" c2~ 8 r4
}
```



Si necesitamos ligar muchas notas a través de las líneas divisorias, nos podría resultar más fácil utilizar la división automática de las notas, véase Sección 2.3.6 [División automática de las notas], página 97. Este mecanismo divide automáticamente las notas largas y las liga a través de las barras de compás.

Cuando se aplica una ligadura de unión a un acorde, se conectan todas las cabezas de las notas cuyas alturas coinciden. Si no coincide ningún par de cabezas, no se crea ninguna ligadura. Los acordes se pueden ligar parcialmente colocando las ligaduras dentro del acorde.

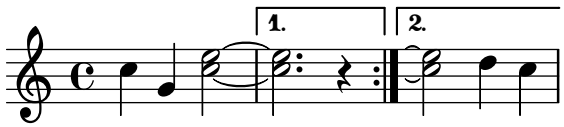
```
\relative c' {
  <c e g>2~ 2 |
  <c e g>4~ <c e g c>
    <c~ e g~ b> <c e g b> |
}
```



Cuando una ligadura de unión se prolonga hasta el interior de una casilla de repetición de primera o segunda vez, es necesario especificar la ligadura repetida como sigue:

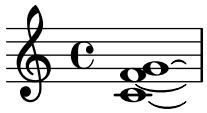
```
\relative {
  \repeat volta 2 { c'' g <c e>2~ }
  \alternative {
```

```
    % the following note is tied normally
    \volta 1 { <c e>2. r4 }
    % the following note has a repeated tie
    \volta 2 { <c e>2\repeatTie d4 c }
  }
}
```



Las ligaduras *L.v.* (*laissez vibrer*, dejar vibrar) indican que las notas no se deben apagar al final. Se usan en la notación para piano, arpa y otros instrumentos de cuerda y percusión. Se pueden introducir de la siguiente manera:

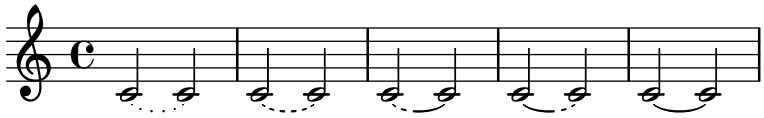
```
<c' f' g'>1\laissezVibrer
```



Es posible hacer manualmente que las ligaduras de unión se curven hacia arriba o hacia abajo; véase Sección 36.1 [Dirección y posición], página 783.

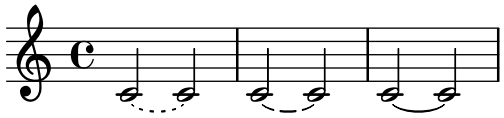
Se puede hacer que las ligaduras de expresión sean discontinuas, punteadas o una combinación de continuas y discontinuas.

```
\relative c' {
  \tieDotted
  c2~ 2
  \tieDashed
  c2~ 2
  \tieHalfDashed
  c2~ 2
  \tieHalfSolid
  c2~ 2
  \tieSolid
  c2~ 2
}
```



Se pueden especificar patrones de discontinuidad personalizados:

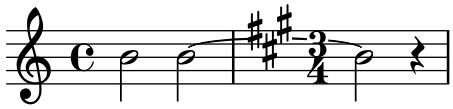
```
\relative c' {
  \tieDashPattern 0.3 0.75
  c2~ 2
  \tieDashPattern 0.7 1.5
  c2~ 2
  \tieSolid
  c2~ 2
}
```



Las definiciones de patrones de discontinuidad para las ligaduras de unión tienen la misma estructura que las definiciones de patrones de discontinuidad para las ligaduras de expresión. Para ver más información acerca de los patrones de discontinuidad complejos, consulte los fragmentos de código bajo Sección 3.2.1 [Ligaduras de expresión], página 171.

Sobreescriba las propiedades de disposición *whiteout* y *layer* de los objetos que puedan causar una discontinuidad en las ligaduras de unión.

```
\relative {
  \override Tie.layer = -2
  \override Staff.TimeSignature.layer = -1
  \override Staff.KeySignature.layer = -1
  \override Staff.TimeSignature.whiteout = ##t
  \override Staff.KeySignature.whiteout = ##t
  b'2 b~
  \time 3/4
  \key a \major
  b r4
}
```



Instrucciones predefinidas

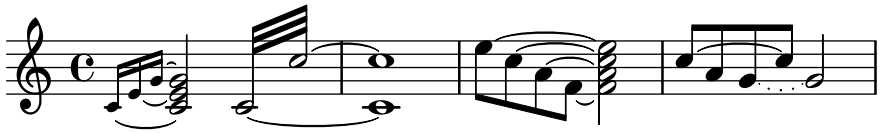
`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`, `\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

Fragmentos de código seleccionados

Uso de ligaduras en los arpegios

En ocasiones se usan ligaduras de unión para escribir los arpegios. En este caso, las dos notas ligadas no tienen que ser consecutivas. Esto se puede conseguir estableciendo la propiedad `tieWaitForNote` al valor `#t`. La misma funcionalidad es de utilidad, por ejemplo, para ligar un trémolo a un acorde, pero en principio también se puede usar para notas normales consecutivas.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



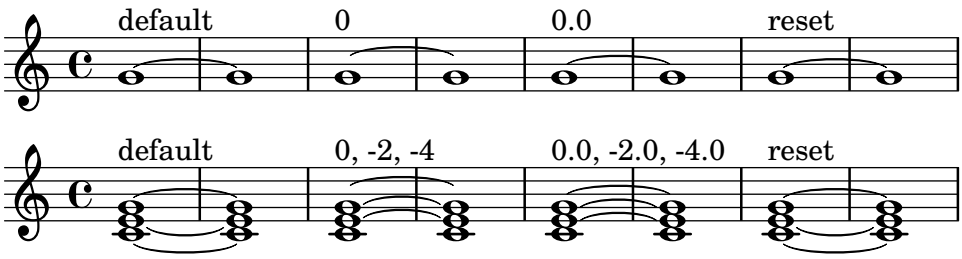
Grabado manual de las ligaduras

Se puede grabar a mano una ligadura modificando la propiedad `staff-position` (un desplazamiento) del objeto gráfico `Tie`; si hay más de una ligadura en el mismo momento musical, se pueden ajustar manualmente cambiando la propiedad `tie-configuration` del objeto `TieColumn`.

El desplazamiento indica la distancia a partir de la tercera línea del pentagrama en espacios de pentagrama, y la dirección puede ser 1 (hacia arriba) o -1 (hacia abajo).

Observe que LilyPond distingue entre valores exactos e inexactos para el primer número. Si se está usando un valor exacto (p.ej., un entero o una fracción como `(/ 4 5)`), el valor sirve como una posición vertical aproximada que después se ajusta por parte de LilyPond para que la ligadura evite las líneas del pentagrama. Si se usa un valor inexacto, como un número de coma flotante, se toma como la posición vertical precisa sin más ajustes posteriores.

```
\relative c' {
  <>^"default"
  g'1 ^~ g

  <>^"0"
  \once \override Tie.staff-position = 0
  g1 ^~ g

  <>^"0.0"
  \once \override Tie.staff-position = 0.0
  g1 ^~ g

  <>^"reset"
  \revert Tie.staff-position
  g1 ^~ g
}

\relative c' {
  \override TextScript.outside-staff-priority = ##f
  \override TextScript.padding = 0

  <>^"default"
  <c e g>1~ <c e g>

  <>^"0, -2, -4"
  \override TieColumn.tie-configuration =
    #'((0 . 1) (-2 . 1) (-4 . 1))
  <c e g>1~ <c e g>

  <>^"0.0, -2.0, -4.0"
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>1~ <c e g>

  <>^"reset"
  \override TieColumn.tie-configuration = ##f
  <c e g>1~ <c e g>
}
```



Véase también

Glosario musical: Sección “tie” en *Glosario Musical*, Sección “laissez vibrer” en *Glosario Musical*.

Referencia de la notación: Sección 3.2.1 [Ligaduras de expresión], página 171, Sección 2.3.6 [División automática de las notas], página 97.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*, Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “LaissezVibrerTie” en *Referencia de Funcionamiento Interno*, Sección “LaissezVibrerTieColumn” en *Referencia de Funcionamiento Interno*, Sección “TieColumn” en *Referencia de Funcionamiento Interno*, Sección “Tie” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Un cambio de pentagrama cuando hay una ligadura activa no produce una ligadura inclinada.

Los cambios de clave o de octava durante una ligadura de unión no están realmente bien definidos. En estos casos puede ser preferible una ligadura de expresión.

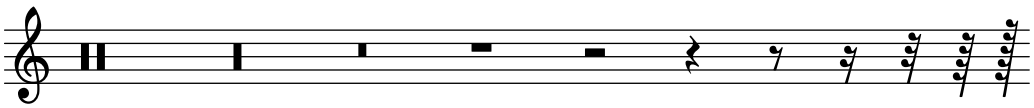
2.2 Escritura de silencios

Los silencios se escriben como parte de la música dentro de las expresiones musicales.

2.2.1 Silencios

Los silencios se introducen como notas con el nombre `r`. Las duraciones mayores que la redonda utilizan las instrucciones predefinidas que se muestran aquí:

```
\new Staff {
  % These two lines are just to prettify this example
  \time 16/1
  \omit Staff.TimeSignature
  % Print a maxima rest, equal to four breves
  r\maxima
  % Print a longa rest, equal to two breves
  r\longa
  % Print a breve rest
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
}
```



Los silencios de un compás, centrados en medio del compás, se deben hacer con silencios multicompás. Se pueden usar para un solo compás así como para muchos compases, y se tratan en Sección 2.2.3 [Silencios de compás completo], página 72.

Para especificar explícitamente la posición vertical de un silencio, escriba una nota seguida de `\rest`. Se colocará un silencio en la posición en que debería aparecer la nota. Esto posibilita la aplicación manual precisa de formato a la música polifónica, ya que el formateador automático de colisiones de silencios no mueve estos silencios.

```
\relative { a'4\rest d4\rest }
```

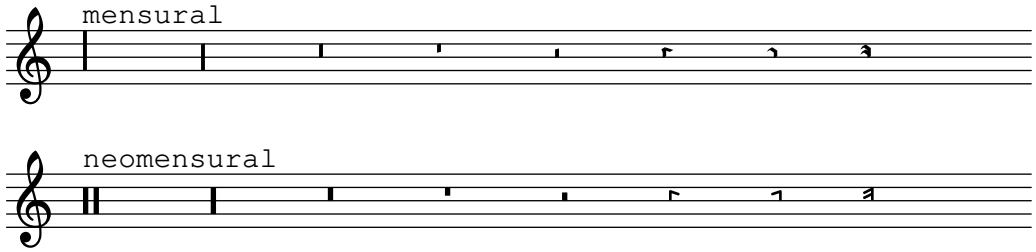


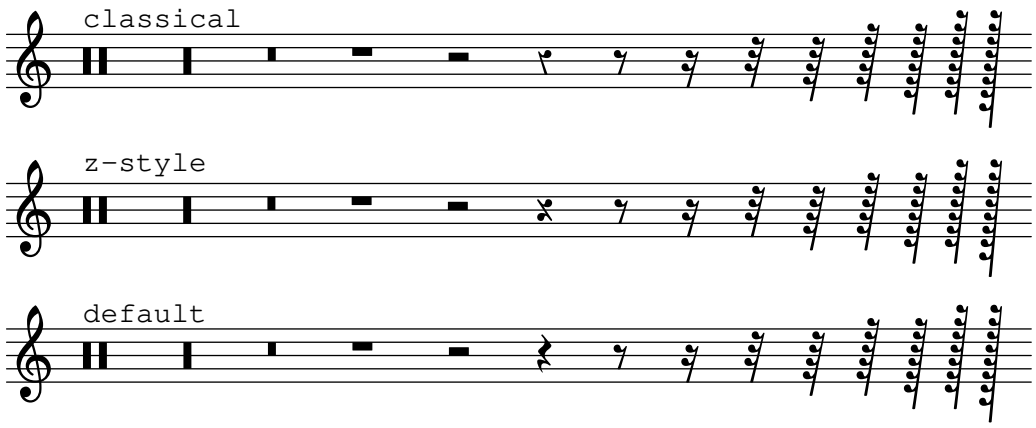
Fragmentos de código seleccionados

Estilos de silencios

Los silencios se pueden imprimir en distintos estilos.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
  r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}
```





Véase también

Glosario musical: Sección “breve” en *Glosario Musical*, Sección “longa” en *Glosario Musical*, Sección “maxima” en *Glosario Musical*.

Referencia de la notación: Sección 2.2.3 [Silencios de compás completo], página 72.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Rest” en *Referencia de Funcionamiento Interno*.

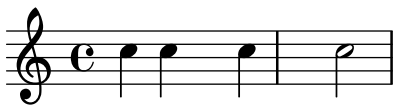
Advertencias y problemas conocidos

No existe un límite fundamental respecto de las duraciones de los silencios (tanto para el más corto como para el más largo), pero el número de glifos es limitado: hay silencios desde 1/1024 de redonda (sin nombre en español) hasta la máxima (8 redondas).

2.2.2 Silencios invisibles

Existen dos formas de silencios invisibles: el *silencio de separación* que lleva el nombre ‘s’, y la instrucción `\skip`. El silencio de separación es una nota que no produce ninguna salida impresa. Como cualquier otra nota o silencio, su duración fija el valor por omisión de las notas siguientes.

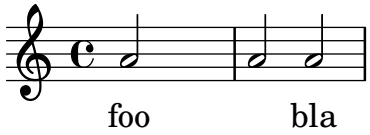
```
\relative c'' {
  c4 c s c |
  s2 c |
}
```



Los silencios de separación solo están disponible en el modo de notas y en el modo de acordes. En otras situaciones, por ejemplo, cuando se introduce la letra, se usa la instrucción `\skip` para producir un desplazamiento de una cierta magnitud temporal. La instrucción `\skip` acepta una duración explícita o bien un fragmento de música como argumento y se salta la duración de ese argumento. La duración del `\skip` se ignora si las sílabas de la letra toman sus duraciones de las notas de una melodía asociada, a través de `\addlyrics` o de `\lyricsto`.

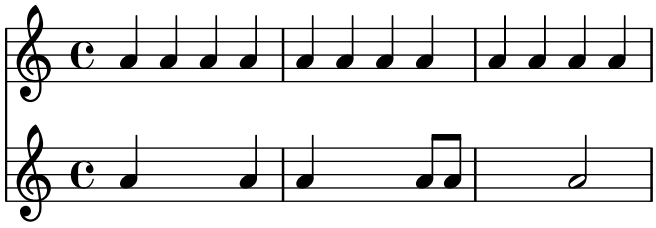
```
<<
  {
    a'2 \skip2 a'2 a'2
  }
  \new Lyrics {
    \lyricmode {
```

```
      foo2 \skip 1 bla2
    }
  }
>>
```



Cuando el argumento de \skip es una expresión musical, la duración por omisión de la nota siguiente se establece implícitamente por la última nota del argumento. Sin embargo, para preservar la compatibilidad hacia atrás con la implementación en desuso de la instrucción \skip, cualquier duración numérica no afecta a la duración de la nota siguiente.

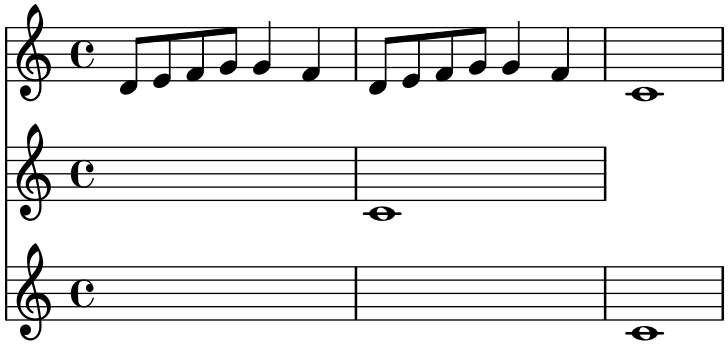
```
<<
  {
    \repeat unfold 12 { a'4 }
  }
  {
    a'4 \skip 2 a' |
    a'4 \skip { a'8 a' a' a' } a' a' |
    s2 a'
  }
>>
```



La instrucción \skip preserva el efecto de una instrucción unfoldRepeats que la encierre, a diferencia de la función de Scheme skip-of-length.

```
MyCadenza = \fixed c' {
  \repeat volta 2 {
    d8 e f g g4 f4
  }
}

music = <<
  \new Staff {
    \MyCadenza
    c'1
  }
  \new Staff {
    #(skip-of-length MyCadenza)
    c'1
  }
  \new Staff {
    \skip \MyCadenza
    c'1
```

```
  }
>>

\unfoldRepeats \music
```



La instrucción \skip tan solo desplaza un tiempo musical; no produce ninguna salida, de ningún tipo.

```
% This is valid input, but does nothing
{ \skip 1 \skip1 \skip 1 }
```

Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 7.1.4 [Notas ocultas], página 291, Sección 36.7 [Visibilidad de los objetos], página 794.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SkipMusic” en *Referencia de Funcionamiento Interno*.

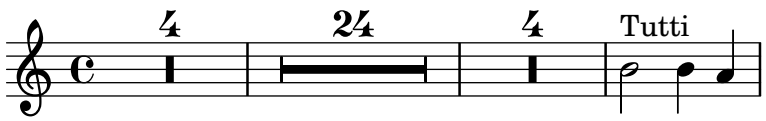
Advertencias y problemas conocidos

Como la duración y los argumentos musicales de la instrucción \skip afectan de manera diferente a la duración de la música que sigue, se considera una buena práctica escribir explícitamente la duración de la música que esté inmediatamente a continuación de esta instrucción.

2.2.3 Silencios de compás completo

Los silencios de uno o más compases completos se introducen como notas con el nombre R en mayúscula. Su duración se escribe de forma idéntica a la que se utiliza para las notas, incluso con la capacidad de utilizar factores multiplicadores de la duración, como se explica en Sección 2.1.3 [Escalar las duraciones], página 62:

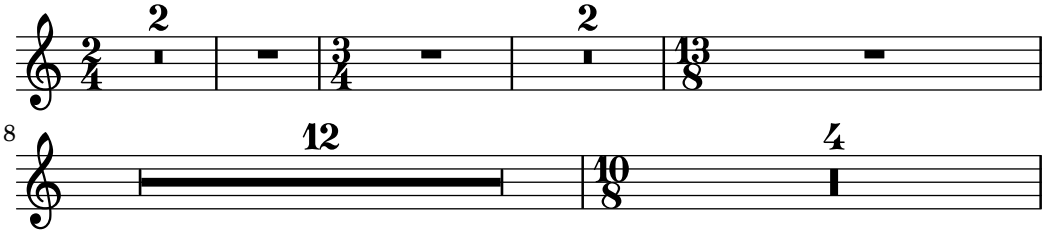
```
% Rest measures contracted to single measure
\compressMMRests {
  R1*4
  R1*24
  R1*4
  b'2^"Tutti" b'4 a'4
}
```



El ejemplo anterior también muestra cómo comprimir varios compases vacíos seguidos, como se explica en Sección 6.3.4 [Compresión de los compases vacíos], página 277.

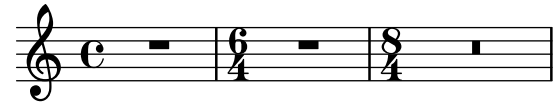
La duración de un silencio multicompás siempre debe ser igual a la longitud de uno o varios compases. Por tanto, algunas indicaciones de compás requieren la utilización de puntillos o fracciones:

```
\compressMMRests {
  \time 2/4
  R1 | R2 |
  \time 3/4
  R2. | R2.*2 |
  \time 13/8
  R1*13/8 | R1*13/8*12 |
  \time 10/8
  R4*5*4 |
}
```



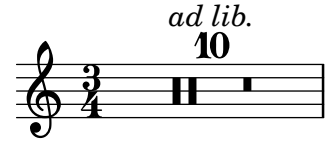
Un silencio de un compás completo se imprime como un silencio de redonda o de breve, centrado en el compás, según el tipo de compás vigente.

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```



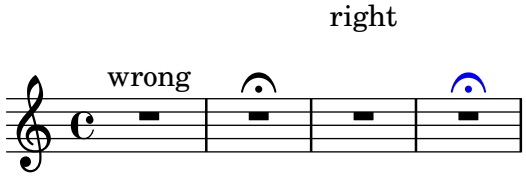
Se pueden añadir elementos de marcado a los silencios multicompás.

```
\compressMMRests {
  \time 3/4
  R2.*10^\markup { \italic "ad lib." }
}
```



Nota: Los elementos de marcado y articulaciones que se añaden a los silencios multicompás son objetos del tipo `MultiMeasureRestText` y `MultiMeasureRestScript`, no `TextScript` y `Script`. Las sobreescrituras de propiedades deben ir dirigidas hacia el objeto correcto, o se ignorarán. Véase el ejemplo siguiente:

```
% This fails, as the wrong object name is specified
\override TextScript.padding = 5
\override Script.color = #blue
R1^"wrong"
R1\fermata
% This is the correct object name to be specified
\override MultiMeasureRestText.padding = 5
\override MultiMeasureRestScript.color = #blue
R1^"right"
R1\fermata
```



Cuando un silencio multicompás sigue inmediatamente al establecimiento de un compás parcial con \partial, es posible que no se emitan las advertencias correspondientes de comprobación de compás.

Instrucciones predefinidas

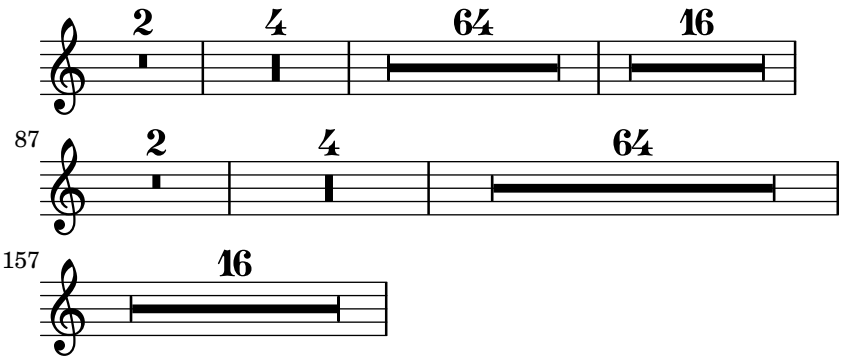
\textLengthOn, \textLengthOff, \compressMMRests.

Fragmentos de código seleccionados

Control de la anchura de un silencio multicompás

Los silencios de varios compases tienen una longitud que concuerda con su duración total, que está bajo el control de la propiedad space-increment del grob MultiMeasureRest; su valor predeterminado es 2.

```
\relative c' {
  \omit Staff.TimeSignature
  \compressEmptyMeasures

  R1*2 R1*4 R1*64 R1*16 \break
  \override MultiMeasureRest.space-increment = 4
  R1*2 R1*4 R1*64 R1*16
}

\layout {
  ragged-right = ##t
}
```

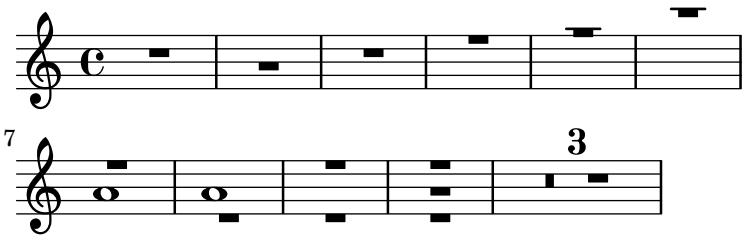


Posicionar los silencios multicompás

A diferencia de los silencios normales, no existe una instrucción predefinida para modificar la posición predeterminada de un símbolo de silencio multicompás sobre el pentagrama, adjuntándolo a una nota, independientemente de cuál sea su forma. Sin embargo, en la música polifónica los silencios multicompás de las voces de numeración par e impar están separados verticalmente.

Este fragmento de código muestra la forma de controlar la colocación de los silencios multicompás.

```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position = 6 R1
  \revert MultiMeasureRest.staff-position
  \break

  % Odd-numbered voices are under the top line.
  << { R1 } \\ { a1 } >>
  % Even-numbered voices are under the bottom line.
  << { a1 } \\ { R1 } >>
  % Multi-measure rests in both voices remain separate.
  << { R1 } \\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override or tweak.
  << { R1 } \\ { R1 } \\ { \tweak staff-position -2 R1 } >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 } \\
    \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```



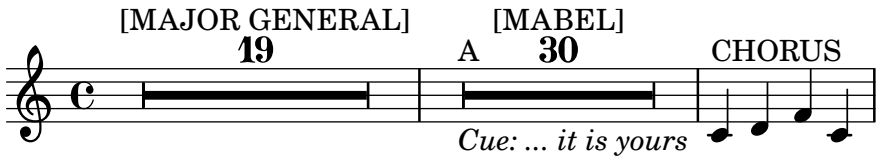
Marcado de silencios multicompás

Los elementos de marcado aplicados a un silencio multicompás se centran encima o debajo de éste. Los elementos de marcado extensos que se adjuntan a silencios multicompás no producen

la expansión del compás. Para expandir un silencio multicompás de forma que quepa todo el marcado, utilice un acorde vacío con un marcado aplicado antes del silencio multicompás.

El texto aplicado a un siencio sparador de esta forma se alinea por la izquierda a la posición en que la nota estaría situada dentro del compás, pero si la longitud del compás está determinada por la longitud del texto, éste aparecerá centrado.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



Véase también

Glosario musical: Sección “multi-measure rest” en *Glosario Musical*.

Referencia de la notación: Capítulo 2 [Duraciones], página 53, Sección 2.1.3 [Escalar las duraciones], página 62, Sección 6.3.4 [Compresión de los compases vacíos], página 277, Capítulo 8 [Texto], página 311, Sección 8.2 [Formatear el texto], página 323, Sección 8.1.2 [Inscripciones textuales], página 314.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MultiMeasureRest” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestNumber” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestScript” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestText” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

En las digitaciones encima de silencios multicompás (p. ej., `R1*10-4`), la cifra de la digitación puede chocar con el número del compás.

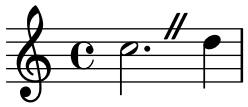
No exite ninguna forma de condensar automáticamente varios silencios normales en un solo silencio multicompás.

Los silencios multicompás no intervienen en la detección de colisiones de silencios.

2.2.4 Cesuras

La instrucción `\caesura` obedece a un silencio no medido: normalmente una corta interrupción del sonido que no reduce el valor de la nota anterior.

```
\fixed c'' { c2. \caesura d4 }
```



En los cánticos e himnos, \caesura puede servir de forma más general como una división de la frase; para más información, véanse las referencias listadas al final de esta sección. Para un corte del sonido que reduce el valor de la nota anterior, véase Sección 3.2.3 [Marcas de respiración], página 177.

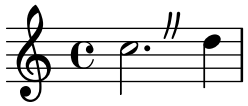
Después de una \caesura pueden ponerse articulaciones para indicar la duración relativa o la importancia de la interrupción; estos crean grobs del tipo CaesuraScript.

```
\fixed c'' { c2. \caesura \fermata d4 }
```



De manera predeterminada, \caesura crea un objeto gráfico BreathingSign. El elemento breath del contexto caesuraType controla qué símbolo de \caesura crea, de entre varios símbolos predefinidos. Véase Sección B.14 [Lista de signos de respiración], página 936.

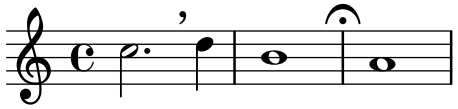
```
\fixed c'' {
  \set Score.caesuraType = #'((breath . curvedcaesura))
  c2. \caesura d4
}
```



Para designar que uno o más objetos gráficos CaesuraScript se creen como la parte normal de una cesura sin articular, fije el elemento scripts de la propiedad de contexto caesuraType (también se pueden adjuntar inscripciones adicionales como articulaciones). En combinación con el elemento breath, las inscripciones listadas en el elemento script se adjuntan al BreathingSign; en otro caso, si está presente una BarLine, se adjuntan a ella.

El valor de la propiedad de contexto caesuraTypeTransform se puede establecer a una función de Scheme para activar un cierto grado de adaptación automática. El generador de función at-bar-line-substitute-caesura-type contempla estilos en que la notación es diferente sobre una línea divisoria.

```
\fixed c' {
  \set Score.caesuraType =
    #'((breath . spacer)
       (scripts . (outsidecomma)))
  \set Score.caesuraTypeTransform =
    #(at-bar-line-substitute-caesura-type
      '((scripts . (fermata))))
  c'2. \caesura d'4
  b1 \caesura
  a1
}
```



Instrucciones predefinidas

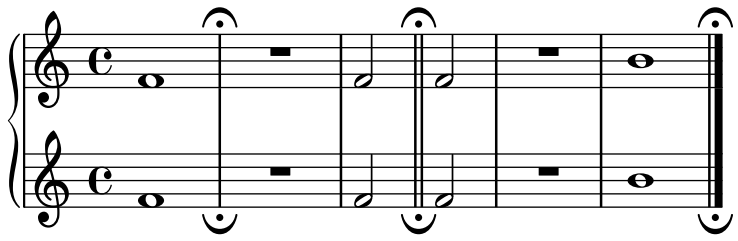
`\caesura`.

Fragmentos de código seleccionados

Colocar calderones opuestos sobre la barra de compás

Este fragmento es una demostración de una instrucción que imprime calderones encima y debajo de la línea divisoria al mismo tiempo. Si no hay ninguna barra de compás, añade una doble barra. Semánticamente, la instrucción codifica una cesura más larga de lo normal, lo que se puede considerar un mal uso dependiendo de la situación.

```
twoWayFermata = {
  \once \set Staff.caesuraType = #'((underlying-bar-line . "||"))
  \once \set Staff.caesuraTypeTransform = ##f
  \caesura ^\fermata _\fermata
}

music = {
  f'1 \twoWayFermata
  R1
  f'2 \twoWayFermata f'2
  R1
  b'1 \twoWayFermata \fine
}

\new GrandStaff <<
  \new Staff \music
  \new Staff \music
>>
```



Véase también

Glosario musical: Sección “caesura” en *Glosario Musical*.

Referencia de la notación: Sección 3.2.3 [Marcas de respiración], página 177, Sección 17.4.4 [Divisiones], página 554, Sección 17.5.5 [Líneas divisorias del canto kievano], página 564, Sección 9.7.4 [Barras de frase en himnos], página 411.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

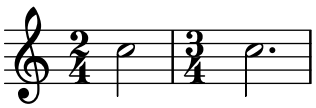
Referencia de funcionamiento interno: Sección “BreathingSign” en *Referencia de Funcionamiento Interno*, Sección “Caesura_engraver” en *Referencia de Funcionamiento Interno*, Sección “CaesuraEvent” en *Referencia de Funcionamiento Interno*, Sección “CaesuraScript” en *Referencia de Funcionamiento Interno*, Sección “Tunable context properties” en *Referencia de Funcionamiento Interno*.

2.3 Impresión de las duraciones

2.3.1 Indicación de compás

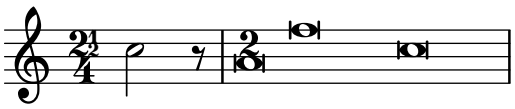
Para poner una indicación de compás básica, use la instrucción `\time` con la fracción como argumento.

```
\time 2/4
c''2

\time 3/4
c''2.
```



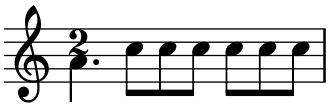
Las indicaciones de compás fraccionales y los denominadores más largos que una redonda, requieren sintaxis de Scheme.

```
\time #'(5/2 . 4)
c''2 r8

\override Timing.TimeSignature.denominator-style = #'note
\time #'(2 . 1/2)
f''\breve c''
```



Se pueden usar otros denominadores poco usuales para añadir puntillos en el estilo 'nota sobre nota'; sin embargo, para aprovechar al máximo las posibilidades que se explican en el lugar oportuno, en tales casos se recomienda usar una fracción convencional para `\time` y sobreescribir `TimeSignature.time-signature`.

```
\override Timing.TimeSignature.denominator-style = #'note
\once \override Timing.TimeSignature.time-signature = #'(2 . 8/3)
\time 6/8
c''8 8 8 8 8 8
```

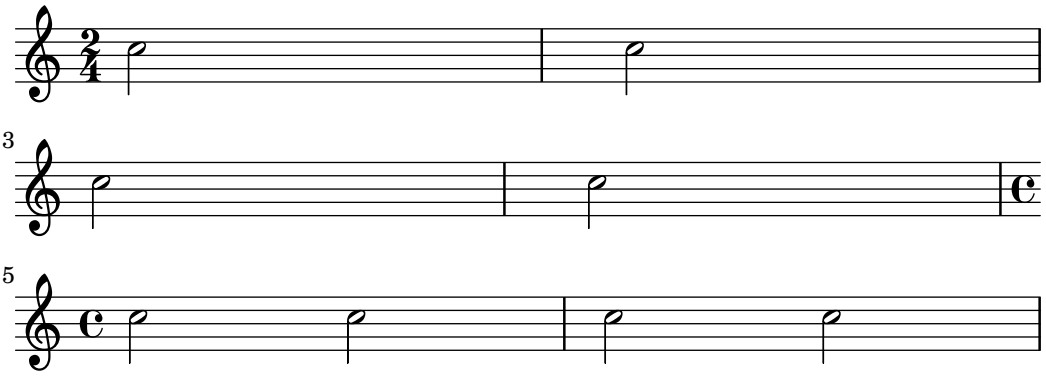


Los cambios de indicación de compás a mitad de un compás aparecen explicados en Sección 2.3.3 [Anacrusas], página 90.

La indicación de compás se imprime al comienzo de una pieza y siempre que hay un cambio de compás. Si se produce un cambio al final de una línea, se imprime una indicación de advertencia en dicho lugar. Se puede modificar este comportamiento predeterminado, véase Sección 36.7 [Visibilidad de los objetos], página 794.

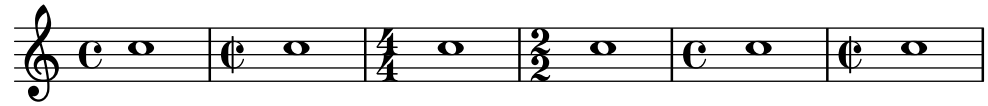
```
\relative c'' {
  \time 2/4
  c2 c
  \break
  c c
```

```
  \break
  \time 4/4
  c c c c
}
```



El símbolo de compás que se usa en 2/2 y 4/4 se puede cambiar a un estilo numérico:

```
\relative c'' {
  % Default style
  \time 4/4 c1
  \time 2/2 c1
  % Change to numeric style
  \numericTimeSignature
  \time 4/4 c1
  \time 2/2 c1
  % Revert to default style
  \defaultTimeSignature
  \time 4/4 c1
  \time 2/2 c1
}
```

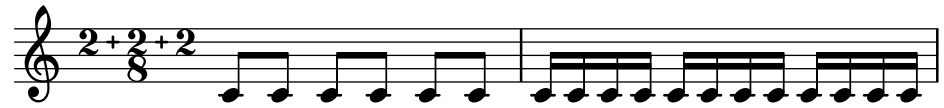


Las indicaciones de compás de la música mensural se tratan en Sección 17.3.3 [Indicaciones de compás de la música mensural], página 546.

Indicaciones de compás complejo o de amalgama

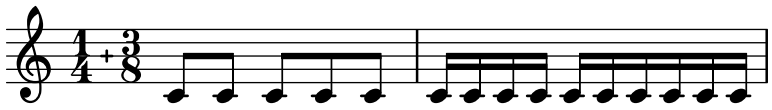
La función `\time` contempla diversos compases complejos o de amalgama representados como expresiones de Scheme. Las fracciones se representan como parejas, (*numerador . denominador*), donde el denominador siempre es un número. El numerador es un número o una lista de dos o más números. Una lista representa una concatenación.

```
\fixed c' {
  \time #'((2 2 2) . 8)
  \repeat unfold 6 c8
  \repeat unfold 12 c16
}
```

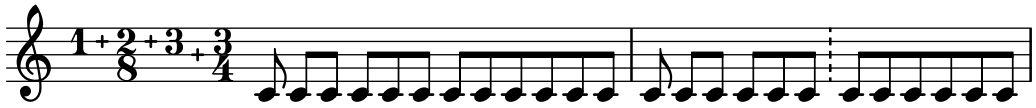


Se pueden construir compases que alternan de forma estricta una serie de fracciones, agrupándolas en una lista. Los ajustes de barrado automático se ajustarán dependiendo de los valores. Se pueden habilitar barras de compás secundarias como ayuda para aclarar los ritmos complejos.

```
\fixed c' {
  \time #'((1 . 4) (3 . 8))
  \repeat unfold 5 c8
  \repeat unfold 10 c16
}
```

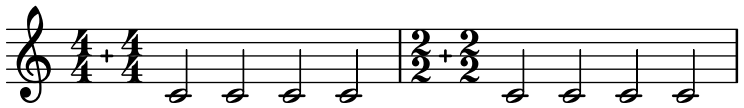


```
\fixed c' {
  \time #'(((1 2 3) . 8) (3 . 4))
  \repeat unfold 12 c8
  \submeasureBarsOn
  \repeat unfold 12 c8
}
```



De forma predeterminada, los tipos de compás que se alternan estrictamente y que consisten en 4/4 and 2/2 se imprimen en un estilo histórico que utiliza varios símbolos de C o de C partida. No es necesario usar esta interpretación de las indicaciones de compás para el compás subyacente: se puede sobreescribir `TimeSignature.time-signature` para imprimir estos símbolos al mismo tiempo que se establece un compás distinto con `\time`.

```
\fixed c' {
  \time #'((4 . 4) (4 . 4))
  \repeat unfold 4 c2
  \once \override Timing.TimeSignature.time-signature
    = #'((2 . 2) (2 . 2))
  \time 4/2
  \repeat unfold 4 c2
}
```



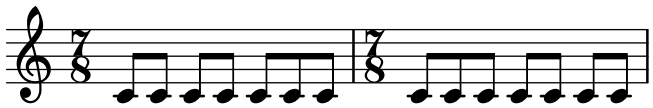
Propiedades métricas

Además de ajustar la indicación de compás que se imprime, la instrucción `\time` establece también los valores de las propiedades de contexto `beatBase`, `beatStructure`, `submeasureStructure` y `beamExceptions`. Los valores predefinidos de estas propiedades están en `scm/time-signature-settings.scm`.

El valor predeterminado de `beatStructure` puede sobreescribirse dentro de la propia instrucción `\time` escribiéndolo como primer argumento opcional:

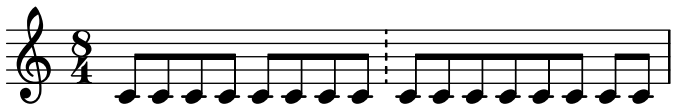
```
\score {
  \new Staff {
    \relative {
```

```
      \time 2,2,3 7/8
      \repeat unfold 7 { c'8 } |
      \time 3,2,2 7/8
      \repeat unfold 7 { c8 } |
    }
  }
}
```



De forma alternativa, los valores predeterminados de estas propiedades de las métricas como un grupo para una indicación de compás dada. Los nuevos valores tienen efecto en una instrucción `\time` posterior con ese mismo compás:

```
\score {
  \new Staff {
    \relative c' {
      \overrideTimeSignatureSettings
        8/4             % timeSignature
        #1/4            % beatBase
        #'((2 2) (3 1)) % beatStructure, measureStructure
        #'()            % beamExceptions
      \time 8/4
      \submeasureBarsOn
      \repeat unfold 16 { c8 } |
    }
  }
}
```



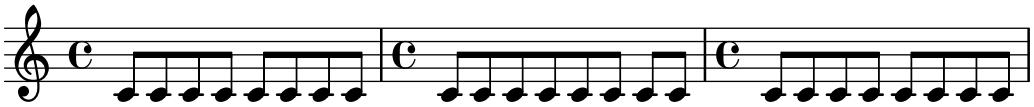
`\overrideTimeSignatureSettings` toma cuatro argumentos:

1. *`timeSignature`*, una fracción que describe el tipo de compás al que se aplican estos valores.
2. *`beatBase`*, la duración que corrresponde a una unidad de *beatStructure*.
3. *`structure`*, una lista de Scheme que describe la longitud de cada pulso del compás en unidades de *beatBase* (es decir, `beatStructure`), agrupadas opcionalmente con listas internas que describen subdivisiones mayores del compás (es decir, `submeasureStructure`).
4. *`beamExceptions`*, una lista-A que contiene cualesquiera reglas de barrado para el tipo de compás que vayan más allá de terminar en cada pulso, como se describe en Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104.

Los valores modificados de las propiedades predeterminadas del tipo de compás se pueden restaurar a los valores originales:

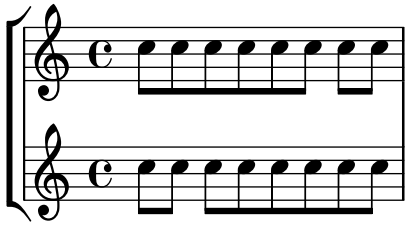
```
\score {
  \relative {
    \repeat unfold 8 { c'8 } |
    \overrideTimeSignatureSettings
      4/4       % timeSignature
      #1/4      % beatBase
```

```
      3,1         % beatStructure
      #'()        % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
    \revertTimeSignatureSettings 4/4
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}
```



Se pueden establecer diferentes valores de las propiedades predeterminadas del tipo de compás para los distintos pentagramas mediante el uso de notación polimétrica (véase Sección 2.3.5 [Notación polimétrica], página 93).

```
\score {
  \new StaffGroup <<
    \new Staff {
      \overrideTimeSignatureSettings
        4/4         % timeSignature
        #1/4        % beatBase
        3,1         % beatStructure
        #'()        % beamExceptions
      \time 4/4
      \repeat unfold 8 {c''8}
    }
    \new Staff {
      \overrideTimeSignatureSettings
        4/4         % timeSignature
        #1/4        % beatBase
        1,3         % beatStructure
        #'()        % beamExceptions
      \time 4/4
      \repeat unfold 8 {c''8}
    }
  >>
  \layout {
    \enablePerStaffTiming
  }
}
```



Otro método para cambiar estas propiedades métricas que evita la reimpresión de la indicación de compás en el momento del cambio, se muestra en Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104.

Instrucciones predefinidas

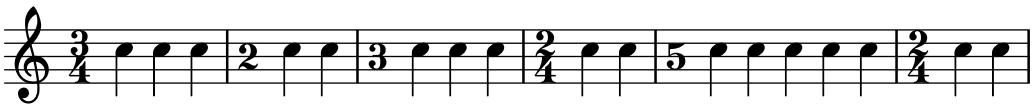
`\numericTimeSignature`, `\defaultTimeSignature`.

Fragmentos de código seleccionados

Indicación de compás imprimiendo solo el numerador (en lugar de la fracción)

A veces, la indicación de compás no debe imprimir la fracción completa (p.ej. 7/4), sino solo el numerador (el dígito 7 en este caso). Esto se puede hacer fácilmente utilizando `\override Staff.TimeSignature.style = #'single-number` para cambiar el estilo permanentemente. Usando `\revert Staff.TimeSignature.style`, se puede revertir el cambio. Para aplicar el estilo de un número único a una sola indicación de compás, use `\tweak`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-number
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-number style only for the next time signature
  \tweak style #'single-number \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```



Véase también

Glosario musical: Sección “time signature” en *Glosario Musical*

Referencia de la notación: Sección 17.3.3 [Indicaciones de compás de la música mensural], página 546, Sección 2.3.5 [Notación polimétrica], página 93, Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104, Sección 2.6.3 [Gestión del tiempo], página 155.

Archivos instalados: `scm/time-signature-settings.scm`.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

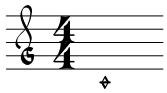
Referencia de funcionamiento interno: Sección “TimeSignature” en *Referencia de Funcionamiento Interno*, Sección “Timing_translator” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Ni `\numericTimeSignature` ni `\defaultTimeSignature` tienen efecto alguno cuando se usan dentro de un `MensuralStaff`. Para usar estas indicaciones de comás modernas dentro de un `MensuralStaff`, selecciónelas desde el momento de la creación del contexto:

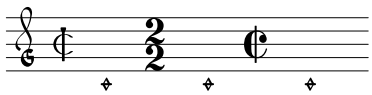
```
\new MensuralStaff \with { \numericTimeSignature } {
```

```
  c'1
}
```



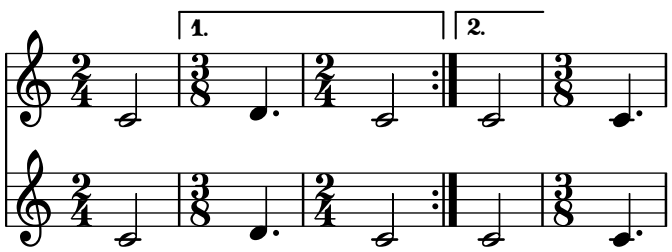
o bien use una sobreescritura \override explícita:

```
\new MensuralStaff {
  \time 2/2
  c'1
  \override MensuralStaff.TimeSignature.style = #'numbered
  \time 2/2
  c'
  \override MensuralStaff.TimeSignature.style = #'default
  \time 2/2
  c'
}
```



Si hay más de un pentagrama, y una indicación de compás está al comienzo de una casilla de repetición de primera vez pero no en la de segunda vez, es necesario ayudar a LilyPond a sincronizar la situación añadiendo una indicación de compás explícita pero invisible en la casilla de segunda vez.

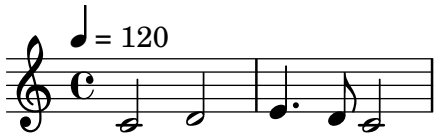
```
music = {
  \repeat volta 2 {
    \time 2/4 c'2 |
    \alternative {
      \volta 1 {
        \time 3/8 d'4. |
        \time 2/4 c'2 | }
      \volta 2 {
        \once \omit Staff.TimeSignature
        \time 2/4 c'2 |
      }
    }
  }
  \time 3/8 c'4. |
}

<<
  \new Staff \music
  \new Staff \music
>>
```



2.3.2 Indicaciones metronómicas

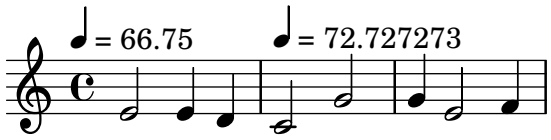
Es muy sencillo escribir una indicación metronómica básica:

```
\relative {
  \tempo 4 = 120
  c'2 d
  e4. d8 c2
}
```



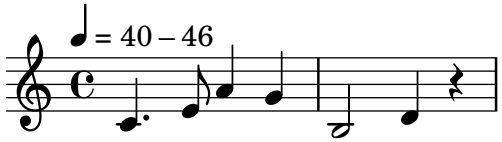
El tempo indicado no tiene que ser un número entero, pero debe ser un número exacto. El formateador por omisión redondea al 0.25 más cercano; esto se controla a través de la propiedad de contexto `tempoCountPrecision`.

```
tempoI = 100
\fixed c' {
  \tempo 4 = #(* tempoI 2/3)   % 66.66...
  e2 e4 d
  \once \set Timing.tempoCountPrecision = #1/1000000
  \tempo 4 = #(* tempoI 8/11)  % 72.72...
  c2 g2
  g4 e2 f4
}
```



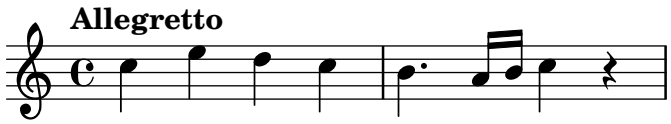
También se pueden imprimir indicaciones metronómicas como un intervalo entre dos números:

```
\relative {
  \tempo 4 = 40 - 46
  c'4. e8 a4 g
  b,2 d4 r
}
```



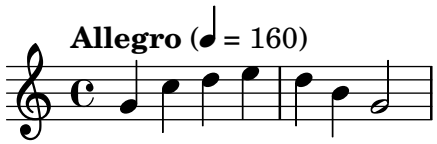
Se pueden usar indicaciones de tempo con texto:

```
\relative {
  \tempo "Allegretto"
    c''4 e d c
    b4. a16 b c4 r4
}
```



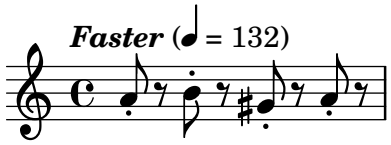
La combinación de una indicación metronómica y un texto hace que la marca de metrónomo se encierre entre paréntesis automáticamente:

```
\relative {
  \tempo "Allegro" 4 = 160
  g'4 c d e
  d4 b g2
}
```



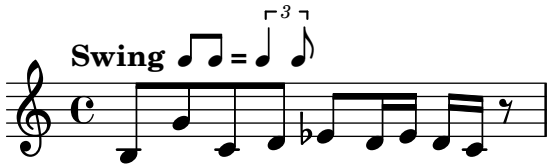
En general, el texto puede ser cualquier objeto de marcado:

```
\relative {
  \tempo \markup { \italic Faster } 4 = 132
  a'8-. r8 b-. r gis-. r a-. r
}
```



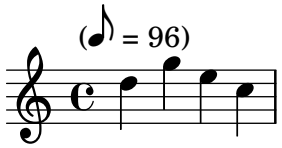
Una instrucción de marcado que resulta especialmente útil es `\rhythm`, que imprime un patrón rítmico. Véase Sección A.1.4 [Markup for music and musical symbols], página 856.

```
\relative {
  \tempo \markup {
    Swing
    \hspace #0.4
    \rhythm { 8[ 8] } = \rhythm { \tuplet 3/2 { 4 8 } }
  }
  b8 g' c, d ees d16 ees d c r8
}
```



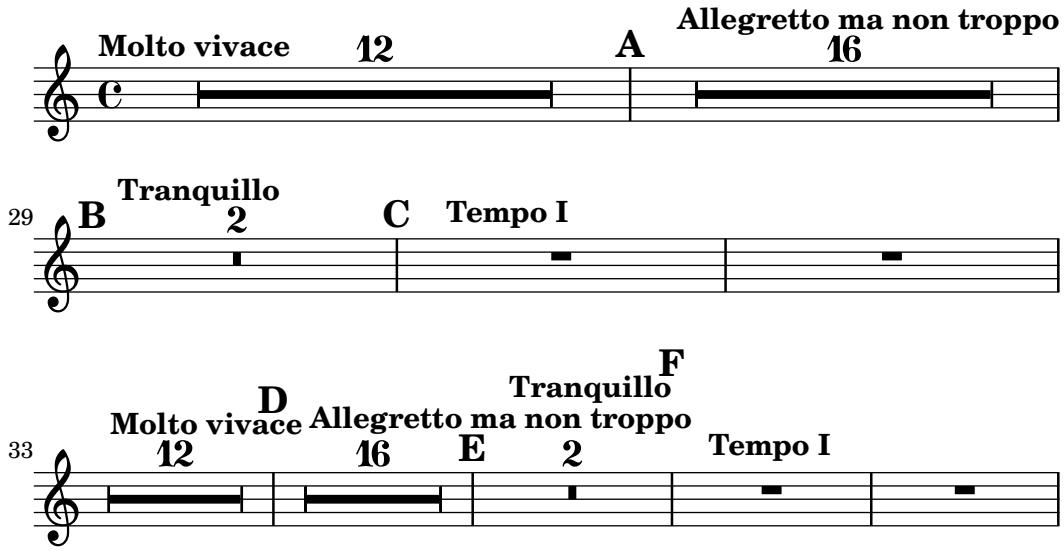
Se puede escribir una indicación metronómica entre paréntesis sin ninguna indicación textual, escribiendo una cadena vacía en la entrada:

```
\relative {
  \tempo "" 8 = 96
  d''4 g e c
}
```



En una particella de un instrumento con períodos de silencio largos (véase Sección 2.2.3 [Silencios de compás completo], página 72) ocurre con bastante frecuencia que las indicaciones de tempo y las llamadas de ensayo y textuales a veces se suceden muy seguidas entre sí. La instrucción `\markLengthOn` añade espacio adicional horizontal (y vertical) para evitar que estas marcas se superpongan en el sentido horizontal, produciendo muchas veces un apilamiento vertical no deseado. Use `\markLengthOff` para volver al comportamiento por omisión: ignorar las marcas por parte del algoritmo de espaciado horizontal.

```
\compressMMRests {
  \markLengthOn
  \tempo "Molto vivace"
  R1*12
  \mark \default \tempo "Allegretto ma non troppo"
  R1*16
  \mark \default \tempo "Tranquillo"
  R1*2
  \markLengthOff
  \mark \default \tempo "Tempo I"
  R1 R1 \break

  \markLengthOff
  \tempo "Molto vivace"
  R1*12
  \mark \default \tempo "Allegretto ma non troppo"
  R1*16
  \mark \default \tempo "Tranquillo"
  R1*2
  \mark \default \tempo "Tempo I"
  R1 R1
}
```

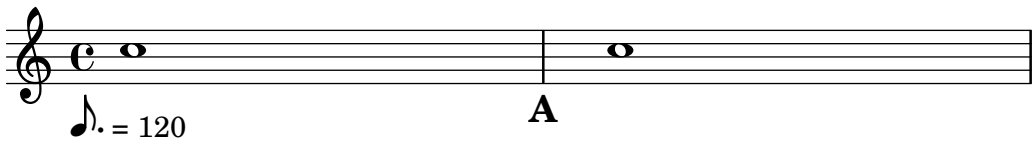


Fragmentos de código seleccionados

Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama

De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente la propiedad `direction` de `MetronomeMark` o de `RehearsalMark`.

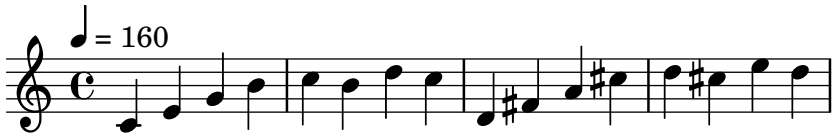
```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Cambiar el tempo sin indicación metronómica

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:

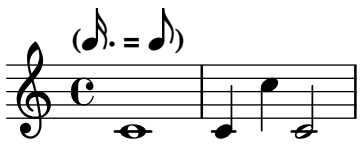
```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Crear indicaciones metronómicas en modo de marcado

Se pueden crear indicaciones metronómicas nuevas en modo de marcado, pero no cambian el tempo en la salida MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
      \smaller \general-align #Y #DOWN \note { 16. } #UP
      " = "
      \smaller \general-align #Y #DOWN \note { 8 } #UP
      )
    }
  }
  c1
  c4 c' c,2
}
```



Para ver más detalles, consulte Sección 8.2 [Formatear el texto], página 323.

Véase también

Glosario musical: Sección “metronome” en *Glosario Musical*, Sección “metronomic indication” en *Glosario Musical*, Sección “tempo indication” en *Glosario Musical*, Sección “metronome mark” en *Glosario Musical*.

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323, Capítulo 24 [Creación de salida MIDI], página 655, Sección 2.2.3 [Silencios de compás completo], página 72.

Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno:: Sección “MetronomeMark” en *Referencia de Funcionamiento Interno*.

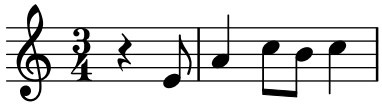
2.3.3 Anacrusas

Los compases parciales como las *anacrusas* o partes *al alzar* se escriben usando la instrucción `\partial`:

```
\partial duración
```

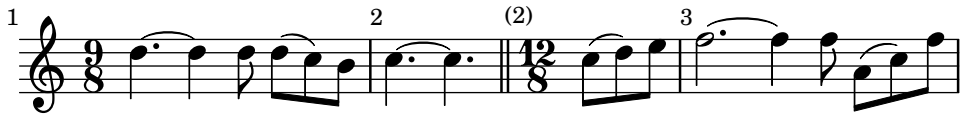
Cuando se usa `\partial` al principio de la partitura, *duración* es el tiempo de anacrusa, la longitud de la música que precede al primer compás.

```
\relative {
  \time 3/4
  \partial 4.
  r4 e'8 | a4 c8 b c4 |
}
```



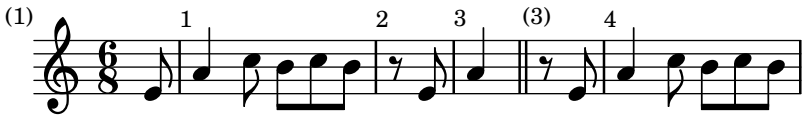
Cuando se usa \partial después del comienzo de la partitura, *duración* es la longitud *restante* del compás actual. No crea un nuevo compás con numeración.

```
\relative {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \override Score.BarNumber.break-visibility =
            #end-of-line-invisible
  \time 9/8
  d''4.~ 4 d8 d( c) b | c4.~ 4. \bar "||"
  \time 12/8
  \partial 4.
  c8( d) e | f2.~ 4 f8 a,( c) f |
}
```



La instrucción \partial *es necesaria* cuando cambia la indicación de compás en medio de un compás, pero también puede usarse sola.

```
\relative {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \override Score.BarNumber.break-visibility =
            #end-of-line-invisible
  \time 6/8
  \partial 8
  e'8 | a4 c8 b[ c b] |
  \partial 4
  r8 e,8 | a4 \bar "||"
  \partial 4
  r8 e8 | a4
  c8 b[ c b] |
}
```



Por razones técnicas, el argumento de \partial no puede ser una duración nula (como \partial 4*0).

Véase también

Glosario musical: Sección “anacrusis” en *Glosario Musical*.

Referencia de la notación: Sección 2.6.1 [Notas de adorno], página 148.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

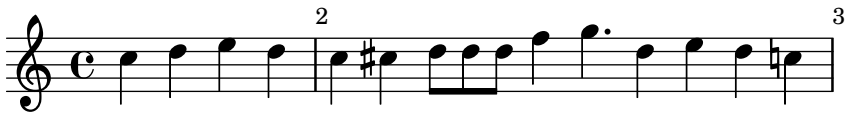
Referencia de funcionamiento interno: Sección “Timing_translator” en *Referencia de Funcionamiento Interno*.

2.3.4 Música sin compasear

En algunos tipos de música como las cadencias, puede ser deseable desactivar la demarcación automática de compases y todo lo que conlleva: numerar los compases, reinicio de las alteraciones

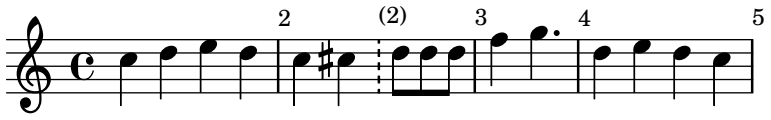
accidentales, etc. La música que está entre \cadenzaOn y \cadenzaOff no cuenta para la duración de un compás.

```
\relative c'' {
  % Show all bar numbers
  \override Score.BarNumber.break-visibility = #all-visible
  c4 d e d
  \cadenzaOn
  c4 cis d8[ d d] f4 g4.
  \cadenzaOff
  d4 e d c
}
```



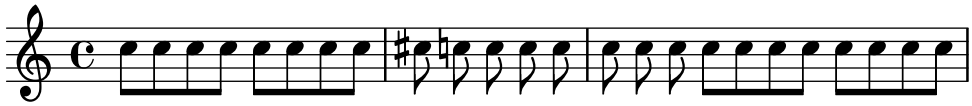
Para dividir un pasaje no medido en varios compases de duración irregular, reactivamos la cuenta de tiempo de forma provisional y usamos \partial para crear un compás diminuto. La instrucción \bar por sí sola no da inicio a un compás nuevo.

```
cadenzaMeasure = {
  \cadenzaOff
  \partial 1024 s1024
  \cadenzaOn
}

\relative c'' {
  % Show all bar numbers
  \override Score.BarNumber.break-visibility = #all-visible
  c4 d e d
  \cadenzaOn
  c4 cis \bar "!" d8[ d d] \cadenzaMeasure f4 g4.
  \cadenzaMeasure
  \cadenzaOff
  d4 e d c
}
```



El barrado automático se desactiva mediante \cadenzaOn. Por tanto, todo el barrado en las cadencias se debe introducir manualmente. Véase Sección 2.4.3 [Barras manuales], página 114.

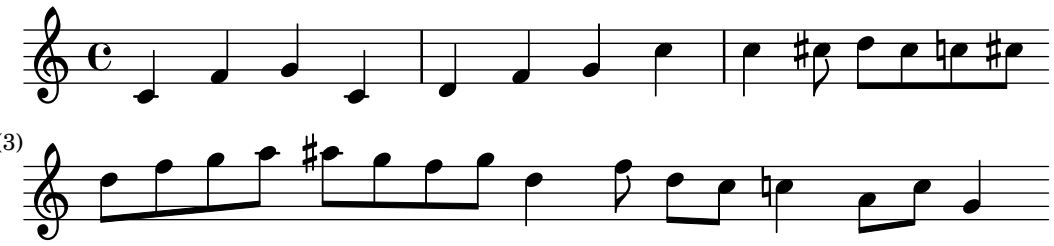
```
\relative {
  \repeat unfold 8 { c''8 }
  \cadenzaOn
  cis8 c c c c
  \bar"|"
  c8 c c
  \cadenzaOff
  \repeat unfold 8 { c8 }
}
```



Estas instrucciones predefinidas afectan a todas las pautas de la partitura, aunque aparezcan en un solo contexto `Voice`. Para cambiar esto, traslade el `Timing_translator` del contexto `Score` al contexto `Staff`, véase Sección 2.3.5 [Notación polimétrica], página 93.

Dentro de una sección de cadencia se desactivan los saltos de línea automáticos: como no hay métrica, no es posible determinar automáticamente cuál sería el lugar apropiado. Por tanto, en un pasaje de cadencia largo, hay que insertar puntos de salto posibles en los lugares adecuados usando la instrucción `\allowBreak` u otras soluciones como las que se dan en Sección 28.1 [Saltos de línea], página 692.

```
\relative {
  c'4 f g c, d f g c
  \cadenzaOn
  c4 cis8
  \allowBreak
  d[ cis c cis]
  \allowBreak
  d[ f g a]
  \allowBreak
  ais[ g f g]
  \allowBreak
  d4 f8
  \allowBreak
  d[ cis] c4
  \allowBreak
  a8[ c] g4
}
```



Instrucciones predefinidas

`\cadenzaOn`, `\cadenzaOff`.

Véase también

Glosario musical: Sección “cadenza” en *Glosario Musical*.

Referencia de la notación: Sección 36.7 [Visibilidad de los objetos], página 794, Sección 2.3.5 [Notación polimétrica], página 93, Sección 2.4.3 [Barras manuales], página 114, Sección 1.1.3 [Alteraciones accidentales], página 8.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

2.3.5 Notación polimétrica

Está contemplada la notación polimétrica, ya sea explícitamente o mediante la modificación manual de la indicación visible del compás y/o el escalado de la duración de las notas.

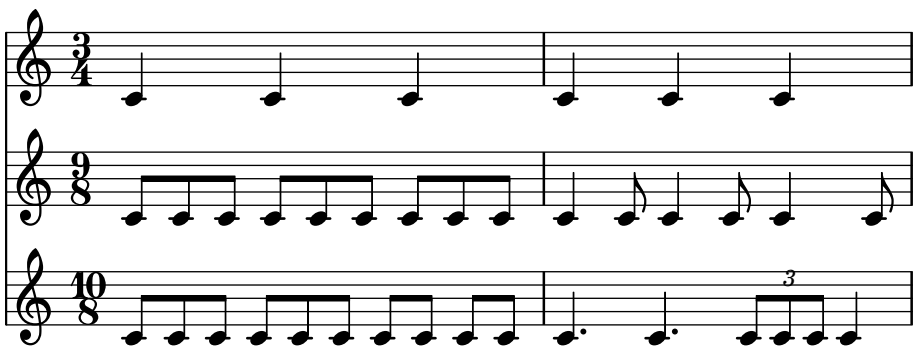
Pentagramas con distintas indicaciones de compás y compases de igual longitud

Use `\polymetric \time` para fijar una indicación de compás local y `\scaleDurations` para hacer que su longitud de compás coincida con la de `Timing.measureLength`. Esto fija el mismo conjunto de propiedades que la instrucción `\time`, excepto en lo que se refiere a `measureLength`.

Para revertir los valores de las propiedades locales y continuar con el uso de los valores del contexto `Timing`, use `\polymetric \default`.

En el siguiente ejemplo, se usan pentagramas con compases de 3/4, 9/8 y 10/8, con 3/4 seleccionada como la longitud del compás de referencia. La música del primer pentagrama presenta un espaciado normal. En el segundo pentagrama, las duraciones mostradas se multiplican por 2/3 (pues 2/3 * 9/8 = 3/4), y en el tercer pentagrama, las duraciones que se muestran están multiplicadas por 3/5 (pues 3/5 * 10/8 = 3/4).

```
\relative <<
  \new Staff {
    \time 3/4
    c'4 c c |
    c4 c c |
  }
  \new Staff {
    \scaleDurations 2/3 {
      \context Staff \polymetric \time 9/8
      \repeat unfold 3 { c8 c c }
      \repeat unfold 3 { c4 c8 }
    }
  }
  \new Staff {
    \scaleDurations 3/5 {
      \context Staff \polymetric \time 3,3,2,2 10/8
      \repeat unfold 2 { c8 c c }
      \repeat unfold 2 { c8 c } |
      c4. c \tuplet 3/2 { c8 c c } c4
    }
  }
>>
```

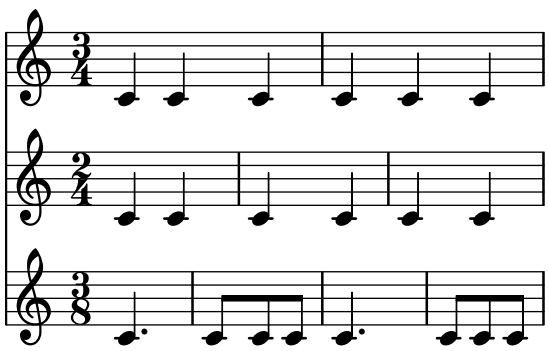


Pentagramas con distintas indicaciones de compás y longitudes de compás distintas

Se puede dar a cada pentagrama su propia indicación de compás independiente en el momento en que se escriba `\enablePerStaffTiming` dentro del bloque `\layout`.

```
\layout {
  \enablePerStaffTiming
```

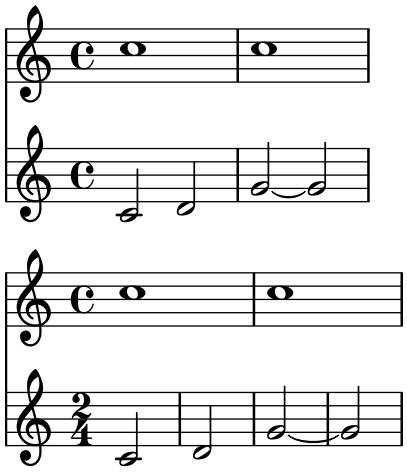
```
}

% Now each staff has its own time signature.

\relative <<
  \new Staff {
    \time 3/4
    c'4 c c |
    c4 c c |
  }
  \new Staff {
    \time 2/4
    c4 c |
    c4 c |
    c4 c |
  }
  \new Staff {
    \time 3/8
    c4. |
    c8 c c |
    c4. |
    c8 c c |
  }
>>
```



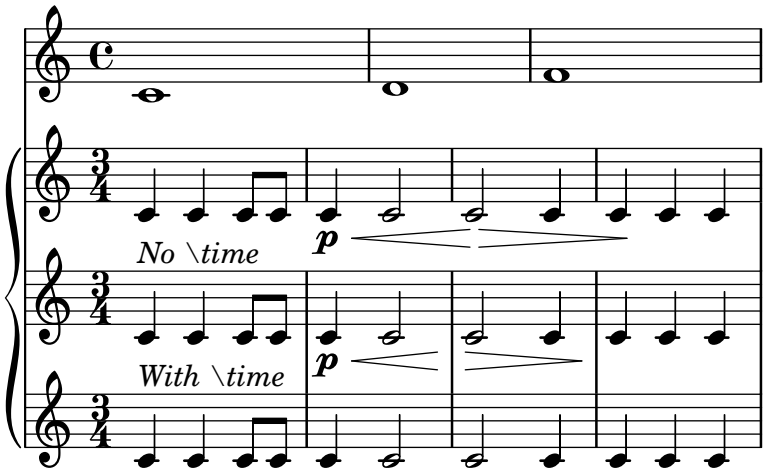
Para tener solo una partitura polimétrica, escriba \enablePerStaffTiming en un bloque \layout dentro del bloque \score.

```
\score {
  <<
    \new Staff { c''1 1 }
    \new Staff { c'2 d' g'2~ 2 }
  >>
}

\score {
  \layout {
    \enablePerStaffTiming
  }
  <<
    \new Staff { \time 4/4 c''1 1 }
    \new Staff { \time 2/4 c'2 d' g'2~ 2 }
```

```
  >>
}
```



Al usar polimetros, todos los pentagramas deben incluir una instrucción `\time` si el compás no es el predeterminado de 4/4. Esto es así incluso para pautas especiales que no tienen líneas, como los contextos `Dynamics` para indicaciones de matiz dinámico, porque la colocación de ciertos elementos extensos como los reguladores se sincroniza con las barras de compás.



Para poder usar esta funcionalidad con la salida MIDI, escriba también `\enablePerStaffTiming` dentro del bloque `\midi`.

```
\layout {
  \enablePerStaffTiming
}

\midi {
  \enablePerStaffTiming
}
```

Véase también

Glosario musical: Sección “polymetric” en *Glosario Musical*, Sección “polymetric time signature” en *Glosario Musical*, Sección “meter” en *Glosario Musical*.

Referencia de la notación: Sección 2.4.1 [Barras automáticas], página 101, Sección 2.4.3 [Barras manuales], página 114, Sección 2.3.1 [Indicación de compás], página 79, Sección 2.1.3 [Escalar las duraciones], página 62.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” en *Referencia de Funcionamiento Interno*, Sección “Timing_translator” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*.

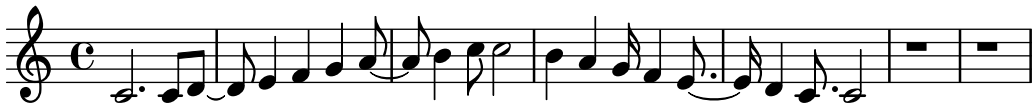
Advertencias y problemas conocidos

Aunque las notas que se producen en el mismo momento en distintos pentagramas se colocan en la misma posición horizontal, las barras de compás en cada pauta pueden hacer que el espaciado de notas sea menos regular según cada tipo de compás.

2.3.6 División automática de las notas

Las notas largas se pueden convertir automáticamente en notas ligadas. Se hace mediante la sustitución del `Note_heads_engraver` por el `Completion_heads_engraver`. De forma similar, los silencios largos que sobrepasan líneas de compás se dividen automáticamente sustituyendo el grabador `Rest_engraver` con el grabador `Completion_rest_engraver`. En el ejemplo siguiente, las notas y los silencios que atraviesan la barra de compás se dividen, y además las notas se unen mediante una ligadura.

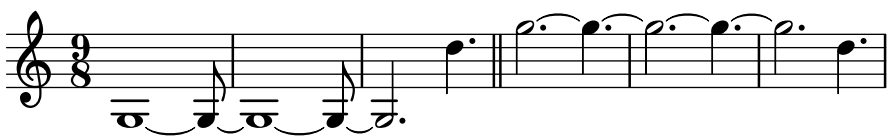
```
\new Voice \with {
  \remove Note_heads_engraver
  \consists Completion_heads_engraver
  \remove Rest_engraver
  \consists Completion_rest_engraver
}
\relative {
  c'2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 r1*2
}
```



Estos grabadores dividen todas las notas y silencios largos en la barra de compás, e inserta ligaduras en las notas. Uno de sus usos es depurar partituras complejas: si los compases no están completos, las ligaduras mostrarán exactamente cuánto le falta a cada compás.

La propiedad `completionUnit` fija una duración preferida para las notas divididas.

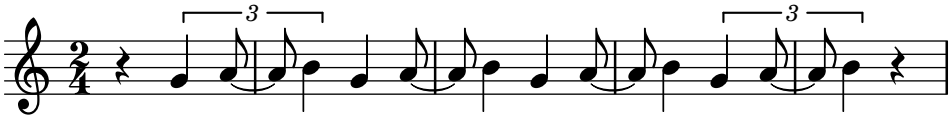
```
\new Voice \with {
  \remove Note_heads_engraver
  \consists Completion_heads_engraver
} \relative {
  \time 9/8 g\breve. d''4. \bar "||"
  \set completionUnit = #3/8
  g\breve. d4.
}
```



Estos grabadores dividen las notas que tienen la duración escalada, como las de los tresillos, en notas que tienen el mismo factor de escala que la nota original de la entrada.

```
\new Voice \with {
```

```
  \remove Note_heads_engraver
  \consists Completion_heads_engraver
} \relative {
  \time 2/4 r4
  \tuplet 3/2 {g'4 a b}
  \scaleDurations 2/3 {g a b}
  g4*2/3 a b
  \tuplet 3/2 {g4 a b}
  r4
}
```



Véase también

Glosario musical: Sección “tie” en *Glosario Musical*

Manual de aprendizaje: Sección “Explicación de los grabadores” en *Manual de Aprendizaje*, Sección “Añadir y eliminar grabadores” en *Manual de Aprendizaje*.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_heads_engraver” en *Referencia de Funcionamiento Interno*, Sección “Completion_heads_engraver” en *Referencia de Funcionamiento Interno*, Sección “Rest_engraver” en *Referencia de Funcionamiento Interno*, Sección “Completion_rest_engraver” en *Referencia de Funcionamiento Interno*, Sección “Forbid_line_break_engraver” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

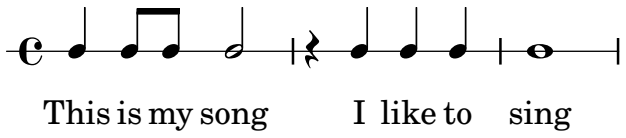
Por consistencia con el comportamiento anterior, las notas y silencios que tienen una duración mayor de un compás, como `c1*2`, se dividen en notas sin ningún factor de escala, `{ c1 c1 }`. La propiedad `completionFactor` controla este comportamiento, y al darle el valor `#f` podemos hacer que las notas y silencios divididos tengan el mismo factor de escala que las duraciones de las notas originales de la entrada.

2.3.7 Mostrar los ritmos de la melodía

A veces podemos querer mostrar solamente el ritmo de una melodía. Esto se puede hacer con un pentagrama de ritmo. Todas las alturas de las notas se convierten en barras inclinadas, y el pentagrama tiene una sola línea:

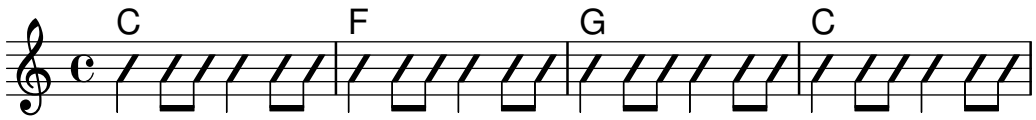
```
<<
  \new RhythmicStaff {
    \new Voice = "myRhythm" \relative {
      \time 4/4
      c'4 e8 f g2
      r4 g g f
      g1
    }
  }
  \new Lyrics {
    \lyricsto "myRhythm" {
      This is my song
      I like to sing
    }
```

```
  }
>>
```



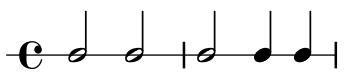
Las tablas de acordes de guitarra ofrecen a menudo los ritmos de rasgueado. Esto se puede hacer con el grabador `Pitch_squash_engraver` y `\improvisationOn`.

```
<<
  \new ChordNames {
    \chordmode {
      c1 f g c
    }
  }
  \new Voice \with {
    \consists Pitch_squash_engraver
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
>>
```



También se puede usar música que contenga acordes como entrada para `RhythmicStaff` y para usarla con el grabador `Pitch_squash_engraver` si los acordes se reducen primero a notas individuales con la función musical `\reduceChords`:

```
\new RhythmicStaff {
  \time 4/4
  \reduceChords {
    <c>2
    <e>2
    <c e g>2
    <c e g>4
    <c e g>4
  }
}
```



Instrucciones predefinidas

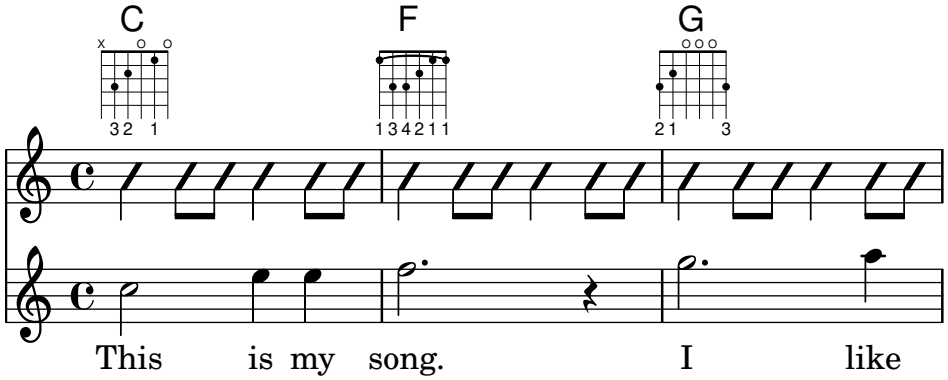
`\improvisationOn`, `\improvisationOff`, `\reduceChords`.

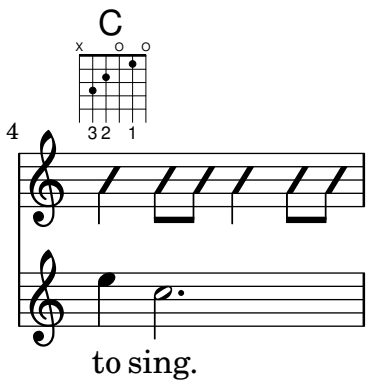
Fragmentos de código seleccionados

Ritmos rasgueados de guitarra

Para la música de guitarra es posible mostrar los ritmos de rasgueo, además de las notas de la melodía, acordes y diagramas de posiciones.

```
\include "predefined-guitar-fretboards.ly"

<<
  \new ChordNames \chordmode {
    c1 | f | g | c
  }
  \new FretBoards \chordmode {
    c1 | f | g | c
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
  \new Voice = "melody" \relative c'' {
    c2 e4 e4
    f2. r4
    g2. a4
    e4 c2.
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
    I like to sing.
  }
>>
```





Véase también

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RhythmicStaff” en *Referencia de Funcionamiento Interno*, Sección “Pitch_squash_engraver” en *Referencia de Funcionamiento Interno*.

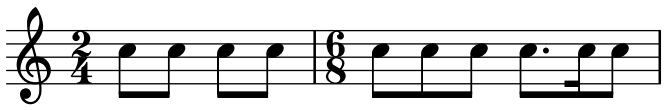
2.4 Barras

LilyPond ofrece dos posibilidades distintas para la introducción de las barras de corchea (y valores menores): la escritura automática y la manual, que pueden también estar mezcladas.

2.4.1 Barras automáticas

De manera predeterminada, las barras de corchea se insertan automáticamente:

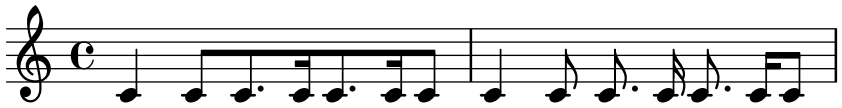
```
\relative c'' {
  \time 2/4 c8 c c c
  \time 6/8 c8 c c c8. c16 c8
}
```



Cuando estas decisiones automáticas no son lo bastante buenas, se pueden escribir los barrados de forma explícita; véase Sección 2.4.3 [Barras manuales], página 114. Las barras *se deben* introducir manualmente si se quieren extender por encima de los silencios.

Si no se necesita el barrado automático, se puede desactivar con `\autoBeamOff` y activarse con `\autoBeamOn`:

```
\relative c' {
  c4 c8 c8. c16 c8. c16 c8
  \autoBeamOff
  c4 c8 c8. c16 c8.
  \autoBeamOn
  c16 c8
}
```



Nota: Si se usan barras para indicar los melismas de las canciones, entonces se debe desactivar el barrado automático con `\autoBeamOff` e indicar las barras manualmente. La utilización de `\partCombine` con `\autoBeamOff` puede producir resultados no deseados. Véanse los fragmentos de código para mayor información.

Se pueden crear patrones de barrado que difieran de los valores automáticos predeterminados; véase Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104.

Instrucciones predefinidas

`\autoBeamOff`, `\autoBeamOn`.

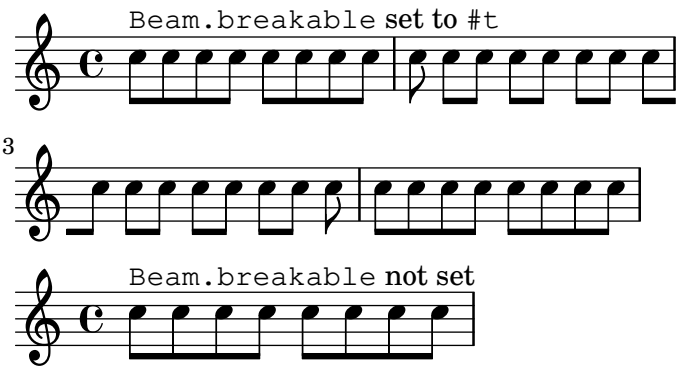
Fragmentos de código seleccionados

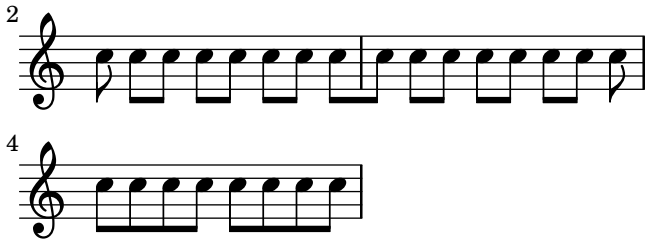
Barras que atraviesan saltos de línea

Normalmente LilyPond rehúsa insertar un salto de línea automático si las barra atraviesa la línea divisoria. Se puede cambiar este comportamiento por medio del establecimiento de la propiedad `Beam.breakable` al valor `#t`.

Esta propiedad no afecta a los saltos manuales insertados con instrucciones como `\break`.

```
music = {
  \repeat unfold 8 c8
  c8 \repeat unfold 7 { c[ c] }  c
  \repeat unfold 8 c8
}

\relative c'' {
  <>^\markup { \typewriter Beam.breakable set to \typewriter "#t" }
  \override Beam.breakable = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Beam.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}
```

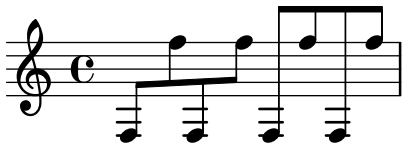




Cambiar el salto de las barras en ángulo

Se insertan automáticamente barras en ángulo cuando se detecta un intervalo muy grande entre las notas. Se puede hacer un ajuste fino de este comportamiento a través de la propiedad `auto-knee-gap`. Se traza una barra doblada si el salto es mayor que el valor de `auto-knee-gap` más el ancho del objeto barra (que depende de la duración de las notas y de la inclinación de la barra). De forma predeterminada `auto-knee-gap` está establecido a 5.5 espacios de pentagrama.

```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



`Partcombine` *y* `\autoBeamOff`

Puede ser difícil comprender la función de `\autoBeamOff` cuando se usa con `\partCombine`. Puede ser preferible usar

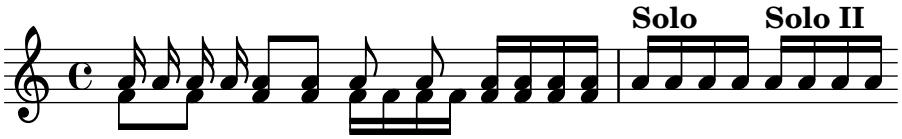
```
\set Staff.autoBeaming = ##f
```

en su lugar, para asegurarnos de que el barrado automático se desactiva para todo el pentagrama. Úselo en un lugar de la partitura en el que no esté aún activo ninguna barra generada por el algoritmo de barrado automático.

Internamente, `\partCombine` funciona con cuatro voces: plica arriba cuando hay una sola voz, plica abajo con la otra, combinadas, y una sola voz. Para poder usar `\autoBeamOff` con el objeto de detener todo el barrado automático cuando se usa con `\partCombine`, es necesario hacer *cuatro* llamadas a `\autoBeamOff`.

```
{
  % \set Staff.autoBeaming = ##f % turns off all auto-beaming

  \partCombine {
    \autoBeamOff   % applies to split up-stems
    \repeat unfold 4 a'16
    % \autoBeamOff % applies to combined stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
    % \autoBeamOff % applies to solo
    \repeat unfold 4 a'16
    r4
  } {
    % \autoBeamOff % applies to split down-stems
    \repeat unfold 4 f'8
```

```
    \repeat unfold 8 f'16 |
    r4
    \repeat unfold 4 a'16
  }
}
```



Véase también

Referencia de la notación: Sección 2.4.3 [Barras manuales], página 114, Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104.

Archivos instalados: `scm/auto-beam.scm`.

Fragmentos de código: Sección "Duraciones" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Auto_beam_engraver" en *Referencia de Funcionamiento Interno*, Sección "Beam_engraver" en *Referencia de Funcionamiento Interno*, Sección "Beam" en *Referencia de Funcionamiento Interno*, Sección "BeamEvent" en *Referencia de Funcionamiento Interno*, Sección "BeamForbidEvent" en *Referencia de Funcionamiento Interno*, Sección "beam-interface" en *Referencia de Funcionamiento Interno*, Sección "unbreakable-spanner-interface" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las propiedades de una barra vienen determinadas al *comienzo* de su construcción y cualquier cambio adicional en las propiedades de la barra que se produzca antes de que la barra se ha completado no tendrá efecto hasta que inicie la *siguiente* barra nueva.

2.4.2 Establecer el comportamiento de las barras automáticas

Cuando está habilitado el barrado automático, la colocación de las barras automáticas viene determinada por tres propiedades de contexto: `beatBase`, `beatStructure` y `beamExceptions`. Los valores predeterminados de estas variables se pueden sobreescribir como se describe más abajo, o de forma alternativa los propios valores predeterminados se pueden cambiar como se explica en Sección 2.3.1 [Indicación de compás], página 79.

Si hay definida una regla de `beamExceptions` para el compás en curso, se usa dicha regla para determinar la colocación de las barras; se ignoran los valores de `beatBase` y `beatStructure`.

Si no está definida ninguna regla de `beamExceptions` para el tipo de compás en curso, la colocación de las barras está determinada por los valores de `beatBase` y `beatStructure`.

Barrado basado en `beatBase` y `beatStructure`

De forma predeterminada, las reglas de `beamExceptions` están definidas para los compases más comunes, y así las reglas de `beamExceptions` se deben desactivar si pretendemos que el barrado automático esté basado en `beatBase` y `beatStructure`. Las reglas de `beamExceptions` se desactivan mediante

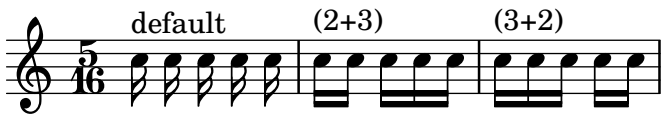
```
\set Timing.beamExceptions = #'()
```

Cuando el valor de `beamExceptions` se ha fijado a `#'()`, ya sea debido a un ajuste explícito o a causa de que no hay ninguna regla de `beamExceptions` definida internamente para el compás actual, los puntos finales de las barras están en los pulsos según viene determinado por las propiedades de contexto `beatBase` y `beatStructure`. `beatStructure` es una lista de Scheme que define la longitud de cada pulso dentro del compás en unidades de `beatBase`. De

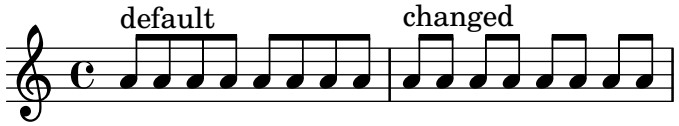
forma predeterminada, `beatBase` es una unidad más que el denominador del compás. De forma predeterminada también, cada unidad de longitud `beatBase` es un único pulso.

Observe que existen valores de `beatStructure` y de `beatBase` diferentes para cada indicación de compás. Los cambios que se hacen a estas variables se aplican solamente al tipo de compás en vigor, por lo que dichos cambios se deben escribir después de la instrucción `\time` que da comienzo a una sección nueva con un tipo de compás distinto, no antes. Los valores nuevos que se dan a una indicación de compás concreta se retienen y se vuelven a aplicar cuando ese tipo de compás vuelve a establecerse.

```
\relative c'' {
  \time 5/16
  c16^"default" c c c c |
  % beamExceptions are unlikely to be defined for 5/16 time,
  % but let's disable them anyway to be sure
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 2,3
  c16^"(2+3)" c c c c |
  \set Timing.beatStructure = 3,2
  c16^"(3+2)" c c c c |
}
```



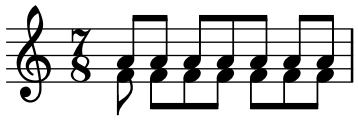

```
\relative {
  \time 4/4
  a'8^"default" a a a a a a a
  % Disable beamExceptions because they are definitely
  % defined for 4/4 time
  \set Timing.beamExceptions = #'()
  \set Timing.beatBase = #1/4
  \set Timing.beatStructure = 1,1,1,1
  a8^"changed" a a a a a a a
}
```



Los cambios en los ajustes de barrado se pueden limitar a contextos específicos. Si no se incluye ningún ajuste en un contexto de nivel más bajo, se aplican los ajustes del contexto circundante.

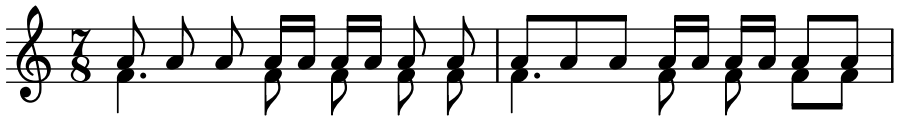
```
\new Staff {
  \time 7/8
  % No need to disable beamExceptions
  % as they are not defined for 7/8 time
  \set Staff.beatStructure = 2,3,2
  <<
    \new Voice = one {
      \relative {
        a'8 a a a a a a
      }
```

```
    }
    \new Voice = two {
      \relative {
        \voiceTwo
        \set Voice.beatStructure = 1,3,3
        f'8 f f f f f f
      }
    }
  >>
}
```



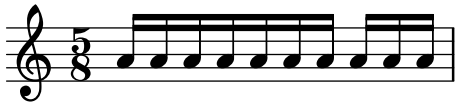
En caso de usar varias voces, se debe especificar el contexto `Staff` si queremos aplicar el barrado a todas las voces del pentagrama:

```
\time 7/8
% rhythm 3-1-1-2
% Change applied to Voice by default --  does not work correctly
% Because of auto-generated voices, all beating will
% be at beatBase #1/8
\set beatStructure = 3,1,1,2
<< \relative {a'8 a a a16 a a a a8 a} \\ \relative {f'4. f8 f f f} >>

% Works correctly with context Staff specified
\set Staff.beatStructure = 3,1,1,2
<< \relative {a'8 a a a16 a a a a8 a} \\ \relative {f'4. f8 f f f} >>
```



El valor de `beatBase` se puede ajustar para cambiar el comportamiento de las barras, si se desea. Cuando se hace, el valor de `beatStructure` se debe fijar de manera que sea compatible con el nuevo valor de `beatBase`.

```
\time 5/8
% No need to disable beamExceptions
% as they are not defined for 5/8 time
\set Timing.beatBase = #1/16
\set Timing.beatStructure = 7,3
\repeat unfold 10 { a'16 }
```



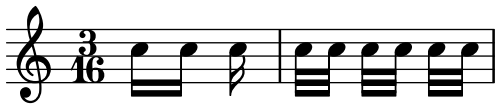
De forma predeterminada `beatBase` está fijado a una unidad más que el denominador del compás. Todas las excepciones a este valor predeterminado están en `scm/time-signature-settings.scm`.

Barrado basado en `beamExceptions`

Las reglas de autobarrado especiales (distintas de terminar una barra sobre un pulso) están definidas en la propiedad `beamExceptions`.

El valor de `beamExceptions`, que es una estructura de datos de Scheme bastante compleja, se genera fácilmente con la función `\beamExceptions`. Esta función recibe uno o más patrones rítmicos barrados manualmente y con un compás de duración (los compases se deben separar por una barra de comprobación de compás `|` porque la función no tiene otra manera de discernir la longitud del compás). He aquí un ejemplo sencillo:

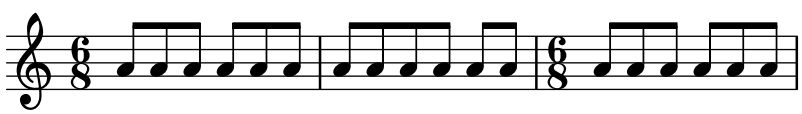
```
\relative c'' {
  \time 3/16
  \set Timing.beatStructure = 2,1
  \set Timing.beamExceptions =
    \beamExceptions { 32[ 32] 32[ 32] 32[ 32] }
  c16 c c |
  \repeat unfold 6 { c32 } |
}
```



> **Nota:** Un valor de `beamExceptions` debe ser una lista de excepciones *completa*. Esto es, toda excepción que se tenga que aplicar debe estar incluida en este ajuste. No es posible añadir, eliminar o cambiar solo una de las excepciones. Aunque esto puede parecer engorroso, significa que no es necesario conocer los ajustes de barrado actuales para poder especificar un patrón de barrado nuevo.

Cuando cambia el compás, se fijan los valores predeterminados de `Timing.beatBase`, `Timing.beatStructure` y `Timing.beamExceptions`. Un ajuste en el tipo de compás da como resultado un reinicio de los ajustes de barrado automático para el contexto `Timing` al comportamiento predeterminado.

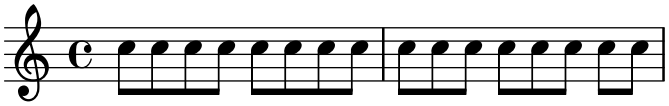
```
\relative a' {
  \time 6/8
  \repeat unfold 6 { a8 }
  % group (4 + 2)
  \set Timing.beatStructure = 4,2
  \repeat unfold 6 { a8 }
  % go back to default behavior
  \time 6/8
  \repeat unfold 6 { a8 }
}
```



Los ajustes de barrado automático predeterminados para un tipo de compás están determinados en el archivo `scm/time-signature-settings.scm`. La forma de cambiar los ajustes predeterminados de barrado automático para un tipo de compás se describe en Sección 2.3.1 [Indicación de compás], página 79.

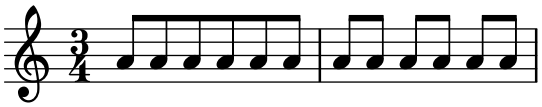
Muchos ajustes de barrado automáticos para un tipo de compás contienen una entrada para beamExceptions. Por ejemplo, el compás de 4/4 trata de unir el compás en dos partes si solo hay corcheas. La regla beamExceptions puede sobreescribir el ajuste beatStructure si no se reinicia beamExceptions.

```
\time 4/4
\set Timing.beatBase = #1/8
\set Timing.beatStructure = 3,3,2
% This won't beam (3 3 2) because of beamExceptions
\repeat unfold 8 {c''8} |
% This will beam (3 3 2) because we clear beamExceptions
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c''8}
```



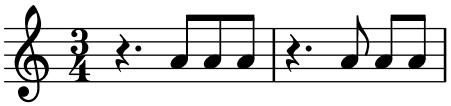
De forma similar, las corcheas en compás de 3/4 se unen mediante una sola barra para todo el compás, de forma predeterminada. Para unir las corcheas en 3/4 mediante una barra en cada parte, reinicie beamExceptions.

```
\time 3/4
% by default we beam in (6) due to beamExceptions
\repeat unfold 6 {a'8} |
% This will beam (1 1 1) due to default beatBase and beatStructure
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a'8}
```



En la música tipografiada de los períodos clásico y romántico, con frecuencia las barras comienzan a mitad de un compás en 3/4, pero la práctica moderna es evitar la falsa impresión de 6/8 (véase Gould, pág. 153). Se producen situaciones similares en el compás de 3/8. Este comportamiento viene controlado mediante la propiedad de contexto beamHalfMeasure, que tiene efecto solamente sobre indicaciones de compás que tienen la cifra 3 en el numerador:

```
\relative a' {
  \time 3/4
  r4. a8 a a |
  \set Timing.beamHalfMeasure = ##f
  r4. a8 a a |
}
```



Cómo funciona el barrado automático

Cuando se habilita el barrado automático, la colocación de las barras de corchea automáticas viene determinada por las propiedades de contexto beatBase, beatStructure y beamExceptions.

Son de aplicación las siguientes reglas, en orden de prioridad, cuando se determina el aspecto de las barras:

- Si está especificada una barra manual con `[...]`, fijar la barra tal y como se ha especificado; en caso contrario,
- si está definida en `beamExceptions` una regla de barrado para este tipo de barra, utilizarla para determinar los lugares válidos en que pueden terminar las barras; en caso contrario,
- si está definida en `beamExceptions` una regla de final de barra para un tipo de barra más largo, utilizarla para determinar los lugares válidos en que pueden terminar las barras; en caso contrario,
- usar los valores de `beatBase` y de `beatStructure` para determinar los finales de los pulsos dentro del compás, y terminar las barras al final de los pulsos.

En las reglas anteriores, el tipo de barra *beam type* es la duración de la nota más breve dentro del grupo unido por una barra.

Las reglas de barrado predefinidas están en el archivo `scm/time-signature-settings.scm`.

Fragmentos de código seleccionados

Subdivisión de barras

Las barras de semicorchea, o de figuras más breves, no se subdividen de forma predeterminada. Esto es, las barras de más de dos plicas se amplían sobre el grupo completo de notas, sin dividirse. Este comportamiento se puede modificar para subdividir las barras en subgrupos mediante el establecimiento de la propiedad `subdivideBeams` al valor `#t`. Cuando está activada, las barras se subdividen a intervalos definidos por el valor actual de la subivisión.

Usando las propiedades `beamMinimumSubdivision` y `beamMaximumSubdivision` se pueden configurar los límites de la subdivisión automática de las barras, concretamente las duraciones mínima y máxima en que las barras secundarios se eliminan. Los valores por omisión son `0` para la primera y `+inf.0` para la segunda, haciendo que LilyPond subdivida las barras tanto como sea posible.

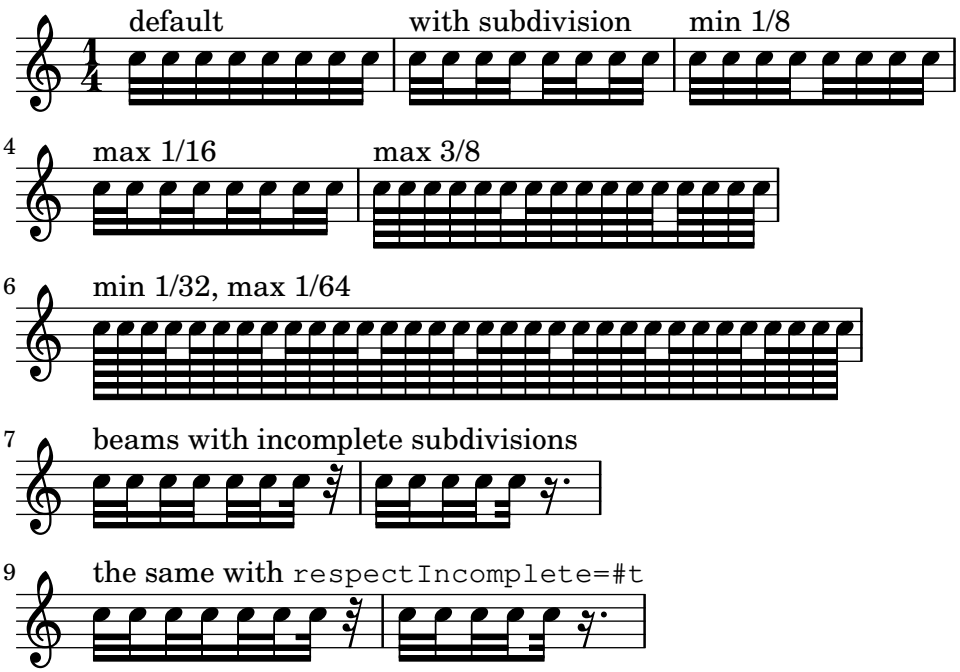
Existen dos casos especiales que tener en cuenta.

- Si el numerador de `beamMaximumSubdivision` no es una potencia de 2, las duraciones candidatas a la subdivisión son `beamMaximumSubdivision` dividido por las potencias de 2 que sean mayores o iguales que `beamMinimumSubdivision`.
- Si `beamMaximumSubdivision` es menor que `beamMinimumSubdivision`, la profundidad de las subdivisiones de las barras está limitada por `beamMaximumSubdivision`, pero no la frecuencia y los intervalos rítmicos, desviándose por tanto, posiblemente, de las longitudes correctas o esperadas.

Si `respectIncompleteBeams` se fija al valor `#t`, las subdivisiones incompletas con más de dos plicas se tratan como una 'extensión' del grupo de subdivisión previo, es decir, la duración del grupo de subdivisión anterior se extiende para que comprenda también la subdivisión incompleta. Si en vez de ello se fija a `#f` (que es lo predeterminado), se inicia un grupo de subdivisión nuevo.

```
\relative c'' {
  \time 1/4

  <>^"default"
  c32 c c c c c c c

  <>^"with subdivision"
  \set subdivideBeams = ##t
  c32 c c c c c c c

  <>^"min 1/8"
```

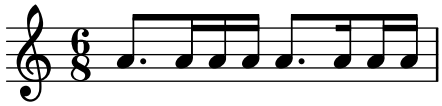
```
  \once \set beamMinimumSubdivision = #1/8
  c32 c c c c c c c

  <>^"max 1/16"
  \once \set beamMaximumSubdivision = #1/16
  c32 c c c c c c c

  <>^"max 3/8"
  \once \set beamMaximumSubdivision = #3/8
  \repeat unfold 16 c64

  <>^"min 1/32, max 1/64"
  % Set maximum beam subdivision interval to 1/64 to limit
  % subdivision depth, despite not being metrically correct.
  \once \set beamMinimumSubdivision = #1/32
  \once \set beamMaximumSubdivision = #1/64
  \repeat unfold 32 c128
  \break

  <>^"beams with incomplete subdivisions"
  c32 c c c c c c r32
  c32 c c c c r16.

  <>^\markup { "the same with"
               \typewriter { "respectIncomplete=#t" } }
  \set respectIncompleteBeams = ##t
  % The incomplete subgroup extends the completed subgroup.
  c32 c c c c c c r32
  % No visual change since we have only two stems in the
  % incomplete subgroup.
  c32 c c c c r16.
}
```



Barras que se atienen al pulso estrictamente

Se puede hacer que las barras secundarias apunten en la dirección del pulso o fracción a que pertenecen. La primera barra evita los corchetes sueltos (que es el comportamiento predeterminado); la segunda barra sigue el pulso o fracción estrictamente.

```
\relative c'' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



Símbolos de dirección y símbolos de agrupación de compás

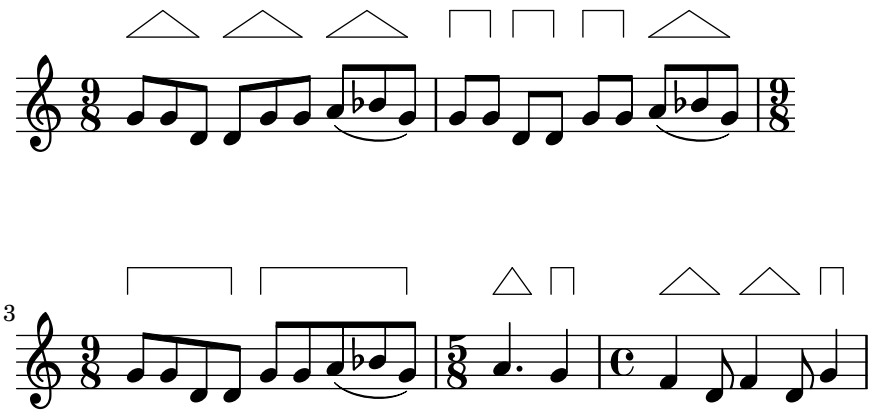
Las propiedades de contexto controlan el agrupamiento de los pulsos dentro de un compás: `beatStructure` lista la longitud de cada pulso en unidades de `beatBase`. Hay establecidos valores de `beatStructure` para muchos tipos de compases en `scm/time-signature-settings.scm`. Estas propiedades se pueden cambiar o establecer con `\set`.

Como alternativa, `\time` acepta opcionalmente una estructura de pulsos para usarla de forma predeterminada. `\time` se aplica al contexto `Timing`, por lo que no restablece los valores de `beatStructure` ni de `beatBase` que se establezcan en otros contextos de nivel inferior, como `Voice`.

Si el grabador `Measure_grouping_engraver` está incluido en uno de los contextos de presentación, se imprimirán signos de agrupación de pulsos. Estos símbolos facilitan la lectura de música moderna rítmicamente compleja. En este ejemplo, el compás de 9/8 se agrupa según dos patrones distintos utilizando los dos métodos, mientras que el compás de 5/8 se agrupa de acuerdo con el ajuste predeterminado que está en `scm/time-signature-settings.scm`. Para el compás de 4/4 hay que fijar explícitamente `beatBase` a corcheas para que se muestre el patrón irregular del compás.

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
    \time 3,3,2 4/4
    \set Timing.beatBase = #1/8
    f4 d8 f4 d8 g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
```

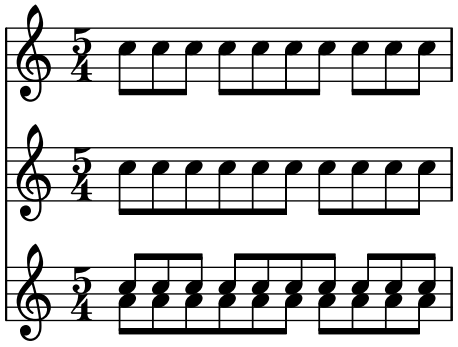
```
}
```



Finales de barra en el contexto Score

Las reglas de final de barra especificadas en el contexto `Score` se aplican a todos los pentagramas, pero se pueden modificar tanto en los niveles de `Staff` como de `Voice`:

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.beatBase = #1/8
  \set Score.beatStructure = 3,4,3
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = 6,4
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = 6,4
          a8 a a a a a a a a a
        }
      >>
    }
  >>
}
```



Véase también

Referencia de la notación: Sección 2.3.1 [Indicación de compás], página 79.

Archivos de inicio: `scm/time-signature-settings.scm`.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

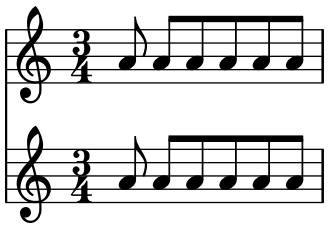
Referencia de funcionamiento interno: Sección “Auto_beam_engraver” en *Referencia de Funcionamiento Interno*, Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “BeamForbidEvent” en *Referencia de Funcionamiento Interno*, Sección “beam-interface” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Si una partitura termina mientras una barra de corchea automática no ha terminado y aún acepta notas, esta última barra no se imprime en absoluto. Lo mismo sirve para las voces polifónicas introducidas con `<< ... \\ ... >>`. Si una voz polifónica termina mientras una barra de corchea automática aún admite notas, no se imprime. El rodeo para estos problemas es aplicar el barrado manual a la última barra de la voz o partitura.

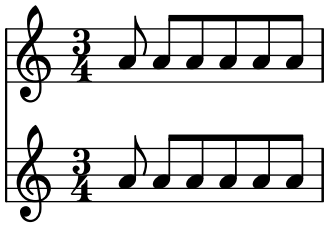
De forma predeterminada, el traductor `Timing` recibe el nombre del contexto `Score` como alias. Esto significa que el establecimiento del compás en ua pauta afectará al barrado de las otras pautas también. Así, un ajuste en el compás en un pentagrama tardío reiniciará el barrado personalizado que se había ajustado en un pentagrama más temprano. Una forma de evitar este problema es ajustar la indicación de compás en un pentagrama solamente.

```
<<
  \new Staff {
    \time 3/4
    \set Timing.beatBase = #1/8
    \set Timing.beatStructure = 1,5
    \set Timing.beamExceptions = #'()
    \repeat unfold 6 { a'8 }
  }
  \new Staff {
    \repeat unfold 6 { a'8 }
  }
>>
```



Los ajustes de barrado predeterminados para dicho compás también se pueden cambiar, de forma que siempre se utilice siempre el barrado deseado. La manera de efectuar cambios en los ajustes de barrado automático para una indicación de compás se describe en Sección 2.3.1 [Indicación de compás], página 79.

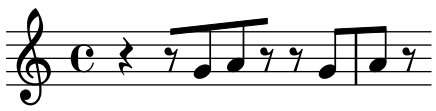
```
<<
  \new Staff {
    \overrideTimeSignatureSettings
      3/4               % timeSignature
      #1/8              % beatBase
      1,5               % beatStructure
      #'()              % beamExceptions
    \time 3/4
    \repeat unfold 6 { a'8 }
  }
  \new Staff {
    \time 3/4
    \repeat unfold 6 { a'8 }
  }
>>
```



2.4.3 Barras manuales

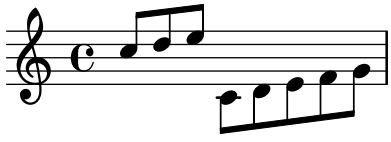
En ciertos casos puede ser preciso sobreescribir el algoritmo de barrado automático. Por ejemplo, el barrador automático no escribe barras por encima de los silencios o las líneas divisorias, y en las partituras corales el barrado se ajusta con frecuencia para que siga la medida de la letra en vez de la de las notas. Tales barras se especifican manualmente marcando los puntos de comienzo y final con `[` y `]`

```
\relative { r4 r8[ g' a r] r g[ | a] r }
```



La dirección de las barras se puede establecer manualmente utilizando indicadores de dirección:

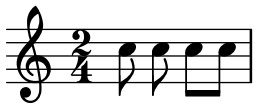
```
\relative { c''8^[ d e] c,_[ d e f g] }
```



Se pueden marcar notas individuales con `\noBeam` para evitar que resulten unidas por una barra:

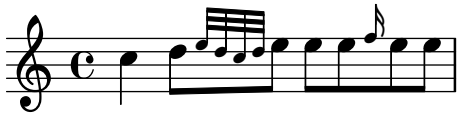
```
\relative {
  \time 2/4
  c''8 c\noBeam c c
```

```
}
```



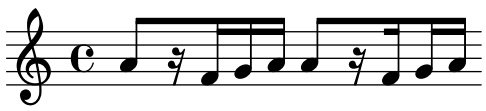
Se pueden producir al mismo tiempo barras de notas de adorno y barras normales. Las notas de adorno sin barra no se colocan dentro de las barras de notas normales.

```
\relative {
  c''4 d8[
  \grace { e32 d c d }
  e8] e[ e
  \grace { f16 }
  e8 e]
}
```



Se puede conseguir un control incluso más estricto sobre las barras estableciendo las propiedades `stemLeftBeamCount` y `stemRightBeamCount`. Especifican el número de barras que se dibujarán en los lados izquierdo y derecho, respectivamente, de la nota siguiente. Si cualquiera de estas dos propiedades está ajustada a un valor, dicho valor se usará una sola vez, y luego se borrará. En este ejemplo, el último Fa se imprime con solo una barra en el lado izquierdo, es decir, la barra de corchea del grupo como un todo.

```
\relative a' {
  a8[ r16 f g a]
  a8[ r16
  \set stemLeftBeamCount = 2
  \set stemRightBeamCount = 1
  f16
  \set stemLeftBeamCount = 1
  g16 a]
}
```



Instrucciones predefinidas

`\noBeam`.

Fragmentos de código seleccionados

Barras recortadas

Las barras recortadas al principio y al final de las barras, junto con las barras unidas a notas sueltas que parecen corchetes planos, son posibles con una combinación de `stemLeftBeamCount`, `stemRightBeamCount` e indicadores de barra `[]` pareados.

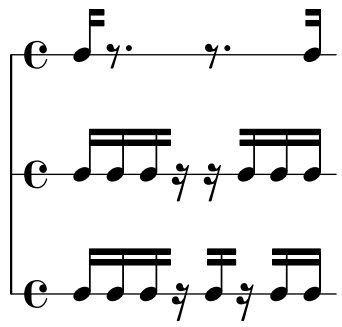
Para imitar los corchetes planos hacia la derecha sobre notas sueltas, indicadores de barra `[]` pareados y fije el valor de `stemLeftBeamCount` a cero. En cambio, para imitar corchetes planos hacia la izquierda sobre notas sueltas, fije el valor de `stemRightBeamCount` a cero (línea uno).

Para las barras recortadas hacia la derecha al final de una serie de noras barradas, fije el valor de `stemRightBeamCount` a un valor positivo. Por su parte, para barras recortadas hacia la derecha al comienzo de una serie de notas barradas, fije `stemLeftBeamCount` (línea dos).

A veces tiene sentido que una nota suelta entre dos silencios lleve dos barras recortadas, hacia la derecha y hacia la izquierda. Esto se hace escribiendo solamente indicadores de barrado `[]` pareados (línea tres).

Observe que `\set stemLeftBeamCount` siempre equivale a `\once \set`. En otras palabras, los ajustes para el número de barras no son "persistentes", o sea que la pareja de barras recortadas unidas a la semicorchea aislada del último ejemplo no tiene nada que ver con la instrucción `\set` de la barra anterior.

```
\score {
  <<
    \new RhythmicStaff {
      \set stemLeftBeamCount = 0
      c16[] r8.
      r8.
      \set stemRightBeamCount = 0
      16[]
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r r
      \set stemLeftBeamCount = 2
      16 16 16
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r16
      16[] r16
      \set stemLeftBeamCount = 2
      16 16
    }
  >>
}
```



Uso de estilos alternativos para los corchetes

Se pueden imprimir formas alternativos del corchete o gancho de las corcheas y figuras menores, mediante la sobreescritura de la propiedad `stencil` del objeto `Flag`. LilyPond porporciona las siguientes funciones: `modern-straight-flag`, `old-straight-flag` y `flat-flag`. Utilice `\revert` para recuperar la forma predeterminada.

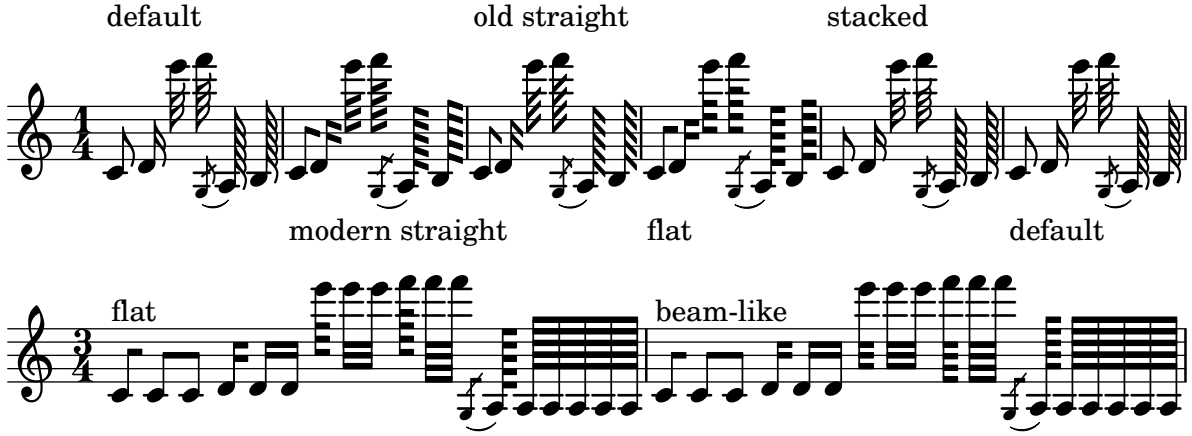
Para obtener corchetes más compactos o apilados verticalmente, llame a la instrucción `\flagStyleStacked`, que se puede anular después con `\flagStyleDefault`.

La sobreescritura de la forma de `Flag` no modifica la manera en que los corchetes se posicionan verticalmente. Esto se nota especialmente en los corchetes rectos: LilyPond no ajusta dinámicamente los huecos verticales entre los corchetes individuales de la misma forma que con las barras de corchea. Una posible solución para armonizar la apariencia es sustituir los corchetes planos por medias barras, como se ve en el segundo pentagrama; sin embargo, esto no se puede hacer automáticamente. En el código de este fragmento, las medias barras se escriben con el prefijo `@`, por ejemplo `@c8`.

Tenga en cuenta que las medias barras *no* son objetos gráficos del tipo `Flag`. Esto significa, particularmente, que la modificación de las propiedades de `Flag` no van a tener ningún efecto sobre ellos (tendríamos que usar en su lugar las propiedades del objeto `Beam`), y las propiedades para su objeto `Stem` asociado se comportarán también como barras.

```
"@" =
#(define-music-function (music) (ly:music?)
  #{ \set stemLeftBeamCount = 0 $music [] #})

testnotes = {
  \autoBeamOff
  c8 d16 e''32 f64 \acciaccatura { g,,,8 } a128 b
}

\relative c' {
  \override TextScript.staff-padding = 6
  \time 1/4
    <>^"default" \testnotes
  \override Flag.stencil = #modern-straight-flag
    <>_"modern straight" \testnotes
  \override Flag.stencil = #old-straight-flag
    <>^"old straight" \testnotes
  \override Flag.stencil = #flat-flag
    <>_"flat" \testnotes
  \revert Flag.stencil

  \flagStyleStacked
    <>^"stacked" \testnotes
  \flagStyleDefault
    <>_"default" \testnotes
}

\relative c' {
  \time 3/4
  \override Flag.stencil = #flat-flag

  <>^"flat" c8 c[ c] d16 d[ d] e''32 e[ e] f64 f[ f]
    \acciaccatura { g,,,8 } a128 a[ a a a]
  <>^"beam-like" @c8 c[ c] @d16 d[ d] @e''32 e[ e] @f64 f[ f]
    \acciaccatura { g,,,8 } @a128 a[ a a a]
}

\layout {
```

```
  indent = 0
  \context {
    \Score
    \override NonMusicalPaperColumn.line-break-permission = ##f
  }
}
```



Véase también

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783, Sección 2.6.1 [Notas de adorno], página 148.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “BeamEvent” en *Referencia de Funcionamiento Interno*, Sección “Beam_engraver” en *Referencia de Funcionamiento Interno*, Sección “beam-interface” en *Referencia de Funcionamiento Interno*, Sección “Stem_engraver” en *Referencia de Funcionamiento Interno*.

2.4.4 Barras progresivas

Las barras progresivas se usan para indicar que un pequeño grupo de notas se debe tocar a una velocidad creciente (o decreciente), sin cambiar el tempo general de la pieza. El ámbito de la barra progresiva se debe indicar manualmente usando `[` y `]`, el efecto de ángulo de la barra se inicia especificando una dirección en la propiedad `grow-direction` del objeto `Beam`.

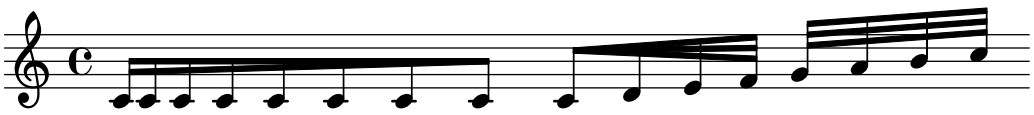
Si queremos que la colocación de las notas y el sonido de la salida MIDI refleje el *ritardando* o *accelerando* indicado por la barra progresiva, las notas se deben agrupar como una expresión musical delimitada por llaves y precedida de una instrucción `\featherDurations` que especifica la razón entre las duraciones de la primera y la última notas dentro del grupo.

Los corchetes rectos muestran el ámbito de la barra y las llaves muestran qué notas han de modificar sus duraciones. Normalmente delimitarían el mismo grupo de notas, pero no es un requisito: las dos instrucciones son independientes.

En el ejemplo siguiente las ocho semicorcheas ocupan exactamente el mismo tiempo que una blanca, pero la primera nota tiene la mitad de duración que la última, con las notas intermedias alargándose gradualmente. Las cuatro primeras fusas se aceleran gradualmente, mientras que las últimas cuatro fusas están a un tempo constante.

```
\relative c' {
  \override Beam.grow-direction = #LEFT
  \featherDurations 2/1
  { c16[ c c c c c c c] }
  \override Beam.grow-direction = #RIGHT
```

```
  \featherDurations 2/3
  { c32[ d e f] }
  % revert to non-feathered beams
  \override Beam.grow-direction = #'()
  { g32[ a b c] }
}
```



El espaciado en la salida impresa representa las duraciones de las notas de una forma solo aproximada, pero la salida MIDI es exacta.

Instrucciones predefinidas

`\featherDurations`.

Véase también

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Advertencias y problemas conocidos

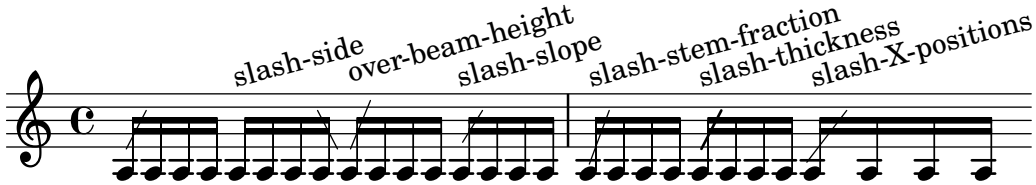
La instrucción `\featherDurations` solo funciona con fragmentos musicales muy breves, y cuando los números de la fracción son pequeños.

2.4.5 Barras cruzadas

Las barras cruzadas se pimprimen usando el procedimiento de sello especial `beam::slashed-stencil`. La barra cruzada se puede imprimir en el lado izquierdo o en el derecho de la barra y se puede personlaizar después por medio de sobreescrituras de las subpropiedades `details` `over-beam-height`, `slash-slope`, `slash-side`, `slash-stem-fraction`, `slash-thickness` y `slash-X-positions`. Observe que esas subpropiedades negocian entre ellas para conseguir un resultado agradable a la vista, esto es, un cambio en una de ellas puede terner un impacto sobre otras.

```
mus = \repeat unfold 4 a16

{
  \override TextScript.rotation = #'(15 1 0)
  \override Beam.stencil = #beam::slashed-stencil
  \mus
  <>^"slash-side" %% default: LEFT
  \once \override Beam.details.slash-side = #RIGHT
  \mus
  <>^"over-beam-height" %% default: 0.75
  \once \override Beam.details.over-beam-height = #1.5
  \mus
  <>^"slash-slope" %% default: 2
  \once \override Beam.details.slash-slope = #1.0
  \mus
  <>^"slash-stem-fraction" %% default: 0.3
  \once \override Beam.details.slash-stem-fraction = #0.6
  \mus
  <>^"slash-thickness" %% default: 0.1
  \once \override Beam.details.slash-thickness = #0.2
```

```
  \mus
  <>^"slash-X-positions" %% default: (-0.5 . 1)
  \once \override Beam.details.slash-X-positions = #'(-1 . 2)
  \mus
}
```



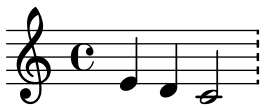
2.5 Compases

2.5.1 Barras de compás

Las líneas divisorias se usan para delimitar a los compases y las secciones, y para indicar repetición. Normalmente, las líneas divisorias normales se insertan de manera automática en la salida impresa en lugares que están de acuerdo con el compás actual. Hay varias instrucciones que insertan otros tipos de barras de compás automáticamente como parte del efecto (véase Sección 2.5.2 [Automatic bar lines], página 130).

Se puede cambiar una barra de compás insertada automáticamente a otro tipo con la instrucción `\bar`:

```
\relative { e'4 d c2 \bar "!" }
```



La última nota de un compás no tiene por qué terminar sobre la línea divisoria automática: se supone que la nota se prolonga sobre el compás siguiente. Pero una secuencia larga de dichos compases prolongados puede hacer que la música aparezca comprimida o incluso que se salga de la página. Esto es a causa de que los saltos de línea automáticos solamente se producen al final de compases completos, es decir, cuando todas las notas han finalizado antes de que el compás termine.

> **Nota:** Una duración incorrecta puede inhibir los saltos de línea, dando como resultado una línea de música con una compresión exagerada o música que se sale de la página.

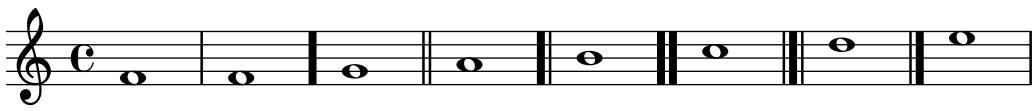
También se permiten saltos de línea en las barras insertadas manualmente incluso dentro de compases incompletos. Para permitir un salto de línea donde no hay ninguna línea divisoria visible, utilice `\allowBreak`; véase Sección 28.1 [Saltos de línea], página 692.

Esta y otras líneas divisorias especiales se pueden insertar manualmente en cualquier punto. Cuando coinciden con el final de un compás, sustituyen a la línea divisoria simple que se habría insertado automáticamente en dicho lugar. Cuando no coinciden con el final de un compás, se inserta la línea especificada en ese punto en la salida impresa.

Las líneas divisorias manuales son puramente visuales. No afectan a ninguna de las propiedades que una barra normal afectaría, como a los números de compás o las alteraciones. No afectan tampoco al cálculo y colocación de las líneas divisorias subsiguientes. Cuando una divisoria se coloca manualmente donde ya existe una divisoria normal, los efectos de la línea original no se alteran.

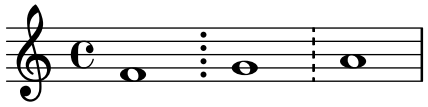
Están disponibles para su inserción manual varios tipos de líneas divisorias simples y dobles:

```
\relative {
  f'1 \bar "|"
  f1 \bar "."
  g1 \bar "||"      % see \section
  a1 \bar ".|"
  b1 \bar ".."
  c1 \bar "|.|"
  d1 \bar "|."      % see \fine
  e1
}
```



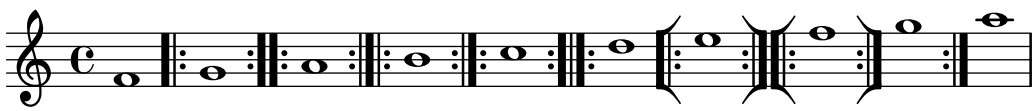
así como la barra de puntos y la discontinua:

```
\relative {
  f'1 \bar ";"
  g1 \bar "!"
  a1
}
```



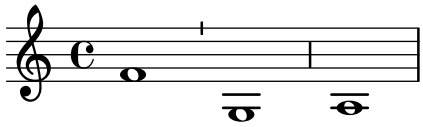
y varios tipos de barra de repetición:

```
\relative {
  f'1 \bar ".|:"
  g1 \bar ":..:"
  a1 \bar ":|.|:"
  b1 \bar ":|.:"
  c1 \bar ":.|.:"
  d1 \bar "[|:"
  e1 \bar ":|][|:"
  f1 \bar ":|]"
  g1 \bar ":|."
  a1
}
```



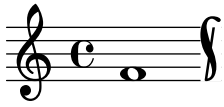
También están disponibles líneas cortas y de tipo coma, aunque en el contexto del canto gregoriano es preferible usar `\divisioMinima` y `\divisioMaior` (véase Sección 17.4.4 [Divisiones], página 554).

```
f'1 \bar "'"
g1 \bar ","
a1
```



LilyPond contempla el uso de notación del canto kievano y provee una línea divisoria especial de este tipo de música:

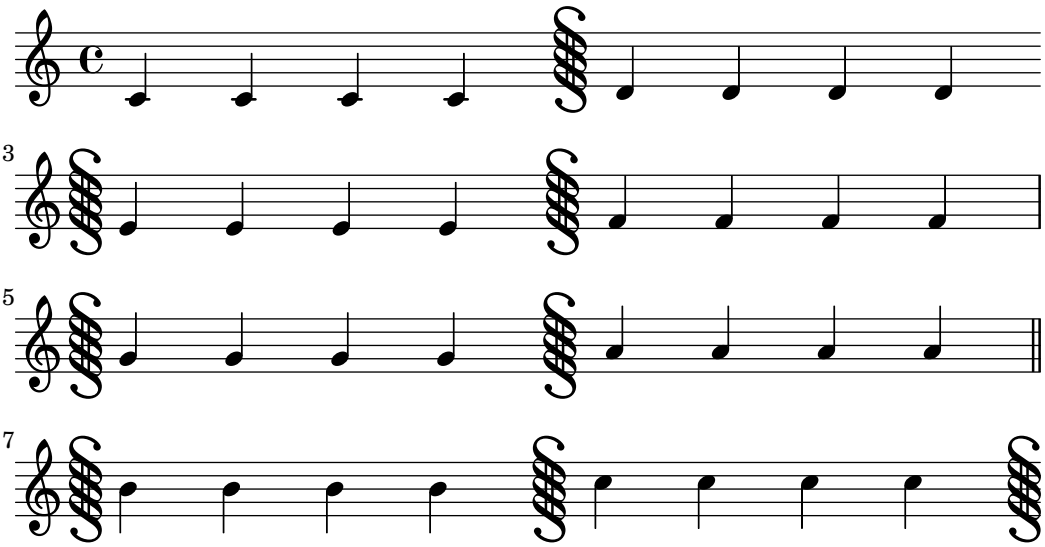
```
f'1 \bar "k"
```

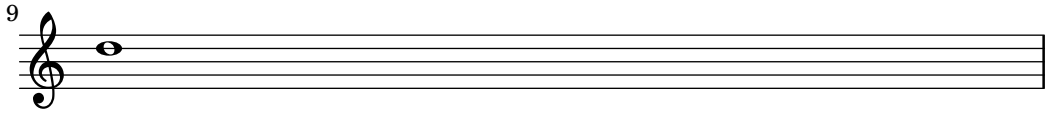


Pueden verse más detalles de esta notación explicados en Sección 17.5 [Tipografiado del canto kievano en notación cuadrada], página 562.

Hay varios símbolos de segno dentro del compás que se diferencian en su comportamiento en los saltos de línea:

```
\fixed c' {
  c4 4 4 4
  \bar "S"
  d4 4 4 4 \break
  \bar "S"
  e4 4 4 4
  \bar "S-|"
  f4 4 4 4 \break
  \bar "S-|"
  g4 4 4 4
  \bar "S-||"
  a4 4 4 4 \break
  \bar "S-||"
  b4 4 4 4
  \bar "S-S"
  c'4 4 4 4 \break
  \bar "S-S"
  d'1
}
```

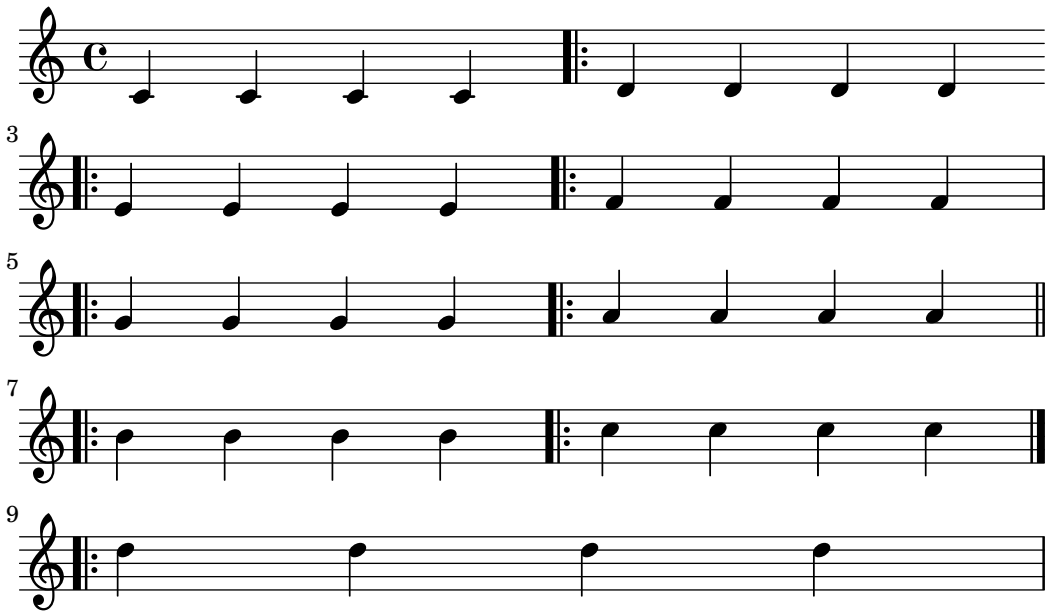




Aunque se pueden insertar manualmente barras de compás con significado de repeticiones, no se reconocen como repeticiones por parte de LilyPond. Las secciones repetidas se introducen mejor utilizando las diversas instrucciones de repetición (véase Capítulo 4 [Repeticiones], página 190), que imprimen automáticamente las barras correspondientes y que se pueden personalizar (véase Sección 2.5.2 [Automatic bar lines], página 130).

Además se puede especificar `".|:-||"`, que equivale a `".|:"` excepto en los saltos de línea, en que produce una doble barra al final de la línea y una repetición izquierda (de comienzo) al principio de la línea siguiente.

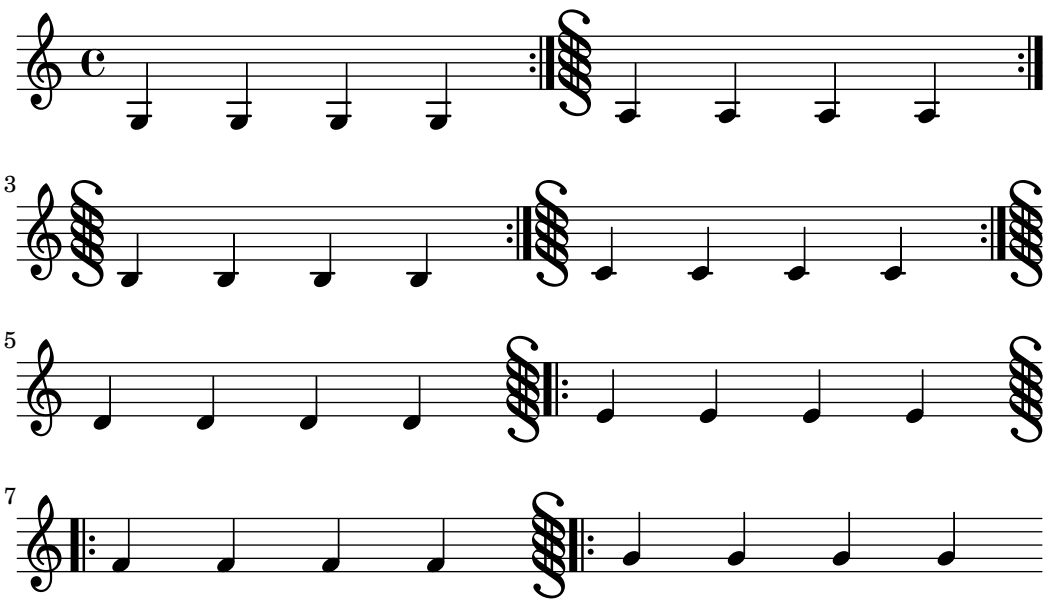
```
\fixed c' {
  c4 4 4 4
  \bar ".|:"
  d4 4 4 4 \break
  \bar ".|:"
  e4 4 4 4
  \bar ".|:-|"
  f4 4 4 4 \break
  \bar ".|:-|"
  g4 4 4 4
  \bar ".|:-||"
  a4 4 4 4 \break
  \bar ".|:-||"
  b4 4 4 4
  \bar ".|:-|."
  c'4 4 4 4 \break
  \bar ".|:-|."
  d'4 4 4 4
}
```

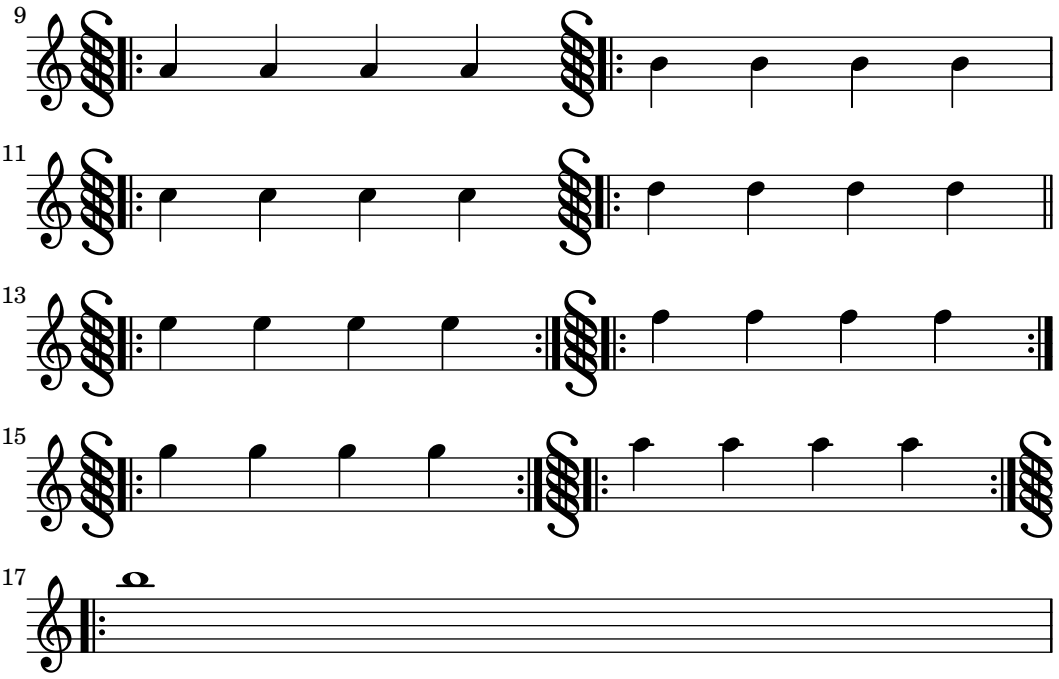


Para combinaciones de repeticiones con el símbolo de segno, existen seis variantes diferentes:

```
\fixed c' {
```

```
  g,4 4 4 4
  \bar ":|.S"
  a,4 4 4 4 \break
  \bar ":|.S"
  b,4 4 4 4
  \bar ":|.S-S"
  c4 4 4 4 \break
  \bar ":|.S-S"
  d4 4 4 4
  \bar "S.|:-S"
  e4 4 4 4 \break
  \bar "S.|:-S"
  f4 4 4 4
  \bar "S.|:"
  g4 4 4 4 \break
  \bar "S.|:"
  a4 4 4 4
  \bar "S.|:-|"
  b4 4 4 4 \break
  \bar "S.|:-|"
  c'4 4 4 4
  \bar "S.|:-||"
  d'4 4 4 4 \break
  \bar "S.|:-||"
  e'4 4 4 4
  \bar ":|.S.|:"
  f'4 4 4 4 \break
  \bar ":|.S.|:"
  g'4 4 4 4
  \bar ":|.S.|:-S"
  a'4 4 4 4 \break
  \bar ":|.S.|:-S"
  b'1
}
```





Muchas de las líneas divisorias de repetición y de segno que se han mostrado arriba se pueden insertar automáticamente con las instrucciones `\repeat` (véase Capítulo 4 [Repeticiones], página 190).

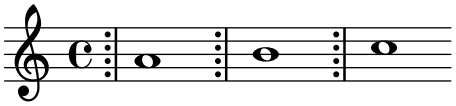
Se pueden definir tipos nuevos de líneas divisorias con `\defineBarLine`:

`\defineBarLine` *`tipo_de_barra`* `#'(`*`final comienzo extensión`*`)`

Dicho brevemente, el argumento *tipo-de-barra* especifica el glifo de línea divisoria que usar en mitad de un pentagrama, y sirve también como el nombre por el que este tipo de línea divisoria se referencia. Los otros argumentos especifican el glifo de la línea divisoria que usar al final o al principio de una línea, o en el trozo que abarca varios pentagramas. Establecer una cualquiera de *eol-bar*, *bol-bar*, o *span-bar* a `#t` significa usar el mismo tipo de barra de compás especificado por *tipo-de-barra* para la posición correspondiente. Establecerlas a `#f` significa no imprimir ninguna línea divisoria en la posición correspondiente.

Más detalladamente, el argumento *tipo-de-barra* es una cadena de caracteres que sirve un doble propósito: especifica el glifo de línea divisoria que imprimir cuando se produce en mitad de un pentagrama; e identifica el objeto de línea divisoria que se puede invocar con `\bar` *`tipo-de-barra`*. Debe tener la forma *midglyph* o *midglyph-annotation* (con un signo menos literal), donde *annotation* es una cadena de caracteres arbitraria, y *midglyph* es una cadena en la que cada uno de sus caracteres es el nombre de uno de los elementos predefinidos de líneas de compás que se listan más abajo. El glifo de línea divisoria resultante que usar en mitad de una línea es la concatenación de estos elementos. Por ejemplo, un *tipo-de-barra* de `";|"` o bien de `";|-other"` especifica una barra de compás compuesta que consiste en una línea de puntos (‘;’) emparejada a una línea continua (‘|’):

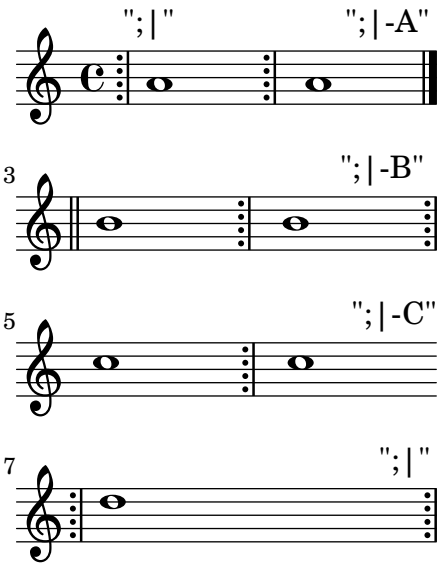
```
\defineBarLine ";|" #'(#t #t #t)
\defineBarLine ";|-other" #'(#f #f #f)
\fixed c' {
   \bar ";|" a1 \bar ";|" b1 \bar ";|-other" c'1 \bar ";|-other"
}
```



La *annotation* (‘`other`’ en el segundo ejemplo de arriba) se usa para diferenciar este tipo de barra de otros que tienen el mismo *midglyph* pero distinto comportamiento de salto de línea o de multi-pauta (por convenio, la cadena que se especifica en *eol-bar* se usa con frecuencia como la anotación, de manera que podríamos haber llamado al segundo ejemplo ‘`";|-f"`’.)

Los arguments *eol-bar* y *bol-bar* especifican la línea divisoria que imprimir al final de la línea y al principio de la siguiente, cuando `\bar` *`tipo-de-barra`* se produce juste en un salto de línea. *bol-bar* se aplica también cuando `\bar` *`tipo-de-barra`* se usa al comienzo de una partitura. El formato de estos argumentos es el mismo que el de *tipo-de-barra*. La cadena de caracteres de los elementos de la línea divisoria especifica el glifo de barra que imprimir en la posición de la línea correspondiente. Además, los argumentos pueden o bien ser `#t` como atajo para copiar el valor de *tipo-de-barra*, o bien `#f` para no imprimir ninguna línea divisoria. Por ejemplo, todos los tipos de barra de compás en este ejemplo imprimen un par de barras punteada/continua en el medio de una línea, pero tienen un comportamiento diferente en los finales y comienzos de los pentagramas:

```
%   dotted-solid everywhere
\defineBarLine  ";|"  #'( #t #t #t)
%   solid-bold at EOL, solid-solid at BOL
\defineBarLine  ";|-A" #'( "|."  "||"  #f )
%   dotted-solid at EOL, nothing at BOL
\defineBarLine  ";|-B" #'( #t  #f  #f )
%   nothing at EOL, dotted-solid at BOL
\defineBarLine  ";|-C" #'( #f  #t  #f )
\relative c'' {
  \bar ";|"  \textMark "\";|\""
  a1  \bar ";|-A"
  a1  \bar ";|-A"  \textEndMark "\";|-A\""  \break
  b1  \bar ";|-B" b \bar ";|-B"  \textEndMark "\";|-B\""  \break
  c1  \bar ";|-C" c \bar ";|-C"  \textEndMark "\";|-C\""   \break
  d1  \bar ";|" \textEndMark "\";|\""
}
```

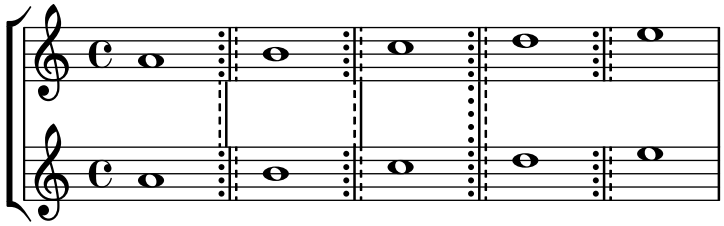


Nota: las cadenas *eol-bar* o *bol-bar* pueden ser nombres de tipos de barra de compás definidos previamente. En un contexto de pentagrama único, esto no tiene importancia: los elementos de la barra de compás en la cadena dada se usan independientemente de cualquier aspecto de la barra definida que se ha nombrado en la cadena. Especialmente, en este contexto cualquier anotación

se ignora. Sin embargo, en un sistema de varios pentagramas es importante que *eol-bar* y *bol-bar* se refieran a tipos de barra de compás previamente definidos (incluyendo el propio *tipo-de-barra*), o que sean `#t` o `#f`. Esto se explica en más detalle abajo.

El argumento *span-bar* solo tiene efecto en sistemas de varios pentagramas (véase Sección 6.1.2 [Agrupar pentagramas], página 246), en los que especifica qué imprimir entre las pautas agrupadas. Este argumento debe ser una cadena de elementos de línea divisoria, de la misma longitud que *tipo-de-barra* o más corto (los elementos adicionales se ignoran). Cada elemento se imprime en línea con el elemento correspondiente de *tipo-de-barra*. Se puede usar un carácter de espacio en blanco (‘ ’) para omitir el elemento de barra, pero hay que dejar espacio para él. Si se fija el valor de *span-bar* a `#t` hace lo mismo que el glifo para la mitad de la línea. Se se fija a `#f`, se omite la barra extendida, y si se fija a `""` (la cadena vacía) se produce una barra extendida de anchura nula. He aquí algunos ejemplos:

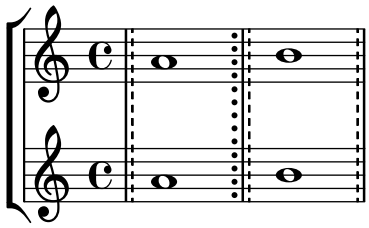
```
\defineBarLine  ";|!-A"  #'(#t #t "!|")
\defineBarLine  ";|!-B"  #'(#t #t " !|")
\defineBarLine  ";|!-C"  #'(#t #t #t)
\defineBarLine  ";|!-D"  #'(#t #t #f)
\fixed c' {
  \new StaffGroup <<
    \new Staff {
      a1 \bar ";|!-A"
      b  \bar ";|!-B"
      c' \bar ";|!-C"
      d' \bar ";|!-D"
      e'
    }
    \new Staff {
      a1 b c' d' e'
    }
  >>
}
```



En los saltos de línea, en lugar de usar *span-bar*, los tipos de barra de compás referenciados por *eol-bar* y *bol-bar* se usan para determinar qué barra de extensión imprimir. En el contexto de un grupo de pautas, si *eol-bar* y *bol-bar* no se refieren a tipos de barra definidos, LilyPond emite una advertencia, y no se imprime ninguna barra de extensión en los saltos de línea. Se permite que *eol-bar* o *bol-bar* sean `#f`, en cuyo caso no se imprime ninguna barra de extensión en la posición de la línea correspondiente. Estos argumentos también pueden ser `#t`, o de forma equivalente, igual a *tipo-de-barra*, en cuyo caso el *span-bar* en curso determina la barra de extensión que imprimir en la posición correspondiente de la línea:

```
\defineBarLine  "!|-t"  #'(#t #t #t)
\defineBarLine  "|!-t"  #'(#t #t #t)
\defineBarLine  ";|!-bad"  #'("|!" "!|" #t)  % fails at line breaks
\defineBarLine  ";|!-good" #'("!|-t" "|!-t" #t)
\relative c'' {
```

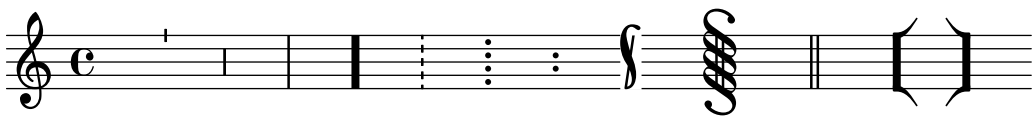
```
  \new StaffGroup <<
    \new Staff {
      \bar ";|!-good"
      a1 \bar ";|!-good"
      % \bar ";|!-bad"     % "WARNING: No span bar glyph defined..."
      b1 \bar ";|!-good"
    }
    \new Staff {
      a1 b1
    }
  >>
}
```



Nota: si *span-bar* es una cadena, debe contener *solamente* elementos de línea divisoriao un espacio (‘ ’), y no una anotación como las que se permiten en otros argumentos. Si *span-bar* contiene una anotación, LilyPond emite una advertencia. La única excepción es que puede ser igual a *tipo-de-barra*, en cuyo caso no se emite ninguna advertencia incluso si *tipo-de-barra* incluye una anotación.

Más abajo se muestran todos los elementoe de línea divisoria que están disponibles. Casi todos ellos también tienen tipos de barra predefinidos (para los argumentos de `\bar`) que los referencian de forma individual. Ciertos elementos están orientados principalmente a que se combinen con otros y por tanto no tienen tipos de barra individual predefinidos.

```
\defineBarLine ":" #'(#f #t #f)
\defineBarLine "=" #'(#t #f #t)
\defineBarLine "[" #'(#f #t #f)
\defineBarLine "]" #'(#t #f #f)

\new Staff {
  s1 \bar "'"
  s1 \bar ","
  s1 \bar "|"
  s1 \bar "."
  s1 \bar "!"
  s1 \bar ";"
  s1 \bar ":"
  s1 \bar "k"
  s1 \bar "S"
  s1 \bar "="
  s1 \bar "["
  s1 \bar "]"
  s1 \bar ""
}
```

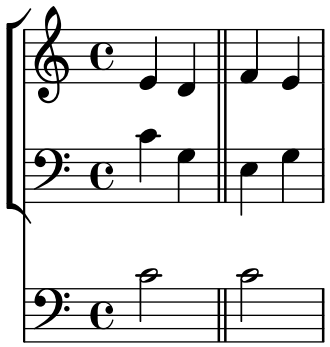


La línea divisoria "=" produce una barra de doble extensión para usarla en combinación con el símbolo de segno. No se recomienda usarla como una doble línea divisoria fina aislada; para ello es casi siempre preferible \bar "||".

Si se necesitan elementos adicionales, LilyPond provee una forma sencilla de definirlos. Para ver más información sobre cómo modificar o añadir barras de compás, consulte el archivo scm/bar-line.scm.

En las partituras con muchos pentagramas, una instrucción \bar en uno de ellos se aplica automáticamente a todos los demás. Las líneas resultantes se conectan entre los distintos pentagramas de un StaffGroup, PianoStaff o GrandStaff.

```
<<
  \new StaffGroup <<
    \new Staff \relative {
      e'4 d
      \bar "||"
      f4 e
    }
    \new Staff \relative { \clef bass c'4 g e g }
  >>
  \new Staff \relative { \clef bass c'2 c2 }
>>
```



El tipo de barra de compás usado para las barras insertadas automáticamente es "|". Esto se puede modificar en cualquier momento con '\set Timing.measureBarType = *bartype*'.

También es posible establecer diferentes tipos de barra en una partitura con varios pentagramas, usando una de las instrucciones o propiedades posibles que se explican abajo (véase Sección 2.5.2 [Automatic bar lines], página 130). Por descontado, esto puede dar lugar auna falta de concordancia entre líneas divisorias y líneas de extensión, debido a su distinta anchura.

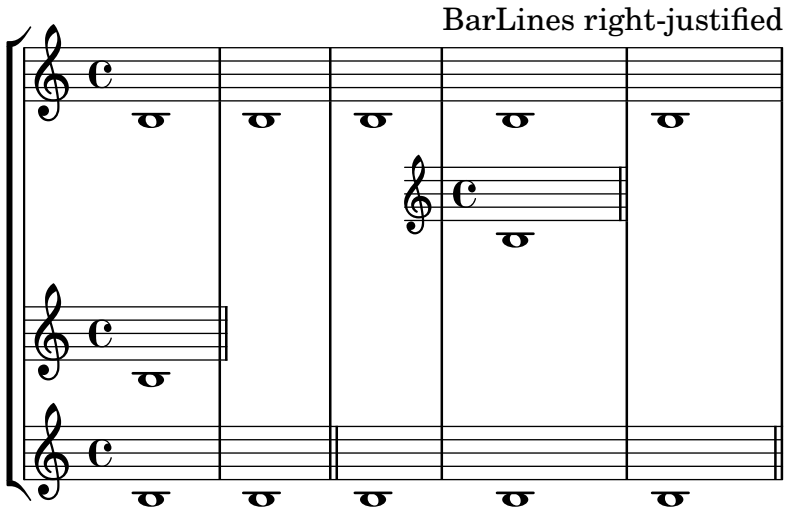
Normalmente las líneas de compás están alineadas a la izquierda (obviando los símbolos de dos puntos como en las barras de repetición). Para hacer que estén alineadas a la derecha, se tiene que aplicar la instrucción

```
[\once] \override Contexto.BarLine.right-justified = ##t
```

, donde *Contexto* es un contexto apropiado para más de un pentagrama, como *Score*, *StaffGroup*, *Grandstaff*, etc.

```
\new StaffGroup
  <<
    \new Staff = "a" {
```

```
    b1  b b
    <<
      { \textMark "BarLines right-justified" b b }
      \new Staff \with { alignAboveContext = "b" }
      {
        \override StaffGroup.BarLine.right-justified = ##t
        b
        \section
      }
    >>
  }
  \new Staff = "b" { b \section }
  \new Staff = "c" { b b \section b b b \section }
>>
```



Después de un salto de línea, las barras de compás nunca están alineadas a la derecha. Para las barras en medio del pentagrama alineadas a la derecha, el punto de ancla (usado para alinear `BarNumber`, `RehearsalMark`, etc.) se mueve de forma correspondiente.

Véase también

Referencia de la notación: Sección 28.1 [Saltos de línea], página 692, Capítulo 4 [Repeticiones], página 190, Sección 6.1.2 [Agrupar pentagramas], página 246.

Archivos instalados: `scm/bar-line.scm`.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BarLine” en *Referencia de Funcionamiento Interno* (creada al nivel de `Staff` (pentagrama)), Sección “SpanBar” en *Referencia de Funcionamiento Interno* (a través de los pentagramas), Sección “Timing_translator” en *Referencia de Funcionamiento Interno* (para las propiedades del contador de tiempo Timing).

2.5.2 Automatic bar lines

Varias instrucciones aparte de `\bar` pueden también crear barras de compás como parte de su efecto. Las barras de compás creadas en estos casos se pueden cambiar estableciendo propiedades de contexto. Si una propiedad se establece a `'()` o se deja sin configurar, se ignora; en otro caso, el valor debe ser un tipo de barra predefinido o uno que se haya definido previamente con la instrucción `\defineBarLine` (véase Sección 2.5.1 [Barras de compás], página 120).

Se pueden aplicar al mismo tiempo varios motivos para crear diferentes barras de compás automáticas. Los conflictos sse resuelven en parte aportando propiedades para combinaciones predeterminadas y en parte mediante un esquema de prioridades. La tabla que aparece más abajo presenta todas las propiedades disponibles en orden de prioridad creciente.

`underlyingRepeatBarType`
: Usado en puntos de repetición u origen de un salto, donde en otro caso no se imrpimiría ninguna barra. Esto es lo que se espera cuando las secciones repetidas no están alineadas con los compases. Varias instrucciones emplean este tipo de barra: `\codaMark`, `\inStaffSegno`, `\repeat segno`, `\repeat volta` y `\segnoMark`.

`caesuraType underlying-bar-line`
: Usado en `\caesura`; véase Sección 9.7.4 [Barras de frase en himnos], página 411.

`submeasureBarType`
: Utilizada en los límites de un sub-compás, especificados por `submeasureStructure`, cuando lo permite `\submeasureBarsOn`; véase Sección 2.3.1 [Indicación de compás], página 79.

`measureBarType`
: Usado en la frontera normal de un compás.

`caesuraType bar-line`
: Usado en `\caesura`; véase Sección 9.7.4 [Barras de frase en himnos], página 411.

`sectionBarType`
: Usado en un salto de sección creado por `\section`.

`fineBarType`
: Usado en `\fine`.

`doubleRepeatBarType`
`doubleRepeatSegnoBarType`
`endRepeatBarType`
`endRepeatSegnoBarType`
`fineSegnoBarType`
`fineStartRepeatSegnoBarType`
`segnoBarType`
`startRepeatBarType`
`startRepeatSegnoBarType`
: Solo uno de estos tipos de barra de compás se usa en un momento dado; la que se use depende de la estructura de la pieza.

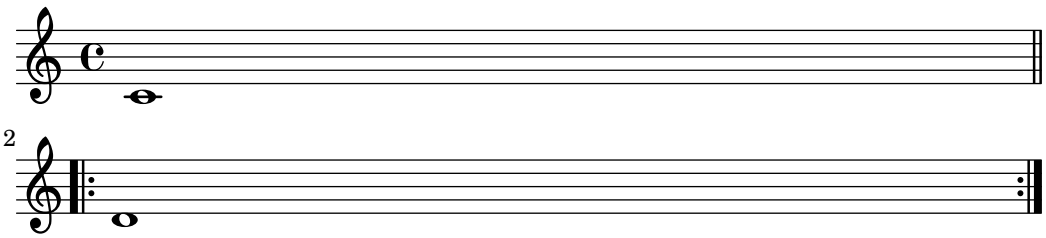
  Las propiedades que llevan `startRepeat` o `endRepeat` en el nombre, se usan al comienzo o al final de una sección repetida creada mediante `\repeat volta`, y las propiedades que tienen `doubleRepeat` en el nombre se usan cuando coinciden el final de una sección repetida y el principio de otra.

  Las propiedades que llevan `segno` en el nombre, se usan en un segno dentro del pentagrama, que se puede crear por parte de `\repeat segno` o de `\segnoMark` cuando la propiedad `segnoStyle` está establecida a `bar-line`, o creada por parte de `\inStaffSegno`.

  Las porpiedades que llevan `fine` en el nombre, se usan en `\fine`.

La prioridad se aplica de forma independiente a las líneas divisorias al principio, mitad o final de los pentagramas, permitiendo que aparezca una barra de compás de menor prioridad donde los tipos de barra de alta prioridad no tienen ningún glifo definido (véase Sección 2.5.1 [Barras de compás], página 120).

```
\fixed c' {
  c1 \section \break
  \repeat volta 2 d1
}
```

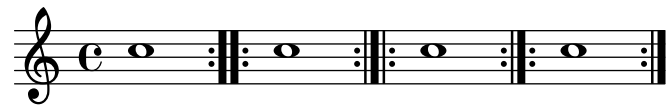


Fragmentos de código seleccionados

Establecer el tipo de repetición doble predeterminado para la primera y segunda vez

Existen distintos estilos de repeticiones dobles para la primera y segunda vez, que se pueden seleccionar utilizando la propiedad de contexto `doubleRepeatBarType`.

```
\relative c'' {
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":..:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":|.|:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = ":|.:"
  \repeat volta 2 { c1 }
}
```



Véase también

Referencia de la notación: Sección 2.5.1 [Barras de compás], página 120, Capítulo 4 [Repeticiones], página 190.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

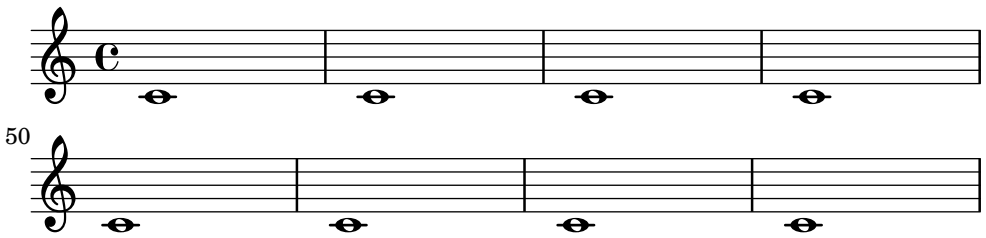
Referencia de funcionamiento interno: Sección “Repeat_acknowledge_engraver” en *Referencia de Funcionamiento Interno*.

2.5.3 Numeración de los compases

Por defecto, los números de compás se imprimen al principio de la línea, excepto la primera. El número propiamente dicho se almacena en la propiedad `currentBarNumber`, que normalmente se actualiza automáticamente para cada compás. También se puede establecer manualmente:

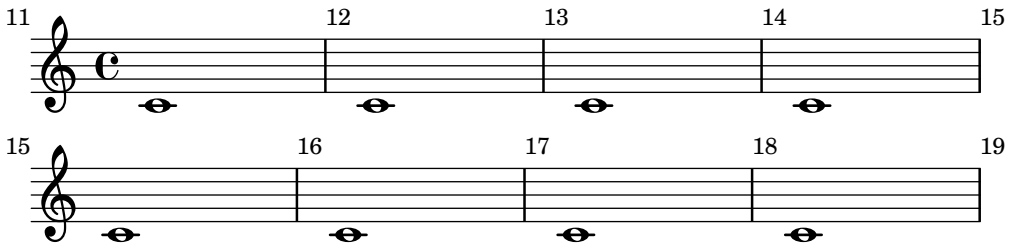
```
\relative c' {
  c1 c c c
  \break
```

```
  \set Score.currentBarNumber = 50
  c1 c c c
}
```



El comportamiento predeterminado de imprimir solo números de compás al comienzo de cada línea se puede modificar a través de la propiedad `break-visibility` de `BarNumber`. Toma tres valores que se pueden fijar al valor `#t` o `#f` para especificar si el número de compás correspondiente es visible o no. El orden de los tres valores es `visible al final de la línea`, `visible en mitad de la línea`, `visible al principio de la línea`. En el ejemplo siguiente se imprimen los números de compás en todos los sitios posibles:

```
\relative c' {
  \override Score.BarNumber.break-visibility = ##(#t #t #t)
  \set Score.currentBarNumber = 11
  c1 | c | c | c |
  \break
  c1 | c | c | c |
}
```

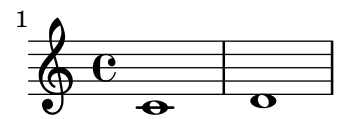


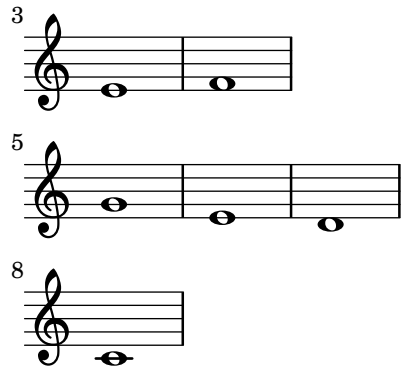
Fragmentos de código seleccionados

Imprimir el número de compás en el primer compás

De forma predeterminada se suprime el número del primer compás de una partitura si es menor o igual a to 1. Esto se puede cambiar fijando la propiedad de contexto `barNumberVisibility` con el valor `all-bar-numbers-visible`

```
\paper {
  line-width = 50\mm
}

\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  c1 | d | e | f \break
  g1 | e | d | c
}
```

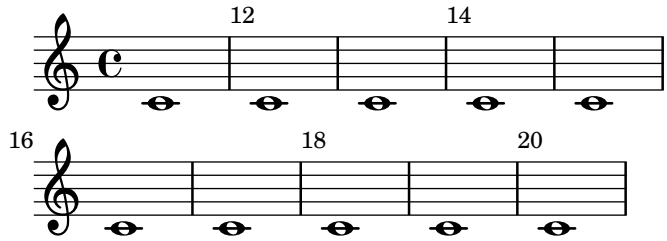




Imprimir números de compás a intervalos regulares

Se pueden imprimir los números de compás a intervalos regulares mediante el establecimiento de la propiedad `barNumberVisibility`. Aquí los números de compás se imprimen a cada dos compases excepto al final de la línea.

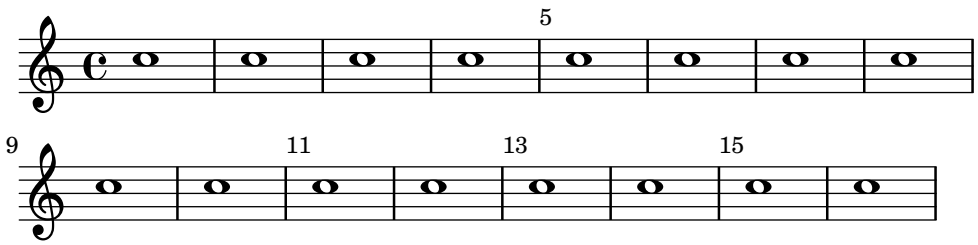
```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = 11
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```

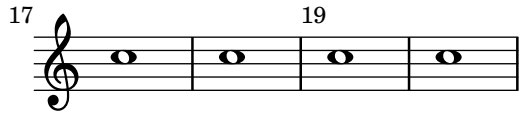


Impresión de los números de compás con intervalos regulares variables

Usando la función de contexto `set-bar-number-visibility`, se pueden cambiar los intervalos de numeración de los compases.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \context Score \applyContext #(set-bar-number-visibility 4)
  \repeat unfold 10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \repeat unfold 10 c
}
```

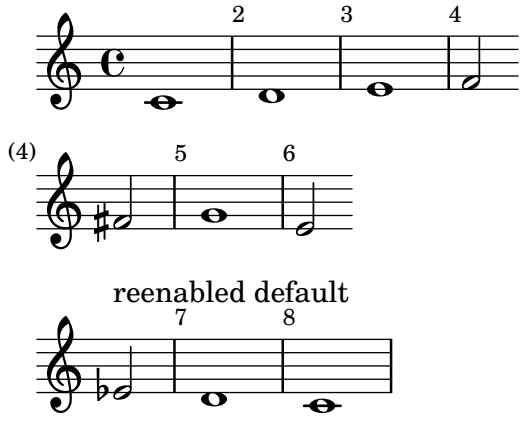




Impresión de números de compás para compases divididos

Por omisión, el número de un compás que está dividido no se repite al comienzo de la línea siguiente. Use `first-bar-number-invisible-save-broken-bars` para `barNumberVisibility` para tener un `BarNumber` encerrado entre paréntesis en ese caso.

```
\layout {
  \context {
    \Score
    barNumberVisibility = #first-bar-number-invisible-save-broken-bars
    \override BarNumber.break-visibility = ##(#f #t #t)
  }
}

\relative c' {
  c1 | d | e | f2 \break
  fis2 | g1 | e2 \break
  <>^"reenabled default"
  % back to default -
  % \unset Score.barNumberVisibility would do so as well
  \set Score.barNumberVisibility =
    #first-bar-number-invisible-and-no-parenthesized-bar-numbers
  es2 | d1 | c
}
```



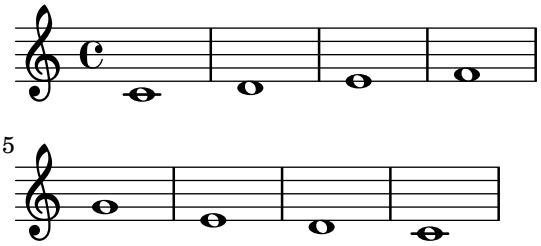
Impresión de los números de compás usando `modulo-bar-number-visible`

Si el resto de dividir el número del compás en curso entre el primer argumento del `modulo-bar-number-visible` es igual al segundo argumento, entonces imprimir el número del compás.

Es útil para imprimir el número de compás a ciertas distancias. Algunos ejemplos:

- `(modulo-bar-number-visible 3 2)` → imprime 2, 5, 8, ...
- `(modulo-bar-number-visible 4 2)` → imprime 2, 6, 10, ...
- `(modulo-bar-number-visible 2 1)` → imprime 3, 5, 7, ...
- `(modulo-bar-number-visible 5 0)` → imprime 5, 10, 15, ...

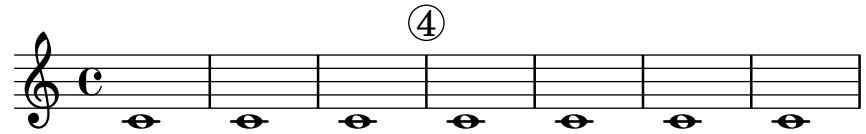
```
\layout {
  \context {
    \Score
    \override BarNumber.break-visibility = ##(#f #t #t)
    barNumberVisibility = #(modulo-bar-number-visible 5 0)
  }
}

\relative c' {
  c1 | d | e | f \break
  g1 | e | d | c
}
```

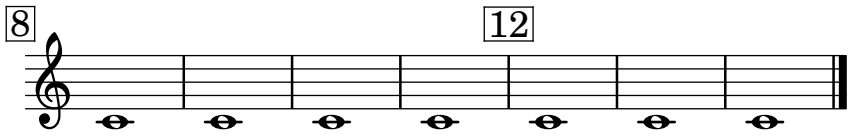


Imprimir números de compás dentro de rectángulos o circunferencias

Los números de compás también se pueden imprimir dentro de rectángulos o de circunferencias.

```
\relative c' {
  % Center bar numbers except at the beginning of a staff.
  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER CENTER 0.3)

  % Prevent bar numbers at the end of a line and permit them elsewhere.
  \override Score.BarNumber.break-visibility = #end-of-line-invisible

  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2.
  \override Score.BarNumber.font-size = 2

  % Draw a circle round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 7 { c1 } \break

  % Draw a box round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 7 { c1 } \bar "|."
}
```

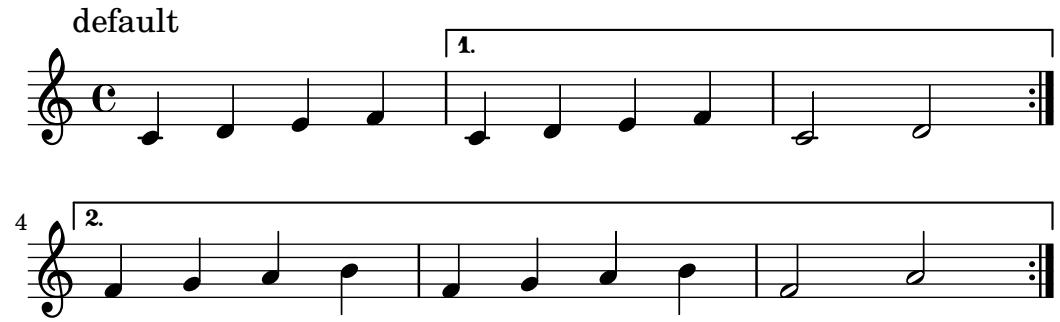


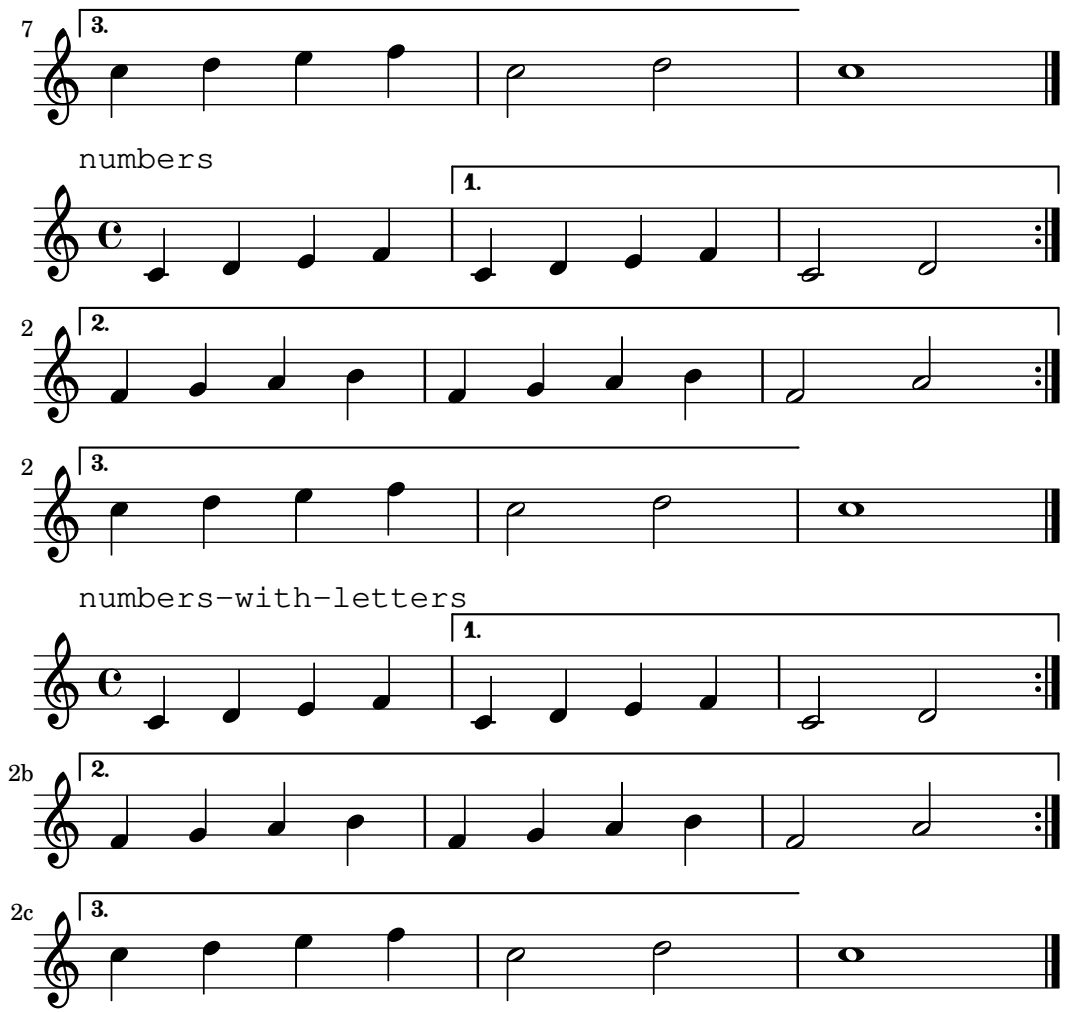


Numeración de compases alternativa

Fijando la propiedad de contexto `alternativeNumberingStyle`, se dispone de dos métodos alternativos adicionales para la numeración de compases dentro de las repeticiones.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|."
}

{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}
```



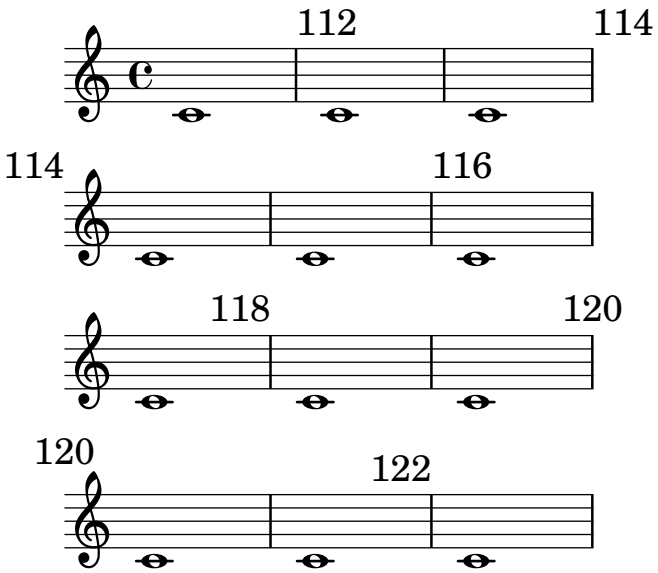


Alineación de los números de compás

La alineación predeterminada de los números de compás depende de su posición: al comienzo de un pentagrama, los números se alinean por la derecha; en cualquier otra posición, se alinean por la izquierda. Usando la función de Scheme `break-alignment-list`, se puede cambiar esto; los tres argumentos de la función son la alineación para las posiciones al final de la línea, en la parte media de la línea y al principio de la línea (en este orden).

```
\relative c' {
  \set Score.currentBarNumber = 111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = 2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)

  c1 | c1 | c1 | \break
  c1 | c1 | c1 | \break

  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER RIGHT CENTER)
  c1 | c1 | c1 | \break
  c1 | c1 | c1 |
}
```

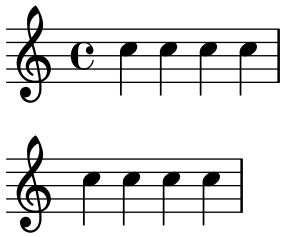
```
\paper {
  line-width = 70\mm
}
```



Suprimir los números de compás de toda la partitura

Se pueden eliminar completamente los números de compás quitando el grabador `Bar_number_engraver` del contexto de `Score`.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    % \remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```

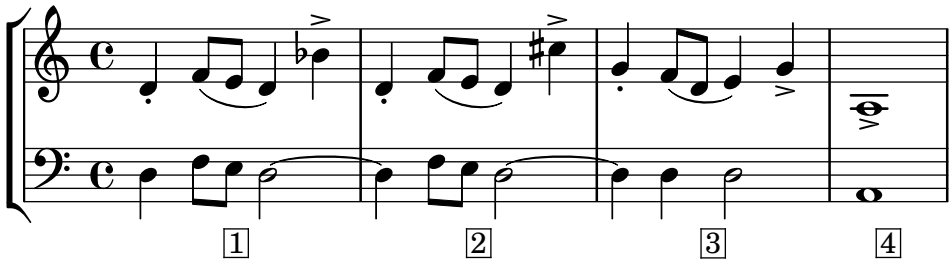


Número del compás centrado entre las barras

Para las partituras de bandas sonoras, es una convención común centrar los números de compás entre las líneas divisorias. Esto se consigue fijando la propiedad de contexto `centerBarNumbers` a `#t`. Cuando se usa esta sobreescritura, el tipo de objeto gráfico del número de compás es `CenteredBarNumber` y no `BarNumber`.

Este ejemplo muestra un número de ajustes: los números de compás centrados están encerrados en un ractángulo y se sitúan debajo de los pentagramas.

```
\layout {
  \context {
    \Score
    centerBarNumbers = ##t
    barNumberVisibility = #all-bar-numbers-visible
    \override CenteredBarNumber.stencil
      = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
    \override CenteredBarNumberLineSpanner.direction = #DOWN
  }
}

\new StaffGroup <<
  \new Staff \relative c' {
    d4-. f8( e d4) bes'-> |
    d,-. f8( e d4) cis'-> |
    g-. f8( d e4) g-> |
    a,1-> |
  }
  \new Staff \relative c {
    \clef bass
    d4 f8 e d2~ |
    4 f8 e d2~ |
    4 4 2 |
    a1 |
  }
>>
```



Véase también

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BarNumber” en *Referencia de Funcionamiento Interno*, Sección “CenteredBarNumber” en *Referencia de Funcionamiento Interno*, Sección “CenteredBarNumberLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “Bar_number_engraver” en *Referencia de Funcionamiento Interno*, Sección “Centered_bar_number_align_engraver” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los números de compás pueden colisionar con el corchete `StaffGroup`, si hay uno en la parte de arriba. Para solucionarlo, se puede usar la propiedad de relleno `padding` de `BarNumber` para colocar el número correctamente. Consulte Sección “StaffGroup” en *Referencia de Funcionamiento Interno* y Sección “BarNumber” en *Referencia de Funcionamiento Interno* para ver más información.

2.5.4 Comprobación de compás y de número de compás

Las comprobaciones de compás ayudan a detectar errores en las duraciones. Una comprobación de compás se escribe usando el símbolo de la barra vertical, `|`, en cualquier lugar donde se espera que caiga una línea divisoria. Si se encuentran líneas de comprobación de compás en otros lugares, se imprime una lista de advertencias en el archivo log de registro, mostrando los números de línea y columna en que han fallado las comprobaciones de compás. En el siguiente ejemplo, la segunda comprobación de compás avisará de un error.

```
\time 3/4 c2 e4 | g2 |
```

Una duración incorrecta produce una partitura completamente desbaratada, especialmente si la partitura es polifónica, de manera que la mejor forma de empezar a corregir la entrada es buscar sistemáticamente la existencia de comprobaciones de compás fallidas y duraciones incorrectas.

Si se producen varias comprobaciones de compás seguidas por valor de la misma duración musical, solo aparece el primer mensaje de advertencia. Así se consigue que el mensaje esté concentrado en la fuente del error de pulso.

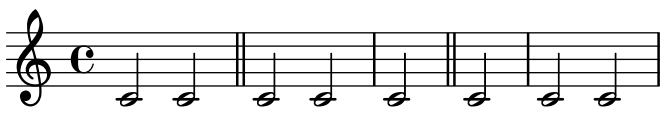
Las comprobaciones de compás también se pueden insertar dentro de la letra de las canciones:

```
\lyricmode {
  \time 2/4
  Twin -- kle | Twin -- kle |
}
```

Observe que las comprobaciones de compás dentro de la letra de las canciones se evalúan en el momento musical de la sílaba *siguiente* a la comprobación de compas que se procesa. Si la letra está asociada con las notas de una voz que tiene un silencio al principio del compás, no se puede poner ninguna sílaba al comienzo de ese compás y se imprimie un mensaje de advertencia si se escribe una comprobación de compás dentro de la letra, en esa posición.

También es posible redefinir la acción que se lleva a cabo cuando se encuentra una barra vertical o símbolo de comprobación de compás, `|`, en el código de entrada, de forma que haga algo distinto a una comprobación de compás. Se hace asignando una expresión musical a `"|"`, el símbolo de barra vertical. En el ejemplo siguiente se establece `|` de forma que inserte una doble línea divisoria cuando aparece en el código de entrada, en vez de comprobar el final de un compás.

```
"|" = \bar "||"
{
  c'2 c' |
  c'2 c'
  c'2 | c'
  c'2 c'
}
```



Al copiar piezas musicales grandes puede servir de ayuda comprobar que el número de compás de LilyPond corresponde al original desde el que está copiando. Esto se puede comprobar con `\barNumberCheck`, por ejemplo:

```
\barNumberCheck 123
```

imprime una advertencia si el `currentBarNumber` (número del compás actual) no es el 123 en el momento de la compilación.

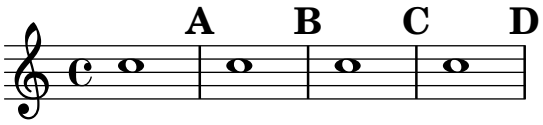
Véase también

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

2.5.5 Llamadas de ensayo

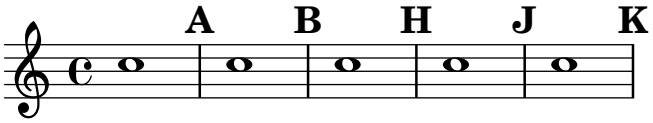
Para imprimir una letra de ensayo, utilice la orden `\mark`.

```
\relative c'' {
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
}
```



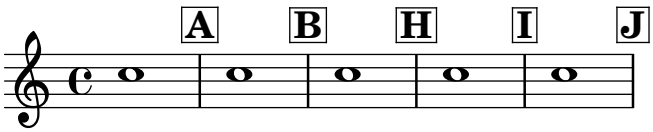
La letra de ensayo se incrementa automáticamente si usa `\mark \default`, pero también puede utilizar un número entero como argumento para establecer la indicación manualmente. El valor que se utilizará se almacena dentro de la propiedad `rehearsalMark`.

```
\relative c'' {
  c1 \mark \default
  c1 \mark \default
  c1 \mark 8
  c1 \mark \default
  c1 \mark \default
}
```



La letra ‘I’ se salta de acuerdo con las tradiciones de grabado. Si quiere incluir la letra ‘I’, use una de las instrucciones siguientes, según el estilo de marcas de ensayo que desee (solo letras, letras dentro de un rectángulo o letras dentro de un círculo).

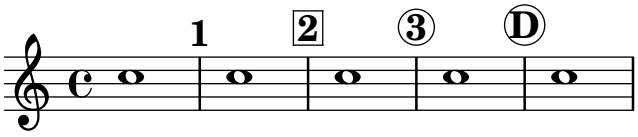
```
\set Score.rehearsalMarkFormatter = #format-mark-alphabet
\set Score.rehearsalMarkFormatter = #format-mark-box-alphabet
\set Score.rehearsalMarkFormatter = #format-mark-circle-alphabet
\relative c'' {
  \set Score.rehearsalMarkFormatter = #format-mark-box-alphabet
  c1 \mark \default
  c1 \mark \default
  c1 \mark 8
  c1 \mark \default
  c1 \mark \default
}
```



El estilo se define por medio de la propiedad `rehearsalMarkFormatter`. Es una función que toma como argumentos la marca en curso (un número entero) y el contexto actual. Debe devolver

un objeto de marcado. En el ejemplo siguiente, se establece rehearsalMarkFormatter con el valor de un procedimiento enlatado. Después de algunos compases, se establece con el valor de una función que produce un número encerrado en una caja.

```
\relative c'' {
  \set Score.rehearsalMarkFormatter = #format-mark-numbers
  c1 \mark \default
  c1 \mark \default
  \set Score.rehearsalMarkFormatter = #format-mark-box-numbers
  c1 \mark \default
  \set Score.rehearsalMarkFormatter = #format-mark-circle-numbers
  c1 \mark \default
  \set Score.rehearsalMarkFormatter = #format-mark-circle-letters
  c1
}
```



El archivo scm/translation-functions.scm contiene las definiciones de format-mark-letters (el formato por omisión), format-mark-box-letters, format-mark-numbers y format-mark-box-numbers. Se pueden usar éstos como inspiración para otras funciones de formateo.

Podemos utilizar format-mark-barnumbers, format-mark-box-barnumbers y format-mark-circle-barnumbers para obtener números de compás en lugar de números o letras secuenciales.

Para los trucajes más habituales en el posicionamiento de las llamadas de ensayo, véase Sección 8.2 [Formatear el texto], página 323. Para un control más preciso, considere break-alignable-interface (véase Sección 36.9 [Alineación de objetos], página 801).

El archivo scm/translation-functions.scm contiene las definiciones de format-mark-numbers y de format-mark-letters. Se pueden usar como inspiración para otras funciones de formateo.

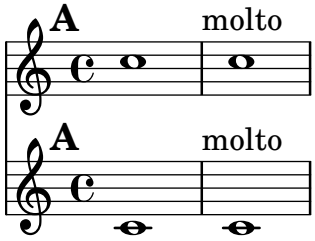
Fragmentos de código seleccionados

Imprimir marcas de ensayo en cualquier pentagrama

Aunque normalmente las marcas de ensayo y textuales solo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
    }
```

```
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}
```



Véase también

Referencia de la notación: Sección B.8 [La tipografía Emmentaler], página 912, Sección 8.2 [Formatear el texto], página 323, Sección 36.9 [Alineación de objetos], página 801, Sección 8.2.5 [Notación musical dentro de elementos de marcado], página 338, Sección 2.3.2 [Indicaciones metronómicas], página 86, Sección 8.1.4 [Etiquetas de sección], página 316.

Archivos instalados: `scm/translation-functions.scm`.

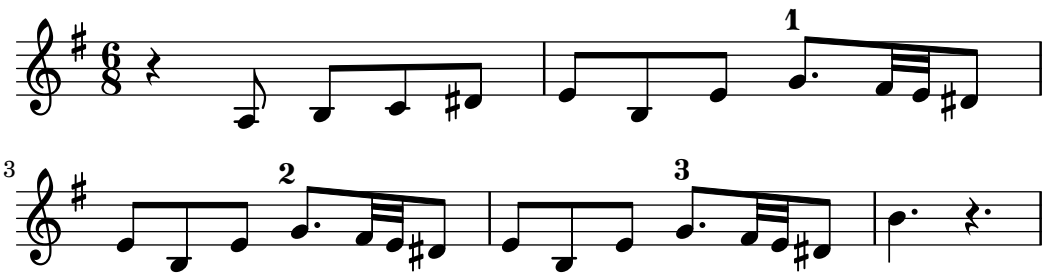
Fragmentos de código: Sección "Duraciones" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "AdHocMarkEvent" en *Referencia de Funcionamiento Interno*, Sección "RehearsalMark" en *Referencia de Funcionamiento Interno*, Sección "RehearsalMarkEvent" en *Referencia de Funcionamiento Interno*.

2.5.6 Cuentas de compases

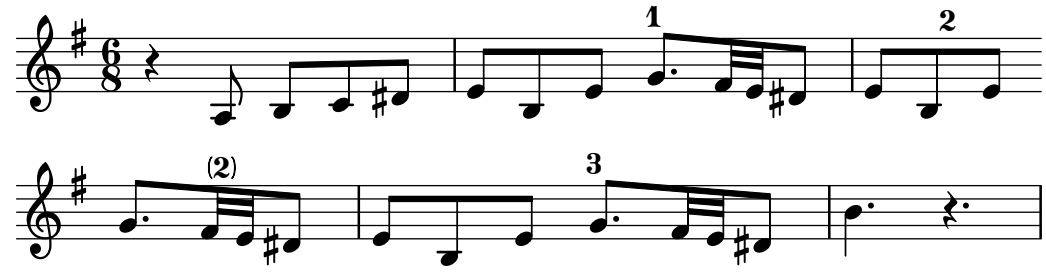
Las cuentas de compases son una forma de numerar compases consecutivos, por ejemplo como ayuda para que los músicos lleven la cuenta durante las repeticiones desplegadas. El uso de esta funcionalidad requiere que se añada el grabador `Measure_counter_engraver` a un tipo de contexto, normalmente `Staff` o `Score`.

```
\layout {
  \context {
    \Staff
    \consists Measure_counter_engraver
  }
}

\relative c' {
  \time 6/8
  \key e \minor
  r4 a8 b c dis
  \startMeasureCount
  \repeat unfold 3 {
    e8 b e g8. fis32 e dis8
  }
  \stopMeasureCount
  b'4. r
}
```



Los compases partidos se numeran entre paréntesis.

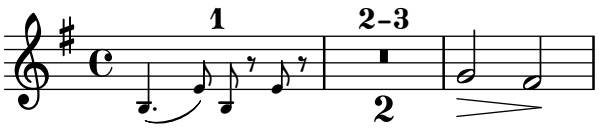
```
\layout {
  \context {
    \Staff
    \consists Measure_counter_engraver
  }
}

\relative c' {
  \time 6/8
  \key e \minor
  r4 a8 b c dis
  \startMeasureCount
  e8 b e g8. fis32 e dis8
  e8 b e \break g8. fis32 e dis8
  e8 b e g8. fis32 e dis8
  \stopMeasureCount
  b'4. r
}
```



Los silencios multicompás comprimidos reciben un tratamiento especial: se muestra todo el rango de compases.

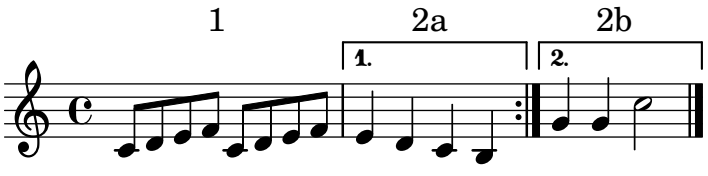
```
\layout {
  \context {
    \Staff
    \consists Measure_counter_engraver
  }
  \context {
    \Voice
    \override MultiMeasureRestNumber.direction = #DOWN
  }
}

\compressMMRests {
  \key e \minor
```

```
  \startMeasureCount
  \new CueVoice {
    b4.( e'8) b8 r e' r
  }
  R1*2
  \stopMeasureCount
  g'2\> fis'2\!
}
```



Los contadores de compás respetan los estilos de numeración alternativos. Si el estilo es `numbers-with-letters`, se imprimen mejor con fuentes tipográficas de texto.

```
\layout {
  \context {
    \Score
    alternativeNumberingStyle = #'numbers-with-letters
  }
  \context {
    \Staff
    \consists Measure_counter_engraver
    \override MeasureCounter.Y-offset = 6
    \override MeasureCounter.font-encoding = #'latin1
    \override MeasureCounter.font-size = 1
  }
}

\relative c' {
  \startMeasureCount
  \repeat volta 2 {
    c8 d e f c d e f
  }
  \alternative {
    { e4 d c b }
    { g'4 g c2 }
  }
  \bar "|."
  \stopMeasureCount
}
```



Instrucciones predefinidas

`\startMeasureCount`, `\stopMeasureCount`.

Véase también

Referencia de la notación: Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752, Sección 6.3.4 [Compresión de los compases vacíos], página 277, Capítulo 28 [Saltos], página 692, Sección 2.5.3 [Numeración de los compases], página 132.

Referencia de funcionamiento interno: Sección "Measure_counter_engraver" en *Referencia de Funcionamiento Interno*, Sección "MeasureCounter" en *Referencia de Funcionamiento Interno*, Sección "measure-counter-interface" en *Referencia de Funcionamiento Interno*.

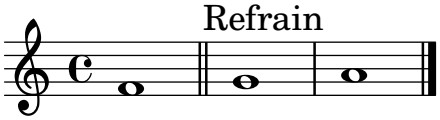
2.5.7 Divisores de sección

La instrucción `\section` marca un punto en el que termina una sección demúsica y comienza otra. No tiene que ir seguida de más música: se puede usar también para enfatizar que el final escrito de la música no es el final de la pieza, como en una indicación *D.C.* o donde un movimiento se continúa directamente con el siguiente sin nungún salto. Normalmente `\section` crea una doble barra, pero su efecto puede depender de otros elementos de la notación, p.ej. barras de repetición.

La instrucción `\fine` finaliza la pieza, normalmente con una doble barra final. Su uso no se limita al final escrito de la música: también puede aparecer dentro de `\repeat` (véase Sección 4.1.5 [Repeticiones Al fine], página 196).

Opcionalmente, una sección puede recibir una etiqueta de nombre con `\sectionLabel` (véase Sección 8.1.4 [Etiquetas de sección], página 316).

```
\fixed c' {
  f1
  \section
  \sectionLabel "Refrain"
  g1
  a1
  \fine
}
```



Para más detalles acerca de las interacciones entre las barras de `\fine` o de `\section` y otros tipos de barras, y opciones para alterar su apariencia, véase Sección 2.5.2 [Automatic bar lines], página 130.

Véase también

GLosario musical: Sección "fine" en *Glosario Musical*.

Referencia de la notación: Sección 2.5.2 [Automatic bar lines], página 130, Sección 17.4.4 [Divisiones], página 554, Sección 4.1.5 [Repeticiones Al fine], página 196, Sección 8.1.4 [Etiquetas de sección], página 316.

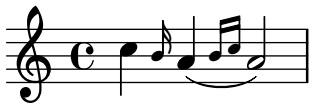
Referencia de funcionamiento interno: Sección "FineEvent" en *Referencia de Funcionamiento Interno*, Sección "SectionEvent" en *Referencia de Funcionamiento Interno*.

2.6 Asuntos rítmicos especiales

2.6.1 Notas de adorno

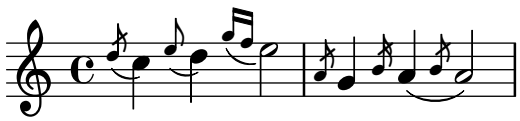
Los mordentes y notas de adorno son ornamentos musicales, impresos en un tipo más pequeño y no ocupan ningún tiempo lógico adicional en el compás.

```
\relative {
  c''4 \grace b16 a4(
  \grace { b16 c16 } a2)
}
```



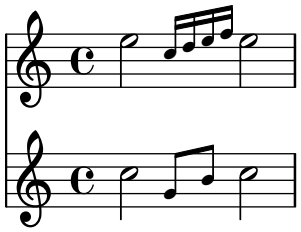
Hay otros tres tipos posibles de notas de adorno; la *acciaccatura* o mordente de una nota (una nota de adorno sin medida que se indica mediante una nota ligada y con el corchete tachado) y la *appoggiatura*, que toma una fracción fija de la nota principal a la que se adjunta, y que se imprime sin tachar. Es posible escribir una nota de adorno con la plica tachada, como la *acciaccatura* pero sin la ligadura, como para colocarla entre notas que están ligadas entre sí, utilizando la función `\slashedGrace`.

```
\relative {
  \acciaccatura d''8 c4
  \appoggiatura e8 d4
  \acciaccatura { g16 f } e2
  \slashedGrace a,8 g4
  \slashedGrace b16 a4(
  \slashedGrace b8 a2)
}
```



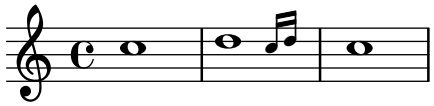
La colocación de notas de adorno se sincroniza entre los distintos pentagramas. En el siguiente ejemplo, hay dos semicorcheas de adorno por cada corchea de adorno:

```
<<
  \new Staff \relative { e''2 \grace { c16 d e f } e2 }
  \new Staff \relative { c''2 \grace { g8 b } c2 }
>>
```



Si queremos terminar una nota con un adorno, usamos la instrucción `\afterGrace`. Toma dos argumentos: la nota principal, y las notas de adorno que siguen a la nota principal.

```
\relative { c''1 \afterGrace d1 { c16[ d] } c1 }
```



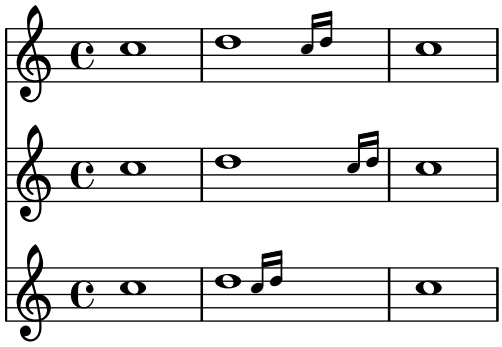
Esto sitúa las notas de adorno *después* del comienzo de la nota principal. El instante temporal en que se colocan las notas de adorno es una fracción dada de la duración de la nota principal. El ajuste predeterminado de

```
afterGraceFraction = 3/4
```

se puede redefinir en el nivel jerárquico superior. Las instrucciones `\afterGrace` individuales pueden, en vez de esto, llevar especificada la fracción directamente, después de la propia instrucción.

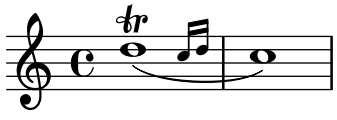
El siguiente ejemplo muestra el resultado de establecer el espacio a su valor predeterminado, establecerlo a `15/16`, y finalmente a `1/2` de la nota principal.

```
<<
  \new Staff \relative {
    c''1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff \relative {
    c''1 \afterGrace 15/16 d1 { c16[ d] } c1
  }
  \new Staff \relative {
    c''1 \afterGrace 1/2 d1 { c16[ d] } c1
  }
>>
```



El efecto de `\afterGrace` también se puede obtener usando espaciadores. El ejemplo siguiente sitúa la nota de adorno después de un espacio que dura 7/8 de la nota principal.

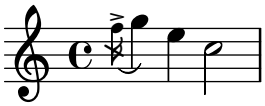
```
\new Voice \relative {
  <<
    { d''1^\trill_( }
    { s2 s4. \grace { c16 d } }
  >>
  c1)
}
```



Una expresión musical `\grace` introduce ajustes de tipografía especiales, por ejemplo para producir un tipo más pequeño y para fijar las direcciones. Por ello, cuando se introducen trucos

para la presentación, deben ir dentro de la expresión de adorno. Las sobreescrituras se deben también revertir dentro de la expresión de adorno. Aquí, la dirección predeterminada de la plica de la nota de adorno se sobreescribe y luego se revierte.

```
\new Voice \relative {
  \acciaccatura {
    \stemDown
    f''16->
    \stemNeutral
  }
  g4 e c2
}
```

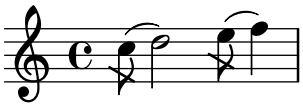


Fragmentos de código seleccionados

Utilizar la barra que tacha las notas de adorno con notas normales

Es posible aplicar la barrita que cruza la barra de las acciaccaturas, en otras situaciones.

```
\relative c'' {
  \override Flag.stroke-style = "grace"
  c8( d2) e8( f4)
}
```

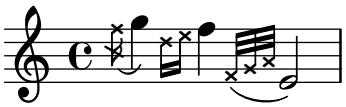


Trucar la disposición de las notas de adorno dentro de la música

La disposición de las expresiones de adorno se puede cambiar usando las funciones `add-grace-property` y `remove-grace-property`.

El ejemplo siguiente borra la definición de la propiedad `direction` (dirección) de los objetos gráficos `Stem` (plicas) para esta nota de adorno, de manera que las plicas no siemmpre apuntan hacia arriba, y cambia la forma predeterminada de las cabezas a aspas.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16 e } f4
       \appoggiatura { f,32 g a } e2
    }
  }
}
```



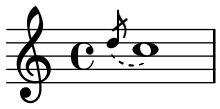
Redefinición de los valores globales predeterminados para notas de adorno

Los valores predeterminados para las notas de adorno están almacenados en los identificadores siguientes.

```
startGraceMusic
stopGraceMusic
startAcciaccaturaMusic
stopAcciaccaturaMusic
startAppoggiaturaMusic
stopAppoggiaturaMusic
```

Todos ellos están definidos en el archivo `ly/grace-init.ly`. Redefiniéndolos se pueden obtener otros efectos.

```
startAcciaccaturaMusic = {
  <>(
  \override Flag.stroke-style = "grace"
  \slurDashed
}

stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>)
}

\relative c'' {
  \acciaccatura d8 c1
}
```



Posicionamiento de las notas de adorno con espacio flotante

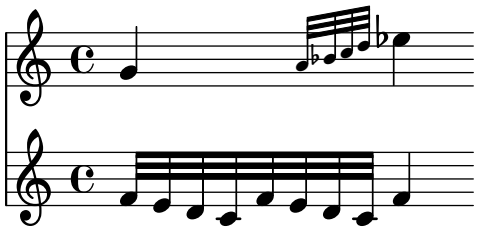
Al establecer la propiedad `strict-grace-spacing` hacemos que las columnas musicales para las notas de adorno sean 'flotantes', es decir, desacopladas de las notas que no son de adorno: primero se aplica el espaciado de las notas normales, y luego se ponen las columnas musicales de las notas de adorno a la izquierda de las columnas musicales de las notas principales.

Sin embargo, a causa del bug número 6876 (`https://gitlab.com/lilypond/lilypond/-/issues/6876`), las alteraciones accidentales se ignoran si esta propiedad está establecida. Este fragmento de código ofrece una solución para este problema.

Otro desafortunado efecto colateral de esta propiedad es que LilyPond no comprueba si hay espacio horizontal suficiente para las notas de adorno (este problema recibe seguimiento como bug número 2630 (`https://gitlab.com/lilypond/lilypond/-/issues/2630`)). Tenemos que asegurarnos de que ha espacio suficiente, por ejemplo, por medio del uso de `\newSpacingSection` junto con un valor adecuado para la duración `base-shortest-duration` del objeto gráfico `SpacingSpanner`.

```
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
   #{
     \override NoteHead.X-offset = #(- offset 0.85)
     \override Stem.X-offset = #offset
```

```
      \grace { $music }
      \revert NoteHead.X-offset
      \revert Stem.X-offset
   #})

\relative c'' <<
  { g4 \shiftedGrace #-1.3 a32 \shiftedGrace #-0.5 { bes c d } es4 }
  { f,32 e d c f e d c f4 }
>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.strict-grace-spacing = ##t
  }
}
```



Véase también

Glosario musical: Sección “grace notes” en *Glosario Musical*, Sección “acciaccatura” en *Glosario Musical*, Sección “appoggiatura” en *Glosario Musical*.

Referencia de la notación: Sección 2.1.3 [Escalar las duraciones], página 62, Sección 2.4.3 [Barras manuales], página 114.

Archivos de inicio: `ly/grace-init.ly`.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

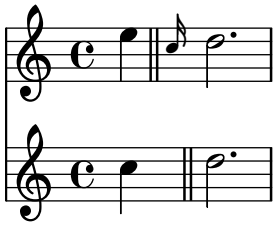
Referencia de funcionamiento interno: Sección “GraceMusic” en *Referencia de Funcionamiento Interno*, Sección “Grace_beam_engraver” en *Referencia de Funcionamiento Interno*, Sección “Grace_auto_beam_engraver” en *Referencia de Funcionamiento Interno*, Sección “Grace_engraver” en *Referencia de Funcionamiento Interno*, Sección “Grace_spacing_engraver” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Una *acciaccatura* de varias notas con una barra se imprime sin tachar, y tiene exactamente la misma apariencia que una *appoggiatura* de varias notas con barra.

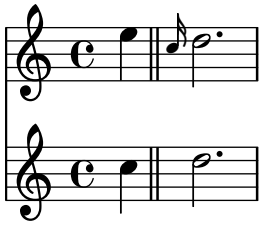
La sincronización de las notas de adorno también puede acarrear sorpresas. La notación de pentagramas, como armaduras, líneas divisorias, etc., se sincronizan también. Ponga cuidado cuando mezcle pentagramas con adornos y sin adornos, por ejemplo

```
<<
  \new Staff \relative { e''4 \section \grace c16 d2. }
  \new Staff \relative { c''4 \section d2. }
>>
```



Esto se puede remediar insertando desplazamientos de adorno de las duraciones correspondientes en los otros pentagramas. Para el ejemplo anterior

```
<<
  \new Staff \relative { e''4 \section \grace c16 d2. }
  \new Staff \relative { c''4 \section \grace s16 d2. }
>>
```



Es obligatorio usar la instrucción \grace para la parte de los desplazamientos, incluso si la parte visual usa \acciaccatura o \appoggiatura porque en caso contrario se imprime una fea ligadura que conecta a la nota de adorno invisible con la nota siguiente.

Las secciones de adorno solo se deben usar dentro de expresiones de música secuenciales. No están contemplados el anidado ni la yuxtaposición de secciones de adorno, y podría producir caídas u otros errores.

Cada nota de adorno en la salida MIDI tiene una longitud que es 1/4 de su duración real. Si la duración combinada de las notas de adorno es mayor que la longitud de la nota precedente, se genera un error "Retrocediendo en el tiempo MIDI". Tiene dos opciones: en primer lugar, puede hacer más corta la duración de las notas de adorno, por ejemplo:

```
c'8 \acciaccatura { c'8[ d' e' f' g'] }
```

se convierte en:

```
c'8 \acciaccatura { c'16[ d' e' f' g'] }
```

Otra opción es cambiar explícitamente la duración musical:

```
c'8 \acciaccatura { \scaleDurations 1/2 { c'8[ d' e' f' g'] } }
```

Véase Sección 2.1.3 [Escalar las duraciones], página 62.

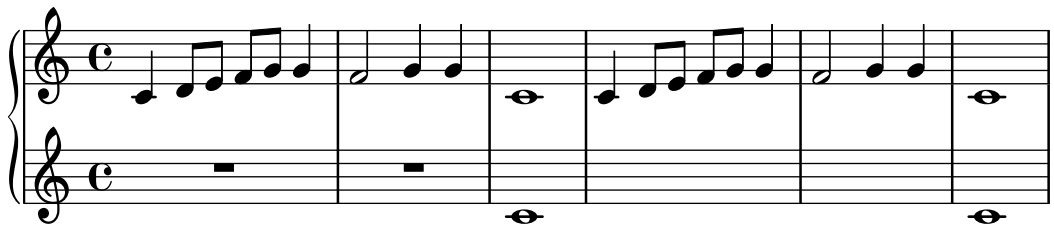
2.6.2 Alinear con una cadenza

En un contexto orquestal, las cadenzas presentan un problema especial: al construir una partitura que tiene una cadenza, todos los demás instrumentos deben saltar tantas notas como la longitud de la cadenza, pues en caso contrario empezarán demasiado pronto o demasiado tarde.

Una solución a este problema son las funciones mmrest-of-length y skip-of-length. Estas funciones de Scheme toman un fragmento de música como argumento y generan un \skip o silencio multicompás, de la longitud exacta del fragmento.

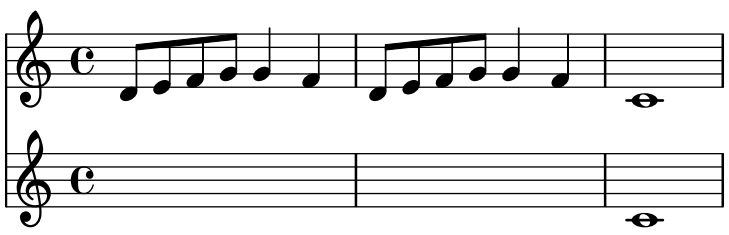
```
MyCadenza = \relative {
  c'4 d8 e f g g4
  f2 g4 g
}
```

```
\new GrandStaff <<
  \new Staff {
    \MyCadenza c'1
    \MyCadenza c'1
  }
  \new Staff {
    #(mmrest-of-length MyCadenza)
    c'1
    #(skip-of-length MyCadenza)
    c'1
  }
>>
```



Otra solución es usar la instrucción \skip.

```
MyCadenza = \fixed c' {
  \repeat volta 2 {
    d8 e f g g4 f4
  }
}

music = <<
  \new Staff {
    \MyCadenza
    c'1
  }
  \new Staff {
    \skip \MyCadenza
    c'1
  }
>>

\unfoldRepeats \music
```



Véase también

Glosario musical: Sección “cadenza” en *Glosario Musical*.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

2.6.3 Gestión del tiempo

El tiempo está administrado por el `Timing_translator`, que de forma predeterminada vive en el contexto de `Score`. Se añade un alias, `Timing`, al contexto en que se coloca el `Timing_translator`. Para asegurar que está disponible el alias `Timing`, quizá tenga que crear explícitamente una instancia del contexto contenedor (como `Voice` o `Staff`).

Se usan las siguientes propiedades de `Timing` para seguir la pista del tiempo dentro de la partitura.

`currentBarNumber`
: El número de compás en curso. Para ver un ejemplo que muestra el uso de esta propiedad, consulte Sección 2.5.3 [Numeración de los compases], página 132.

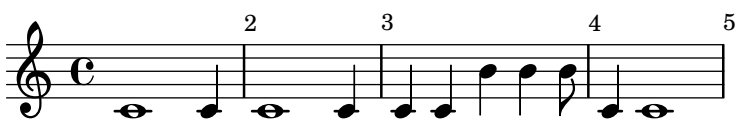
`measureLength`
: La longitud de los compases dentro de la indicación actual de compás. Para un 4/4 esto es 1, y para el 6/8 es 3/4. Su valor determina cuándo se insertan las líneas divisorias y cómo se generan las barras automáticas.

`measurePosition`
: Esta propiedad interna es el punto en curso dentro del compás actual. Cuando alcanza `measureLength`, se reinicia al valor cero y se incrementa `currentBarNumber`. `measurePosition` no se debe configurar explícitamente, pero se puede cambiar con la instrucción `\partial`.

`timing`
: Si tiene el valor `#t`, las variables anteriores se actualizan a cada paso de tiempo. Cuando tiene el valor `#f`, las variables anteriores conservan su valor actual indefinidamente.

La cuenta del tiempo se puede cambiar estableciendo el valor de cualquiera de estas variables. En el siguiente ejemplo, se imprime la indicación de compás predeterminada 4/4, pero `measureLength` está ajustado a 5/4. En los 4/8 hasta el tercer compás, la instrucción `\partial` avanza la posición del compás para dejar un resto de solo 5/8, acortando ese compás en 1/8. La siguiente línea divisoria cae en 9/8 en vez de hacerlo en 5/4.

```
\fixed c' {
  \override Score.BarNumber.break-visibility = #all-visible
  \set Timing.measureLength = #5/4
  c1 c4
  c1 c4
  c4 c \partial 8*5 b4 b b8
  c4 c1
}
```



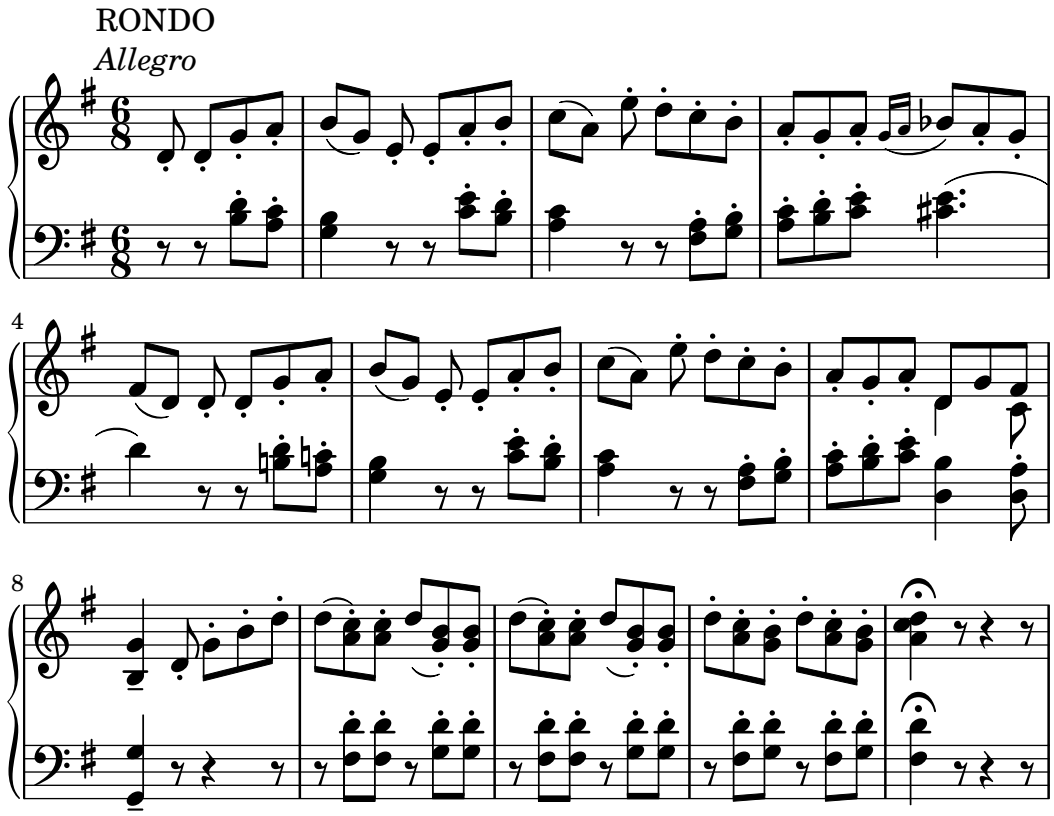
Véase también

Referencia de la notación: Sección 2.5.3 [Numeración de los compases], página 132, Sección 2.3.4 [Música sin compasear], página 91.

Fragmentos de código: Sección “Duraciones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Timing_translator” en *Referencia de Funcionamiento Interno*, Sección “Score” en *Referencia de Funcionamiento Interno*.

3 Expresiones



Esta sección relaciona diversas marcas de expresión que se pueden crear en un a partitura.

3.1 Expresiones adosadas a las notas

Esta sección explica cómo crear marcas expresivas que están aplicadas a notas: articulaciones, ornamentos y matices. También se tratan los métodos para crear las nuevas marcas dinámicas.

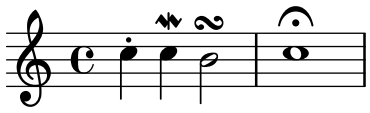
3.1.1 Articulaciones y ornamentos

Se pueden adjuntar a las notas un amplio abanico de símbolos para denotar articulaciones, adornos y otras indicaciones de ejecución, utilizando la sintaxis siguiente:

*nota**nombre*

La lista de los posibles valores de *nombre* está en Sección B.13 [Lista de articulaciones], página 934. Por ejemplo:

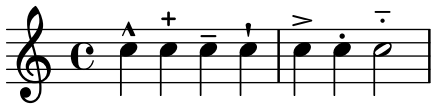
```
\relative {
  c''4\staccato c\mordent b2\turn
  c1\fermata
}
```



Algunas de estas articulaciones tienen abreviaturas que facilitan su escritura. Las abreviaturas se escriben detrás del nombre de la nota, y su sintaxis consiste en un guión - seguido de un símbolo que especifica la articulación. Existen abreviaturas predefinidas para el *marcato*, *stopped*

(nota apagada), *tenuto*, *staccatissimo*, *accent* (acento), *staccato* (picado), y *portato*. La salida correspondiente a estas articulaciones aparece de la siguiente manera:

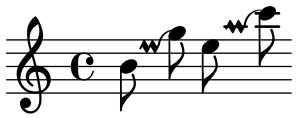
```
\relative {
  c''4-^ c-+ c-- c-!
  c4-> c-. c2-_
}
```



Las reglas para la colocación predeterminada de las articulaciones se encuentran definidas en el archivo `scm/script.scm`. Las articulaciones y ornamentos se pueden colocar manualmente por encima o por debajo del pentagrama; véase Sección 36.1 [Dirección y posición], página 783.

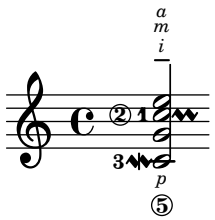
El `bachschleifer` se posiciona a la izquerda del objeto `NoteHead`. Este ornamento puede tener líneas adicionales.

```
{
  \autoBeamOff
  b'8 g''\bachschleifer
  e'' c'''\bachschleifer
}
```



También es posible posicionar articulaciones y ornamentos comunes a la izquierda y a la derecha de una cabeza de nota sobreescribiendo la propiedad `side-axis` y, si es necesario, la propiedad `direction`. Las funciones de conveniencia `\atLeft` o `\atRight` se ocupan de esto.

```
{
  \set fingeringOrientations = #'(left)
  \set stringNumberOrientations = #'(left down)
  <
   c'-3\5_\rightHandFinger #1 \atLeft \mordent
   g'^\rightHandFinger #2
   c''-1\2^\rightHandFinger #3 \atRight \prall
   e'' ^\rightHandFinger #4
  >2^\tenuto
}
```



Instrucciones predefinidas

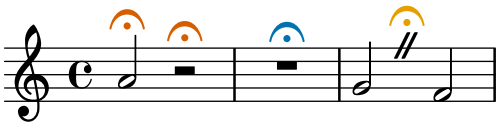
`\atLeft`, `\atRight`.

El tipo de grob que crea una articulación depende de a qué se adjunta.

- Obre las notas o silencios ordinarios, las articulaciones crean objetos `Script`.

- Sobre los silencios multicompás, las articulaciones crean objetos `MultiMeasureRestScript`.
- Sobre una `\caesura`, las articulaciones crean objetos `CaesuraScript`.

```
\override Score.Script.color = #(universal-color 'vermillion)
\override Score.MultiMeasureRestScript.color = #(universal-color 'blue)
\override Score.CaesuraScript.color = #(universal-color 'orange)
a'2\fermata r\fermata
R1\fermata
g'2 \caesura \fermata f'2
```



Además de articulaciones, se pueden adjuntar textos y elementos de marcado a las notas. Véase Sección 8.1.2 [Inscripciones textuales], página 314.

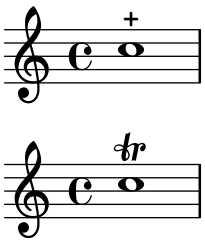
Para ver más información acerca de la ordenación de los elementos `Script` y `TextScript` que se adjuntan a las notas, consulte Sección "Colocación de los objetos" en *Manual de Aprendizaje*.

Fragmentos de código seleccionados

Modificar los valores predeterminados para la notación abreviada de las articulaciones

Las abreviaturas se encuentran definidas dentro del archivo `ly/script-init.ly`, donde las variables `dashHat`, `dashPlus`, `dashDash`, `dashBang`, `dashLarger`, `dashDot` y `dashUnderscore` reciben valores predeterminados. Se pueden modificar estos valores predeterminados para las abreviaturas. Por ejemplo, para asociar la abreviatura `-+` (`dashPlus`) con el símbolo del *semitrino* en lugar del símbolo predeterminado "+", asigne el valor `\trill` a la variable `dashPlus`:

```
\relative c'' { c1-+ }

dashPlus = \trill

\relative c'' { c1-+ }
```



Controlar la ordenación vertical de las inscripciones

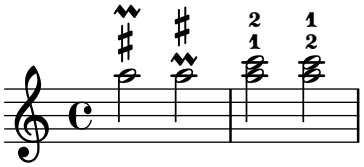
El orden vertical que ocupan las inscripciones gráficas está controlado con la propiedad `script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo, el `TextScript` (el símbolo del *sostenido*) tiene primero la prioridad más baja, por lo que se sitúa en la posición más baja en el primer ejemplo. En el segundo, el `Script` (el símbolo del *semitrino*) es el que la tiene más baja, por lo que se sitúa en la parte interior. Cuando dos objetos tienen la misma prioridad, el orden en que se introducen determina cuál será el que aparece en primer lugar.

Observe que para los objetos gráficos `Fingering`, `StringNumber` y `StrokeFinger`, si se usan dentro de un acorde, el orden vertical también viene determinado por la posición vertical de la

cabeza de la nota asociada, que se añade a (o, dependiendo de la dirección, se sustrae de) el valor `script-priority` del objeto gráfico. Esto nos asegura que para las digitaciones que están por encima de un acorde, la nota inferior está asociada con la digitación inferior (y viceversa para la otra dirección); no importa si escribimos las notas del acorde de arriba a abajo o de abajo a arriba.

De manera predeterminada, las inscripciones de carácter menos técnico se sitúan más cerca de la cabeza de la nota; el orden es más o menos: articulación, armónico, digitación, digitación de la mano derecha, número de cuerda, calderón, indicaciones de arco, e inscripción de texto.

```
\relative c''' {
  \once \override TextScript.script-priority = -100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = -100
  a2^\prall^\markup { \sharp }

  \set fingeringOrientations = #'(up)
  <c-2 a-1>2
  <a-1 c\tweak script-priority -100 -2>2
}
```

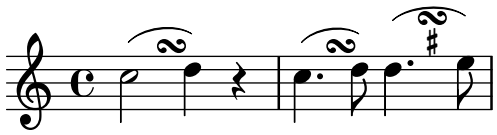


Véase Sección B.18 [Valores por omisión para `script-priority`], página 944.

Crear un grupeto de anticipación

La creación de un grupeto circular de anticipación entre dos notas, donde la nota inferior del grupeto utiliza una alteración, requiere varias sobreescrituras de propiedades. La propiedad `outside-staff-priority` se debe establecer al valor `#f`, pues en caso contrario tendría prioridad sobre la propiedad `avoid-slur`. Cambiando el primer argumento de `\after` (que es una duración) puede ajustarse la posición horizontal.

```
\relative c'' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
  d4.( e8)
}
```



Véase también

Glosario musical: Sección “tenuto” en *Glosario Musical*, Sección “accent” en *Glosario Musical*, Sección “staccato” en *Glosario Musical*, Sección “portato” en *Glosario Musical*.

Manual de aprendizaje: Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 8.1.2 [Inscripciones textuales], página 314, Sección 36.1 [Dirección y posición], página 783, Sección B.13 [Lista de articulaciones], página 934, Sección 3.3.3 [Trinos], página 187.

Archivos instalados: `scm/script.scm`.

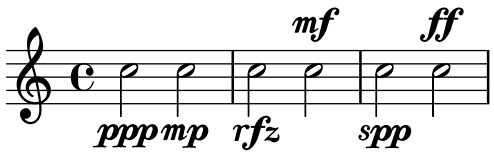
Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CaesuraScript” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestScript” en *Referencia de Funcionamiento Interno*, Sección “Script” en *Referencia de Funcionamiento Interno*, Sección “TextScript” en *Referencia de Funcionamiento Interno*.

3.1.2 Matices dinámicos

Las marcas dinámicas de matiz absoluto se especifican usando una instrucción después de una nota: `c4\ff`. Las marcas dinámicas disponibles son `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz`, `\rfz` y `\n`. Las indicaciones dinámicas se pueden colocar manualmente por encima o por debajo del pentagrama; para ver más detalles, consulte Sección 36.1 [Dirección y posición], página 783.

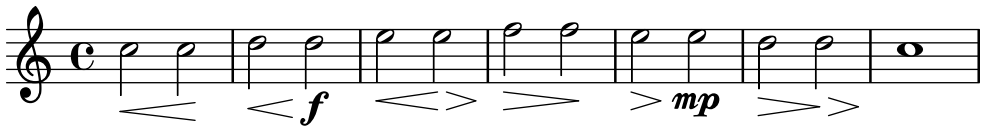
```
\relative c'' {
  c2\ppp c\mp
  c2\rfz c^\mf
  c2_\spp c^\ff
}
```



Una indicación de *crescendo* se comienza con `\<` y se termina con `\!`, con un matiz absoluto o con otra indicación de crescendo o de decrescendo. Una indicación de *decrescendo* comienza con `\>` y se termina también con `\!`, con un matiz dinámico absoluto o con otra indicación de crescendo o de decrescendo. Se pueden usar `\cr` y `\decr` en lugar de `\<` y `\>`. También es posible usar `\endcr` y `\enddecr` en lugar de `\!` para finalizar un crescendo o un decrescendo, respectivamente. De forma predeterminada, se tipografían reguladores en ángulo cuando se utiliza esta notación.

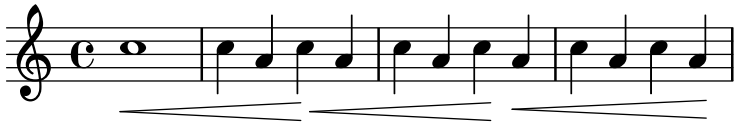
```
\relative c'' {
  c2\< c\!
  d2\< d\f
  e2\< e\>
  f2\> f\!
  e2\> e\mp
  d2\> d\>
  c1\!
```

```
}
```



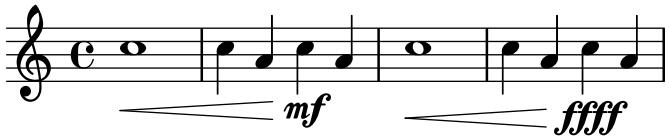
Un regulador terminado mediante \! finaliza en el borde derecho de la nota que lleva el \! adosado. En el caso en que esté terminado con el comienzo de otra indicación *crescendo* o *decrescendo*, terminará en el centro de la nota que tiene adosada la siguiente indicación \< ó \>. El regulador siguiente empezará entonces en el borde derecho de la misma nota en lugar del borde izquierdo, como sería usual si hubiese terminado con \! previamente. Si un regulador termina en el inicio de un compás, finalizará sobre la línea divisoria precedente.

```
\relative {
  c''1\< | c4 a c\< a | c4 a c\! a\< | c4 a c a\!
}
```



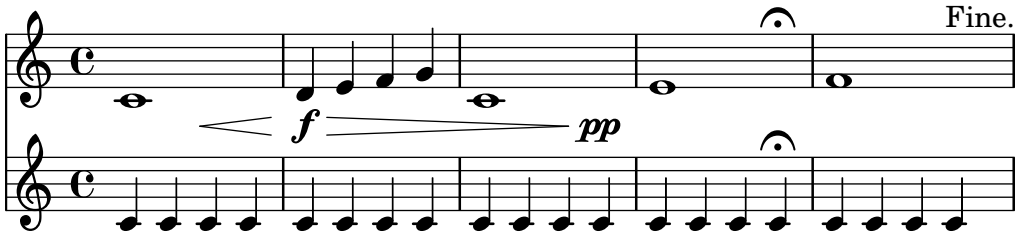
Los reguladores que terminan con indicaciones dinámicas absolutas en lugar de con \! también se tipografían de manera parecida. Sin embargo, la longitud de la propia indicación dinámica puede alterar el punto en que finaliza el regulador anterior.

```
\relative {
  c''1\< | c4 a c\mf a | c1\< | c4 a c\ffff a
}
```



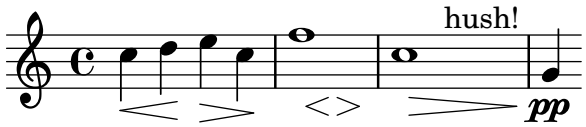
Con frecuencia, las marcas como *crescendo* o *decrescendo* deben comenzar o terminar en el mismo instante temporal dentro de la duración de una nota sostenida. Esto se puede conseguir con \after, que se puede usar también para crear articulaciones o inscripciones de texto retrasadas:

```
<<
  \relative {
    \after 2 \< c'1
    d4\f\> e f g
    \after 2. \pp c,1
    \after 2. \fermata e
    \after 2. ^"Fine." f
  }
  \relative {
    \repeat unfold 12 c'4
    c c c c\fermata
    c c c c
  }
>>
```



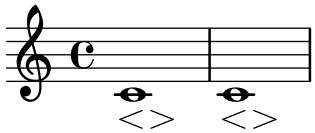
Se pueden usar varias instancias de \after para tipografiar varias indicaciones dinámicas sobre una sola nota. Esto es útil especialmente para añadir un crescendo y un decrescendo a la misma nota:

```
\relative {
  c''4\< d\! e\> c\!
  \after 4 \< \after 2\> \after 2. \! f1
  \textLengthOn
  \after 4 \> \after 2. ^"hush!" c1
  g4\pp
}
```



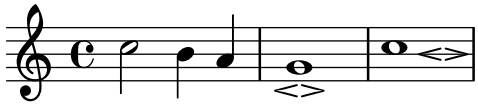
Si la primera de dicha secuencia de marcas sobre una única nota se supone que debe coincidir con el ataque de la nota, es conveniente adjuntarla a un acorde vacío <>. De esta manera, todas las marcas se pueden introducir en su orden visual natural:

```
{
  <>\< \after 4 \> \after 2 \! c'1
  % easier to write and read than:
  \after 4 \> \after 2 \! c'1\<
}
```



También se puede usar la instrucción \espressivo para indicar un crescendo y un decrescendo sobre la misma nota. Sin embargo, advierta que esta indicación está implementada como una articulación, no como una expresión dinámica.

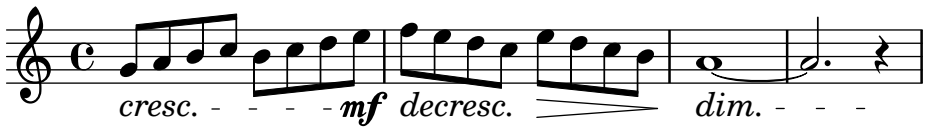
```
\relative {
  c''2 b4 a
  g1\espressivo
  \after 2. \espressivo c
}
```



Las indicaciones de crescendo textuales se inician con \cresc. Los decrescendos textuales se inician con \decresc o con \dim. Se trazan líneas extensoras cuando es necesario.

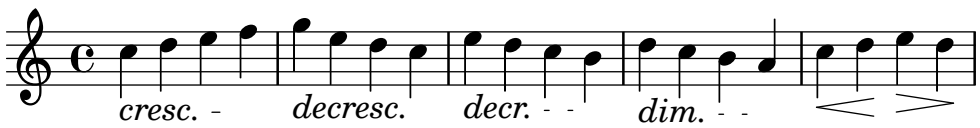
```
\relative {
```

```
  g'8\cresc a b c b c d e\mf |
  f8\decresc e d c e\> d c b |
  a1\dim ~ |
  a2. r4\! |
}
```



También pueden sustituirse los reguladores gráficos por cambios de dinámica textuales:

```
\relative c'' {
  \crescTextCresc
  c4\< d e f\! |
  \dimTextDecresc
  g4\> e d c\! |
  \dimTextDecr
  e4\> d c b\! |
  \dimTextDim
  d4\> c b a\! |
  \crescHairpin
  \dimHairpin
  c4\< d\! e\> d\! |
}
```



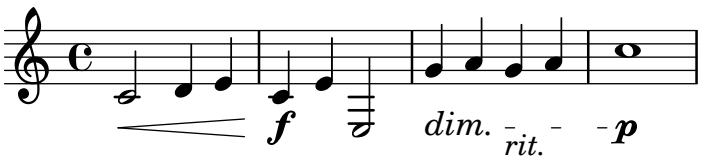
Para crear nuevas indicaciones de matiz absoluto o de texto que deba alinearse con los matices; véase Sección 3.1.3 [Indicaciones dinámicas nuevas], página 169.

La colocación vertical de las indicaciones de dinámica se maneja por parte de Sección "DynamicLineSpanner" en *Referencia de Funcionamiento Interno*.

Está disponible un contexto `Dynamics` para tipografiar las indicaciones de matiz dinámico en su propia línea horizontal. Utilice silencios de separación para indicar los tiempos (las notas que están dentro de un contexto `Dynamics` también ocupan tiempo musical, pero no se imprimen). El contexto `Dynamics` puede contener otros elementos como inscripciones textuales, objetos extensores de texto e indicaciones de pedal de piano.

```
<<
  \new Staff \relative {
    c'2 d4 e |
    c4 e e,2 |
    g'4 a g a |
    c1 |
  }
  \new Dynamics {
    s1\< |
    s1\f |
    s2\dim s2-"rit." |
    s1\p |
  }
```

```
>>
```



Nota: Incluso si solo hay una única indicación dinámica absoluta como `\p` en toda la partitura, LilyPond siempre crea *dos* objetos para ella, un objeto `DynamicText` y un objeto `DynamicLineSpanner`, y las propiedades para controlar la indicación dinámica se comparten entre los dos objetos. Por ejepmlo, el tamaño se puede cambiar con la propiedad `DynamicText.font-size`, mientras que la posición vertical se controla por medio de `DynamicLineSpanner.Y-offset`.

Como consecuencia, un código como

```
\tweak font-size 5 \p
```

funciona, pero

```
\tweak Y-offset 5 \p
```

no funciona. Tenemos de escribir

```
\tweak DynamicLineSpanner.Y-offset 5 \p
```

en su lugar.

Instrucciones predefinidas

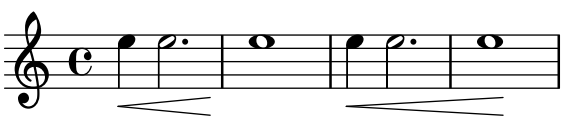
`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`, `\crescTextCresc`, `\dimTextDim`, `\dimTextDecr`, `\dimTextDecresc`, `\crescHairpin`, `\dimHairpin`.

Fragmentos de código seleccionados

Establecer el comportamiento de los reguladores en las barras de compás

Si la nota que da fin a un regulador cae sobre la primera parte de un compás, el regulador se detiene en la línea divisoria inmediatamente precedente. Se puede controlar este comportamiento sobreescribiendo la propiedad `to-barline`.

```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```

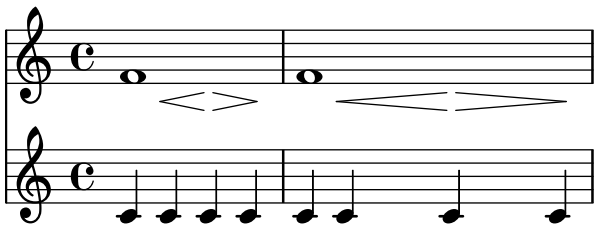


Ajustar la longitud mínima de los reguladores

Si los reguladores son demasiado cortos, se pueden alargar modificando la propiedad `minimum-length` del objeto `Hairpin`.

```
<<
```

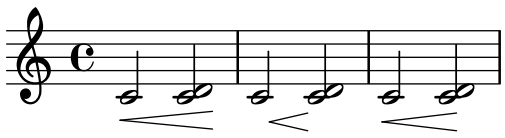
```
  {
    \after 4 \< \after 2 \> \after 2. \! f'1
    \override Hairpin.minimum-length = 8
    \after 4 \< \after 2 \> \after 2. \! f'1
  }
  {
    \repeat unfold 8 c'4
  }
>>
```



Alinear los extremos de los reguladores con las direcciones de `NoteColumn`

Se pueden alinear los extremos de los reguladores con los puntos izquierdo, central o derecho de los objetos gráficos `NoteColumn` por medio de la sobreescritura de la propiedad `endpoint-alignments`, que es una pareja de números que representan los extremos izquierdo y derecho de un regulador. Se espera que los `endpoint-alignments` sean direcciones (-1, 0 o 1). Cualquier otro valor se transforma acompañado de una advertencia. La derecha de un regulador que termina en un silencio no resulta afectada, terminando siempre en el borde izquierdo del silencio.

```
{
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #'(1 . -1)
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #`(,LEFT . ,CENTER)
  c'2\< <c' d'>\! |
}
```

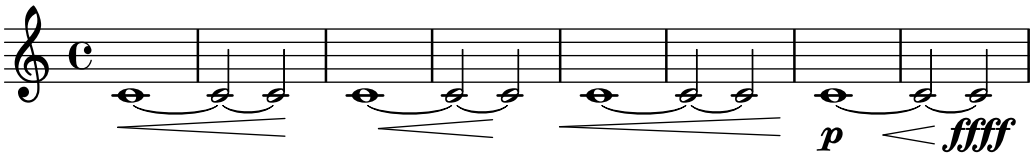


Desplazar los extremos de un regulador

Se pueden desplazar los extremos de los reguladores fijando la propiedad `shorten-pair` del objeto `Hairpin`. Los valores positivos mueven los extremos hacia la derecha, los negativos hacia la izquierda. A diferencia de la propiedad `minimum-length`, esta propiedad afecta solo a la apariencia del regulador; no ajusta el espaciado horizontal (incluida la posición de los matices circundantes). Este método es así adecuado para el ajuste fino de un regulador dentro del espacio de que dispone.

```
{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
```

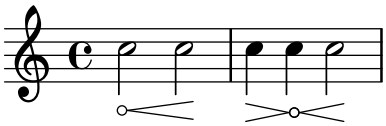
```
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}
```



Impresión de reguladores utilizando la notación «al niente»

Se pueden imprimir reguladores con un círculo en la punta (notación «al niente») estableciendo la propiedad `circled-tip` del objeto `Hairpin` al valor `#t`.

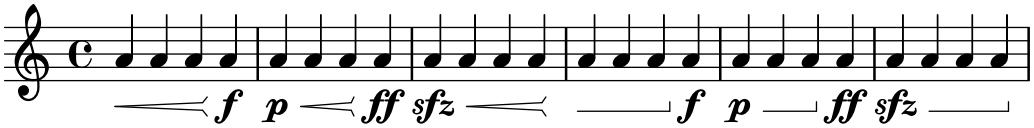
```
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```

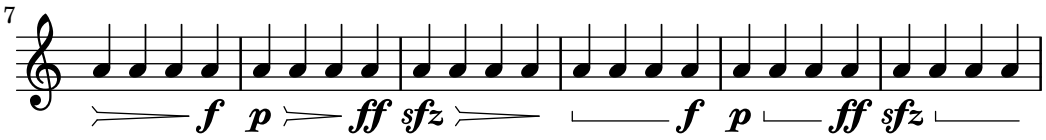


Imprimir reguladores en diversos estilos

Los reguladores se pueden crear en una amplia variedad de estilos.

```
\relative c'' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
}
```



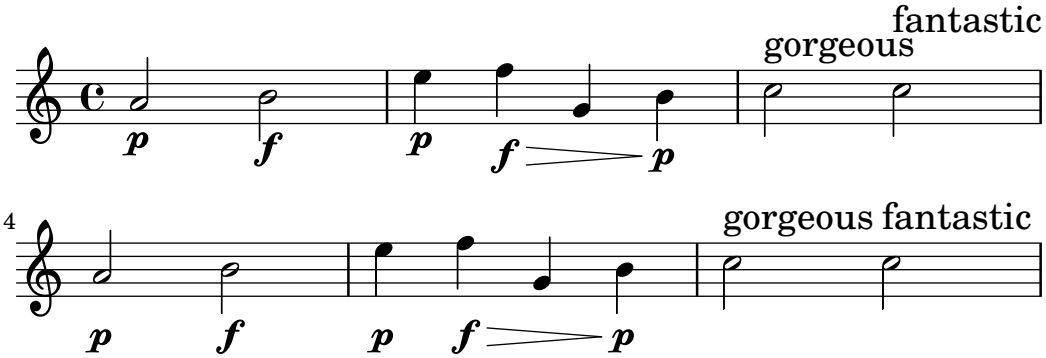


Indicaciones dinámicas y textuales alineadas verticalmente

Para todos los objetos DynamicLineSpanner (esto es, reguladores y textos de dinámica), la distancia vertical mínima entre su línea de referencia y el pentagrama viene dada por el valor dentro de la propiedad staff-padding, a no ser que otros elementos de notación los fuercen a situarse en una posición más alejada. El establecimiento de esta propiedad a un valor suficientemente grande, permite alinear las indicaciones de matiz dinámico.

Se usa una idea similar junto a \textLengthOn para alinear las inscripciones de texto a lo largo de su línea de base.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
  \textLengthOn
  \override TextScript.staff-padding = 1
  \music
}
```



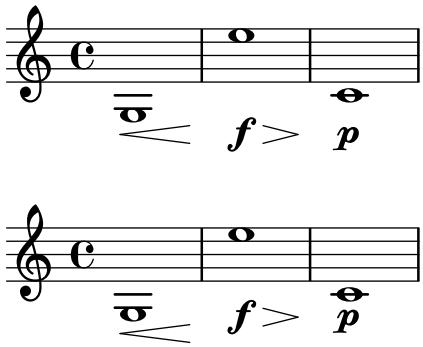
Desalinear indicaciones dinámicas y textuales verticalmente

De forma predeterminada, LilyPond usa ojetos gráficos DynamicLineSpanner para alinear verticalmente objetos de matiz dinámico sucesivos como reguladores y dinamicas textuales. Sin embargo esto no siempre es deseable. Insertando \breakDynamicSpan, que finaliza el objeto extenso de alineamiento de forma prematura, se puede evitar este alineamiento vertical.

Véase también el fragmento de código “Desalinear indicaciones dinámicas y textuales horizontalmente”.

```
{ g1\< |
  e''\f\> |
  c'\p }

{ g1\< |
```

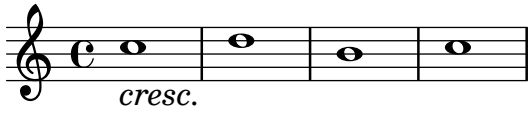
```
  e''\breakDynamicSpan\f\> |
  c'\p }
```



Ocultar la línea de extensión de las expresiones textuales de dinámica

Los cambios de dinámica con estilo de texto (como *cresc.* y *dim.*) se imprimen con una línea intermitente que muestra su alcance. Esta línea se puede suprimir de la siguiente manera:

```
\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}
```

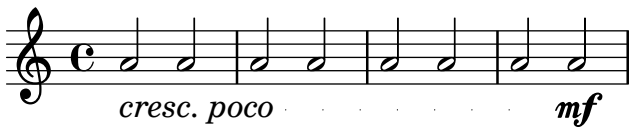


Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales

Se puede modificar el texto empleado para los *crescendos* y *decrescendos* modificando las propiedades de contexto `crescendoText` y `decrescendoText`.

El estilo de la línea de extensión se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `dashed-line` (línea discontinua), y entre otros valores posibles se encuentran `line` (línea), `dotted-line` (línea de puntos) y `none` (nada):

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Véase también

Glosario musical: Sección “al niente” en *Glosario Musical*, Sección “crescendo” en *Glosario Musical*, Sección “decrescendo” en *Glosario Musical*, Sección “hairpin” en *Glosario Musical*.

Manual de aprendizaje: Sección “Articulaciones y matices dinámicos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783, Sección 3.1.3 [Indicaciones dinámicas nuevas], página 169, Sección 24.9 [Enriquecimiento de la salida MIDI], página 666, Sección 24.4 [Control de las dinámicas del MIDI], página 657.

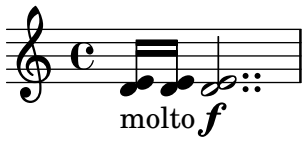
Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “DynamicText” en *Referencia de Funcionamiento Interno*, Sección “Hairpin” en *Referencia de Funcionamiento Interno*, Sección “DynamicLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “Dynamics” en *Referencia de Funcionamiento Interno*.

3.1.3 Indicaciones dinámicas nuevas

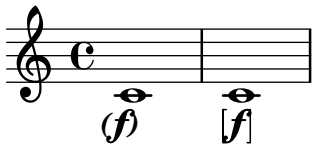
La manera más fácil de crear indicaciones dinámicas es usar objetos de marcado (`\markup`).

```
moltoF = \markup { molto \dynamic f }

\relative {
  <d' e>16_\moltoF <d e>
  <d e>2..
}
```



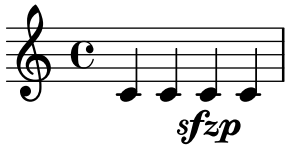
En el modo de marcado se pueden crear indicaciones dinámicas editoriales (entre paréntesis o corchetes). La sintaxis del modo de marcado se describe en Sección 8.2 [Formatear el texto], página 323.

```
roundF = \markup {
    \center-align \concat { \bold { \italic ( }
           \dynamic f \bold { \italic ) } } }
boxF = \markup { \bracket { \dynamic f } }
\relative {
  c'1_\roundF
  c1_\boxF
}
```



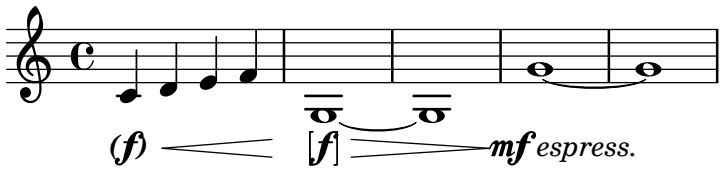
Las indicaciones dinámicas sencillas y centradas se crean fácilmente con la función `make-dynamic-script`.

```
sfzp = #(make-dynamic-script "sfzp")
\relative {
  c'4 c c\sfzp c
}
```



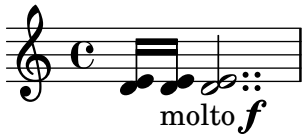
En general `make-dynamic-script` toma cualquier objeto de marcado como argumento. La fuente tipográfica de matices solo contiene los caracteres `f`, `m`, `p`, `r`, `s`, `z` y `n`; por lo que si se desea obtener una indicación dinámica que incluya texto normal o signos de puntuación, es necesario utilizar instrucciones de marcado que devuelvan los ajustes de la familia de fuente tipográfica y su codificación a las del texto normal, por ejemplo `\normal-text`. El uso de `make-dynamic-script` en lugar de un elemento de marcado corriente asegura la alineación vertical de los objetos de marcado y reguladores que se aplican a la misma cabeza de nota.

```
roundF = \markup { \center-align \concat {
           \normal-text { \bold { \italic ( } }
           \dynamic f
           \normal-text { \bold { \italic ) } } } }
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
              \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
\relative {
  c'4_\roundFdynamic\< d e f
  g,1~_\boxFdynamic\>
  g1
  g'1~\mfEspressDynamic
  g1
}
```



Se puede utilizar en su lugar la forma Scheme del modo de marcado. Su sintaxis se explica en Sección “Construcción de elementos de marcado en Scheme” en *Extender*.

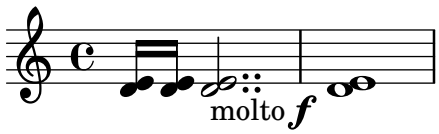
```
moltoF = #(make-dynamic-script
            (markup #:normal-text "molto"
                    #:dynamic "f"))
\relative {
  <d' e>16 <d e>
  <d e>2..\moltoF
}
```



Para alinear el texto del matiz dinámico a la izquierda en lugar de centrarlo sobre una nota, utilice un `\tweak`:

```
moltoF = \tweak DynamicText.self-alignment-X #LEFT
```

```
        #(make-dynamic-script
           (markup #:normal-text "molto"
                   #:dynamic "f"))
\relative {
  <d' e>16 <d e>
  <d e>2..\moltoF <d e>1
}
```



Los ajustes para las fuentes tipográficas en el modo de marcado se describen en Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327.

Véase también

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323, Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327, Sección 24.9 [Enriquecimiento de la salida MIDI], página 666, Sección 24.4 [Control de las dinámicas del MIDI], página 657.

Extender LilyPond: Sección “Construcción de elementos de marcado en Scheme” en *Extender*.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

3.2 Expresiones como curvas

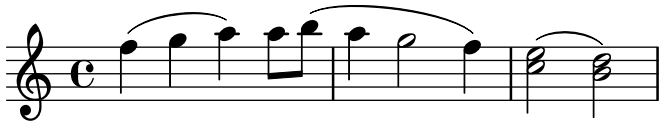
Esta sección explica cómo crear varias marcas expresivas de forma curva: ligaduras de expresión y de fraseo, respiraciones, caídas y elevaciones de tono.

3.2.1 Ligaduras de expresión

Las *Ligaduras de expresión* se introducen utilizando paréntesis:

> **Nota:** En música polifónica, las ligaduras de expresión deben terminar en la misma voz en la que empiezan.

```
\relative {
  f''4( g a) a8 b(
  a4 g2 f4)
  <c e>2( <b d>2)
}
```



Las ligaduras de expresión se pueden colocar manualmente por encima o por debajo de las notas; véase Sección 36.1 [Dirección y posición], página 783.

Debe tenerse cuidado con las ligaduras de expresión simultáneas o superpuestas. En casi todos los casos, las ligaduras de expresión externas indican, ciertamente, fraseo, y las ligaduras de fraseo pueden superponerse a las ligaduras de expresión normales, véase Sección 3.2.2 [Ligaduras de fraseo], página 175. Cuando se necesitan varias ligaduras de expresión normales dentro de una sola

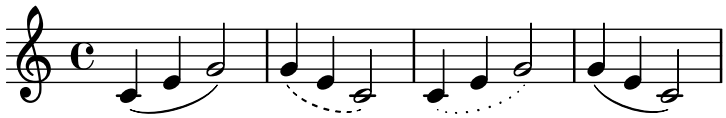
voz, los comienzos y finales correspondientes de las ligaduras deben etiquetarse precediéndolos por \= seguido de una clave identificativa (un símbolo o un entero no negativo).

```
\fixed c' {
  <c~ f\=1( g\=2( >2 <c e\=1) a\=2) >
}
```



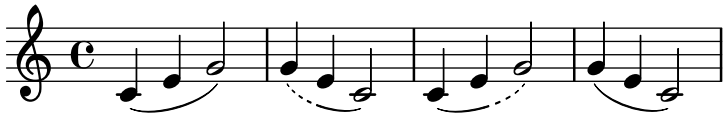
Las ligaduras de expresión pueden ser continuas, discontinuas o de puntos. El estilo predeterminado de las ligaduras de expresión es el continuo:

```
\relative {
  c'4( e g2)
  \slurDashed
  g4( e c2)
  \slurDotted
  c4( e g2)
  \slurSolid
  g4( e c2)
}
```



También se puede hacer que las ligaduras de expresión sean semi-discontinuas (la primera mitad discontinua y la segunda mitad continua) o semi-continuas (la primera mitad continua y la segunda discontinua):

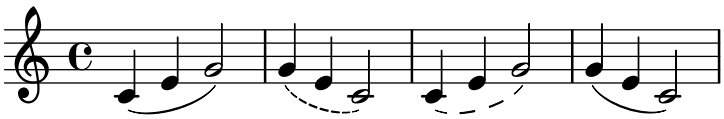
```
\relative {
  c'4( e g2)
  \slurHalfDashed
  g4( e c2)
  \slurHalfSolid
  c4( e g2)
  \slurSolid
  g4( e c2)
}
```



Se pueden definir patrones de discontinuidad para las ligaduras de expresión:

```
\relative {
  c'4( e g2)
  \slurDashPattern 0.7 0.75
  g4( e c2)
  \slurDashPattern 0.5 2.0
  c4( e g2)
  \slurSolid
  g4( e c2)
```

```
}
```



Instrucciones predefinidas

\slurUp, \slurDown, \slurNeutral, \slurDashed, \slurDotted, \slurHalfDashed, \slurHalfSolid, \slurDashPattern, \slurSolid.

Fragmentos de código seleccionados

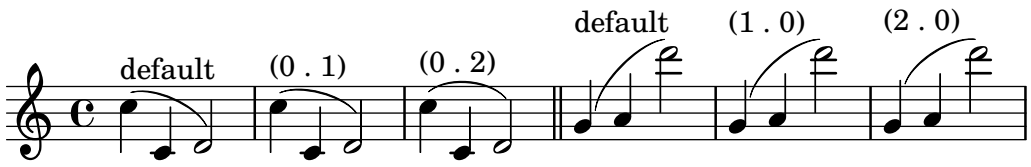
Ajustar las posiciones de las ligaduras verticalmente

Usando \override Slur.positions es posible fijar la posición vertical de los puntos de inicio y de fin de una ligadura de expresión a valores absolutos (o mejor dicho, forzar al algoritmo de ligaduras de LilyPond para que tenga en cuenta estos valores según se desee). En muchos casos, esto implica bastante ensayo y error hasta que se encuentran unos valores aceptables. Porblablemente habrá probado la instrucción \offset solo para comprobar que no funciona con las ligaduras de expresión, sino que emite una advertencia en su lugar.

El código de este fragmento nos permite trucar las posiciones de comienzo y final especificando cambios *relativos*, de manera parecida a \offset.

La sintaxis es: \offsetPositions #'(*dy1 . dy2*)

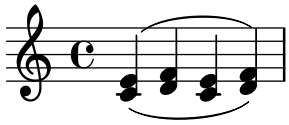
```
offsetPositions =
#(define-music-function (offsets) (number-pair?)
  #{
     \once \override Slur.control-points =
       #(lambda (grob)
          (match-let ((((_ . y1) _ _ (_ . y2))
                       (ly:slur::calc-control-points grob))
                      ((off1 . off2) offsets))
            (set! (ly:grob-property grob 'positions)
                  (cons (+ y1 off1) (+ y2 off2)))
            (ly:slur::calc-control-points grob)))
  #})

\relative c'' {
  c4(^"default" c, d2)
  \offsetPositions #'(0 . 1)
  c'4(^"(0 . 1)" c, d2)
  \offsetPositions #'(0 . 2)
  c'4(^"(0 . 2)" c, d2)
  \bar "||"
  g4(^"default" a d'2)
  \offsetPositions #'(1 . 0)
  g,,4(^"(1 . 0)" a d'2)
  \offsetPositions #'(2 . 0)
  g,,4(^"(2 . 0)" a d'2)
}
```



Utilizar ligaduras dobles para acordes legato

Algunos compositores escriben dos *ligaduras* cuando quieren acordes legato. Esto se puede conseguir estableciendo la propiedad de contexto `doubleSlurs`.

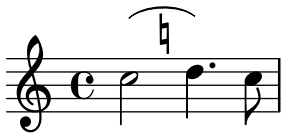
```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```



Situar los elementos de marcado de texto por dentro de las ligaduras

Los elementos de marcado de texto deben tener la propiedad `outside-staff-priority` establecida a `#f` para que se impriman por dentro de las ligaduras de expresión.

```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



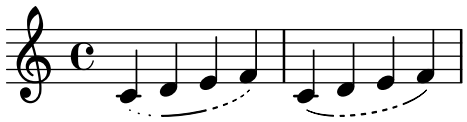
Hacer ligaduras de expresión con estructura compleja de discontinuidad

Las ligaduras de expresión se pueden construir con patrones de discontinuidad complejos mediante la definición de la propiedad `dash-definition`, que es una lista de segmentos de la ligadura, que a su vez son listas de parámetros que definen el comportamiento de discontinuidad del segmento dado.

Los segmentos de la ligadura se definen en términos del parámetro de bezier *t*, cuyo rango va de 0 a la izquierda de la ligadura hasta 1 en el extremo derecho de la ligadura. Cada segmento tiene la forma `(`*`t-inicio t-final fracción-discontinuidad período-discontinuidad`*`)`. En el segmento que va desde *t-inicio* hasta *t-final*, el patrón se define por los valores de *fracción-discontinuidad* y *período-discontinuidad*. La *fracción-discontinuidad* especifica el período en que el trazo es continuo y de color negro. Si se establece al valor de 1 se obtiene un segmento de ligadura continuo. Las unidades de *período-discontinuidad* son espacios de pentagrama.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'((  0  0.3  0.1 0.75)
                              (0.3  0.6  1   1   )
                              (0.65 1.0  0.4 0.75))
  c4( d e f)
```

```
  \once \override
    Slur.dash-definition = #'((0    0.25  1   1   )
                              (0.3  0.7   0.4 0.75)
                              (0.75 1.0   1   1   ))
  c4( d e f)
}
```



Véase también

Glosario musical: Sección "slur" en *Glosario Musical*.

Manual de aprendizaje: Sección "Acerca de la no anidabilidad de llaves y ligaduras" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783, Sección 3.2.2 [Ligaduras de fraseo], página 175.

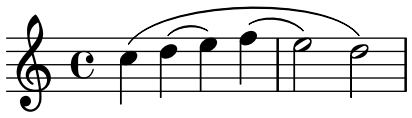
Fragmentos de código: Sección "Marcas de expresión" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Slur" en *Referencia de Funcionamiento Interno*.

3.2.2 Ligaduras de fraseo

Las *ligaduras de fraseo* (o marcas de fraseo) que indican una frase musical se escriben usando las instrucciones `\(` y `\)` respectivamente:

```
\relative {
  c''4\( d( e) f(
  e2) d\)
}
```

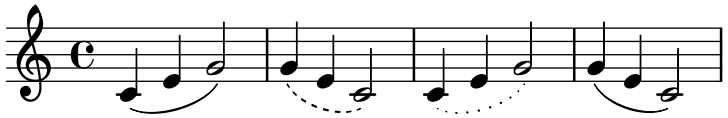


Tipográficamente, la ligadura de fraseo se comporta casi exactamente igual que una ligadura de expresión normal. Sin embargo, se tratan como objetos diferentes. Una `\slurUp` no tendrá ningún efecto sobre una ligadura de fraseo. El fraseo se puede situar manualmente por encima o por debajo de las notas; véase Sección 36.1 [Dirección y posición], página 783.

Las ligaduras de fraseo simultáneas o superpuestas se introducen usando `\=` igual que se hace con las ligaduras de expresión normales, véase Sección 3.2.1 [Ligaduras de expresión], página 171.

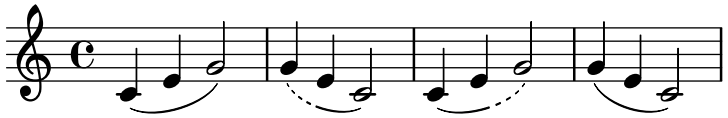
Las ligaduras de fraseo pueden ser continuas, de puntos o de rayas. El estilo predeterminado para las ligaduras de fraseo es el continuo:

```
\relative {
  c'4\( e g2\)
  \phrasingSlurDashed
  g4\( e c2\)
  \phrasingSlurDotted
  c4\( e g2\)
  \phrasingSlurSolid
  g4\( e c2\)
}
```



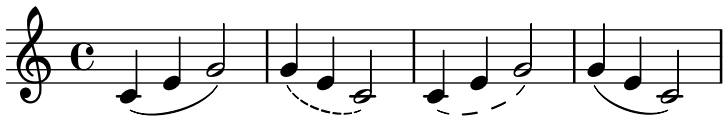
Se puede hacer también que las ligaduras de fraseo sean semi-discontinuas (la primera mitad discontinua y la segunda mitad continua) o semi-continuas (la primera mitad continua y la segunda mitad discontinua):

```
\relative {
  c'4\( e g2\)
  \phrasingSlurHalfDashed
  g4\( e c2\)
  \phrasingSlurHalfSolid
  c4\( e g2\)
  \phrasingSlurSolid
  g4\( e c2\)
}
```



Se pueden definir patrones de discontinuidad para las ligaduras de fraseo:

```
\relative {
  c'4\( e g2\)
  \phrasingSlurDashPattern 0.7 0.75
  g4\( e c2\)
  \phrasingSlurDashPattern 0.5 2.0
  c4\( e g2\)
  \phrasingSlurSolid
  g4\( e c2\)
}
```



Las definiciones de patrones de discontinuidad tienen la misma estructura que las definiciones de patrones de discontinuidad de las ligaduras de expresión. Para ver más información acerca de patrones de discontinuidad complejos, consulte los fragmentos de código bajo Sección 3.2.1 [Ligaduras de expresión], página 171.

Instrucciones predefinidas

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`, `\phrasingSlurDashed`, `\phrasingSlurDotted`, `\phrasingSlurHalfDashed`, `\phrasingSlurHalfSolid`, `\phrasingSlurDashPattern`, `\phrasingSlurSolid`.

Véase también

Manual de aprendizaje: Sección “Acerca de la no anidabilidad de llaves y ligaduras” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783, Sección 3.2.1 [Ligaduras de expresión], página 171.

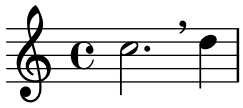
Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PhrasingSlur” en *Referencia de Funcionamiento Interno*.

3.2.3 Marcas de respiración

Las respiraciones (acortando la nota anterior para tomar aire) se introducen utilizando `\breathe`:

```
\fixed c'' { c2. \breathe d4 }
```

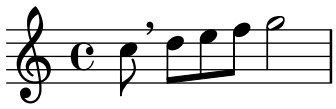


Para un corte breve del sonido que no se extrae de la nota anterior, véase Sección 2.2.4 [Cesuras], página 76.

A diferencia de otras marcas de expresión, los símbolos de respiración no están asociados con la nota anterior, sino que son eventos independientes. Por ello, todas las marcas de expresión que están adjuntas a la nota anterior, cualquier corchete recto que indique barrado manual, y cualquier paréntesis que indique ligaduras de expresión y de fraseo, deben escribirse antes de `\breathe`. La propia `\breathe` no acepta articulaciones, pero véase Sección 2.2.4 [Cesuras], página 76.

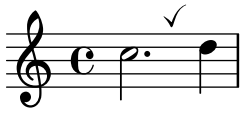
Las marcas de respiración dan por terminadas las barras automáticas; para sobreescribir este comportamiento, véase Sección 2.4.3 [Barras manuales], página 114.

```
\fixed c'' { c8 \breathe d e f g2 }
```



La propiedad de contexto `breathMarkType` controla cuál de las varias indicaciones de respiración predefinidas crea la instrucción `\breathe`. Véase Sección B.14 [Lista de signos de respiración], página 936.

```
\fixed c'' {
  \set breathMarkType = #'tickmark
  c2. \breathe d4
}
```



Véase también

Glosario musical: Sección “caesura” en *Glosario Musical*.

Referencia de la notación: Sección 2.2.4 [Cesuras], página 76, Sección 17.4.4 [Divisiones], página 554.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BreathingEvent” en *Referencia de Funcionamiento Interno*, Sección “BreathingSign” en *Referencia de Funcionamiento Interno*, Sección “Breathing_sign_engraver” en *Referencia de Funcionamiento Interno*.

3.2.4 Caídas y elevaciones

Se pueden expresar *caídas* y *subidas* de tono (falls y doits) añadidas a las notas mediante la instrucción `\bendAfter`. La dirección de la caída o elevación se indica con un signo más o menos (arriba o abajo). El número indica el intervalo de alturas sobre el que se extiende la caída o elevación *partiendo de* la nota principal.

```
\relative c'' {
  c2\bendAfter 4
  c2\bendAfter -4
  c2\bendAfter 6.5
  c2\bendAfter -6.5
  c2\bendAfter 8
  c2\bendAfter -8
}
```



Fragmentos de código seleccionados

Ajustar la forma de las subidas y caídas de tono

Puede ser necesario trucar la propiedad `shortest-duration-space` para poder ajustar el tamaño de las caídas y subidas de tono («falls» y «doits»).

```
\relative c'' {
  \override Score.SpacingSpanner.shortest-duration-space = 4.0
  c2-\bendAfter 5
  c2-\bendAfter -4.75
  c2-\bendAfter 8.5
  c2-\bendAfter -6
}
```



Véase también

Glosario musical: Sección “fall” en *Glosario Musical*, Sección “doit” en *Glosario Musical*.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

3.3 Expresiones como líneas

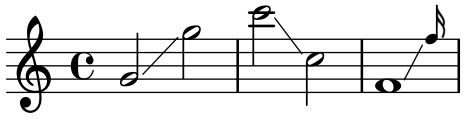
Esta sección explica cómo crear diversas marcas de expresión que siguen un camino lineal: glissandi, arpegios y trinos.

3.3.1 Glissando

Se crea un *glissando* añadiendo `\glissando` después de la nota:

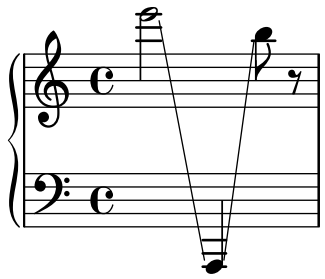
```
\relative {
  g'2\glissando g'
  c2\glissando c,
  \afterGrace f,1\glissando f'16
```

```
}
```



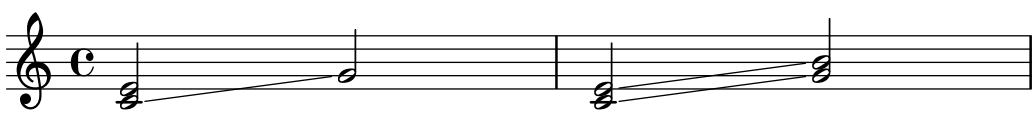
Un glissando puede conectar notas de un pentagrama a otro:

```
\new PianoStaff <<
  \new Staff = "right" {
    e'''2\glissando
    \change Staff = "left"
    a,,4\glissando
    \change Staff = "right"
    b''8 r |
  }
  \new Staff = "left" {
    \clef bass
    s1
  }
>>
```



Un glissando puede conectar las notas dentro de un acorde. Si se necesita hacer algo que no sea una conexión de una nota con otra entre los dos acordes, las conexiones entre las notas se definen estableciendo `\glissandoMap` a una lista de Scheme. Los elementos son pares de números enteros; cada par *(x . y)* crea una línea de glissando desde la nota *x*-ésima del primer acorde hasta la nota *y*-ésima del segundo acorde. Las notas se numeran desde cero en el orden en que aparecen en el archivo `.ly` de entrada. No todas las notas necesitan formar parte de un glissando.

```
\relative {
  <c' e>2\glissando g'
  <c, e>\glissando <g' b>
  \break
  \set glissandoMap = #'((0 . 1) (1 . 0))
  <c, g'>\glissando <d a'>
  \set glissandoMap = #'((0 . 0) (0 . 1) (0 . 2))
  c\glissando <d f a>
  \set glissandoMap = #'((2 . 2) (0 . 0))
  <f d a'>\glissando <c f c'>
}
```





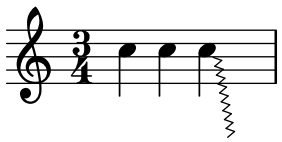
Se pueden crear distintos tipos de glissando. Para ver más detalles, consulte Sección 36.5 [Estilos de línea], página 791.

Fragmentos de código seleccionados

Glissando contemporáneo

Se puede tipografiar un gissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



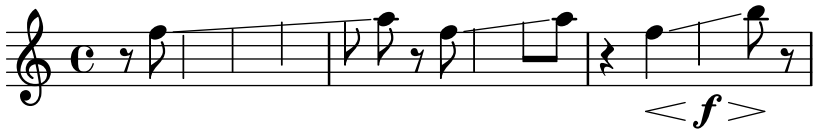
Añadir marcas de tiempo a glissandos largos

Los pulsos que se saltan en glissandos muy largos se indican a veces mediante marcas de tiempo que consisten en figuras sin cabeza. Estas plicas se pueden usar también para albergar indicaciones expresivas intermedias.

Si las plicas no quedan bien alineadas con el glissando, podría ser necesario recolocarlas ligeramente.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando \glissandoSkipOn f4 g a |
  a8\noBeam \glissandoSkipOff a8
```

```
    r8 f8\glissando \glissandoSkipOn g4 a8 \glissandoSkipOff a8 |
  r4 f\glissando\< \glissandoSkipOn a4\f\> \glissandoSkipOff b8\! r |
}
```

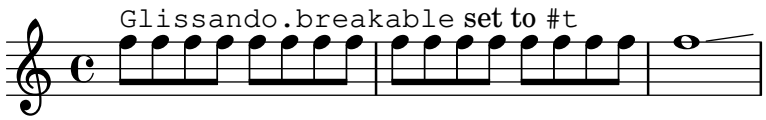


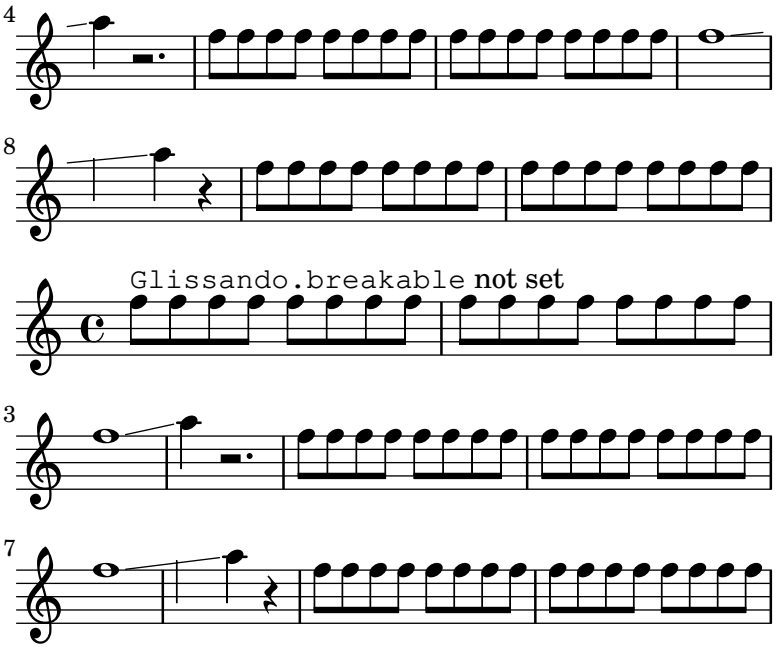
Hacer que los glissandos se puedan dividir en el salto de línea

Normalmente, LilyPond rehúsa insertar un salto de línea automático en el punto en que un glissando cruza la barra de compás. Este comportamiento se puede cambiar estableciendo el valor de la propiedad `Glissando.breakable` a `#t`. También, estableciendo el valor de la propiedad `after-line-breaking` a `#t` hace que la línea del glissando continúe después del salto.

La propiedad `breakable` no afecta a los saltos manuales insertados con instrucciones como `\break`.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

music = {
  \repeat unfold 16 f8 |
  f1\glissando |
  a4 r2. |
  \repeat unfold 16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \repeat unfold 16 f8
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable
               set to \typewriter "#t" }
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}
```





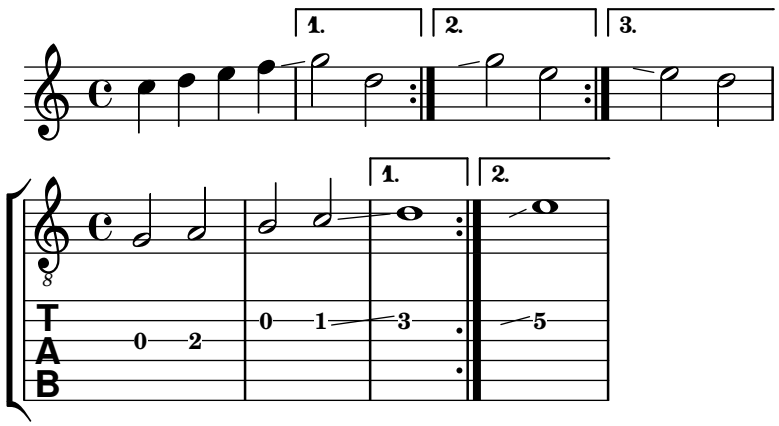
Extender glissandos sobre repeticiones

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
```

```
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```



Véase también

Glosario musical: Sección “glissando” en *Glosario Musical*.

Referencia de la notación: Sección 36.5 [Estilos de línea], página 791.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Glissando” en *Referencia de Funcionamiento Interno*.

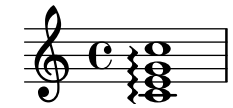
Advertencias y problemas conocidos

La impresión de texto sobre la línea (como *gliss.*) no está contemplada.

3.3.2 Arpegio

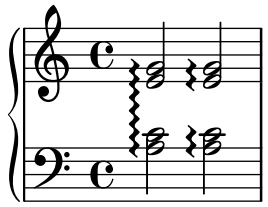
Un signo de *acorde arpegiado* (conocido también como acorde quebrado) sobre un acorde, se denota adjuntando `\arpeggio` al acorde:

```
\relative { <c' e g c>1\arpeggio }
```



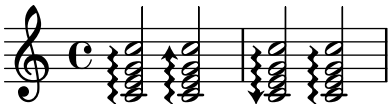
Se pueden conectar arpegios entre pentagramas o voces distintas mediante el establecimiento de la propiedad `connectArpeggios` dentro de un contexto de agrupación de pentagramas. Esto conecta a todos los arpegios que estén dentro de dicho contecto:

```
\new PianoStaff \fixed c' <<
  \new Staff {
    \once \set PianoStaff.connectArpeggios = ##t
    <e g>2\arpeggio q\arpeggio
  }
  \new Staff {
    \clef "bass"
    <a, c>2\arpeggio q\arpeggio
  }
>>
```



Existen instrucciones para especificar la dirección en la ejecución. `\arpeggioNormal` produce la vuelta al arpegio normal:

```
\relative {
  <c' e g c>2\arpeggio

  \arpeggioArrowUp
  <c e g c>2\arpeggio

  \arpeggioArrowDown
  <c e g c>2\arpeggio

  \arpeggioNormal
  <c e g c>2\arpeggio
}
```

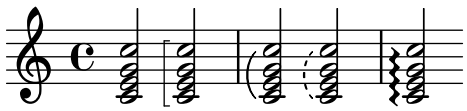


Estas instrucciones predefinidas modifican internamente la propiedad `arpeggio-direction`; véanse sus definiciones completas en el archivo `ly/property-init.ly`.

Se puede cambiar el símbolo que crea la instrucción `\arpeggio`:

```
\relative {
  <c' e g c>2

  \arpeggioBracket
  <c e g c>2\arpeggio

  \arpeggioParenthesis
  <c e g c>2\arpeggio
```

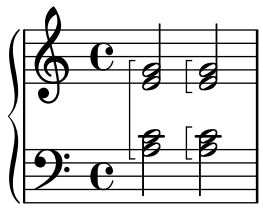
```
  \arpeggioParenthesisDashed
  <c e g c>2\arpeggio

  \arpeggioNormal
  <c e g c>2\arpeggio
}
```



Estas instrucciones predefinidas sobreescriben internamente la propiedad `stencil` del objeto `Arpeggio`, y puede también adaptar su `X-extent` (esto es, la dimensión horizontal que le supone no colisionar con otros objetos).

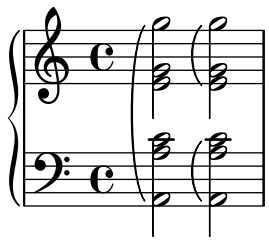
Para hacer un corchete que designe un acorde no arpegiado, es mejor usar `\nonArpeggiato` que usar `\arpeggio` y alterar la apariencia. Esto crea un objeto `ChordBracket`.

```
\new PianoStaff \fixed c' <<
  \new Staff {
    \once \set PianoStaff.connectChordBrackets = ##t
    <e g>2\nonArpeggiato q\nonArpeggiato
  }
  \new Staff {
    \clef "bass"
    <a, c>2\nonArpeggiato q\nonArpeggiato
  }
>>
```



Para hacer una ligadura vertical que designa un acorde *quasi no arpegiado*, es mejor usar `\chordSlur` que usar `\arpeggio` y alterar la apariencia. Esto crea un objeto `ChordSlur`.

```
\new PianoStaff \fixed c <<
  \new Staff {
    \once \set PianoStaff.connectChordSlurs = ##t
    <e' g' g''>2\chordSlur q\chordSlur
  }
  \new Staff {
    \clef "bass"
    <a, a c'>2\chordSlur q\chordSlur
  }
>>
```



Las propiedades de discontinuidad de ChordSlur y del paréntesis del arpegio se controlan mediante la propiedad dash-definition (véase Sección 3.2.1 [Ligaduras de expresión], página 171).

Los acordes arpegiados se pueden desarrollar explícitamente utilizando *ligaduras de unión*. Para más información, véase Sección 2.1.4 [Ligaduras de unión], página 63.

Instrucciones predefinidas

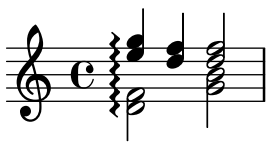
\arpeggio, \arpeggioArrowUp, \arpeggioArrowDown, \arpeggioNormal, \arpeggioBracket, \arpeggioParenthesis, \arpeggioParenthesisDashed, \chordSlur, \nonArpeggiato.

Fragmentos de código seleccionados

Crear arpegios entre notas de voces distintas

Se puede trazar un símbolo de *arpeggio* entre notas de distintas voces que están sobre el mismo pentagrama si el grabador Span_arpeggio_engraver se traslada al contexto de Staff.

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Véase también

Glosario musical: Sección “arpeggio” en *Glosario Musical*.

Referencia de la notación: Sección 3.2.1 [Ligaduras de expresión], página 171, Sección 2.1.4 [Ligaduras de unión], página 63.

Archivos instalados: ly/property-init.ly.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Arpeggio” en *Referencia de Funcionamiento Interno*, Sección “ChordBracket” en *Referencia de Funcionamiento Interno*, Sección “ChordSlur” en *Referencia de Funcionamiento Interno*, Sección “Slur” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las instrucciones predefinidas como \arpeggioArrowUp solamente se aplican al contexto en curso, y por ello no afectan a los arpegios que abarcan varias voces o pentagramas. En tales casos, estas instrucciones requieren utilizarse en un bloque \context dentro de \layout, o en un bloque \with, como se explica en Sección 33.5 [Cambiar los valores por omisión de los contextos], página 753. De manera alternativa, en lugar de usar las abreviaturas predefinidas, se recomienda sebreescribir directamente las propiedades adecuadas para el objeto Arpeggio dentro del contexto correspondiente; por ejemplo:

```
\override Staff.Arpeggio.stencil = #ly:chord-bracket::print
```

```
\override Staff.Arpeggio.X-extent = #ly:chord-bracket::width
```

para imprimir corchetes de arpegio que comprenden más de una voz en el nivel de `Staff`, o bien

```
\override PianoStaff.Arpeggio.arpeggio-direction = #UP
```

para imprimir arpegios con punta de flecha (hacia arriba) que comprendan más de un pentagrama dentro de un contexto `PianoStaff`.

No es posible mezclar arpegios conectados y no conectados en un `PianoStaff` en el mismo instante de tiempo.

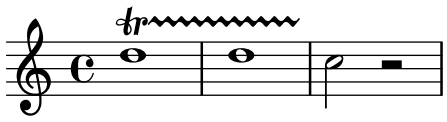
La forma sencilla de especificar el corchete en los arpegios de paréntesis no funciona para los arpegios de pentagrama cruzado; véase [Corchetes de acorde de pentagrama cruzado], página 423.

3.3.3 Trinos

Los trinos cortos sin línea extensora se imprimen con `\trill`; véase Sección 3.1.1 [Articulaciones y ornamentos], página 156.

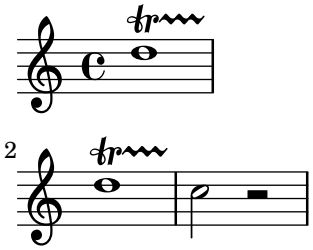
Los *trinos* largos mantenidos, con línea de extensión, se hacen con `\startTrillSpan` y `\stopTrillSpan`:

```
\relative {
  d''1\startTrillSpan
  d1
  c2\stopTrillSpan
  r2
}
```



Un trino extendido que atraviesa un salto de línea recomienza exactamente encima de la primera nota de la nueva línea.

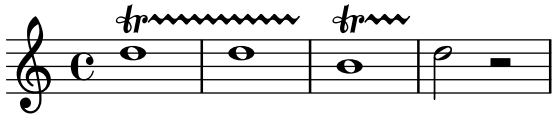
```
\relative {
  d''1\startTrillSpan
  \break
  d1
  c2\stopTrillSpan
  r2
}
```



Los trinos extendidos consecutivos funcionan sin necesidad de instrucciones `\stopTrillSpan` explícitas, porque cada trino se convertirá automáticamente en el borde derecho del trino anterior.

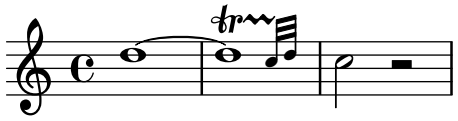
```
\relative {
  d''1\startTrillSpan
  d1
  b1\startTrillSpan
  d2\stopTrillSpan
```

```
  r2
}
```



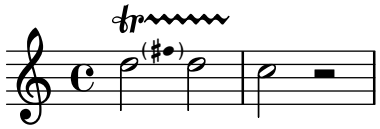
También se pueden combinar los trinos con notas de adorno. La sintaxis de esta construcción y el método para colocar las notas de adorno con precisión están descritos en Sección 2.6.1 [Notas de adorno], página 148.

```
\relative {
  d''1~\afterGrace
  d1\startTrillSpan { c32[ d]\stopTrillSpan }
  c2 r2
}
```



Los trinos que se tienen que ejecutar sobre notas auxiliares explícitas se pueden tipografiar con la instrucción `pitchedTrill`. El primer argumento es la nota principal, y el segundo es la nota *trinada*, que se imprime como una cabeza de nota, sin plica y entre paréntesis.

```
\relative {
  \pitchedTrill
  d''2\startTrillSpan fis
  d2
  c2\stopTrillSpan
  r2
}
```



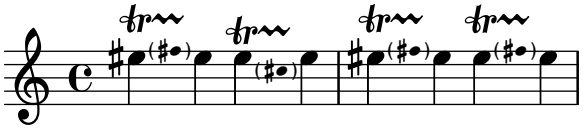
La alteración del primer trino con altura definida dentro de un compás siempre se imprime, incluso si es un becuadro.

```
{
  \key d \major
  \pitchedTrill
  d'2\startTrillSpan cis d\stopTrillSpan
  \pitchedTrill
  d2\startTrillSpan c d\stopTrillSpan
  \pitchedTrill
  d2\startTrillSpan e d\stopTrillSpan
}
```



Es necesario añadir manualmente las alteraciones subsiguientes (de la misma nota dentro del mismo compás).

```
\relative {
  \pitchedTrill
  eis''4\startTrillSpan fis
  eis4\stopTrillSpan
  \pitchedTrill
  eis4\startTrillSpan cis
  eis4\stopTrillSpan
  \pitchedTrill
  eis4\startTrillSpan fis
  eis4\stopTrillSpan
  \pitchedTrill
  eis4\startTrillSpan fis!
  eis4\stopTrillSpan
}
```



Instrucciones predefinidas

`\startTrillSpan`, `\stopTrillSpan`.

Véase también

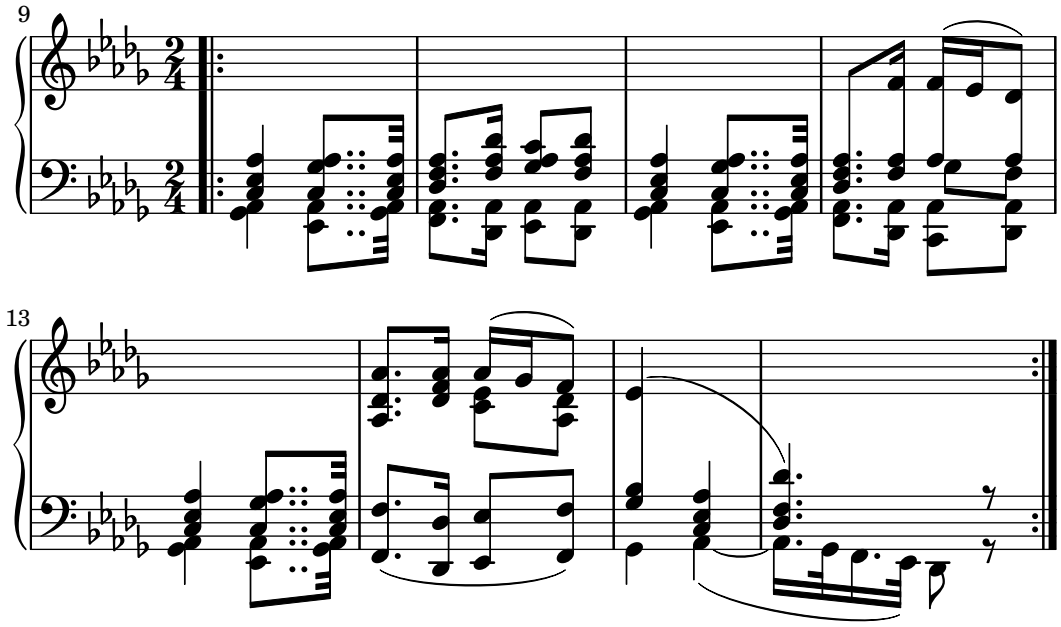
Glosario musical: Sección “trill” en *Glosario Musical*.

Referencia de la notación: Sección 3.1.1 [Articulaciones y ornamentos], página 156, Sección 2.6.1 [Notas de adorno], página 148.

Fragmentos de código: Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TrillSpanner” en *Referencia de Funcionamiento Interno*, Sección “TrillPitchHead” en *Referencia de Funcionamiento Interno*, Sección “TrillPitchAccidental” en *Referencia de Funcionamiento Interno*, Sección “TrillPitchParentheses” en *Referencia de Funcionamiento Interno*, Sección “TrillPitchGroup” en *Referencia de Funcionamiento Interno*, Sección “Pitched_trill_engraver” en *Referencia de Funcionamiento Interno*.

4 Repeticiones



La repetición es un concepto fundamental en música, y existen varios tipos de notación para las repeticiones. LilyPond contempla los siguientes tipos de repetición:

`volta (primera y segunda vez)`
: Esta es la notación estándar de las repeticiones con o sin finales alternativos (primera y segunda vez). La sección que se repite está encerrada entre barras de repetición, pero la barra inicial no se escribe cuando la sección que se repite aparece al prinicipio de la pieza. Los finales alternativos se imprimen en secuencia, con líneas de casilla de primera y segunda vez, y numerados con la repetición o repeticiones a las que se aplican.

`segno`
: Este tipo contempla varios casos de *da capo* y *dal segno*. La sección que se repite comienza con una marca de segno mark, excepto si está al comienzo de la pieza. Los finales alternativos se imprimen en secuencia y se marcan con símbolos de coda, y se puede aplicar opcionalmente una etiqueta de sección como 'Coda' a la última alternativa. La sección que se repite termina con una instrucción como *D.S.*

`unfold (desplegada)`
: La música repetida se imprime en su totalidad el número de veces especificado.

`percent (porcentaje)`
: Este tipo consiste en repeticiones de pulsos o compases enteros. Son como barras inclinadas sueltas o símbolos de porcentaje.

`tremolo (trémolo)`
: Este tipo se usa para escribir barras de trémolo.

Las construcciones de acorde se pueden repetir usando el símbolo de repetición de acordes, q. Véase Sección 5.1.2 [Repetición de acordes], página 220.

4.1 Repeticiones largas

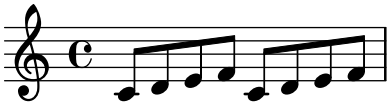
Esta sección trata sobre la forma de introducir repeticiones largas, normalmente de varios compases.

4.1.1 Repeticiones explícitas

La instrucción `\repeat unfold` repite la música escribiéndola entera un cierto número de veces. La sintaxis es la misma que la de las instrucciones `\repeat volta` y `\repeat segno`, que aparecen documentadas en las secciones siguientes.

También existe una forma abreviada, `\*`, que puede convenir dentro de una pieza que use profusamente repeticiones desplegadas:

```
\*2 { c'8 d' e' f' }
```



Para evitar la redundancia, el desplegado de repeticiones no se estudia aquí con mucho detalle; sin embargo, algunos de los ejemplos de las secciones siguientes ilustran las repeticiones de varias maneras usando la instrucción `\unfoldRepeats` para convertir las formas `volta` o `segno` a la forma `unfold`. Para ver otro importante uso de la instrucción `\unfoldRepeats`, véase Sección 24.6 [Uso de las repeticiones con el MIDI], página 662.

Existen ciertos puntos de interés que son específicos de la instrucción `\repeat unfold`.

En algunos casos, especialmente dentro de un contexto `\relative`, el resultado del despliegue de repeticiones no es el mismo que el de escribir la expresión musical de entrada varias veces, p.ej.,

```
\repeat unfold 2 { a'4 b c d | }
```

se diferencia de lo siguiente en un cambio de octava:

```
a'4 b c d |
a'4 b c d |
```

También, el anidado de `\repeat unfold` puede resultar práctico de maneras que el anidado de `\repeat volta` o `\repeat segno` no lo serían.

> **Nota:** Si se incluye `\relative` dentro de `\repeat` sin instanciar explícitamente el contexto `Voice`, aparecerán pentagramas extra no deseados. Véase Sección "An extra staff appears" en *Utilización del Programa*.

Véase también

Fragmentos de código: Sección "Repeticiones" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "UnfoldedRepeatedMusic" en *Referencia de Funcionamiento Interno*.

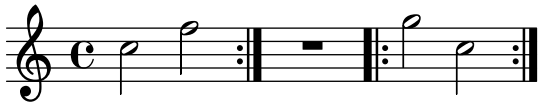
4.1.2 Repeticiones sencillas

Esta es la sintaxis de una repetición sin variación:

```
\repeat volta número de veces expresión
```

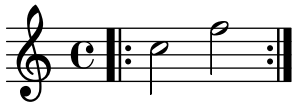
donde *expresión* es la expresión musical que se ha de repetir.

```
\fixed c'' {
  \repeat volta 2 { c2 f }
  R1
  \repeat volta 2 { g2 c }
}
```



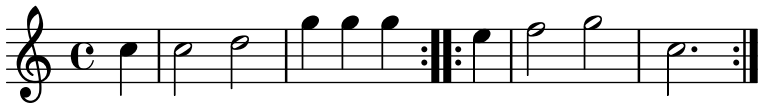
De forma predeterminada no se imprimen las dobles barras de apertura de repetición al principio de una pieza, siguiendo las convenciones de la tipografía musical clásica. Sin embargo, sí se escriben tradicionalmente en ciertos contextos, como en las hojas guía de acordes para estándares de jazz. Esto se puede conseguir configurando la propiedad `printInitialRepeatBar`.

```
\fixed c'' {
  \set Score.printInitialRepeatBar = ##t
  \repeat volta 2 { c2 f }
}
```



Una sección repetida que comienza en mitad de un compás termina generalmente en la misma posición dentro de un compás posterior de tal forma que los dos extremos hacen un compás entero. En estos casos, las líneas divisorias de repetición no constituyen la frontera de un compás, por lo que no se debe poner ahí ninguna comprobación de compás. De forma similar, no se debe escribir ninguna instrucción `\partial` dentro del fragmento repetido, porque los compases están completos; sin embargo, hay que escribir una instrucción `\partial` antes de la repetición cuando aparece un compás verdaderamente incompleto la primera vez.

```
\fixed c'' {
  \partial 4
  \repeat volta 2 {
    c4
    c2 d
    g4 g g
  }
  \repeat volta 2 {
    e4
    f2 g
    c2.
  }
}
```



4.1.3 Finales alternativos (primera y segunda vez)

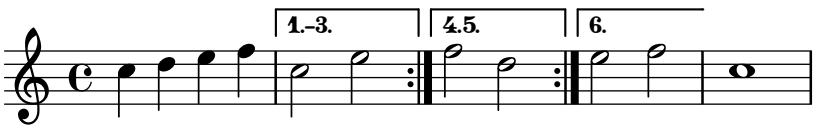
Las repeticiones con finales alternativos de primera y segunda vez se pueden escribir de dos formas. Esta es la sintaxis recomendada:

```
\repeat volta número_de_veces {
  expresión...
  \alternative {
    \volta lista_de_números expresión
    \volta lista_de_números expresión
    ...
  }
```

```
}
```

donde *expresión* es una expresión musical, *expresión...* es un número cualquiera de expresiones musicales, y *lista_de_números* es una lista, separada por comas, de números de volta (o vez) en el rango desde 1 hasta *número_de_veces*.

```
\fixed c'' {
  \repeat volta 6 {
    c4 d e f
    \alternative {
      \volta 1,2,3 { c2 e }
      \volta 4,5 { f2 d }
      \volta 6 { e2 f }
    }
  }
  c1
}
```

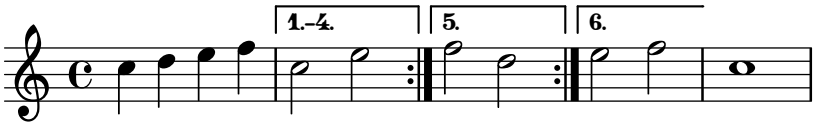


Una sintaxis anterior, en la que el bloque `\alternative` se escribía a continuación fuera de la expresión musical repetida, aún está contemplada y tiene el mismo efecto. Esta sintaxis requiere que `\repeat` aparezca literalmente: una cláusula `\alternative` escrita después no se enlaza con una función musical ni variable que contenga `\repeat`.

```
\repeat volta número_de_veces expresión
\alternative {
  \volta lista_de_números expresión
  \volta lista_de_números expresión
  ...
}
```

Las especificaciones de `\volta` dentro de un bloque `\alternative` son opcionales según el criterio de todos o ninguno. Si se omiten, las alternativas se usan una vez cada una, pero la primera se repite las veces necesarias para que se cumpla el número de veces de la repetición.

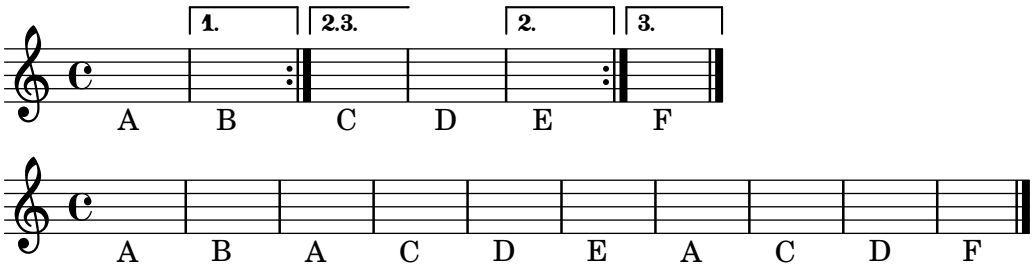
```
\fixed c'' {
  \repeat volta 6 {
    c4 d e f
    \alternative {
      { c2 e }
      { f2 d }
      { e2 f }
    }
  }
  c1
}
```



Los bloques `\alternative` se pueden anidar unos dentro de otros.

```
music = \fixed c' {
```

```
  \repeat volta 3 {
    s1_"A"
    \alternative {
      \volta 1 { s1_"B" }
      \volta 2,3 {
        \once \override Score.VoltaBracket.musical-length =
          \musicLength 1
        s1_"C"
        s1_"D"
        \alternative {
          \volta 2 { s1_"E" }
          \volta 3 { s1_"F" }
        }
      }
    }
  }
  \fine
}

\score { \music }
\score { \unfoldRepeats \music }
```



Nota: Cada elemento de un bloque `\alternative` se trata como un final alternativo. Algo tan simple como una comprobación de compás en el lado incorrecto de una llave puede producir resultados inesperados.

Nota: Si incluye `\relative` dentro de un `\repeat` sin crear explícitamente una instancia del contexto `Voice`, aparecerán pentagramas añadidos e indeseados. Véase Sección "An extra staff appears" en *Utilización del Programa*.

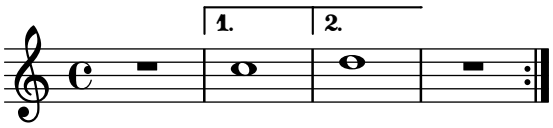
Nota: Cuando se habilita una numeración alternativa para los compases, se aplica a las alternativas encerradas entre llaves que están en la parte más externa.

4.1.4 Otras variantes dentro de las secciones repetidas

Un bloque `\alternative` se puede usar dentro de un bloque `\repeat` para producir una notación similar a los finales alternativos (véase Sección 4.1.3 [Finales alternativos (primera y segunda vez)], página 192).

```
\fixed c'' {
```

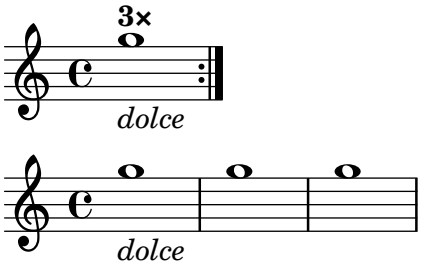
```
  \repeat volta 2 {
    R1
    \alternative {
      \volta 1 { c1 }
      \volta 2 { d1 }
    }
    R1
  }
}
```



La instrucción \volta no se limita a su uso sobre los elementos de un bloque \alternative. Se puede emplear en cualquier lugar dentro de \repeat para designar música para algunas vueltas en particular, aunque no crea casillas de repetición de primera y segunda vez en otros casos.

Cuando se despliega un \repeat, se omite la música específica de los finales alternativos de todas aquellas vueltas a las que no se aplica. Si se da una lista vacía de Scheme en lugar de los números de repetición, la música desaparece por completo.

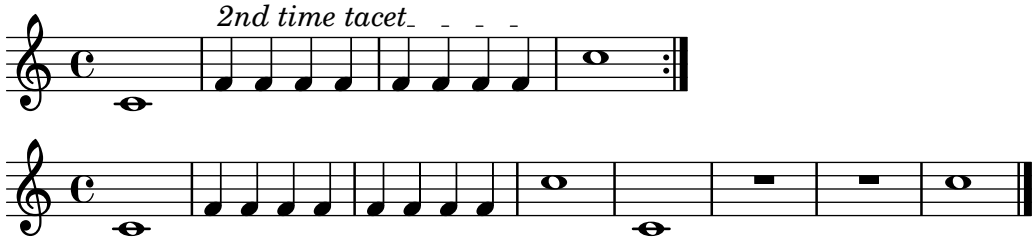
```
music = \repeat volta 3 {
  \volta #'() { <>^\markup { \bold "3×" } }
  \volta 1 { <>_\markup { \italic dolce } }
  g''1
}

\score { \music }
\score { \unfoldRepeats \music }
```



Cuando se despliega un \repeat, puede ser deseable no solo filtrar toda la música que no es específica de los finales alternativos, sino también añadir la música que no estaba presente en la forma plegada. La instrucción \unfolded designa la música que queremos que se ignore hasta que se despliegue el \repeat que lo encierra.

```
music = \fixed c' {
  \repeat volta 2 {
    c1
    <<
      \volta #'() {
        \once \override TextSpanner.bound-details.left.text =
          "2nd time tacet"
        s4*7\startTextSpan s4\stopTextSpan
      }
      \volta 1 { f4 f f f | f f f f }
```

```
      \volta 2 { \unfolded { R1*2 } }
    >>
    c'1
  }
  \fine
}

\score { \music }
\score { \unfoldRepeats \music }
```

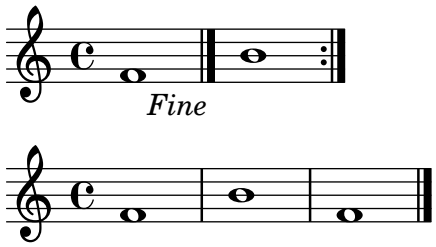


Nota: las instrucciones `\volta` y `\unfolded` funcionan con respecto a la repetición más interior que las incluye.

4.1.5 Repeticiones Al fine

La instrucción `\fine` marca el final de la música pero no obliga a que lo sea. Cuando se despliega una repetición que contiene `\fine`, la instrucción `\fine` se despliega como cualquier otra música. Para obtener un desplegado correcto, es necesario especificar el final alternativo en que se debe ejecutar el *Fine* y los finales alternativos en los que toda música que aparezca a continuación deba ser ejecutada (véase Sección 4.1.4 [Otras variantes dentro de las secciones repetidas], página 194).

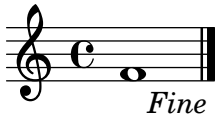
```
music = \fixed c' {
  \repeat volta 2 {
    f1
    \volta 2 \fine
    \volta 1 b1
  }
}

\score { \music }
\score { \unfoldRepeats \music }
```



Como se acaba de mostrar, en el final escrito de la música, `\fine` crea una barra de compás de final sin ninguna indicación de *Fine*. Para forzar que aparezca el *Fine* en estos casos, establezca a verdadero el valor de la propiedad de contexto `finalFineTextVisibility`.

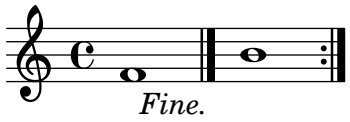
```
\fixed c' {
  \set Score.finalFineTextVisibility = ##t
```

```
  f1
  \fine
}
```



Para modificar el texto que imprime la instrucción `\fine`, configure el valor de la propiedad de contexto `fineText`.

```
\fixed c' {
  \set Score.fineText = "Fine."
  \repeat volta 2 {
    f1
    \volta 2 \fine
    \volta 1 b1
  }
}
```



Para ver más detalles acerca de las interacciones con otros tipos de líneas divisorias y opciones para modificar su aspecto, véase Sección 2.5.2 [Automatic bar lines], página 130.

Véase también

Glosario musical: Sección “fine” en *Glosario Musical*.

Referencia de la notación: Sección 2.5.2 [Automatic bar lines], página 130, Sección 4.1.4 [Otras variantes dentro de las secciones repetidas], página 194, Sección 2.5.7 [Divisores de sección], página 147.

Fragmentos de código: Sección “Repeticiones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “FineEvent” en *Referencia de Funcionamiento Interno*, Sección “Jump_engraver” en *Referencia de Funcionamiento Interno*, Sección “JumpScript” en *Referencia de Funcionamiento Interno*.

4.1.6 Estructura de las repeticiones de Segno

`\repeat segno` se diferencia de `\repeat volta` solo en la notación que resulta impresa. Consulte las secciones anteriores para ver información general sobre la escritura de música con repeticiones, finales alternativos y variaciones. Esta sección se ocupa de los detalles de la notación de segno sin volver a explicar la sintaxis del código de entrada.

`\repeat segno` efectúa la notación de las repeticiones que tienen indicaciones *D.C.* o *D.S.*. Marca con un símbolo de segno el comienzo de la sección que se repite cuando no está al comienzo de la pieza. También marca los finales alternativos con símbolos de coda en lugar de casillas de primera y segunda vez, siempre que los finales se suponga que se ejecutan en orden, esto es, no `\volta 1,3` y después `\volta 2,4`.

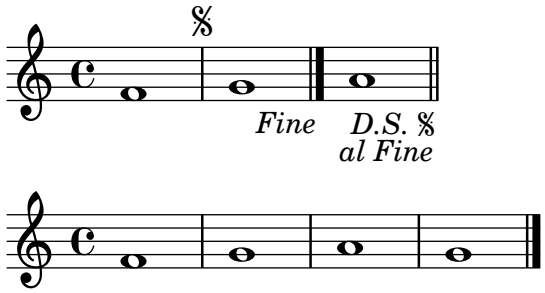
Cuando se habilita la numeración de compases de los finales alternativos, también se aplica a las alternativas cuya notación presenta casillas de primera y segunda vez si se han creado con `\repeat segno` o con `\repeat volta`, pero no se aplica a los finales alternativos que contienen símbolos de coda.

A continuación aparecen muestras de algunos usos comunes.

al fine

Las indicaciones de repetición incluyen *al Fine* si aparece una instrucción `\fine` en cualquier momento anterior (véase Sección 4.1.5 [Repeticiones Al fine], página 196).

```
music = \fixed c' {
  f1
  \repeat segno 2 {
    g1
    \volta 2 \fine
    \volta 1 a1
  }
  \section
}

\score { \music }
\score { \unfoldRepeats \music }
```

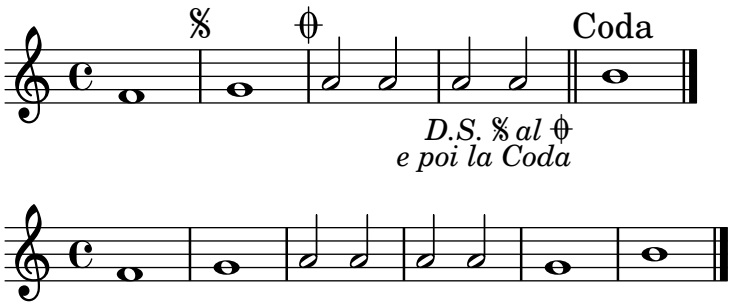


alla coda

El comienzo de cada final alternativo se marca con un `\codaMark \default` implícito. Las indicaciones de repetición dentro de las alternativas incluyen '*al . . . e poi la . . .*' que se refiere a la marca de la primera alternativa y la marca a la que se salta. Si la duración de la última alternativa sea cero, se suprime la llamada automática, permitiendo en su lugar establecer una etiqueta de sección.

```
music = \fixed c' {
  f1
  \repeat segno 2 {
    g1
    \alternative {
      \volta 1 { \repeat unfold 4 { a2 } }
      \volta 2 \volta #'() {
        \section
        \sectionLabel "Coda"
      }
    }
  }
  b1
  \fine
}

\score { \music }
```

```
\score { \unfoldRepeats \music }
```

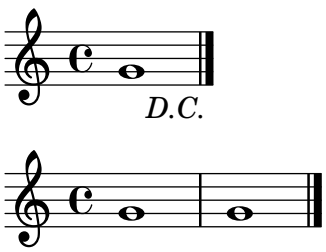


Se puede abreviar la indicacion de retorno preparando un procedimiento de formato alternativo (véase Sección 4.1.7 [Apariencia de las repeticiones de Segno], página 201).

da capo

Las indicaciones de repetición incluyen *D.C.* cuando la sección que se repite comienza al principio de la partitura. Los casos contemplados de *da capo* reflejan los casos contemplados de *dal segno*.

```
music = \fixed c' {
  \repeat segno 2 {
    g1
  }
  \fine
}

\score { \music }
\score { \unfoldRepeats \music }
```

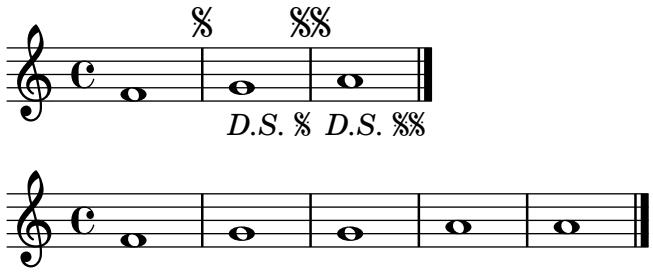


dal segno

Las indicaciones de repetición incluyen *D.S.* cuando la sección que se repite comienza después del principio de la partitura. El comienzo de la sección que se repite viene marcada con una llamada `\segnoMark \default` implícita.

```
music = \fixed c' {
  f1
  \repeat segno 2 {
    g1
  }
  \repeat segno 2 {
    a1
  }
  \fine
}
```

```
\score { \music }
\score { \unfoldRepeats \music }
```

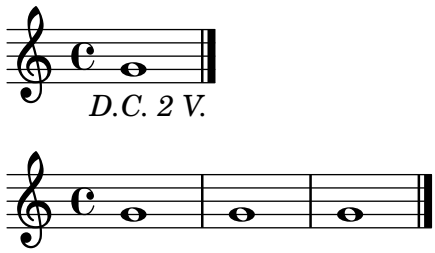


Se puede forzar una repetición del tipo *dal-segno* que comience al principio de la partitura (véase Sección 4.1.7 [Apariencia de las repeticiones de Segno], página 201).

retorno múltiple

Una indicación de repetición que se debe ejecutar más de una vez incluye una '*. . . V.*'. El número de veces que la indicación se ejecuta es una menos que el número de veces que se ejecuta el pasaje completo.

```
music = \fixed c' {
  \repeat segno 3 {
    g1
  }
  \fine
}

\score { \music }
\score { \unfoldRepeats \music }
```



Véase también

Glosario musical: Sección "da capo" en *Glosario Musical*, Sección "dal segno" en *Glosario Musical*, Sección "fine" en *Glosario Musical*.

Referencia de la notación: Sección 4.1.5 [Repeticiones Al fine], página 196, Sección 4.1.3 [Finales alternativos (primera y segunda vez)], página 192, Sección 2.5.2 [Automatic bar lines], página 130, Sección 4.1.4 [Otras variantes dentro de las secciones repetidas], página 194, Sección 2.5.7 [Divisores de sección], página 147, Sección 8.1.4 [Etiquetas de sección], página 316, Sección 4.1.2 [Repeticiones sencillas], página 191.

Fragmentos de código: Sección "Repeticiones" en *Fragmentos de código*.

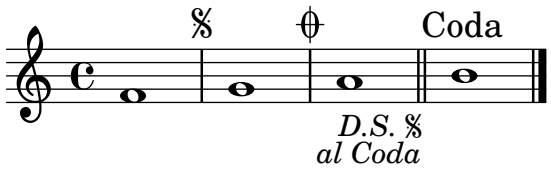
Referencia de funcionamiento interno: Sección "CodaMark" en *Referencia de Funcionamiento Interno*, Sección "JumpScript" en *Referencia de Funcionamiento Interno*, Sección "SectionLabel" en *Referencia de Funcionamiento Interno*, Sección "SegnoMark" en *Referencia de Funcionamiento Interno*.

4.1.7 Apariencia de las repeticiones de Segno

Las indicaciones de llamada y retorno que crea la instrucción `\repeat segno` son ajustables en ciertos aspectos.

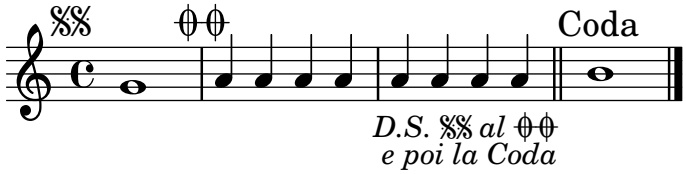
Las indicaciones de retorno se formatexn con el procedimiento de Scheme especificado en la propiedad `dalSegnoTextFormatter`. Hay un formateador alternativo predefinido que crea indicaciones más breves.

```
\fixed c' {
  \set Score.dalSegnoTextFormatter = #format-dal-segno-text-brief
  f1
  \repeat segno 2 {
    g1
    \alternative {
      \volta 1 { a1 }
      \volta 2 \volta #'() {
        \section
        \sectionLabel "Coda"
      }
    }
  }
  b1
  \fine
}
```



Los números de secuencia de las llamadas al principio de la sección que se repite y el comienzo de la primera alternativa se deben fijar explícitamente sin que interfieran con las indicaciones de retorno automáticas (véase Sección 4.1.8 [Marcas de repetición manual], página 207).

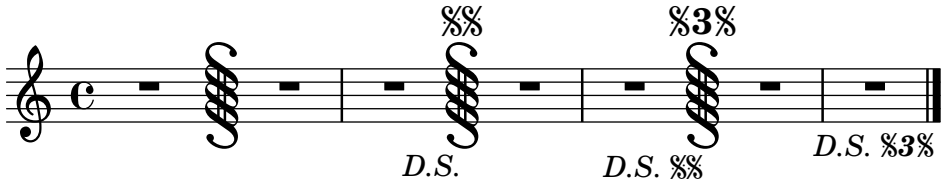
```
\fixed c' {
  \repeat segno 2 {
    \volta #'() { \segnoMark 2 }
    g1
    \alternative {
      \volta 1 {
        \volta #'() { \codaMark 2 }
        \repeat unfold 8 { a4 }
      }
      \volta 2 \volta #'() {
        \section
        \sectionLabel "Coda"
      }
    }
  }
  b1
  \fine
}
```



Sin la instrucción `\segnoMark 2` explícita, el ejemplo anterior se habría impreso como una repetición del tipo *da-capo*.

Como alternativa a la impresión de un símbolo de segno como llamada encima del pentagrama, es posible imprimirlo como una línea divisoria estableciendo el valor de la propiedad `segnoStyle` a `bar-line`. Para evitar la ambigüedad, solo permanece sin llamada el primer compás de segno.

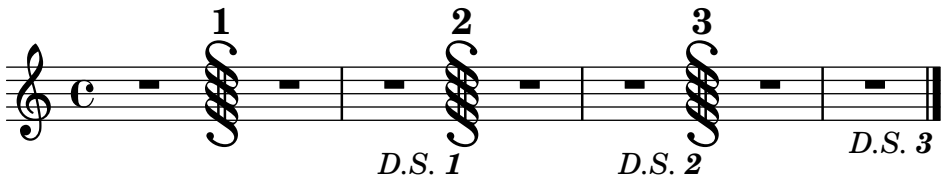
```
\fixed c' {
  \set Score.segnoStyle = #'bar-line
  R1
  \repeat unfold 3 {
    \repeat segno 2 {
      R1*2
    }
  }
  \fine
}
```



Allí donde un compás de segno coincide con otras líneas divisorias especiales, se selecciona automáticamente una línea de compás a partir de un conjunto predeterminado. Para cada combinación que se contempla, la línea divisoria se puede personalizar configurando una cierta propiedad de contexto (véase Sección 2.5.2 [Automatic bar lines], página 130).

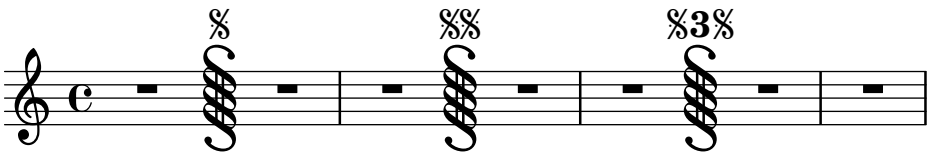
Las llamadas de segno y coda se formatean con procedimientos especificados en las propiedades `segnoMarkFormatter` y `codaMarkFormatter`. Estos son intercambiables con los procedimientos usados para formatear las llamadas de ensayo (véase Sección 2.5.5 [Llamadas de ensayo], página 142).

```
\fixed c' {
  \set Score.segnoMarkFormatter = #format-mark-numbers
  \set Score.segnoStyle = #'bar-line
  R1
  \repeat unfold 3 {
    \repeat segno 2 {
      R1*2
    }
  }
  \fine
}
```



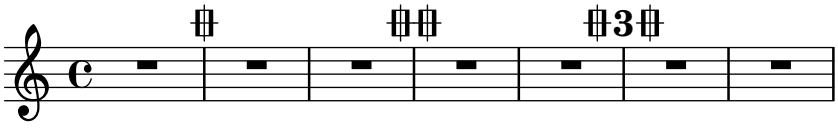
Hay un formateador alternativo predefinido para el segno que imprime una llamada incluso sobre la primera línea divisoria:

```
\fixed c' {
  \set Score.segnoMarkFormatter = #format-segno-mark
  \set Score.segnoStyle = #'bar-line
  R1
  \repeat unfold 3 {
    \segnoMark \default
    R1*2
  }
}
```



También hay un formateador alternativo predefinido de llamadas de coda que usa símbolos de `\varcoda`.

```
\fixed c' {
  \set Score.codaMarkFormatter = #format-varcoda-mark
  R1
  \repeat unfold 3 {
    \codaMark \default
    R1*2
  }
}
```



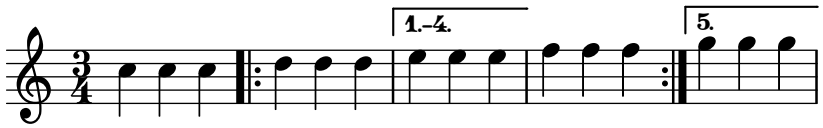
Fragmentos de código seleccionados

Acortar los corchetes de primera y segunda vez

De forma predeterminada, los corchetes de primera y segunda vez se trazan encima de los finales alternativos completos, pero es posible acortartlos sobreescribiendo el valor de `VoltaBracket.musical-length`. En el ejemplo siguiente, el corchete solo abarca un compás, que corresponde a una duración de 3/4.

```
\fixed c'' {
  \time 3/4
  c4 c c
  \repeat volta 5 {
    d4 d d
    \alternative {
      \volta 1,2,3,4 {
        \once \override Score.VoltaBracket.musical-length =
          \musicLength 2.
        e4 e e
        f4 f f
      }
```

```
      \volta 5 {
        g4 g g } } }
}
```

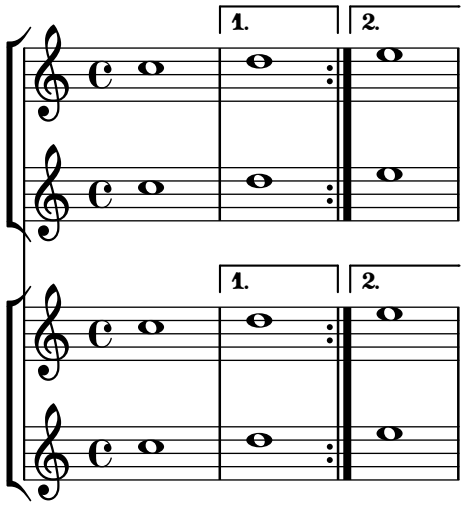


Casillas de primera y segunda vez en varios pentagramas

Añadiendo el grabador `Volta_engraver` al pentagrama deseado, las casillas de primera y segunda vez se pueden poner sobre pautas distintas a la del extremo superior de la partitura.

Las instrucciones `\repeat` y otras relacionadas deben estar presentes en todos los pentagramas.

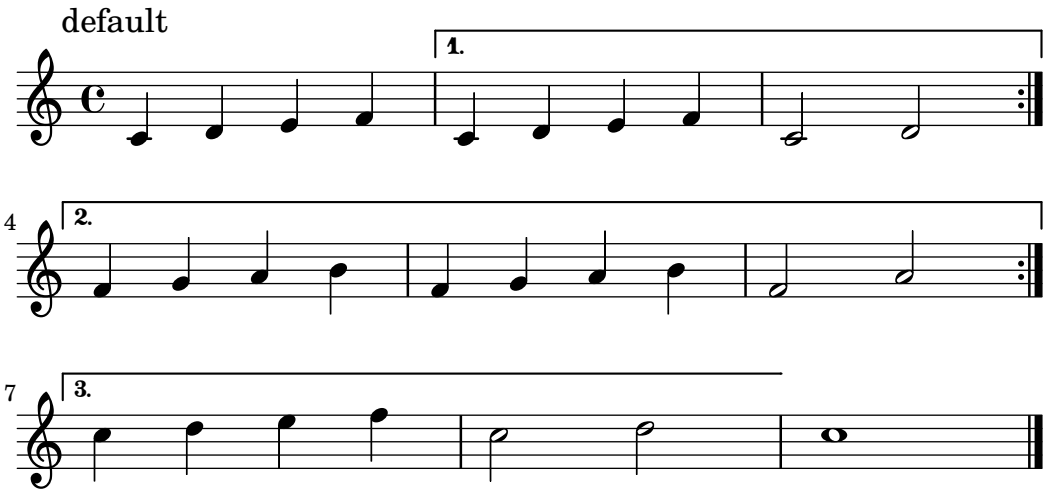
```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```

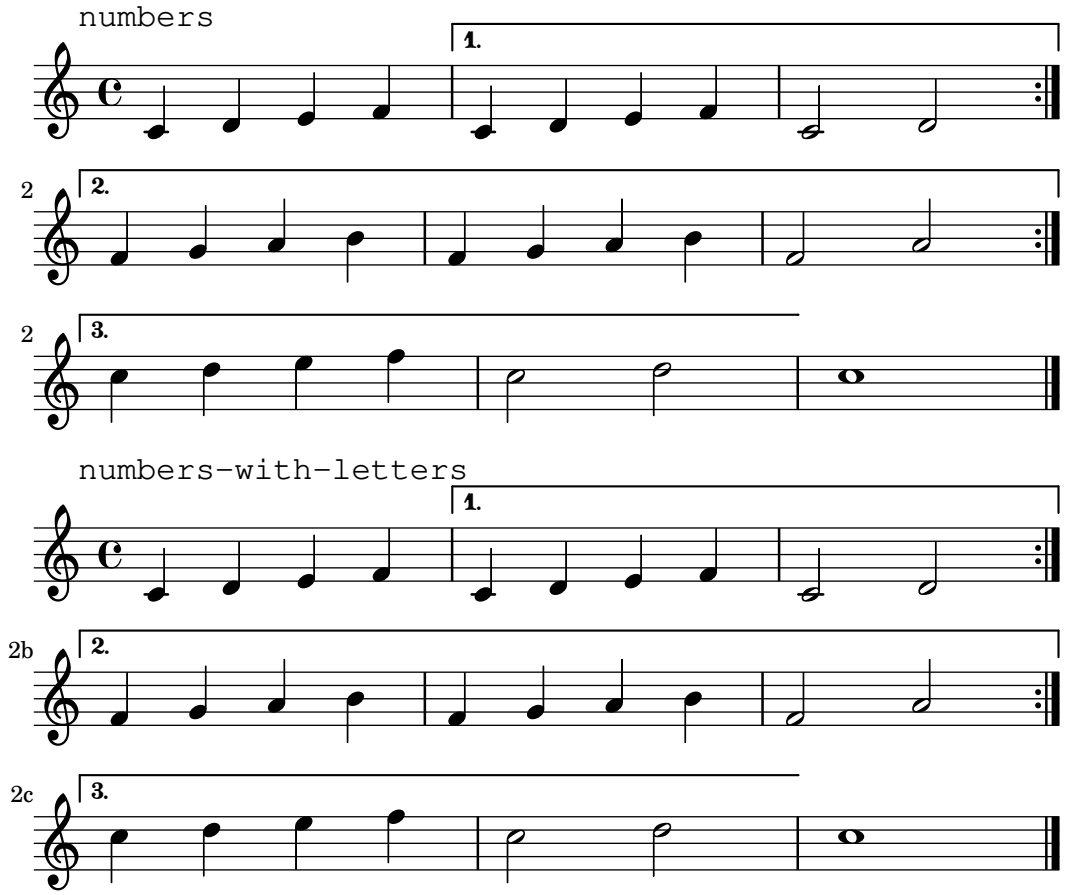


Numeración de compases alternativa

Fijando la propiedad de contexto `alternativeNumberingStyle`, se dispone de dos métodos alternativos adicionales para la numeración de compases dentro de las repeticiones.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|."
}

{
  \textMark \markup \large "default"
  \music
}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}
```





Véase también

Glosario musical: Sección “repeat” en *Glosario Musical*, Sección “volta” en *Glosario Musical*.

Referencia de la notación: Sección 2.5.2 [Automatic bar lines], página 130, Sección 2.5.1 [Barras de compás], página 120, Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752, Sección 36.11.1 [Modificación de ligaduras de unión y de expresión], página 808, Sección 2.6.3 [Gestión del tiempo], página 155.

Archivos instalados: `ly/engraver-init.ly`.

Fragmentos de código: Sección “Repeticiones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” en *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “UnfoldedRepeatedMusic” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Para las repeticiones en la forma `volta` (primera y segunda vez), los elementos extensos (ligaduras de expresión, etc.) que atraviesan los límites de las alternativas funcionan solo para la primera alternativa. De igual forma, no pueden abarcar desde el final de una alternativa hacia atrás hasta el comienzo de la sección que se repite.

La apariencia visual de una ligadura de unión o de expresión que continúa en alternativas subsecuentes se puede conseguir con la instrucción `\repeatTie` si la ligadura se extiende solo sobre una nota dentro del bloque alternativo, aunque este método no funciona en `TabStaff`; véase [Repeat tie], página 64. Otros métodos que se pueden confeccionar para indicar ligaduras que se continúan sobre varias notas dentro de los bloques alternativos, y que también funcionan en los contextos `TabStaff`, se muestran en Sección 36.11.1 [Modificación de ligaduras de unión y de expresión], página 808.

La apariencia visual de un glissando que continúa en las casillas de alternativa se puede conseguir como un glissando que comienza en una nota de adorno oculta. Véase [Extending glissandi across repeats], página 182.

Si una repetición que comienza en un compás incompleto tiene un bloque `\alternative` que contiene modificaciones a la propiedad `measureLength`, el uso de `\unfoldRepeats` dará lugar a líneas divisorias descolocadas y advertencias de comprobación de compás.

Una repetición anidada que usa la sintaxis antigua `\alternative` es ambigua:

```
\repeat ...
\repeat ...
\alternative
```

No está claro a qué `\repeat` pertenece la `\alternative`. Esta ambigüedad se resuelve haciendo que la `\alternative` pertenezca siempre a la `\repeat` más interna. Para más claridad, se recomienda usar llaves y poner el bloque `\alternative` dentro del cuerpo de su propia repetición.

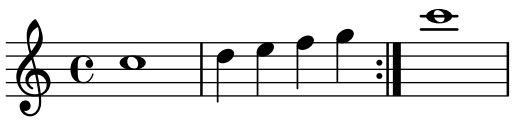
4.1.8 Marcas de repetición manual

> **Nota:** Estos métodos solo se utilizan para realizar construcciones de repetición poco usuales, y pueden tener un comportamiento distinto al esperado. En casi todas las situaciones, se deben crear las repeticiones utilizando la instrucción estándar `\repeat` o imprimiendo las barras de compás correspondientes. Para ver más información, consulte Sección 2.5.1 [Barras de compás], página 120.

Se puede usar la propiedad `repeatCommands` para controlar la disposición de las repeticiones. Su valor es una lista de Scheme. En general, cada elemento es a su vez una lista, `'(`*instrucción argumentos...*`)`, pero una lista sin argumentos se puede abreviar como un símbolo; p.ej., `'((start-repeat))` se puede escribir como `'(start-repeat)`.

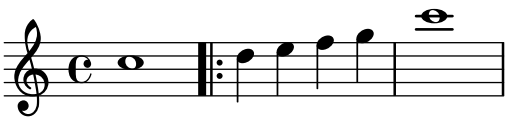
`end-repeat`
: Termina una sección repetida.

```
\relative {
  c''1
  d4 e f g
  \set Score.repeatCommands = #'(end-repeat)
  c1
}
```



`start-repeat`
: Inicia una sección repetida.

```
\relative {
  c''1
  \set Score.repeatCommands = #'(start-repeat)
  d4 e f g
  c1
}
```

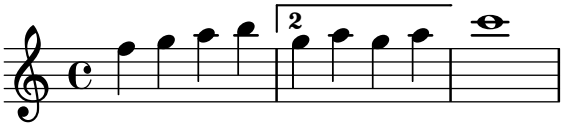


En la práctica habitual del grabado no se imprimen signos de repetición al principio de la pieza.

volta *texto*

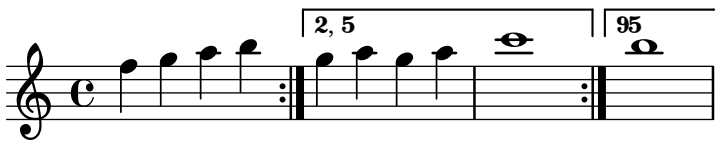
Si *texto* es un elemento de marcado, iniciar una casilla de alternativa con esa etiqueta; si *texto* es `#f`, terminar la casilla de alternativa. Una casilla de volta o alternativa que no se hace terminar explícitamente no llegará a imrpimirse.

```
\relative {
  f''4 g a b
  \set Score.repeatCommands =
    #`((volta ,#{ \markup \volta-number "2" #}))
  g4 a g a
  \set Score.repeatCommands = #'((volta #f))
  c1
}
```



Se pueden producir varias instrucciones de repetición en el mismo punto:

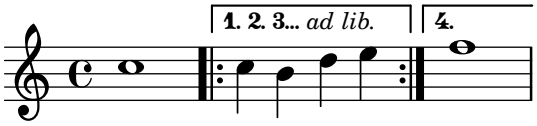
```
\relative {
  f''4 g a b
  \set Score.repeatCommands =
    #`((volta ,#{ \markup { \concat { \volta-number 2 , }
                            \volta-number 5 } #})
       end-repeat)
  g4 a g a
  c1
  \set Score.repeatCommands =
    #`((volta #f)
       (volta ,#{ \markup \volta-number 95 #})
       end-repeat)
  b1
  \set Score.repeatCommands = #'((volta #f))
}
```



Se puede incluir texto dentro de la casilla de primera y segunda vez. El texto puede ser un número o números, o un elemento de marcado, véase Sección 8.2 [Formatear el texto], página 323. La forma más fácil de usar texto de marcado es definir el marcado previamente, y luego incluirlo dentro de una lista de Scheme.

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... }
                       \italic { ad lib. } }
\relative {
```

```
  c''1
  \set Score.repeatCommands = #`((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands =
    #`((volta #f)
       (volta ,#{ \markup \volta-number "4." #})
       end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```

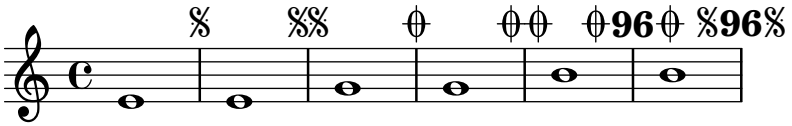


Se pueden crear llamadas de segno y coda con las instrucciones \segnoMark, \inStaffSegno y \codaMark. Esta es la sintaxis para las instrucciones de llamada:

```
\codaMark n
\segnoMark n
```

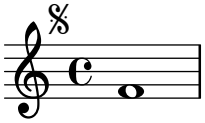
donde *n* es un número de secuencia, o \default para usar el siguiente automáticamente. Las llamadas de ensayo, de segno y de coda se numeran de manera independiente.

```
\fixed c' {
  e1
  \segnoMark \default
  e1
  \segnoMark \default
  g1
  \codaMark \default
  g1
  \codaMark \default
  b1
  \codaMark 96
  b1
  \segnoMark 96
}
```



Al comienzo de una pieza, las instrucciones \segnoMark \default y \codaMark \default no producen ningún símbolo de llamada. Especifique '1' para forzar una llamada.

```
\fixed c' {
  \segnoMark 1
  f1
}
```



La instrucción \inStaffSegno es equivalente a \segnoMark \default con el efecto añadido de establecer temporalmente la propiedad segnoStyle al valor bar-line para forzar que se imprima como una barra de compás.

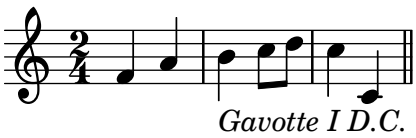
```
\fixed c' {
  e1
  \inStaffSegno
  g1
  \segnoMark \default
  b1
}
```



Para ver más información acerca de cómo modificar la aperiencia visual de las llamadas de segno y coda, véase Sección 4.1.7 [Apariencia de las repeticiones de Segno], página 201.

Para crear indicaciones de salto arbitrarias, use la instrucción \jump.

```
\fixed c' {
  \time 2/4
  f4 a
  b4 c'8 d'
  c'4 c
  \jump "Gavotte I D.C."
  \section
}
```



Véase también

Referencia de la notación: Sección 2.5.1 [Barras de compás], página 120, Sección 8.2 [Formatear el texto], página 323, Sección 2.5.5 [Llamadas de ensayo], página 142, Sección 4.1.7 [Apariencia de las repeticiones de Segno], página 201.

Fragmentos de código: Sección “Repeticiones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CodaMark” en *Referencia de Funcionamiento Interno*, Sección “Jump_engraver” en *Referencia de Funcionamiento Interno*, Sección “JumpScript” en *Referencia de Funcionamiento Interno*, Sección “Mark_engraver” en *Referencia de Funcionamiento Interno*, Sección “SegnoMark” en *Referencia de Funcionamiento Interno*, Sección “SegnoRepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “VoltaBracket” en *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” en *Referencia de Funcionamiento Interno*.

4.2 Repeticiones cortas

Esta sección trata de cómo introducir repeticiones cortas. Las repeticiones cortas pueden adoptar dos formas: barras inclinadas o signos de porcentaje si estamos representando repeticiones de una sola nota, un compás o dos compases, y trémolos en caso contrario.

4.2.1 Repeticiones de compás o parte de ellos

Los patrones cortos que se repiten se imprimen una sola vez, y el patrón repetido se sustituye por un símbolo especial.

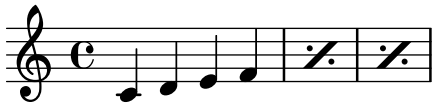
La sintaxis es:

```
\repeat percent número expresión_musical
```

donde *expresión_musical* es una expresión musical.

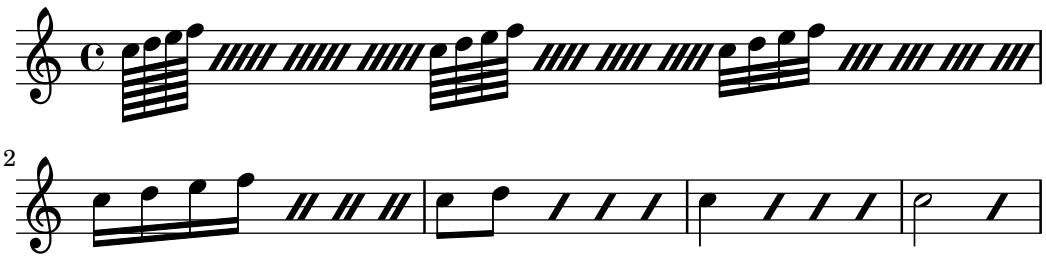
También existe una forma abreviada, \%, que puede convenir dentro de una pieza que use profusamente repeticiones de porcentaje:

```
\%3 { c'4 d' e' f' }
```



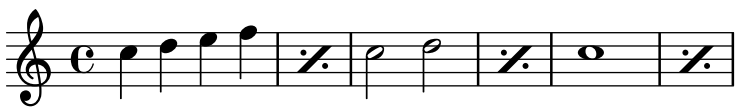
Los patrones más cortos que un compás se sustituyen por barras inclinadas.

```
\relative c'' {
  \repeat percent 4 { c128 d e f }
  \repeat percent 4 { c64 d e f }
  \repeat percent 5 { c32 d e f }
  \repeat percent 4 { c16 d e f }
  \repeat percent 4 { c8 d }
  \repeat percent 4 { c4 }
  \repeat percent 2 { c2 }
}
```

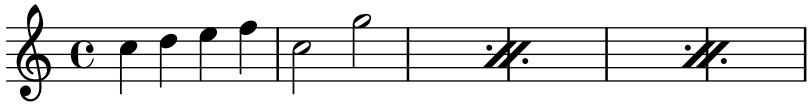


Los patrones de repetición de uno o dos compases se sustituyen por signos parecidos al símbolo de porcentaje.

```
\relative c'' {
  \repeat percent 2 { c4 d e f }
  \repeat percent 2 { c2 d }
  \repeat percent 2 { c1 }
}
```

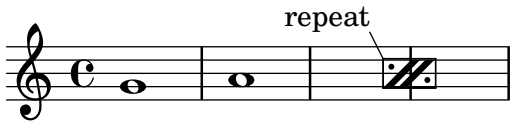


```
\relative {
  \repeat percent 3 { c''4 d e f | c2 g' }
}
```



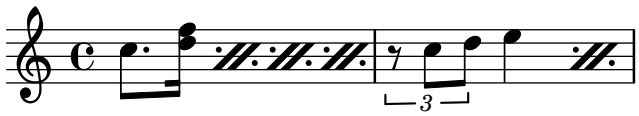
Observe que el tiempo de desplazamiento correcto para acceder al grob DoublePercentRepeat es el comienzo del segundo compás de la repetición, que se puede conseguir fácilmente con \after.

```
\new Voice \with { \consists Balloon_engraver }
  { \after 1*3
      \balloonGrobText DoublePercentRepeat #'(-1 . 2) "repeat"
    \repeat percent 2 { g'1 | a'1 } }
```



Los patrones más cortos que un compás pero que contienen duraciones mezcladas utilizan un símbolo de porcentaje doble.

```
\relative {
  \repeat percent 4 { c''8. <d f>16 }
  \repeat percent 2 { \tuplet 3/2 { r8 c d } e4 }
}
```

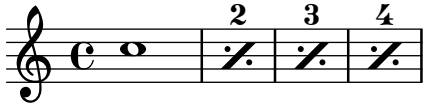


Fragmentos de código seleccionados

Contador de repeticiones de tipo porcentaje

Las repeticiones de compases completos de más de dos repeticiones pueden llevar impreso un contador si se ajusta el valor de la propiedad de contexto countPercentRepeats.

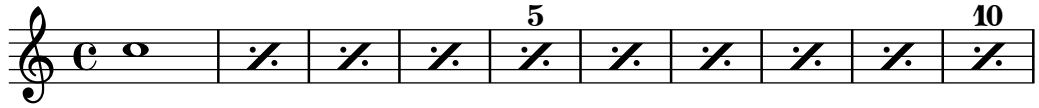
```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```

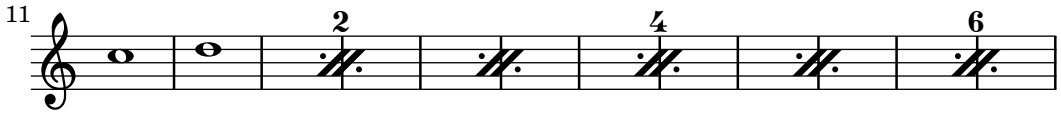


Visibilidad del contador de repeticiones de tipo porcentaje

Se pueden mostrar los contadores de las repeticiones del tipo porcentaje a intervalos regulares mediante el establecimiento de la propiedad de contexto repeatCountVisibility.

```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```

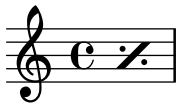




Símbolos de porcentaje sueltos

También se pueden imprimir símbolos de porcentaje sueltos.

```
makePercent =
#(define-music-function (note) (ly:music?)
   "Make a percent repeat the same length as NOTE."
   (make-music 'PercentEvent
               'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



Véase también

Glosario musical: Sección “percent repeat” en *Glosario Musical*, Sección “simile” en *Glosario Musical*.

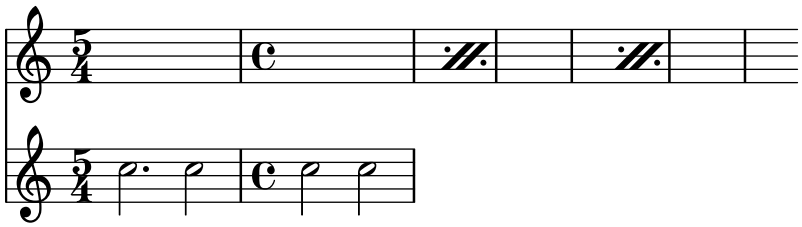
Fragmentos de código: Sección “Repeticiones” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RepeatSlash” en *Referencia de Funcionamiento Interno*, Sección “RepeatSlashEvent” en *Referencia de Funcionamiento Interno*, Sección “DoubleRepeatSlash” en *Referencia de Funcionamiento Interno*, Sección “PercentRepeat” en *Referencia de Funcionamiento Interno*, Sección “PercentRepeatCounter” en *Referencia de Funcionamiento Interno*, Sección “PercentRepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “Percent_repeat_engraver” en *Referencia de Funcionamiento Interno*, Sección “DoublePercentEvent” en *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeat” en *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeatCounter” en *Referencia de Funcionamiento Interno*, Sección “Double_percent_repeat_engraver” en *Referencia de Funcionamiento Interno*, Sección “Slash_repeat_engraver” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Las repeticiones de porcentaje no contienen nada más aparte del propio signo de porcentaje; especialmente, los cambios de indicación de compás no se repiten.

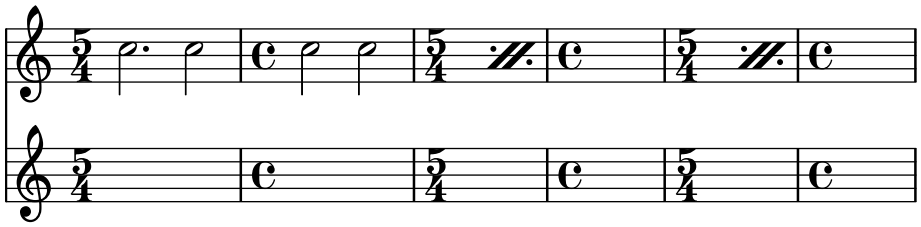
```
\repeat percent 3 { \time 5/4 c2. 2 \time 4/4 2 2 }
```



Cualquier cambio de compás o instrucción \partial se tiene que producir en pasajes paralelos *fuera* de la repetición de porcentaje, p. ej. en una pista especial para la gestión del compás.

```
<<
  \repeat percent 3 { c2. 2 2 2 }
```

```
  \repeat unfold 3 { \time 5/4 s4*5 \time 4/4 s1 }
>>
```

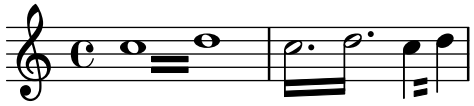


4.2.2 Repeticiones de trémolo

Los trémolos pueden adoptar dos formas: alternancia entre dos acordes o dos notas, y repetición rápida de una sola nota o acorde. Los trémolos que consisten en una alternancia se indican por medio de la adición de barras entre las notas o acordes que se alternan, mientras que los trémolos que consisten en la repetición rápida de una sola nota se indican mediante la adición de barras cruzadas a una nota única.

Para colocar marcas de trémolo entre las notas, use `\repeat` con el estilo `tremolo` (trémolo):

```
\relative c'' {
  \repeat tremolo 8 { c16 d }
  \repeat tremolo 6 { c16 d }
  \repeat tremolo 2 { c16 d }
}
```



La sintaxis de `\repeat tremolo` espera que se escriban exactamente dos notas dentro de las llaves, y el número de repeticiones debe corresponderse con un valor que se pueda expresar con figuras normales o con puntillo. Así, `\repeat tremolo 7` es válido y produce una nota con doble puntillo, pero `\repeat tremolo 9` no es válido.

La duración del trémolo es igual a la duración de la expresión entre llaves, multiplicada por el número de repeticiones: `\repeat tremolo 8 { c16 d16 }` da como resultado un trémolo de redonda, escrito como dos redondas unidas por barras de trémolo.

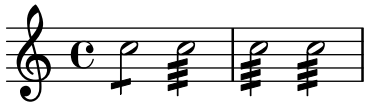
Existen dos maneras de colocar marcas de trémolo sobre una única nota. Incluso aquí se puede utilizar la sintaxis `\repeat tremolo`, en cuyo caso la nota no debe ir encerrada entre llaves:

```
\repeat tremolo 4 c'16
```



El mismo resultado se puede obtener escribiendo `:`*N* después de la nota, donde *N* indica la duración de la subdivisión (debe ser 8 como mínimo). Si *N* es 8, se añade una barra de corchea a la plica de la nota. Si *N* se omite, se utiliza el último valor:

```
\relative {
  c''2:8 c:32
  c: c:
}
```

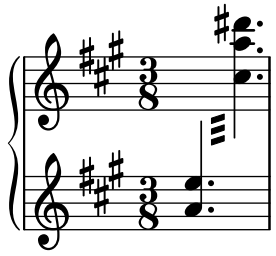


Fragmentos de código seleccionados

Trémolos de pentagrama cruzado

Dado que `\repeat tremolo` espera exactamente dos argumentos musicales para los trémolos de acorde, la nota o acorde que cambia de pentagrama en un trémolo que cruza el pentagrama se debe colocar dentro de llaves curvas junto a su instrucción `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```



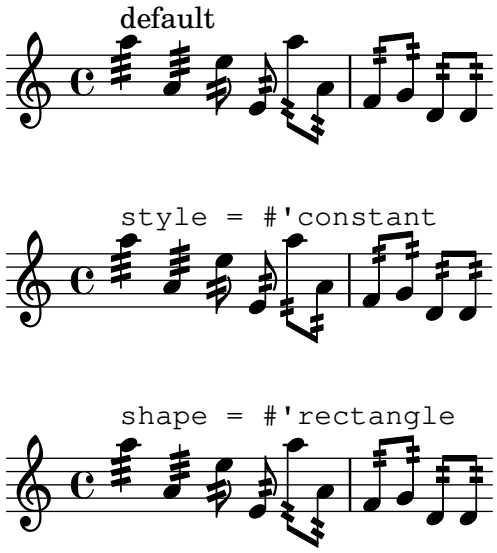
Controlar el aspecto de las barras de trémolo

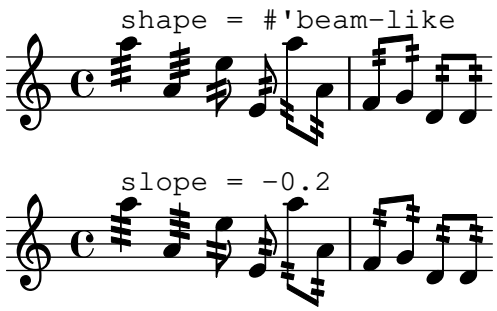
Usando diversas propiedades del objeto gráfico `StemTremolo` es posible controlar la apariencia de las barras de trémolo.

- La propiedad `slope` fija la inclinación de las barras de trémolo.
- La propiedad `shape` determina si las barras de trémolo tienen forma de rectángulo (valor `rectangle`) o de barras de corchea pequeñas (valor `beam-like`).
- La propiedad `style` fija la inclinación y la forma dependiendo de si la nota tiene corchete, barra o solo plica. Esto existe en contraste con las dos propiedades anteriores, que cambian la inclinación y la forma incondicionalmente. Hay dos estilos definidos.
  - `default`: las barras para los corchetes de plica hacia abajo son más largos y más inclinados que las barras para los corchetes de plica hacia arriba; las barras de trémolo sobre notas barradas tienen una forma rectangular y son paralelas a la barra de corchea.
  - `constant`: todas las barras de trémolo tienen forma de barra de corchea y tienen la misma inclinación excepto para los corchetes de plica hacia abajo.

```
music = {
```

```
  a''4:32 a':
  e''8: \noBeam e':
  a'':[ a':]
  f':[ g':]
  d':[ d':]
}

\new Staff {
  <>^\markup "default"
  \music
}

\new Staff {
  <>^\markup \typewriter "style = #'constant"
  \override StemTremolo.style = #'constant
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'rectangle"
  \override StemTremolo.shape = #'rectangle
  \music
}

\new Staff {
  <>^\markup \typewriter "shape = #'beam-like"
  \override StemTremolo.shape = #'beam-like
  \music
}

\new Staff {
  <>^\markup \typewriter "slope = -0.2"
  \override StemTremolo.slope = -0.2
  \music
}
```





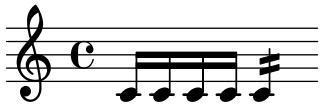
Véase también

Fragmentos de código: Sección “Repeticiones” en *Fragmentos de código*.

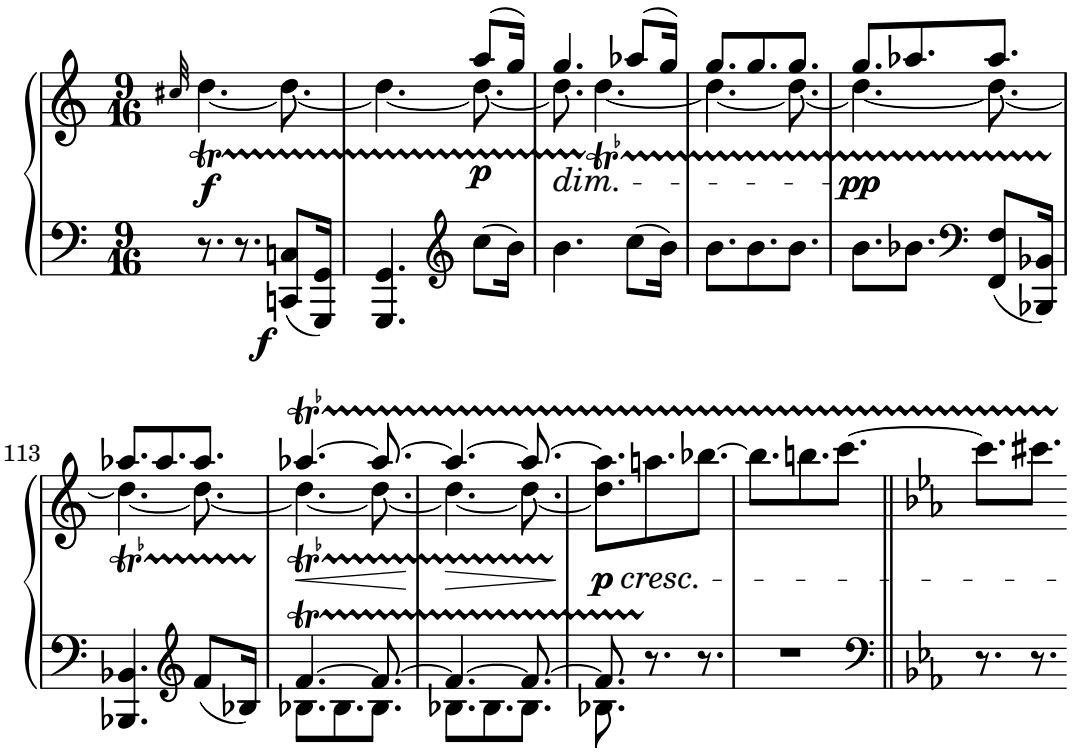
Advertencias y problemas conocidos

Aunque la sintaxis de `\repeat tremolo` se despliega por medio de `\unfoldRepeats`, su abreviatura correspondiente no lo hace (el seguimiento de este problema está en el bug número 6145 (`https://gitlab.com/lilypond/lilypond/-/issues/6145`)):

```
{
  \unfoldRepeats \repeat tremolo 4 c'16
  \unfoldRepeats c'4:16
}
```



5 Notas simultáneas



La palabra 'polifonía' en música hace referencia a tener más de una voz en una pieza cualquiera de música. En LilyPond la polifonía hace referencia a tener más de una voz en el mismo pentagrama.

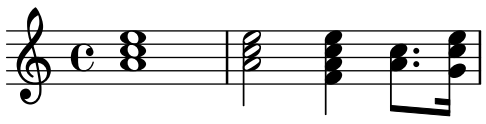
5.1 Una voz única

Esta sección trata de la notas simultáneas dentro de la misma voz.

5.1.1 Notas en acorde

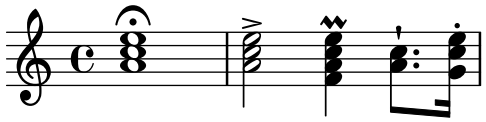
Un acorde se forma encerrando un conjunto de notas entre < y >. Un acorde puede ir seguido de una duración, como si fueran simples notas.

```
\relative {
  <a' c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16
}
```



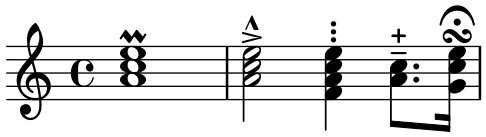
Los acordes también pueden ir seguidos de articulaciones, de nuevo como si fueran simples notas.

```
\relative {
  <a' c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^! <g c e>16-.
}
```



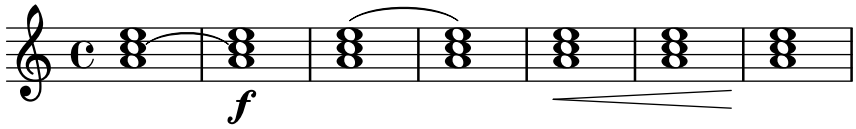
Las notas dentro del propio acorde también pueden ir seguidas de articulaciones y ornamentos.

```
\relative {
  <a' c\prall e>1 <a-> c-^ e>2 <f-. a c-. e-.>4
  <a-+ c-->8. <g\fermata c e\turn>16
}
```



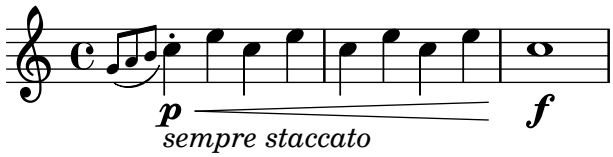
Sin embargo, algunos elementos de notación tales como las expresiones de matices dinámicos y los reguladores se deben unir al acorde y no a las notas que integran el mismo, pues en caso contrario no se imprimirán. Otros elementos de notación como las digitaciones y las ligaduras de expresión se posicionan de forma muy diferente cuando se adjuntan a las notas de un acorde en lugar de hacerlo a redondas o notas sueltas.

```
\relative {
  <a'\f c( e>1 <a c) e>\f <a\< c e>( <a\! c e>)
  <a c e>\< <a c e> <a c e>\!
}
```



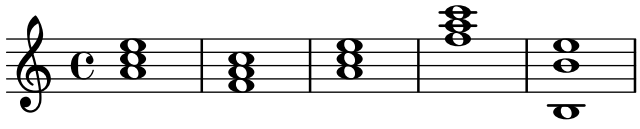
Un acorde funciona como un mero contenedor para las notas que lo componen y sus articulaciones y otros elementos adjuntos. Como consecuencia, un acorde sin ninguna nota en su interior no tiene realmente ninguna duración. Cualqier articulación adjunta se producirá en el mismo momento musical que la nota o acorde siguiente y se puede combinar con ellos (para ver posibilidades más complejas de combinaciones de estos elementos, consulte Sección 5.1.3 [Expresiones simultáneas], página 222):

```
\relative {
  \grace { g'8( a b }
  <> ) \p \< -. -\markup \italic "sempre staccato"
  \repeat unfold 4 { c4 e }  c1\f
}
```



Se puede usar el modo relativo para la altura de las notas de los acordes. La primera nota del acorde siempre es relativa a la primera nota del acorde anterior, o en caso de que el elemento precedente no sea un acorde, la altura de la última nota que vino antes del acorde. El resto de las notas del acorde son relativas a la nota anterior *dentro del mismo acorde*.

```
\relative {
  <a' c e>1 <f a c> <a c e> <f' a c> <b, e b,>
}
```



Para ver más información sobre los acordes, consulte Capítulo 15 [Notación de acordes], página 510.

Véase también

Glosario musical: Sección “chord” en *Glosario Musical*.

Manual de aprendizaje: Sección “Combinar notas para formar acordes” en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 15 [Notación de acordes], página 510, Sección 3.1.1 [Articulaciones y ornamentos], página 156, Sección 1.1.2 [Escritura de octava relativa], página 4, Sección 5.2 [Varias voces], página 224.

Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

Advertencias y problemas conocidos

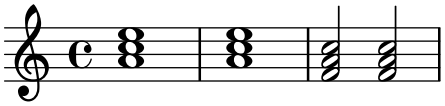
Los acordes que contienen más de dos alturas en el mismo espacio de pentagrama, como ‘`<e f! fis!>`’, crean notas cuyas cabezas se superponen. Dependiendo de la situación, una mejor representación puede requerir:

- el uso temporal de varias voces, véase Sección 5.2 [Varias voces], página 224, ‘`<< f! \\ <e fis!> >>`’,
- la transcripción enarmónica de una o más notas, ‘`<e f ges>`’, or
- Culsters o racimos; véase Sección 5.1.4 [Racimos (clusters)], página 223.

5.1.2 Repetición de acordes

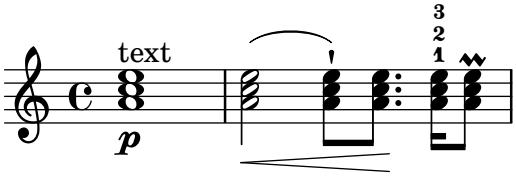
Para reducir el tecleo, se puede usar una abreviatura de repetición del acorde anterior. El símbolo de repetición de acordes es q:

```
\relative {
  <a' c e>1 q <f a c>2 q
}
```



Como en los acordes normales, el símbolo de repetición de acordes puede usarse con duraciones, articulaciones, elementos de marcado, ligaduras de expresión, barras, etc., pues sólo se duplican las notas del acorde precedente.

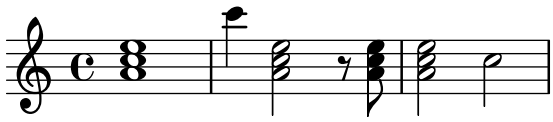
```
\relative {
  <a' c e>1\p^"text" q2\<( q8)[-! q8.]\! q16-1-2-3 q8\prall
}
```



El símbolo de repetición de acordes siempre recuerda la última ocurrencia de un acorde, por lo que es posible repetir el acorde más reciente incluso si se han escrito en medio otras notas que no están en un acorde, o silencios.

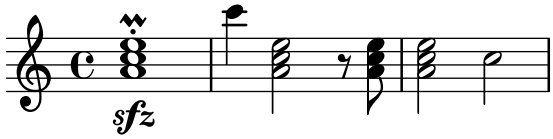
```
\relative {
  <a' c e>1 c'4 q2 r8 q8 |
  q2 c, |
```

```
}
```



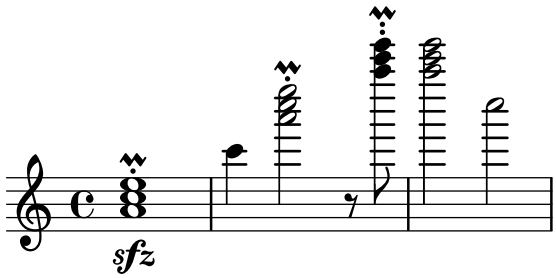
Sin embargo, el símbolo de repetición de acordes no retiene los matices dinámicos, las articulaciones ni los ornamentos que contiene o que están adosados al acorde anterior.

```
\relative {
  <a'-. c\prall e>1\sfz c'4 q2 r8 q8 |
  q2 c, |
}
```



Para poder retener algunos de ellos, se puede llamar explícitamente a la función `\chordRepeats` con un argumento adicional que especifique una lista de *tipos de evento* que conservar, a no ser que los eventos de ese tipo ya estén presentes en el propio acorde `q`.

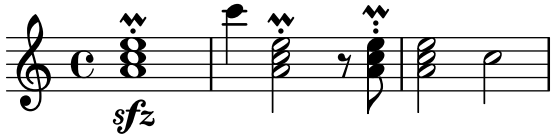
```
\relative {
  \chordRepeats #'(articulation-event)
  { <a'-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c, |
}
```



Aquí, el uso de `\chordRepeats` dentro de una construcción `\relative` produce un resultado inseperado: una vez se han expandido los eventos de acorde, no pueden distinguirse de aquellos introducidos como acordes normales, haciendo que `\relative` asigne una octava basada en su contexto actual.

Dado que las instancias anidadas de `\relative` no se afectan mutuamente, se puede usar otra instrucción `\relative` dentro de `\chordRepeats` para establecer las relaciones de octava antes de expandir los acordes repetidos. En tal caso, todo el contenido de la instrucción `\relative` interior no afecta a la exterior; de aquí la distinta escritura de octava en la última nota de este ejemplo.

```
\relative {
  \chordRepeats #'(articulation-event)
  \relative
  { <a'-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c'' |
}
```



Las interacciones con `\relative` se producen solamente con las llamadas explícitas de `\chordRepeats`: la expansión implícita al comienzo del proceso de tipografiado se hace en un momento en que todas las instancias de `\relative` ya se han procesado.

Véase también

Referencia de la notación: Capítulo 15 [Notación de acordes], página 510, Sección 3.1.1 [Articulaciones y ornamentos], página 156.

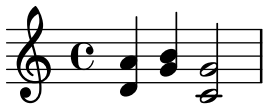
Archivos de inicio: `ly/chord-repetition-init.ly`.

5.1.3 Expresiones simultáneas

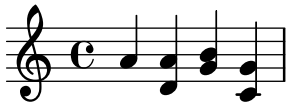
Una o más expresiones musicales encerradas entre ángulos dobles se entienden como simultáneas. Si la primera expresión comienza con una sola nota o si toda la expresión simultánea aparece explícitamente dentro de una sola voz, la expresión completa se sitúa sobre un solo pentagrama; en caso contrario los elementos de la expresión simultánea se sitúan en pentagramas distintos.

Los ejemplos siguientes muestran expresiones simultáneas sobre un solo pentagrama:

```
\new Voice {  % explicit single voice
  << \relative { a'4 b g2 }
     \relative { d'4 g c,2 } >>
}
```

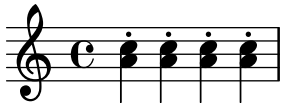


```
\relative {
  % single first note
  a' << \relative { a'4 b g }
        \relative { d'4 g c, } >>
}
```



Esto puede ser de utilidad si las secciones simultáneas tienen idénticas duraciones, pero se producirán errores si se intentan poner notas de distinta duración sobre la misma plica. Las notas, articulaciones y cambios de propiedades que están dentro de *un solo* contexto 'Voice' se recolectan y se representan en el orden musical:

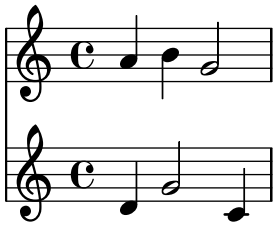
```
\relative {
  <a' c>4-.  <>-. << c a >>  << { c-. <c a> } { a s-. } >>
}
```



Tratar de poner más de una plica o barra de corchea, o distintas duraciones o propiedades en el mismo momento musical, requiere el uso de más de una voz.

El ejemplo siguiente muestra cómo las expresiones simultáneas pueden generar varios pentagramas de forma implícita:

```
% no single first note
<< \relative { a'4 b g2 }
   \relative { d'4 g2 c,4 } >>
```



Aquí no hay problema en tener distintas duraciones porque se interpretan en voces distintas.

Advertencias y problemas conocidos

Si hay notas que proceden de dos o más voces, sin especificar ningún desplazamiento horizontal, y tienen las plicas en la misma dirección, aparece el mensaje

```
Advertencia: esta voz requiere un ajuste de \voiceXx o \shiftXx
```

durante la compilación. Este mensaje se puede suprimir mediante

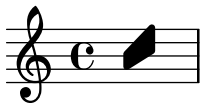
```
\override NoteColumn.ignore-collision = ##t
```

Sin embargo, esto no solamente suprime la advertencia sino que puede evitar cualquier resolución de colisiones y puede tener otros efectos no deseados (consulte también *Problemas conocidos* en Sección 5.2.3 [Resolución de las colisiones], página 229).

5.1.4 Racimos (clusters)

Un «cluster» o racimo indica que se deben tocar simultáneamente un conjunto de notas consecutivas. Se escriben aplicando la función `\makeClusters` a una secuencia de acordes, p.ej.:

```
\relative \makeClusters { <g' b>2 <c g'> }
```

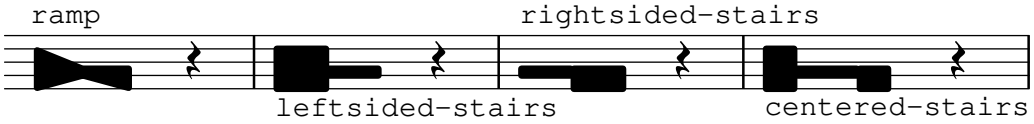


Se pueden mezclar en el mismo pentagrama notas normales y clusters, incluso al mismo tiempo. En tal caso, no se hace ningún intento de evitar automáticamente las colisiones entre clusters y notas normales.

Están contemplados los siguientes estilos de cluster: `ramp`, `leftsided-stairs`, `rightsided-stairs` y `centered-stairs`.

```
fragment = { <e' d''>4 <g' a'> <e' a'> r }

{
  \omit Staff.Clef
  \omit Staff.TimeSignature

  <>^\markup \typewriter "ramp"
  \override ClusterSpanner.style = #'ramp
  \makeClusters \fragment

  <>_\markup \typewriter "leftsided-stairs"
  \override ClusterSpanner.style = #'leftsided-stairs
```

```
  \makeClusters \fragment

  <>^\markup \typewriter "rightsided-stairs"
  \override ClusterSpanner.style = #'rightsided-stairs
  \makeClusters \fragment

  <>_\markup \typewriter "centered-stairs"
  \override ClusterSpanner.style = #'centered-stairs
  \makeClusters \fragment
}
```



Véase también

Glosario musical: Sección “cluster” en *Glosario Musical*.

Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ClusterSpanner” en *Referencia de Funcionamiento Interno*, Sección “ClusterSpannerBeacon” en *Referencia de Funcionamiento Interno*, Sección “Cluster_spanner_engraver” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los clusters sólo tienen un buen aspecto cuando abarcan un mínimo de dos acordes. En caso contrario aparecerán excesivamente estrechos.

Los clusters no llevan plica y por sí mismos no pueden indicar las duraciones, pero la longitud del cluster que se imprime viene determinada por la duración de los acordes que lo definen. Los racimos separados necesitan silencios de separación entre ellos.

Los clusters no producen ninguna salida MIDI.

5.2 Varias voces

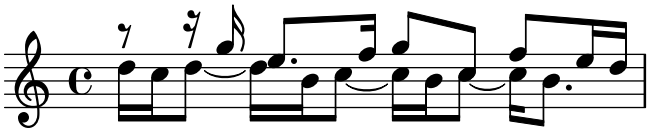
Esta sección trata las notas simultáneas en varias voces o varios pentagramas.

5.2.1 Polifonía en un solo pentagrama

Instanciar las voces explícitamente

La estructura básica necesaria para obtener varias voces independientes en un mismo pentagrama se ilustra en el ejemplo siguiente:

```
\new Staff <<
  \new Voice = "first"
    \relative { \voiceOne r8 r16 g'' e8. f16 g8[ c,] f e16 d }
  \new Voice= "second"
    \relative { \voiceTwo d''16 c d8~ 16 b c8~ 16 b c8~ 16 b8. }
>>
```



Aquí se crean explícitamente instancias de voces, cada una de las cuales recibe un nombre. Las instrucciones `\voiceOne ... \voiceFour` (voz uno hasta voz cuatro) preparan las voces de manera que la primera y segunda voces llevan las plicas hacia arriba, las voces segunda y cuarta llevan las plicas hacia abajo, las cabezas de las notas en las voces tercera y cuarta se desplazan horizontalmente, y los silencios de las voces respectivas se desplazan también automáticamente para evitar las colisiones. La instrucción `\oneVoice` (una voz) devuelve todos los ajustes de las voces al estado neutro predeterminado.

Observe que `Voice` es un contexto del nivel inferior (véase Sección 33.1.5 [Contextos del nivel más bajo. Voces], página 744). Dentro de `TabStaff` se debe usar en su lugar `TabVoice`.

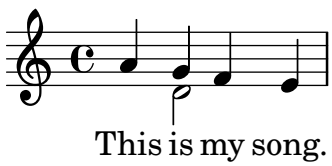
Pasajes polifónicos temporales

Se puede crear un pasaje polifónico temporal con la construcción siguiente:

```
<< { \voiceOne ... }
  \new Voice { \voiceTwo ... }
>> \oneVoice
```

Aquí, la primera expresión dentro de un pasaje polifónico temporal se coloca en el contexto `Voice` que estaba en uso inmediatamente antes del pasaje polifónico, y ese mismo contexto `Voice` continua después de la sección temporal. Otras expresiones dentro de los ángulos se asignan a distintas voces temporales. Esto permite asignar la letra de forma continua a una voz antes, durante y después de la sección polifónica:

```
\relative <<
  \new Voice = "melody" {
    a'4
    <<
      {
        \voiceOne
        g f
      }
      \new Voice {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }
  \new Lyrics \lyricsto "melody" {
  This is my song.
  }
>>
```

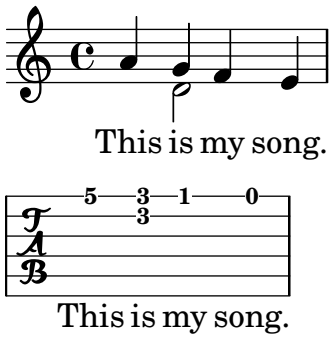


Aquí, las instrucciones `\voiceOne` y `\voiceTwo` son necesarias para definir los ajustes de cada voz.

In `TabStaff` one needs to use `TabVoice`.

Si la misma música debe aparecer en `Staff` *y* en `TabStaff` se puede usar el contexto general `Bottom` (véase Sección 33.1.5 [Contextos del nivel más bajo. Voces], página 744).

```
mus =
\relative
  \new Bottom = "melody" {
    a'4
    <<
      {
        \voiceOne
        g f
      }
      \new Bottom {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }

<<
  \new Staff \mus
  \new Lyrics \lyricsto "melody" {
    This is my song.
  }
>>

<<
  \new TabStaff \mus
  \new Lyrics \lyricsto "melody" {
    This is my song.
  }
>>
```



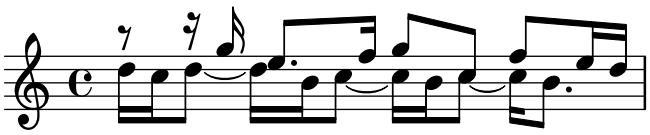
La construcción de la doble barra invertida

La construcción << {...} \\ {...} >>, en que las dos expresiones (o más) están separadas por doble barra invertida, se comporta de forma distinta a la construcción similar sin las dobles barras: *todas* las expresiones dentro de esta construcción se asignan a contextos `Bottom` nuevos del tipo actual, normalmente `Voice` o `TabVoice` (véase Sección 33.1.5 [Contextos del nivel más bajo. Voces], página 744). Estos contextos `Bottom` nuevos se crean implícitamente y reciben los nombres fijos `"1"`, `"2"`, etc.

El primer ejemplo podría haberse tipografiado de la manera siguiente:

```
<<
```

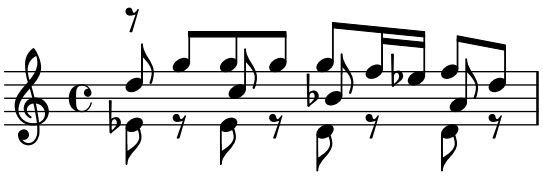
```
  \relative { r8 r16 g'' e8. f16 g8[ c,] f e16 d }
  \\
  \relative { d''16 c d8~ 16 b c8~ 16 b c8~ 16 b8. }
>>
```



Esta sintaxis se puede usar siempre que no nos importe que se creen nuevas voces temporales que después serán descartadas. Estas voces creadas implícitamente reciben ajustes equivalentes al efecto de las instrucciones `\voiceOne ... \voiceFour`, en el orden en que aparecen en el código.

En el siguiente ejemplo, la voz intermedia lleva las plicas hacia arriba, de manera que la introducimos en tercer lugar para que pase a ser la voz tres, que tiene las plicas hacia arriba tal y como queremos. Se usan silencios de separación para evitar la aparición de silencios duplicados.

```
<<
  \relative { r8 g'' g  g g f16 ees f8 d }
  \\
  \relative { ees'8 r ees r d r d r }
  \\
  \relative { d''8 s c s bes s a s }
>>
```



En todas las partituras excepto las más simples, se recomienda crear contextos `Voice` explícitos como aparece explicado en Sección “Contextos y grabadores” en *Manual de Aprendizaje* y en Sección “Voces explícitas” en *Manual de Aprendizaje*.

Orden de las voces

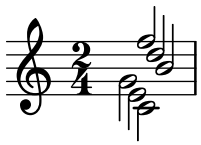
Al escribir varias voces en el archivo de entrada, utilice el orden siguiente:

```
Voz 1: las más aguda
Voz 2: la más grave
Voz 3: la segunda más aguda
Voz 4: la segunda más grave
Voz 5: la tercera más aguda
Voz 6: la tercera más grave
etc.
```

A pesar de que esto puede parecer contrario a la intuición, simplifica el proceso de disposición automática de las figuras. Observe que las voces de numeración impar reciben plicas hacia arriba, y las de numeración par reciben plicas hacia abajo:

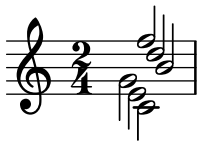
```
\new Staff <<
  \time 2/4
  { f''2 }  % 1: highest
  \\
  { c'2  }  % 2: lowest
  \\
```

```
  { d''2 }  % 3: second-highest
  \\
  { e'2  }  % 4: second-lowest
  \\
  { b'2  }  % 5: third-highest
  \\
  { g'2  }  % 6: third-lowest
>>
```



Si se desea introducir las voces en un orden distinto, puede ser de utilidad la instrucción `\voices`:

```
\new Staff \voices 1,3,5,6,4,2 <<
  \time 2/4
  { f''2 }  % 1: highest
  \\
  { d''2 }  % 3: second-highest
  \\
  { b'2  }  % 5: third-highest
  \\
  { g'2  }  % 6: third-lowest
  \\
  { e'2  }  % 4: second-lowest
  \\
  { c'2  }  % 2: lowest
>>
```

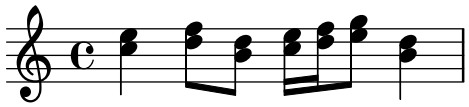


Nota: No se pueden crear letras ni elementos de extensión (como ligaduras, reguladores, etc.) que se crucen de una voz a otra.

Duraciones idénticas

En el caso especial en que queremos tipografiar fragmentos de música que discurre en paralelo y con las mismas duraciones, se pueden combinar en un solo contexto de voz, formando así acordes. Para conseguirlo, las incorporamos dentro de una construcción de música simultánea, dentro de una voz creada explícitamente:

```
\new Voice <<
  \relative { e''4 f8 d e16 f g8 d4 }
  \relative { c''4 d8 b c16 d e8 b4 }
>>
```



Este método conduce a barrados extraños y advertencias si los fragmentos de música no tienen las mismas duraciones exactas.

Instrucciones predefinidas

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

Véase también

Manual de aprendizaje: Sección “Las voces contienen música” en *Manual de Aprendizaje*, Sección “Voces explícitas” en *Manual de Aprendizaje*.

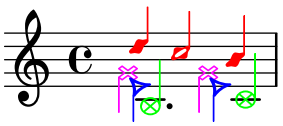
Referencia de la notación: Sección 13.1.5 [Pautas de percusión], página 492, Sección 2.2.2 [Silencios invisibles], página 70, Sección 7.1.9 [Plicas], página 300.

Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

5.2.2 Estilos de voz

Se pueden aplicar colores y formas distintos a las voces para permitir identificarlas fácilmente:

```
<<
  \relative { \voiceOneStyle d''4 c2 b4 }
  \\
  \relative { \voiceTwoStyle e'2 e }
  \\
  \relative { \voiceThreeStyle b2. c4 }
  \\
  \relative { \voiceFourStyle g'2 g }
>>
```



Para recuperar la presentación normal se utiliza la instrucción `\voiceNeutralStyle`.

Instrucciones predefinidas

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`, `\voiceNeutralStyle`.

Véase también

Manual de aprendizaje: Sección “Oigo voces” en *Manual de Aprendizaje*, Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

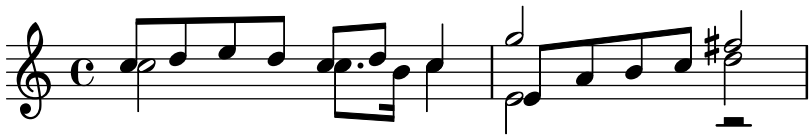
Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

5.2.3 Resolución de las colisiones

Las cabezas de notas que están en diferentes voces y tienen la misma altura, la misma forma de cabeza, y dirección opuesta de la plica, se combinan automáticamente, pero las que tienen cabezas distintas o la misma dirección de la plica no se combinan. Los silencios que se encuentran en el lado opuesto de una plica en otra voz se desplazan verticalmente. El ejemplo siguiente muestra tres circunstancias distintas, sobre los pulsos 1 y 3 en el primer compás y sobre el pulso 1 del segundo compás, donde la combinación automática falla.

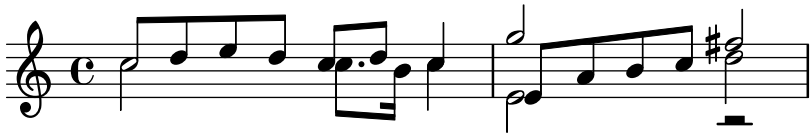
```
<<
  \relative {
    c''8 d e d c d c4
```

```
    g'2 fis
  } \\
  \relative {
    c''2 c8. b16 c4
    e,2 r
  } \\
  \relative {
    \oneVoice
    s1
    e'8 a b c d2
  }
>>
```



Las cabezas de notas diferentes se pueden combinar como se muestra más abajo. En este ejemplo las cabezas del pulso 1 del primer compás sí se combinan:

```
<<
  \relative {
    \mergeDifferentlyHeadedOn
    c''8 d e d c d c4
    g'2 fis
  } \\
  \relative {
    c''2 c8. b16 c4
    e,2 r
  } \\
  \relative {
    \oneVoice
    s1
    e'8 a b c d2
  }
>>
```

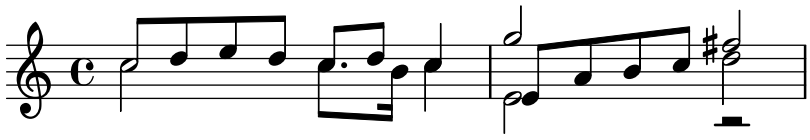


Las figuras negras y blancas no se combinan de esta manera, pues sería difícil poder distinguirlas.

También se pueden combinar cabezas con puntillos diferentes, como se muestra en el tercer pulso del primer compás:

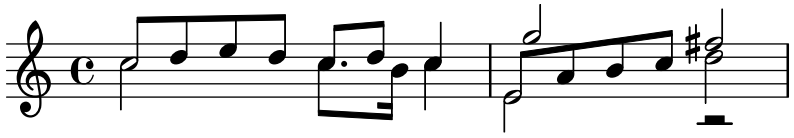
```
<<
  \relative {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c''8 d e d c d c4
    g'2 fis
  } \\
```

```
  \relative {
    c''2 c8. b16 c4
    e,2 r
  } \\
  \relative {
    \oneVoice
    s1
    e'8 a b c d2
  }
>>
```



La blanca y la corchea en el comienzo del segundo compás no están correctamente combinadas porque la combinación automática no es capaz de completarse satisfactoriamente cuando se encuentran tres o más notas alineadas en la misma columna de notas, y en este caso la cabeza combinada es incorrecta. Para conseguir que la combinación seleccione la cabeza correcta, se debe aplicar un `\shift` o desplazamiento a la nota que no se debe combinar. Aquí, se aplica `\shiftOn` para mover el sol agudo fuera de la columna, y entonces `\mergeDifferentlyHeadedOn` hace su trabajo correctamente.

```
<<
  \relative {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c''8 d e d c d c4
    \shiftOn
    g'2 fis
  } \\
  \relative {
    c''2 c8. b16 c4
    e,2 r
  } \\
  \relative {
    \oneVoice
    s1
    e'8 a b c d2
  }
>>
```



La instrucción `\shiftOn` permite (aunque no fuerza) que las notas de una voz se puedan desplazar. Cuando`\shiftOn` se aplica a una voz, una nota o acorde de esta voz se desplaza solamente si su plica chocase en caso contrario con una plica de otra voz, y sólo si las plicas en colisión apuntan en la misma dirección. La instrucción `\shiftOff` evita la posibilidad de este tipo de desplazamiento.

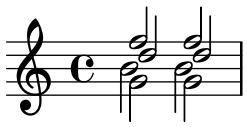
De forma predeterminada, las voces externas (normalmente las voces uno y dos) tienen `\shiftOff` especificado (desplazamiento desactivado), mientras que las voces interiores (tres

y siguientes) tienen especificado \shiftOn (desplazamiento activado). Cuando se aplica un desplazamiento, las voces con plicas hacia arriba (voces de numeración impar) se desplazan hacia la derecha, y las voces con las plicas hacia abajo (voces con numeración par) se desplazarn a la izquierda.

He aquí un ejemplo que le ayudará a visualizar la forma en que se expanidría internamente una expresión polifónica.

> **Nota:** Observe que con tres o más voces, el orden vertical de las mismas dentro de su archivo de entrada ¡no sería el mismo que el orden vertical de las voces en el pentagrama!

```
\new Staff \relative {
  %% abbreviated entry
  <<
    { f''2  }  % 1: highest
    \\
    { g,2 }  % 2: lowest
    \\
    { d'2 }  % 3: upper middle
    \\
    { b2  }  % 4: lower middle
  >>
  %% internal expansion of the above
  <<
    \new Voice = "1" { \voiceOne   \shiftOff f'2 }
    \new Voice = "2" { \voiceTwo   \shiftOff g,2 }
    \new Voice = "3" { \voiceThree \shiftOn  d'2 } % shifts right
    \new Voice = "4" { \voiceFour  \shiftOn  b2  } % shifts left
  >>
}
```



Dos instrucciones adicionales, \shiftOnn y \shiftOnnn, ofrecen niveles de desplazamiento mayores que se pueden especificar temporalmente para resolver colisiones en situaciones complejas; véase Sección “Ejemplos reales de música” en *Manual de Aprendizaje*.

Sólo se combinan las notas si tienen la plica en direcciones opuestas (como la tienen, por ejemplo, en las voces uno y dos de forma predeterminada o cuando las plicas se establecen explícitamente en direcciones opuestas).

Instrucciones predefinidas

\mergeDifferentlyDottedOn, \mergeDifferentlyDottedOff, \mergeDifferentlyHeadedOn, \mergeDifferentlyHeadedOff.

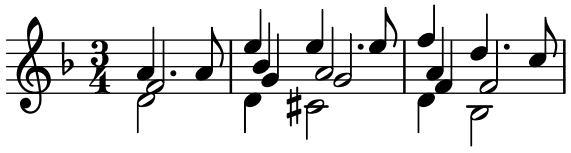
\shiftOn, \shiftOnn, \shiftOnnn, \shiftOff.

Fragmentos de código seleccionados

Voces adicionales para evitar colisiones

En ciertos casos de polifonía compleja, se necesitan voces adicionales para evitar colisiones entre las notas. Si se necesitan más de cuatro voces paralelas, las voces adicionales se añaden definiendo una variable que utiliza la función de Scheme `context-spec-music`.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice  {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }
    \new Voice {
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
  >>
}
```

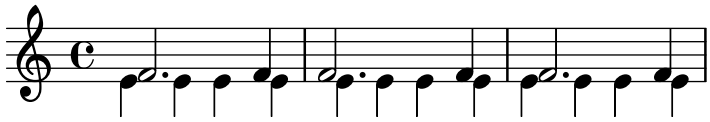


Desplazar las notas con puntillo en polifonía

Cuando se puede una nota en la voz superior para evitar la colisión con una nota de otra voz, el comportamiento predeterminado es desplazar la nota superior a la derecha. Se puede cambiar modificando la propiedad `prefer-dotted-right` del objeto gráfico `NoteCollision`.

```
\new Staff \relative c' <<
  {
```

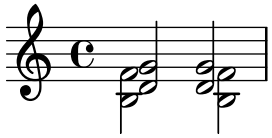
```
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```



Forzar el desplazamiento horizontal de las notas

Cuando el motor de tipografiado no es capaz de todo, se puede usar la sintaxis siguiente para sobreescribir las decisiones de tipografía. Las unidades de medida que se usan aquí son espacios de pentagrama.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = 1.7
    <b f'>2
  }
>>
```



Véase también

Glosario musical: Sección “polyphony” en *Glosario Musical*.

Manual de aprendizaje: Sección “Varias notas a la vez” en *Manual de Aprendizaje*, Sección “Las voces contienen música” en *Manual de Aprendizaje*, Sección “Ejemplos reales de música” en *Manual de Aprendizaje*.

Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

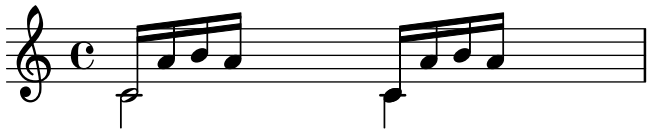
Referencia de funcionamiento interno: Sección “NoteColumn” en *Referencia de Funcionamiento Interno*, Sección “NoteCollision” en *Referencia de Funcionamiento Interno*, Sección “RestCollision” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El uso de `\override NoteColumn.ignore-collision = ##t` hace que las notas con distinta cabeza en voces diferentes se mezclen incorrectamente.

```
\mergeDifferentlyHeadedOn
<< \relative { c'16 a' b a } \\ \relative { c'2 } >>
\override NoteColumn.ignore-collision = ##t
```

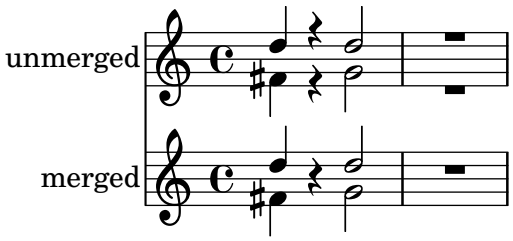
```
<< \relative { c'16 a' b a } \\ \relative { c'2 } >>
```



5.2.4 Combinación de silencios

Al usar más de una voz, es frecuente combinar los silencios que aparecen en las dos partes. Esto puede hacerse mediante el uso del grabador `Merge_rests_engraver`.

```
voiceA = \relative { d''4 r d2 | R1 | }
voiceB = \relative { fis'4 r g2 | R1 | }
\score {
  <<
    \new Staff \with {
      instrumentName = "unmerged"
    }
    <<
      \new Voice { \voiceOne \voiceA }
      \new Voice { \voiceTwo \voiceB }
    >>
    \new Staff \with {
      instrumentName = "merged"
      \consists Merge_rests_engraver
    }
    <<
      \new Voice { \voiceOne \voiceA }
      \new Voice { \voiceTwo \voiceB }
    >>
  >>
}
```



Al establecer el valor de la propiedad de contexto `suspendRestMerging` a `#t` podemos desactivar temporalmente la combinación de silencios.

5.2.5 Combinación automática de las partes

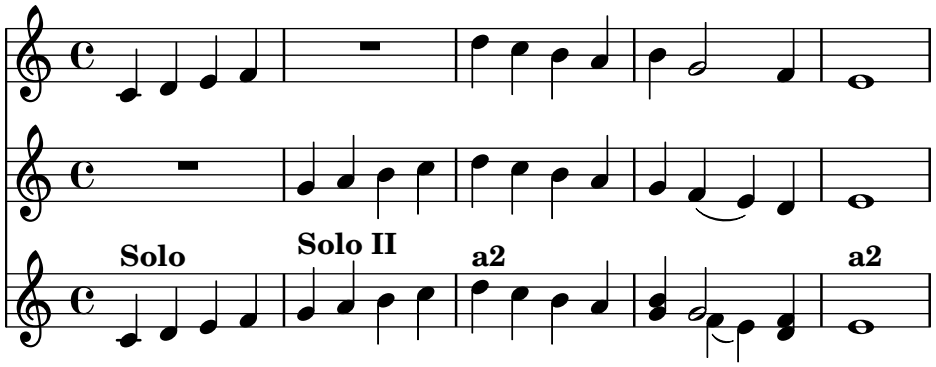
La combinación automática de particellas se usa para mezclar dos partes musicales distintas sobre un pentagrama. Esto puede ser de gran ayuda especialmente al tipografiar partituras orquestales. Se imprime una sola voz cuando la música de las dos voces es la misma, pero en aquellos lugares en que difieren, se imprime una segunda voz. Las direcciones de las plicas se establecen hacia arriba o hacia abajo según proceda, al tiempo que se identifican y quedan marcadas las partes de solista y de dúo.

La sintaxis para la combinación automática de las partes es:

```
\partCombine expresión_musical_1 expresión_musical_2
```

El ejemplo siguiente ejemplifica la funcionalidad básica, poniendo las partes en un solo pentagrama en forma polifónica, y estableciendo las direcciones de las plicas de forma adecuada. Se utilizan las mismas variables para las partes independientes y el pentagrama combinado.

```
instrumentOne = \relative {
  c'4 d e f |
  R1 |
  d'4 c b a |
  b4 g2 f4 |
  e1 |
}

instrumentTwo = \relative {
  R1 |
  g'4 a b c |
  d4 c b a |
  g4 f( e) d |
  e1 |
}

<<
  \new Staff \instrumentOne
  \new Staff \instrumentTwo
  \new Staff \partCombine \instrumentOne \instrumentTwo
>>
```

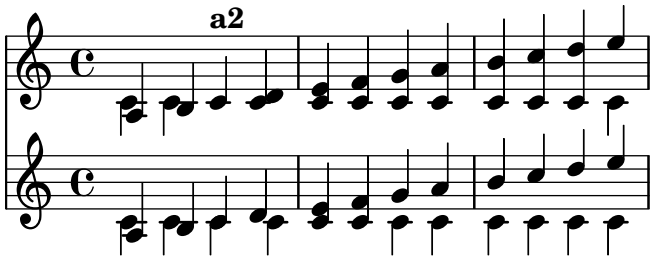


Las dos partes tienen notas idénticas en el tercer compás, por lo que aparecen solamente una vez. Las direcciones de las plicas y ligaduras se establecen de forma automática, según se trate de un solo o de un unísono. Cuando se necesita en situaciones de polifonía, la primera parte (que recibe el nombre de contexto `one`) recibe las plicas hacia arriba, mientras que la segunda (llamada `two`) siempre recibe las plicas hacia abajo. En los fragmentos de solo, las partes se marcan con "Solo" y "Solo II", respectivamente. Las partes en unísono (*a due*) se marcan con el texto "a2".

De forma predeterminada, el combinador de partes mezcla dos notas de la misma altura como una nota *a due*, combina las notas que tengan la misma duración y estén a menos de una novena de distancia como acordes, y dispone en voces separadas las notas que estén a más de una novena de distancia (o cuando hay cruce de voces). Este comportamiento se puede sobreescribir con un argumento opcional que consiste en una pareja de números después de la instrucción `\partCombine`: el primero especifica el intervalo donde las notas comienzan a estar combinadas (el valor predeterminado es cero) y el segundo donde las notas se dividen en voces distintas. El ajuste del segundo argumento al valor cero significa que el combinador de partes separa las notas

que estén dentro de un intervalo de segunda o más, establecerlo al valor uno separa las notas de una tercera o más, y así sucesivamente.

```
instrumentOne = \relative {
  a4 b c d |
  e f g a |
  b c d e |
}

instrumentTwo = \relative {
  c'4 c c c |
  c c c c |
  c c c c |
}

<<
  \new Staff \partCombine \instrumentOne \instrumentTwo
  \new Staff \partCombine #'(2 . 3) \instrumentOne \instrumentTwo
>>
```



Los dos argumentos de \partCombine se interpretan como contextos de Sección "Voice" en *Referencia de Funcionamiento Interno* separados, por lo que si la música está escrita en modo relativo, entonces *las dos* partes han de incluir la función \relative, es decir:

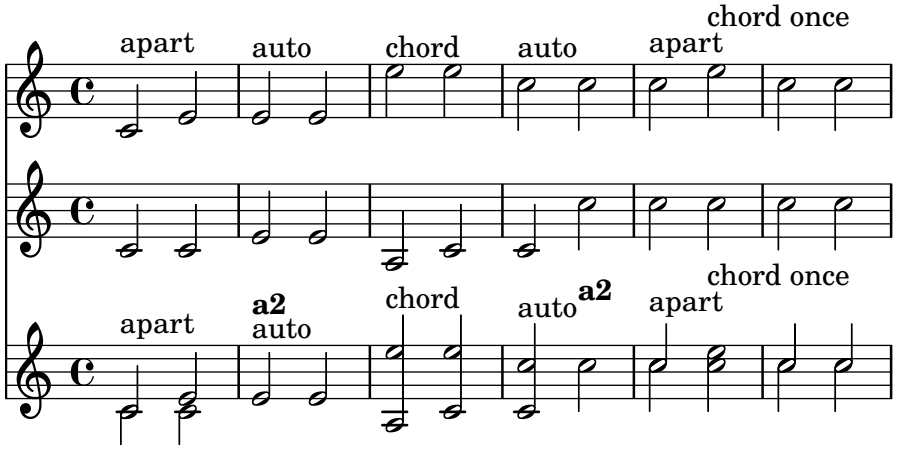
```
\partCombine
  \relative ... expresión_musical_1
  \relative ... expresión_musical_2
```

Una sección \relative que se encuentra fuera de \partCombine no tiene ningún efecto sobre las notas de *expresión_musical_1* y *expresión_musical_2*.

En las partituras profesionales, las voces con frecuencia se mantienen separadas entre sí durante pasajes prolongados incluso si algunas notas coninciden entre ambas, y podrían escribirse fácilmente como un unísono. Por ello, la combinación de las notas en un acorde, o mostrar una voz como solo, no son soluciones ideales porque la función \partCombine considera cada nota de forma individual. En este caso, el resultado de la función \partCombine se puede alterar o corregir con una de las instrucciones siguientes. Todas las instrucciones se pueden anteceder por \once para que se apliquen solamente a la nota siguiente dentro de la expresión musical.

- \partCombineApart mantiene las notas como dos voces separadas incluso si se pueden combinar en un acorde o unísono.
- \partCombineChords combina las notas en un acorde.
- \partCombineUnisono combina las dos voces como unísono y marca el resultado como "unison".
- \partCombineSoloI muestra solo la voz uno y la marca como "Solo".
- \partCombineSoloII imprime solo la voz dos y la marca como "Solo".
- \partCombineAutomatic termina el efecto de las instrucciones anteriores y retorna a la funcionalidad estándar de \partCombine.

```
instrumentOne = \relative c' {
  \partCombineApart c2^"apart" e |
  \partCombineAutomatic e2^"auto" e |
  \partCombineChords e'2^"chord" e |
  \partCombineAutomatic c2^"auto" c |
  \partCombineApart c2^"apart"
    \once \partCombineChords e^"chord once" |
  c2 c |
}
instrumentTwo = \relative {
  c'2 c |
  e2 e |
  a,2 c |
  c2 c' |
  c2 c |
  c2 c |
}

<<
  \new Staff { \instrumentOne }
  \new Staff { \instrumentTwo }
  \new Staff { \partCombine \instrumentOne \instrumentTwo }
>>
```



Uso de \partCombine con letra

La instrucción \partCombine no está diseñada para funcionar con la letra de las canciones; si una de las voces recibe un nombre explícito para poder adjuntar una letra, el combinador de partes no funciona. Sin embargo, se puede conseguir este efecto usando un contexto NullVoice. Véase Sección 9.2.6 [Polifonía con letras compartidas], página 380.

Fragmentos de código seleccionados

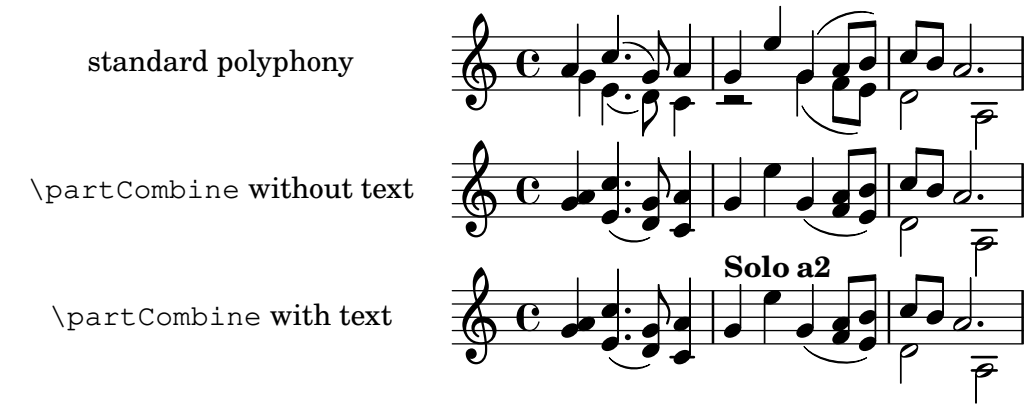
Combinar dos partes sobre el mismo pentagrama

La herramienta de combinación de partes (o sea, la instrucción \partCombine) permite la combinación de varias partes diferentes sobre el mismo pentagrama. Las indicaciones textuales tales como “solo” o “a2” se añaden de forma predeterminada; para quitarlas, sencillamente establezca la propiedad printPartCombineTexts al valor #f. Para partituras vocales (como himnos), no hay necesidad de añadir los textos “solo” o “a2”, por lo que se deben desactivar. Sin

embargo, podría ser mejor no usarlos si hay solos, porque éstos no se indicarán. En tales casos podría ser preferible la notación polifónica estándar.

Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, \partCombine sin textos, y \partCombine con textos.

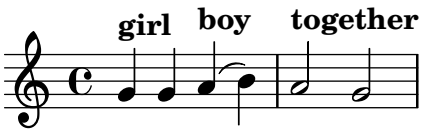
```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\ \musicDown >>

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" without text}
      printPartCombineTexts = ##f
    } \partCombine \musicUp \musicDown

    \new Staff \with {
      instrumentName =
        \markup { \typewriter "\\partCombine" with text}
    } \partCombine \musicUp \musicDown
  >>

  \layout {
    indent = 6.0\cm
    \context {
      \Score
      % Setting this to a large value avoids a bar line at the
      % beginning that would connect the three staves otherwise.
      \override SystemStartBar.collapse-height = 30
    }
  }
}
```



Cambiar los textos de `\partCombine`

Al utilizar la posibilidad de combinación automática de partes, se puede modificar el texto que se imprime para las secciones de solo y de unísono:

```
\new Staff <<
  \set Staff.soloText = "girl"
  \set Staff.soloIIText = "boy"
  \set Staff.aDueText = "together"
  \partCombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
>>
```



Véase también

Glosario musical: Sección “a due” en *Glosario Musical*, Sección “part” en *Glosario Musical*.

Referencia de la notación: Sección 6.3 [Escritura de las particellas], página 265.

Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PartCombineMusic” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Todas las funciones `\partCombine...` admiten exclusivamente dos voces.

Las funciones `\partCombine...` no se pueden escribir dentro de los bloques `\tuplet` ni `\relative`.

Si `printPartCombineTexts` está establecido y las dos voces tocan y terminan las mismas notas en el mismo compás, el combinador de partes puede tipografiar `a2` más de una vez en ese compás.

\partCombine solo tiene en cuenta el momento de inicio de las notas dentro de una voz Voice; por ejemplo, no puede recordar si una nota dentro de una voz ya ha comenzado cuando las notas que se combinan se han iniciado justo ahora en la otra voz. Esto puede conducir a cierto número de problemas inesperados entre los que se incluye la impresión incorrecta de marcas de "Solo" o de "Unison".

\partCombine conserva todos los elementos extensos (ligaduras, reguladores, etc.) dentro del mismo contexto Voice de forma que si uno cualquiera de estos elementos extensos inicia o termina en un contexto Voice diferente, puede no imprimirse o hacerlo incorrectamente.

Si la función \partCombine no puede combinar las dos expresiones musicales (es decir, cuando las dos voces tienen duraciones distintas), otorgará internamente sus propios nombres a las voces: one y two respectivamente. Esto significa que si se produce un cambio a un contexto Voice que tenga un nombre distinto, se ignorarán los eventos dentro del contexto Voice que tiene el nombre distinto.

A causa de que \partCombine es una funcionalidad en dos pasos, se debe tener cuidado de no confundir las dos fases. Por ejemplo, este código

```
uno = { e''2 \tag #'score f''
            \tag #'part fis'' g''1 }
dos = { e''2 d'' g'1 }

\removeWithTag #'score \partCombine \uno \dos
```

no funciona porque el primer paso no sabe que estamos eliminando música etiquetada, por lo que la información que registra no es consistente con la música cuando llega el momento de la segunda fase. Si queremos usar \partCombine con filtrados, repeticiones desplegadasu otras transformaciones, tenemos que hacer la transformación primero para que la música se encuentre en su forma definitiva antes de la primera fase, por ejemplo,

```
...
\partCombine
  \removeWithTag #'score \uno
  \removeWithTag #'score \dos
```

Consulte también el apartado *Advertencias y problemas conocidos* en la sección Sección 12.1.3 [Tablaturas predeterminadas], página 436, al utilizar \partCombine con tablaturas, y la *Nota* de Sección 2.4.1 [Barras automáticas], página 101, al utilizar barrado automático.

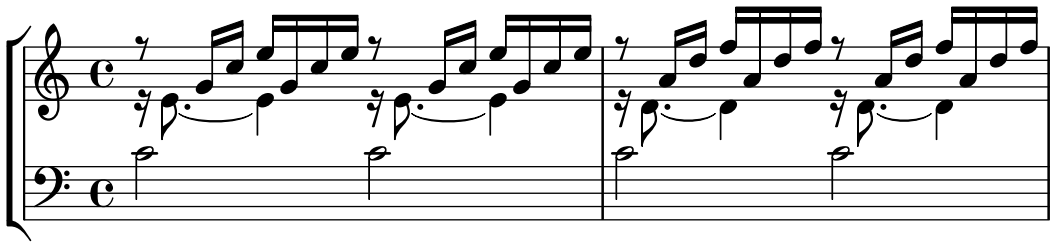
5.2.6 Escribir música en paralelo

La música para varias partes se puede intercalar dentro del código de entrada. La función \parallelMusic admite una lista que contiene los nombres de las variables que se van a crear, y una expresión musical. El contenido de los compases alternativos extraídos de la expresión se convierten en el valor de las variables respectivas, de manera que podemos utilizarlas más tarde para imprimir la música.

> **Nota:** Es obligatorio utilizar comprobaciones de compás |, y los compases deben tener la misma longitud.

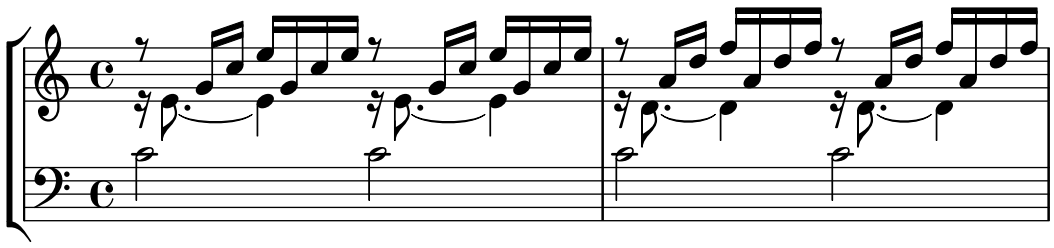
```
\parallelMusic voiceA,voiceB,voiceC {
  % Bar 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~   4              r16 e'8.~   4              |
  c'2                        c'2                        |

  % Bar 2
```

```
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~   4             r16 d'8.~   4              |
  c'2                       c'2                        |

}
\new StaffGroup <<
  \new Staff << \voiceA \\ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>
```



Se puede usar el modo relativo. Observe que la instrucción \relative no se utiliza dentro del propio bloque \parallelMusic. Las notas guardan relación con la nota anterior en la misma voz, no con la nota anterior dentro del código de entrada (dicho de otra manera, las notas relativas de vozA ignoran a las notas que hay en vozB.

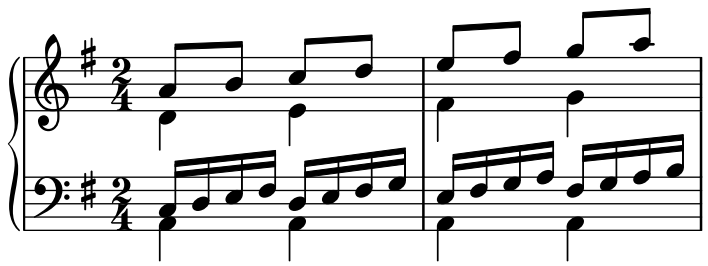
```
\parallelMusic voiceA,voiceB,voiceC {
  % Bar 1
  r8 g16 c e g, c e r8 g,16 c e g, c e  |
  r16 e8.~ 4        r16 e8.~  4         |
  c2                c                   |

  % Bar 2
  r8 a,16 d f a, d f r8 a,16 d f a, d f |
  r16 d8.~  4        r16 d8.~  4        |
  c2                 c                  |

 }
\new StaffGroup <<
  \new Staff << \relative c'' \voiceA \\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>
```



Esto funciona aceptablemente bien para música de piano. El siguiente ejemplo asigna cada cuatro compases consecutivos a cuatro variables:

```
global = {
  \key g \major
  \time 2/4
}
```

```
\parallelMusic voiceA,voiceB,voiceC,voiceD {
  % Bar 1
  a8    b     c   d     |
  d4          e         |
  c16 d e fis d e fis g  |
  a4          a         |

  % Bar 2
  e8      fis  g     a   |
  fis4         g         |
  e16 fis g  a fis g a b  |
  a4           a         |

  % Bar 3 ...
}

\score {
  \new PianoStaff <<
     \new Staff {
       \global
       <<
         \relative c'' \voiceA
         \\
         \relative c'  \voiceB
       >>
     }
     \new Staff {
       \global \clef bass
       <<
         \relative c \voiceC
         \\
         \relative c \voiceD
       >>
     }
  >>
}
```

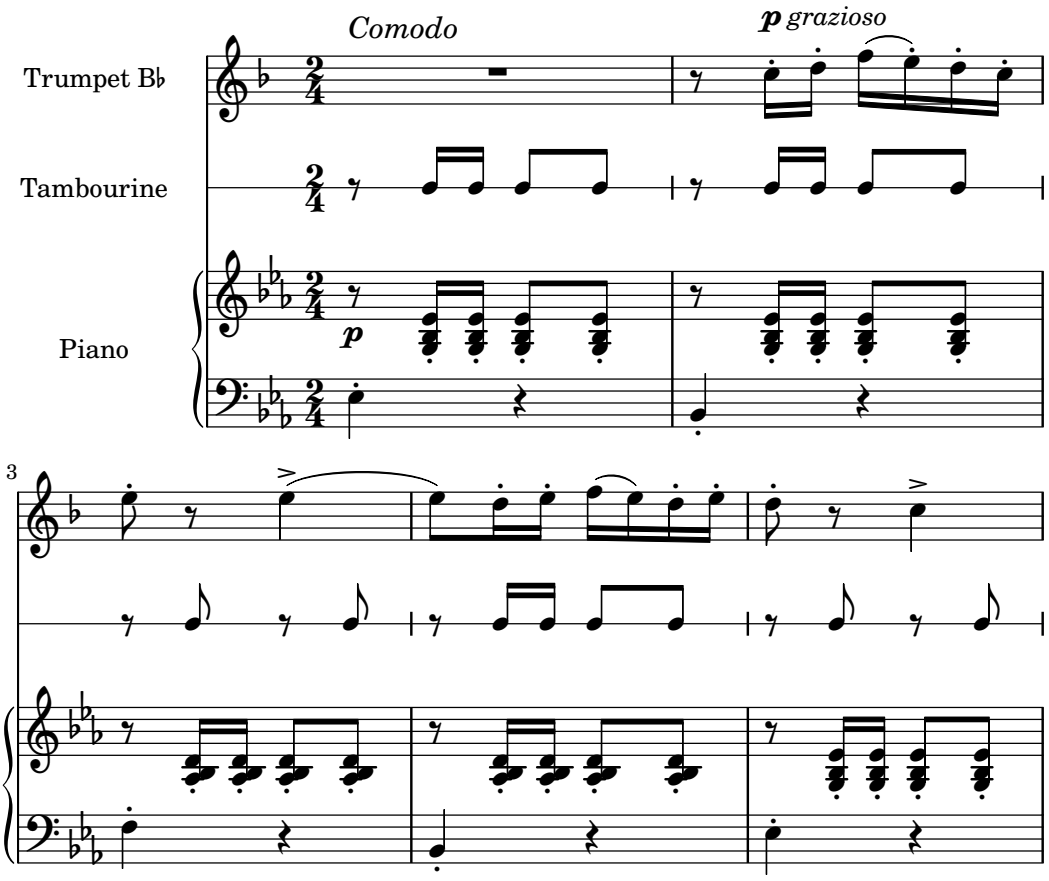


Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*

Fragmentos de código: Sección “Notas simultáneas” en *Fragmentos de código*.

6 Notación de los pentagramas



Esta sección explica cómo influir sobre la apariencia de los pentagramas, cómo imprimir partituras con más de un pentagrama y cómo añadir indicaciones de tempo y notas guía a los pentagramas.

6.1 Impresión de los pentagramas

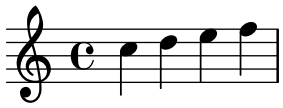
Esta sección describe los distintos métodos de creación de pentagramas y grupos de ellos.

6.1.1 Crear instancias de pentagramas nuevos

Las *pautas* y los *pentagramas* o pautas de cinco líneas se crean con las instrucciones `\new` o `\context`. Para ver más detalles, consulte Sección 33.2 [Crear y referenciar contextos], página 746.

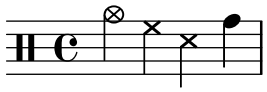
El contexto básico de pentagrama es `Staff`:

```
\new Staff \relative { c''4 d e f }
```



El contexto `DrumStaff` crea una pauta de cinco líneas preparada para un conjunto de batería típico. Cada instrumento se presenta con un símbolo distinto. Los instrumentos se escriben en el modo de percusión que sigue a una instrucción `\drummode`, con cada instrumento identificado por un nombre. Para ver más detalles, consulte Sección 13.1.5 [Pautas de percusión], página 492.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



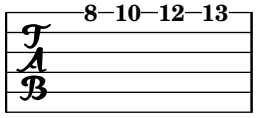
RhythmicStaff crea una pauta de una sola línea que sólo muestra las duraciones de la entrada. Se preservan las duraciones reales. Para ver más detalles, consulte Sección 2.3.7 [Mostrar los ritmos de la melodía], página 98.

```
\new RhythmicStaff { c4 d e f }
```



TabStaff crea una tablatura con seis cuerdas en la afinación estándar de guitarra. Para ver más detalles, consulte Sección 12.1.3 [Tablaturas predeterminadas], página 436.

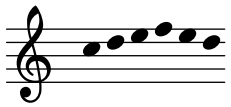
```
\new TabStaff \relative { c''4 d e f }
```



Existen contextos de pauta específicos para la notación de música antigua, por ejemplo MensuralStaff y VaticanaStaff. Se describen en Sección 17.2.1 [Contextos predefinidos], página 542.

GregorianTranscriptionStaff crea una pauta para la notación moderna de canto gregoriano. Representa las *divisiones* gregorianas como líneas divisorias, pero no muestra barras de compás.

```
\new GregorianTranscriptionStaff \relative { c''4 d e f e d }
```



Se pueden definir contextos nuevos de pentagrama único. Para ver más detalles, consulte Sección 33.6 [Definir contextos nuevos], página 758.

Véase también

Glosario musical: Sección “staff” en *Glosario Musical*, Sección “staves” en *Glosario Musical*.

Referencia de la notación: Sección 33.2 [Crear y referenciar contextos], página 746, Sección 13.1.5 [Pautas de percusión], página 492, Sección 2.3.7 [Mostrar los ritmos de la melodía], página 98, Sección 12.1.3 [Tablaturas predeterminadas], página 436, Sección 17.2.1 [Contextos predefinidos], página 542, Sección 6.2.1 [El símbolo del pentagrama], página 252, Sección 17.4.1 [Contextos del canto gregoriano], página 552, Sección 17.3.1 [Contextos de la música mensural], página 544, Sección 33.6 [Definir contextos nuevos], página 758.

Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

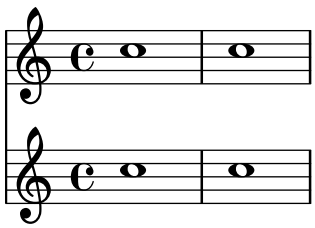
Referencia de funcionamiento interno: Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “DrumStaff” en *Referencia de Funcionamiento Interno*, Sección “GregorianTranscriptionStaff” en *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” en *Referencia de Funcionamiento Interno*, Sección “TabStaff” en *Referencia de Funcionamiento Interno*, Sección “MensuralStaff” en *Referencia de Funcionamiento Interno*, Sección “VaticanaStaff” en *Referencia de Funcionamiento Interno*, Sección “StaffSymbol” en *Referencia de Funcionamiento Interno*.

6.1.2 Agrupar pentagramas

Existen varios contextos para agrupar pentagramas individuales formando sistemas. Cada contexto de agrupación establece el estilo del delimitador de comienzo del sistema y el comportamiento de las barras de compás.

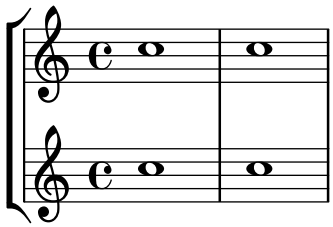
Si no se especifica ningún contexto, se usan las propiedades predeterminadas: el grupo comienza con una línea vertical y las barras de compás no están conectadas.

```
<<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



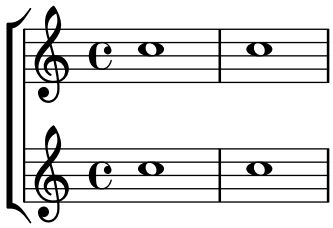
En el contexto `StaffGroup`, el grupo se inicia con un corchete y las barras de compás se dibujan atravesando todos los pentagramas.

```
\new StaffGroup <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



En un `ChoirStaff` (sistema de coro), el grupo se inicia con un corchete, pero las barras de compás no están conectadas.

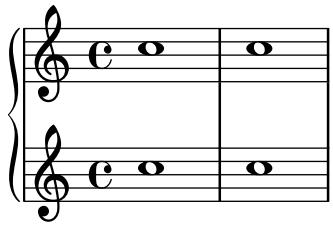
```
\new ChoirStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



En un `GrandStaff` (sistema de piano), el grupo se inicia con una llave y las barras de compás se conectan entre los pentagramas.

```
\new GrandStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
```

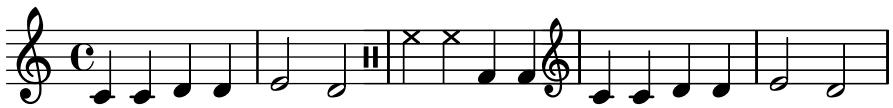
```
>>
```



El `PianoStaff` (sistema de piano) es idéntico a `GrandStaff`, excepto que sus pautas se suprimen siempre juntas, nunca por separado; véase Sección 6.2.3 [Ocultar pentagramas], página 260.

El contexto `OneStaff` es un grupo de pentagramas que sitúa contextos separados dentro de la misma alineación vertical. Este ejemplo muestra tres pautas que comparten el mismo espacio. Aquí, el grabador `Time_signature_engraver` se ha movido del contexto de los pentagramas al contexto `OneStaff` para evitar que se repita para cada pauta.

```
\layout {
  \context {
    \OneStaff
    \consists Time_signature_engraver
  }
  \context {
    \Staff
    \remove Time_signature_engraver
  }
  \context {
    \DrumStaff
    \remove Time_signature_engraver
  }
}

\new OneStaff {
  \new Staff {
    c'4 4 d'4 4 e'2 d'
  }
  \drums {
    hihat4 hh bassdrum bd
  }
  \new Staff {
    c'4 4 d'4 4 e'2 d'
  }
}
```



Cada contexto de grupo de pentagramas fija la propiedad del delimitador de inicio `systemStartDelimiter` a uno de los siguientes valores: `SystemStartBar` (línea), `SystemStartBrace` (llave) o `SystemStartBracket` (corchete). También está disponible un cuarto delimitador, `SystemStartSquare` (corchete en ángulo recto), pero se debe especificar explícitamente.

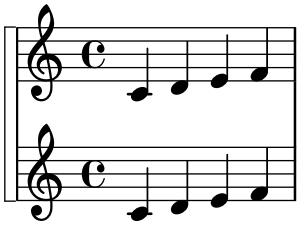
Se pueden definir contextos nuevos de grupo de pentagramas. Para ver más detalles, consulte Sección 33.6 [Definir contextos nuevos], página 758.

Fragmentos de código seleccionados

Uso del corchete recto al comienzo de un grupo de pentagramas

Se puede usar el delimitador de comienzo de un sistema `SystemStartSquare` estableciéndolo explícitamente dentro de un contexto `StaffGroup` o `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```

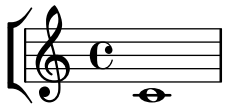


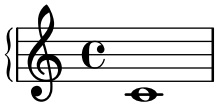
Mostrar corchete o llave en grupos de un solo pentagrama

Si hay un solo pentagrama en un contexto `ChoirStaff` o `StaffGroup`, el comportamiento predeterminado es que no se imprima el corchete en la barra inicial. Esto se puede cambiar estableciendo la propiedad `collapse-height` a un valor que sea menor que el número de líneas en la pauta.

Observe que en contextos como `PianoStaff` y `GrandStaff` en que los sistemas empiezan con una llave en lugar de un corchete, se debe establecer el valor de una propiedad distinta, como se ve en el segundo sistema del ejemplo.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
```

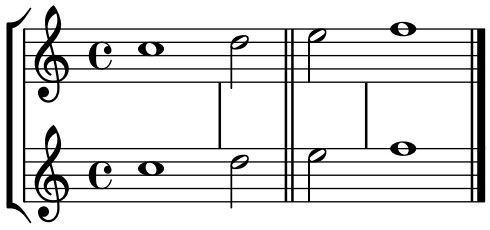




Disposición Mensurstriche (líneas divisorias entre pentagramas)

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir fijando `measureBarType` al valor `"-span|"` y usando un contexto de agrupamiento que admita barras extendidas, como `StaffGroup`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Véase también

Glosario musical: Sección “brace” en *Glosario Musical*, Sección “bracket” en *Glosario Musical*, Sección “grand staff” en *Glosario Musical*, Sección “mensurstrich” en *Glosario Musical*.

Referencia de la notación: Sección 6.3.1 [Nombres de instrumentos], página 265, Sección 33.6 [Definir contextos nuevos], página 758.

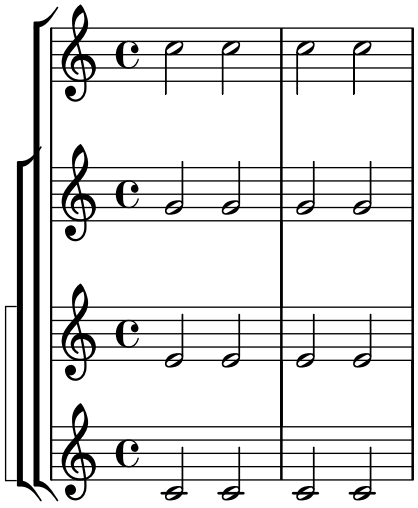
Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “StaffGroup” en *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” en *Referencia de Funcionamiento Interno*, Sección “GrandStaff” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*, Sección “OneStaff” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBracket” en *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” en *Referencia de Funcionamiento Interno*.

6.1.3 Grupos de pentagramas anidados

Los contextos de grupos de pentagramas se pueden anidar hasta una profundidad arbitraria. En este caso, cada contexto descendiente crea un corchete nuevo adyacente al corchete de su grupo padre.

```
\new StaffGroup <<
  \new Staff \relative { c''2 c | c2 c }
  \new StaffGroup <<
    \new Staff \relative { g'2 g | g2 g }
    \new StaffGroup \with {
      systemStartDelimiter = #'SystemStartSquare
    }
    <<
      \new Staff \relative { e'2 e | e2 e }
      \new Staff \relative { c'2 c | c2 c }
    >>
  >>
>>
```



Se pueden definir nuevos contextos de grupos de pentagramas anidados. Para ver más detalles, consulte Sección 33.6 [Definir contextos nuevos], página 758.

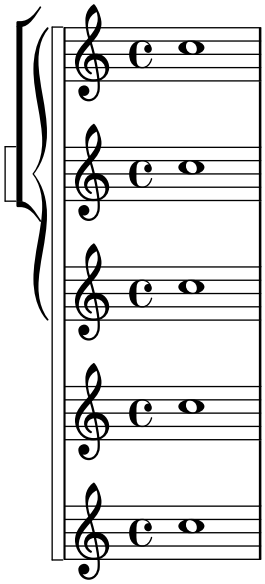
Fragmentos de código seleccionados

Anidado de grupos de pentagramas

Se puede utilizar la propiedad `systemStartDelimiterHierarchy` para crear grupos de pentagramas anidados de forma más compleja. La propiedad `systemStartDelimiterHierarchy` del contexto `StaffGroup` toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierren las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` y `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
```

```
            (SystemStartBrace
              (SystemStartBracket a
                (SystemStartSquare b))
              c)
            d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Véase también

Referencia de la notación: Sección 6.1.2 [Agrupar pentagramas], página 246, Sección 6.3.1 [Nombres de instrumentos], página 265, Sección 33.6 [Definir contextos nuevos], página 758.

Fragmentos de código: Sección "Notación de los pentagramas" en *Fragmentos de código*.

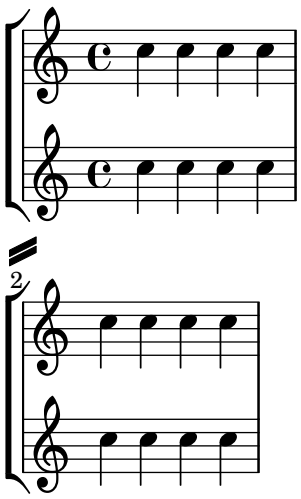
Referencia de funcionamiento interno: Sección "StaffGroup" en *Referencia de Funcionamiento Interno*, Sección "ChoirStaff" en *Referencia de Funcionamiento Interno*, Sección "SystemStartBar" en *Referencia de Funcionamiento Interno*, Sección "SystemStartBrace" en *Referencia de Funcionamiento Interno*, Sección "SystemStartBracket" en *Referencia de Funcionamiento Interno*, Sección "SystemStartSquare" en *Referencia de Funcionamiento Interno*.

6.1.4 Separación de sistemas

Si el número de sistemas por página varía de una página a otra, es costumbre separar los sistemas colocando una marca separadora entre ellos. De forma predeterminada, el separador de sistemas es nulo, pero se puede activar con una opción de \paper.

```
\book {
  \score {
    \new StaffGroup <<
      \new Staff {
        \relative {
          c''4 c c c
          \break
```

```
          c4 c c c
        }
      }
      \new Staff {
        \relative {
          c''4 c c c
          \break
          c4 c c c
        }
      }
    >>
  }
  \paper {
    system-separator-markup = \slashSeparator
    tagline = ##f
  }
}
```



Véase también

Referencia de la notación: Capítulo 26 [Disposición de la página], página 673.

Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

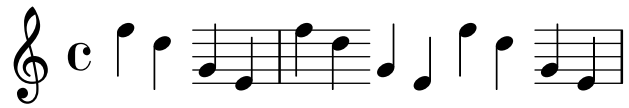
6.2 Modificación de pentagramas sueltos

Esta sección explica cómo cambiar los atributos específicos de un pentagrama: por ejemplo, cambiar el número de líneas o el tamaño de la pauta. También se describen los métodos para comenzar y terminar los pentagramas, y establecer secciones de ossia.

6.2.1 El símbolo del pentagrama

Se pueden usar las instrucciones `\stopStaff` y `\startStaff` para detener y (re)iniciar, respectivamente, la impresión de las líneas de la pauta en cualquier punto de la partitura.

```
\relative {
  \stopStaff f''4 d \startStaff g, e
  f'4 d \stopStaff g, e
  f'4 d \startStaff g, e
}
```



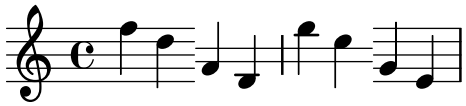
Instrucciones predefinidas

`\startStaff`, `\stopStaff`.

Las líneas de la pauta pertenecen al grob `StaffSymbol` (incluidas las líneas adicionales) y se pueden modificar usando las propiedades de `StaffSymbol`, pero esas modificaciones deben hacerse antes de que el pentagrama se (re)inicie.

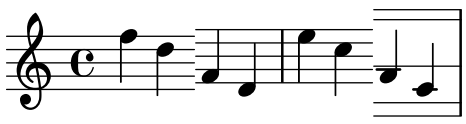
Se puede cambiar el número de líneas de la pauta:

```
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-count = 2
  \startStaff g, e |

  f'4 d \stopStaff
  \revert Staff.StaffSymbol.line-count
  \startStaff g, e |
}
```



La posición de cada una de las líneas de la pauta también puede cambiarse. Una lista de números fija la posición de cada línea. `0` corresponde a la línea central normal, y las posiciones normales de las líneas son `(-4 -2 0 2 4)`. Se imprime una sola línea de pauta por cada valor que se introduce, de manera que el número de líneas así como su posición en la pauta pueden cambiarse con una sola operación de sobreescritura (así, la propiedad `line-count` no se tiene en cuenta si está establecida la propiedad `line-positions`).

```
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-positions = #'(1 3 5 -1 -3)
  \startStaff g, e |
  f'4 d \stopStaff
  \override Staff.StaffSymbol.line-positions = #'(8 6.5 -6 -8 -0.5)
  \startStaff g, e |
}
```

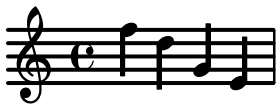


Para preservar las direcciones típicas de las plicas (hacia arriba para la mitad inferior del pentagrama y hacia abajo para la mitad superior), debemos alinear la línea (o espacio) central de la pauta personalizada con la posición de la línea central normal (0). Puede ser necesario un ajuste de la posición de la clave y del Do central para que se correspondan a las líneas nuevas. Véase Sección 1.3.1 [Clave], página 20.

Puede cambiarse el grosor de las líneas de la pauta. También resultan afectadas, de forma predeterminada, las líneas adicionales y no la plica de las figuras.

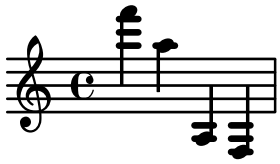
```
\new Staff \with {
```

```
  \override StaffSymbol.thickness = 3
} \relative {
  f''4 d g, e
}
```



También es posible fijar el grosor de las líneas adicionales de forma independiente del de las líneas de la pauta.

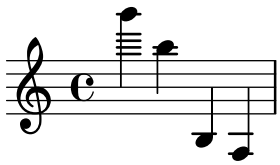
```
\new Staff \with {
  \override StaffSymbol.thickness = 2
  \override StaffSymbol.ledger-line-thickness = #'(0.5 . 0.4)
} \relative {
  f'''4 a, a,, f
}
```



El primer valor se multiplica por el grosor de las líneas del pentagrama, el segundo por el ancho de un espacio del pentagrama, y después los dos valores se suman para obtener el grosor de las líneas adicionales.

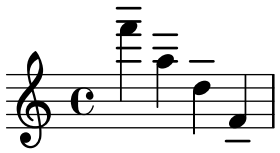
Se pueden alterar las posiciones de las líneas adicionales:

```
\new Staff \with {
  \override StaffSymbol.ledger-positions = #'(-3 -2 -1 2 5 6)
} \relative {
  f'''4 a, a,, f
}
```



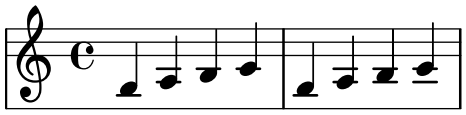
Puede hacerse que las líneas adicionales añadidas aparezcan por encima o por debajo de la cabeza de las figuras, dependiendo de la posición actual relativa a otras cabezas de figura que tienen asimismo sus propias líneas adicionales.

```
\new Staff \with {
  \override StaffSymbol.ledger-extra = 4
} \relative {
  f'''4 a, d, f,
}
```



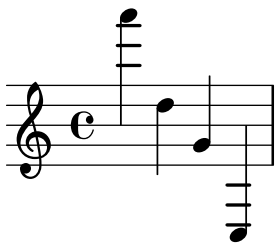
También puede hacerse que las líneas adicionales aparezcan dentro del pentagrama allí donde se requieren líneas de pauta personalizadas. El ejemplo muestra la posición predeterminada de las líneas adicionales cuando el valor de posición explícito `ledger-position` se ha fijado o no. La instrucción `\stopStaff` se necesita en el ejemplo para revertir la instrucción de sobreescritura `\override` para todo el `StaffSymbol` (el símbolo de pauta).

```
\relative d' {
  \override Staff.StaffSymbol.line-positions = #'(-8 0 2 4)
  d4 e f g
  \stopStaff
  \startStaff
  \override Staff.StaffSymbol.ledger-positions = #'(-8 -6 (-4 -2) 0)
  d4 e f g
}
```



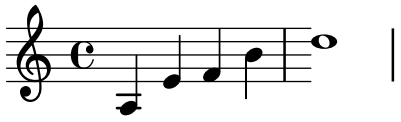
Puede alterarse la distancia entre líneas de la pauta. Esto afecta también al espaciado de las líneas adicionales.

```
\new Staff \with {
  \override StaffSymbol.staff-space = 1.5
} \relative {
  f'''4 d, g, e,
}
```



Se puede modificar la anchura de un pentagrama. Las unidades son espacios de pentagrama. El espaciado de los objetos dentro de la pauta no resulta afectado por este ajuste.

```
\new Staff \with {
  \override StaffSymbol.width = 23
}
\relative { a4 e' f b | d1 }
```

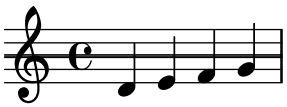


Fragmentos de código seleccionados

Hacer unas líneas del pentagrama más gruesas que las otras

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Véase también

Glosario musical: Sección “line” en *Glosario Musical*, Sección “ledger line” en *Glosario Musical*, Sección “staff” en *Glosario Musical*.

Referencia de la notación: Sección 1.3.1 [Clave], página 20.

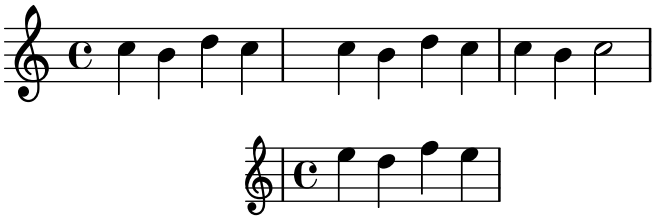
Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” en *Referencia de Funcionamiento Interno*, Sección “staff-symbol-interface” en *Referencia de Funcionamiento Interno*.

6.2.2 Pentagramas de Ossia

Los pentagramas *Ossia* se pueden preparar mediante la creación de un pentagrama simultáneo nuevo en la posición adecuada:

```
\new Staff \relative {
  c''4 b d c
  <<
    { c4 b d c }
    \new Staff { e4 d f e }
  >>
  c4 b c2
}
```

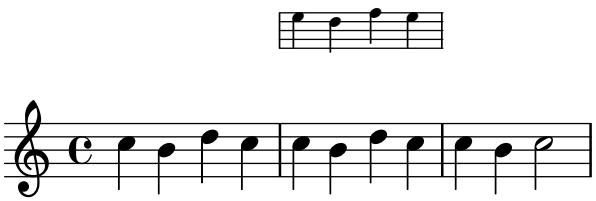


Sin embargo, el ejemplo anterior no es lo que normalmente se desea. Para crear pentagramas de ossia que estén encima del pentagrama original, que no tengan compás ni clave, y que tengan un tamaño menor de tipografía se deben usar varios trucos. El Manual de aprendizaje describe una técnica específica para llegar a este objetivo, empezando por Sección “Anidado de expresiones musicales” en *Manual de Aprendizaje*.

El ejemplo siguiente utiliza la propiedad `alignAboveContext` para alinear el pentagrama de ossia. Este método es muy conveniente cuando se necesitan sólo algunos pentagramas de ossia.

```
\new Staff = "main" \relative {
```

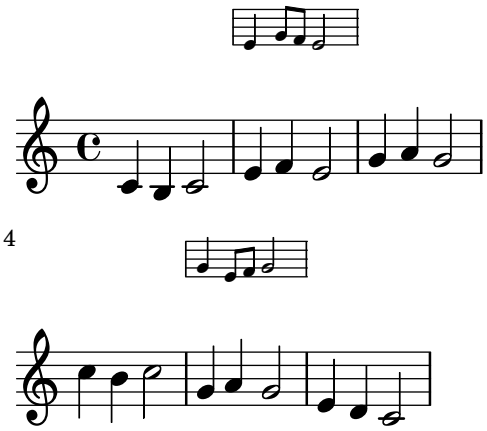
```
  c''4 b d c
  <<
    { c4 b d c }

    \new Staff \with {
      \remove Time_signature_engraver
      alignAboveContext = "main"
      \magnifyStaff #2/3
      firstClef = ##f
    }
    { e4 d f e }
  >>
  c4 b c2
}
```



Si se requieren muchos pentagramas de ossia aislados, puede ser más conveniente la creación de un contexto `Staff` vacío con un *identificador de contexto* específico; después se pueden crear los pentagramas de ossia *llamando* a este contexto y usando `\startStaff` y `\stopStaff` en los puntos deseados. Las ventajas de este método son más patentes si la pieza es más larga que en ejemplo siguiente.

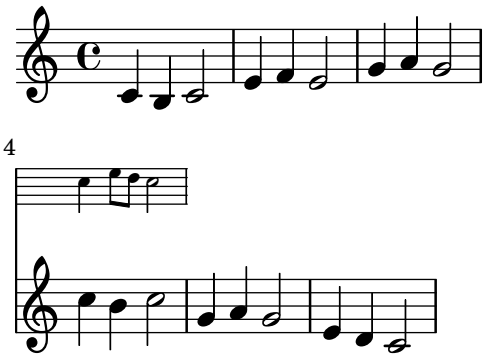
```
<<
  \new Staff = "ossia" \with {
    \remove Time_signature_engraver
    \hide Clef
    \magnifyStaff #2/3
  }
  { \stopStaff s1*6 }

  \new Staff \relative {
    c'4 b c2
    <<
      { e4 f e2 }
      \context Staff = "ossia" {
        \startStaff e4 g8 f e2 \stopStaff
      }
    >>
    g4 a g2 \break
    c4 b c2
    <<
      { g4 a g2 }
      \context Staff = "ossia" {
        \startStaff g4 e8 f g2 \stopStaff
      }
    >>
    e4 d c2
  }
```

```
>>
```



Como alternativa, se puede usar la instrucción `\RemoveAllEmptyStaves` para crear pentagramas de ossia. Este método es muy conveniente cuando los pentagramas de ossia aparecen inmediatamente después de un salto de línea. Para ver más información sobre `\RemoveAllEmptyStaves`, consulte Sección 6.2.3 [Ocultar pentagramas], página 260.

```
<<
  \new Staff = "ossia" \with {
    \remove Time_signature_engraver
    \hide Clef
    \magnifyStaff #2/3
    \RemoveAllEmptyStaves
  } \relative {
    R1*3
    c''4 e8 d c2
  }
  \new Staff \relative {
    c'4 b c2
    e4 f e2
    g4 a g2 \break
    c4 b c2
    g4 a g2
    e4 d c2
  }
>>
```

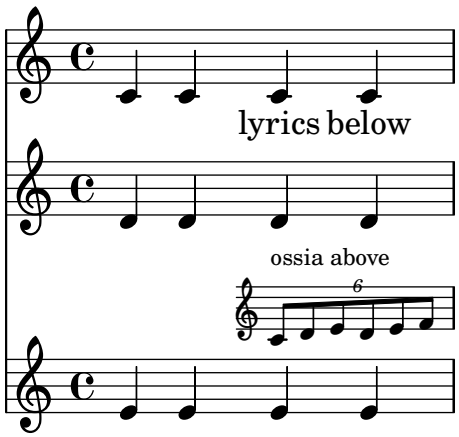


Fragmentos de código seleccionados

Alineación vertical de la letra y los compases de ossia

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```
\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
        \override VerticalAxisGroup.staff-staff-spacing =
           #'((minimum-distance . 0)
              (basic-distance . 0)
              (padding . 1))
      } {
        \tuplet 6/4 {
          \override TextScript.padding = 2
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Véase también

Glosario musical: Sección “ossia” en *Glosario Musical*, Sección “staff” en *Glosario Musical*, Sección “Frenched staff” en *Glosario Musical*.

Manual de aprendizaje: Sección “Anidado de expresiones musicales” en *Manual de Aprendizaje*, Sección “Tamaño de los objetos” en *Manual de Aprendizaje*, Sección “Longitud y grosor de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 6.2.3 [Ocultar pentagramas], página 260.

Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” en *Referencia de Funcionamiento Interno*.

6.2.3 Ocultar pentagramas

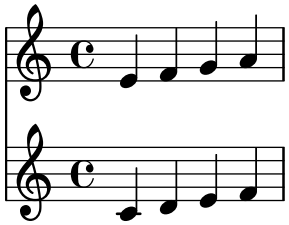
Se pueden ocultar las líneas del pentagrama quitando el grabador `Staff_symbol_engraver` del contexto de `Staff`. Como alternativa se puede utilizar la instrucción `\stopStaff`.

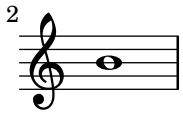
```
\new Staff \with {
  \remove Staff_symbol_engraver
}
\relative { a''8 f e16 d c b a2 }
```

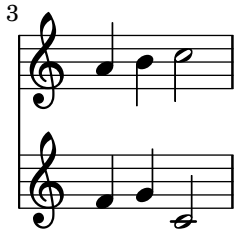


Se pueden ocultar los pentagramas vacíos (para hacer la que se conoce como ‘partitura a la francesa’) aplicando la instrucción `\RemoveEmptyStaves` sobre un contexto, lo cual se puede hacer globalmente (dentro del bloque `\layout`) así como para pautas específicas solamente (dentro de un bloque `\with`). Esta instrucción elimina todas las pautas vacías de la partitura excepto las del primer sistema. Si queremos ocultar también las del primer sistema, usamos `\RemoveAllEmptyStaves`.

```
\layout {
  \context {
    \Staff
    \RemoveEmptyStaves
  }
}

\relative <<
  \new Staff {
    e'4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
>>
```







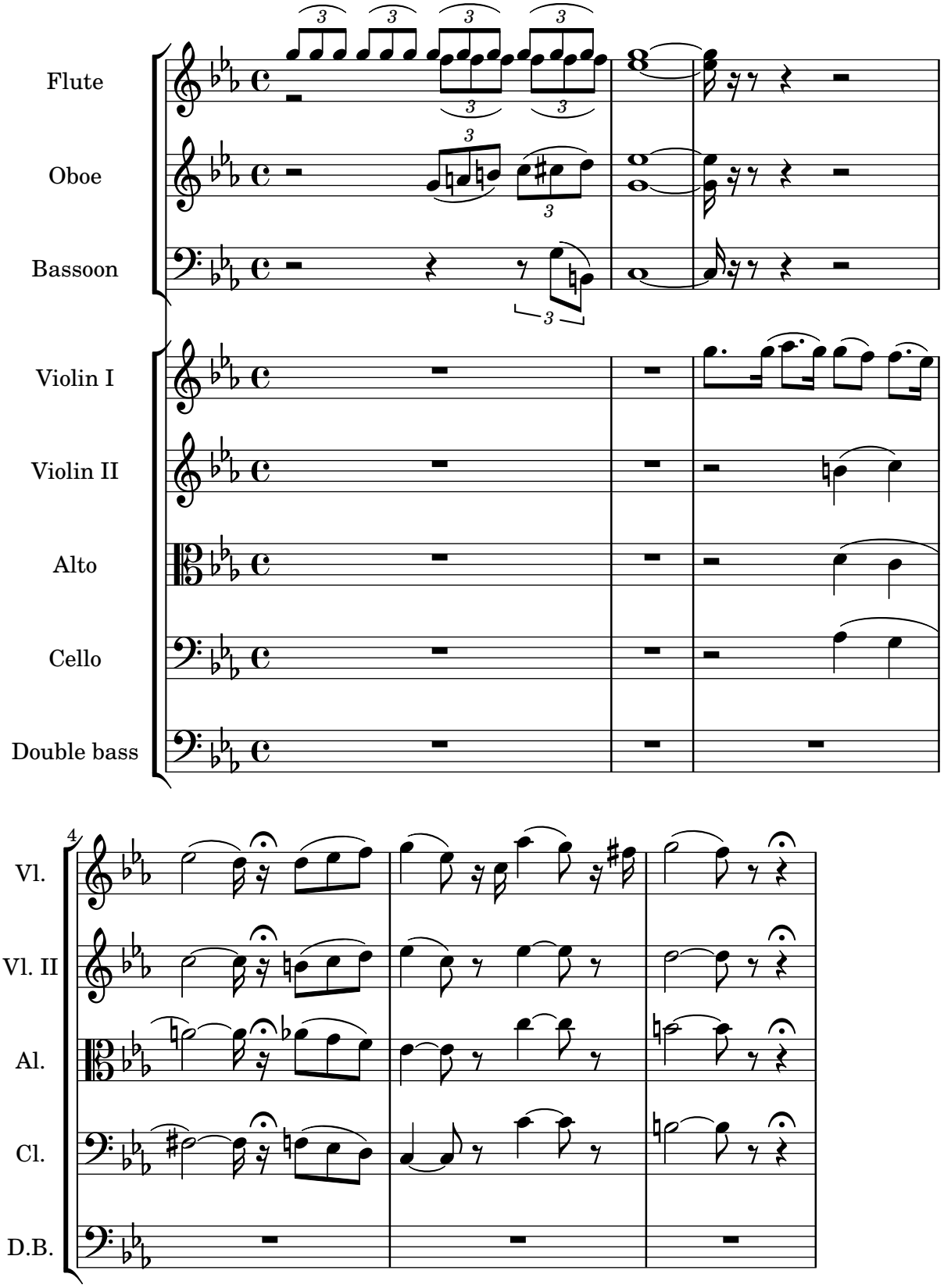
Se considera que un pentagrama está vacío cuando contiene solamente silencios multicompás, silencios normales, silencios oscultos o saltos, o una combinación de todos los elementos anteriores. *Cualquier otro* objeto musical (que hace que un pentagrama no se considere como vacío) está en la lista de la propiedad de contexto `keepAliveInterfaces`, como se establece inicialmente en el archivo `ly/engraver-init.ly`.

Tanto `\RemoveEmptyStaves` como `\RemoveAllEmptyStaves` son atajos predefinidos que establecen propiedades tales como `remove-empty` y `remove-first` para el objeto `VerticalAxisGroup`, como aparece explicado en Sección B.22 [Identificadores de modificación de contextos], página 968.

El grabador `Keep_alive_together_engraver` solo permite que grupos de pentagramas se puedan eliminar juntos y no individualmente. Por defecto, forma parte del contexto `PianoStaff`: una parte de piano solo se oculta cuando sus dos pentagramas están vacíos. De forma semejante, es una práctica común del tipografiado de las partituras orquestales eliminar grupos de pentagramas vacíos en lugar de pautas individuales; esto es posible conseguirlo añadiendo el grabador `Keep_alive_together_engraver` al contexto de agrupamiento de pentagramas adecuado, como aparece explicado en Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752, (consulte Sección 6.1.2 [Agrupar pentagramas], página 246, para ver los nombres de los contextos).

```
\layout {
  \context {
    \StaffGroup
    \RemoveEmptyStaves
    \consists Keep_alive_together_engraver
  }
}
```

En el ejemplo siguiente, los pentagramas de instrumentos de viento son eliminados en el segundo sistema; sin embargo, el contrabajo no se elimina porque forma parte de la sección de cuerda, que está tocando en ese momento.



El grabador `Keep_alive_together_engraver` usa internamente la propiedad `remove-layer` del `VerticalAxisGroup` de un pentagrama para decidir si imprimirlo o no, cuando se considera vacío. Dicha propiedad puede también establecerse directamente, en cuyo caso actúa como un índice de prioridad: los valores más cercanos a cero tienen precedencia frente a los números más

altos, y de esta forma los pentagramas cuyo `remove-layer` es más alto quedarán enmascarados en favor de los que tienen un número más bajo.

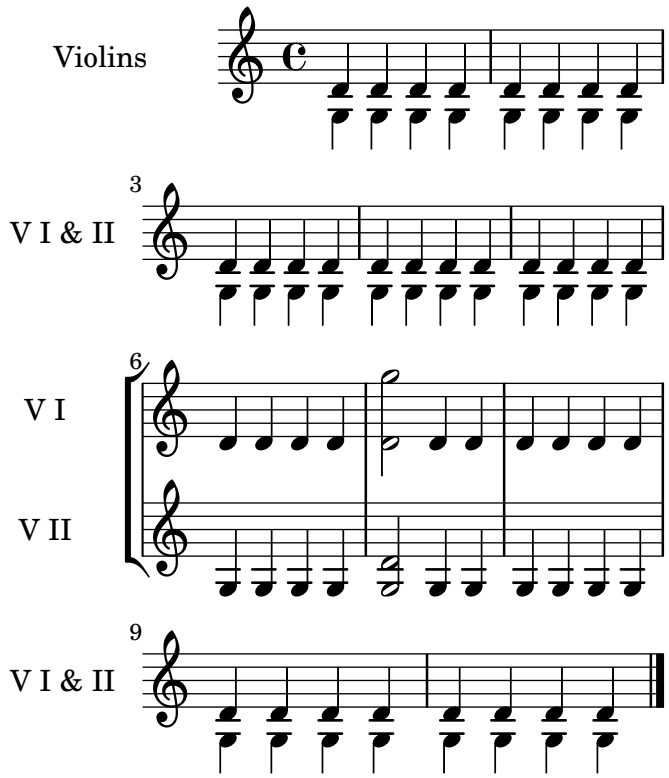
Esto es especialmente útil para los pentagramas en 'divisi', donde ciertas partes individuales (véase Sección 6.3 [Escritura de las particellas], página 265) a veces tienen que expandirse a más de un pentagrama. En el ejemplo siguiente, dos partes resultan dirigidas a un total de *tres* pentagramas; sin embargo, los tres pentagramas nunca se imprimen todos al mismo tiempo:

- en los primeros sistemas, solo uno de ellos se imprime, porque la propiedad `keepAliveInterfaces` se ha establecido a una lista vacía; por tanto, los otros dos pentagramas se considera que están vacíos y por ello ocultos, sin que importe su contenido;
- cuando la propiedad resulta des-establecida (volviendo así a su valor predeterminado), ya no impide que los otros dos pentagramas se impriman; sin embargo, como el valor de `remove-layer` de éstos es menor que el del pentagrama suelto, estos dos pentagramas se imprimen ahora en su lugar.

Tales sustituciones se aplican no solamente a notas, acordes y otros eventos musicales que tienen lugar inmediatamente después de cambiar el valor, sino a todo el sistema dentro del cual ocurre el cambio.

```
\layout {
  short-indent = 2\cm
  indent = 3\cm
  \context {
    \Staff
    keepAliveInterfaces = #'()
  }
}

violI = {
  \repeat unfold 24 { d'4 }
  \once \unset Staff.keepAliveInterfaces
  <d' g''>2
  \repeat unfold 14 { d'4 }
  \bar "|."
}

violII = {
  \repeat unfold 24 { g4 }
  <g d'>2
  \repeat unfold 14 { g4 }
  \bar "|."
}

\new StaffGroup \with { \consists Keep_alive_together_engraver } <<
  \new Staff \with {
    instrumentName = "Violins"
    shortInstrumentName = "V I & II"
    \override VerticalAxisGroup.remove-layer = 2
  } <<  \violI \\ \violII  >>
  \new Staff \with {
    instrumentName = "Violin I"
    shortInstrumentName = "V I"
    \RemoveAllEmptyStaves
    \override VerticalAxisGroup.remove-layer = 1
```

```
  } \violI
  \new Staff \with {
    instrumentName = "Violin II"
    shortInstrumentName = "V II"
    \RemoveAllEmptyStaves
    \override VerticalAxisGroup.remove-layer = 1
  } \violII
>>
```



También se puede usar `\RemoveAllEmptyStaves` para crear secciones de ossia para un pentagrama. Para ver más detalles, consulte Sección 6.2.2 [Pentagramas de Ossia], página 256.

Instrucciones predefinidas

`\RemoveEmptyStaves`, `\RemoveAllEmptyStaves`.

Véase también

Glosario musical: Sección "Frenched staff" en *Glosario Musical*.

Manual de aprendizaje: Sección "Visibilidad y color de los objetos" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 33.5 [Cambiar los valores por omisión de los contextos], página 753, Sección 6.2.1 [El símbolo del pentagrama], página 252, Sección 6.2.2 [Pentagramas de Ossia], página 256, Sección 7.1.4 [Notas ocultas], página 291, Sección 2.2.2 [Silencios invisibles], página 70, Sección 36.7 [Visibilidad de los objetos], página 794, Sección B.22 [Identificadores de modificación de contextos], página 968, Sección 6.1.2 [Agrupar pentagramas], página 246, Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752.

Archivos instalados: `ly/engraver-init.ly`.

Fragmentos de código: Sección "Notación de los pentagramas" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "ChordNames" en *Referencia de Funcionamiento Interno*, Sección "FiguredBass" en *Referencia de Funcionamiento Interno*, Sección "Lyrics"

en *Referencia de Funcionamiento Interno*, Sección "Staff" en *Referencia de Funcionamiento Interno*, Sección "VerticalAxisGroup" en *Referencia de Funcionamiento Interno*, Sección "Staff_symbol_engraver" en *Referencia de Funcionamiento Interno*, Sección "Axis_group_engraver" en *Referencia de Funcionamiento Interno*, Sección "Keep_alive_together_engraver" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La eliminación del grabador `Staff_symbol_engraver` también oculta las barras de compás. Si se fuerza la visibilidad de la barra de compás, pueden ocurrir errores de formato visual. En este caso, utilice las siguientes sobreescrituras de valores en vez de quitar el grabador:

```
\omit StaffSymbol
\override NoteHead.no-ledgers = ##t
```

Para ver los fallos y problemas conocidos, así como las advertencias asociadas con `\RemoveEmptyStaves`, consulte Sección 33.5 [Cambiar los valores por omisión de los contextos], página 753.

6.3 Escritura de las particellas

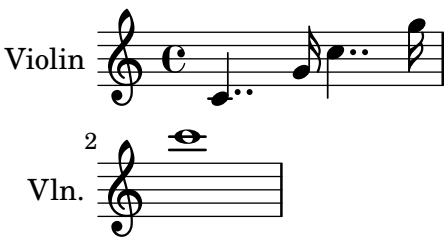
Esta sección explica cómo preparar el papel de atril, o particellas, para música orquestal o de conjunto, lo que frecuentemente requiere insertar nombres de instrumentos en la partitura. También se describen métodos para citar otras voces y dar formato a las notas guía, así como una manera de contraer varios compases vacíos consecutivos dentro de las partes individuales.

Además, puede verse un método para la impresión de los pentagramas de *divisi*, que se usan a veces en las particellas, en Sección 6.2.3 [Ocultar pentagramas], página 260.

6.3.1 Nombres de instrumentos

Se pueden imprimir los nombres de los instrumentos en el lado izquierdo de los pentagramas dentro de los contextos `Staff`, `PianoStaff`, `StaffGroup`, `GrandStaff` y `ChoirStaff`. El valor de `instrumentName` se usa para el primer pentagrama, y el valor de `shortInstrumentName` se usa para todos los pentagramas siguientes.

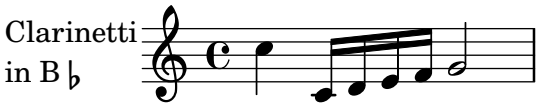
```
\new Staff \with {
  instrumentName = "Violin "
  shortInstrumentName = "Vln. "
} \relative {
  c'4.. g'16 c4.. g'16 \break | c1 |
}
```



También podemos usar `\markup` para construir nombres de instrumento más complicados:

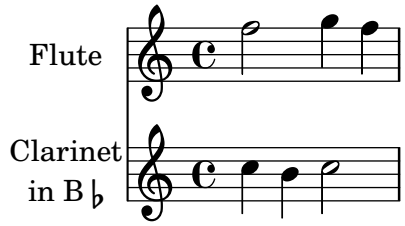
```
\new Staff \with {
  instrumentName = \markup {
    \column { "Clarinetti"
      \line { "in B" \smaller \flat }
    }
  }
```

```
} \relative {
  c''4 c,16 d e f g2
}
```



Cuando se agrupan dos o más contextos de pentagrama, los nombres de instrumento y los nombres cortos aparecen centrados de forma predeterminada. Para centrar nombres de instrumento de varias líneas, se debe utilizar `\center-column`:

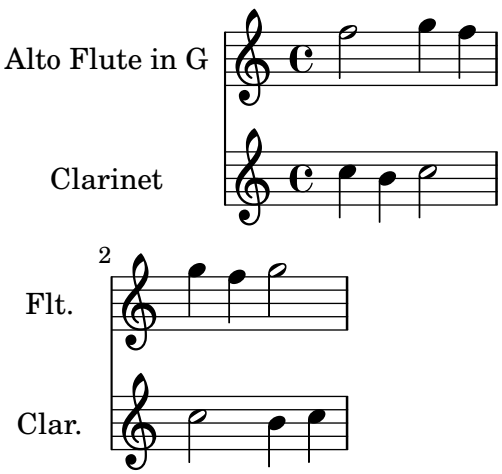
```
<<
  \new Staff \with {
    instrumentName = "Flute"
  } \relative {
    f''2 g4 f
}
  \new Staff \with {
    instrumentName = \markup {
      \center-column { "Clarinet"
        \line { "in B" \smaller \flat }
      }
    }
  } \relative { c''4 b c2 }
>>
```



Sin embargo, si los nombres de instrumento son más largos, éstos no se centran para un grupo de pentagramas a no ser que se aumenten los valores del sangrado, `indent`, y del sangrado corto, `short-indent`. Para ver más detalles sobre estos ajustes, consulte Sección 26.5.3 [Variables de \paper para desplazamientos y sangrados], página 681.

```
<<
  \new Staff \with {
    instrumentName = "Alto Flute in G"
    shortInstrumentName = "Flt."
  } \relative {
    f''2 g4 f \break
    g4 f g2
  }
  \new Staff \with {
    instrumentName = "Clarinet"
    shortInstrumentName = "Clar."
  } \relative {
    c''4 b c2 \break
    c2 b4 c
  }
```

```
>>

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}
```

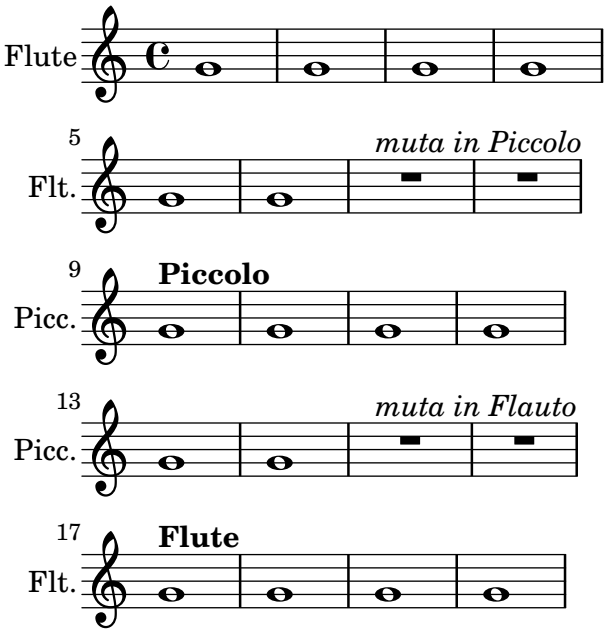


Para añadir nombres de instrumento a otros contextos (como ChordNames o FiguredBass), debemos añadir el grabador Instrument_name_engraver a dicho contexto. Para ver más detalles, consulte Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752.

El nombre de instrumento corto shortInstrumentName se puede cambiar en mitad de una pieza, igual que otros ajustes que pueden ser necesarios para el instrumento nuevo; sin embargo, sólo el primer instrumentName se imprime y el resto de ellos se ignoran:

```
prepPiccolo = <>^\markup \italic { muta in Piccolo }

prepFlute = <>^\markup \italic { muta in Flauto }

setPiccolo = {
  <>^\markup \bold { Piccolo }
  \transposition c''
}

setFlute = {
  <>^\markup \bold { Flute }
  \transposition c'
}

\new Staff \with {
  instrumentName = "Flute"
  shortInstrumentName = "Flt."
}
\relative {
  g'1 g g g \break
  g1 g \prepPiccolo R R \break
  \set Staff.instrumentName = "Piccolo"
  \set Staff.shortInstrumentName = "Picc."
  \setPiccolo
  g1 g g g \break
```

```
  g1 g \prepFlute R R \break
  \set Staff.instrumentName = "Flute"
  \set Staff.shortInstrumentName = "Flt."
  \setFlute
  g1 g g g
}
```



Véase también

Referencia de la notación: Sección 26.5.3 [Variables de \paper para desplazamientos y sangrados], página 681, Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752.

Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “InstrumentName” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*.

6.3.2 Citar otras voces

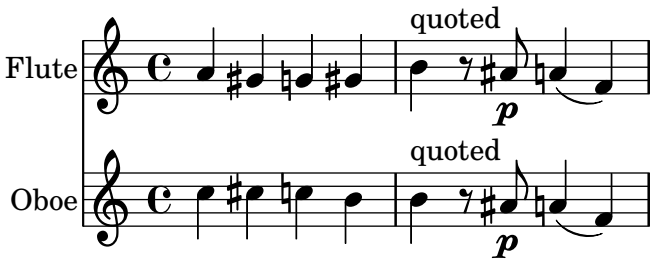
Es muy frecuente que una voz use las mismas notas que otra voz. Por ejemplo, los violines primero y segundo tocando la misma frase durante un determinado pasaje musical. Esto se hace dejando que una voz *cite* a la otra, sin tener que volver a introducir la música para la segunda voz.

La instrucción \addQuote, utilizada en el ámbito del nivel sintáctico superior, define un flujo de música desde el que es posible citar fragmentos.

La instrucción \quoteDuring se usa para indicar el punto en que comienza la cita. Va seguida por dos argumentos: el nombre de la voz citada, tal y como se definió con \addQuote, y una expresión musical que indica la duración de la cita.

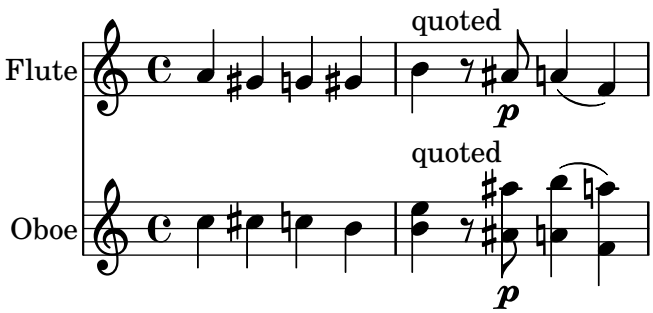
```
fluteNotes = \relative {
  a'4 gis g gis | b4^"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "flute" { s1 }
}
```

```
\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Si la expresión musical que se usa para \quoteDuring contiene cualquier cosa que no sea un silencio de separación o un silencio multicompás, se produce una situación de polifonía, lo que no suele ser deseable:

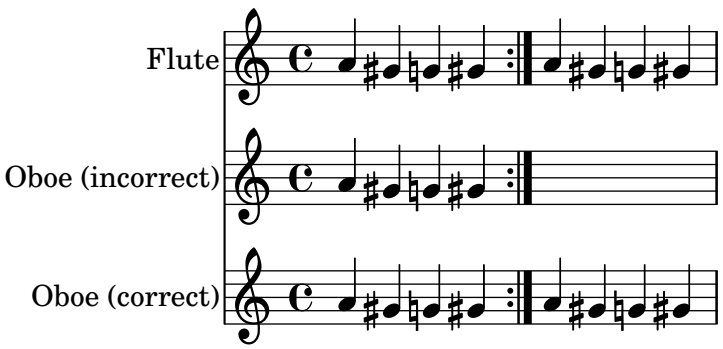
```
fluteNotes = \relative {
  a'4 gis g gis | b4^"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "flute" { e4 r8 ais b4 a }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Si una instrucción \unfoldRepeats dentro de una expresión musical se necesita imprimir al usar \quoteDuring, entonces debe contener también su propia instrucción \unfoldRepeats;

```
fluteNotes = \relative {
  \repeat volta 2 { a'4 gis g gis }
}
```

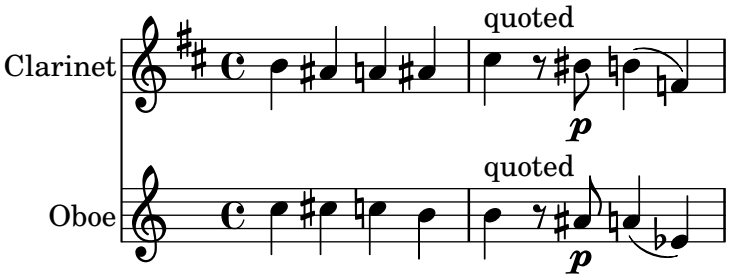
```
oboeNotesDW = \relative {
  \repeat volta 2 \quoteDuring "incorrect" { s1 }
}

oboeNotesW = \relative {
  \repeat volta 2 \quoteDuring "correct" { s1 }
}


\addQuote "incorrect" { \fluteNotes }

\addQuote "correct" { \unfoldRepeats \fluteNotes }

\score {
  \unfoldRepeats
  <<
    \new Staff \with { instrumentName = "Flute" }
    \fluteNotes
    \new Staff \with { instrumentName = "Oboe (incorrect)" }
    \oboeNotesDW
    \new Staff \with { instrumentName = "Oboe (correct)" }
    \oboeNotesW
  >>
}
```



La instrucción \quoteDuring usa los ajustes de \transposition tanto de la parte citada como de la que cita, para producir notas que tienen la misma altura de sonido en la parte que cita y en las de la parte citada.

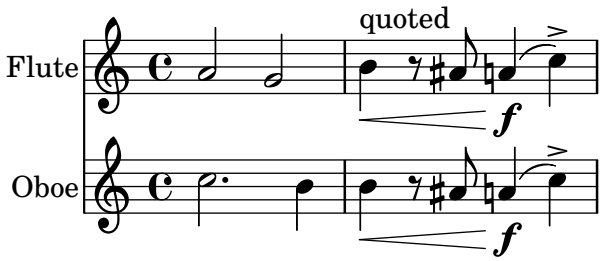
```
clarinetNotes = \relative c'' {
  \transposition bes
  \key d \major
  b4 ais a ais | cis4^"quoted" r8 bis\p b4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "clarinet" { s1 }
}

\addQuote "clarinet" { \clarinetNotes }
```

```
\score {
  <<
    \new Staff \with { instrumentName = "Clarinet" } \clarinetNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



De forma predeterminada, la música citada incluye todas las articulaciones, matices dinámicos, elementos de marcado, etc. de la expresión citada. Es posible elegir cuáles de estos objetos de la música que se cita se imprimen, mediante la propiedad de contexto `quotedEventTypes`.

```
fluteNotes = \relative {
  a'2 g2 |
  b4\<^"quoted" r8 ais a4\f( c->)
 }

oboeNotes = \relative {
  c''2. b4 |
  \quoteDuring "flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \set Score.quotedEventTypes = #'(note-event articulation-event
                                     crescendo-event rest-event
                                     slur-event dynamic-event)
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
 }
```



Las citas también se pueden etiquetar, véase Sección 22.2.2 [Uso de etiquetas], página 630.

Véase también

Referencia de la notación: Sección 1.3.4 [Transposición de los instrumentos], página 30, Sección 22.2.2 [Uso de etiquetas], página 630.

Archivos de inicio: `scm/define-event-classes.scm`.

Fragmentos de código: Sección "Notación de los pentagramas" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Music classes" en *Referencia de Funcionamiento Interno*, Sección "QuoteMusic" en *Referencia de Funcionamiento Interno*, Sección "Voice" en *Referencia de Funcionamiento Interno*.

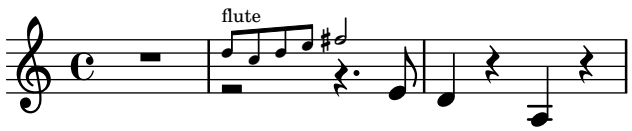
Advertencias y problemas conocidos

Solo el contenido de la primera voz de una instrucción `\addQuote` se tiene en cuenta para la cita, de manera que si la expresión musical contiene enunciados `\new` o `\context Voice`, sus contenidos no serían citados. La cita de notas de adorno no está contemplada y puede hacer que LilyPond termine de forma abrupta; la cita de unos tresillos dentro de otros puede dar como resultado una notación de pobre calidad.

6.3.3 Formateo de las notas guía

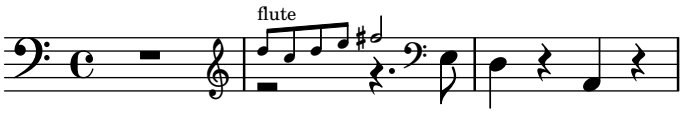
La manera más sencilla de dar formato a las notas guía es crear explícitamente un contexto `CueVoice` dentro de la parte.

```
\relative {
  R1
  <<
    { e'2\rest r4. e8 }
    \new CueVoice {
      \stemUp d'8^"flute" c d e fis2
    }
  >>
  d,4 r a r
}
```



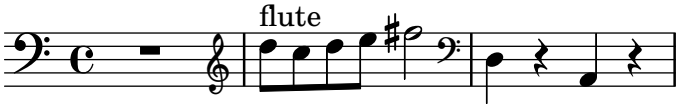
La instrucción `\cueClef` puede usarse también con un contexto `CueVoice` explícito si se requiere un cambio de clave, e imprime una clave del tamaño adecuado para las notas guía. Después puede utilizarse la instrucción `\cueClefUnset` para volver a la clave original, de nuevo con un signo de clave del tamaño adecuado.

```
\relative {
  \clef "bass"
  R1
  <<
    { e'2\rest r4. \cueClefUnset e,8 }
    \new CueVoice {
      \cueClef "treble" \stemUp d''8^"flute" c d e fis2
    }
  >>
  d,,4 r a r
}
```



Las instrucciones \cueClef y \cueClefUnset se pueden usar también sin un CueVoice si es necesario.

```
\relative {
  \clef "bass"
  R1
  \cueClef "treble"
  d''8^"flute" c d e fis2
  \cueClefUnset
  d,,4 r a r
}
```



En casos de colocación más compleja de notas guía, por ejemplo con transposición o insertando notas guía procedentes de más de una fuente, pueden usarse las instrucciones \cueDuring o \cueDuringWithClef. Son una forma más especializada de \quoteDuring, véase Sección 6.3.2 [Citar otras voces], página 268, en la sección anterior.

La sintaxis es

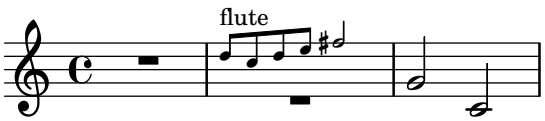
```
\cueDuring nombre_de_la_cita dirección música
```

y

```
\cueDuringWithClef nombre_de_la_cita dirección clave música
```

La música procedente de los compases correspondientes del elemento *nombre_de_la_cita* se añade como un contexto CueVoice y ocurre simultáneamente con la *música*, lo que produce una situación polifónica. La variable *dirección* toma un argumento #UP (arriba) o #DOWN (abajo), y corresponde a las voces primera y segunda, respectivamente, determinando cómo se imprimen las notas guía en relación a la otra voz.

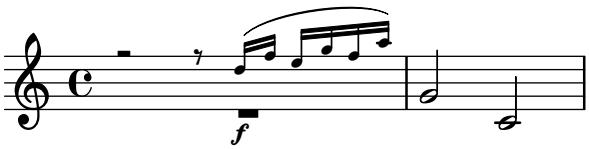
```
fluteNotes = \relative {
  r2. c''4 | d8 c d e fis2 | g2 d |
}

oboeNotes = \relative c'' {
  R1
  <>^\markup \tiny { flute }
  \cueDuring "flute" #UP { R1 }
  g2 c,
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \oboeNotes
}
```



Es posible ajustar qué aspectos de la música se citan con `\cueDuring` estableciendo el valor de la propiedad `quotedCueEventTypes`. Su valor por omisión es `'(note-event rest-event tie-event beam-event tuplet-span-event)`, lo que significa que solamente las notas, silencios, ligaduras de unión, barras y grupos especiales se citan, pero no las articulaciones, marcas dinámicas, elementos de marcado, etc.

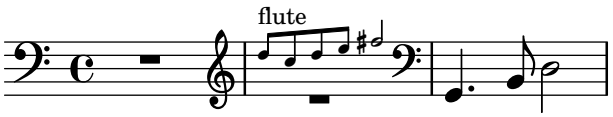
> **Nota:** Cuando un contexto `Voice` da comienzo con `\cueDuring`, como en el ejemplo siguiente, el contexto `Voice` se debe declarar explícitamente, pues en caso contrario toda la expresión musical pertenecería al contexto `CueVoice`.

```
oboeNotes = \relative {
  r2 r8 d''16(\f f e g f a)
  g8 g16 g g2.
}
\addQuote "oboe" { \oboeNotes }

\new Voice \relative c'' {
  \set Score.quotedCueEventTypes = #'(note-event rest-event tie-event
                                      beam-event tuplet-span-event
                                      dynamic-event slur-event)
  \cueDuring "oboe" #UP { R1 }
  g2 c,
}
```



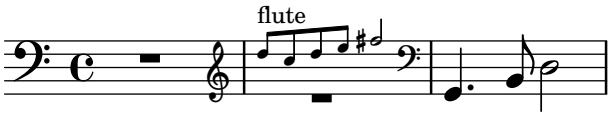
Se pueden usar elementos de marcado para mostrar el nombre del instrumento citado. Si las notas guía requieren un cambio de clave, puede hacerse manualmente pero la clave original se debe restaurar al final de las notas guía.

```
fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \clef treble
  <>^\markup \tiny { flute }
  \cueDuring "flute" #UP { R1 }
  \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}
```



De forma alternativa, puede usarse la función \cueDuringWithClef en su lugar. Esta instrucción admite un argumento adicional para especificar el cambio de clave que se necesita imprimir para las notas guía, pero después imprime automáticamente la clave original una vez que ha finalizado la serie de notas guía.

```
fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  <>^\markup { \tiny "flute" }
  \cueDuringWithClef "flute" #UP "treble" { R1 }
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}
```

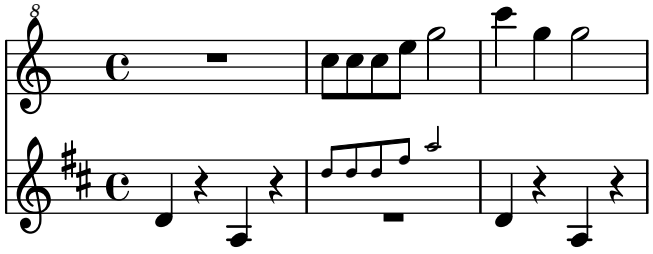


Como \quoteDuring, \cueDuring tiene en cuenta las transposiciones instrumentales. Las notas guía se producen en las alturas en que se escribirían para el instrumento que recibe la cita, para así producir las alturas de sonido del instrumento original.

Para transportar las notas guía de forma diferente, use \transposedCueDuring. Esta instrucción acepta un argumento adicional para especificar (en modo absoluto) la altura impresa con que queremos representar el sonido de un Do central de concierto. Esto es útil para extraer citas de un instrumento que está en un registro completamente diferente.

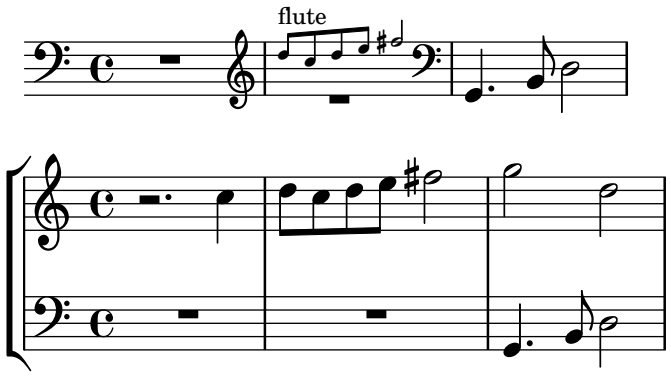
```
piccoloNotes = \relative {
  \clef "treble^8"
  R1
  c'''8 c c e g2
  c4 g g2
}

bassClarinetNotes = \relative c' {
  \key d \major
  \transposition bes,
  d4 r a r
  \transposedCueDuring "piccolo" #UP d { R1 }
  d4 r a r
}
```

```
\addQuote "piccolo" { \piccoloNotes }

<<
  \new Staff \piccoloNotes
  \new Staff \bassClarinetNotes
>>
```



La instrucción \killCues elimina las notas guía de una expresión musical, de forma que la misma expresión musical pueda utilizarse para producir la particella instrumental (con notas guía) y la partitura de conjunto. La instrucción \killCues elimina solamente las notas y eventos que se han citado mediante \cueDuring. Otros elementos de marcado asociados con las guías, como los cambios de clave o una etiqueta identificativa del instrumento fuente, pueden marcarse para su inclusión selectiva dentro de la partitura; véase Sección 22.2.2 [Uso de etiquetas], página 630.

```
fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \tag #'part {
    \clef treble
    <>^\markup \tiny { flute }
  }
  \cueDuring "flute" #UP { R1 }
  \tag #'part \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

\new StaffGroup <<
  \new Staff {
    \fluteNotes
  }
  \new Staff {
    \removeWithTag #'part { \killCues { \bassoonNotes } }
  }
>>
```



Véase también

Referencia de la notación: Sección 6.3.2 [Citar otras voces], página 268, Sección 1.3.4 [Transposición de los instrumentos], página 30, Sección 6.3.1 [Nombres de instrumentos], página 265, Sección 1.3.1 [Clave], página 20, Sección 9.6.3 [Guías musicales], página 397, Sección 22.2.2 [Uso de etiquetas], página 630.

Fragmentos de código: Sección "Notación de los pentagramas" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "CueVoice" en *Referencia de Funcionamiento Interno*, Sección "Voice" en *Referencia de Funcionamiento Interno*.

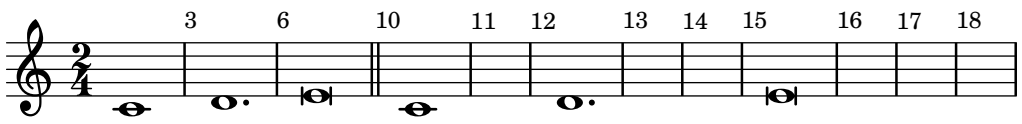
Advertencias y problemas conocidos

Pueden ocurrir colisiones con los silencios si se usa `\cueDuring`, entre los contextos de `Voice` y de `CueVoice`. Al usar `\cueDuringWithClef` o `\transposedCueDuring` el argumento adicional requerido para cada caso debe ir después de la cita y de la dirección.

6.3.4 Compresión de los compases vacíos

De forma predeterminada, todos los compases se imprimen aunque estén vacíos (cosa que puede ocurrir si un evento rítmico, como notas, silencios o saltos invisibles, es tan largo que abarca varios compases. Este comportamiento se puede cambiar contrayendo todos los compases vacíos en uno solo, como se ilustra aquí (la segunda parte del ejemplo, con compases expandidos, en realidad vuelve de nuevo al comportamiento predeterminado):

```
\override Score.BarNumber.break-visibility = ##(#f #t #t)
\time 2/4
\compressEmptyMeasures
c'1 d'1. e'\breve
\bar "||"
\expandEmptyMeasures
c'1 d'1. e'\breve
```

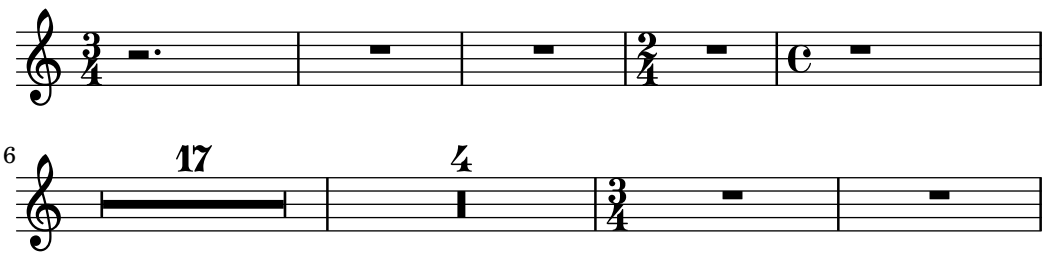


Aunque esta notación es correcta sintácticamente, puede dar lugar a confusión desde el punto de vista musical, como queda ilustrado en el ejemplo anterior; de ahí la necesidad de imprimir explícitamente los números de compás, usando la sintaxis que se describe en Sección 36.7.4 [Uso de break-visibility (visibilidad en el salto)], página 796.

Donde esta notación puede ser más útil es cuando se aplica a los Sección 2.2.3 [Silencios de compás completo], página 72. Un silencio multicompás se imprimirá entonces como un solo

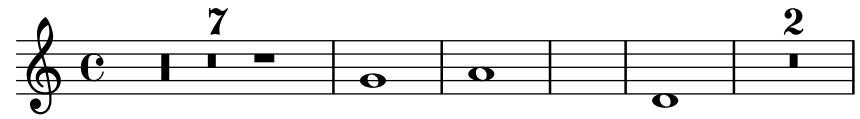
compás que contiene un símbolo de silencio multicompás, con el número de compases de duración de este silencio impreso encima del compás:

```
% Default behavior
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Rest measures contracted to single measure
\compressEmptyMeasures
r1 | R1*17 | R1*4 |
\expandEmptyMeasures
% Rest measures expanded again
\time 3/4
R2.*2 |
```



A diferencia de `\compressEmptyMeasures`, la función musical `\compressMMRests` solo se aplica a los silencios, dejando sin comprimir todos los demás eventos. Dado que es una función, y no un ajuste del valor de una propiedad, su sintaxis difiere ligeramente en que debe ir seguida de una expresión musical:

```
\compressMMRests {
  % Rests are compressed...
  R1*7
  % ... but notes can still span multiple measures.
  g'1 a'1*2 d'1
  R1*2
}
```



Todas las instrucciones descritas en esta sección, descansan en realidad en la propiedad interna `skipBars`, que se establece dentro del contexto de `Score`, véase Sección 35.2 [Las instrucciones \set y \unset], página 768.

Instrucciones predefinidas

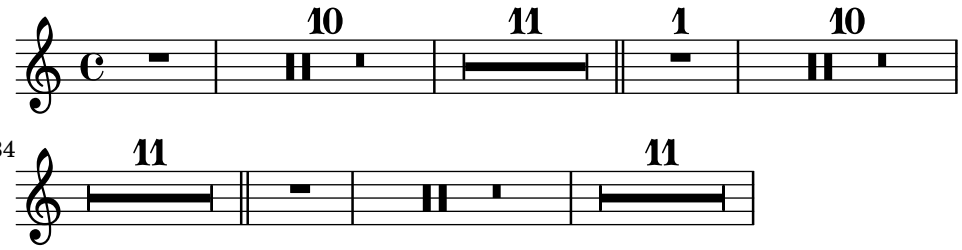
`\compressEmptyMeasures`, `\expandEmptyMeasures`, `\compressMMRests`.

Fragmentos de código seleccionados

Numerar silencios seguidos de un compás

Los silencios de varios compases muestran su duración con un número excepto los que tienen un solo compás. Esto se puede cambiar fijando el valor de `restNumberThreshold`.

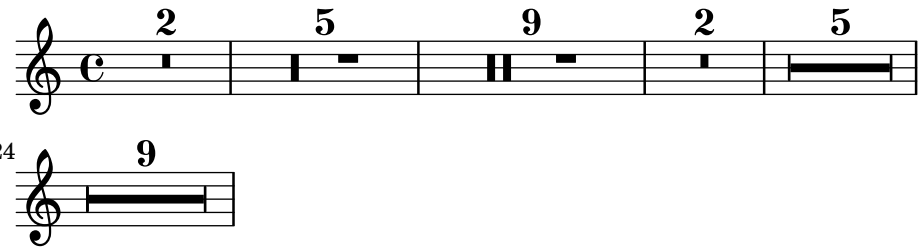
```
{
  \compressEmptyMeasures
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 0
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 10
  R1 R1*10 R1*11
}
```



Cambiar la forma de los silencios multicompás

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y longa (conocidos en alemán como "Kirchenpausen", «silencios eclesiásticos»); en caso contrario se muestra una barra gruesa y larga. Este valor predeterminado de diez se puede cambiar sobreescribiendo la propiedad `expand-limit`:

```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = 3
    R1*2 | R1*5 | R1*9
  }
}
```

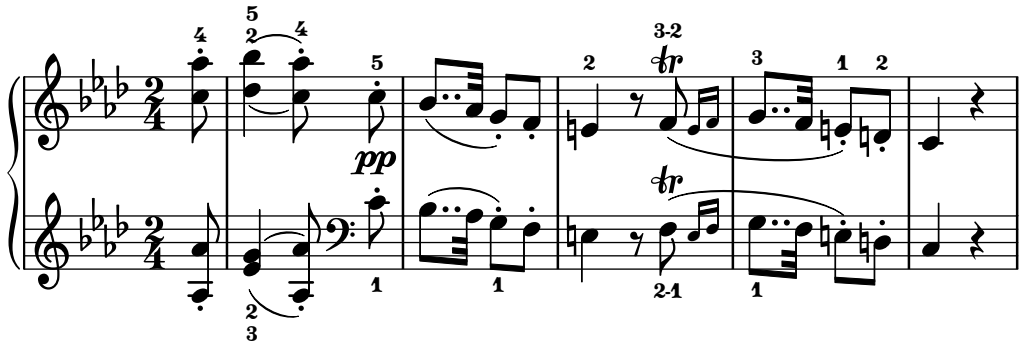


Véase también

Referencia de la notación: Sección 36.7.4 [Uso de break-visibility (visibilidad en el salto)], página 796, Sección 2.2.3 [Silencios de compás completo], página 72, Sección 35.2 [Las instrucciones `\set` y `\unset`], página 768.

Referencia de funcionamiento interno: Sección "MultiMeasureRest" en *Referencia de Funcionamiento Interno*, Sección "MultiMeasureRestNumber" en *Referencia de Funcionamiento Interno*, Sección "MultiMeasureRestScript" en *Referencia de Funcionamiento Interno*, Sección "MultiMeasureRestText" en *Referencia de Funcionamiento Interno*.

7 Anotaciones editoriales



Esta sección trata de las diversas maneras de modificar el aspecto de las notas y de aplicar énfasis analítico o educativo.

7.1 Dentro del pentagrama

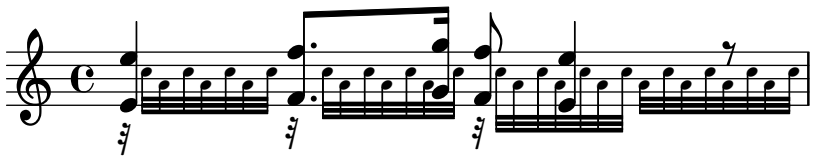
Esta sección trata sobre cómo aplicar énfasis a los elementos situados dentro del pentagrama.

7.1.1 Seleccionar el tamaño de la tipografía para la notación

> **Nota:**
> Para los tamaños de fuente del texto Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327.
> Para el tamaño de la pauta, véase Sección 27.2 [Establecer el tamaño del pentagrama], página 688.
> Para las notas guía, véase Sección 6.3.3 [Formateo de las notas guía], página 272.
> Para los pentagramas de ossia, véase Sección 6.2.2 [Pentagramas de Ossia], página 256.

Para modificar el tamaño de la notación sin cambiar el tamaño del pentagrama, especifique un factor de ampliación con la instrucción `\magnifyMusic`:

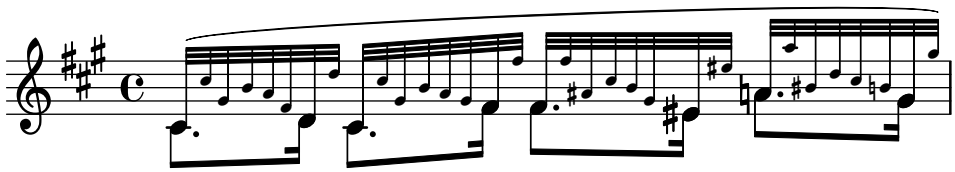
```
\new Staff <<
  \new Voice \relative {
    \voiceOne
    <e' e'>4 <f f'>8. <g g'>16 <f f'>8 <e e'>4 r8
  }
  \new Voice \relative {
    \voiceTwo
    \magnifyMusic 0.63 {
      \override Score.SpacingSpanner.spacing-increment = #(* 1.2 0.63)
      r32 c'' a c a c a c r c a c a c a c
      r c a c a c a c a c a c a c a c
    }
  }
>>
```



El \override del ejemplo anterior es una solución provisional para solventar un fallo del programa. Consulte el apartado de "Problemas y advertencias conocidos" al final de esta sección.

Si una figura con un tamaño normal de cabeza se mezcla con otra más pequeña, el tamaño de la menor podría necesitar reiniciarse (con \once \normalsize) para que las plicas y las alteraciones accidentales mantengan una buena alineación:

```
\new Staff <<
  \key fis \minor
  \mergeDifferentlyDottedOn
  \new Voice \relative {
    \voiceOne
    \magnifyMusic 0.63 {
      \override Score.SpacingSpanner.spacing-increment =
        #(* 1.2 0.63)

      \once \normalsize cis'32( cis' gis b a fis
        \once \normalsize d d'
      \once \normalsize cis, cis' gis b a gis
        \once \normalsize fis fis'
      \once \normalsize fis, fis' ais, cis b gis
        \once \normalsize eis eis'
      \once \normalsize a, a' bis, d cis b
        \once \normalsize gis gis')
    }
  }
  \new Voice \relative {
    \voiceTwo
    cis'8. d16 cis8. fis16 fis8. eis16 a8. gis16
  }
>>
```



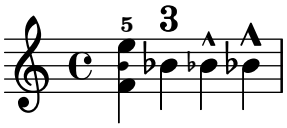
La instrucción \magnifyMusic no está pensada para las notas guía, de adorno, o para los pentagramas de Ossia; existen métodos más apropiados para la introducción de cada uno de esos elementos. En vez de ello, es útil cuando el tamaño de la notación cambia dentro de una sola parte instrumental sobre un pentagrama, y cuando las notas de adorno no son adecuadas, como en pasajes de tipo cadencial o en casos como los de los ejemplos anteriores. Establecer el valor de \magnifyMusic a 0.63 duplica las dimensiones del contexto CueVoice.

> **Nota:** La instrucción \magnifyMusic *no* debe utilizarse si se está cambiando el tamaño del pentagrama al mismo tiempo. Véase Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

Cambiar el temaño de los objetos individuales

Un objeto de presentación individual se puede cambiar de tamaño usando las instrucciones `\tweak` u `\override` para ajustar su propiedad `font-size`:

```
\relative {
  % resize a note head
  <f' \tweak font-size -4 b e>-5
  % resize a fingering
  bes-\tweak font-size 0 -3
  % resize an accidental
  \once \override Accidental.font-size = -4 bes!-^
  % resize an articulation
  \once \override Script.font-size = 4 bes!-^
}
```

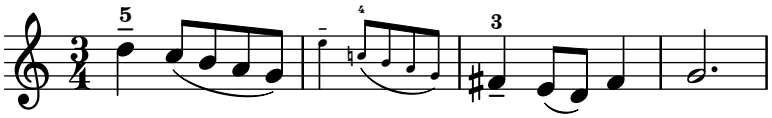


El valor predeterminado de `font-size` para cada objeto de presentación está relacionado en el Manual de Referencia de funcionamiento interno. La propiedad `font-size` solamente se puede fijar para los objetos de presentación que contemplan la interfaz de presentación `font-interface`. Si no está especificado `font-size` en la lista de 'Ajustes estándar' del objeto, su valor es 0. Véase Sección "All layout objects" en *Referencia de Funcionamiento Interno*.

Comprensión de la propiedad `fontSize`

La propiedad de contexto `fontSize` ajusta el tamaño relativo de todos los elementos de notación basados en glifos dentro de un contexto.[1]

```
\relative {
  \time 3/4
  d''4---5 c8( b a g) |
  \set fontSize = -6
  e'4-- c!8-4( b a g) |
  \set fontSize = 0
  fis4---3 e8( d) fis4 |
  g2.
}
```



El valor de `fontSize` es un número que indica el tamaño relativo al tamaño estándar para la altura actual del pentagrama. El valor predeterminado de `fontSize` es 0; la adición de 6 a cualquier valor de `fontSize` duplica el tamaño impreso de los glifos, y la sustracción de 6 reduce el tamaño a la mitad. Cada paso aumenta el tamaño en un 12% aproximadamente.

La función de Scheme `magnification->font-size` se provee por conveniencia, pues las unidades logarítmicas de la propiedad `font-size` no son totalmente intuitivas. Por ejemplo, para ajustar la notación musical al 75% del tamaño predeterminado, usamos:

```
\set fontSize = #(magnification->font-size 0.75)
```

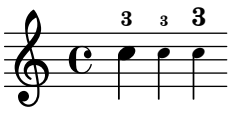
[1] Observe las palabras 'basado en glifos': una plica, por ejemplo, no es un glifo sino que se construye directamente por parte de LilyPond con líneas y curvas; por lo tanto, no resulta afectado. Lo mismo vale para objetos similares como ligaduras o barras de corchea.

La función de Scheme `magstep` tiene el efecto opuesto: convierte un valor de `font-size` en un factor de ampliación.

La propiedad `fontSize` no afecta solamente a los elementos notacionales que se dibujan con glifos, como la cabeza de las figuras, las alteraciones accidentales, las inscripciones textuales, etc. No afecta al tamaño de la propia pauta, ni cambia la escala de las plicas, barras, o espaciado horizontal. Para cambiar la escala de las plicas, barras y el espaciado horizontal junto con el tamaño de la notación (sin cambiar el tamaño de la pauta), use la instrucción `\magnifyMusic` que se ha explicado anteriormente. Para cambiar la escala del tamaño de todo, incluida la pauta, véase Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

Siempre que se establece el valor de la *propiedad de contexto* `fontSize`, su valor se añade al valor de la *propiedad de objeto gráfico* `font-size` para los objetos de presentación individuales, antes de que se imprima cualquiera de los glifos. Esto puede causar confusión cuando se fija el valor de las propiedades `font-size` individuales mientras `fontSize` ya está fijado:

```
% the default font-size for NoteHead is 0
% the default font-size for Fingering is -5
c''4-3

\set fontSize = -3
% the effective font size for NoteHead is now -3
% the effective font size for Fingering is now -8
c''4-3

\override Fingering.font-size = 0
% the effective font size for Fingering is now -3
c''4-3
```

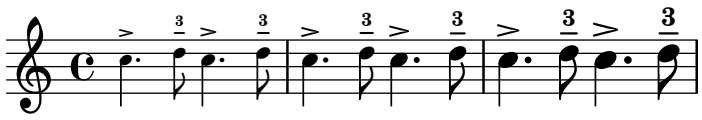


Las siguientes instrucciones de atajo también están disponibles:

Instrucción	Equivalente a	Tamaño relativo
`\teeny`	`\set fontSize = -3`	71%
`\tiny`	`\set fontSize = -2`	79%
`\small`	`\set fontSize = -1`	89%
`\normalsize`	`\set fontSize = 0`	100%
`\large`	`\set fontSize = 1`	112%
`\huge`	`\set fontSize = 2`	126%

```
\relative c'' {
  \teeny
  c4.-> d8---3
  \tiny
  c4.-> d8---3
  \small
  c4.-> d8---3
  \normalsize
  c4.-> d8---3
  \large
  c4.-> d8---3
  \huge
  c4.-> d8---3
```

```
}
```



Los cambios en el tamaño de la fuente se obtienen por medio del escalado del tamaño del diseño que se encuentra más cerca del tamaño deseado. El tamaño estándar para la tipografía (para `font-size = 0`) depende de la altura estándar del pentagrama. Para un pentagrama de 20 pt, se selecciona una fuente de 11 pt.

Instrucciones predefinidas

`\magnifyMusic`, `\teeny` (enano), `\tiny` (muy pequeño), `\small` (pequeño), `\normalsize` (normal), `\large` (grande), `\huge` (enorme).

Véase también

Notation Reference: Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327, Sección 27.2 [Establecer el tamaño del pentagrama], página 688, Sección 6.3.3 [Formateo de las notas guía], página 272, Sección 6.2.2 [Pentagramas de Ossia], página 256.

Installed Files: `ly/music-functions-init.ly`, `ly/property-init.ly`.

Fragmentos de código: Sección "Anotaciones editoriales" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "font-interface" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Actualmente hay dos fallos del programa que impiden un correcto espaciado horizontal al usar `\magnifyMusic`. Solamente existe una forma de esquivarlo, y su funcionamiento correcto no está garantizado en todos los casos. En el ejemplo que aparece a continuación, sustituya la variable *mag* con su valor preferido. También puede tratar de quitar una o ambas instrucciones `\newSpacingSection`, y/o las instrucciones `\override` y `\revert`:

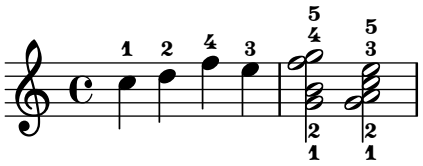
```
\magnifyMusic mag {
  \newSpacingSection
  \override Score.SpacingSpanner.spacing-increment = #(* 1.2 mag)
  [music]
  \newSpacingSection
  \revert Score.SpacingSpanner.spacing-increment
}
```

7.1.2 Indicaciones de digitación

LilyPond provee dos grabadores para el manejo de las indicaciones de digitaición. En ambos casos, las instrucciones de digitación se pueden introducir usando '*nota-dígito*':

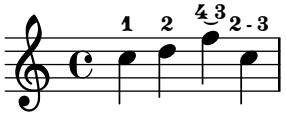
- El primer grabador es `Fingering_engraver`, que se ocupa de las indicaciones de digitación que están fuera de las construcciones de acorde (es decir, fuera de `<...>`). El orden en que las digitaciones aparecen en el código de entrada se refleja directamente en la salida, y el marcado de digitaciones siempre se apila de forma vertical dentro de una columna única encima o debajo de la nota o acorde.

```
\relative {
  c''4-1 d-2 f-4 e-3 |
  <g, b f' g>2_2_1^4^5 <g a c e>_2_1^3^5
}
```



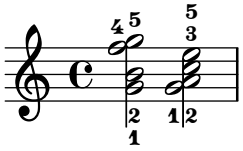
Si quiere textos de marcado o cadenas de caracteres para la digitación, use en su lugar la instrucción `\finger`.

```
\relative {
  c''4-1 d-2 f\finger \markup \tied-lyric "4~3" c\finger "2 - 3"
}
```



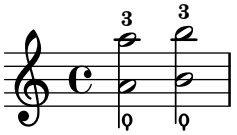
- El segundo grabador es `New_fingering_engraver`, que maneja instrucciones de digitación, articulaciones y cabezas de nota de armónicos dentro de los acordes (es decir, dentro de `<...>`).

```
\relative {
  <g'-1 b-2 f'-4 g-5>2 <e'-5 c-3 a-2 g-1>
}
```



Puede usar la digitación de pulgar para indicar que una nota se debe tocar con el pulgar (p.ej. en música de violoncello).

```
\relative { <a'_\thumb a'-3>2 <b_\thumb b'-3> }
```



Las indicaciones de digitación se pueden situar manualmente encima o debajo del pentagrama, véase Sección 36.1 [Dirección y posición], página 783.

Consulte la sección siguiente para ver ejemplos de fragmentos de código que muestran cómo controlar la posición de las indicaciones de digitación.

Fragmentos de código seleccionados

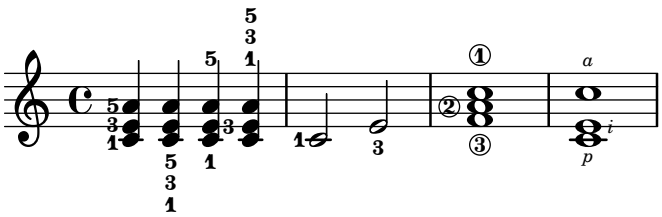
Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece up en la lista), por debajo (si aparece

down), a la izquierda (si aparece left), o a la derecha (si aparece right). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que left y right se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

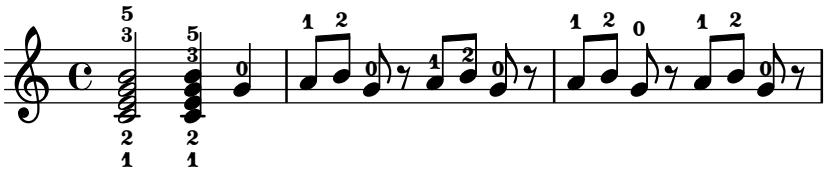
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeteminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Véase también

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783.

Fragmentos de código: Sección “Anotaciones editoriales” en *Fragmentos de código*.

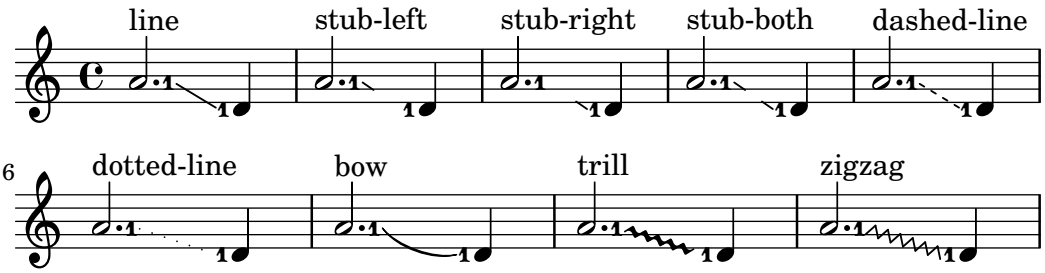
Referencia de funcionamiento interno: Sección “FingeringEvent” en *Referencia de Funcionamiento Interno*, Sección “fingering-event” en *Referencia de Funcionamiento Interno*, Sección “Fingering_engraver” en *Referencia de Funcionamiento Interno*, Sección “New_fingering_engraver” en *Referencia de Funcionamiento Interno*, Sección “Fingering” en *Referencia de Funcionamiento Interno*.

7.1.3 Deslizamientos de dedo

En los instrumentos de cuerda se suele indicar el deslizamiento de un dedo mediante una línea que conecta el mismo dedo usado para notas que están en distintos lugares de la cuerda. Esta línea se inicia con `\glide` escrito antes de la digitación con `Fingering` y termina con la siguiente ocurrencia del mismo dedo. La línea se puede imprimir en varios estilos diferentes.

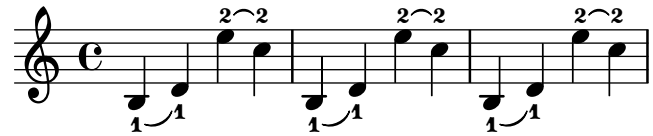
```
mus = {
  \set fingeringOrientations = #'(right)
  <a'\glide-1>2.
  \set fingeringOrientations = #'(left)
  <d'-1>4
}

{
  <>^"line"
  \mus
  <>^"stub-left"
  \override FingerGlideSpanner.style = #'stub-left
  \mus
  <>^"stub-right"
  \override FingerGlideSpanner.style = #'stub-right
  \mus
  <>^"stub-both"
  \override FingerGlideSpanner.style = #'stub-both
  \mus
  <>^"dashed-line"
  \override FingerGlideSpanner.style = #'dashed-line
  \mus
  \break
  <>^"dotted-line"
  \override FingerGlideSpanner.style = #'dotted-line
  \mus
  <>^"bow"
  \override FingerGlideSpanner.style = #'bow
  \mus
  <>^"trill"
  \override FingerGlideSpanner.style = #'trill
```

```
  \mus
  <>^"zigzag"
  \override FingerGlideSpanner.style = #'zigzag
  \mus
}
```



Si se establece `style` a `'bow` la dirección del arco puede ajustarse usando modificadores de dirección.

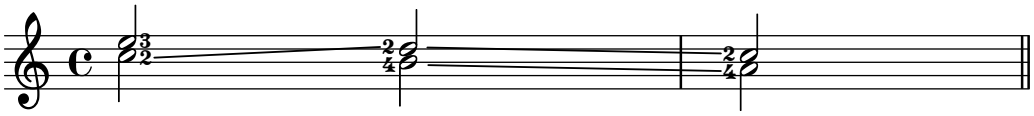
```
{
  \override FingerGlideSpanner.style = #'bow
  \set fingeringOrientations = #'(down)
  <b\glide-1>4 <d'-1>
  \set fingeringOrientations = #'(up)
  <e''\glide-2> <c''-2>

  \set fingeringOrientations = #'(down)
  <b^\glide-1>4 <d'-1>
  \set fingeringOrientations = #'(up)
  <e''^\glide-2> <c''-2>

  \set fingeringOrientations = #'(down)
  <b_\glide-1>4 <d'-1>
  \set fingeringOrientations = #'(up)
  <e''_\glide-2> <c''-2>
}
```



Si se traslada el grabador `Finger_glide_engraver` al contexto de `Staff`, se puedne conectar entre sí los objetos gráficos del tipo `Fingering` de distintos contextos `Voice`.

```
\score {
  \new Staff <<
    \new Voice {
      \voiceOne
      \set fingeringOrientations = #'(right)
      <e''-3>2
      \set fingeringOrientations = #'(left)
      <d''-\tweak bound-details.left.padding 2.5 \glide-2>
      <c''-2>
      \bar "||"
    }
```

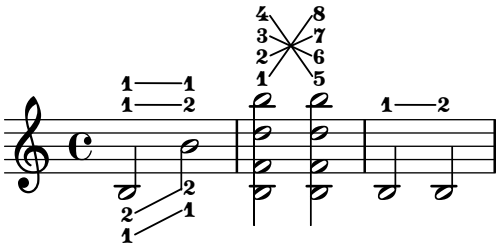
```
    \new Voice {
      \voiceTwo
      \set fingeringOrientations = #'(right)
      <c''\glide-2>
      \set fingeringOrientations = #'(left)
      <b'-\tweak bound-details.left.padding 2.5 \glide-4>
      <a'-4>
    }
  >>
  \layout {
    ragged-right = ##f
    \context {
      \Voice
      \remove Finger_glide_engraver
    }
    \context {
      \Staff
      \consists Finger_glide_engraver
    }
  }
}
```



Para conectar distintos dedos o apariciones diversas del mismo dedo, establezca la propiedad `id` con `\=` y escribiendo un número no negativo o un símbolo, o haga un truco con `\tweak` sobre la propiedad `text`.

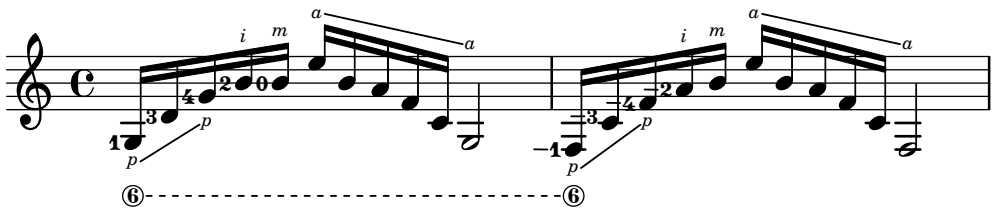
```
{
  b2 \glide \= #'foo ^1
     \glide \= #'bar ^1
     \glide _2
     \glide _1
  b' \= #'foo ^2
     \= #'bar ^1
     _2
     _1

  \set fingeringOrientations = #'(up)
  <
   b\glide \=1 -1
   f'\glide \=2 -2
   d''\glide \=3 -3
   b''\glide \=4 -4
  >
  <
   b\=4 -5
   f'\=3 -6
   d''\=2 -7
   b''\=1 -8
  >
```

```
  b\glide -1 b\tweak text "2" -1
}
```



El objeto de extensión FingerGlideSpanner también puede conectar números de cuerda indicando que se toque sobre la misma cuerda, o pulsar los dedos indicando un *arrastre*.

```
{
  \override StringNumber.staff-padding = 7
  \omit TupletNumber
  \set fingeringOrientations = #'(left)
  \tuplet 5/4 4 {
    \set strokeFingerOrientations = #'(down)
    <
     g-\tweak style #'stub-right \glide-1
       \glide \rightHandFinger #1
       -\tweak style #'dashed-line \glide _\6
    >16
    <d'-\tweak style #'stub-right \glide -3 >
    <g'-\tweak style #'stub-right \glide -4 \rightHandFinger #1 >
    \set strokeFingerOrientations = #'(up)
    <b'-\tweak style #'stub-right \glide -2 \rightHandFinger #2 >
    <b'-0\rightHandFinger #3 >
    e''\glide \rightHandFinger #4
    b' a' f' c'
  }
  g2\rightHandFinger #4

  \tuplet 5/4 4 {
    \set strokeFingerOrientations = #'(down)
    <f-1 \glide \rightHandFinger #1 _\6 >16
    % Raise a bit, otherwise the stub line would be hidden
    % by the ledger line.
    <c'\tweak Y-offset #0.5 -3>
    <f' -4 \rightHandFinger #1 >
    \set strokeFingerOrientations = #'(up)
    <a'-2\rightHandFinger #2 >
    b'\rightHandFinger #3
    e''\glide \rightHandFinger #4
    b' a' f' c'
  }
  f2\rightHandFinger #4
}
```



Véase también

Glosario musical: Sección “arrastre” en *Glosario Musical*.

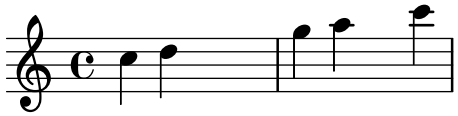
Referencia de la notación: Sección 36.1 [Dirección y posición], página 783.

Referencia de funcionamiento interno: Sección “FingerGlideEvent” en *Referencia de Funcionamiento Interno*, Sección “finger-glide-event” en *Referencia de Funcionamiento Interno*, Sección “Finger_glide_engraver” en *Referencia de Funcionamiento Interno*, Sección “finger-glide-interface” en *Referencia de Funcionamiento Interno*, Sección “FingerGlideSpanner” en *Referencia de Funcionamiento Interno*.

7.1.4 Notas ocultas

Las notas ocultas (o invisibles, o transparentes) pueden ser de utilidad en la preparación de ejercicios de teoría o de composición.

```
\relative {
  c''4 d
  \hideNotes
  e4 f
  \unHideNotes
  g a
  \hideNotes
  b
  \unHideNotes
  c
}
```



La cabeza, la plica y el corchete de las figuras, así como los silencios, son invisibles. Las barras son invisibles si comienzan en una nota oculta. Los objetos de notación que están anclados a notas invisibles, son a pesar de ello visibles.

```
\relative c'' {
  e8(\p f g a)--
  \hideNotes
  e8(\p f g a)--
}
```



Instrucciones predefinidas

`\hideNotes`, `\unHideNotes`.

Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Notation Reference: Sección 2.2.2 [Silencios invisibles], página 70, Sección 36.7 [Visibilidad de los objetos], página 794, Sección 6.2.3 [Ocultar pentagramas], página 260.

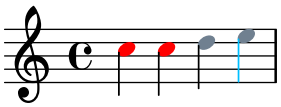
Fragmentos de código: Sección “Anotaciones editoriales” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_spacing_engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteSpacing” en *Referencia de Funcionamiento Interno*.

7.1.5 Colorear objetos

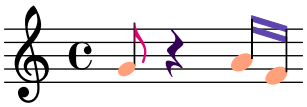
Se pueden asignar colores a los objetos individuales. Los nombres de color válidos se encuentran relacionados en la Sección B.7 [Lista de colores], página 907.

```
\override NoteHead.color = #red
c''4 c''
\override NoteHead.color = #(x11-color 'SlateGrey)
d''
\override Stem.color = "deepskyblue"
e''
```



Además de un conjunto limitado de colores sencillos que están disponibles como variables predefinidas (véase ‘Colores normales’ dentro de Sección B.7 [Lista de colores], página 907), se puede introducir cualquier color como una cadena de caracteres. Dicha cadena puede ser un nombre predefinido de color al estilo de CSS (`https://www.w3.org/Style/CSS/`), o un código de color hexadecimal con el carácter ‘`#`’ como prefijo (*dentro* de las comillas):

```
\override NoteHead.color = "lightsalmon"
\override Flag.color = "#E30074"
\override Beam.color = "#5e45ad"
\override Rest.color = "#3058"
g'8 \huge r4 a'16 f'
```



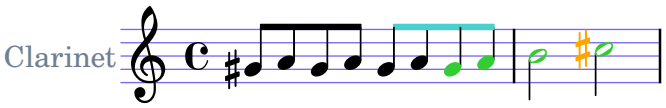
Si dicho color incluye un canal alfa para la semitransparencia mediante el uso de un código de ocho caracteres del tipo "*#RRGGBBAA*" o su forma abreviada "*#RGBA*", se usará en la salida de SVG pero no en la salida de PostScript o PDF. En el ejemplo anterior, el silencio es semitransparente solo si el código se compila con el backend de SVG, como se explica en Sección 23.3 [Formatos de salida alternativos], página 651.

De manera completamente distinta, se puede acceder al rango de colores completo definido para X11 (`https://en.wikipedia.org/wiki/X11_color_names`) mediante el uso de la función de Scheme `x11-color`. Esta función admite un argumento que puede ser un símbolo, como `'DarkSeaGreen4`, o una cadena de caracteres, como `"DarkSeaGreen4"`. La primera manera es más rápida de escribir y ligeramente más eficiente; sin embargo la segunda forma permite además especificar colores de X11 como frases de varias palabras: en el ejemplo, `"dark sea green 4"`.

Si `x11-color` no entiende el parámetro, el color predeterminado que se devuelve es el negro.

```
\new Staff \with {
  instrumentName = \markup {
```

```
    \with-color #(x11-color 'SlateGrey) "Clarinet"
    }
  }
  \relative c'' {
  \override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
  gis8 a
  \override Beam.color = #(x11-color "medium turquoise")
  gis a
  \override Accidental.color = #(x11-color 'orange)
  gis a
  \override NoteHead.color = #(x11-color "LimeGreen")
  gis a
  % this is deliberate nonsense; note that the stems remain black
  \override Stem.color = #(x11-color 'Boggle)
  b2 cis
}
```



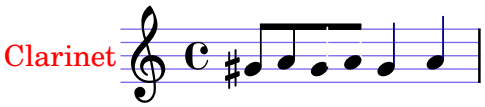
LilyPond contempla también un conjunto de ocho nombres de colores (`https://jfly.uni-koeln.de/color`) no ambiguos para personas con ceguera a los colores o sin ella. Use `universal-color` para acceder a ellos.

```
\markup \with-color #(universal-color 'vermillion) vermillion
```

vermillion

Se pueden especificar colores RGB exactos utilizando la función de Scheme `rgb-color`. Esta función admite tres argumentos que se utilizan respectivamente para los canales *rojo*, *verde* y *azul*, y un número de *alfa* opcional para la semitransparencia (todos los valores deben ser números entre 0 y 1). De nuevo, la transparencia solo se contempla en la salida de SVG; por ejemplo, en el fragmento siguiente la clave es semitransparente cuando se produce un documento SVG.

```
\new Staff \with {
    instrumentName = \markup {
      \with-color #(x11-color 'red) "Clarinet"
      }
    \override Clef.color = #(rgb-color 0 0 0 0.5)
  }
\relative c'' {
  \override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
  \override Stem.color = #(rgb-color 0 0 0)
  gis8 a
  \override Stem.color = #(rgb-color 1 1 1)
  gis8 a
  \override Stem.color = #(rgb-color 0 0 0.5)
  gis4 a
}
```



Véase también

Referencia de la notación: Sección B.7 [Lista de colores], página 907, Sección 35.6 [Las instrucciones `\tweak` y `\single`], página 773.

Fragmentos de código: Sección "Anotaciones editoriales" en *Fragmentos de código*.

Advertencias y problemas conocidos

Un color de X11 no es necesariamente de la misma tonalidad exacta que un color normal de nombre similar.

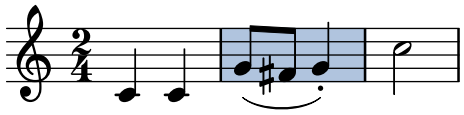
No todos los colores de X11 se distinguen entre sí en un navegador web, es decir, un navegador de web podría no mostrar ninguna diferencia entre `LimeGreen` (verde lima) y `ForestGreen` (verde bosque). Para la web se recomiendan los colores de CSS, como aparece detallado en Sección B.7 [Lista de colores], página 907.

Las notas de un acorde no se pueden colorear individualmente con `\override`; en su lugar utilice `\tweak` o su equivalente `\single\override` antes de la nota respectiva, véase Sección 35.6 [Las instrucciones `\tweak` y `\single`], página 773.

7.1.6 Resaltados en el pentagrama

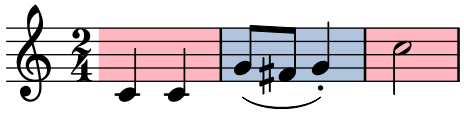
Para un propósito de análisis o pedagógico, puede ser de utilidad "resaltar" pasajes musicales, por ejemplo para mostrar una región tonal. Esto se puede hacer usando la instrucción `\staffHighlight`, que recibe un color como argumento. Para conocer todas las formas de introducir colores, véase Sección 7.1.5 [Colorear objetos], página 292. El resaltado se finaliza usando `\stopStaffHighlight`.

```
\relative {
  \time 2/4
  c'4 4
  \staffHighlight "lightsteelblue"
  g'8( fis g4)-.
  \stopStaffHighlight
  c2
}
```



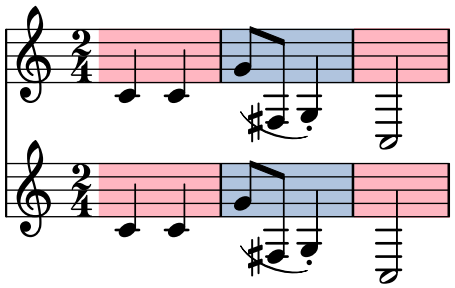
Si hay varios resaltados seguidos, no es necesario escibir `\stopStaffHighlight`, porque `\staffHighlight` también termina implícitamente el resaltado en curso, si hay uno. De forma similar, no es necesario escibir `\stopStaffHighlight` al final de la pieza. Esto es especialmente práctico si se quieren resaltar todos y cada uno de los compases.

```
\relative {
  \time 2/4
  \staffHighlight "lightpink"
  c'4 4
  \staffHighlight "lightsteelblue"
  g'8( fis g4)-.
  \staffHighlight "lightpink"
  c2
}
```



De forma predeterminada, los pentagramas se resaltan por separado.

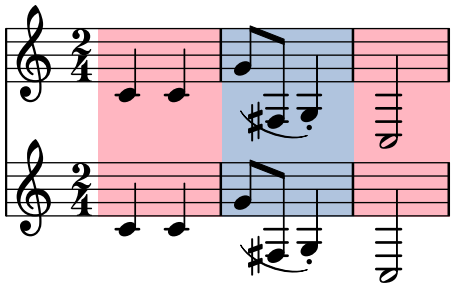
```
music = {
  \time 2/4
  \staffHighlight "lightpink"
  c'4 4
  \staffHighlight "lightsteelblue"
  g'8( fis g4)-.
  \staffHighlight "lightpink"
  c2
}

<<
  \new Staff \music
  \new Staff \music
>>
```



Sin embargo, se pueden resaltar varios pentagramas al mismo tiempo trasladando el grabador `Staff_highlight_engraver` a un contexto más elevado que `Staff` (o `RhythmicStaff`, o similar). Esto se hace usando las instrucciones `\consists` y `\remove`; Véase Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752, para más información. Por ejemplo, si el grabador se traslada a `Score`, los resaltados se comparten por parte de todos los pentagramas.

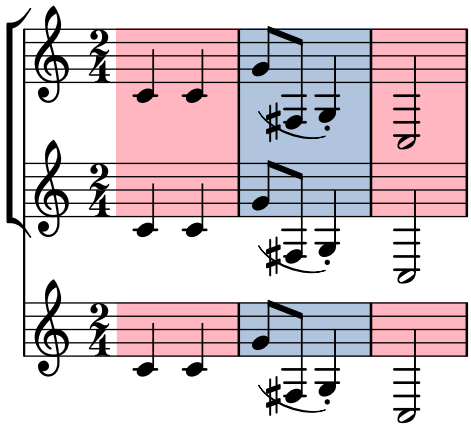
```
\layout {
  \context {
    \Staff
    \remove Staff_highlight_engraver
  }
  \context {
    \Score
    \consists Staff_highlight_engraver
  }
}

music = {
  \time 2/4
  \staffHighlight "lightpink"
  c'4 4
  \staffHighlight "lightsteelblue"
  g'8( fis g4)-.
  \staffHighlight "lightpink"
```

```
  c2
}

<<
  \new Staff \music
  \new Staff \music
>>
```



También se puede trasladar el grabador `Staff_highlight_engraver` a contextos de nivel intermedio como `StaffGroup`.

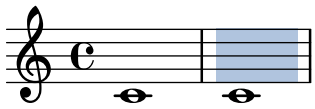
```
music = {
  \time 2/4
  \staffHighlight "lightpink"
  c'4 4
  \staffHighlight "lightsteelblue"
  g'8( fis g4)-.
  \staffHighlight "lightpink"
  c2
}

<<
  \new StaffGroup \with { \consists Staff_highlight_engraver } <<
    \new Staff \with { \remove Staff_highlight_engraver } \music
    \new Staff \with { \remove Staff_highlight_engraver } \music
  >>
  \new Staff \music
>>
```



Se puede usar la propiedad `StaffHighlight.shorten-pair` para trucar el comienso y final horizontales del objeto de extensión del resaldado.

```
{
```

```
  c'1
  \once \override Staff.StaffHighlight.shorten-pair = #'(1.0 . 1.0)
  \staffHighlight lightsteelblue
  c'1
}
```



Instrucciones predefinidas

`\staffHighlight`, `\stopStaffHighlight`.

Véase también

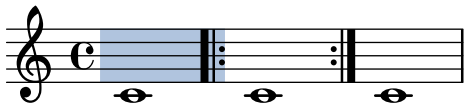
Referencia de la notación: Sección 7.1.5 [Colorear objetos], página 292, Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752.

Referencia de funcionamiento interno: Sección “StaffHighlight” en *Referencia de Funcionamiento Interno*, Sección “staff-highlight-interface” en *Referencia de Funcionamiento Interno*, Sección “Staff_highlight_engraver” en *Referencia de Funcionamiento Interno*, Sección “StaffHighlightEvent” en *Referencia de Funcionamiento Interno*, Sección “staff-highlight-event” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El comportamiento de los resaltados puede no ser el esperado en las barras de inicio de una repetición. Se puede usar la propiedad `shorten-pair` mostrada anteriormente para resolver el problema.

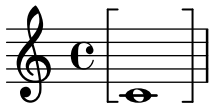
```
{
  \staffHighlight "lightsteelblue"
  c'1
  \stopStaffHighlight
  \repeat volta 2 { c'1 }
  c'1
}
```



7.1.7 Corchetes para material opcional

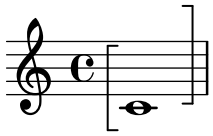
El material adicional opcional se puede encerrar entre corchetes que atraviesan todo el pentagrama.

```
{
  \startOptionalMaterial
  c'1
  \stopOptionalMaterial
}
```



Se pueden sobreescribir las posiciones de las notas que el corchete encierra. Los puntos extremos del intervalo se miden en espacios de pentagrama a partir de la línea central hasta el centro de la nota.

```
{
  \tweak OptionalMaterialBracket.positions #'(-4 . 1)
  \startOptionalMaterial
  c'1
  \once \override Staff.OptionalMaterialBracket.positions =
  #'(-2 . 4)
  \stopOptionalMaterial
}
```



Instrucciones predefinidas

\startOptionalMaterial, \stopOptionalMaterial.

Véase también

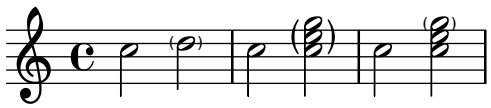
Fragmentos de código: Sección “Anotaciones editoriales” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Optional_material_bracket_engraver” en *Referencia de Funcionamiento Interno*, Sección “OptionalMaterialBracket” en *Referencia de Funcionamiento Interno*, Sección “optional-material-bracket-interface” en *Referencia de Funcionamiento Interno*.

7.1.8 Paréntesis

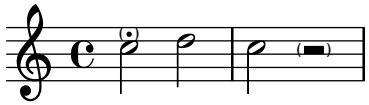
Los objetos se pueden encerrar entre paréntesis anteponiendo \parenthesize al evento musical.

```
\relative {
  c''2 \parenthesize d
  c2 \tweak Parentheses.font-size 2 \parenthesize <c e g>
  c2 <c e \parenthesize g>
}
```



Los objetos que no son notas también se pueden poner entre paréntesis. Para las articulaciones se necesita un signo menos antes de la instrucción \parenthesize.

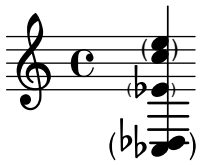
```
\relative {
  c''2-\parenthesize -. d
  c2 \parenthesize r
}
```



Para encerrar entre paréntesis un grupo de notas dentro de un acorde, use la construcción de música en paralelo << ... >>.

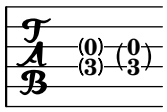
```
\new Voice \relative c {
  <<
    { \tweak Parentheses.font-size 0 \parenthesize <ces des> }
```

```
    { \parenthesize ees' }
    { \tweak Parentheses.font-size -2 \parenthesize <c' e> }
  >>
}
```



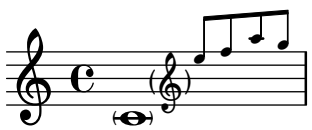
En una tablatura especificamos NoteColumn para encerrar el acorde entre paréntesis.

```
\new TabVoice {
  \override Parentheses.font-size = 0
  \parenthesize <f g>
  \parenthesize NoteColumn <f g>
}
```



Esta segunda forma de la instrucción \parenthesize implica el uso de una ruta de objetos gráficos: o bien \parenthesize *ContextName.GrobName* o simplemente \parenthesize *GrobName* (este último implica al contexto de nivel inferior, normalmente Voice). Se debe añadir antes del momento musical, como una sobreescritura \once \override. Esta forma hace posible encerrar entre paréntesis elementos gráficos que solo se originan indirectamente a partir de eventos.

```
\new Staff \relative <<
  {
    \parenthesize NoteHead
    c'1
  }
  \new CueVoice {
    s2
    \voiceOne
    \once \override Staff.Parentheses.font-size = 3
    \parenthesize Staff.CueClef
    \cueClef treble
    e'8 f a g
  }
>>
```



Véase también

Fragmentos de código: Sección “Anotaciones editoriales” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Parenthesis_engraver” en *Referencia de Funcionamiento Interno*, Sección “Parentheses” en *Referencia de Funcionamiento Interno*, Sección “parentheses-interface” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Al poner un acorde entre paréntesis, se encierra cada una de las notas individuales entre paréntesis, en vez de un solo paréntesis grande rodeando al acorde completo.

7.1.9 Plicas

Cuando se encuentra con una nota, se crea automáticamente un objeto `Stem` (plica). Para las redondas y los silencios, también se crean pero se hacen invisibles.

Se puede hacer manualmente que las plicas apunten hacia arriba o hacia abajo; véase Sección 36.1 [Dirección y posición], página 783.

Instrucciones predefinidas

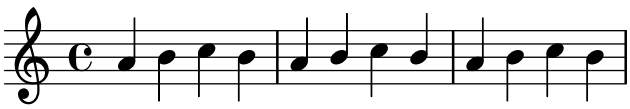
`\stemUp`, `\stemDown`, `\stemNeutral`.

Fragmentos de código seleccionados

Dirección predeterminada de las plicas sobre la tercera línea del pentagrama

La dirección predeterminada de las plicas sobre la tercera línea del pentagrama está determinada por la propiedad `neutral-direction` del objeto `Stem`.

```
\relative c'' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}
```



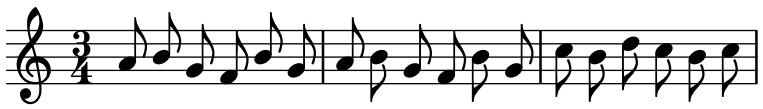
Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía

LilyPond puede alterar la dirección de la plica de las notas que van en la tercera línea de un pentagrama de forma que siga la melodía, mediante la adición del grabador `Melody_engraver` al contexto Voice.

Se puede usar la propiedad de contexto `suspendMelodyDecisions` para desactivar este comportamiento localmente.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a  b g f b g |
  \unset suspendMelodyDecisions
  c  b d c b c |
}

\layout {
  \context {
    \Voice
```

```
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```



Véase también

Referencia de la notación: Sección 36.1 [Dirección y posición], página 783.

Fragmentos de código: Sección "Anotaciones editoriales" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Stem_engraver" en *Referencia de Funcionamiento Interno*, Sección "Stem" en *Referencia de Funcionamiento Interno*, Sección "stem-interface" en *Referencia de Funcionamiento Interno*.

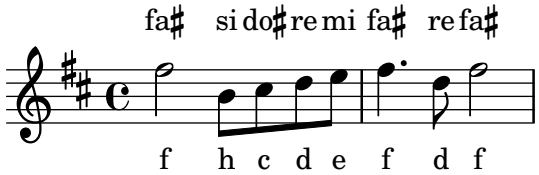
7.2 Fuera del pentagrama

Esta sección trata sobre cómo enfatizar elementos que están dentro del pentagrama, desde fuera del pentagrama.

7.2.1 Nombre de las notas

Se pueden imprimir los nombres de las notas como texto, usando el contexto `NoteNames`. Si se usa simultáneamente con una pauta ordinaria, es posible sincronizar cada nota con su nombre impreso por encima o por debajo de la pauta.

```
\language "italiano"
melody = \relative do'' {
  fad2 si,8 dod re mi fad4. re8 fad2
}

<<
  \new NoteNames { \melody }
  \new Staff { \key si \minor \melody }
  \new NoteNames {
    \set printNotesLanguage = "deutsch"
    \set printAccidentalNames = ##f
    \melody
  }
>>
```

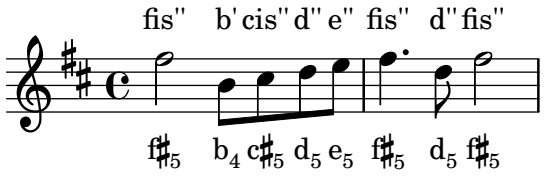


De manera predeterminada, los nombres de nota se imprimen en el mismo idioma que se ha usado para la entrada de la música; sin embargo, la propiedad `printNotesLanguage` permite seleccionar cualquier otro idioma disponible (véase Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10). A través de la propiedad `printAccidentalNames` se determina si las alteraciones se imprimen o no.

Estableciendo tanto la mencionada propiedad a un símbolo, como la propiedad `printOctaveNames` al valor `#t` (verdadero), se pueden obtener nombres de notas de una manera

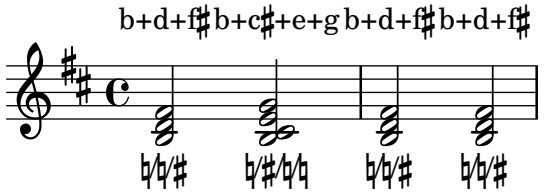
muy similar a la sintaxis de entrada de LilyPond. Para un resultado más general, es posible conseguir nombres de nota 'científicos'.

```
melody = \relative c'' {
  fis2 b,8 cis d e fis4. d8 fis2
}

<<
  \new NoteNames {
    \set printOctaveNames = ##t
    \set printAccidentalNames = #'lily
    \melody
  }
  \new Staff { \key b \minor \melody }
  \new NoteNames {
    \set printOctaveNames = #'scientific
    \melody
  }
>>
```



La propiedad `noteNameSeparator` define cómo se imprimen los acordes. Se pueden definir otras funciones de formateado como `noteNameFunction`; tales funciones deben admitir los argumentos `altura` y `contexto`, incluso si alguno de ellos puede ignorarse.

```
somechords = \relative c' {
  <b d fis>2 <b cis e g> <b d fis> q
}

<<
  \new NoteNames {
    \set noteNameSeparator = "+"
    \somechords
  }
  \new Staff { \key b \minor \somechords }
  \new NoteNames {
    \set noteNameFunction =
      #(lambda (pitch ctx)
         (accidental->text-markup
          (ly:pitch-alteration pitch)))
    \somechords
  }
>>
```



Véase también

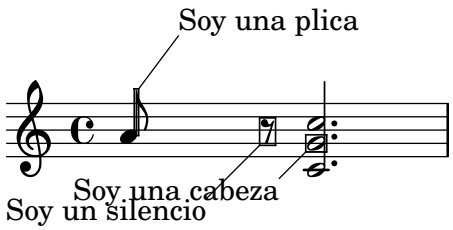
Referencia de la notación: Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10.

Referencia de funcionamiento interno: Sección “NoteName” en *Referencia de Funcionamiento Interno*, Sección “NoteNames” en *Referencia de Funcionamiento Interno*, Sección “Note_name_engraver” en *Referencia de Funcionamiento Interno*.

7.2.2 Globos de ayuda

Los elementos de notación se pueden marcar y nombrar con la ayuda de un cartel o globo de ayuda rectangular. El propósito principal de esta funcionalidad es la explicación de la notación.

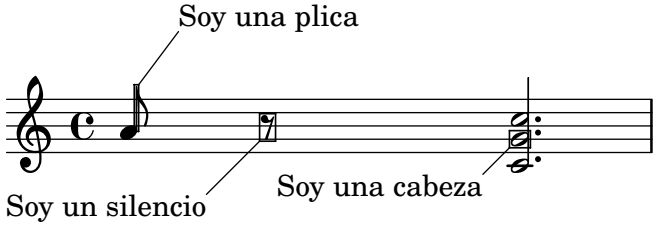
```
\new Voice \with { \consists Balloon_engraver }
\relative c'' {
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}
```



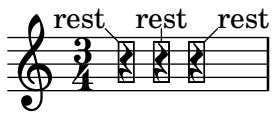
Existen dos funciones musicales, `balloonText` y `balloonGrobText`; el primero se usa como `\tweak`, pero solo dentro de acordes, para adjuntar un texto a una nota individual. del acorde, y el último se usa como `\once \override`, para adjuntar un texto a cualquier grob y en cualquier lugar.

Los textos de globo no influyen en el espaciado de las notas, pero esto puede cambiarse:

```
\new Voice \with { \consists Balloon_engraver }
\relative c'' {
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}
```

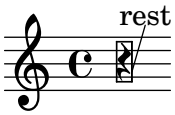


El comportamiento predeterminado del punto de inserción de la línea en el marco y el alineamiento del texto del globo, se muestran más abajo.



El cálculo por omisión del punto de inserción se puede imponer usando las propiedades `X-attachment` y `Y-attachment`, que toman valores entre -1 y 1, con límites que corresponden al extremo izquierdo (respectivamente la parte inferior) y el derecho (respectivamente la parte superior) del marco. La alineación del texto se controla mediante `text-alignment-X` y `text-alignment-Y`, que tienen una forma similar.

```
\new Voice \with {
  \consists Balloon_engraver
}
{
  \once \override BalloonText.Y-attachment = -0.5
  \once \override BalloonText.text-alignment-X = 0.0
  \balloonGrobText Rest #'(1 . 3.5) "rest"
  r4
}
```



Instrucciones predefinidas

`\balloonLengthOn`, `\balloonLengthOff`.

Véase también

Fragmentos de código: Sección “Anotaciones editoriales” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Balloon_engraver” en *Referencia de Funcionamiento Interno*, Sección “BalloonText” en *Referencia de Funcionamiento Interno*, Sección “balloon-interface” en *Referencia de Funcionamiento Interno*.

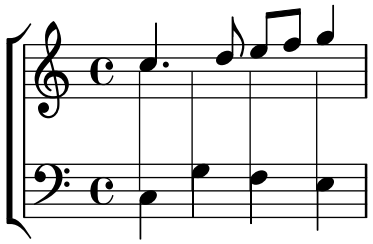
7.2.3 Líneas de rejilla

Se pueden dibujar líneas verticales entre los pentagramas sincronizadas con las notas.

Se debe usar el grabador `Grid_point_engraver` para crear los puntos extremos de las líneas, mientras que el grabador `Grid_line_span_engraver` se debe utilizar para trazar efectivamente las líneas. De forma predeterminada, esto centra las líneas de rejilla horizontalmente debajo y al lado izquierdo de la cabeza de las notas. Las líneas de rejilla se extienden a partir de línea media de los pentagramas. El intervalo `gridInterval` debe especificar la duración entre las líneas de rejilla.

```
\layout {
  \context {
    \Staff
    \consists Grid_point_engraver
    gridInterval = #1/4
  }
  \context {
    \Score
    \consists Grid_line_span_engraver
  }
}
```

```
\score {
  \new ChoirStaff <<
    \new Staff \relative {
      \stemUp
      c''4. d8 e8 f g4
    }
    \new Staff \relative {
      \clef bass
      \stemDown
      c4 g' f e
    }
  >>
}
```



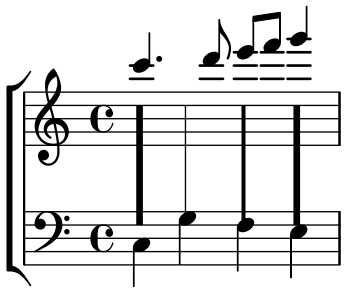
Fragmentos de código seleccionados

Líneas de rejilla: modificar su aspecto

Se puede cambiar el aspecto de las líneas de rejilla sobreescribiendo algunas de sus propiedades.

```
\new ChoirStaff <<
  \new Staff {
    \relative c'' {
      \stemUp
      c'4. d8 e8 f g4
    }
  }
  \new Staff {
    \relative c {
      % this moves them up one staff space from the default position
      \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
      \stemDown
      \clef bass
      \once \override Score.GridLine.thickness = 5.0
      c4
      \once \override Score.GridLine.thickness = 1.0
      g'4
      \once \override Score.GridLine.thickness = 3.0
      f4
      \once \override Score.GridLine.thickness = 5.0
      e4
    }
  }
>>
```

```
\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #1/4
    }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = -0.5
  }
}
```



Véase también

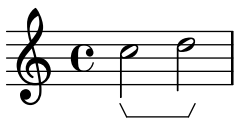
Fragmentos de código: Sección “Anotaciones editoriales” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Grid_line_span_engraver” en *Referencia de Funcionamiento Interno*, Sección “Grid_point_engraver” en *Referencia de Funcionamiento Interno*, Sección “GridLine” en *Referencia de Funcionamiento Interno*, Sección “GridPoint” en *Referencia de Funcionamiento Interno*, Sección “grid-line-interface” en *Referencia de Funcionamiento Interno*, Sección “grid-point-interface” en *Referencia de Funcionamiento Interno*.

7.2.4 Corchetes de análisis

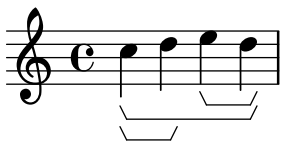
Los corchetes se usan en análisis musical para indicar la estructura de las piezas musicales. Están contemplados los corchetes horizontales simples.

```
\layout {
  \context {
    \Voice
    \consists Horizontal_bracket_engraver
  }
}
\relative {
  c''2\startGroup
  d\stopGroup
}
```



Los corchetes de análisis se pueden anidar.

```
\layout {
  \context {
    \Voice
    \consists Horizontal_bracket_engraver
  }
}
\relative {
  c''4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}
```



Fragmentos de código seleccionados

Corchetes de análisis encima del pentagrama

De forma predeterminada se añaden corchetes de análisis sencillos debajo del pentagrama. El ejemplo siguiente muestra una manera de colocarlos por encima.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



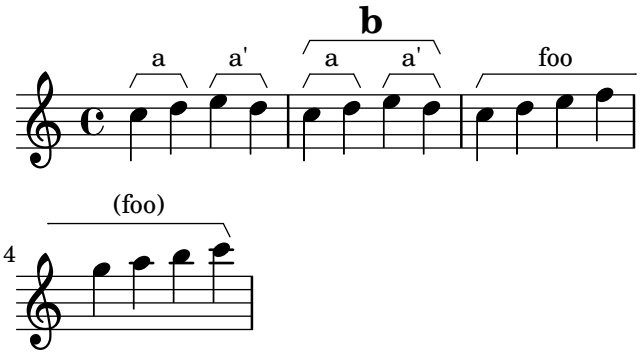
Corchetes de análisis con etiquetas

Se pueden añadir indicaciones de marcado de texto a los corchetes de análisis por medio de la propiedad `text` del objeto gráfico `HorizontalBracketText`. Si se quieren añadir textos diferentes al corchetes que comienzan en el mismo tiempo, es necesario usar la instrucción `\tweak`.

El texto del corchete se encierra entre paréntesis después de un salto de línea. El orden vertical de los corchetes anidados se puede controlar con la propiedad `outside-staff-priority`.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
```

```
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
    c''\startGroup d''\stopGroup
    \once\override HorizontalBracketText.text = "a'"
    e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
      \tweak HorizontalBracketText.text
        \markup \bold \huge "b" \startGroup
     -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```

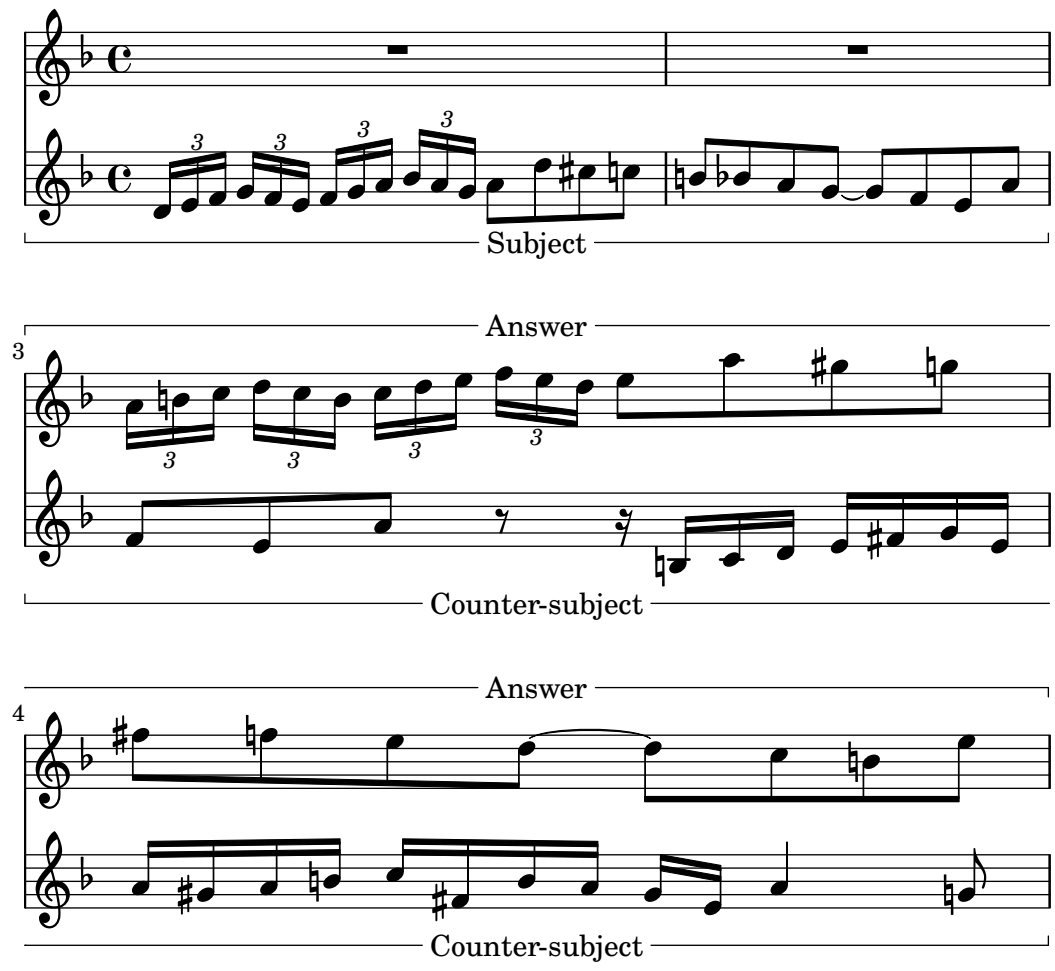


Corchete que abarca varios compases

Los elementos de extensión de compases son una forma alternativa de imprimir corchetes anotados. A diferencia de los corchetes horizontales, estos se extienden entre dos líneas divisorias en lugar de hacerlo entre dos notas. El texto se imprime en el centro del corchete.

```
\layout {
  \context {
    \Staff
    \consists Measure_spanner_engraver
  }
}

<<
  \new Staff \relative c'' {
    \key d \minor
    R1*2
    \tweak text "Answer"
    \startMeasureSpanner
    \tuplet 3/2 8 {
      a16[ b c] d[ c b]  c[ d e] f[ e d]
    }
```

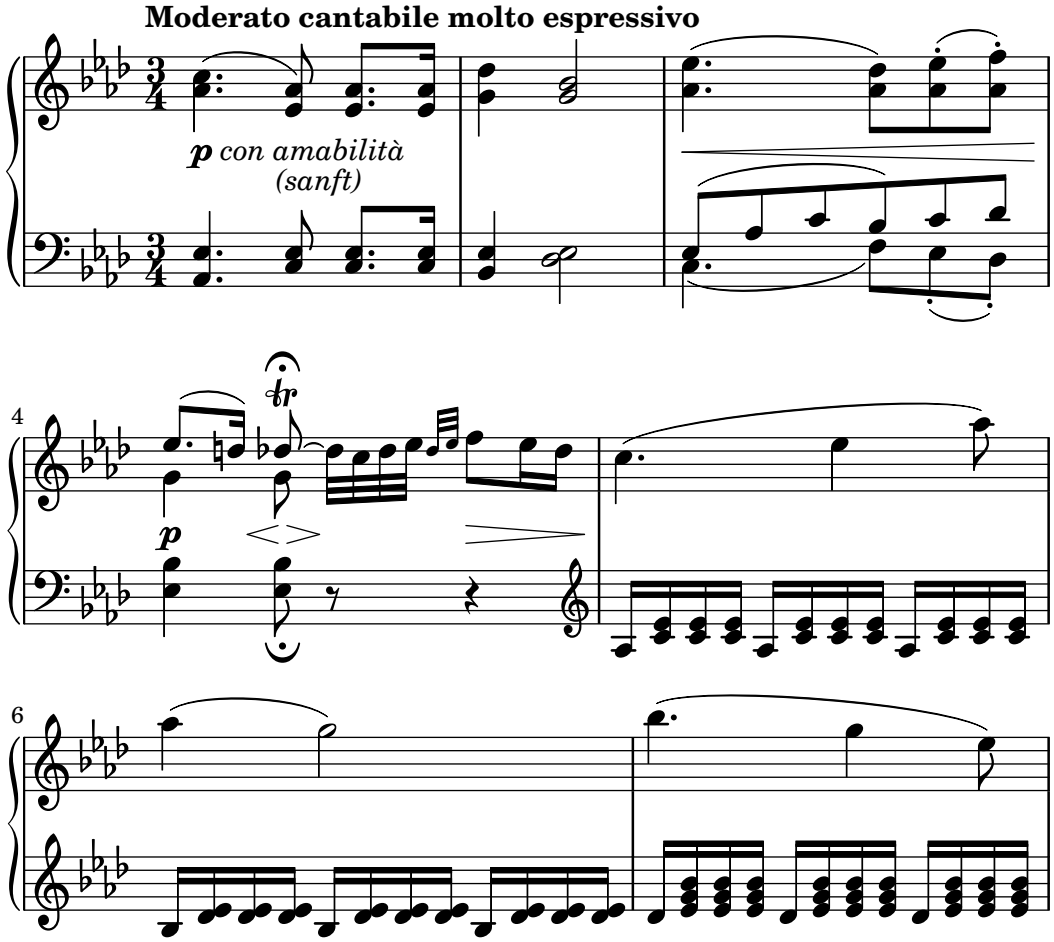
```
    e8 a gis g
    fis f e d~ d c b e
    \stopMeasureSpanner
  }
  \new Staff \relative c' {
    \key d \minor
    \tweak text "Subject"
      \tweak direction #DOWN
      \startMeasureSpanner
    \tuplet 3/2 8 {
      d16[ e f] g[ f e] f[ g a] bes[ a g]
    }
    a8 d cis c
    b bes a g~ g f e a
    \stopMeasureSpanner
    \tweak text "Counter-subject"
      \tweak direction #DOWN
      \startMeasureSpanner
    f8 e a r r16 b, c d e fis g e
    a gis a b c fis, b a gis e a4 g8
    \stopMeasureSpanner
  }
>>
```



Véase también

Referencia de funcionamiento interno: Sección “Horizontal_bracket_engraver” en *Referencia de Funcionamiento Interno*, Sección “HorizontalBracket” en *Referencia de Funcionamiento Interno*, Sección “horizontal-bracket-interface” en *Referencia de Funcionamiento Interno*, Sección “HorizontalBracketText” en *Referencia de Funcionamiento Interno*, Sección “horizontal-bracket-text-interface” en *Referencia de Funcionamiento Interno*, Sección “Measure_spanner_engraver” en *Referencia de Funcionamiento Interno*, Sección “MeasureSpanner” en *Referencia de Funcionamiento Interno*, Sección “measure-spanner-interface” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*.

8 Texto



Esta sección explica cómo incluir texto (con diversos estilos de formateo) en las partituras.

8.1 Escritura del texto

Esta sección presenta las distintas formas de añadir texto a una partitura.

> **Nota:** Para escribir texto con caracteres acentuados y especiales (como los de idiomas distintos del inglés), sencillamente inserte los caracteres directamente en el archivo de LilyPond. El archivo se debe guardar como UTF-8. Para ver más información, consulte Sección 22.4.1 [Codificación del texto], página 647.

8.1.1 Panorámica de los objetos de texto

Los objetos de texto sencillos se introducen como simples cadenas entrecomilladas (que son opcionales para una sola palabra). El modo de marcado es una herramienta más flexible que puede aceptar una variedad de posibilidades avanzadas de formato y gráficas, como se detalla en la sección Sección 8.2 [Formatear el texto], página 323.

Así, los bloques de marcado se pueden utilizar:

- dentro de cualquier objeto del tipo `TextScript` (aplicado a una nota con `-`, `^` o `_`); véase Sección 8.1.2 [Inscripciones textuales], página 314;

- como 'spanners' u objetos extensos, cuando ciertas indicaciones se prolongan sobre varios pulsos o compases. Véase Sección 8.1.3 [Objetos extensos de texto], página 315;
- dentro de cualquier marca impresa por encima de los pentagramas, tales como obletos `RehearsalMark` (letras de ensayo) o `MetronomeMark` (indicaciones de tempo) introducidos respectivamente con las palabras clave `\mark` y `\tempo`; véase Sección 8.1.5 [Marcas de texto], página 317;
- como bloques de texto autosuficientes, que se introducen en el nivel jerárquico superior fuera de cualquier bloque `\score` (en este caso es obligatoria la instrucción `\markup` o `\markuplist`, y no se puede omitir en favor de una simple cadena de texto entrecomillada); véase Sección 8.1.6 [Texto separado], página 321;
- en cualquier definición dentro del bloque `\header` (como `title`, `subtitle`, `composer`), o en elementos específicos definidos dentro del bloque `\paper` tales como `evenHeaderMarkup` para los números de página. Esto se explica en la sección Capítulo 21 [Títulos y encabezamientos], página 598.

Muchos otros objetos basados en texto se pueden escribir como bloques de marcado, incluso cuando este no sea su uso principal.

- Las digitaciones se pueden sustituir fácilmente con bloques de marcado, si se escriben con la instrucción `\finger`; véase Sección 7.1.2 [Indicaciones de digitación], página 284.
- Las sílabas de la letra de una canción se pueden formatear por medio de la instrucción `\markup`; véase Sección 9.1 [Notación común para música vocal], página 349.
- Los nombres de acordes están, de hecho, definidos como bloques de marcado, y por tanto se pueden redefinir de la misma forma para personalizar los modificadores o las excepciones de acorde; véase Sección 15.2 [Imprimir los acordes], página 515.
- Los matices dinámicos se escriben habitualmente de una forma sencilla; sin embargo, es posible definir Sección 3.1.3 [Indicaciones dinámicas nuevas], página 169, como objetos de marcado. Ciertas dinámicas como *crescendo* se imprimen como objetos extensos y se pueden redefinir a través de propiedades como `crescendoText`; véase Sección 3.1.2 [Matices dinámicos], página 160.
- Otros objetos menos frecuentes también están hechos con bloques de marcado, como las indicaciones del tipo *globos de ayuda* (véase Sección 7.2.2 [Globos de ayuda], página 303).

De hecho es posible usar la instrucción `\markup` para personalizar la apariencia de prácticamente cualquier objeto gráfico (o 'grob'), bien sea sobreescribiendo su propiedad `text`, si la tiene, o bien su propiedad `stencil`. Una parte de la lógica que hace posible todo esto, está explicada en Sección "Arquitectura flexible" en *Ensayo*.

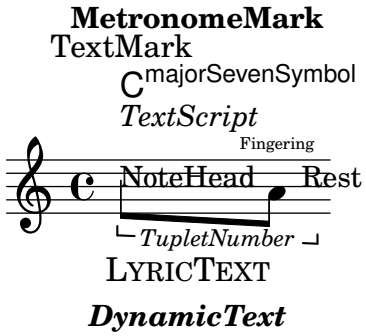
El ejemplo que aparece a continuación ilustra la omnipresencia de los bloques de marcado, no solo como algunos de los objetos relacionados más arriba, sino también en sustitución de objetos musicales por objetos de texto a través de distintos métodos.

```
\header { title = \markup "Header" }

dyn =
#(make-dynamic-script #{ \markup \serif "DynamicText" #})

\markup \box "Top-level markup"

\score {
  <<
    \new ChordNames
    \with {
      majorSevenSymbol = \markup "majorSevenSymbol"
```

```
    }
    \chordmode { c1:maj7 }
    \new Staff {
      \tempo \markup "MetronomeMark"
      \textMark "TextMark"
      \once \override TupletNumber.text =
        \markup "TupletNumber"
      \tuplet 3/2 {
        \once \override NoteHead.stencil =
          #ly:text-interface::print
        \once \override NoteHead.text =
          \markup \lower #0.5 "NoteHead"
        c''8^\markup \italic "TextScript"
        a'\finger \markup \serif "Fingering"
        \once \override Rest.stencil =
          #(lambda (grob)
             (grob-interpret-markup grob #{
               \markup  "Rest"
               #}))
        r
      }
    }
    \new Lyrics \lyricmode {
      \markup \smallCaps "LyricText" 1
    }
    \new Dynamics { s1\dyn }
  >>
}
```

Header

Top-level markup



Véase también

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323, Sección 8.1.2 [Inscripciones textuales], página 314, Sección 8.1.3 [Objetos extensos de texto], página 315, Sección 8.1.5 [Marcas de texto], página 317, Sección 8.1.6 [Texto separado], página 321, Sección 7.1.2 [Indicaciones de digitación], página 284, Sección 9.1 [Notación común para música vocal], página 349, Sección 15.2 [Imprimir los acordes], página 515, Sección 3.1.3 [Indicaciones dinámicas nuevas], página 169, Sección 3.1.2 [Matices dinámicos], página 160, Sección 7.2.2 [Globos de ayuda], página 303.

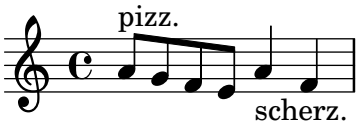
Ensayo sobre grabado musical automatizado: Sección “Arquitectura flexible” en *Ensayo*.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

8.1.2 Inscripciones textuales

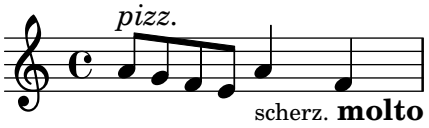
Es posible añadir indicaciones textuales a una partitura escribiéndolas entre comillas como se muestra en el ejemplo siguiente. Estas indicaciones se pueden colocar manualmente por encima o por debajo del pentagrama, utilizando la sintaxis que se describe en Sección 36.1 [Dirección y posición], página 783.

```
\relative { a'8^"pizz." g f e a4-"scherz." f }
```



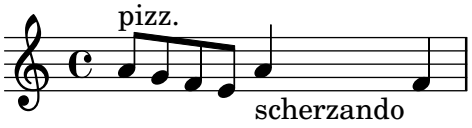
Esta sintaxis es en realidad una abreviatura; se puede añadir explícitamente a una nota un formateado de texto más complejo utilizando un bloque \markup, como se describe bajo Sección 8.2 [Formatear el texto], página 323.

```
\relative {
  a'8^\markup { \italic pizz. } g f e
  a4_\markup { \tiny scherz. \bold molto } f }
```



De forma predeterminada, las indicaciones de texto no afectan al espaciado de las notas. Sin embargo, su anchura sí podría tenerse en cuenta: en el siguiente ejemplo, la primera cadena de texto no afecta al espaciado, pero la segunda sí afecta.

```
\relative {
  a'8^"pizz." g f e
  \textLengthOn
  a4_"scherzando" f
}
```



Se pueden adjuntar articulaciones a las notas, además de inscripciones de texto. Para ver más información, consulte Sección 3.1.1 [Articulaciones y ornamentos], página 156.

Para ver más información sobre el orden relativo de las inscripciones de texto y las articulaciones, consulte Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

Instrucciones predefinidas

\textLengthOn, \textLengthOff.

Véase también

Manual de aprendizaje: Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323, Sección 36.1 [Dirección y posición], página 783, Sección 3.1.1 [Articulaciones y ornamentos], página 156.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

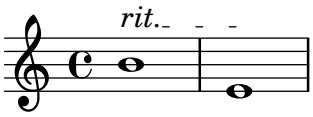
La comprobación necesaria para asegurar que las inscripciones de texto y la letra de las canciones se mantienen dentro de los márgenes, requiere cálculos adicionales. En caso de que desee un proceso ligeramente más rápido, puede utilizar

```
\override Score.PaperColumn.keep-inside-line = ##f
```

8.1.3 Objetos extensos de texto

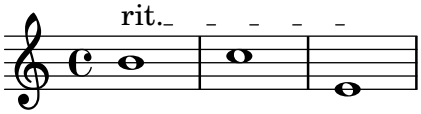
Ciertas indicaciones de ejecución, p.ej., *rallentando* o *accelerando*, se escriben como texto y se extienden sobre muchos compases mediante líneas de puntos. Estos objetos, que reciben el nombre de 'spanners' u objetos de extensión, se pueden crear desde una nota hasta otra usando la siguiente sintaxis:

```
\relative {
  \override TextSpanner.bound-details.left.text = "rit."
  b'1\startTextSpan
  e,\stopTextSpan
}
```



La cadena de texto que imprimir se establece a través de propiedades de objeto. De forma predeterminada se imprime en estilo cursiva, pero se pueden conseguir distintos efectos de formato utilizando bloques `\markup`, como se describe en Sección 8.2 [Formatear el texto], página 323.

```
\relative {
  \override TextSpanner.bound-details.left.text =
    \markup { \upright "rit." }
  b'1\startTextSpan c
  e,\stopTextSpan
}
```



Tanto el estilo de la línea como la cadena de texto, se pueden definir como una propiedad de objeto. Esta sintaxis se describe en Sección 36.5 [Estilos de línea], página 791.

Instrucciones predefinidas

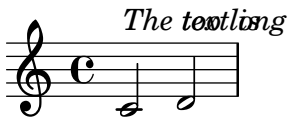
`\textSpannerUp`, `\textSpannerDown`, `\textSpannerNeutral`, `\startTextSpan`, `\stopTextSpan`.

Advertencias y problemas conocidos

LilyPond solo puede manejar un objeto de extensión de texto por cada voz.

Los textos de los extremos de un elemento de extensión de texto pueden presentar colisiones.

```
{
  \once \override TextSpanner.bound-details.left.text = "The text is"
  \once \override TextSpanner.bound-details.right.text = "too long"
  c'2\startTextSpan d'2\stopTextSpan
}
```



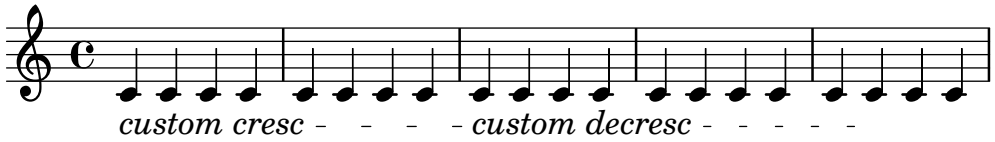
En Capítulo 30 [Espaciado horizontal], página 718, podemos encontrar soluciones alternativas.

Fragmentos de código seleccionados

Dynamics spanner with custom text

Postfix functions for custom crescendo text spanners. The spanners should start on the first note of the measure. One has to use -\mycresc, otherwise the spanner start will rather be assigned to the next note.

```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'CrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'DecrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4 c4\! c4 c4
}
```



Véase también

Referencia de la notación: Sección 36.5 [Estilos de línea], página 791, Sección 3.1.2 [Matices dinámicos], página 160, Sección 8.2 [Formatear el texto], página 323.

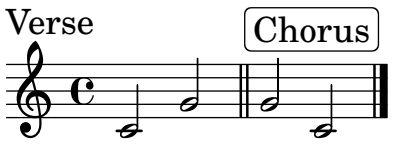
Fragmentos de código: Sección “Texto” en *Fragmentos de código*, Sección “Marcas de expresión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextSpanner” en *Referencia de Funcionamiento Interno*.

8.1.4 Etiquetas de sección

La instrucción \sectionLabel marca el comienzo de un pasaje con nombre. Su uso es adecuado en una división de sección creada con \section, pero no implica a \section y se puede usar de forma aislada.

```
\fixed c' {
  \sectionLabel "Verse"
  c2 g
  \section
  \sectionLabel \markup { \rounded-box { Chorus } }
  g2 c
  \bar "|."
}
```



Véase también

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323, Sección 2.5.5 [Llamadas de ensayo], página 142, Sección 2.5.7 [Divisores de sección], página 147.

Fragmentos de código: Sección "Texto" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "SectionLabel" en *Referencia de Funcionamiento Interno*, Sección "SectionLabelEvent" en *Referencia de Funcionamiento Interno*.

8.1.5 Marcas de texto

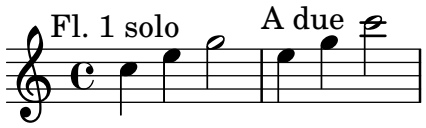
Las marcas o llamadas de texto son objetos textuales que, a diferencia de las inscripciones de texto (véase Sección 8.1.2 [Inscripciones textuales], página 314), no se imprimen sobre las notas sino entre ellas, a menudo alineadas a una barra de compás.

> **Nota:** Versiones antiguas de LilyPond usaban la instrucción `\mark` para las marcas de texto, aunque está indicada principalmente para las llamadas de ensayo (véase Sección 2.5.5 [Llamadas de ensayo], página 142). Las instrucciones `\textMark` y `\textEndMark` están más indicadas para marcas de texto en cuanto a su configuración predeterminada, así como a la posibilidad de tener varias marcas de texto en el mismo punto. Por tanto, se recomienda usar `\textMark` y `\textEndMark` en lugar de `\mark "Texto"` o `\mark \markup ...`. Observe que si estamos convirtiendo texto que usa `\mark` para las marcas de texto, las sobreescrituras que usan `RehearsalMark` se deben cambiar por `TextMark`.

Antes de usar marcas o llamadas de texto, se recomienda buscar una instrucción más específica, si hay alguna disponible. Para los textos que identifican una sección, use `\sectionLabel` (véase Sección 8.1.4 [Etiquetas de sección], página 316). Para las indicaciones de salto, use `\jump` (véase Sección 4.1.8 [Marcas de repetición manual], página 207). Estas instrucciones tienen distintas configuraciones de disposición predeterminadas, y crean objetos separados que se pueden formatear de forma diferente de las marcas de texto genéricas, en las hojas de estilo.

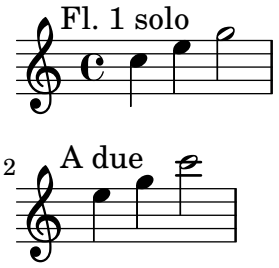
Se introduce una marca de texto usando `\textMark` o `\textEndMark`. La instrucción `\textMark` dibuja una marca alineada a la izquierda.

```
\fixed c'' {
  \textMark "Fl. 1 solo"
  c4 e g2
  \textMark "A due"
  e4 g c'2
}
```



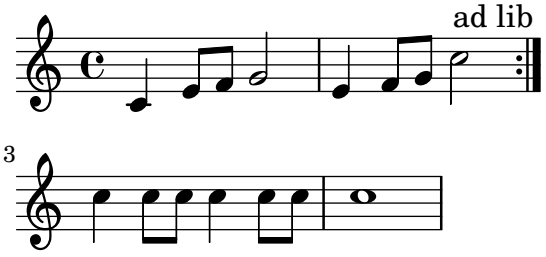
Si se produce un salto de línea en el lugar donde se ha usado la `\textMark`, el texto aparece en el sistema siguiente.

```
\fixed c'' {
  \textMark "Fl. 1 solo"
  c4 e g2
  \break
  \textMark "A due"
  e4 g c'2
}
```



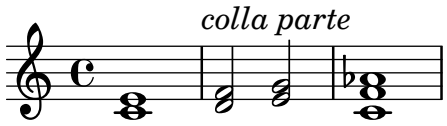
A diferencia de `\textMark`, `\textEndMark` creao una marca alineada por la derecha. Si se produce sobre un salto de línea, se imprime en el sistema anterior.

```
\fixed c' {
  \repeat volta 2 {
    c4 e8 f g2
    e4 f8 g c'2
    \textEndMark "ad lib"
  }
  \break
  c'4 8 8 4 8 8
  c'1
}
```



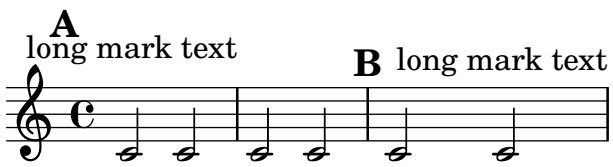
Se puede hacer un formateo complejo de los textos usando un bloque `\markup` (véase Sección 8.2 [Formatear el texto], página 323).

```
\relative {
  <c' e>1
  \textMark \markup { \italic { colla parte } }
  <d f>2 <e g>
  <c f aes>1
}
```



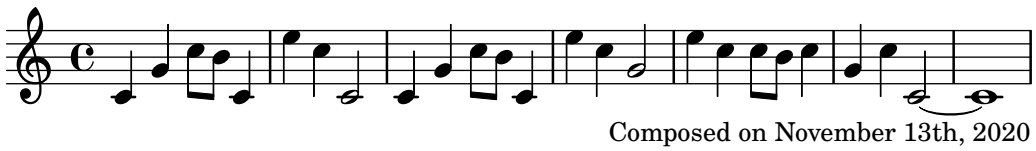
Las instrucciones \markLengthOn y \markLengthOff (véase Sección 2.3.2 [Indicaciones metronómicas], página 86) también se pueden usar en las marcas de texto.

```
{
  \mark \default
  c'2 2
  \textEndMark "long mark text"
  \markLengthOn
  2 2
  \mark \default
  2 2
  \textEndMark "long mark text"
}
```



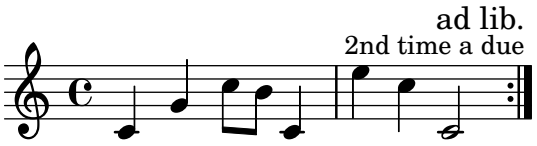
Las marcas de texto se pueden imprimir debajo del pentagrama.

```
\fixed c' {
  c4 g c'8 b c4 e' c' c2
  c4 g c'8 b c4 e' c' g2
  e'4 c' c'8 b c'4
  g4 c' c2~ c1
  \tweak direction #DOWN
    \tweak font-size -1
      \textEndMark "Composed on November 13th, 2020"
}
```



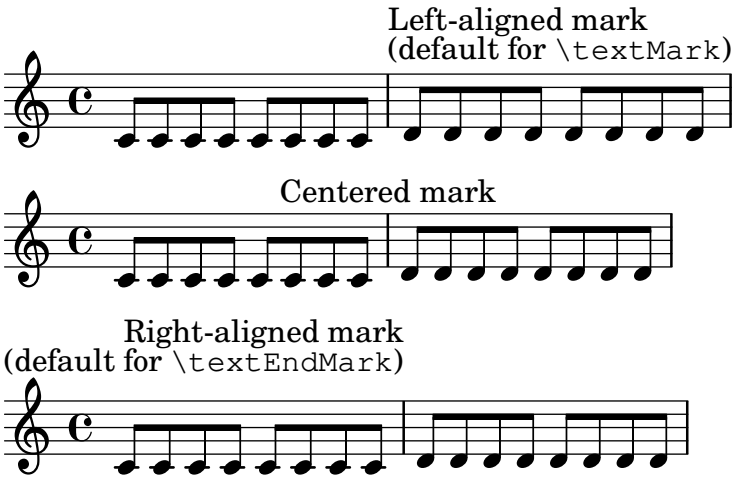
Puede haber varias marcas de texto en el mismo momento. El orden en que se apilan se puede sobreescribir usando la propiedad outside-staff-priority (véase Sección 29.3 [Evitar las colisiones verticales], página 715). Como alternativa, una marca de texto única con \markup \column { ... } puede conseguir el mismo efecto.

```
\fixed c' {
  \repeat volta 2 {
    c4 g c'8 b c4 e' c' c2
    \textEndMark "ad lib."
    \tweak outside-staff-priority 1200
      \tweak font-size -1
        \textEndMark "2nd time a due"
  }
}
```



Se puede alterar la alineación de una marca de texto.

```
{
  c'8 8 8 8 8 8 8 8
  \textMark \markup \column {
    \line { Left-aligned mark }
    \line { (default for \concat { \typewriter "\\textMark" ) } }
  }
  d'8 8 8 8 8 8 8 8
}

{
  c'8 8 8 8 8 8 8 8
  \tweak self-alignment-X #CENTER
    \textMark "Centered mark"
  d'8 8 8 8 8 8 8 8
}

{
  c'8 8 8 8 8 8 8 8
  \tweak self-alignment-X #RIGHT
    \textMark \markup \right-column {
      \line { Right-aligned mark }
      \line { (default for \concat { \typewriter "\\textEndMark" ) } }
    }
  d'8 8 8 8 8 8 8 8
}
```



Instrucciones predefinidas

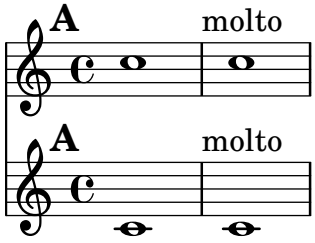
`\textMark`, `\textEndMark`, `\markLengthOn`, `\markLengthOff`.

Fragmentos de código seleccionados

Imprimir marcas de ensayo en cualquier pentagrama

Aunque normalmente las marcas de ensayo y textuales solo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}
```



Véase también

Referencia de la notación: Sección 2.3.2 [Indicaciones metronómicas], página 86, Sección 2.5.5 [Llamadas de ensayo], página 142, Sección 8.1.4 [Etiquetas de sección], página 316, Sección 8.2 [Formatear el texto], página 323, Sección 8.2.5 [Notación musical dentro de elementos de marcado], página 338, Sección B.8 [La tipografía Emmentaler], página 912.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextMarkEvent” en *Referencia de Funcionamiento Interno*, Sección “Text_mark_engraver” en *Referencia de Funcionamiento Interno*, Sección “TextMark” en *Referencia de Funcionamiento Interno*.

8.1.6 Texto separado

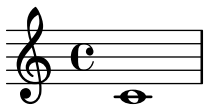
Un bloque `\markup` o `\markuplist` puede existir de forma independiente, fuera de cualquier bloque `\score`, como una “expresión de nivel superior”. Esta sintaxis se describe en Sección 20.5 [Estructura del archivo], página 594.

```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

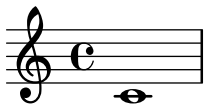
Tomorrow, and tomorrow, and tomorrow...

Esto hace posible imprimir texto separado de la música, lo que es especialmente útil cuando el archivo de entrada contiene varias piezas, tal y como se describe en Sección 20.2 [Varias partituras en un libro], página 591.

```
\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Using \markuplist, los bloques de texto independientes pueden abarcar varias páginas, posibilitando la impresión de documentos de texto o libros íntegramente desde LilyPond. Para ver una descripción de esta funcionalidad y la sintaxis que requiere, véase Sección 8.2.1 [Introducción al marcado de texto], página 323.

Instrucciones predefinidas

\markup, \markuplist.

Fragmentos de código seleccionados

Elemento de marcado de texto independiente en dos columnas

Los textos independientes se pueden disponer en varias columnas utilizando instrucciones \markup

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
    \hspace #2
    \column \italic {
      \line { O sacred feast }
      \line { in which Christ is received, }
```

```
        \line { the memory of His Passion is renewed, }
        \line { the mind is filled with grace, }
        \line { and a pledge of future glory is given to us. }
        \line { Amen. }
      }
      \hspace #1
    }
  }
```

O sacrum convivium	*O sacred feast*
in quo Christus sumitur,	*in which Christ is received,*
recolitur memoria passionis ejus,	*the memory of His Passion is renewed,*
mens impletur gratia,	*the mind is filled with grace,*
futurae gloriae nobis pignus datur.	*and a pledge of future glory is given to us.*
Amen.	*Amen.*

Véase también

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323, Sección 20.5 [Estructura del archivo], página 594, Sección 20.2 [Varias partituras en un libro], página 591.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

8.2 Formatear el texto

Esta sección presenta los formateados básico y avanzado de texto, usando la sintaxis específica del modo de *markup* o de marcado.

8.2.1 Introducción al marcado de texto

Se usa un bloque `\markup` o `\markuplist` para tipografiar texto con una sintaxis ampliable que se denomina “modo de marcado”. Tales bloques se pueden usar en muchos contextos (véase Sección 8.1.1 [Panorámica de los objetos de texto], página 311).

Dentro del modo de marcado, las palabras se imprimen como están. Una palabra única no necesita comillas.

```
\markup intenso
```

intenso

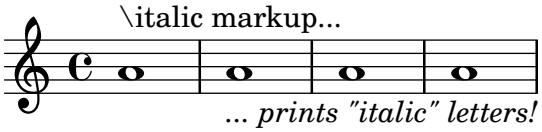
Se pueden agrupar varias palabras juntas encerrándolas entre comillas.

```
\markup "molto intenso"
```

molto intenso

Aparte de agrupar, el entrecomillado también permite escribir caracteres especiales como ‘\’ y ‘#’ sin que afecten al formateado del texto. El propio símbolo de comillas dobles se puede imprimir haciéndolo preceder de una barra inclinada invertida.

```
\relative {
  a'1^"\\italic markup..."
  a_\markup { \italic "... prints \"italic\" letters!" }
  a a
}
```



El formateo se hace por medio de las instrucciones de marcado. Su nombre se escribe precedido de una barra invertida. Estas instrucciones esperan un número variable de argumentos que son específicos de cada una. Para ver una lista exhaustiva de las instrucciones específicas de `\markup`, véase Sección A.1 [Instrucciones de marcado de texto], página 815.

```
\markup \italic "string. assai"
\markup \with-color "red" intenso
```

string. assai

intenso

Se pueden anidar las instrucciones de marcado. El bloque de marcado termina cuando todas las instrucciones han recibido sus correspondientes argumentos.

```
\markup \with-color "red" \italic intenso
```

intenso

Se pueden agrupar varias expresiones de marcado encerrándolas entre llaves, para formar lo que se conoce como una *lista de marcados*. Sin ningún formateo posterior, los elementos de una lista de marcados se tipografían en fila.

```
\markup { molto \italic intenso }
```

molto *intenso*

Ciertas instrucciones no esperan un elemento de marcado, sino una lista de marcados, permitiendo arreglos de texto más complejos que la mera impresión en fila.

```
\markup \center-column {
  \bold "Des Simplicius Simplicissimus Jugend"
  "Karl Amadeus Hartmann"
}
```

Des Simplicius Simplicissimus Jugend
Karl Amadeus Hartmann

Asimismo, algunas instrucciones no devuelven un marcado, sino una lista. El resultado se puede usar luego allí donde se espere una lista de marcados. Para ver una lista de estas instrucciones, véase Sección A.2 [Instrucciones de lista de marcado de texto], página 891.

```
\markup \string-lines
    "Twinkle, twinkle, little star,
     How I wonder what you are!"
```

Twinkle, twinkle, little star, How I wonder what you are!

```
\markup \center-column \string-lines
    "Twinkle, twinkle, little star,
     How I wonder what you are!"
```

Twinkle, twinkle, little star,
How I wonder what you are!

Los elementos de una lista de marcados anidada se tratan simplemente como elementos de la lista de marcados principal.

```
\markup \center-column {
  \bold "Des Simplicius Simplicissimus Jugend"
  { Karl Amadeus \smallCaps Hartmann }
}
```

Des Simplicius Simplicissimus Jugend
Karl
Amadeus
HARTMANN

Para agrupar los elementos de una lista de marcados anidada en una fila, aplique la instrucción `\line` a la lista de marcados. Esto apila horizontalmente los elementos de la lista de marcados, formando un marcado único.

```
\markup \center-column {
  \bold "Des Simplicius Simplicissimus Jugend"
  \line { Karl Amadeus \smallCaps Hartmann }
}
```

Des Simplicius Simplicissimus Jugend
Karl Amadeus HARTMANN

Hay una funcionalidad especial que es el manejo de instrucciones que toman elementos de marcado cuando se aplican a listas. Cuando una instrucción espera un elemento de marcado como último argumento, y se da una lista como argumento, la instrucción se aplica a cada uno de los marcados individuales dentro de la lista.

```
\markup \box { Karl Amadeus \smallCaps Hartmann }
```

Karl Amadeus HARTMANN

En este caso, el resultado es a su vez una lista de marcados, que se puede pasar a una instrucción que espere una lista de marcados o a una que espere un marcado sencillo, de nuevo con el comportamiento de asignación que se ha descrito en el caso anterior.

```
\markup \center-column \box { Karl Amadeus \smallCaps Hartmann }
\markup \rotate #30 \box { Karl Amadeus \smallCaps Hartmann }
```

Karl
Amadeus
HARTMANN

Karl Amadeus HARTMANN

Aplique `\line` a una lista de marcados para hacer que se la trate como un argumento de marcado sencillo.

```
\markup \box { Karl Amadeus \smallCaps Hartmann }
\markup \box \line { Karl Amadeus \smallCaps Hartmann }
```

Karl Amadeus HARTMANN

Karl Amadeus HARTMANN

Cuando el contenido completo de una expresión de \markup es una lista de marcados, se tipografía implícitamente usando la instrucción \line. Así, los elementos se apilan horizontalmente y se agrupan como un bloque de texto único e indivisible. La instrucción \markuplist actúa de forma diferente: espera una lista de marcados, e imprime los marcados individuales sobre la página apilándolos verticalmente, permitiendo saltos de página. El ejemplo siguiente ilustra la diferencia.

```
\markup \box \wordwrap {
  Lorem ipsum dolor sit amet, consectetur
  adipisicing elit, sed do eiusmod tempor incididunt
  ut labore et dolore magna aliqua.  Ut enim ad minim
  veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat.
}

\markuplist \box \wordwrap-lines {
  Lorem ipsum dolor sit amet, consectetur
  adipisicing elit, sed do eiusmod tempor incididunt
  ut labore et dolore magna aliqua.  Ut enim ad minim
  veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat.
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

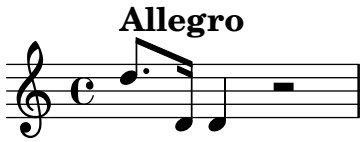
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea

commodo consequat.

Los marcados se pueden almacenar dentro de variables, para reutilizarlos en cualquier contexto en el que se acepte un elemento de marcado. Por ejemplo, estas variables se pueden adjuntar directamente a las notas:

```
allegro = \markup \bold \large Allegro

{
  d''8.^\allegro
  d'16 d'4 r2
}
```



La sintaxis \etc permite definir instrucciones abreviadas que se pueden usar como instrucciones de marcado.

```
\markup reddish = \markup \with-color "tomato" \etc
```

```
\markup { molto \reddish intenso }
```

molto intenso

El funcionamiento interno de las instrucciones de marcado y cómo implementar instrucciones más complejas, se explica en Sección “Markup functions” en *Extender*.

Véase también

Referencia de la notación: Sección 8.1.1 [Panorámica de los objetos de texto], página 311, Sección A.1 [Instrucciones de marcado de texto], página 815, Sección A.2 [Instrucciones de lista de marcado de texto], página 891.

Manual de extensión: Sección “Funciones de marcado” en *Extender*.

Archivos instalados: `scm/markup.scm`, `scm/define-markup-commands.scm`.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

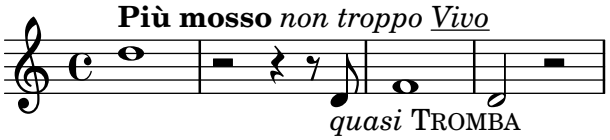
Advertencias y problemas conocidos

Los mensajes de error de sintaxis para el modo de marcado suelen ser bastante confusos.

8.2.2 Seleccionar la tipografía y su tamaño

Está contemplado de forma básica el cambio de la fuente tipográfica en el modo de marcado:

```
\relative {
  d''1^\markup {
    \bold { Più mosso }
    \italic { non troppo \underline Vivo }
  }
  r2 r4 r8
  d,_\markup { \italic quasi \smallCaps Tromba }
  f1 d2 r
}
```

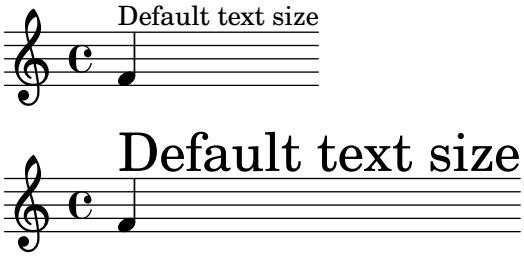


El tamaño de texto global para los elementos de marcado se puede fijar con la variable de papel `text-font-size`. Es útil para ajustar el tamaño a una fuente tipográfica distinta que puede aparecer más pequeña o más grande aunque tenga el mismo tamaño nominal de fuente. El valor se da en puntos (sin especificar las unidades); el valor predeterminado depende de la altura del pentagrama y se calcula como `(altura-del-pentagrama / 20 * 11)`.

Véase Sección 36.2 [Distancias y medidas], página 784, para más información acerca de las dimensiones usadas por LilyPond.

```
\score {
  { f'^"Default text size" }
  \layout { text-font-size = 10 }
}

\score {
```

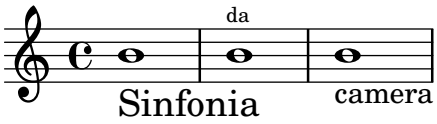
```
  { f'^"Default text size" }
  \layout { text-font-size = 20 }
}
```



El tamaño de la fuente tipográfica se puede alterar en relación al tamaño global del texto, de una serie de formas como se ve a continuación.

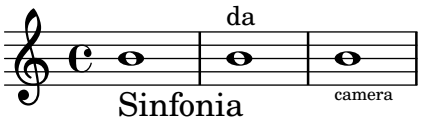
Se puede fijar a un tamaño predefinido:

```
\relative b' {
  b1_\markup { \huge Sinfonia }
  b1^\markup { \teeny da }
  b1-\markup { \normalsize camera }
}
```



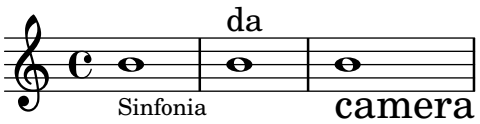
Se puede establecer a un tamaño relativo al valor anterior:

```
\relative b' {
  b1_\markup { \larger Sinfonia }
  b1^\markup { \smaller da }
  b1-\markup { \magnify #0.6 camera }
}
```



Se puede aumentar o disminuir de forma relativa al valor fijado por el tamaño global del pentagrama:

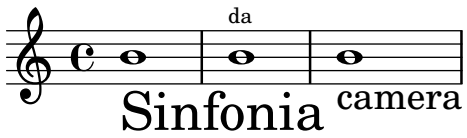
```
\relative b' {
  b1_\markup { \fontsize #-2 Sinfonia }
  b1^\markup { \fontsize #1 da }
  b1-\markup { \fontsize #3 camera }
}
```



También se puede establecer a un tamaño de puntos fijo, independientemente del tamaño global del pentagrama:

```
\relative b' {
  b1_\markup { \abs-fontsize #20 Sinfonia }
```

```
  b1^\markup { \abs-fontsize #8 da }
  b1-\markup { \abs-fontsize #14 camera }
}
```



Si el texto incluye espacios, es mejor escribir todo entre comillas, de forma que el tamaño de cada espacio sea correcto para el tamaño de los otros caracteres.

```
\markup \fontsize #6 \bold { Sinfonia da camera }
\markup \fontsize #6 \bold { "Sinfonia da camera" }
```

Sinfonia da camera
Sinfonia da camera

El texto se puede imprimir como subíndice o como superíndice. De forma predeterminada se imprimen en un tamaño menor, pero también se puede usar un tamaño normal:

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement
      \sub { (part two) }  }
  }
}
```

1 st movement
1 st movement (part two)

El modo de marcado ofrece una manera fácil de elegir familias de tipografía alternativas. A no ser que se especifique de otro modo, se selecciona automáticamente la fuente predeterminada con serifa, de tipo romano: en la última línea del ejemplo siguiente, no hay diferencia entre la primera y la segunda palabra.

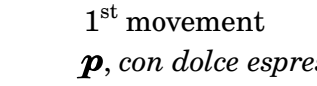
```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \serif Valentine and Proteus. }
  }
}
```

Act 1
Scene I.
Verona. An open place.
Enter Valentine and Proteus.

Algunas de estas familias, usadas para elementos específicos como números o matices, no ofrecen todos los caracteres, como se explica en Sección 3.1.3 [Indicaciones dinámicas nuevas], página 169, y Sección 4.1.8 [Marcas de repetición manual], página 207.

Si se usan dentro de una palabra, algunas instrucciones de cambio de tipografía o de formateo pueden producir un espacio vacío no deseado. Esto se puede solucionar concatenando en uno solo los distintos elementos de texto:

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
      \italic { con dolce espressione }
    }
  }
}
```



Se puede encontrar una lista exhaustiva de instrucciones de cambio y utilización personalizada de las fuentes tipográficas en Sección A.1.1 [Font markup], página 815.

También es posible definir conjuntos personalizados de fuentes, tal y como se explica en Sección 8.3 [Fuentes tipográficas], página 341.

Instrucciones predefinidas

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, `\smaller`, `\larger`.

Véase también

Referencia de la notación: Sección A.1.1 [Font markup], página 815, Sección 3.1.3 [Indicaciones dinámicas nuevas], página 169, Sección 4.1.8 [Marcas de repetición manual], página 207, Sección 8.3 [Fuentes tipográficas], página 341.

Archivos de inicio: `scm/define-markup-commands.scm`.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La utilización de las instrucciones de tamaño de texto `\teeny`, `\tiny`, `\small`, `\normalsize`, `\large` y `\huge` conducen a un espaciado de las líneas inconsistente, comparado con el uso de `\fontsize`.

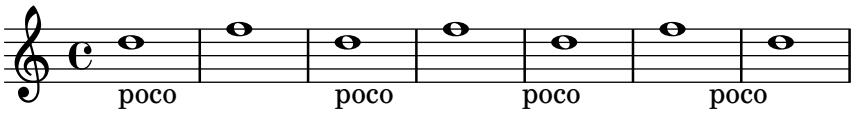
8.2.3 Alineación de texto

Esta subsección trata sobre cómo colocar texto en el modo de marcado. Los objetos de marcado también se pueden mover como un todo, usando la sintaxis que se describe en Sección “Mover objetos” en *Manual de Aprendizaje*.

Los objetos de marcado se pueden alinear de distintas maneras. De forma predeterminada, una indicación de texto se alinea sobre el borde izquierdo: en el ejemplo siguiente, no existe diferencia entre los marcados primero y segundo. El ejemplo muestra también varias formas sintácticamente correctas de ubicar las instrucciones de alineación:

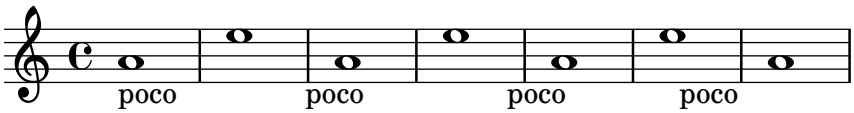
```
\relative {
```

```
  d''1-\markup { poco }
  f
  d-\markup { \left-align poco }
  f
  d-\markup { \center-align { poco } }
  f
  d-\markup \right-align { poco }
}
```



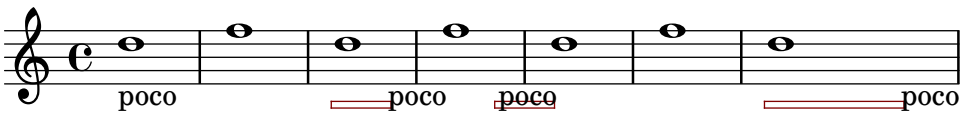
Se puede realizar un ajuste fino de la alineación horizontal usando un valor numérico:

```
\relative {
  a'1-\markup { \halign #-1 poco }
  e'
  a,-\markup { \halign #0 poco }
  e'
  a,-\markup { \halign #0.5 poco }
  e'
  a,-\markup { \halign #2 poco }
}
```



Por último, las palabras y otros objetos se pueden desplazar horizontalmente si los hacemos preceder de \hspace. También es posible usar un valor negativo que hará moverse a los objetos subsiguientes en la dirección opuesta. Aquí, escribimos \hspace dentro de la caja para mostrar mejor el efecto.

```
\relative {
  d''1-\markup { poco }
  f
  d-\markup \concat { \with-color #darkred \box \hspace #4 poco }
  f
  d-\markup \concat { \with-color #darkred \box \hspace #-4 poco }
  f
  d-\markup \concat { \with-color #darkred \box \hspace #10 poco }
}
```

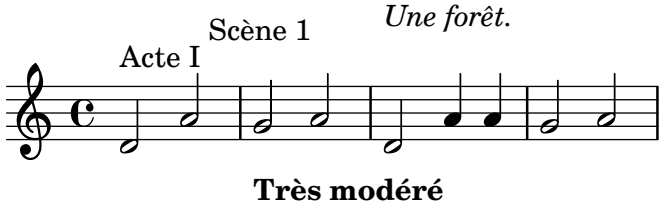


Ciertos objetos pueden poseer sus propios procedimientos de alineación, y por tanto no resultan afectados por estas instrucciones. Es posible mover estos objetos de marcado como un todo, como se muestra por ejemplo en Sección 8.1.5 [Marcas de texto], página 317.

La alineación vertical se puede establecer de una forma similar. Como se ha dicho más arriba, los objetos de marcado se pueden mover como un todo; sin embargo, también es posible mover elementos específicos dentro de un bloque de marcado.

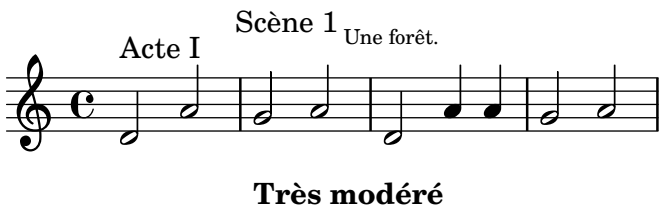
```
\relative {
```

```
  d'2^\markup {
    Acte I
    \raise #2 { Scène 1 }
  }
  a'
  g_\markup {
    \lower #4 \bold { Très modéré }
  }
  a
  d,^\markup \raise #4 \italic {
    Une forêt.
  }
  a'4 a g2 a
}
```



Algunas instrucciones pueden afectar tanto a la alineación horizontal como a la vertical de los objetos de texto en el modo de marcado:

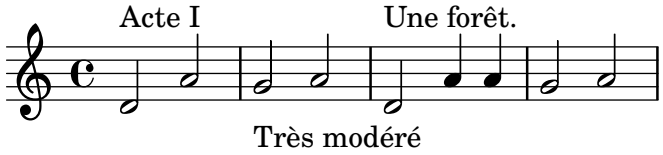
```
\relative {
  d'2^\markup {
    Acte I
    \translate #'(2 . 2) "Scène 1"
  }
  a'
  g_\markup {
    \general-align #Y #5 \bold "Très modéré"
  }
  a
  d,^\markup \translate-scaled #'(-3 . 2) \teeny {
   "Une forêt."
  }
  a'4 a g2 a
}
```



Observe que en general *no* se puede usar \vspace para subir o bajar objetos arbitrarios dentro de un marcado \column debido a la forma en que este está implementado. Así, el enfoque ingenuo que aparece a continuación no funciona.

```
\relative {
  d'2^\markup {
    Acte I
```

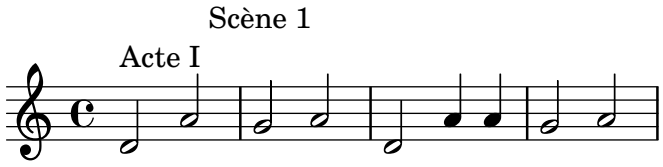
```
    \column {
      \vspace #-2
      "Scène 1"
    }
  }
  a'
  g_\markup \column {
    \vspace #1
    "Très modéré"
  }
  a
  d,^\markup \column {
    "Une forêt."
    \vspace #2
  }
  a'4 a g2 a
}
```



Lo que sí funciona es escribir \vspace entre dos objetos que tienen dimensiones no vacías.

```
\relative {
  d'2^\markup {
    Acte I
    \column {
      " "
      \vspace #-2
      "Scène 1"
    }
  }
  a'
  g_\markup \column {
    " "
    \vspace #1
    "Très modéré"
  }
  a
  d,^\markup \column {
    "Une forêt."
    \vspace #2
    " "
  }
  a'4 a g2 a
}
```

Une forêt.



Très modéré

Un objeto de marcado puede incluir varias líneas de texto. En el ejemplo siguiente, cada elemento o expresión se sitúa en su propia línea, ya sea alineada por la izquierda o centrada:

```
\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}
```

a
b c
d e f

a
b c
d e f

De forma similar, una lista de elementos o expresiones se puede repartir de forma que rellene por completo el ancho de la línea horizontal (si hay un solo elemento, se centra en el papel). A su vez, estas expresiones pueden incluir texto de varias líneas o cualquier otra expresión de marcado:

```
\markup {
  \fill-line {
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
    \line { Sir Arthur Sullivan }
  }
}
\markup {
  \fill-line { 1885 }
}
```

William S. Gilbert THE MIKADO Sir Arthur Sullivan

or

THE TOWN OF TITIPU

1885

Los elementos se pueden extender para que cubran cualquier anchura especificada mediante la sobreescritura de la propiedad `line-width`. De forma predetermianda está establecida a `#f` lo que implica toda la línea:

```
\markup {
  \column {
    \fill-line { left center right }
    \null
    \override #'(line-width . 30)
    \fill-line { left center right }
  }
}
```

left center right

left center right

Las indicaciones de texto largas se pueden también ajustar automáticamente según un ancho de línea dado. Estarán alineados por la izquierda o justificados, como se muestra en el ejemplo siguiente.

```
\markup {
  \column {
    \line  \smallCaps { La vida breve }
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada.  Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
    \hspace #0

    \line \bold { Acto II }
    \override #'(line-width . 50)
    \justify \italic {
      (Calle de Granada.  Fachada de la casa de Carmela
      y su hermano Manuel con grandes ventanas abiertas
      a través de las que se ve el patio
      donde se celebra una alegre fiesta)
    }
  }
}
```

La vida breve

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Hay una lista exhaustiva de instrucciones de alineación de texto en Sección A.1.2 [Markup for text alignment], página 827.

Véase también

Manual de aprendizaje: Sección "Mover objetos" en *Manual de Aprendizaje*.

Referencia de la notación: Sección A.1.2 [Markup for text alignment], página 827, Sección 8.1.5 [Marcas de texto], página 317.

Archivos de inicio: `scm/define-markup-commands.scm`.

Fragmentos de código: Sección "Texto" en *Fragmentos de código*.

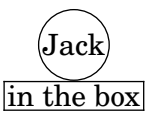
Referencia de funcionamiento interno: Sección "TextScript" en *Referencia de Funcionamiento Interno*.

8.2.4 Notación gráfica dentro de elementos de marcado

Se puede añadir diversos objetos gráficos a una partitura, utilizando instrucciones de marcado.

Ciertas instrucciones de marcado permiten la decoración de elementos de texto con gráficos, como se muestra en el ejemplo siguiente.

```
\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}
```

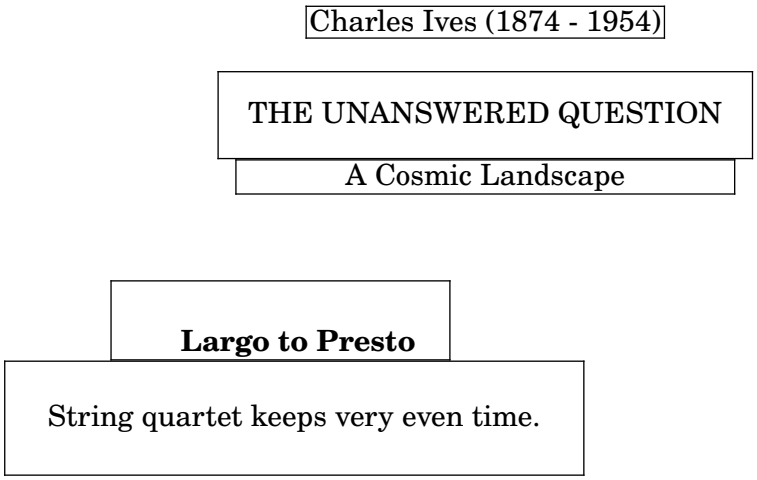


Erik Satie [1866 - 1925]

Prelude

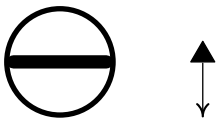
Algunas instrucciones pueden requerir un aumento del relleno alrededor del texto: esto se puede conseguir con algunas instrucciones de marcado descritas exhaustivamente en Sección A.1.2 [Markup for text alignment], página 827.

```
\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
      \bold "Largo to Presto"
  }
  \box \pad-around #3 "String quartet keeps very even time."
}
```



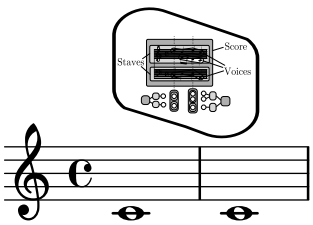
Se pueden imprimir otros símbolos o elementos gráficos sin que se requiera ningún otro texto. De igual manera que en el caso de las expresiones de marcado, dichos objetos se pueden combinar:

```
\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}
```



Entre las funcionalidades gráficas avanzadas se encuentran la posibilidad de incluir archivos de imagen externos convertidos al formato de PostScript Encapsulado (*eps*), y la inclusión directa de gráficos dentro del archivo de entrada, usando código PostScript nativo. En tal caso puede ser de utilidad la especificación explícita del tamaño del dibujo, como se ejemplifica a continuación:

```
c'1^\markup {
  \combine
    \epsfile #X #10 "./context-example.eps"
    \with-dimensions #'(-0.5 . 10.5) #'(-2 . 8)
    \postscript "
      -0.3 1 translate
      2.7 2 scale
      newpath
      2 -1 moveto
      4 -2 4 1 1 arct
      4 2 3 3 1 arct
      0 4 0 3 1 arct
      0 0 1 -1 1 arct
      closepath
      stroke"
  }
c'
```



Hay una lista exhaustiva de instrucciones específicas de gráficos en Sección A.1.3 [Graphical markup], página 845.

Véase también

Referencia de la notación: Sección A.1.2 [Markup for text alignment], página 827, Sección 36.3 [Dimensiones], página 785, Capítulo 7 [Anotaciones editoriales], página 280, Sección A.1.3 [Graphical markup], página 845.

Archivos de inicio: `scm/define-markup-commands.scm`, `scm/stencil.scm`.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

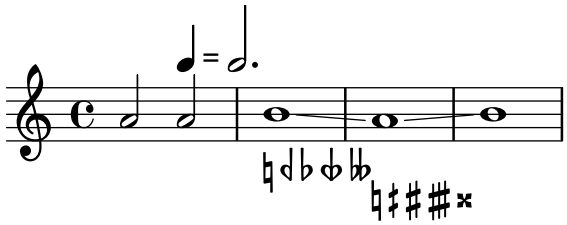
8.2.5 Notación musical dentro de elementos de marcado

Se pueden añadir a la partitura diversos elementos de notación musical, dentro de un objeto de marcado.

Las notas y las alteraciones se pueden escribir utilizando instrucciones de marcado:

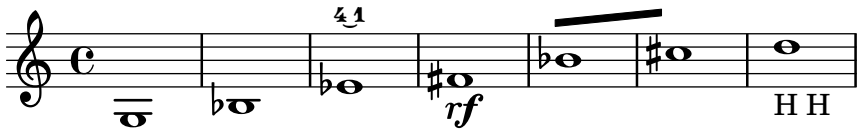
```
a'2 a'^\markup {
  \note {4} #1
```

```
  =
  \note-by-number #1 #1 #1.5
}
b'1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a'1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}
\glissando b'
```



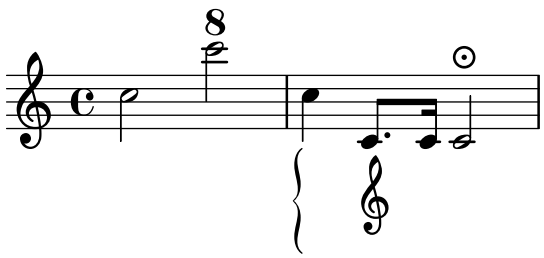
Otros objetos de notación se pueden también imprimir en el modo de marcado:

```
\relative {
  g1 bes
  ees\finger \markup \tied-lyric "4~1"
  fis_\markup { \dynamic rf }
  bes^\markup {
    \beam #8 #0.1 #0.5
  }
  cis
  d-\markup {
    \markalphabet #8
    \markletter #8
  }
}
```



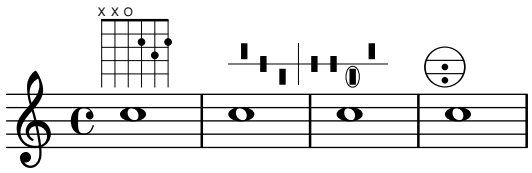
De forma más general, cualquier símbolo musical disponible se puede incluir por separado dentro de un objeto de marcado, como se ejemplifica a continuación; hay una lista exhaustiva de estos símbolos y sus nombres en Sección B.8 [La tipografía Emmentaler], página 912.

```
\relative {
  c''2
  c'^\markup { \musicglyph "eight" }
  c,4_\markup { \left-brace #40 }
  c,8._\markup { \musicglyph "clefs.G_change" }
  c16
  c2^\markup { \musicglyph "timesig.neomensural94" }
}
```



El modo de marcado también contempla diagramas para instrumentos específicos:

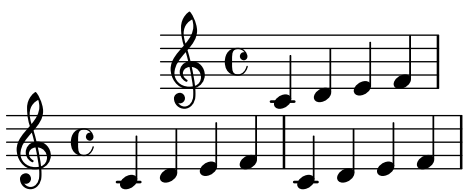
```
\relative {
  c''1^\markup {
    \fret-diagram-terse "x;x;o;2;3;2;"
  }
  c^\markup {
    \harp-pedal "^-v|--ov^"
  }
  c
  c^\markup {
    \combine
      \musicglyph "accordion.discant"
      \combine
        \raise #0.5 \musicglyph "accordion.dot"
        \raise #1.5 \musicglyph "accordion.dot"
  }
}
```



Dichos diagramas se encuentran documentados en Sección A.1.6 [Instrument-specific markup], página 869.

Incluso una partitura completa se puede incluir dentro de un objeto de marcado:

```
\relative {
  c'4 d^\markup {
    \score {
      \relative { c'4 d e f }
    }
  }
  e f |
  c d e f
}
```



Hay una lista exhaustiva de instrucciones relacionadas con la notación musical en Sección A.1.4 [Markup for music and musical symbols], página 856.

Véase también

Referencia de la notación: Sección A.1.4 [Markup for music and musical symbols], página 856, Sección B.8 [La tipografía Emmentaler], página 912.

Archivos de inicio: `scm/define-markup-commands.scm`, `scm/fret-diagrams.scm`, `scm/harp-pedals.scm`.

Fragmentos de código: Sección "Texto" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TextScript" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El espaciado vertical de un `\score` dentro de un objeto de marcado está controlado por `baseline-skip`. Todo ajuste del bloque `\paper` se ignora.

8.3 Fuentes tipográficas

Las fuentes tipográficas se manejan en LilyPond por parte de varias bibliotecas; dos de ellas son relevantes para el usuario: *FontConfig* (`https://fontconfig.org`) se usa para detectar las fuentes disponibles, y las fuentes seleccionadas se renderizan después por parte de *Pango* (`https://pango.org`) para imprimir las cadenas de texto.

Esta sección muestra cómo acceder a las fuentes tipográficas en LilyPond, y cómo cambiarlas dentro de las partituras.

8.3.1 Tipos de fuente no soportados

En primer lugar unas palabras de precaución: LilyPond crea el resultado final en PDF utilizando un archivo PosrtScript intermedio. Como consecuencia, algunos formatos de fuente tipográfica no se pueden usar, porque siendo PostScript un antiguo estándar que ya no recibe ninguna actualización, adolece de una falta de soporte para aquellas. He aquí una lista de los formatos de fuente que no funcionan.

CFF2 En los últimos años este se convirtió en el formato de fuente predeterminado para las tipografías OpenType (`.otf`). Aunque su predecesor (CFF, que usaba igualmente la extensión `.otf`) estaba soportado, el CFF2, más compacto, no lo está. LilyPond emite una advertencia en la consola si encuentra una fuente de ese tipo, y la conversión a PDF puede fallar o bien crearse sin los glifos de esta fuente.

Font Variations
Estas fuentes normalmente tienen las letras 'VF' como parte del nombre; desgraciadamente no están soportadas independientemente de si son Font Variations de TrueType o de OpenType, y por ello se rechazan por adelantado.

fuentes en color
No soportadas; esto afecta en particular a las fuentes de Emoji escalables. Algunas fuentes tipográficas, sin embargo, también contienen versiones en blanco y negro de los Emojis (como glifos de contorno normales), y estos deberían funcionar perfectamente (siempre y cuando vinieran en un formato de fuente que estuviera soportado).

colecciones de OpenType
Esto es un formato contenedor (usualmente con la extensión `.otc`) que alberga varias fuentes dentro del mismo archivo. Hoy en día se usan como fuente de último recurso, esto es, se usa por parte del sistema operativo si por algún motivo no se encuentra ninguna otra que sea válida.

fuentes de bitmap
: No están soportadas en absoluto y se rechazan automáticamente antes de que LilyPond llegue a ver la lista de las fuentes disponibles. Esto incluye algunas fuentes antiguas de Emojis en color.

Muchas fuentes se encuentran disponibles en los dos formatos, TrueType y OpenType; si falla la versión `.otf`, pruebe con la versión `.ttf`. Si no encuentra ninguna alternativa a una fuente CFF2, pruebe la opción de la línea de órdenes `-dbackend=cairo` de LilyPond; este *backend*, aún experimental, puentea la fase de PostScript y crea directamente salida en PDF, siendo así capaz de soportar el formato CFF2 (y también las colecciones de OpenType).

8.3.2 Búsqueda de fuentes

Además de cualquier fuente tipográfica que ya esté instalada en el sistema operativo, se pueden añadir fuentes adicionales al conjunto que detecta FontConfig (y por ello disponibles en las partituras de LilyPond) a través de las instrucciones siguientes:

```
#(ly:font-config-add-font "ruta/hacia/archivo-de-fuente")
#(ly:font-config-add-directory "ruta/hacia/directorio/")
```

Las dos instrucciones admiten rutas absolutas y relativas, lo que permite compilar una partitura en cualquier sistema, simplemente distribuyendo los archivos de fuente necesarios junto al código fuente de la partitura.

Para verificar que las fuentes deseadas se encuentran por parte de FontConfig, use la instrucción `#(ly:font-config-display-fonts)`, que imprime en el archivo de registro de la consola la lista completa de fuentes disponibles. También muestra los nombres de fuente efectivos a usar con LilyPond; estos pueden ser distintos de los propios nombres de archivo. Como alternativa, lanzando en un terminal la orden `lilypond -dshow-available-fonts` se consigue el mismo efecto.

8.3.3 Familias de fuentes

Están disponibles tres alias genéricos para las familias[1] de fuentes de texto: 'serif', 'sans', y 'máquina de escribir'. Dependiendo del *backend*, estas familias se asignan a diferentes alias de familias de fuente.

Para el *backend* `svg`:

familia genérica	**familia de fuente SVG**
serif	serif
sans	sans-serif
máquina de escribir	monospace

'serif', 'sans-serif', y 'monoespaciado' son la 'generic-family' en las especificaciones SVG y CSS.

Para otros *backends*:

familia genérica	**alias predeterminado de la familia**	**familias contenidas en el alias**
serif	LilyPond Serif	C059, Century SchoolBook URW, Century Schoolbook L, TeX Gyre Schola, DejaVu Serif, . . . , serif
sans	LilyPond Sans Serif	Nimbus Sans, Nimbus Sans L, TeX Gyre Heros, DejaVu Sans, . . . , sans-serif
máquina de escribir	LilyPond Monospace	Nimbus Mono PS, Nimbus Mono, Nimbus Mono L, TeX Gyre Cursor, DejaVu Sans Mono, . . . , monospace

[1] En su forma más simple, una *familia de fuentes* contiene normalmente fuentes en estilos romano, itálica, negrita e itálica negrita.

Si un carácter no existe en una fuente determinada de la primera familia de la lista, se usa en su lugar para ese carácter la fuente adecuada de la siguiente familia de la lista.

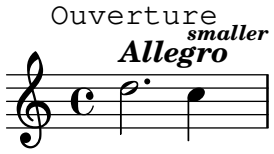
Observe que las familias de fuente URW distribuidas con LilyPond ('C059', 'Nimbus Sans' y 'Nimbus Mono PS') tienen una peculiaridad: por omisión, además de las ligaduras estándar como 'fl' o 'ffi', sustituyen la cadena de carcateres 'Nr.' con el símbolo de Número (U+2116) si se selecciona el script 'latn'. Para resolver el problema localmente, inserte un carácter *zero-width non-joiner* (ZWNJ, U+200C) entre los caracteres 'N' y 'r'. Para solucionarlo globalmente, use el siguiente código para hacer que LilyPond siempre inserte el caácter ZWNJ.

```
\paper {
  #(add-text-replacements!
      `(("Nr." . ,(format #f "N~ar." (ly:wide-char->utf-8 #x200C)))))
}
```

'LilyPond Serif', 'LilyPond Sans Serif' y 'LilyPond Monospace' son alias de familia de fuente definidos en el archivo de configuración adicional de FontConfig `00-lilypond-fonts.conf`, que se encuentra normalmente en el directorio `/usr/local/share/lilypond/2.26.0/fonts`, y que se usa exclusivamente por parte de LilyPond.

Cada familia de fuentes puede incluir distintas formas y series. El ejemplo siguiente muestra esto, incluyendo código para cambiar también el tamaño. El valor proporcionado a `font-size` se toma en relación al tamaño predeterminado de la tipografía.

```
\override Score.TextMark.font-family = #'typewriter
\textMark "Ouverture"
\override Voice.TextScript.font-shape = #'italic
\override Voice.TextScript.font-series = #'bold
d''2.^\markup "Allegro"
\override Voice.TextScript.font-size = -3
c''4^"smaller"
```



Se puede usar una sintaxis similar en el modo de marcado, aunque en este caso es preferible usar la sintaxis sencilla que se explica en Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327:

```
\markup {
  \column {
    \line {
      \override #'((font-shape . italic) (font-size . 4))
      Idomeneo,
    }
    \line {
      \override #'(font-family . typewriter) {
        \override #'(font-series . bold) re
        di
      }
      \override #'(font-family . sans) Creta
    }
  }
}
```

Idomeneo,
re di Creta

8.3.4 Funcionalidades de fuente

Al usar fuentes tipográficas de OpenType, pueden usarse las funcionalidades de fuente o *font features*.[2] Observe que no todas las fuentes de OpenType tienen todas las funciones. Si se requiere una funcionalidad que no existe en la fuente seleccionada, la funcionalidad sencillamente se ignora. El ejemplo que aparece más abajo usa la fuente 'TeX Gyre Schola' (esto es, el estilo romano de la familia).

```
\paper {
  property-defaults.fonts.serif = "TeX Gyre Schola"
}

\markup "normal style: Hello HELLO"

\markup \caps "small caps: Hello"

\markup \override #'(font-features . ("smcp")) "true small caps: Hello"

\markup "normal number style: 0123456789"

\markup \override #'(font-features . ("onum"))
  "old number style: 0123456789"

\markup \override #'(font-features . ("salt 0"))
  "stylistic alternate 0: εφπρθ"

\markup \override #'(font-features . ("salt 1"))
  "stylistic alternate 1: εφπρθ"

\markup \override #'(font-features . ("onum" "smcp" "salt 1"))
  "multiple features: Hello 0123456789 εφπρθ"
```

normal style: Hello HELLO

SMALL CAPS: HELLO

TRUE SMALL CAPS: HELLO

normal number style: 0123456789

old number style: 0123456789

stylistic alternate 0: εφπρθ

stylistic alternate 1: εϕϖϱϑ

MULTIPLE FEATURES: HELLO 0123456789 εϕϖϱϑ

[2] No se pueden especificar los idiomas ni los scripts de OpenType.

Para ver la lista completa de funcionalidades de las fuentes OpenType, consulte `https://www.microsoft.com/typography/otspec/featurelist.htm`; para la identificación de las funcionalidades de las fuentes de OpenType véase `https://lists.gnu.org/archive/html/lilypond-devel/2017-08/msg00004.html`.

Véase también

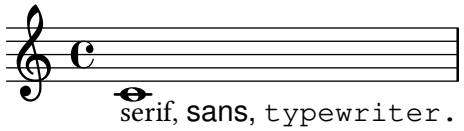
Referencia de la notación: Sección B.8 [La tipografía Emmentaler], página 912, Sección 8.2.5 [Notación musical dentro de elementos de marcado], página 338, Sección 36.8 [Rotación de objetos], página 801, Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327, Sección A.1.1 [Font markup], página 815. Sección 8.3.2 [Búsqueda de fuentes], página 342, Sección 8.3.3 [Familias de fuentes], página 342, Sección 8.4 [Cambiar las fuentes tipográficas], página 345.

Fragmentos de código: Sección "Texto" en *Fragmentos de código*.

8.4 Cambiar las fuentes tipográficas

Es posible cambiar las fuentes usadas en las familias de la fuente predeterminada de LilyPond.

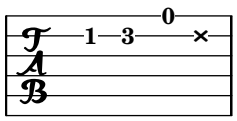
```
\paper {
  property-defaults.fonts.serif = "Linux Libertine O"
  property-defaults.fonts.sans = "DejaVu Sans"
  property-defaults.fonts.typewriter = "DejaVu Sans Mono"
}

\relative c'{
  c1-\markup {
    serif,
    \sans sans,
    \typewriter typewriter. }
}
```



Se puede usar la misma sintaxis para cambiar la fuente de la música; véase Sección 23.5 [Sustituir la tipografía de la notación], página 652.

Para cambiar las fuentes usadas para un objeto gráfico específico, o para una parte específica de un elemento de marcado, sobreescriba la propiedad `fonts`. El siguiente ejemplo cambia la fuente para las "cabezas de nota" normales de tablatura (que usan la familia `serif`) manteniendo al mismo tiempo la fuente predeterminada para las que se dibujan como unas aspas (que usan la familia `music`).

```
\layout {
  \override TabVoice.TabNoteHead.property-defaults.fonts.serif =
    "Linux Libertine O"
}

\new TabStaff { c' d' e' \deadNote c' }
```



Más abajo aparece un ejemplo de la sobreescritura de `fonts` en los elementos de marcado:

```
\markup \override #'(fonts . ((serif . "Linux Libertine O")
                              (typewriter . "DejaVu Sans Mono"))) {
  Copyright © John Doe \typewriter john@doe.org
}
```

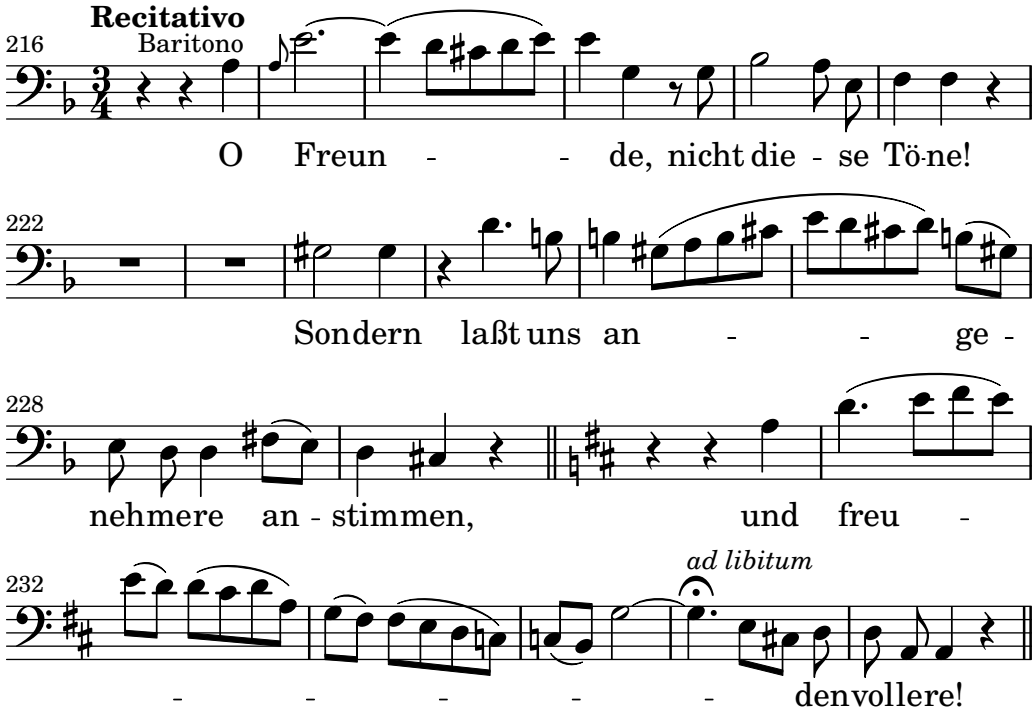
Copyright © John Doe `john@doe.org`

Véase también

Referencia de la notación: Sección 8.3.2 [Búsqueda de fuentes], página 342, Sección 8.3.3 [Familias de fuentes], página 342, Sección 8.2.2 [Seleccionar la tipografía y su tamaño], página 327, Sección A.1.1 [Font markup], página 815, Sección 23.5 [Sustituir la tipografía de la notación], página 652.

Notación especializada

9 Música vocal



Esta sección explica cómo tipografiar música vocal, y cómo asegurarse de que la letra se alinea con las notas de su melodía correspondiente.

9.1 Notación común para música vocal

Esta sección se ocupa de las cuestiones comunes a la mayoría de los tipos de música vocal.

9.1.1 Referencias para música vocal

Esta sección indica dónde encontrar los detalles de las cuestiones de notación que pueden surgir en cualquier tipo de música vocal.

- Casi todos los estilos de música vocal utilizan texto escrito como letra. Hay una introducción a esta notación en Sección “Elaborar canciones sencillas” en *Manual de Aprendizaje*.
- La música vocal probablemente requiere el uso del modo de marcado o `markup`, ya sea para la letra o para otros elementos de texto (nombres de los personajes, etc.). Esta sintaxis está descrita en Sección 8.2.1 [Introducción al marcado de texto], página 323.
- Los *ambitus* o indicaciones de tesitura vocal se pueden añadir al principio de los pentagramas vocales, como se explica en Sección 1.3.7 [Tesitura], página 41.
- De forma predeterminada, las indicaciones dinámicas se sitúan debajo del pentagrama, pero en la música coral se suelen disponer por encima del pentagrama para evitar la letra, como se explica en Sección 9.5.2 [Disposiciones de la partitura para música coral], página 392.

Véase también

Glosario musical: Sección “ambitus” en *Glosario Musical*.

Manual de aprendizaje: Sección “Elaborar canciones sencillas” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 8.2.1 [Introducción al marcado de texto], página 323, Sección 1.3.7 [Tesitura], página 41, Sección 9.5.2 [Disposiciones de la partitura para música coral], página 392.

Fragmentos de código: Sección “Música vocal” en *Fragmentos de código*.

9.1.2 Introducir la letra

La letra de las canciones se introduce en un modo de entrada especial que se inicia mediante la palabra clave `\lyricmode`, o bien mediante `\addlyrics` ó `\lyricsto`. En este modo especial de entrada, el carácter `d` no se analiza como una nota, sino más bien como una sílaba de una sola letra. Dicho de otra forma, las sílabas se introducen como las notas, pero la altura de las notas se sustituye por texto.

Por ejemplo:

```
\lyricmode { Three4 blind mice,2 three4 blind mice2 }
```

Existen dos métodos principales para especificar la colocación horizontal de las sílabas: mediante la indicación de la duración de cada sílaba explícitamente, como en el ejemplo anterior, o dejando que la letra se alinee automáticamente con una melodía u otra voz de música, utilizando `\addlyrics` o `\lyricsto`. El primer método se describe a continuación bajo el epígrafe Sección 9.1.5 [Duración manual de las sílabas], página 356. El segundo método se describe en Sección 9.1.4 [Duración automática de las sílabas], página 353.

Una palabra o sílaba de la letra comienza con un carácter alfabético (y algunos otros caracteres, véase más abajo) y termina con un espacio o un dígito. El resto de los caracteres dentro de la sílaba pueden ser cualesquier caracteres que no sean dígitos o espacios.

A causa de que cualquier carácter que no es un dígito o un espacio se considera parte de la sílaba, una palabra es válida incluso si termina con `}`, lo que a menudo lleva al siguiente fallo:

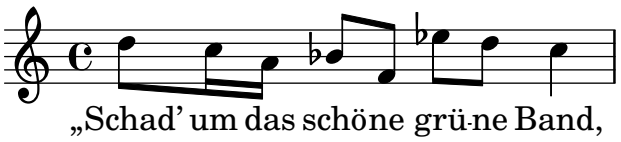
```
\lyricmode { la la la}
```

En este ejemplo, el símbolo `}` se encuentra incluido dentro de la sílaba final, por tanto la llave de apertura no se compensa con la correspondiente llave de cierre y el archivo de entrada probablemente no se podrá procesar. En vez de ello, las llaves siempre deben ir rodeadas por espacios:

```
\lyricmode { la la la }
```

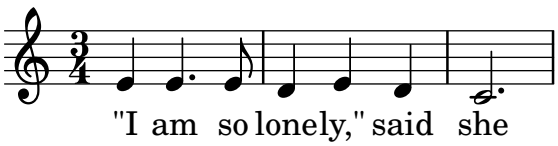
Los signos de puntuación, las letras con caracteres acentuados o de cualquier idioma distinto del inglés, o los caracteres especiales (como el símbolo del corazón o las comillas inclinadas), se pueden insertar directamente en el archivo de entrada, siempre que éste se grabe con la codificación UTF-8. Para más información, consulte Sección 22.4 [Caracteres especiales], página 647.

```
\relative { d''8 c16 a bes8 f ees' d c4 }
\addlyrics { „Schad’ um das schö -- ne grü -- ne Band, }
```



Se pueden usar las comillas normales dentro de la letra, pero deben ir precedidas por el carácter de barra invertida y la sílaba en su conjunto se debe encerrar en otro par de comillas. Por ejemplo,

```
\relative { \time 3/4 e'4 e4. e8 d4 e d c2. }
\addlyrics { "\"I" am so lone -- "ly,\"" said she }
```



La definición completa del comienzo de una palabra en el modo Lyrics (letra) es algo más compleja. Una palabra en el modo Lyrics comienza por: un carácter alfabético, `_`, `?`, `!`, `:`, `'`, los

caracteres de control desde ^A hasta ^F, desde ^Q hasta ^W, ^Y, ^^, cualquier carácter de 8 bits con código ASCII por encima del 127, o una combinación de dos caracteres consistente en la combinación de una barra invertida seguida por `, ', " ó ^.

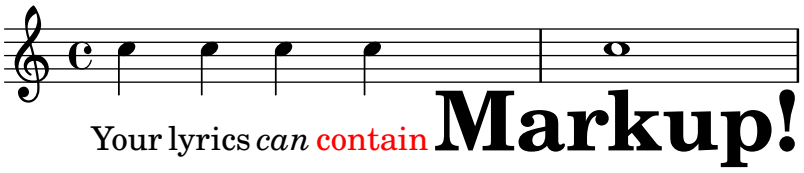
Un enorme control sobre la apariencia de la letra proviene del uso de elementos de marcado \markup dentro de la propia letra. Para ver una explicación de muchas de las opciones, consulte Sección 8.2 [Formatear el texto], página 323.

Fragmentos de código seleccionados

Dar formato a sílabas de la letra

Es posible usar la instrucción \markup dentro de un bloque \lyricmode para dar formato a las sílabas individuales dentro de la letra.

```
mel = \relative c'' { c4 c c c c1 }
lyr = \lyricmode {
  Your lyrics \markup { \italic can }
  \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



Véase también

Manual de aprendizaje: Sección “Canciones” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 9.1.4 [Duración automática de las sílabas], página 353, Sección 8.3 [Fuentes tipográficas], página 341, Sección 8.2 [Formatear el texto], página 323, Capítulo 19 [Modos de entrada], página 587, Sección 9.1.5 [Duración manual de las sílabas], página 356, Sección 22.4 [Caracteres especiales], página 647.

Referencia de funcionamiento interno: Sección “LyricText” en *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Texto” en *Fragmentos de código*.

9.1.3 Alineación de la letra a una melodía

La letra se interpreta en el modo \lyricmode y se imprime dentro del contexto llamado Lyrics, véase Sección 33.1 [Explicación de los contextos], página 742.

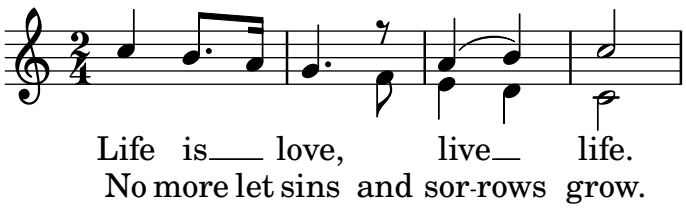
```
\new Lyrics \lyricmode { ... }
```

Dos variantes de \lyricmode además disponen un contexto asociado que se usa para sincronizar las sílabas de la letra con la música. \addlyrics, que es más conveniente, sigue inmediatamente al contenido musical del contexto de Voz con el que se debe sincronizar, creando implícitamente un contexto Lyrics completo. \lyricsto es más versátil y requiere tanto que se especifique el contexto Voice asociado por su nombre, como que se cree explícitamente un contexto contenedor Lyrics. Para ver más detalles, consulte Sección 9.1.4 [Duración automática de las sílabas], página 353.

La letra se puede alinear con la melodía de dos maneras principales:

- Se puede alinear la letra automáticamente, tomándose las duraciones de las sílabas de otra voz de música, o (en circunstancias especiales) una melodía asociada, usando `\addlyrics`, `\lyricsto`, o estableciendo el valor de la propiedad `associatedVoice`. Para ver más detalles, consulte Sección 9.1.4 [Duración automática de las sílabas], página 353.

```
<<
  \new Staff <<
    \time 2/4
    \new Voice = "one" \relative {
      \voiceOne
      c''4 b8. a16 g4. r8 a4 ( b ) c2
    }
    \new Voice = "two" \relative {
      \voiceTwo
       s2 s4. f'8 e4 d c2
    }
  >>

% takes durations and alignment from notes in "one"
  \new Lyrics \lyricsto "one" {
    Life is __ _ love, live __ life.
  }

% takes durations and alignment from notes in "one" initially
% then switches to "two"
  \new Lyrics \lyricsto "one" {
    No more let
    \set associatedVoice = "two"  % must be set one syllable early
    sins and sor -- rows grow.
  }
>>
```



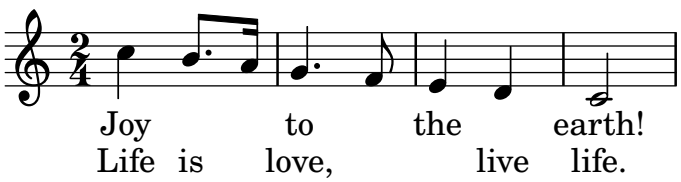
La primera línea de letra muestra la forma normal de introducir la letra.

La segunda línea de texto muestra cómo se puede cambiar la voz de la que se toman las duraciones para las sílabas. Esto es útil si el texto de las distintas estrofas corresponde a las mismas notas de formas diferentes, y todas las duraciones están disponibles dentro de contextos de voz. Para ver más detalles, consulte Sección 9.3 [Versos], página 382.

- El texto se puede alinear independientemente de la duración de cualquier nota si las duraciones de las sílabas se especifican explícitamente, y se escriben con `\lyricmode`.

```
<<
  \new Voice = "one" \relative {
    \time 2/4
    c''4 b8. a16 g4. f8 e4 d c2
  }
```

```
% uses previous explicit duration of 2;
  \new Lyrics \lyricmode {
    Joy to the earth!
  }

% explicit durations, set to a different rhythm
  \new Lyrics \lyricmode {
    Life4 is love,2. live4 life.2
  }
>>
```



El primer verso no está alineado con las notas porque no se especificaron las duraciones, y se utiliza el valor previo de 2 (blanca) para todas las sílabas.

El segundo verso muestra cómo las palabras se pueden alinear de forma independiente de las notas. Esto es útil si el texto de los distintos versos se corresponde con las notas de maneras diferentes, pero las duraciones requeridas no están disponibles en un contexto de música. Para ver más detalles, consulte Sección 9.1.5 [Duración manual de las sílabas], página 356. Esta técnica también es útil cuando se quiere preparar un diálogo encima de la música; para ver ejemplos que lo muestran, consulte Sección 9.6.5 [Diálogos encima de la música], página 401.

Véase también

Manual de aprendizaje: Sección "Alineación de la letra a una melodía" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 33.1 [Explicación de los contextos], página 742, Sección 9.1.4 [Duración automática de las sílabas], página 353, Sección 9.3 [Versos], página 382, Sección 9.1.5 [Duración manual de las sílabas], página 356, Sección 9.6.5 [Diálogos encima de la música], página 401, Sección 9.1.5 [Duración manual de las sílabas], página 356.

Referencia de funcionamiento interno: Sección "Lyrics" en *Referencia de Funcionamiento Interno*.

9.1.4 Duración automática de las sílabas

Las sílabas de la letra se pueden alinear automáticamente con las notas de una melodía de tres formas:

- especificando por su nombre el contexto Voice que contiene la melodía, con `\lyricsto`,
- introduciendo la letra con `\addlyrics` y colocándola inmediatamente después del contexto de voz que contiene la melodía,
- estableciendo un valor para la propiedad `associatedVoice`, la alineación de la letra se puede mover a un contexto de voz distinto, especificado por su nombre, en cualquier momento musical.

En los tres métodos se pueden trazar guiones de separación entre las sílabas de una palabra y líneas extensoras después del final de una palabra. Para ver más detalles, véase Sección 9.1.8 [Líneas de extensión y guiones], página 361.

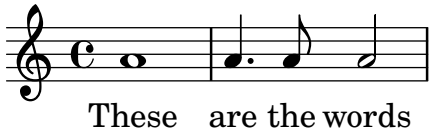
El contexto `Voice` que contiene la melodía con que se está alineando la letra, no debe haber "muerto", o se perderá la letra a partir de este punto. Esto puede ocurrir si existen períodos en

que dicha voz no tiene nada que hacer. Para ver métodos para mantener vivos los contextos, consulte Sección 33.3 [Mantener vivos los contextos], página 749.

Uso de \lyricsto

Las sílabas de la letra se pueden alinear automáticamente bajo las notas de una melodía especificando por su nombre el contexto de voz que contiene la melodía, con `\lyricsto`:

```
<<
  \new Voice = "melody" \relative {
    a'1 a4. a8 a2
  }
  \new Lyrics \lyricsto "melody" {
    These are the words
  }
>>
```



Esto alinea las sílabas con las notas del contexto `Voice` nombrado, que debe existir previamente. Por tanto, normalmente se especifica primero el contexto `Voice` seguido del contexto `Lyrics`. La letra en sí sigue a la instrucción `\lyricsto`. La instrucción `\lyricsto` invoca automáticamente el modo de letra. De forma predeterminada, la letra se coloca por debajo de las notas. Para otras colocaciones, consulte Sección 9.2.2 [Posicionamiento vertical de la letra], página 364.

Uso de \addlyrics

La instrucción `\addlyrics` es realmente tan sólo una forma cómoda de escribir una estructura de LilyPond más complicada que establece la letra.

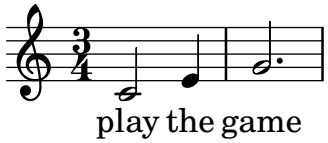
```
{ MÚSICA }
\addlyrics { LETRA }
```

es lo mismo que

```
\new Voice = "blabla" { MÚSICA }
\new Lyrics \lyricsto "blabla" { LETRA }
```

He aquí un ejemplo:

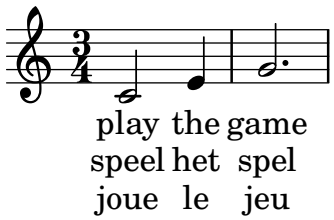
```
{
  \time 3/4
  \relative { c'2 e4 g2. }
  \addlyrics { play the game }
}
```



Se pueden añadir más versos poniendo más secciones `\addlyrics`:

```
{
  \time 3/4
  \relative { c'2 e4 g2. }
  \addlyrics { play the game }
```

```
  \addlyrics { speel het spel }
  \addlyrics { joue le jeu }
}
```



La instrucción `\addlyrics` no es capaz de manejar situaciones de polifonía. Asimismo, no puede usarse para asociar letra a un contexto de tablatura `TabVoice`. Para estos casos debería usar `\lyricsto`.

Uso de `associatedVoice`

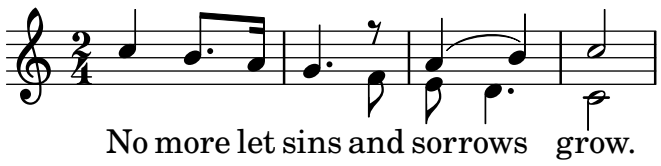
Se puede cambiar la melodía a la que se alinea la letra mediante el establecimiento de la propiedad `associatedVoice`,

```
\set associatedVoice = "lala"
```

El valor de la propiedad (aquí: `"lala"`) debe ser el nombre de un contexto `Voice`. Por razones técnicas, la instrucción `\set` se debe escribir una sílaba antes de aquella a la que se quiere aplicar el cambio de voz.

He aquí un ejemplo que muestra su uso:

```
<<
  \new Staff <<
    \time 2/4
    \new Voice = "one" \relative {
      \voiceOne
      c''4 b8. a16 g4. r8 a4 ( b ) c2
    }
    \new Voice = "two" \relative {
      \voiceTwo
       s2 s4. f'8 e8 d4. c2
    }
  >>
% takes durations and alignment from notes in "one" initially
% then switches to "two"
  \new Lyrics \lyricsto "one" {
    No more let
    \set associatedVoice = "two"  % must be set one syllable early
    sins and sor -- rows grow.
  }
>>
```



Véase también

Referencia de la notación: Sección 9.1.8 [Líneas de extensión y guiones], página 361, Sección 33.3 [Mantener vivos los contextos], página 749, Sección 9.2.2 [Posicionamiento vertical de la letra], página 364.

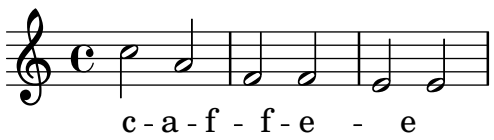
9.1.5 Duración manual de las sílabas

En ciertas músicas vocales complejas puede ser deseable colocar la letra de forma completamente independiente de las notas. En este caso, no use `\lyricsto` ni `\addlyrics` y no establezca un valor para `associatedVoice`. Las sílabas se escriben como notas – pero sustituyendo los nombres de las notas por texto – y la duración de cada sílaba se escribe explícitamente después de la sílaba.

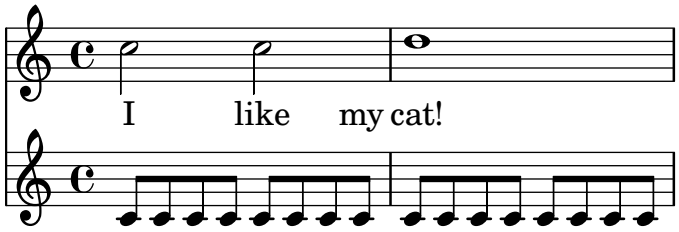
Se pueden trazar líneas separadores entre las sílabas, como es usual, pero no se pueden trazar líneas extensoras cuando no hay una voz asociada.

He aquí dos ejemplos:

```
<<
  \new Voice = "melody" \relative {
    c''2 a f f e e
  }
  \new Lyrics \lyricmode {
    c4. -- a -- f -- f -- e2. -- e
  }
>>
```



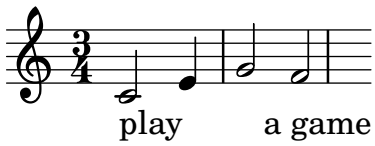

```
<<
  \new Staff {
    \relative {
      c''2 c2
      d1
    }
  }
  \new Lyrics {
    \lyricmode {
      I2 like4. my8 cat!1
    }
  }
  \new Staff {
    \relative {
      c'8 c c c c c c c
      c8 c c c c c c c
    }
  }
>>
```



Esta técnica es muy útil cuando se escriben diálogos encima de la música, véase Sección 9.6.5 [Diálogos encima de la música], página 401.

Para cambiar la alineación de las sílabas, simplemente sobreescriba la propiedad `self-alignment-X`:

```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    c'2 e4 g2 f
  }
  \new Lyrics \lyricmode {
    \override LyricText.self-alignment-X = #LEFT
    play1 a4 game4
  }
>>
```



Véase también

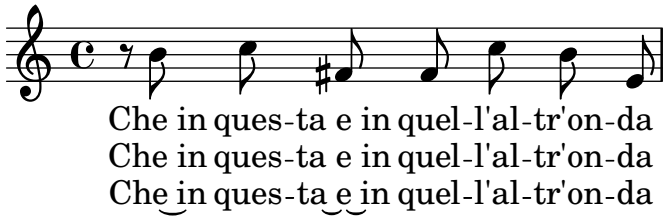
Referencia de la notación: Sección 33.3 [Mantener vivos los contextos], página 749.

Referencia de funcionamiento interno: Sección “Lyrics” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*.

9.1.6 Varias sílabas sobre una nota

Para asignar más de una sílaba a una única nota con espacios entre las sílabas, podemos encerrar la frase entre comillas o utilizar un carácter de guión bajo _. De forma alternativa, podemos usar el símbolo de tilde curva (~) para obtener una ligadura de texto.

```
{
  \relative {
    \autoBeamOff
    r8 b' c fis, fis c' b e,
  }
  \addlyrics
  {
    % Ensure hyphens are visible
    \override LyricHyphen.minimum-distance = 1.0
    Che_in ques -- ta_e_in quel -- l'al -- tr'on -- da
  }
  \addlyrics { "Che in" ques -- "ta e in" quel -- l'al -- tr'on -- da }
  \addlyrics { Che~in ques -- ta~e~in quel -- l'al -- tr'on -- da }
}
```



Véase también

Referencia de funcionamiento interno: Sección "LyricCombineMusic" en *Referencia de Funcionamiento Interno*.

9.1.7 Varias notas sobre una sílaba

A veces, y sobre todo en la música medieval y del Barroco, varias notas se cantan sobre una sílaba única; tales vocalizaciones reciben el nombre de melismas, o melismata (véase Sección "melisma" en *Glosario Musical*). La sílaba de un melisma se suele alinear por la izquierda con la primera nota del melisma.

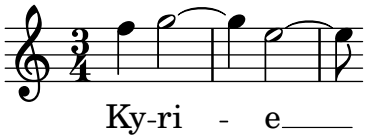
Cuando se produce un melisma sobre una sílaba distinta de la última de una palabra, dicha sílaba se suele unir a la siguiente con un guión separador. Esto se indica en el código de entrada escribiendo un doble guión, `--`, inmediatamente después de la sílaba.

De forma alternativa, cuando se produce un melisma sobre la última o sobre la única sílaba de una palabra, se suele trazar una línea extensora desde el final de la sílaba hasta la última nota del melisma. Esto se indica en el código de entrada escribiendo una doble barra baja o carácter de subrayado, `__`, inmediatamente después de la palabra.

Existen cinco formas de indicar los melismas:

- Se crean melismas automáticamente sobre las notas unidas mediante ligadura de unión:

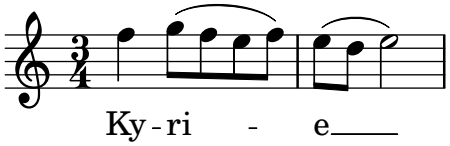
```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    f''4 g2 ~ |
    4 e2 ~ |
    8
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e __
  }
>>
```



- Se pueden crear melismas automáticamente a partir de la música escribiendo ligaduras de expresión sobre las notas de cada melisma. Ésta es la forma usual de escribir la letra:

```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    f''4 g8 ( f e f )
    e8 ( d e2 )
  }
  \new Lyrics \lyricsto "melody" {
```

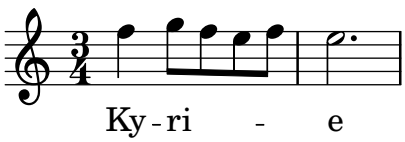
```
    Ky -- ri -- e __
  }
>>
```



Observe que las ligaduras de fraseo no afectan a la creación de melismas.

- Se considera que las notas forman un melisma cuando se unen manualmente mediante barra, siempre y cuando el barrado automático esté desactivado. Véase Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104.

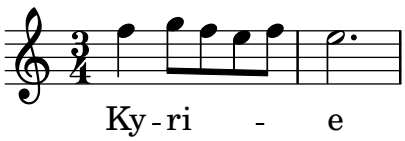
```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    \autoBeamOff
    f''4 g8[ f e f]
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```



Claramente, esto no es adecuado para los melismas formados por notas de duración más larga que la corchea.

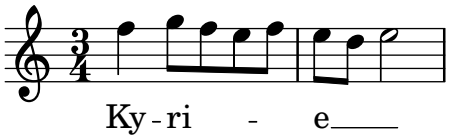
- Un grupo de notas sin ligadura de expresión se trata como un melisma si están comprendidas entre `\melisma` y `\melismaEnd`.

```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    f''4 g8
    \melisma
    f e f
    \melismaEnd
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```



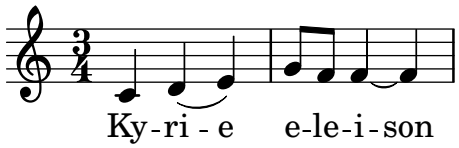
- Se puede definir un melisma enteramente dentro de la letra escribiendo un carácter de barra baja suelto, _, por cada nota adicional que se quiere añadir al melisma.

```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    f''4 g8 f e f
    e8 d e2
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ _ _ e __ _ _
  }
>>
```



Es posible tener ligaduras de unión o de expresión y barras manuales en la melodía sin que indiquen melisma. Para hacerlo, ajuste el valor de `melismaBusyProperties`:

```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    \set melismaBusyProperties = #'()
    c'4 d ( e )
    g8 [ f ] f4 ~ 4
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e e -- le -- i -- son
  }
>>
```



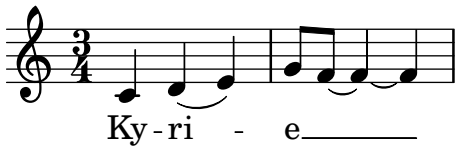
Se pueden usar otros valores para `melismaBusyProperties` si queremos incluir o excluir selectivamente las ligaduras de unión, ligaduras de expresión o barras de la detección automática de los melismas; véase `melismaBusyProperties` en el apartado Sección "Tunable context properties" en *Referencia de Funcionamiento Interno*.

Como alternativa, si todas las indicaciones de melismas se van a ignorar, se puede establecer `ignoreMelismata` al valor verdadero; véase Sección 9.3.4 [Versos con ritmos distintos], página 384.

Si se requiere un melisma en el transcurso de un pasaje en el que `melismaBusyProperties` está activo, puede indicarse colocando una sola barra baja en la letra por cada nota que se debe incluir dentro del melisma:

```
<<
  \new Voice = "melody" \relative {
    \time 3/4
    \set melismaBusyProperties = #'()
    c'4 d ( e )
    g8 [ f ] ~ 4 ~ f
```

```
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ e __ _ _ _
  }
>>
```



Instrucciones predefinidas

`\autoBeamOff`, `\autoBeamOn`, `\melisma`, `\melismaEnd`.

Véase también

Glosario musical: Sección "melisma" en *Glosario Musical*.

Manual de aprendizaje: Sección "Alineación de la letra a una melodía" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 9.1.3 [Alineación de la letra a una melodía], página 351, Sección 9.1.4 [Duración automática de las sílabas], página 353, Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104, Sección 9.3.4 [Versos con ritmos distintos], página 384.

Referencia de funcionamiento interno: Sección "Tunable context properties" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No se crean automáticamente líneas extensoras bajo los melismas; se deben insertar manualmente con un doble guión bajo.

9.1.8 Líneas de extensión y guiones

Los melismas se indican mediante una línea horizontal centrada entre una sílaba y la siguiente. Esta línea recibe el nombre de línea de extensión, y se escribe manualmente como ' `__` ' (fíjese en los espacios antes y después de los dos guiones bajos). También se pueden crear líneas de extensión automáticamente fijando el valor de `autoExtenders` a `#t`.

> **Nota:** Los melismas se indican en la partitura con líneas extensoras, que se escriben con un doble guión bajo; pero también se pueden introducir melismas cortos mediante saltos de notas individuales, que se escriben como caracteres de guión bajo sueltos; de forma predeterminada éstos no producen la impresión de una línea de extensión.

Los guiones centrados se escriben como ' `--` ' entre sílabas de una misma palabra (fíjese en los espacios antes y después de los dos guiones). El guión quedará centrado entre las sílabas, y su longitud se ajustará en función del espacio que exista entre ellas.

En la música grabada de modo muy apretado, se pueden quitar los guiones. Se puede controlar si esto ocurre o no, mediante las propiedades `minimum-distance` (distancia mínima entre las dos sílabas) y la `minimum-length` (umbral por debajo del cual se suprimen los guiones) de `LyricHyphen`.

De manera predeterminada no se repiten los guiones después de un salto de sistema cuando la línea siguiente comienza con una sílaba distinta. Si se establece la propiedad `after-line-breaking` al valor `#t`, se pueden imprimir los guiones en estas situaciones.

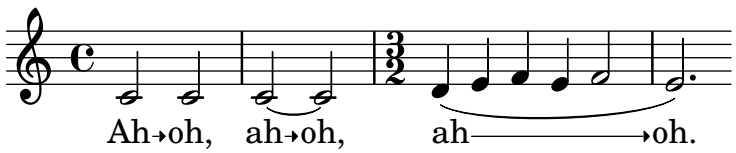
Véase también

Referencia de funcionamiento interno: Sección “LyricExtender” en *Referencia de Funcionamiento Interno*, Sección “LyricHyphen” en *Referencia de Funcionamiento Interno*.

9.1.9 Cambios de vocal graduales

Las transiciones de vocal (cambios graduales de una vocal o de una consonante mantenida), que pueden venir indicadas por flechas entre dos sílabas, se introducen con la instrucción `\vowelTransition` (véase Gould págs. 452–453). La flecha muestra la longitud de la transición, y por defecto siempre se imprime (se añade espacio en caso necesario si la música está muy apretada). Se indica que no hay ninguna articulación, a pesar del cambio de vocal, mediante ligaduras de unión o de expresión. La longitud mínima de las flechas puede ajustarse con la propiedad `minimum-length` del objeto `VowelTransition`.

```
{
  c'2 c'
  \set melismaBusyProperties = #'()
  c'2 ~ c'
  \time 3/2
  d'4( e' f' e' f'2
  e'2.)
}
\addlyrics
{
  Ah \vowelTransition oh,
  ah \vowelTransition oh,
  ah \vowelTransition _ _ _ _
  oh.
}
```



Véase también

GLosario musical: Sección “vowel transition” en *Glosario Musical*.

Referencia de funcionamiento interno: Sección “VowelTransition” en *Referencia de Funcionamiento Interno*.

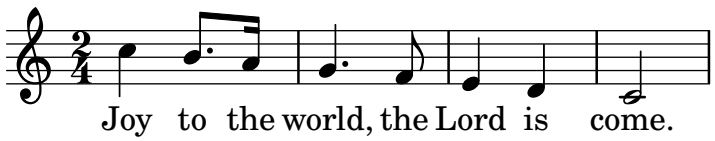
9.2 Técnicas específicas para la letra

9.2.1 Trabajar con letra y variables

Se pueden crear variables que contienen letra, pero la letra se debe introducir en el modo de letra:

```
musicOne = \relative {
  c''4 b8. a16 g4. f8 e4 d c2
}
verseOne = \lyricmode {
  Joy to the world, the Lord is come.
}
\score {
```

```
  <<
    \new Voice = "one" {
      \time 2/4
      \musicOne
    }
    \new Lyrics \lyricsto "one" {
      \verseOne
    }
  >>
}
```

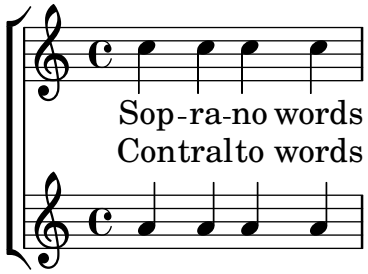


No es necesario escribir las duraciones si la variables se va a invocar con `\addlyrics` o con `\lyricsto`.

Para disposiciones distintas o más complejas, la mejor forma es definir en primer lugar las variables de la música y de la letra, después preparar una jerarquía de pentagramas y letras, omitiendo la letra en sí, y después poner la letra utilizando `\context` por debajo. Esto asegura que las voces referenciadas por `\lyricsto` siempre han sido definidas previamente. Por ejemplo:

```
sopranoMusic = \relative { c''4 c c c }
contraltoMusic = \relative { a'4 a a a }
sopranoWords = \lyricmode { Sop -- ra -- no words }
contraltoWords = \lyricmode { Con -- tral -- to words }

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \sopranoMusic
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos"
    \new Staff {
      \new Voice = "contraltos" {
        \contraltoMusic
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        \sopranoWords
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        \contraltoWords
      }
    }
  >>
```

```
}
```



Véase también

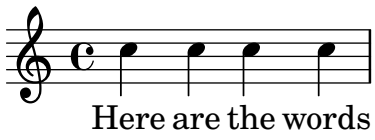
Referencia de la notación: Sección 9.2.2 [Posicionamiento vertical de la letra], página 364.

Referencia de funcionamiento interno: Sección “LyricCombineMusic” en *Referencia de Funcionamiento Interno*, Sección “Lyrics” en *Referencia de Funcionamiento Interno*.

9.2.2 Posicionamiento vertical de la letra

Dependiendo del tipo de música, la letra puede colocarse sobre el pentagrama, debajo del pentagrama, o entre dos pentagramas. Lo más fácil es situar la letra debajo de la pauta asociada, y se puede conseguir simplemente definiendo el contexto Lyrics por debajo del contexto Staff:

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative { c''4 c c c }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



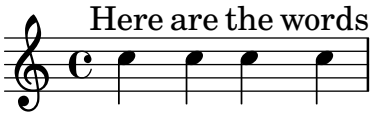
Se puede colocar la letra sobre un pentagrama usando uno de los dos métodos disponibles. El más sencillo (y preferible) es usar la misma sintaxis que se ha visto más arriba y especificar explícitamente la posición de la letra.

El argumento de `alignAboveContext` es el nombre de un contexto de `Staff` o uno de sus hermanos (no funciona si se usa `Voice`).

Use `alignBelowContext` para posicionar la letra debajo de la pauta. Normalmente esto solo es necesario para forzar la alineación entre la letra y una pauta que no es la predeterminada.

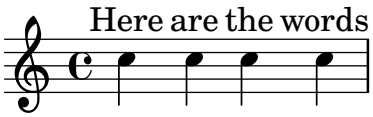
```
\score {
  <<
    \new Staff = "staff" {
      \new Voice = "melody" {
        \relative { c''4 c c c }
```

```
    }
  }
  \new Lyrics \with { alignAboveContext = "staff" } {
    \lyricsto "melody" {
      Here are the words
    }
  }
>>
}
```



Como alternativa, se puede usar un proceso en dos pasos. En primer lugar se declara el contexto Lyrics (sin ningún contenido) antes de los contextos Staff y Voice, y después se escribe la instrucción `\lyricsto` después de la declaración de Voice a que hace referencia, usando `\context`, como se ve a continuación:

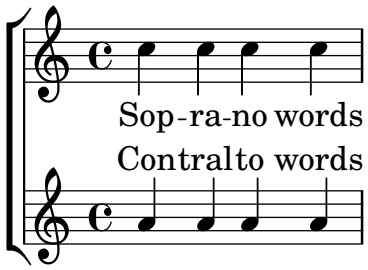
```
\score {
  <<
    \new Lyrics = "lyrics" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "melody" {
        \relative { c''4 c c c }
      }
    }
    \context Lyrics = "lyrics" {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Cuando hay dos voces sobre pentagramas separados, la letra se puede colocar entre ellos usando cualquiera de estos métodos. He aquí un ejemplo del segundo método:

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \relative { c''4 c c c }
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos" \with {
```

```
      % lyrics above a staff should have this override
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "contraltos" {
        \relative { a'4 a a a }
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        Sop -- ra -- no words
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        Con -- tral -- to words
      }
    }
  >>
}
```



Es posible generar otras combinaciones de letra y pautas mediante la elaboración de los ejemplos anteriores, o examinando las plantillas del Manual de Aprendizaje, véase Sección "Plantillas de conjuntos vocales" en *Manual de Aprendizaje*.

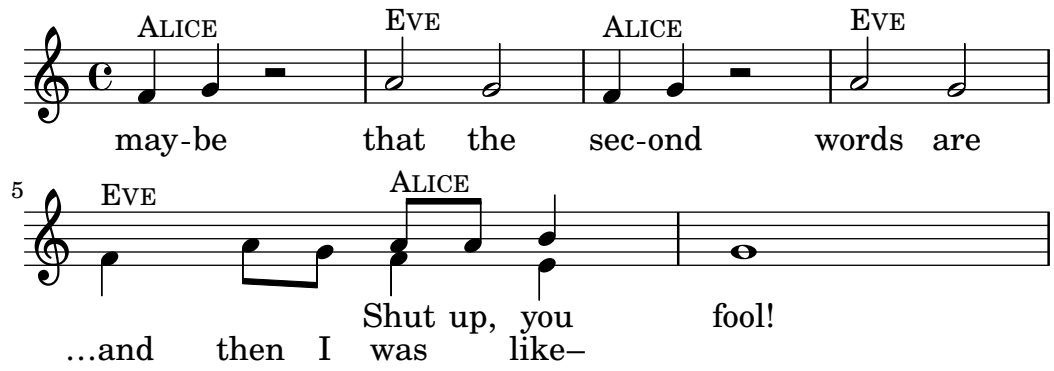
Fragmentos de código seleccionados

Disposición de distintas letras en la misma línea

A veces queremos poner la letra que corresponde a distintos personajes sobre una sola línea, por ejemplo allí donde el texto alterna rápidamente. Este fragmento muestra cómo puede hacerse ajustando la propiedad `nonstaff-nonstaff-spacing` del objeto gráfico `VerticalAxisGroup`.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
              .nonstaff-nonstaff-spacing
              .minimum-distance = ##f
  }
}

aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

<<
```

```
\new Staff <<
  \new Voice = "alice" {
    f'4^\aliceSings g' r2 |
    s1 |
    f'4^\aliceSings g' r2 |
    s1 | \break
    % ...

    \voiceOne
    s2 a'8^\aliceSings a' b'4 |
    \oneVoice
    g'1
  }
  \new Voice = "eve" {
    s1 |
    a'2^\eveSings g' |
    s1 |
    a'2^\eveSings g'
    % ...

    \voiceTwo
    f'4^\eveSings a'8 g' f'4 e' |
    \oneVoice
    s1
  }
>>

\new Lyrics \lyricsto "alice" {
  may -- be
  sec -- ond
  % ...
  Shut up, you fool!
}

\new Lyrics \lyricsto "eve" {
  that the
  words are
  % ...
  ...and then I was like--
}
>>
```



Véase también

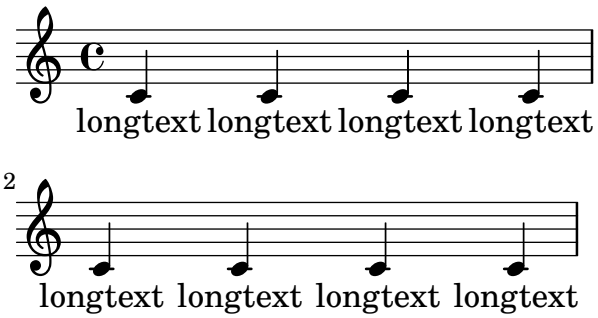
Manual de aprendizaje: Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 33.7 [Orden de disposición de los contextos], página 762, Sección 33.2 [Crear y referenciar contextos], página 746.

9.2.3 Colocación horizontal de las sílabas

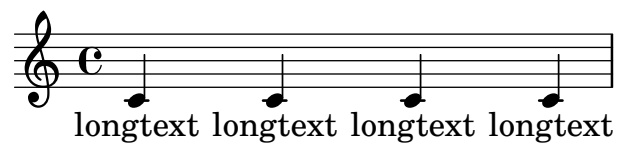
Para aumentar la separación entre las líneas de texto, establezca la propiedad `minimum-distance` de `LyricSpace`.

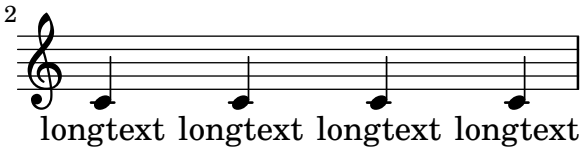
```
\relative c' {
  c c c c
  \override Lyrics.LyricSpace.minimum-distance = 1.0
  c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}
```



Para efectuar este cambio en todas las letras de la partitura, ajuste la propiedad dentro del bloque `\layout`.

```
\score {
  \relative {
  c' c c c
  c c c c
  }
  \addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
  }
  \layout {
    \context {
      \Lyrics
      \override LyricSpace.minimum-distance = 1.0
    }
  }
}
```



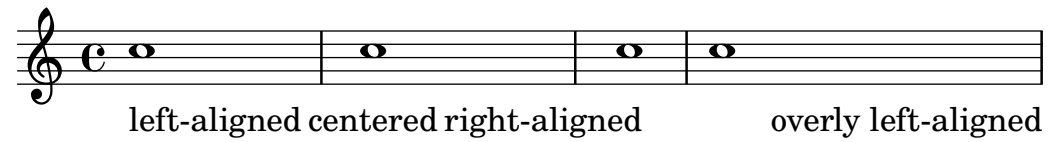


Fragmentos de código seleccionados

Alineación de la letra

La alineación horizontal de la letra se puede ajustar sobreescribiendo la propiedad `self-alignment-X` del objeto `LyricText`. El valor -1 significa alineado por la izquierda, 0 es centrado y 1 es derecha; de forma alternativa también podemos usar los valores de Scheme `#LEFT`, `#CENTER` y `#RIGHT` en lugar de números. También son posibles otros valores numéricos. Importante: no olvide añadir el prfijo de Scheme '#' para los números negativos.

```
\layout {
  ragged-right = ##f
}

\relative c'' {
  c1 c c c
}

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "overly left-aligned"
}
```

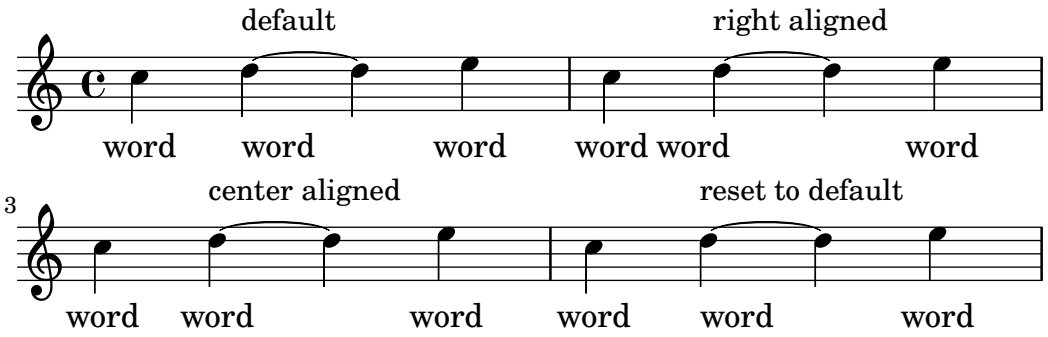


Alineación de sílabas con melisma

De forma predeterminada, las sílabas de la letra que comienzan un melisma se alinean a la izquierda sobre su nota correspondiente. Se puede alterar la alineación usando la propiedad `lyricMelismaAlignment`.

```
<<
  \new Staff {
    \new Voice = "vocal" \relative c'' {
      \override TextScript.staff-padding = #2
      c d~^\markup default  d e
      c d~^\markup "right aligned" d e
      c d~^\markup "center aligned" d e
      c d~^\markup "reset to default" d e
    }
  }
```

```
  \new Lyrics \lyricsto "vocal" {
    word word word
    \set lyricMelismaAlignment = #RIGHT
    word word word
    \set lyricMelismaAlignment = #CENTER
    word word word
    \unset lyricMelismaAlignment
    word word word
  }
>>

\layout {
  ragged-right = ##f
}
```



Advertencias y problemas conocidos

La comprobación necesaria para asegurar que las inscripciones de texto están dentro de los márgenes, requiere cálculos adicionales. Para acelerar ligeramente el procesado, se puede desactivar esta funcionalidad:

```
\override Score.PaperColumn.keep-inside-line = ##f
```

Para hacer que el texto evite también las líneas divisorias, use

```
\layout {
  \context {
    \Lyrics
      \consists Bar_engraver
      \consists Separating_line_group_engraver
      \hide BarLine
  }
}
```

9.2.4 Letra y repeticiones

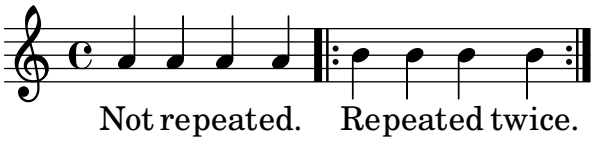
Repeticiones sencillas

Las repeticiones *en la música* se describen extensamente en alguna otra parte de este manual; véase Capítulo 4 [Repeticiones], página 190. Esta sección explica cómo añadir letra a secciones de música que se repiten.

La letra que va con una sección de música que se repite, debiera estar encerrada en la misma construcción de repeticiones exacta que la música, si el texto no cambia.

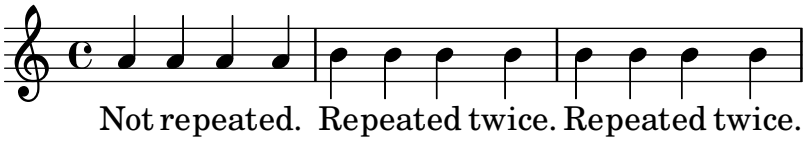
```
\score {
  <<
    \new Staff {
```

```
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- ed twice. }
      }
    }
  >>
}
```



El texto se expandirá correctamente de esta manera si las repeticiones se despliegan.

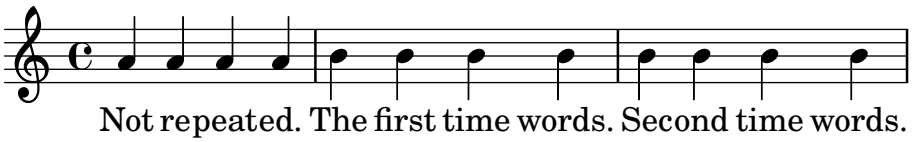
```
\score {
  \unfoldRepeats {
    <<
      \new Staff {
        \new Voice = "melody" {
          \relative {
            a'4 a a a
            \repeat volta 2 { b4 b b b }
          }
        }
      }
      \new Lyrics {
        \lyricsto "melody" {
          Not re -- peat -- ed.
          \repeat volta 2 { Re -- peat -- ed twice. }
        }
      }
    >>
  }
}
```



Si la sección repetida ha de desplegarse y tiene distinta letra, escriba simplemente todo el texto:

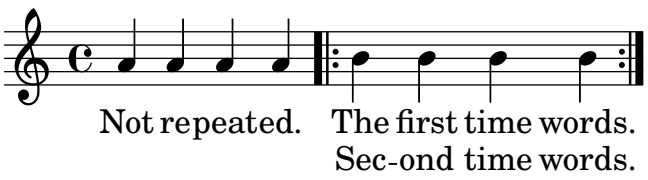
```
\score {
  <<
    \new Staff {
```

```
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat unfold 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        The first time words.
        Sec -- ond time words.
      }
    }
  >>
}
```



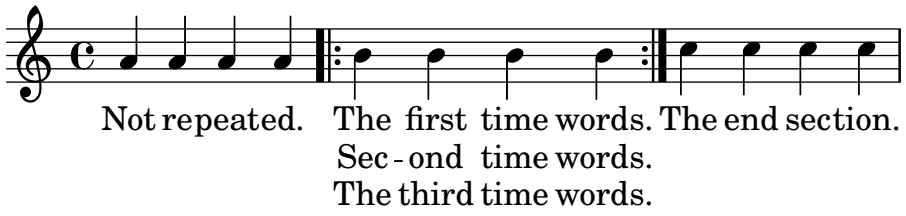
Cuando el texto de las secciones de primera o segunda vez es distinto, la letra de cada repetición se debe escribir dentro de contextos `Lyrics` diferentes, anidados correctamente en secciones paralelas:

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics \lyricsto "melody" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
          \set associatedVoice = "melody"
          Sec -- ond time words.
        }
      >>
    }
  >>
}
```



Se pueden añadir más estrofas de una manera similar:

```
\score {
  <<
    \new Staff {
      \new Voice = "singleVoice" {
        \relative {
          a'4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics \lyricsto "singleVoice" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
          \set associatedVoice = "singleVoice"
          Sec -- ond time words.
        }
        \new Lyrics {
          \set associatedVoice = "singleVoice"
          The third time words.
        }
      >>
      The end sec -- tion.
    }
  >>
}
```

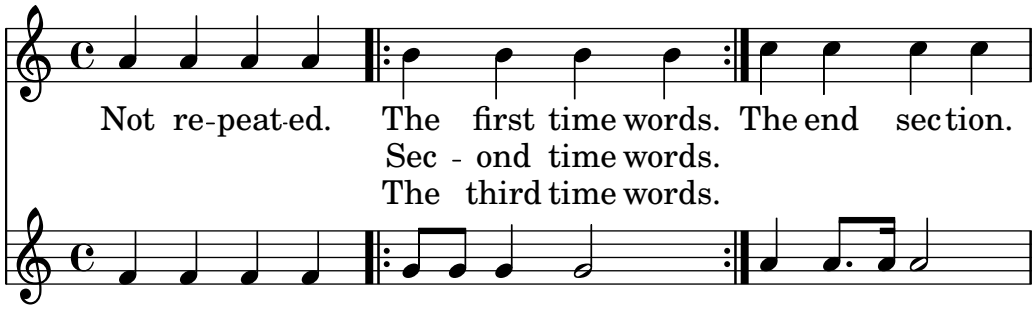


Sin embargo, si esta construcción se inserta dentro de un contexto de varios pentagramas tal como un `ChoirStaff`, la letra de los versos segundo y tercero aparecerán debajo del pentagrama inferior.

Para posicionarlos correctamente utilice `alignBelowContext`:

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
```

```
    \new Lyrics = "firstVerse" \lyricsto "melody" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics = "secondVerse"
        \with { alignBelowContext = "firstVerse" } {
          \set associatedVoice = "melody"
          Sec -- ond time words.
        }
        \new Lyrics = "thirdVerse"
        \with { alignBelowContext = "secondVerse" } {
          \set associatedVoice = "melody"
          The third time words.
        }
      >>
      The end sec -- tion.
    }
    \new Voice = "harmony" {
      \relative {
        f'4 f f f
        \repeat volta 3 { g8 g g4 g2 }
        a4 a8. a16 a2
      }
    }
  >>
}
```

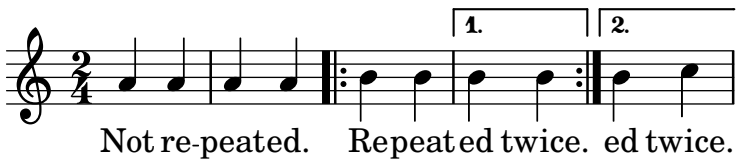


Repeticiones con finales alternativos (primera y segunda vez)

Si el texto de la sección repetida es el mismo, y ninguno de los bloques `\alternative` comienza con un silencio, se puede usar exactamente la misma estructura tanto para la letra como para la música. Esto tiene la ventaja de que `\unfoldRepeats` expande correctamente tanto la música como la letra.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b }
          \alternative {
            \volta 1 { b b }
```

```
            \volta 2 { b c }
          }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- }
        \alternative {
          \volta 1 { ed twice. }
          \volta 2 { ed twice. }
        }
      }
    }
  >>
}
```



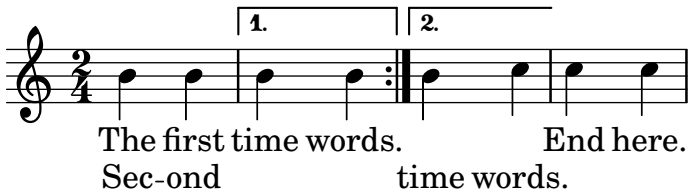
Pero cuando la sección repetida tiene un texto distinto, o cuando uno de los bloques `\alternative` comienza con un silencio, no se puede usar una construcción de repetición para el texto y es necesario insertar manualmente instrucciones `\skip` para saltar sobre las notas en las secciones alternativas que no correspondan.

Nota: no utilice un guión bajo, _, para saltar notas: el guión bajo indica un melisma, y hace que la sílaba anterior resulte alineada por la izquierda.

> **Nota:** La instrucción `\skip` debe ir seguida por un número, pero este número se ignora dentro de la letra, que deriva las duraciones de las notas de una melodía asociada a través de `\addlyrics` o `\lyricsto`. Cada `\skip` salta una sola nota de cualquier valor, independientemente del valor del número que se escribe a continuación.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b }
          \alternative {
            \volta 1 { b b }
            \volta 2 { b c }
          }
          c4 c
        }
      }
    }
```

```
    \new Lyrics {
      \lyricsto "melody" {
        The first time words.
        \repeat unfold 2 { \skip 1 }
        End here.
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Sec -- ond
        \repeat unfold 2 { \skip 1 }
        time words.
      }
    }
  >>
}
```

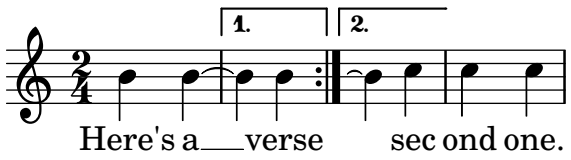


Cuando una nota está ligada sobre dos o más finales alternativos (sobre la primera nota de una casilla de primera y segunda vez), se usa una ligadura de unión para arrastrar la nota dentro del primer final alternativo y se usa una instrucción `\repeatTie` de ligadura de repetición en las casillas de repetición siguientes. Esta estructura produce problemas de alineación difíciles cuando hay letra de por medio; aumentar la longitud de las secciones alternativas de forma que las notas ligadas se vean contenidas completamente dentro de ellas, puede ofrecer un resultado más aceptable.

La ligadura de unión crea un melisma dentro de la casilla de primera vez pero no en la segunda vez y siguientes; por tanto, para alinear la letra correctamente es necesario inhabilitar la creación automática de melismas sobre la sección de finales alternativos e insertar saltos manuales.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \set melismaBusyProperties = #'()
          \repeat volta 2 { b'4 b ~}
          \alternative {
            \volta 1 { b b }
            \volta 2 { b \repeatTie c }
          }
          \unset melismaBusyProperties
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
```

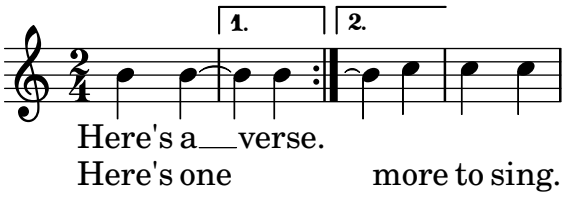
```
        \repeat volta 2 { Here's a __ }
        \alternative {
          \volta 1 { \skip 1 verse }
          \volta 2 { \skip 1 sec }
        }
        ond one.
      }
    }
  >>
}
```



Observe que si se usa \unfoldRepeats sobre una sección que contiene alguna \repeatTie, la \repeatTie se debe eliminar para evitar que se impriman al mismo tiempo los dos tipos de ligadura.

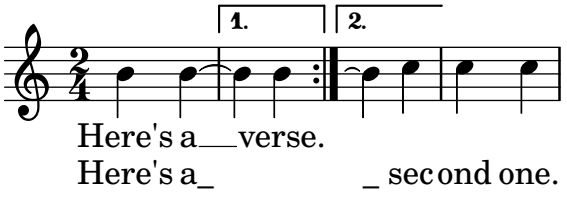
Cuando la sección repetida tiene textos distintos, no se puede usar \repeat para la letra y se deben insertar instrucciones \skip manualmente, como se vio antes.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b ~}
          \alternative {
            \volta 1 { b b }
            \volta 2 { b \repeatTie c }
          }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's one
        \repeat unfold 2 { \skip 1 }
        more to sing.
      }
    }
  >>
}
```



Si queremos imprimir líneas extensoras y guiones que entran o salen de las casillas de primera y segunda vez, hemos de insertarlos manualmente.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b ~}
          \alternative {
            \volta 1 { b b }
            \volta 2 { b \repeatTie c }
          }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's "a_"
        \skip 1
        "_" sec -- ond one.
      }
    }
  >>
}
```



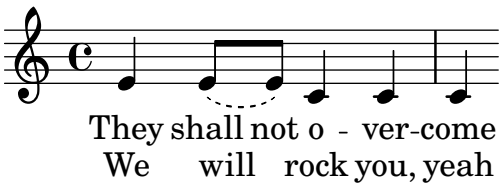
Véase también

Referencia de la notación: Sección 33.3 [Mantener vivos los contextos], página 749, Capítulo 4 [Repeticiones], página 190.

9.2.5 Letras en divisi

Cuando solamente difieren las palabras y las duraciones de las dos partes mientras que la altura de las notas es la misma, el método adecuado puede ser la desactivación temporal de la detección automática de los melismas e indicar el melisma dentro de la letra:

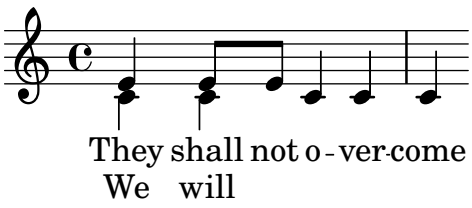
```
\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        \set melismaBusyProperties = #'()
        \slurDashed
        e4 e8( e) c4 c |
        \unset melismaBusyProperties
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "melody" {
      We will _ rock you, yeah
    }
  >>
}
```



Cuando difieren tanto la música como las palabras, puede ser mejor imprimir la música y la letra que difieren mediante el nombrado de contextos de voz y adjuntando la letra a estos contextos específicos:

```
\score {
  <<
    \new Voice = "melody" {
      \relative {
        <<
          {
            \voiceOne
            e'4 e8 e
          }
          \new Voice = "splitpart" {
            \voiceTwo
            c4 c
          }
        >>
        \oneVoice
        c4 c |
        c
      }
```

```
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "splitpart" {
      We will
    }
  >>
}
```

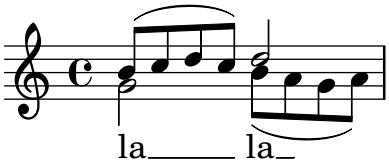


En la música coral es frecuente que una parte vocal se divida en dos durante varios compases. La construcción << {...} \\ {...} >>, en la que dos (o más) expresiones musicales están separadas por dos barras invertidas, puede en principio parecer la forma más adecuada de realizar las voces divididas. Sin embargo, esta construcción asigna **todas** las expresiones que están dentro de ella a **contextos de voz NUEVOS** lo que hace que no se asigne a ellas *ninguna letra* porque la letra se asigna al contexto de voz original, que no es lo que pretendemos, en general. La construcción correcta que debemos usar es el pasaje polifónico temporal, véase la sección *Pasajes polifónicos temporales* dentro de Sección 5.2.1 [Polifonía en un solo pentagrama], página 224.

9.2.6 Polifonía con letras compartidas

Cuando dos voces cuyas notas tienen distintas duraciones comparten la misma letra, la alineación de ésta a una de las voces puede dar lugar a problemas en la otra voz. Por ejemplo, la segunda línea extensora que se ve debajo es demasiado corta, porque la letra está alineada solamente a la voz superior:

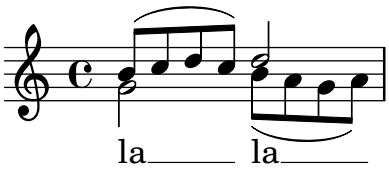
```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice = "sopranoVoice" { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
  \new Lyrics \lyricsto "sopranoVoice" \words
>>
```



Para obtener el resultado deseado, alinee la letra con un contexto `NullVoice` ('Voz Nula') nuevo que contenga una combinación adecuada de las dos voces. Las notas del contexto `NullVoice` no aparecen en la página impresa, pero pueden utilizarse para alinear la letra adecuadamente:

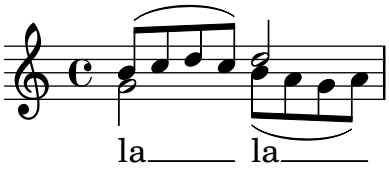
```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
```

```
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>
```



Este método se puede usar trambién con la función \partCombine, que en sí no permite incluir letra:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice \partCombine \soprano \alto
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>
```



Advertencias y problemas conocidos

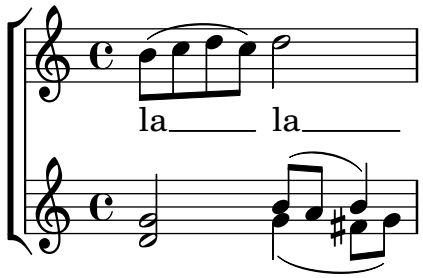
La función \addlyrics solo funciona con letras de Voice, por lo que no puede utilizarse con NullVoice.

La función \partCombine se describe en Sección 5.2.5 [Combinación automática de las partes], página 235.

Por último, este método se puede usar incluso si las voces están en distintos pentagramas, y no se limita a dos voces únicamente:

```
soprano = \relative { b'8( c d c) d2 }
altoOne = \relative { g'2 b8( a b4) }
altoTwo = \relative { d'2 g4( fis8 g) }
aligner = \relative { b'8( c d c) d( d d d) }
words = \lyricmode { la __ la __ }

\new ChoirStaff \with {\accepts NullVoice } <<
  \new Staff \soprano
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
  \new Staff \partCombine \altoOne \altoTwo
```

```
>>
```

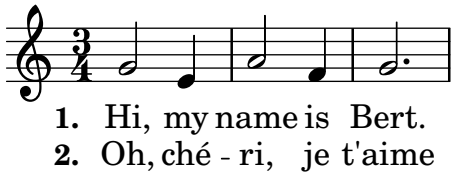


9.3 Versos

9.3.1 Añadir números de verso

Los números de los versos se pueden añadir estableciendo stanza, p.ej.,

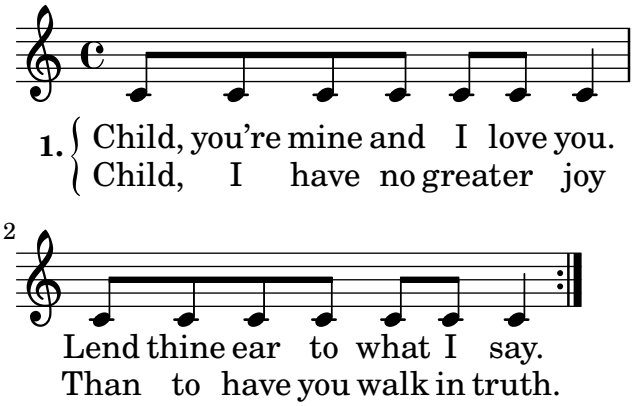
```
\new Voice \relative {
  \time 3/4 g'2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = "1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = "2. "
  Oh, ché -- ri, je t'aime
}
```



Estos números aparecerán inmediatamente antes de la primera sílaba. También pueden agruparse dos líneas de estrofa, por ejemplo en el caso de una repetición con letras diferentes:

```
stanzaOneOne = \lyricmode {
  \set stanza = \markup {
    \column {
      \vspace #.2
      \line { "1." \left-brace #30 }
    }
  }
  Child, you're mine and I love you.
  Lend thine ear to what I say.
}

stanzaOneThree = \lyricmode {
  Child, I have no great -- er joy
  Than to have you walk in truth.
}

\new Voice {
  \repeat volta 2 {
    c'8 c' c' c' c' c' c'4
```

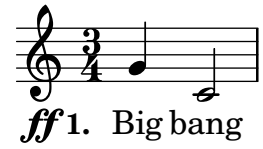
```
    c'8 c' c' c' c' c' c'4
  }
}
\addlyrics { \stanzaOneOne }
\addlyrics { \stanzaOneThree }
```



9.3.2 Añadir expresiones dinámicas a los versos

Los versos que difieren en su sonoridad se pueden especificar escribiendo una indicación dinámica antes de cada verso. En LilyPond, todo lo que aparece delante de un verso está dentro del objeto `StanzaNumber`; las indicaciones dinámicas no son diferentes. Por razones técnicas, tendrá que establecer el valor de la sección stanza (verso) fuera de `\lyricmode`:

```
text = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}

<<
  \new Voice = "tune" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "tune" \text
>>
```

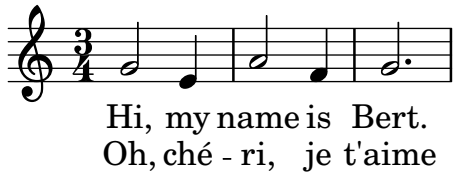


9.3.3 Añadir el nombre de los cantantes a los versos

También se pueden poner los nombres de los cantantes. Se imprimen al comienzo de la línea, igual que los nombres de instrumento. Se crean estableciendo un valor para `vocalName`. Se puede definir una versión abreviada como `shortVocalName`.

```
\new Voice \relative {
  \time 3/4 g'2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = "Bert "
```

```
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = "Ernie "
  Oh, ché -- ri, je t'aime
}
```



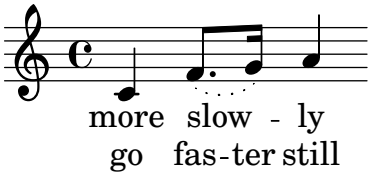
9.3.4 Versos con ritmos distintos

Con frecuencia, los distintos versos de una canción encajan de formas ligeramente diferentes con la misma melodía. Estas variaciones se pueden capturar con el uso de `\lyricsto`.

Ignorar los melismas

Existe la posibilidad de que el texto tenga un melisma en un verso, pero varias sílabas en otro. Una solución es hacer que la voz más rápida ignore el melisma. Esto se consigue estableciendo `ignoreMelismata` en el contexto Lyrics.

```
<<
  \relative \new Voice = "lahlah" {
    \set Staff.autoBeaming = ##f
    c'4
    \slurDotted
    f8.[( g16])
    a4
  }
  \new Lyrics \lyricsto "lahlah" {
    more slow -- ly
  }
  \new Lyrics \lyricsto "lahlah" {
    go
    \set ignoreMelismata = ##t
    fas -- ter
    \unset ignoreMelismata
    still
  }
>>
```



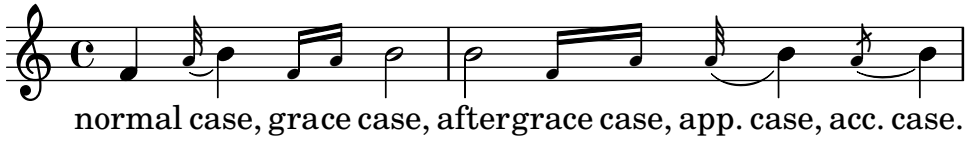
Advertencias y problemas conocidos

A diferencia de casi todas las instrucciones `\set`, `\set ignoreMelismata` no funciona si va precedido de `\once`. Es necesario utilizar `\set` y `\unset` para delimitar la letra en que se quieren ignorar los melismas.

Aplicar sílabas a notas de adorno

De forma predeterminada, las notas de adorno (p.ej. insertadas por medio de \grace) no pueden recibirla asignación de sílabas si se utiliza \lyricsto, pero este comportamiento puede cambiarse:

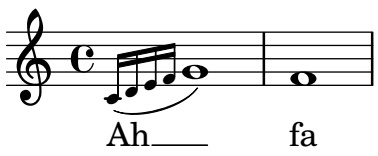
```
<<
  \new Voice = melody \relative {
    f'4 \appoggiatura a32 b4
    \grace { f16 a16 } b2
    \afterGrace b2 { f16[ a16] }
    \appoggiatura a32 b4
    \acciaccatura a8 b4
  }
  \new Lyrics
  \lyricsto melody {
    normal
    \set includeGraceNotes = ##t
    case,
    gra -- ce case,
    after -- grace case,
    \set ignoreMelismata = ##t
    app. case,
    acc. case.
  }
>>
```



Advertencias y problemas conocidos

Como ocurre con associatedVoice, includeGraceNotes se tiene que establecer al menos una sílaba antes de la que se va a colocar bajo una nota de adorno. En caso de una nota de adorno al comienzo mismo de una pieza, se recomienda usar un bloque \with o \context dentro de \layout:

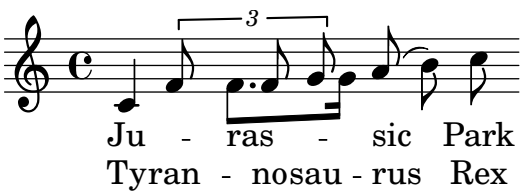
```
<<
  \new Voice = melody \relative c' {
    \grace { c16( d e f }
    g1) f
  }
  \new Lyrics \with { includeGraceNotes = ##t }
  \lyricsto melody {
    Ah __ fa
  }
>>
```



Cambio a una melodía alternativa

Son posibles variaciones más complejas en la coordinación de la letra y la música. La melodía a la que se está alineando la letra puede cambiarse desde dentro de la letra mediante el establecimiento de la propiedad `associatedVoice`:

```
<<
  \relative \new Voice = "lahlah" {
    \set Staff.autoBeaming = ##f
    c'4
    <<
      \new Voice = "alternative" {
        \voiceOne
        \tuplet 3/2 {
          % show associations clearly.
          \override NoteColumn.force-hshift = -3
          f8 f g
        }
      }
      {
        \voiceTwo
        f8.[ g16]
        \oneVoice
      } >>
    a8( b) c
  }
  \new Lyrics \lyricsto "lahlah" {
    Ju -- ras -- sic Park
  }
  \new Lyrics \lyricsto "lahlah" {
    % Tricky: need to set associatedVoice
    % one syllable too soon!
    \set associatedVoice = "alternative" % applies to "ran"
    Ty --
    ran --
    no --
    \set associatedVoice = "lahlah" % applies to "rus"
    sau -- rus Rex
  } >>
```



El texto del primer verso se fija a la melodía llamada ‘lahlah’ de la forma usual, pero el segundo verso se fija inicialmente al contexto `lahlah` y después se cambia a la melodía `alternative` para las sílabas desde ‘ran’ hasta ‘sau’ mediante las líneas:

```
\set associatedVoice = "alternative" % se aplica a "ran"
Ty --
ran --
no --
```

```
\set associatedVoice = "lahlah" % se aplica a "rus"
sau -- rus Rex
```

Aquí, `alternative` es el nombre del contexto de `Voice` que contiene el tresillo.

Observe la posición de la instrucción `\set associatedVoice`: parece estar situada una sílaba antes de lo normal, pero está bien así.

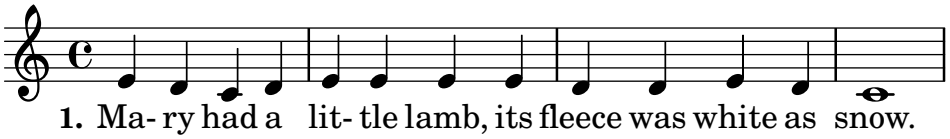
> **Nota:** La instrucción `\set associatedVoice` se debe escribir una sílaba *antes* de aquella en la que queremos que se produzca el cambio a la voz nueva. Dicho de otra forma, el cambio de la voz asociada se produce una sílaba más tarde de lo esperado. Esto es por razones técnicas, y no es un fallo.

9.3.5 Imprimir los versos al final

En ocasiones es conveniente tener un verso ajustado a la música, y el resto añadido en forma de estrofa al final de la pieza. Esto se puede conseguir escribiendo los versos adicionales dentro de una sección `\markup` fuera del bloque Score principal de la partitura. Tenga en cuenta que existen varias maneras diferentes de forzar los saltos de línea al usar `\markup`. Para introducir una cadena completa podemos usar `\string-lines` con saltos de línea `\n` insertados manualmente o saltos automáticos como los que aparecen al usar `\wordwrap-string`. Si se usa este código de formateo interior, se recomiendo una combinación de líneas y columnas introducidas con `\line` y `\column`.

```
melody = \relative {
  e' d c d | e e e e |
  d d e d | c1 | }

text = \lyricmode {
  \set stanza = "1."
  Ma- ry had a lit- tle lamb,
  its fleece was white as snow. }

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
}

\markup \column \string-lines
  "Verse 2. \n Everywhere that Mary went \n The lamb was sure to go."

\markup \column \string-lines
  "Verse 3.
   All the children laughed and played,
   To see a lamb at school."

\markup \column {
  \line \italic { Verse 4. }
  \line { And so the teacher turned it out, }
  \line { But still it lingered near. } }
```

```
\markup \wordwrap-string "
  Verse 5.

  Mary took it home again,

  It was against the rule."
```



Verse 2.
Everywhere that Mary went
The lamb was sure to go.

Verse 3.
All the children laughed and played,
To see a lamb at school.

Verse 4.
And so the teacher turned it out,
But still it lingered near.

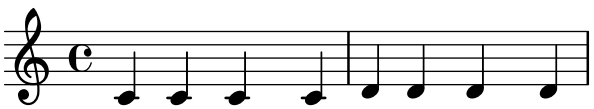
Verse 5.
Mary took it home again,
It was against the rule.

9.3.6 Imprimir los versos al final en varias columnas

Cuando una pieza tiene muchos versos, a menudo se imprimen en varias columnas a lo largo de toda la página. Con frecuencia un número de verso fuera del margen precede a cada verso. El ejemplo siguiente muestra cómo producir dicha salida en LilyPond.

```
melody = \relative {
  c'4 c c c | d d d d
}

text = \lyricmode {
  \set stanza = "1." This is verse one.
  It has two lines.
}

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}

\markup {
  \fill-line {
    % moves the column off the left margin;
    % can be removed if space on the page is tight
```

```
    \hspace #0.1
    \column {
      \line { \bold "2."
        \column {
          "This is verse two."
          "It has two lines."
        }
      }
      % adds vertical spacing between verses
      \combine \null \vspace #0.1
      \line { \bold "3."
        \column {
          "This is verse three."
          "It has two lines."
        }
      }
    }
    % adds horizontal spacing between columns
    \hspace #0.1
    \column {
      \line { \bold "4."
        \column {
          "This is verse four."
          "It has two lines."
        }
      }
      % adds vertical spacing between verses
      \combine \null \vspace #0.1
      \line { \bold "5."
        \column {
          "This is verse five."
          "It has two lines."
        }
      }
    }
    % gives some extra space on the right margin;
    % can be removed if page space is tight
    \hspace #0.1
  }
}
```



1. This is verse one. It has two lines.

2. This is verse two.
It has two lines.

3. This is verse three.
It has two lines.

4. This is verse four.
It has two lines.

5. This is verse five.
It has two lines.

Véase también

Referencia de funcionamiento interno: Sección “LyricText” en *Referencia de Funcionamiento Interno*, Sección “StanzaNumber” en *Referencia de Funcionamiento Interno*.

9.4 Canciones

9.4.1 Referencias para canciones

Por lo general, las canciones se escriben sobre tres pentagramas con la melodía del cantante en el pentagrama superior y dos pentagramas de acompañamiento de piano en la parte inferior. La letra del primer verso se imprime inmediatamente debajo del pentagrama superior. Si existe solamente una pequeña cantidad de versos adicionales, pueden imprimirse inmediatamente debajo del primero, pero si hay más versos de los que caben cómodamente allí, el segundo verso y siguientes se imprimen después de la música, como texto independiente.

Todos los elementos notacionales necesarios para escribir canciones se describen completamente en otras secciones de la documentación:

- Para montar la disposición de las pautas, véase Sección 6.1 [Impresión de los pentagramas], página 244.
- Para escribir música para piano, véase Capítulo 10 [Teclados y otros instrumentos de varios pentagramas], página 415.
- Para imprimir la letra de una línea melódica, véase Sección 9.1 [Notación común para música vocal], página 349.
- Para colocar la letra, véase Sección 9.2.2 [Posicionamiento vertical de la letra], página 364.
- Para escribir versos véase Sección 9.3 [Versos], página 382.
- Las canciones se imprimen frecuentemente con los acordes indicados mediante cifrado americano (los nombres de los acordes en letras mayúsculas) sobre las pautas. Esto se describe en Sección 15.2 [Imprimir los acordes], página 515.
- Para imprimir diagramas de posiciones de los acordes para el acompañamiento de guitarra o para otros instrumentos con trastes en el mástil, consulte “Marcados de diagramas de posiciones de acorde” dentro de Sección 12.1 [Notación común para cuerdas con trastes], página 433.

Véase también

Manual de aprendizaje: Sección “Canciones” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 9.1 [Notación común para música vocal], página 349, Sección 15.2 [Imprimir los acordes], página 515, Sección 6.1 [Impresión de los pentagramas], página 244, Capítulo 10 [Teclados y otros instrumentos de varios pentagramas], página 415, Sección 9.2.2 [Posicionamiento vertical de la letra], página 364, Sección 9.3 [Versos], página 382.

Fragmentos de código: Sección “Música vocal” en *Fragmentos de código*.

9.4.2 Hojas guía de acordes

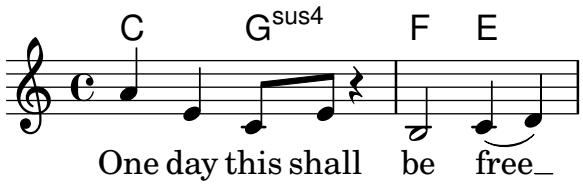
Se pueden imprimir hojas guía de acordes combinando partes vocales y el ‘modo de acordes’; esta sintaxis se explica en Capítulo 15 [Notación de acordes], página 510.

Fragmentos de código seleccionados

Hoja guía de acordes o lead sheet sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet».

```
<<
  \chords { c2 g:sus4 f e }
  \new Staff \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Véase también

Referencia de la notación: Capítulo 15 [Notación de acordes], página 510.

9.5 Música coral

Esta sección trata los asuntos de notación más directamente relacionados con la música coral. Esto incluye los anthems, las canciones por partes, los oratorios, etc.

9.5.1 Referencias para música coral

La notación de música coral se realiza normalmente sobre dos, tres o cuatro pentagramas dentro de un grupo de pautas `ChoirStaff`. Si se necesita un acompañamiento, se sitúa debajo dentro de un grupo `PianoStaff`, que se suele reducir de tamaño, para ensayar las obras corales *a cappella*. Las notas de cada parte vocal se sitúan dentro de un contexto `Voice`, y cara pentagrama recibe o una sola parte vocal (es decir, una `Voice`) o un par de partes vocales (es decir, dos `Voices`).

Los textos se disponen en contextos `Lyrics`, bien debajo de cada pauta de música correspondiente, o bien una encima y una debajo de la pauta de música, si ésta contiene la música de las dos partes.

En otras partes del manual se describen completamente algunos otros temas sobre música coral:

- Hay una introducción a la creación de partituras vocales SATB en el Manual de aprendizaje, véase Sección “Partitura vocal a cuatro voces SATB” en *Manual de Aprendizaje*. También existe una plantilla incorporada que simplifica la introducción de música vocal SATB, véase see Sección “Plantillas incorporadas” en *Manual de Aprendizaje*.
- También en el Manual de aprendizaje hay varias plantillas adecuadas para diversos estilos de música coral, véase Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.
- Para ver más información acerca de `ChoirStaff` y `PianoStaff`, consulte Sección 6.1.2 [Agrupar pentagramas], página 246.
- Las figuras con cabezas de formas, como las que se usan en la notación del estilo Arpa Sacra y otros similares, se describen en Sección 1.4.3 [Cabezas de notas con formas diversas], página 48.

- Cuando dos partes vocales comparten un pentagrama, las plicas, ligaduras, etc. de la parte aguda se orientan hacia arriba, y los de la parte grave hacia abajo. Para hacerlo, utilice `\voiceOne` y `\voiceTwo`. Véase Sección 5.2.1 [Polifonía en un solo pentagrama], página 224.
- Cuando una parte vocal se divide temporalmente, debemos usar *Pasajes polifónicos temporales* (véase Sección 5.2.1 [Polifonía en un solo pentagrama], página 224).

Instrucciones predefinidas

`\oneVoice`, `\voiceOne`, `\voiceTwo`.

Véase también

Manual de aprendizaje: Sección "Partitura vocal a cuatro voces SATB" en *Manual de Aprendizaje*, Sección "Plantillas de conjuntos vocales" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 33.7 [Orden de disposición de los contextos], página 762, Sección 6.1.2 [Agrupar pentagramas], página 246, Sección 1.4.3 [Cabezas de notas con formas diversas], página 48, Sección 5.2.1 [Polifonía en un solo pentagrama], página 224.

Fragmentos de código: Sección "Música vocal" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "ChoirStaff" en *Referencia de Funcionamiento Interno*, Sección "Lyrics" en *Referencia de Funcionamiento Interno*, Sección "PianoStaff" en *Referencia de Funcionamiento Interno*.

9.5.2 Disposiciones de la partitura para música coral

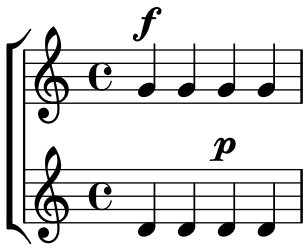
La música coral que contiene cuatro pentagramas, con o sin acompañamiento de piano, se suele disponer con dos sistemas por página. Dependiendo del tamaño de la página, conseguirlo puede requerir cambios en ciertos ajustes predeterminados. Se deben tener en cuenta los siguientes ajustes:

- Se puede modificar el tamaño global de pentagrama para alterar el tamaño general de los elementos de la partitura. Véase Sección 27.2 [Establecer el tamaño del pentagrama], página 688.
- Las diferentes distancias entre sistemas, pentagramas y líneas de texto se pueden ajustar de manera independiente. Véase Capítulo 29 [Espaciado vertical], página 701.
- Se pueden imprimir las dimensiones de las variables de disposición verticales como ayuda para el ajuste del espaciado vertical. Esta y otras posibilidades para hacer encajar la música en un número de páginas menor, se describen en Capítulo 31 [Encajar la música en menos páginas], página 734.
- Si el número de sistemas por página cambia de uno a dos, es costumbre indicarlo mediante una marca separadora de sistemas entre los dos sistemas. Véase Sección 6.1.4 [Separación de sistemas], página 251.
- Para ver más detalles sobre otras propiedades de formateo de las páginas, consulte Capítulo 26 [Disposición de la página], página 673.

Las indicaciones dinámicas se colocan por omisión debajo del pentagrama, pero en la música coral se suelen situar encima del pentagrama para evitar las colisiones con el texto. La instrucción predefinida `\dynamicUp` hace esta tarea para las indicaciones dinámicas en un único contexto `Voice`. Si hay muchos contextos de voz, esta instrucción predefinida tendría que colocarse en cada uno de ellos. Como alternativa, se puede usar su forma expandida para poner las indicaciones dinámicas de toda la partitura encima de sus pautas respectivas, como se muestra aquí:

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice {
```

```
        \relative { g'4\f g g g }
      }
    }
    \new Staff {
      \new Voice {
        \relative { d'4 d d\p d }
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override DynamicText.direction = #UP
      \override DynamicLineSpanner.direction = #UP
    }
  }
}
```



Instrucciones predefinidas

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

Véase también

Referencia de la notación: Sección 31.2 [Cambiar el espaciado], página 735, Sección 31.1 [Mostrar el espaciado], página 734, Capítulo 31 [Encajar la música en menos páginas], página 734, Capítulo 26 [Disposición de la página], página 673, Capítulo 27 [Disposición de la partitura], página 686, Sección 6.1.4 [Separación de sistemas], página 251, Sección 27.2 [Establecer el tamaño del pentagrama], página 688, Capítulo 28 [Saltos], página 692, Capítulo 29 [Espaciado vertical], página 701.

Referencia de funcionamiento interno: Sección "VerticalAxisGroup" en *Referencia de Funcionamiento Interno*, Sección "StaffGrouper" en *Referencia de Funcionamiento Interno*.

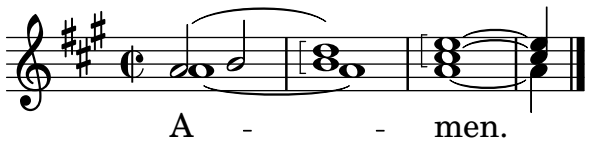
Fragmentos de código seleccionados

Using a bracket to clarify divisi

The `\nonArpeggiato` command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
```

```
        \new Voice = "upper" <<
          {
            \voiceOne
            a2( b2
            <b d>1\nonArpeggiato)
            <cs e>\nonArpeggiato ~
            <cs e>4
            \fine
          }
          \addlyrics { \lyricmode { A -- men. } }
        >>
        \new Voice = "lower" {
          \voiceTwo
          a1 ~
          a
          a ~
          a4
          \fine
        }
      >>
    }
}
```



Véase también

Referencia de la notación: Sección 3.3 [Expresiones como líneas], página 178.

9.6 Ópera y musicales

La música, letra y diálogos de las óperas y musicales se disponen normalmente de una o más de las siguientes formas:

- Una *Partitura del director* que contiene todas las partes orquestales y vocales, junto a notas guía del libreto si existen pasajes hablados.
- *Particellas orquestales* que contienen la música de los instrumentos individuales de la orquesta o banda.
- Una *Partitura vocal* que contiene todas las partes vocales con acompañamiento de piano. El acompañamiento es normalmente una reducción de orquesta, y en este caso es frecuente indicar el nombre del instrumento original de la orquesta. Las partituras vocales a veces incluyen indicaciones de escena y notas guía del libreto.
- Un *Libro vocal* que contiene sólo las partes vocales (sin acompañamiento), a veces en combinación con el libreto.
- Un *Libreto* que contiene los diálogos completos que normalmente hay en los musicales, junto a la letra de las partes cantadas. Suelen incluirse también las indicaciones de escena. Se puede utilizar LilyPond para tipografiar libretos, pero dado que no contienen música, puede ser preferible algún método alternativo.

En la sección de referencias que aparece a continuación están relacionadas las secciones de la documentación de LilyPond que se ocupan de los temas necesarios para crear partituras en los

estilos habituales de la ópera y los musicales. Después aparecen secciones que cubren las técnicas peculiares de la tipografía musical de las partituras de ópera y de musicales.

9.6.1 Referencias para ópera y musicales

La gran mayoría de las nociones que aparecen a continuación se pueden aplicar a prácticamente cualquier música orquestal y de conjunto, además de a la música vocal y de escena:

- Una partitura de director contiene muchos pentagramas y legras agrupados. En Sección 6.1.2 [Agrupar pentagramas], página 246, se explican maneras de agrupar pentagramas. Para anidar unos grupos dentro de otros, consulte Sección 6.1.3 [Grupos de pentagramas anidados], página 250.
- La impresión de pentagramas vacíos en las partituras del director y vocales se suele inhibir. Para crear estas partituras, conocidas como "Partituras francesas" consulte Sección 6.2.3 [Ocultar pentagramas], página 260.
- La escritura de partituras orquestales se estudia en Sección 6.3 [Escritura de las particellas], página 265. Pueden ser de relevancia otras secciones del capítulo de Notación especializada, dependiendo de la orquestación utilizada. Muchos instrumentos son transpositores, consulte Sección 1.3.4 [Transposición de los instrumentos], página 30.
- Si el número de sistemas por página varía de una a otra página, es costumbre separar los sistemas con una marca separadora de sistemas. Véase See Sección 6.1.4 [Separación de sistemas], página 251.
- Para ver detalles acerca de otras propiedades de formateo de las páginas, consulte Capítulo 26 [Disposición de la página], página 673.
- Se pueden insertar notas guía de diálogos, instrucciones de escena y notas al pie, véase Sección 21.4 [Crear notas al pie], página 612, y Capítulo 8 [Texto], página 311. También se pueden añadir indicaciones de escena extensas con una sección de elementos de marcado independientes entre dos bloques `\score`, véase Sección 8.1.6 [Texto separado], página 321.

Véase también

Glosario musical: Sección "Frenched score" en *Glosario Musical*, Sección "Frenched staves" en *Glosario Musical*, Sección "transposing instrument" en *Glosario Musical*.

Referencia de la notación: Sección 21.4 [Crear notas al pie], página 612, Sección 6.1.2 [Agrupar pentagramas], página 246, Sección 6.2.3 [Ocultar pentagramas], página 260, Sección 1.3.4 [Transposición de los instrumentos], página 30, Sección 6.1.3 [Grupos de pentagramas anidados], página 250, Capítulo 26 [Disposición de la página], página 673, Sección 6.1.4 [Separación de sistemas], página 251, Sección 1.2.2 [Transposición], página 14, Sección 6.3 [Escritura de las particellas], página 265, Sección 8.1 [Escritura del texto], página 311.

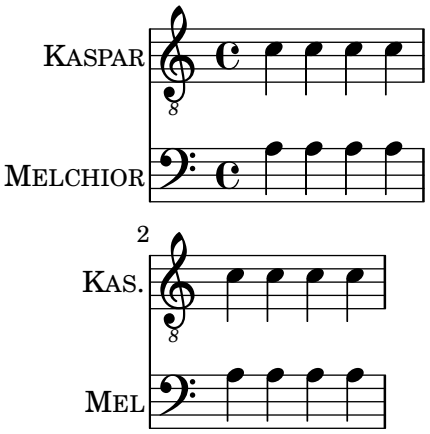
Snippets: Sección "Música vocal" en *Fragmentos de código*.

9.6.2 Nombres de los personajes

Se suelen mostrar los nombres de los personajes a la izquierda del pentagrama cuando éste está dedicado a dicho personaje solamente:

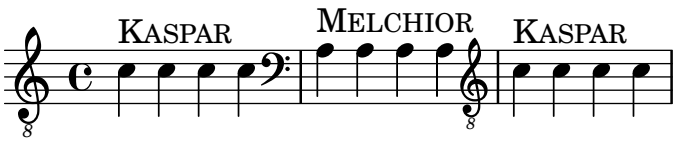
```
\score {
  <<
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Kaspar
      \set Staff.shortVocalName = \markup \smallCaps Kas.
      \relative {
        \clef "G_8"
        c'4 c c c
```

```
        \break
        c4 c c c
      }
    }
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Melchior
      \set Staff.shortVocalName = \markup \smallCaps Mel
      \clef "bass"
      \relative {
        a4 a a a
        a4 a a a
      }
    }
  >>
}
```



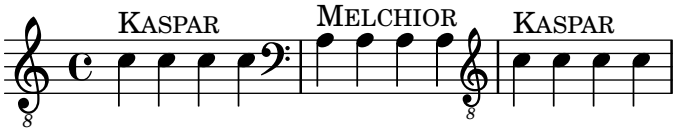
Cuando dos o más personajes comparten un pentagrama, el nombre del personaje se suele imprimir encima del pentagrama al principio de cada sección correspondiente a dicho personaje. Esto se puede hacer con elementos de marcado. Con frecuencia se usa un tipo de letra especial para este propósito.

```
\relative c' {
  \clef "G_8"
  c4^\markup \fontsize #1 \smallCaps Kaspar
  c c c
  \clef "bass"
  a4^\markup \fontsize #1 \smallCaps Melchior
  a a a
  \clef "G_8"
  c4^\markup \fontsize #1 \smallCaps Kaspar
  c c c
}
```



Como alternativa, si hay muchos cambios de personajes, puede ser más fácil establecer variables que contengan las definiciones de cada personaje, para así poder indicar el cambio de personaje de manera fácil y concisa.

```
kaspar = {
  \clef "G_8"
  \set Staff.shortVocalName = "Kas."
  \set Staff.midiInstrument = "voice oohs"
  <>^\markup \smallCaps "Kaspar"
}

melchior = {
  \clef "bass"
  \set Staff.shortVocalName = "Mel."
  \set Staff.midiInstrument = "choir aahs"
  <>^\markup \smallCaps "Melchior"
}

\relative c' {
  \kaspar
  c4 c c c
  \melchior
  a4 a a a
  \kaspar
  c4 c c c
}
```



Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 8 [Texto], página 311, Sección A.1 [Instrucciones de marcado de texto], página 815.

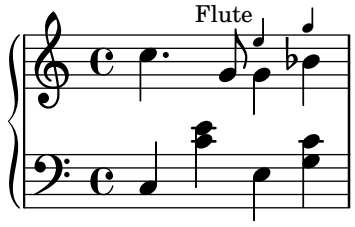
9.6.3 Guías musicales

Se pueden insertar guías musicales en las partituras vocales, en los libros vocales y en las partes orquestales para indicar qué música de otra parte precede inmediatamente a una entrada. Asimismo, las guías se suelen insertar en la reducción de piano en las partituras vocales para indicar lo que está tocando cada parte orquestas. Esto sirve de ayuda al director cuando no está disponible una partitura del director.

El mecanismo básico para insertar guías se explica de forma exhaustiva en el texto principal, véase Sección 6.3.2 [Citar otras voces], página 268, y Sección 6.3.3 [Formateo de las notas guía], página 272. Pero cuando se deben insertar muchas guías, por ejemplo, como ayuda para el director en una partitura vocal, el nombre del instrumento se debe colocar cuidadosamente justo antes del comienzo de las notas guía, y cerca de ellas. El siguiente ejemplo muestra cómo se hace esto.

```
flute = \relative {
  s4 s4 e'' g
}
\addQuote "flute" { \flute }

pianoRH = \relative {
```

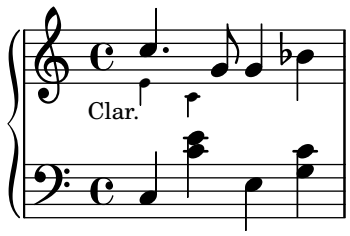
```
  c''4. g8
  % position name of cue-ing instrument just before the cue notes,
  % and above the staff
  <>^\markup { \right-align { \tiny "Flute" } }
  \cueDuring "flute" #UP { g4 bes4 }
}
pianoLH = \relative { c4 <c' e> e, <g c> }

\score {
  \new PianoStaff <<
    \new Staff {
      \pianoRH
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
}
```



Si un instrumento transpositor se está citando, la parte instrumental debe especificar la tonalidad de manera que la conversión de las notas citadas se realice automáticamente. El ejemplo de abajo muestra esta transposición para un clarinete en Si bemol. Las notas de este ejemplo son bajas en el pentagrama, por lo que se especifica DOWN en \cueDuring (para que las plicas se orienten hacia abajo) y el nombre del instrumento se coloca debajo del pentagrama.

```
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
  \transposition c'
  % position name of cue-ing instrument below the staff
  <>_\markup { \right-align { \tiny "Clar." } }
  \cueDuring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}
pianoLH = \relative { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
      \new Staff {
        \new Voice {
```

```
            \pianoRH
          }
        }
        \new Staff {
          \clef "bass"
          \pianoLH
        }
      >>
    >>
}
```



A partir de estos dos ejemplos debería quedar claro que insertar muchas notas guía en una partitura vocal puede ser tedioso, y la visión de las notas de la parte de piano quedaría entorpecida. Sin embargo, como muestra el siguiente fragmento de código, es posible definir una función musical para reducir la cantidad de texto que se teclea y para hacer que las notas del piano estén más claras.

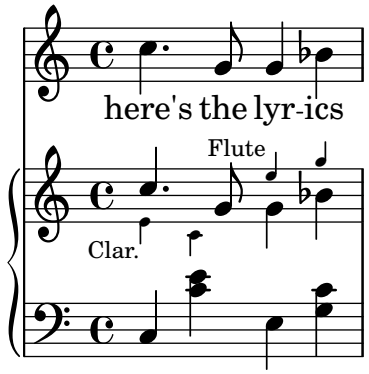
Fragmentos de código seleccionados

Añadir notas guía orquestales a una partitura vocal

Este ejemplo muestra una forma de simplificar la adición de muchas notas guía orquestales a la reducción de piano en una partitura vocal. La función musical `\cueWhile` toma cuatro argumentos: la música de la que se toma la cita, como viene definida por `\addQuote`, el nombre que insertar antes de las notas guía, y después `UP` o `DOWN` para especificar `\voiceOne` con el nombre encima del pentagrama o bien `\voiceTwo` con el nombre debajo del pentagrama, y finalmente la música de piano con la que las notas guía deben aparecer en paralelo. El nombre del instrumento citado se posiciona a la izquierda de las notas guía. Se pueden citar muchos pasajes como guía, pero no se pueden superponer en el tiempo entre ellos.

```
cueWhile =
#(define-music-function
   (instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
       \once \override TextScript.direction = $dir
       <>-\markup { \tiny #name }
       $music
     }
   #})

flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
```

```
\addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



Véase también

Glosario musical: Sección “cue-notes” en *Glosario Musical*.

Referencia de la notación: Sección 36.9 [Alineación de objetos], página 801, Sección 36.1 [Dirección y posición], página 783, Sección 6.3.3 [Formateo de las notas guía], página 272, Sección 6.3.2 [Citar otras voces], página 268, Sección 22.3 [Uso de las funciones musicales], página 641.

Fragmentos de código: Sección “Música vocal” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CueVoice” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

`\cueDuring` inserta automáticamente un contexto `CueVoice` y todas las notas guía se colocan dentro de este contexto. Ello significa que no es posible tener dos secuencias de notas guía superpuestas mediante esta técnica. Las secuencias superpuestas se podrían introducir declarando explícitamente contextos `CueVoice` separados y usando `\quoteDuring` para extraer e insertar las notas citadas.

9.6.4 Música hablada

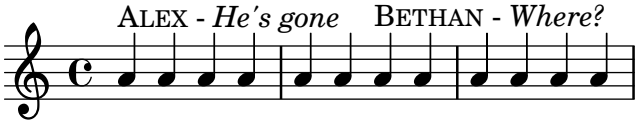
Efectos tales como el ‘parlato’ o el ‘Sprechgesang’ requieren que los intérpretes hablen sin altura definida pero con el ritmo adecuado; la notación de estos estilos se realiza mediante figuras con la cabeza en forma de cruz, como se muestra en Sección 1.4.1 [Cabezas de nota especiales], página 44.

9.6.5 Diálogos encima de la música

Los diálogos sobre la música se imprimen normalmente por encima de los pentagramas en un estilo inclinado, con el comienzo de cada frase ligado a un momento musical en particular.

Para interjecciones cortas, basta un simple elemento de marcado.

```
\relative {
  a'4^\markup { \smallCaps { Alex - } \italic { He's gone } } a a a
  a4 a a^\markup { \smallCaps { Bethan - } \italic Where? } a
  a4 a a a
}
```



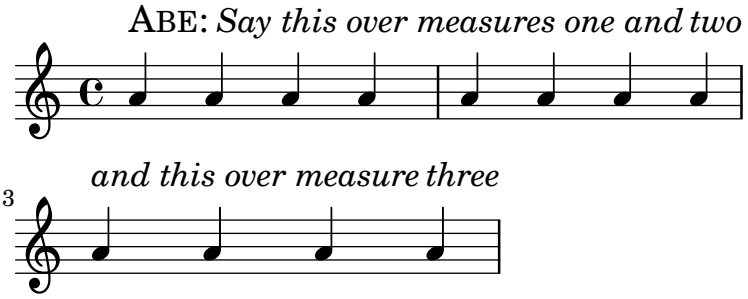
Para frases más largas puede ser necesario expandir la música de forma que que las palabras quepan holgadamente. No está previsto en LilyPond hacer esto de manera completamente automática, y puede ser necesaria cierta intervención manual sobre la disposición.

Para frases largas o pasajes con un diálogo en disposición apretada, puede dar mejores resultados la utilización de un contexto de letra. El contexto Lyrics no se debe asociar con una voz musical; en vez de ello, cada sección del diálogo recibe una duración explícita. Si hay una pausa en medio del diálogo, se debe separar la palabra final del resto y dividirse la duración entre ellas de forma que la música subyacente reciba un espaciado continuo y adecuado.

Si el diálogo se extiende a más de una línea, será necesario insertar \breaks manualmente y ajustar la colocación del diálogo para evitar que se extienda más allá del margen derecho. La última palabra del último compás de una línea debe separarse también, como se ha visto antes.

He aquí un ejemplo que ilustra cómo puede hacerse.

```
music = \relative {
  \repeat unfold 3 { a'4 a a a }
}

dialogue = \lyricmode {
  \markup {
    \fontsize #1 \upright \smallCaps Abe:
    "Say this over measures one and"
  }4*7
  "two"4 |
  \break
  "and this over measure"4*3
  "three"4 |
}

\score {
  <<
    \new Lyrics \with {
      \override LyricText.font-shape = #'italic
      \override LyricText.self-alignment-X = #LEFT
    }
    { \dialogue }
    \new Staff {
      \new Voice { \music }
    }
  >>
}
```



Véase también

Referencia de la notación: Sección 9.1.5 [Duración manual de las sílabas], página 356, Capítulo 8 [Texto], página 311.

Referencia de funcionamiento interno: Sección “LyricText” en *Referencia de Funcionamiento Interno*.

9.7 Cánticos salmos e himnos

La música y la letra de los cánticos, salmos e himnos suelen seguir un formato bien establecido dentro de cualquier iglesia en particular. Aunque los formatos pueden diferir de una iglesia a otra, los problemas de tipografiado que surgen son bastante similares, y se estudian en esta sección.

9.7.1 Referencias para cánticos y salmos

El tipografiado del canto gregoriano en distintos estilos de notación antigua se describe en Capítulo 17 [Notación antigua], página 540.

Véase también

Referencia de la notación: Capítulo 17 [Notación antigua], página 540.

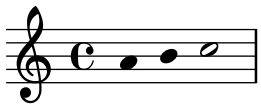
Snippets: Sección “Música vocal” en *Fragmentos de código*.

9.7.2 Preparar un cántico

La preparación moderna de los cánticos utiliza notación moderna con un número variable de elementos tomados de la notación antigua. Algunos de los elementos y métodos que considerar se presentan aquí.

Los cánticos usan con frecuencia figuras negras sin plica para indicar la altura, tomando las duraciones del ritmo hablado del texto.

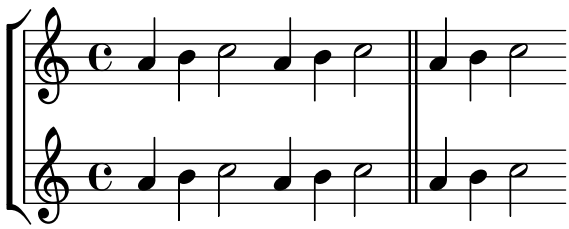
```
stemOff = { \hide Staff.Stem }

\relative c' {
  \stemOff
  a'4 b c2 |
}
```



Los cánticos con frecuencia omiten las líneas divisorias o utilizan barras de compás recortadas o discontinuas para indicar pausas en la música. Para preparar un cántico sin metro musical, véase Sección 2.3.4 [Música sin compasear], página 91. Para retener todos los efectos de una indicación de compás pero inhabilitando las barras de compás automáticas, fije `measureBarType` al valor `'()`. Si no hay líneas divisorias normales, puede que queramos tomar medidas para controlar el espaciado horizontal; véase Sección 28.1 [Saltos de línea], página 692.

```
\score {
  \new StaffGroup <<
    \new Staff {
      \relative {
        a'4 b c2 |
        a4 b c2 | \section
        a4 b c2 |
      }
    }
    \new Staff {
      \relative {
        a'4 b c2 |
        a4 b c2 | \section
        a4 b c2 |
      }
```

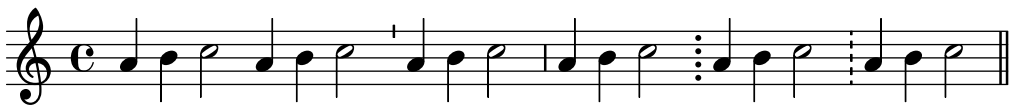
```
    }
  >>
  \layout {
    \context {
      \Staff
      measureBarType = #'()
      forbidBreakBetweenBarLines = ##f
    }
  }
}
```



Las líneas divisorias también se pueden modificar pentagrama a pentagrama, véase Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752,

Los silencios o pausas en los cánticos pueden indicarse mediante barras de compás modificadas.

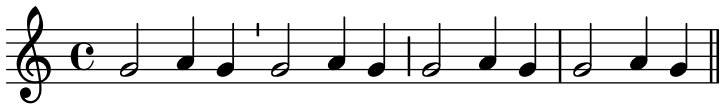
```
\relative a' {
  a4
  \cadenzaOn
  b c2
  a4 b c2
  \bar "'"
  a4 b c2
  \bar ","
  a4 b c2
  \bar ";"
  a4 b c2
  \bar "!"
  a4 b c2
  \bar "||"
}
```



Como alternativa, se usa a veces la notación del canto gregoriano para las pausas o silencios, aunque el resto de la notación sea moderna. Este ejemplo utiliza una marca `\breathe` modificada:

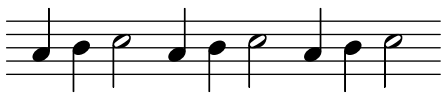
```
\score {
  \relative {
    g'2 a4 g
    \divisioMinima
    g2 a4 g
    \divisioMaior
    g2 a4 g
    \divisioMaxima
    g2 a4 g
```

```
    \finalis
  }
  \layout {
    \context {
      \Staff
      \remove Caesura_engraver
      \consists Divisio_engraver
      \EnableGregorianDivisiones
      caesuraType = #'((breath . chantquarterbar))
      measureBarType = #'()
      forbidBreakBetweenBarLines = ##f
    }
  }
}
```



Los cánticos suelen omitir la indicación de compás y con frecuencia omiten la clave también.

```
\score {
  \new Staff {
    \relative {
      a'4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  \layout {
    \context {
      \Staff
      \remove Time_signature_engraver
      \remove Clef_engraver
      measureBarType = #'()
      forbidBreakBetweenBarLines = ##f
    }
  }
}
```

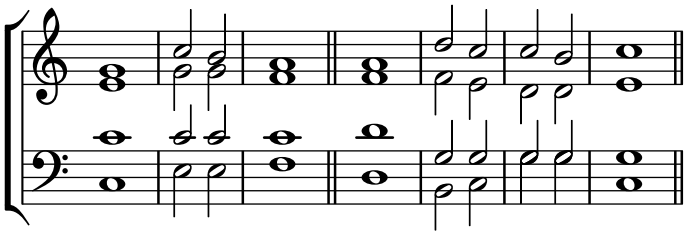


Los cánticos de salmos en la tradición anglicana suelen ser *sencillos*, con siete compases de música, o *dobles*, con dos períodos de siete compases. Cada grupo de siete compases se divide en dos mitades, que corresponden a las dos mitades de cada verso, normalmente separados mediante una doble línea divisoria. Sólo se utilizan redondas y blancas. El primer compás de cada mitad siempre contiene un único acorde en redondas. Ésta es la "nota de recitado". Los cánticos se suelen centrar sobre la página.

```
SopranoMusic = \relative {
  g'1 | c2 b | a1 |
  a1 | d2 c | c b | c1 |
}
```

```
AltoMusic = \relative {
  e'1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative {
  c'1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
  \skip 1*3 \section
  \skip 1*4 \fine
}

% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score {  % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
              \voiceOne
              \SopranoMusic
            >>
            \new Voice = "Alto" <<
              \voiceTwo
              \AltoMusic
            >>
          >>
          \new Staff <<
            \clef "bass"
            \global
            \new Voice = "Tenor" <<
              \voiceOne
              \TenorMusic
            >>
            \new Voice = "Bass" <<
              \voiceTwo
              \BassMusic
            >>
          >>
```

```
        >>
      >>
      \layout {
        \context {
          \Score
          \override SpacingSpanner.base-shortest-duration =
            \musicLength 2
          fineBarType = "||"
        }
        \context {
          \Staff
          \remove Time_signature_engraver
        }
      }
    }  % End score
  }
}  % End markup
```



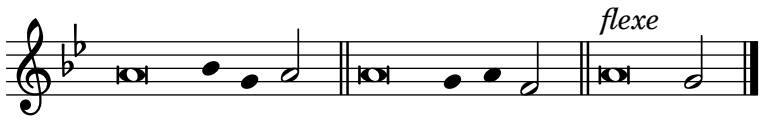
Otros enfoques para la preparación de este cántico se muestran en el primero de los siguientes fragmentos de código.

Fragmentos de código seleccionados

Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```

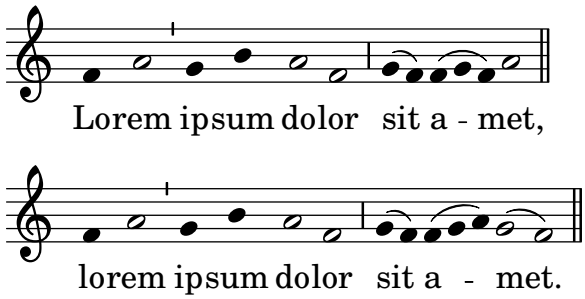


Los cánticos y otros textos litúrgicos pueden disponerse de una forma más libre, y posiblemente usen elementos notacionales tomados de la música antigua. Con frecuencia la letra se imprime por debajo y alineada con las notas. En este caso, las notas se disponen espaciadas de acuerdo con las sílabas en vez de hacerlo con las duraciones de las notas.

Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}
```



Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*, Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 17 [Notación antigua], página 540, Sección 2.5.1 [Barras de compás], página 120, Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752, Sección 17.4 [Tipografiado del canto gregoriano], página 552, Sección 2.3.4 [Música sin compasear], página 91, Sección 36.7 [Visibilidad de los objetos], página 794.

9.7.3 Puntuación de un salmo

El texto de un salmo anglicano se suele imprimir en versos separados debajo del canto.

Los cantos sencillos (con siete compases) se repiten para cada verso. Los cantos dobles (con 14 compases) se repiten para cada par de versos. Se insertan marcas dentro del texto para indicar la forma en que encajan con el canto. Cada verso se divide en dos mitades. Se suele utilizar un signo de dos puntos para indicar esta división. Esto corresponde a la doble línea divisoria de la música. El texto antes de los dos puntos se canta con los tres primeros compases de la música; el texto después de los dos puntos se canta con los últimos cuatro compases.

Se insertan líneas de compás simples (o en algunos casos una coma invertida u otro símbolo similar) entre el texto para indicar dónde caen las líneas divisorias de la música. En el modo de marcado, se puede introducir una línea de compás simple con el símbolo de comprobación de compás, la barra vertical, `|`.

```
\markup {
  \fill-line {
    \column {
      \line { O come let us sing | unto the | Lord : let }
      \line { us heartily rejoice \bold \italic in the | strength of | our }
      \line { sal- | -vation. }
    }
  }
}
```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

Otros símbolos pueden requerir glifos de las fuentes tipográficas `fetaMusic`. Para ver más detalles, consulte Sección 8.3 [Fuentes tipográficas], página 341.

```
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \line { O come let us sing \tick unto the \tick Lord : let }
      \line { us heartily rejoice \bold \italic in the \tick strength of \tick our }
      \line { sal \tick vation. }
    }
  }
}
```

O come let us sing ' unto the ' Lord : let
us heartily rejoice in the ' strength of ' our
sal ' vation.

Donde hay una redonda en un compás, todo el texto que corresponde a ese compás se recita sobre esa nota en el ritmo de la palabra. Donde hay dos notas en un compás, suele haber solamente una o dos sílabas correspondientes. Si hay más de dos sílabas, se suele insertar un punto para indicar dónde se produce el cambio de nota.

```
dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
```

```
}
\markup {
  \fill-line {
    \column {
      \line { O come let us sing \tick unto \dot the \tick Lord : let }
      \line { us heartily rejoice \bold \italic in the \tick strength of \tick our }
      \line { sal \tick vation. }
    }
  }
}
```

O come let us sing ' unto • the ' Lord : let
us heartily rejoice in the ' strength of ' our
sal ' vation.

En algunos salterios (libros de salmos) se usa un asterisco para indicar un corte en una sección recitada en lugar de una coma y las sílabas acentuadas o ligeramente alargadas se indican en estilo negrita.

```
dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \line { Today if ye will hear his voice * }
      \line { \concat { \bold hard en } |
              not your | hearts : as \bold \italic in the pro- }
      \line { vocation * and as \bold \italic in the \bold day of tempt- | }
      \line { -ation | \bold \italic in the | wilderness. }
    }
  }
}
```

Today if ye will hear his voice *
harden | not your | hearts : as in the pro-
vocation * and as in the **day** of tempt- |
-ation | in the | wilderness.

En otros salterios se coloca un acento sobre la sílaba para indicar el énfasis.

```
tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \line { O come let us \concat { si \combine \tick ng } |
              unto the | Lord : let }
      \line { us heartily \concat { rejo \combine \tick ice }
              \bold \italic in the | strength of | our }
```

```
      \line { sal- | -vation. }
    }
  }
}
```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

El uso del marcado para centrar texto y para disponer las líneas en columnas se describe en Sección 8.2 [Formatear el texto], página 323.

Casi todos estos elementos se muestran en uno u otro de los dos versos en la plantilla, véase Sección "Salmos" en *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Salmos" en *Manual de Aprendizaje*, Sección "Plantillas de conjuntos vocales" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 8.3 [Fuentes tipográficas], página 341, Sección 8.2 [Formatear el texto], página 323.

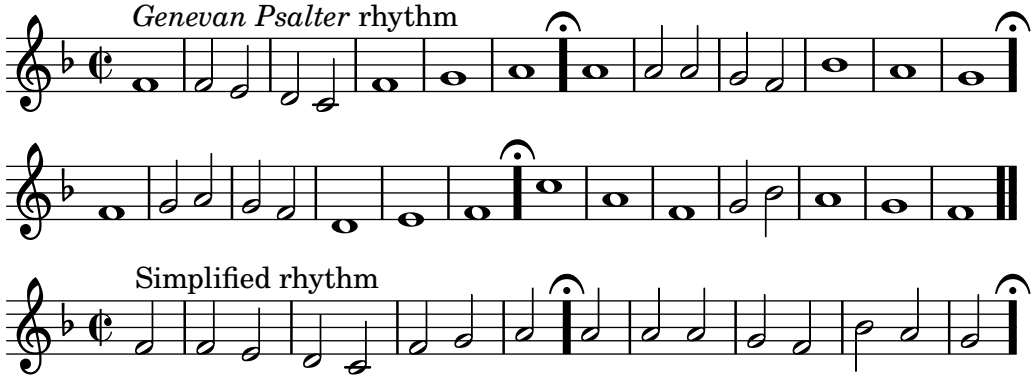
9.7.4 Barras de frase en himnos

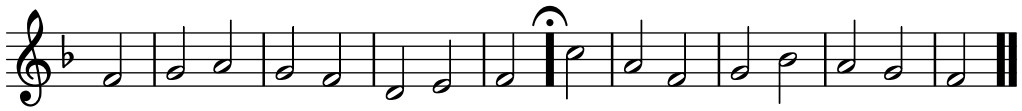
Se puede configurar la instrucción `\caesura` para crear líneas divisorias de fraseo que interactúan bien con las otras líneas divisorias automáticas (véase Sección 2.5.2 [Automatic bar lines], página 130).

En los ejemplos siguientes, el código fuente de la melodía de `old-hundredth-example.ly` usa `\caesura` entre los versos de la letra y `\fine` al final. No hay ninguna instrucción `\fermata` ni `\bar`; estos símbolos aparecen en el resultado a causa de la configuración de la cesura.

The Boston Handel and Haydn Society Collection of Church Music (1830) tiene una línea divisoria gruesa después de cada frase. Para esta melodía, también tiene calderones sobre las líneas divisorias:

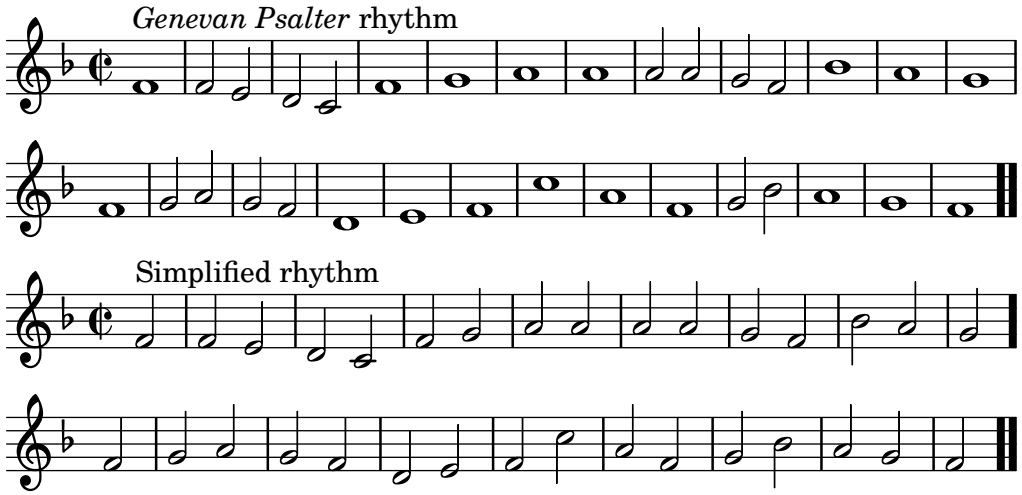
```
\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . ".")
                     (scripts . (fermata)))
    fineBarType = ".."
  }
}
\include "old-hundredth-example.ly"
```





Original Sacred Harp de J.S. James (1911) tiene una línea divisoria gruesa cuando se salta de línea a mitad del compás al final de una frase.

```
\layout {
  \context {
    \Score
    caesuraType = #'((underlying-bar-line . "x-."))
    fineBarType = ".."
  }
}
\include "old-hundredth-example.ly"
```



9.7.5 Compases parciales en melodías de himno

Con frecuencia las melodías de himno comienzan y terminan cada línea de música con compases parciales de forma que cada línea de música corresponde exactamente con una línea de texto. Esto requiere una instrucción `\partial` al principio de la música e instrucciones `\bar "|"` ó `\bar "||"` al final de cada línea.

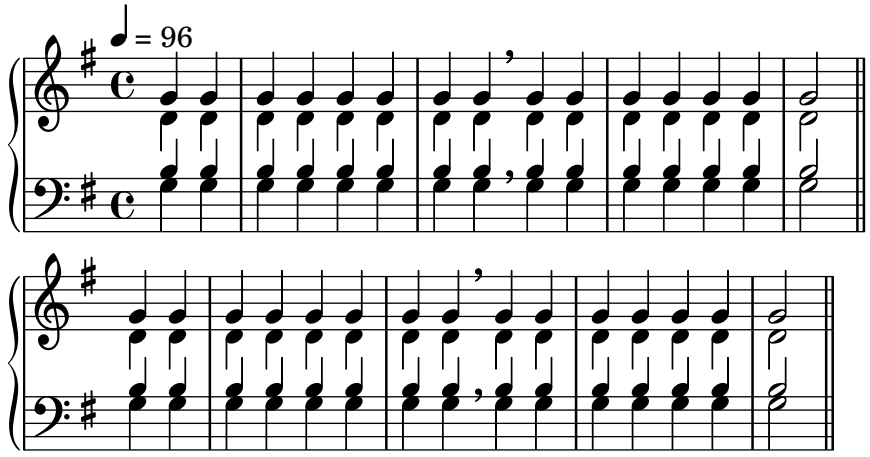
Plantilla para himnos

Este fragmento de código muestra una forma de preparar un himno en el que cada línea comienza con un compás parcial. También muestra cómo añadir los versos como texto independiente debajo de la música.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \caesura \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \fine
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}
```

```
AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
 \key g \major
}

\score {  % Start score
  \new PianoStaff <<  % Start pianostaff
    \new Staff <<  % Start Staff = RH
      \global
      \clef "treble"
      \new Voice = "Soprano" <<  % Start Voice = "Soprano"
        \Timeline
        \voiceOne
        \SopranoMusic
      >>  % End Voice = "Soprano"
      \new Voice = "Alto" <<  % Start Voice = "Alto"
        \Timeline
        \voiceTwo
        \AltoMusic
      >>  % End Voice = "Alto"
    >>  % End Staff = RH

    \new Staff <<  % Start Staff = LH
      \global
      \clef "bass"
      \new Voice = "Tenor" <<  % Start Voice = "Tenor"
        \Timeline
        \voiceOne
        \TenorMusic
      >>  % End Voice = "Tenor"
      \new Voice = "Bass" <<  % Start Voice = "Bass"
        \Timeline
        \voiceTwo
        \BassMusic
      >>  % End Voice = "Bass"
    >>  % End Staff = LH
  >>  % End pianostaff
```

```
}  % End score

\markup \fill-line {
  \left-column {
    "This is line one of the first verse"
    "This is line two of the same"
    \null
    "And here's line one of the second verse"
    "And the next line of the same"
  }
}

\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . "||"))
    fineBarType = "||"
  }
}

\paper {  % Start paper block
  indent = 0         % don't indent first system
  line-width = 130   % shorten line length to suit music
  tagline = ##f      % Don't print tag line, can be removed
}  % End paper block
```



This is line one of the first verse
This is line two of the same

And here's line one of the second verse
And the next line of the same

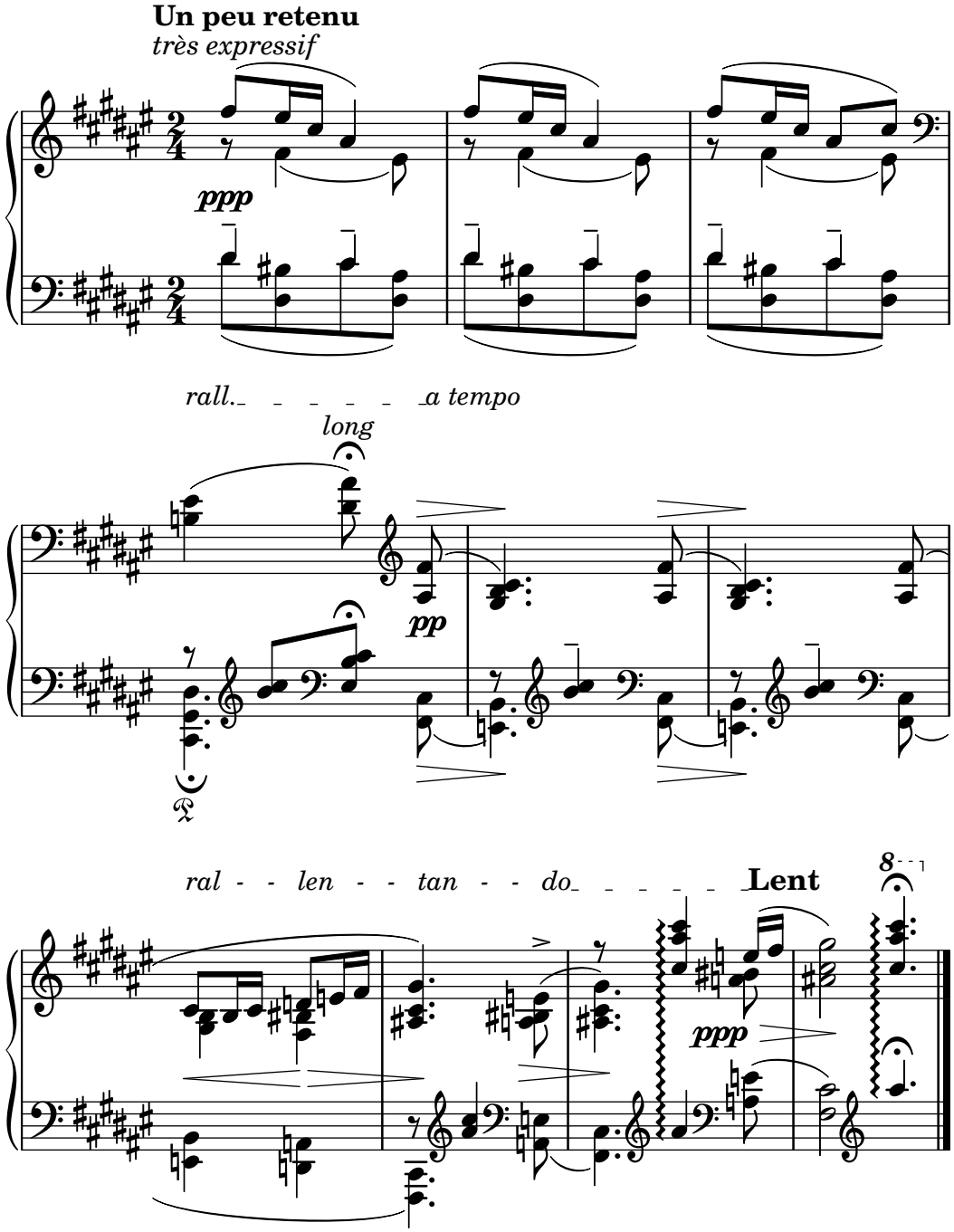
9.8 Música vocal antigua

Está contemplada la música vocal en estilo de tipografía antiguo, como se explica en Capítulo 17 [Notación antigua], página 540.

Véase también

Referencia de la notación: Capítulo 17 [Notación antigua], página 540.

10 Teclados y otros instrumentos de varios pentagramas



Esta sección se ocupa de varios aspectos de la notación musical que son exclusivos de los instrumentos de teclado y otros instrumentos cuya notación se realiza sobre varios pentagramas, como el arpa o el vibráfono. A los efectos de nomenclatura, en esta sección se denominan abreviadamente "teclados" a todo este grupo de instrumentos de varios pentagramas, aunque algunos de ellos no tienen teclado.

10.1 Notación común para instrumentos de teclado

Esta sección trata aspectos de notación que pueden aparecer en casi todos los instrumentos de teclado.

10.1.1 Referencias para teclados

La notación de los instrumentos de teclado se suele realizar con sistemas de piano. Éstos son dos pentagramas normales agrupados mediante una llave. También se utiliza esta notación para otros instrumentos de teclado. La música de órgano se escribe normalmente con dos pentagramas dentro de un grupo `PianoStaff` y un tercer pentagrama normal para los pedales.

Los pentagramas son hasta cierto punto independientes, pero a veces las voces pueden cruzarse entre los dos pentagramas. Esta sección trata técnicas de notación que son particulares de la música de teclado.

En otros lugares se tratan varios problemas comunes de la música de teclado:

- La música para teclado suele contener varias voces y el número de éstas puede variar ampliamente; esto se describe en Sección 5.2.3 [Resolución de las colisiones], página 229.
- La música para teclado se puede escribir en paralelo, como se describe en Sección 5.2.6 [Escribir música en paralelo], página 241.
- Los matices dinámicos se pueden colocar en un contexto `Dynamics`, entre los dos contextos `Staff` para alinear las indicaciones de matiz sobre una línea horizontal equidistante de los dos pentagramas; véase Sección 3.1.2 [Matices dinámicos], página 160.
- Las digitaciones se indican como puede verse en Sección 7.1.2 [Indicaciones de digitación], página 284.
- Las indicaciones de pedal de órgano se insertan como articulaciones, véase Sección 10.3.1 [Indicaciones de pedal de órgano], página 425, y Sección B.13 [Lista de articulaciones], página 934.
- Las líneas verticales de rejilla se pueden mostrar como se describe en Sección 7.2.3 [Líneas de rejilla], página 304.
- La música para teclado incluye con frecuencia ligaduras *Laissez vibrer* así como ligaduras sobre arpegios y trémolos, descritos en Sección 2.1.4 [Ligaduras de unión], página 63.
- La colocación de arpegios a través de varias voces y pentagramas se cubre en Sección 3.3.2 [Arpegio], página 183.
- Las indicaciones de trémolo se describen en Sección 4.2.2 [Repeticiones de trémolo], página 214.
- Varios de los trucos que pueden ser necesarios en música para teclado se muestran en Sección "Ejemplos reales de música" en *Manual de Aprendizaje*.
- Las notas ocultas se pueden usar para producir ligaduras de unión que se cruzan entre voces, como se explica en Sección "Otras aplicaciones de los trucos" en *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Ejemplos reales de música" en *Manual de Aprendizaje*, Sección "Otras aplicaciones de los trucos" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 6.1.2 [Agrupar pentagramas], página 246, Sección 6.3.1 [Nombres de instrumentos], página 265, Sección 5.2.3 [Resolución de las colisiones], página 229, Sección 5.2.6 [Escribir música en paralelo], página 241, Sección 7.1.2 [Indicaciones de digitación], página 284, Sección B.13 [Lista de articulaciones], página 934, Sección 7.2.3 [Líneas de rejilla], página 304, Sección 2.1.4 [Ligaduras de unión], página 63, Sección 3.3.2 [Arpegio], página 183, Sección 4.2.2 [Repeticiones de trémolo], página 214.

Referencia de funcionamiento interno: Sección "PianoStaff" en *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección "Teclados y otros instrumentos de varios pentagramas" en *Fragmentos de código*.

10.1.2 Cambiar de pentagrama manualmente

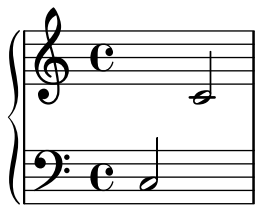
Las voces se pueden cambiar de pentagrama manualmente usando la instrucción

```
\change Staff = nombre_del_pentagrama
```

La cadena *nombre_del_pentagrama* es el nombre del pentagrama. Cambia la voz actual de su pentagrama en curso al pentagrama de nombre *nombre_del_pentagrama*. Los valores más corrientes para *nombre_del_pentagrama* son `"arriba"` y `"abajo"`, o bien `"MD"` y `"MI"`.

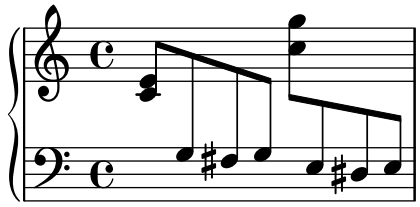
El pentagrama al que salta la voz debe existir en el momento del salto. Si es necesario, los pentagramas se tienen que "mantener con vida", véase Sección 33.3 [Mantener vivos los contextos], página 749, o instanciados explícitamente, por ejemplo mediante la utilización del acorde vacío, <>, véase Sección 5.1.1 [Notas en acorde], página 218.

```
\new PianoStaff <<
  \new Staff = "up" {
    % enforce creation of all contexts at this point of time
    <>
    \change Staff = "down" c2
    \change Staff = "up" c'2
  }
  \new Staff = "down" {
    \clef bass
    % keep staff alive
    s1
  }
>>
```



Las notas de pentagrama cruzado reciben la barra automáticamente:

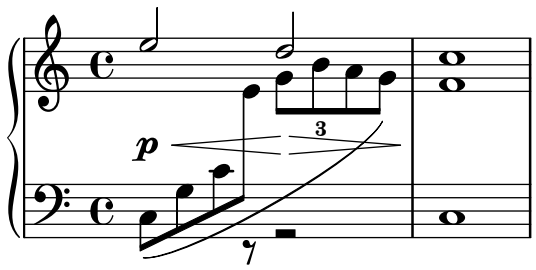
```
\new PianoStaff <<
  \new Staff = "up" {
    <e' c'>8
    \change Staff = "down"
    g8 fis g
    \change Staff = "up"
    <g'' c''>8
    \change Staff = "down"
    e8 dis e
    \change Staff = "up"
  }
  \new Staff = "down" {
    \clef bass
    % keep staff alive
    s1
  }
>>
```



Si el barrado necesita algún trucaje, efectúe en primer lugar los cambios en las direcciones de las plicas. Las posiciones de las barras se miden posteriormente a partir del centro del pentagrama que está más cerca de la barra. Para ver un ejemplo sencillo de trucaje de las barras, consulte Sección “Arreglar notación con superposiciones” en *Manual de Aprendizaje*.

Se puede producir una superposición en la notación cuando las voces se cruzan de un pentagrama a otro:

```
\new PianoStaff <<
  \new Staff = "up" {
    \voiceOne
    % Make space for fingering in the cross-staff voice
    \once\override DynamicLineSpanner.staff-padding = 4
    e''2\p\< d''\>
    c''1\!
  }
  \new Staff = "down" <<
    {
      \clef bass
      s4. e,8\rest g,2\rest
      c1
    } \\ {
      c8\( g c'
      \change Staff = "up"
      e' g' b'-3 a' g'\)
      f'1
    }
  >>
>>
```



La plica y la ligadura de expresión se superponen a la línea del símbolo de dinámica porque la resolución automática de colisiones se suspende para las barras, ligaduras y otros objetos de extensión que conectan notas situadas en distintos pentagramas, así como para las plicas y articulaciones si su colocación está afectada por un objeto de extensión de pentagrama cruzado. Las colisiones resultantes se pueden resolver manualmente donde sea necesario, utilizando los métodos descritos en la sección Sección “Arreglar notación con superposiciones” en *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Arreglar notación con superposiciones” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 7.1.9 [Plicas], página 300, Sección 2.4.1 [Barras automáticas], página 101, Sección 33.3 [Mantener vivos los contextos], página 749.

Fragmentos de código: Sección “Teclados y otros instrumentos de varios pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “ContextChange” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

El mecanismo para evitar las colisiones de barras de corchea no funciona para las barras automáticas que terminan justo antes de un cambio de pauta. En este caso, utilice barras manuales.

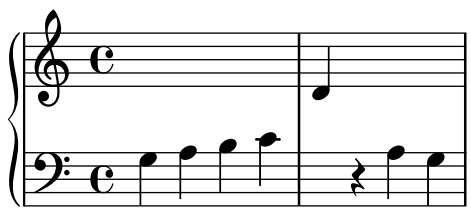
10.1.3 Cambiar de pentagrama automáticamente

Se puede hacer que las voces cambien automáticamente entre los pentagramas inferior y superior. La sintaxis para conseguir esto es

```
\autoChange ...música...
```

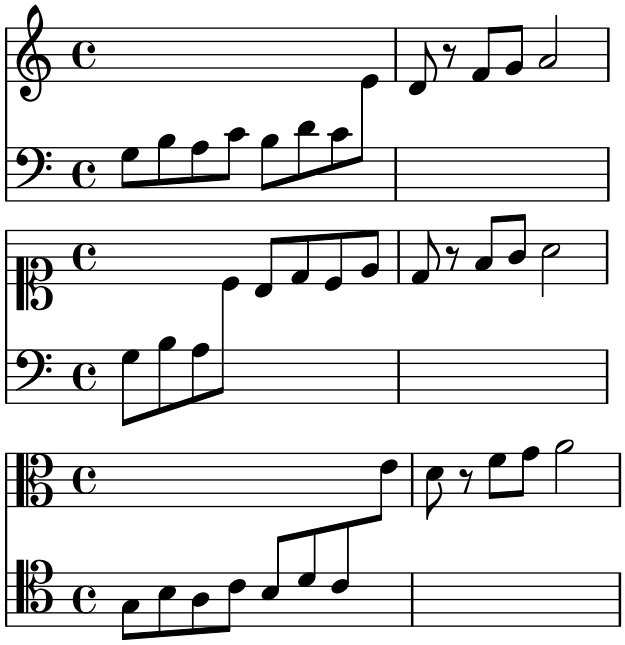
Esto creará dos pentagramas dentro del grupo actual de pentagramas (normalmente un `PianoStaff`), llamados `"up"` (arriba) y `"down"` (abajo). El pentagrama inferior estará en clave de Fa de forma predeterminada. El cambiador automático hace el cambio basándose en la altura (el Do central es el punto de inflexión), y se adelanta saltando sobre los silencios para hacer cambios por adelantado.

```
\new PianoStaff {
  \autoChange {
    g4 a b c'
    d'4 r a g
  }
}
```



Se pueden especificar otras notas para el punto de corte. Si los pentagramas no han sido instanciados de forma explícita, se pueden usar otras claves.

```
music = {
  g8 b a c' b8 d' c'8 e'
  d'8 r f' g' a'2
}

\autoChange d' \music
\autoChange b \with { \clef soprano } \music
\autoChange d' \with { \clef alto } \with { \clef tenor } \music
```



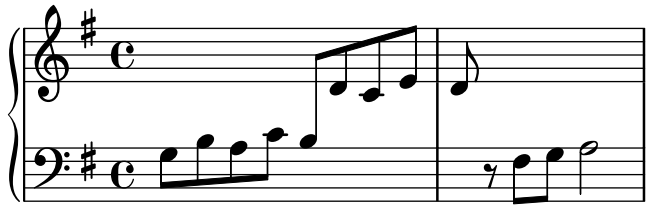
Una sección marcada `\relative` que se encuentra fuera de `\autoChange` no tiene efecto sobre las notas de la música; por tanto, en caso necesario escriba `\relative` dentro de `\autoChange`.

Si se necesita más control sobre los pentagramas individuales, se pueden crear manualmente con los nombres `"up"` y `"down"`. Entonces, la instrucción `\autoChange` alternará su voz entre los pentagramas existentes.

> **Nota:** Si se crean los pentagramas manualmente, *se deben* llamar `"up"` y `"down"` (en inglés).

Por ejemplo, esto es necesario para colocar una armadura de tonalidad en el pentagrama inferior:

```
\new PianoStaff <<
  \new Staff = "up" {
    \new Voice = "melodiaUno" {
      \key g \major
      \autoChange \relative {
        g8 b a c b d c e
        d8 r fis, g a2
      }
    }
  }
  \new Staff = "down" {
    \key g \major
    \clef bass
  }
>>
```



Véase también

Referencia de la notación: Sección 10.1.2 [Cambiar de pentagrama manualmente], página 417.

Fragmentos de código: Sección “Teclados y otros instrumentos de varios pentagramas” en *Fragmentos de código*.

Advertencias y problemas conocidos

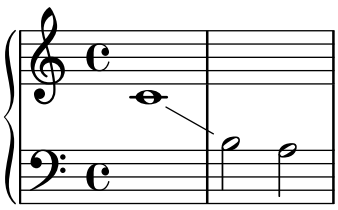
Los cambios de pentagrama pueden no acabar en lugares óptimos. Para conseguir una salida de alta calidad es mejor especificar manualmente los cambios de pentagrama.

Los acordes no se dividen entre los pentagramas; se asignan a un pentagrama en función de la primera nota nombrada dentro de la construcción del acorde.

10.1.4 Líneas de cambio de pentagrama

Cuando una voz cambia a otro pentagrama, se puede imprimir automáticamente una línea que conecte las notas:

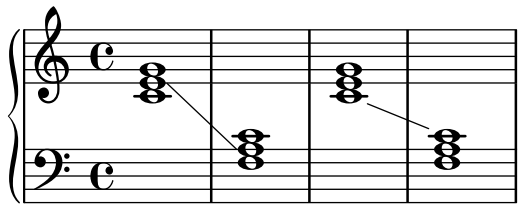
```
\new PianoStaff <<
  \new Staff = "one" {
    \showStaffSwitch
    c'1
    \change Staff = "two"
    b2 a
  }
  \new Staff = "two" {
    \clef bass
    s1*2
  }
>>
```



Una línea de cambio de pauta entre acordes conecta las “últimas notas” de cada uno de ellos según se han escrito en el archivo de entrada; Esto se puede usar para ajustar rápidamente las posiciones verticales del comienzo y el final de la línea.

```
\new PianoStaff <<
  \new Staff = "one" {
    <c' e' g'>1
    \showStaffSwitch
    \change Staff = "two"
    <a c' f>1
    \hideStaffSwitch
    \change Staff = "one"
    <e' g' c'>1
    \showStaffSwitch
    \change Staff = "two"
    <f a c'>1
  }
  \new Staff = "two" {
```

```
    \clef bass
    s1*4
  }
>>
```



Instrucciones predefinidas

`\showStaffSwitch`, `\hideStaffSwitch`.

Véase también

Fragmentos de código: Sección “Teclados y otros instrumentos de varios pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note_head_line_engraver” en *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” en *Referencia de Funcionamiento Interno*.

Fragmentos de código seleccionados

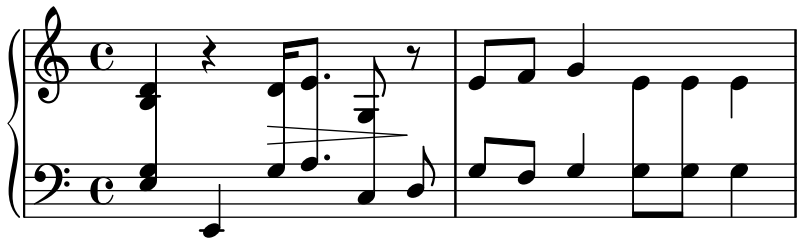
Plicas de pentagrama cruzado

Este fragmento de código muestra el uso del grabador `Span_stem_engraver` y de `\crossStaff` para conectar automáticamente plicas de un pentagrama a otro.

No es necesario especificar las longitudes de las plicas porque la distancia variable entre las cabezas de las notas y los pentagramas se calcula automáticamente. Sin embargo, es importante aplicar `\crossStaff` a la voz o pentagrama correctos (esto es, en el lado opuesto de donde está o estaría posicionada la barra de corcheas) para conseguir el efecto deseado

```
\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
      \voiceTwo
      % Down to lower staff
      \crossStaff { e'8 e'8 } e'4 |
  }

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
```

```
    g8 f g4 \voiceTwo g8 g g4 |
  }
>>
```

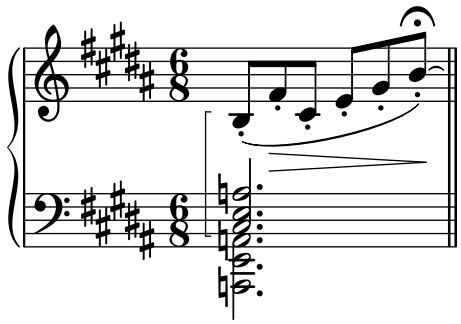


Indicating cross-staff chords with a bracket

A non-arpeggiato bracket can indicate that notes on two different staves are to be played with the same hand. In order to do this, the `PianoStaff` must be set to accept cross-staff brackets.

The following example typesets measure 65 of Debussy's prelude *Les collines d'Anacapri*.

```
\new PianoStaff <<
  \once \set PianoStaff.connectChordBrackets = ##t

  \new Staff \relative c' {
    \key b \major
    \time 6/8
    b8-.(\nonArpeggiato fis'-.\> cis-.
      e-. gis-. b-.)\!\fermata^\laissezVibrer
    \section
  }

  \new Staff \relative c' {
    \clef bass
    \key b \major
    << { <a e cis>2.\nonArpeggiato } \\
       { <a, e a,>2. } >>
    \section
  }
>>
```



Véase también

Fragmentos de código: Sección "Teclados y otros instrumentos de varios pentagramas" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Stem" en *Referencia de Funcionamiento Interno*.

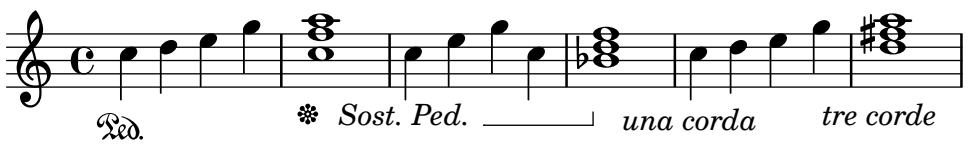
10.2 Piano

Esta sección trata de aspectos de la notación directamente relacionados con el piano.

10.2.1 Pedales de piano

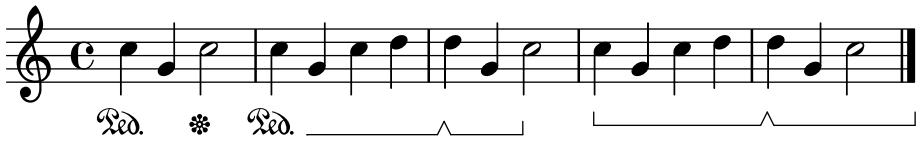
En general, los pianos tienen tres pedales que alteran la forma de producir el sonido: el pedal de mantenimiento (pedal *derecho* o pedal *forte*), el pedal *sostenuto* (*sost.*, pedal central) y el pedal *una corda* (*U.C.*, pedal izquierdo). Los pedales de mantenimiento como el pedal derecho del piano se encuentran también en el vibráfono y en la celesta.

```
\relative {
  c''4\sustainOn d e g
  <c, f a>1\sustainOff
  c4\sostenutoOn e g c,
  <bes d f>1\sostenutoOff
  c4\unaCorda d e g
  <d fis a>1\treCorde
}
```



Existen tres estilos de indicaciones de pedal: textual, corchete y mixta. El pedal derecho y el pedal de una corda utilizan el estilo textual de forma predeterminada, mientras que el pedal sostenuto utiliza el método mixto de forma predeterminada.

```
\relative {
  c''4\sustainOn g c2\sustainOff
  \set Staff.pedalSustainStyle = #'mixed
  c4\sustainOn g c d
  d\sustainOff\sustainOn g, c2\sustainOff
  \set Staff.pedalSustainStyle = #'bracket
  c4\sustainOn g c d
  d\sustainOff\sustainOn g, c2
  \bar "|."
}
```



La colocación de las instrucciones de pedal se corresponden con el movimiento físico del pedal derecho en el transcurso de la ejecución. La pedalización hasta la barra final de compás se indica mediante la omisión de la instrucción final de levantar el pedal.

Las indicaciones de pedal se pueden poner dentro de un contexto `Dynamics`, que los alinea sobre una línea horizontal.

Véase también

Referencia de la notación: Sección 2.1.4 [Ligaduras de unión], página 63.

Fragmentos de código: Sección “Teclados y otros instrumentos de varios pentagramas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SustainPedal” en *Referencia de Funcionamiento Interno*, Sección “SustainPedalLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “SustainEvent” en *Referencia de Funcionamiento Interno*, Sección “SostenutoPedal” en *Referencia de Funcionamiento Interno*, Sección “SostenutoPedalLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “SostenutoEvent” en *Referencia de Funcionamiento Interno*, Sección “UnaCordaPedal” en *Referencia de Funcionamiento Interno*, Sección “UnaCordaPedalLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “UnaCordaEvent” en *Referencia de Funcionamiento Interno*, Sección “PianoPedalBracket” en *Referencia de Funcionamiento Interno*, Sección “Piano_pedal_engraver” en *Referencia de Funcionamiento Interno*.

10.3 Órgano

En esta sección se discuten los problemas de notación relativos al órgano de tubos.

10.3.1 Indicaciones de pedal de órgano

En las partituras para órgano, especialmente en los métodos de enseñanza, podemos encontrar indicaciones sobre cómo usar la punta y el talón de los pies izquierdo y derecho cuando se toca el pedalero. Dependiendo del estilo del libro, se usan distintos glifos para estas indicaciones. Sin embargo, casi siempre las indicaciones de pedal por encima del pentagrama son para el pie derecho, y las que están por debajo son para el pie izquierdo.

Las instrucciones principales para imprimir estas marcas son `\rtoe`, `\ltoe`, `\rheel` y `\lheel`. Use la propiedad de contexto `toeHeelStyle` para seleccionar el estilo de las marcas de pedal. Las opciones disponibles aparecen a continuación.

`default` El ajuste por omisión durante el arranque, especialmente por compatibilidad hacia atrás.

`standard` El esquema maś común que se usa actualmente; las formas son iguales por encima y por debajo del pentagrama.

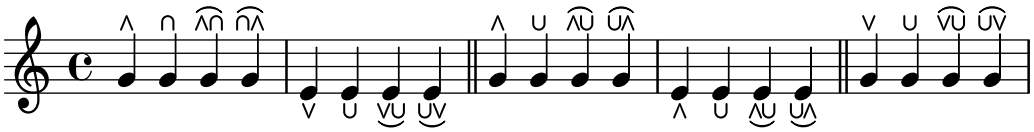
`reversed` La ‘punta’ del glifo apunta hacia el pentagrama.

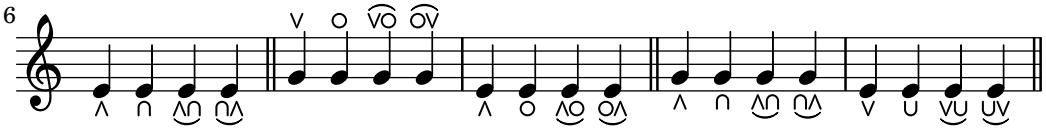
`circleheels` Usar un glifo circular para el talón.

`below` Este es para las raras ocasiones en que la notación del pedal se hace en la misma pauta que la mano izquierda, lo que hace necesario que las marcas de los pies izquierdo y derecho estén por debajo del pentagrama.

Para las sustituciones *toe-heel* (punta-talón) y *heel-toe* (talón-punta) están las instrucciones `\rtoeheel`, `\ltoeheel`, `\rheeltoe` y `\lheeltoe`.

```
music = { g'4_\rtoe g'\rheel g'\rtoeheel g'\rheeltoe |
          e'4\ltoe e'\lheel e'\ltoeheel e'\lheeltoe \bar "||" }

{
  \music            % #'default
  \set toeHeelStyle = #'standard \music
  \set toeHeelStyle = #'reversed \music
  \set toeHeelStyle = #'circleheels \music
  \set toeHeelStyle = #'below \music
}
```

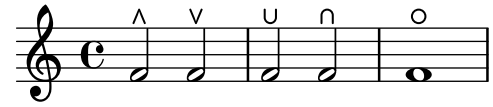




\rtoe y sus instrucciones hermanas tienen una apariencia similar a otras instrucciones de articulación como \flageolet o \prall, y por supuesto son objetos gráficos del tipo Script. Sin embargo, ignoran los cambios de dirección '^', '_' y \tweak direction. Esto es debido al grabador Toe_heel_engraver, que está activado por defecto. La idea que se encuentra detrás de esta decisión es la separación del contenido y su representación, dejando que un estilo sea el que controla la posición de las marcas.

Si surge la necesidad de posicionar las indicaciones de pedal individuales por encima o por debajo de la pauta, use las instrucciones estándar de articulación \toe, \vartoe, \heel, \varheel y \heelcircle.

```
{
  f'2\toe f'2\vartoe |
  f'2\heel f'2\varheel |
  f'1\heelcircle
}
```



10.4 Acordeón

Esta sección se ocupa de la notación que es exclusiva del acordeón.

10.4.1 Símbolos de Discanto

Los acordeones se suelen construir con más de un conjunto de lengüetas que pueden estar al unísono, una octava por encima, o una octava por debajo de la altura escrita. Cada fabricante de acordeones utiliza distintos nombres para los *cambios* que seleccionan las distintas combinaciones de lengüetas, como *oboe*, *musette* o *bandoneón*, de manera que ha entrado en uso un sistema de signos para simplificar las instrucciones de ejecución.

Puede verse una lista completa de todos los registros de acordeón disponibles en Sección A.1.7 [Accordion registers], página 874,

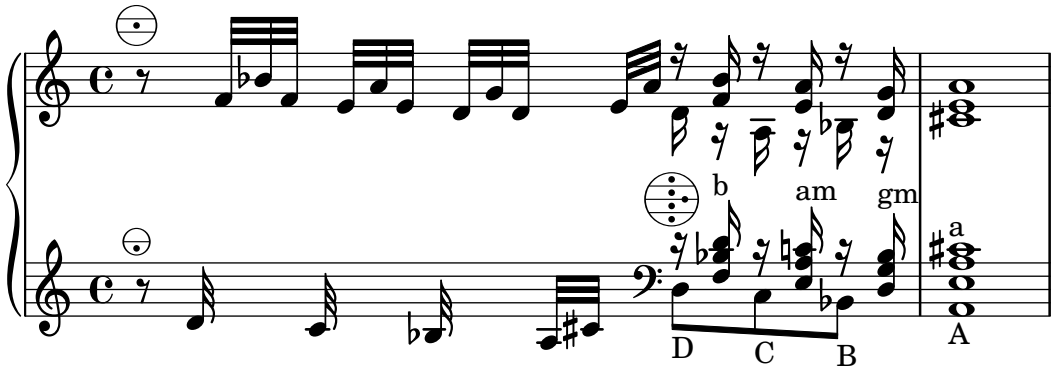
Fragmentos de código seleccionados

Símbolos de registración de acordeón

Los símbolos de registración de acordeón están disponibles como elementos de \markup y como eventos musicales autocontenidos (pues los cambios de registro tienden a ocurrir entre eventos de música). Los registros del bajo no están demasiado estandarizados. Las instrucciones existentes están disponibles en el apartado 'Símbolos de acordeón' en la Referencia de la notación (https://lilypond.org/doc/v2.24/Documentation/notation/accordion#discant-symbols).

```
#(use-modules (lily accreg))

\new PianoStaff <<
  \new Staff \relative {
    \clef treble
    \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
```

```
    <<
      { r16 <f bes> r <e a> r <d g> }
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
      <e a cis>1^"a" }
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```



Véase también

Fragmentos de código: Sección “Teclados y otros instrumentos de varios pentagramas” en *Fragmentos de código*.

10.5 Arpa

Esta sección trata sobre asuntos de notación específicos del arpa.

10.5.1 Referencias para notación de arpa

Ciertas características comunes de la música de arpa se estudian en otras partes de la documentación:

- El glissando es la técnica más característica del arpa, Sección 3.3.1 [Glissando], página 178.
- El *bisbigliando* se escribe como un trémolo, Sección 4.2.2 [Repeticiones de trémolo], página 214.
- Los armónicos naturales se estudian bajo el epígrafe Sección 11.1.3 [Armónicos], página 430.
- Para los arpegios dirigidos y los no arpegios, véase Sección 3.3.2 [Arpegio], página 183.

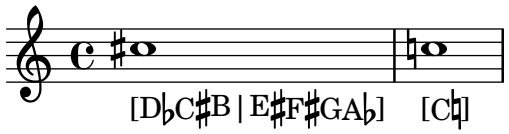
Véase también

Referencia de la notación: Sección 4.2.2 [Repeticiones de trémolo], página 214, Sección 3.3.1 [Glissando], página 178, Sección 3.3.2 [Arpegio], página 183, Sección 11.1.3 [Armónicos], página 430.

10.5.2 Pedales de arpa

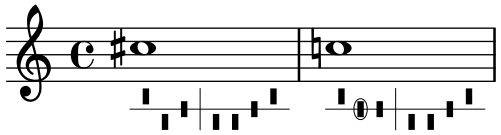
Las arpas tienen siete cuerdas por octava que pueden sonar a la altura natural, bemol o sostenido. En el arpa de palancas o arpa celta, cada cuerda se ajusta individualmente, pero en las arpas de pedales todas las cuerdas con igual nombre de nota se controlan por un único pedal. Visto de izquierda a derecha desde la posición del ejecutante, los pedales son Re, Do y Si a la izquierda y Mi, Fa, Sol y La a la derecha. La posición de los pedales se puede indicar con indicaciones textuales:

```
\textLengthOn
cis''1_\markup \concat \vcenter {
  [D \flat C \sharp B|E \sharp F \sharp G A \flat] }
c''!1_\markup \concat \vcenter {
  [ C \natural ] }
```



o mediante diagramas de pedal:

```
\textLengthOn
cis''1_\markup { \harp-pedal "^v-|vv-^" }
c''!1_\markup { \harp-pedal "^o--|vv-^" }
```

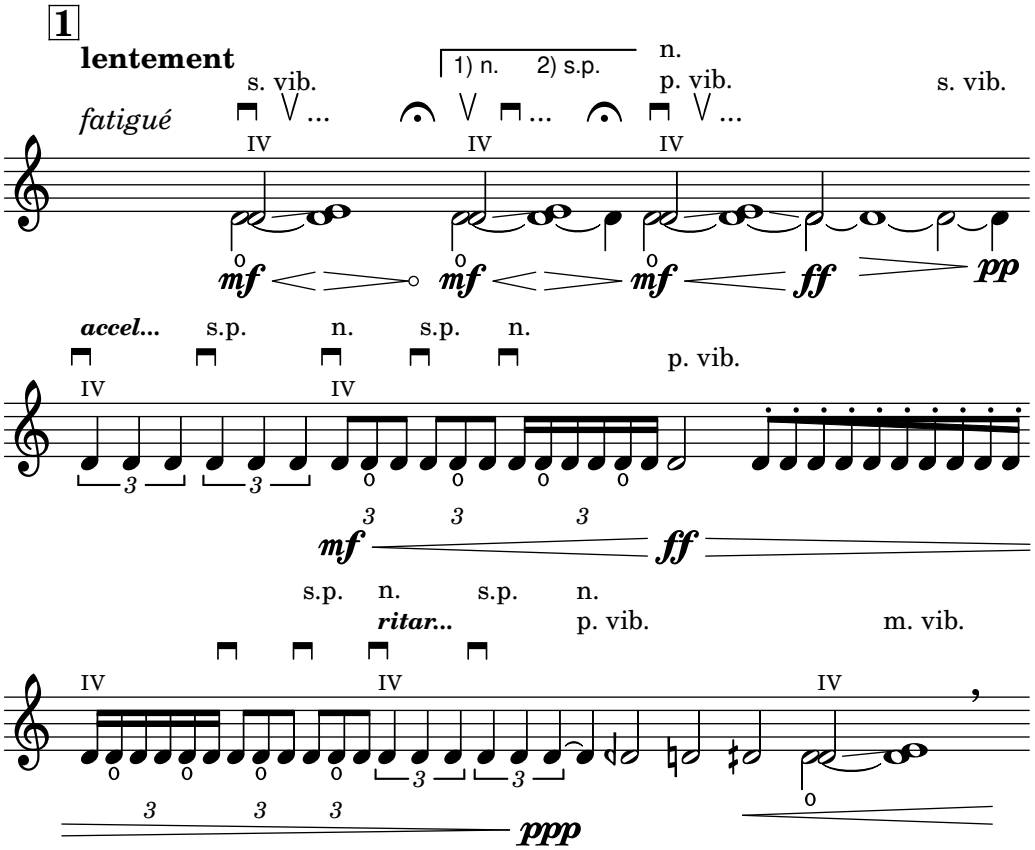


La instrucción `\harp-pedal` acepta una cadena de caracteres en la que `^` es la posición alta del pedal (altura bemol), `-` es la posición intermedia (altura natural), `v` es la posición baja (altura sostenido), y `|` es la línea vertical separadora. Si se antepone `o`, el siguiente símbolo de pedal se rodea por una elipse.

Véase también

Referencia de la notación: Sección 8.1.2 [Inscripciones textuales], página 314, Sección A.1.6 [Instrument-specific markup], página 869.

11 Instrumentos de cuerda sin trastes



Esta sección aporta información y referencias que resultan útiles si se escribe música para instrumentos de cuerda sin trastes, principalmente instrumentos de cuerda orquestales.

11.1 Notación común para cuerdas sin trastes

Hay poca notación especializada para los instrumentos de cuerdas sin trastes. La notación de la música se realiza en un solo pentagrama, y se suele usar una sola voz. Se pueden necesitar dos voces para algunos pasajes en dobles cuerdas o en divisi.

11.1.1 Referencias para cuerdas sin trastes

La mayor parte de la notación que resulta de utilidad para las cuerdas orquestales y otros instrumentos de arco se estudia en otras partes del manual:

- Las indicaciones textuales como “pizz.” y “arco” se añaden como texto simple: véase Sección 8.1.2 [Inscripciones textuales], página 314.
- Las digitaciones, entre ellas la indicación de pulgar, se describen en Sección 7.1.2 [Indicaciones de digitación], página 284.
- Se pueden añadir números de cuerda (en general como números romanos para los instrumentos de cuedra frotada) como se explica en Sección 12.1.2 [Indicación de los números de cuerda], página 434.
- Las dobles cuerdas se indican normalmente escribiendo un acorde, véase Sección 5.1.1 [Notas en acorde], página 218. Hay otras indicaciones que se pueden añadir para la interpretación de acordes, véase Sección 3.3.2 [Arpegio], página 183.
- Hay plantillas para cuarteto de cuerda en Sección “Plantillas de cuarteto de cuerda” en *Manual de Aprendizaje*. Otras se muestran en la sección de fragmentos de código.

Véase también

Manual de aprendizaje: Sección “Plantillas de cuarteto de cuerda” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 8.1.2 [Inscripciones textuales], página 314, Sección 7.1.2 [Indicaciones de digitación], página 284, Sección 5.1.1 [Notas en acorde], página 218, Sección 3.3.2 [Arpegio], página 183.

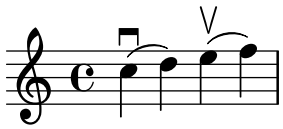
Fragmentos de código: Sección “Instrumentos de cuerda sin trastes” en *Fragmentos de código*.

11.1.2 Indicaciones de arco

Las indicaciones de arco se crean como articulaciones, que se describen en Sección 3.1.1 [Articulaciones y ornamentos], página 156.

Las instrucciones de arco `\upbow` (arco arriba) y `\downbow` (arco abajo) se usan con ligaduras de expresión de la siguiente manera:

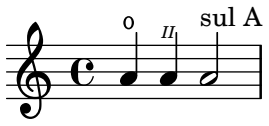
```
\relative { c''4(\downbow d) e(\upbow f) }
```



Se pueden usar números romanos para los números de cuerda (en sustitución de la opción predeterminada, que son cifras arábigas rodeadas por un círculo), tal y como se explica en Sección 12.1.2 [Indicación de los números de cuerda], página 434.

Como alternativa, se pueden imprimir las indicaciones de cuerda mediante instrucciones de marcado; los símbolos de articulación también pueden indicar cuerdas al aire.

```
a'4 \open
\romanStringNumbers
a'\2
a'2^\markup { \small "sul A" }
```



Instrucciones predefinidas

`\downbow`, `\upbow`, `\open`, `\romanStringNumbers`.

Véase también

Referencia de la notación: Sección 3.1.1 [Articulaciones y ornamentos], página 156, Sección 12.1.2 [Indicación de los números de cuerda], página 434, Sección 3.2.1 [Ligaduras de expresión], página 171.

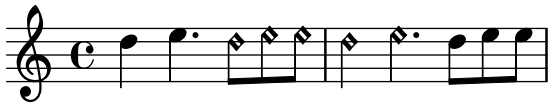
11.1.3 Armónicos

Armónicos naturales

La notación de los armónicos naturales se puede realizar de varias formas. Generalmente, una nota con la cabeza en forma de rombo significa tocar (sin apretar) la cuerda en el lugar donde se pisaría la nota si no fuese un rombo.

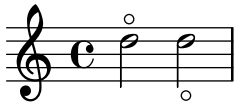
```
\relative d'' {
  d4 e4.
  \harmonicsOn
```

```
  d8 e e
  d4 e4.
  \harmonicsOff
  d8 e e
}
```



Como posibilidad alternativa está la de mostrar una cabeza normal de nota en la altura de la nota que debe sonar, con un pequeño círculo que indica que se debe tocar como armónico:

```
d''2^\flageolet d''_\flageolet
```

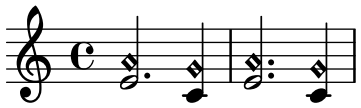


Armónicos artificiales

La notación de los armónicos artificiales se realiza con dos notas, una con una cabeza normal que indica la posición donde se pisa, y otra con una cabeza en forma de rombo hueco para indicar la posición donde se roza la cuerda (sin pisar) para producir el armónico.

Los armónicos artificiales indicados con `\harmonic` no presentan los puntillos. La propiedad de contexto `harmonicDots` se debe activar si se necesitan los puntillos.

```
\relative e' {
  <e a\harmonic>2.  <c g'\harmonic>4
  \set harmonicDots = ##t
  <e a\harmonic>2.  <c g'\harmonic>4
}
```



Véase también

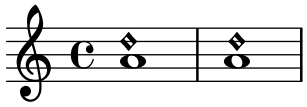
Glosario musical: Sección “harmonics” en *Glosario Musical*.

Referencia de la notación: Sección 1.4.1 [Cabezas de nota especiales], página 44, Sección 11.1.1 [Referencias para cuerdas sin trastes], página 429.

Advertencias y problemas conocidos

Si queremos centrar un armónico sobre una redonda, podemos hacer que forme parte de un acorde.

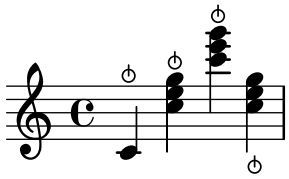
```
{
  << { d''1\harmonic } \\ { a'1 } >>
  <a' d''\harmonic>1
}
```



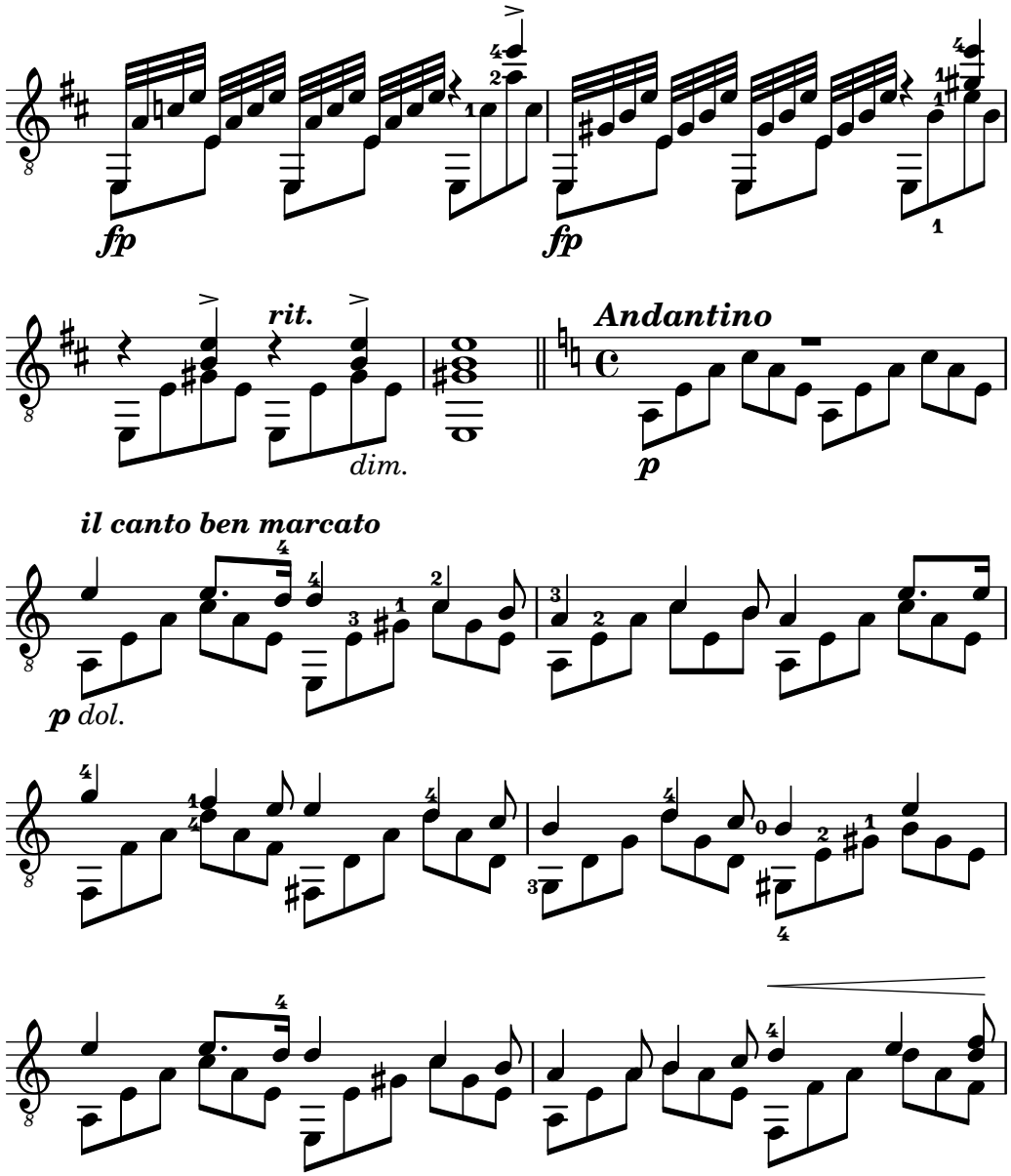
11.1.4 Pizzicato Snap o de Bartók

El *snap pizzicato* (conocido también como "pizzicato de Bartók") es un tipo de pizzicato en el que la cuerda se pulsa hacia arriba (y no lateralmente) de forma deliberada, de forma que golpea la madera del diapasón.

```
\relative {
  c'4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```



12 Instrumentos de cuerda con trastes



Esta sección trata varios aspectos de la notación musical que son exclusivos de los instrumentos de cuerda con trastes.

12.1 Notación común para cuerdas con trastes

Esta sección se ocupa de la notación común exclusiva de los instrumentos de cuerda con trastes.

12.1.1 Referencias para cuerdas con trastes

La notación musical para instrumentos de cuerda con trastes se realiza normalmente en una sola pauta, ya sea en notación musical tradicional o en tablatura. A veces se combinan los dos tipos, y es común en música popular utilizar diagramas de acordes por encima de un pentagrama de notación tradicional. La guitarra y el banjo son instrumentos transpositores, que suenan una octava por debajo de lo escrito. Las partituras para estos instrumentos deben usar la clave de Sol octava baja `"treble_8"` (o `\transposition c` para obtener una salida MIDI correcta).

Otros elementos relativos a los instrumentos de cuerda con trastes se estudian en otras partes de manual:

- Las digitaciones se indican como se explica en Sección 7.1.2 [Indicaciones de digitación], página 284.
- Las instrucciones para ligaduras *Laissez vibrer* así como ligaduras de unión sobre arpegios y trémolos están en Sección 2.1.4 [Ligaduras de unión], página 63.
- Las instrucciones para el manejo de varias voces se encuentran en Sección 5.2.3 [Resolución de las colisiones], página 229.
- Las instrucciones para la indicación de armónicos está en Sección 11.1.3 [Armónicos], página 430.

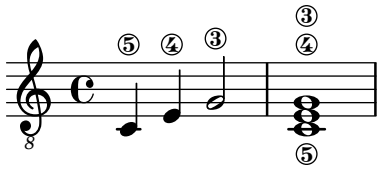
Véase también

Referencia de la notación: Sección 7.1.2 [Indicaciones de digitación], página 284, Sección 2.1.4 [Ligaduras de unión], página 63, Sección 5.2.3 [Resolución de las colisiones], página 229, Sección 6.3.1 [Nombres de instrumentos], página 265, Sección 5.2.6 [Escribir música en paralelo], página 241, Sección 3.3.2 [Arpegio], página 183, Sección B.13 [Lista de articulaciones], página 934, Sección 1.3.1 [Clave], página 20, Sección 1.3.4 [Transposición de los instrumentos], página 30.

12.1.2 Indicación de los números de cuerda

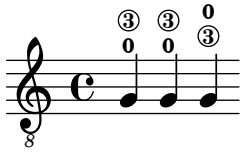
Se puede indicar la cuerda en que se debe tocar una nota añadiendo \ *número* a una nota.

```
\clef "treble_8"
c4\5 e\4 g2\3
<c\5 e\4 g\3>1
```



Cuando se usan indicaciones de cuerdas junto a digitaciones, se puede controlar su ubicación vertical con la propiedad `script-priority`, véase [Controlling the vertical ordering of scripts], página 158, y Sección B.18 [Valores por omisión para `script-priority`], página 944; el orden en el código fuente no influye.

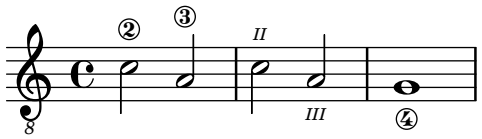
```
\clef "treble_8"
g4\3-0
g-0\3
g-\tweak script-priority 200 -0 \3
```



Los números de cuerda también pueden (así como en los instrumentos sin trastes, donde es obligatorio) imprimirse en números romanos y colocarse bajo el pentagrama, en lugar de encima de él.

```
\clef "treble_8"
c'2\2
a\3
\romanStringNumbers
```

```
c'\2
\set stringNumberOrientations = #'(down)
a\3
\arabicStringNumbers
g1\4
```



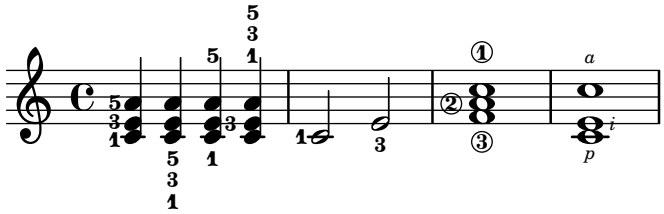
Fragmentos de código seleccionados

Controlar la colocación de las digitaciones de acordes

Se puede controlar con precisión la colocación de los números de digitación usando la propiedad `fingeringOrientation`. Para que se tenga en cuenta la orientación de las digitaciones, la instrucción de digitación se debe utilizar dentro de una construcción de acorde (`<...>`) aunque sea una sola nota. Se puede controlar la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar usando las propiedades `stringNumberOrientation` y `strokeFingerOrientation`, respectivamente.

Estas propiedades se pueden establecer a una lista de entre uno y tres valores. Controlan si las digitaciones se pueden situar por encima (si aparece `up` en la lista), por debajo (si aparece `down`), a la izquierda (si aparece `left`), o a la derecha (si aparece `right`). Por contra, si una localización no aparece en la lista, no se coloca ninguna digitadión en ese lugar. LilyPond admite esas restricciones y trata de encontrar la mejor ubicación para las digitaciones de las notas de los acordes siguientes. Observe que `left` y `right` se excluyen mutuamente (las digitaciones se pueden colocar en un lado o en el otro, no en los dos).

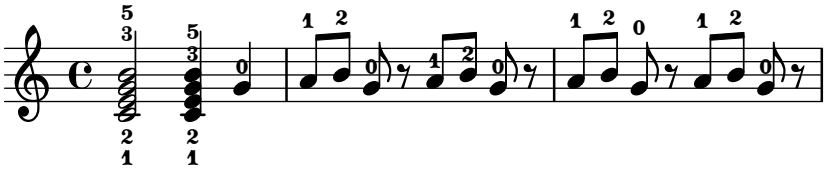
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger 1 e\rightHandFinger 2 c'\rightHandFinger 4 >
}
```



Permitir que las digitaciones se impriman dentro del pentagrama

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeteminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Instrucciones predefinidas

`\arabicStringNumbers`, `\romanStringNumbers`.

Véase también

Referencia de la notación: Sección 7.1.2 [Indicaciones de digitación], página 284.

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

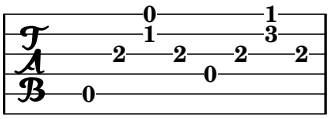
Referencia de funcionamiento interno: Sección “StringNumber” en *Referencia de Funcionamiento Interno*, Sección “Fingering” en *Referencia de Funcionamiento Interno*.

12.1.3 Tablaturas predeterminadas

La notación para instrumentos de cuerda pulsada se realiza con frecuencia utilizando un sistema de dedos y cifras, o tablatura. A diferencia de la notación tradicional, las notas no se designan mediante figuras, sino mediante números (o símbolos semejantes a letras en la intavolatura histórica). Las líneas de la tablatura indican sobre qué cuerda se debe tocar la nota, y el número sobre una línea indica el traste en que la cuerda correspondiente debe pisarse. Las notas que se deben tocar simultáneamente se alinean en sentido vertical.

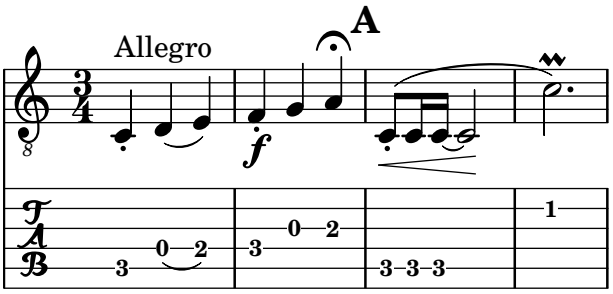
De forma predeterminada, la cuerda 1 es la más aguda y corresponde a la línea superior de la pauta `TabStaff`. La afinación predeterminada de las cuerdas del `TabStaff` es la afinación estándar de guitarra (con 6 cuerdas). Las notas se imprimen como tablatura, usando los contextos `TabStaff` y `TabVoice`. Se añade automáticamente una clave caligráfica de tablatura.

```
\new TabStaff \relative {
  a,8 a' <c e> a
  d,8 a' <d f> a
}
```



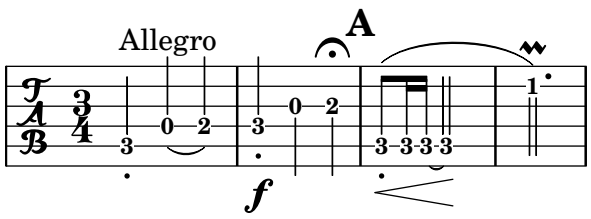
Las tablaturas predeterminadas no contienen símbolos para la duración de las notas ni ningún otro símbolo musical, como p.ej. indicaciones expresivas.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c~ 2\!
  c'2.\prall\)
}

\score {
  <<
    \new Staff { \clef "G_8" \symbols }
    \new TabStaff { \symbols }
  >>
}
```



Si queremos que todos los símbolos musicales que se usan en la notación tradicional aparezcan también en la tablatura, tenemos que aplicar la instrucción `\tabFullNotation` dentro de un contexto `TabStaff`. Tenga en cuenta que, en la tablatura, las blancas tienen doble plica para poder distinguirlas de las negras.

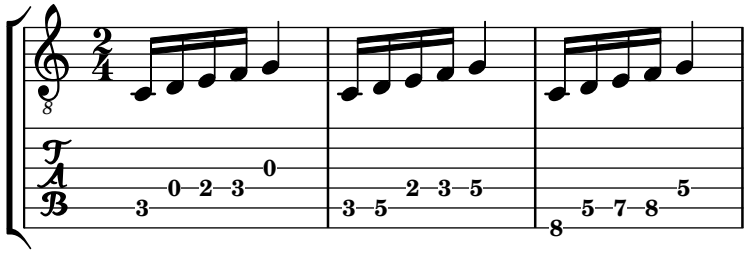
```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c~ 2\!
  c'2.\prall\)
}

\score {
  \new TabStaff {
    \tabFullNotation
    \symbols
  }
}
```



De forma predeterminada, las notas se asignan a la posición más baja del mástil (primera posición). Las cuerdas al aire se toman como preferencia, automáticamente. Si queremos que una nota determinada se toque sobre una cuerda concreta, podemos añadir una indicación de número de cuerda al nombre de la nota. Si no queremos que las indicaciones de número de cuerda aparezcan en la notación tradicional, podemos sobreescribir el sello correspondiente. Generalmente es mucho más cómodo definir la posición utilizando el valor de `minimumFret`. El valor predeterminado de es 0.

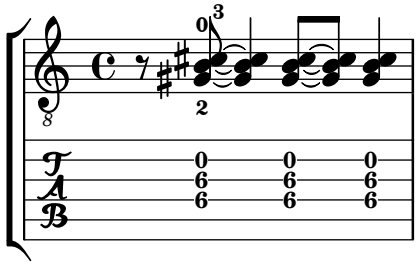
Incluso si `minimumFret` está establecido, se usan las cuerdas al aire siempre que sea posible. Este comportamiento se puede cambiar fijando `restrainOpenStrings` al valor `#t`.

```
\layout { \omit Voice.StringNumber }
\new StaffGroup <<
   \new Staff \relative {
     \clef "treble_8"
     \time 2/4
     c16 d e f g4
     c,16\5 d\5 e\4 f\4 g4\4
     c,16 d e f g4
   }
   \new TabStaff \relative {
     c16 d e f g4
     c,16\5 d\5 e\4 f\4 g4\4
     \set TabStaff.minimumFret = 5
     \set TabStaff.restrainOpenStrings = ##t
     c,16 d e f g4
   }
>>
```



Las construcciones de acorde se pueden repetir mediante el símbolo de repetición de acordes ‘q’ (véase Sección 5.1.2 [Repetición de acordes], página 220). En combinación con las tablaturas, su comportamiento de eliminar números de cuerda y de dedos junto a otros eventos puede llevar a resultados no deseados, sobre todo diferentes posiciones de los trastes. La instrucción `\tabChordRepeats` conserva consistente la digitación de una repetición a otra. En el siguiente ejemplo, la digitación predeterminada para este acorde (sin indicaciones de digitación) sería ‘sol sostenido’ sobre la cuarta cuerda, ‘si’ sobre la tercera, y ‘do sostenido’ sobre la segunda. Como usamos `b-0` en el código de entrada, ‘si’ está en la segunda cuerda, y el ‘do sostenido’ se traslada a la tercera cuerda. `\tabChordRepeats` permite mantener la misma digitación en los slguientes acordes de q:

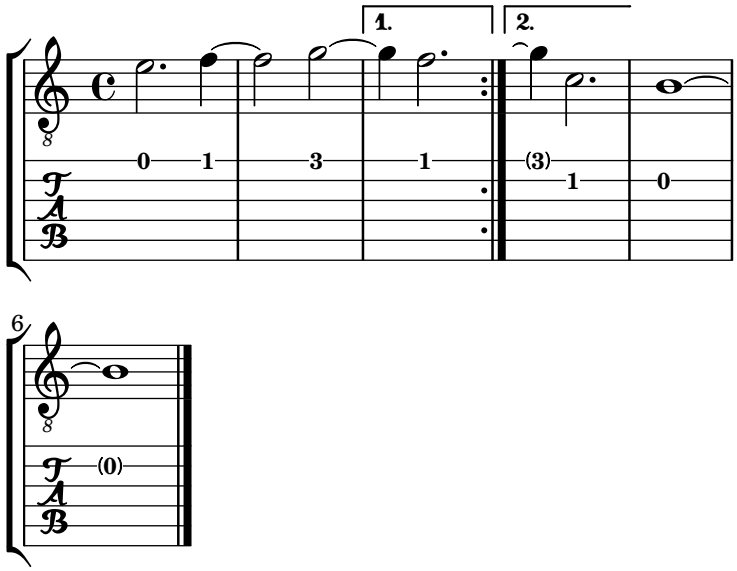
```
guitar = \relative {
  r8 <gis-2 cis-3 b-0>~ q4 q8~ 8 q4
}

\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \guitar
  }
  \new TabStaff {
    \tabChordRepeats \guitar
  }
>>
```



Las ligaduras de unión que cruzan saltos de línea llevan paréntesis de forma predeterminada. Lo mismo rige para la casilla de segunda vez de una repetición.

```
ties = \relative {
  \repeat volta 2 {
    e'2. f4~
    2 g2~
  }
  \alternative {
     \volta 1 { g4 f2. }
     \volta 2 { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|."
}

\score {
  <<
    \new StaffGroup  <<
      \new Staff {
        \clef "treble_8"
        \ties
      }
      \new TabStaff {
        \ties
      }
    >>
  >>
  \layout {
```

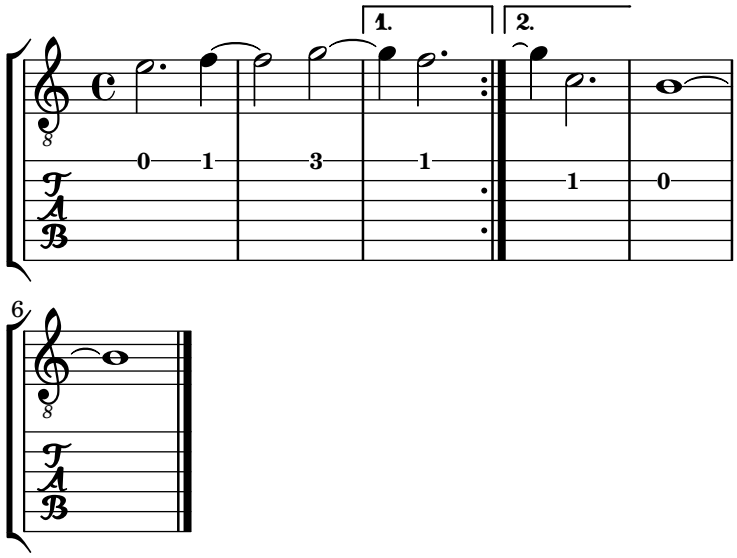
```
  indent = 0
  ragged-right = ##t
  }
}
```



La instrucción \hideSplitTiedTabNotes cancela el comportamiento de imprimir los números de traste entre paréntesis:

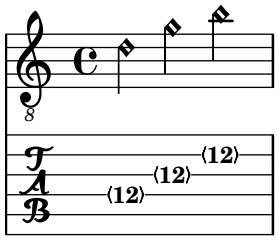
```
ties = \relative {
  \repeat volta 2 {
    e'2. f4~
    2 g2~ }
  \alternative {
    \volta 1 { g4 f2. }
    \volta 2 { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|."
}

\score {
  <<
    \new StaffGroup  <<
      \new Staff {
        \clef "treble_8"
        \ties
      }
      \new TabStaff {
      \hideSplitTiedTabNotes
        \ties
      }
    >>
  >>
  \layout {
```

```
  indent = 0
  ragged-right = ##t
  }
}
```



Se pueden añadir indicaciones de armónicos a la notación de tablatura como sus alturas de sonido:

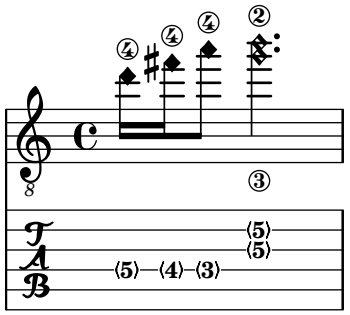
```
\layout { \omit Voice.StringNumber }
firstHarmonic = {
  d'4\4\harmonic
  g'4\3\harmonic
  b'2\2\harmonic
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \firstHarmonic
    }
    \new TabStaff { \firstHarmonic }
  >>
}
```



Observe que la instrucción \harmonic se debe adjuntar siempre a notas individuales (posiblemente dentro de un acorde) en lugar de a acordes completos. Solo tiene sentido para armónicos sobre una cuerda abierta en el duodécimo traste. Cualquier otro armónico debe estar calculado por parte de LilyPond. Se puede conseguir esto indicando el traste en que debe rozar la cuerda el dedo de la mano que pisa.

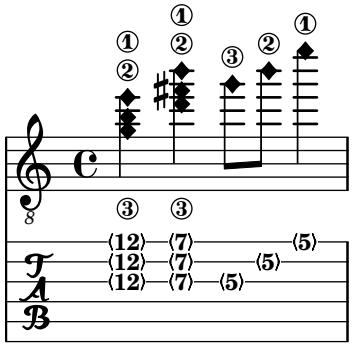
```
fretHarmonics = {
```

```
  \harmonicByFret 5 d16\4
  \harmonicByFret 4 d16\4
  \harmonicByFret 3 d8\4
  \harmonicByFret 5 <g\3 b\2>2.
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \fretHarmonics
    }
    \new TabStaff { \fretHarmonics }
  >>
}
```



De forma alternativa, se pueden calcular los armónicos definiendo la razón de las longitudes de cuerda por encima y por debajo de la digitación de armónico.

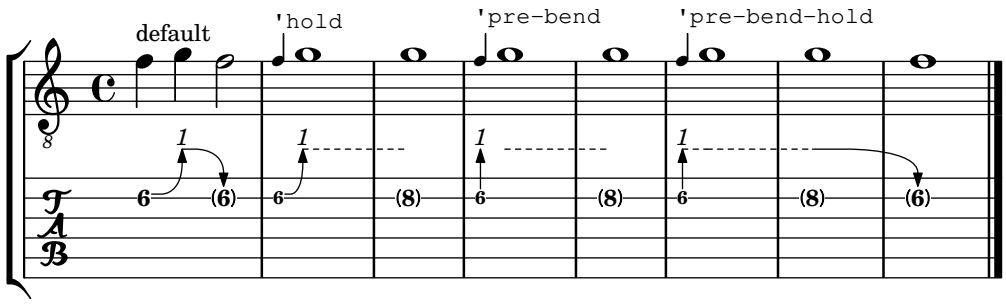
```
ratioHarmonics = {
  \harmonicByRatio #1/2 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/3 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/4 { g8\3 b8\2 e'4\1 }
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \ratioHarmonics
    }
    \new TabStaff { \ratioHarmonics }
  >>
}
```



Se pueden añadir estiramientos de cuerda o curvaturas de tono a la notación de tablatura. Se escribe una curvatura añadiendo `\^` después de la nota o acorde que recibe la curvatura; termina automáticamente en la siguiente nota o acorde. Están disponibles los siguientes estilos: el predeterminado imprime una curva con una punta de flecha hacia arriba o hacia abajo, `'hold` una línea discontinua horizontal, `'pre-bend` una línea vertical con punta de flecha, y `'pre-bend-hold` una línea vertical con flecha que se continúa con una línea discontinua.

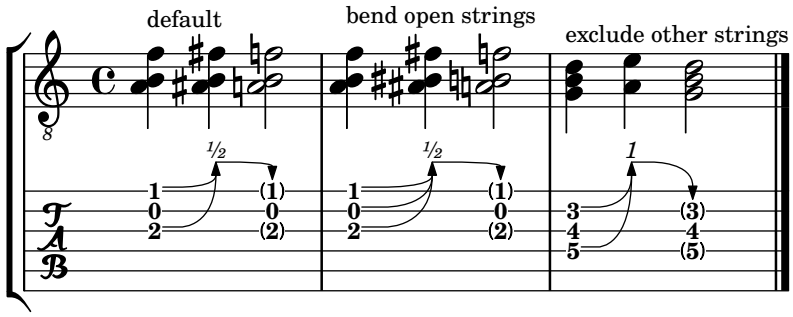
Las instrucciones `\bendHold`, `\preBend` y `\preBendHold` son atajos para la selección del estilo de las curvaturas.

```
bend-styles = {
  <>^"default"
  f'4\^ g'4\^ f'2

  <>^\markup \typewriter "'hold"
  \grace f'4\^ g'1\bendHold \^ g'1

  <>^\markup \typewriter "'pre-bend"
  \grace f'4\preBend \^ g'1\bendHold \^ g'1

  <>^\markup \typewriter "'pre-bend-hold"
  \grace f'4\preBendHold \^ g'1\bendHold \^ g'1\^ f'

  \bar "|."
}

\score {
  \new StaffGroup
  <<
    \new Staff {
      \override TextScript.font-size = -2
      \clef "G_8"
      \bend-styles
    }
    \new TabStaff \bend-styles
  >>
  \layout {
    \context {
      \Voice
      \omit StringNumber
    }
    \context {
      \TabStaff
      minimumFret = 5
    }
  }
}
```



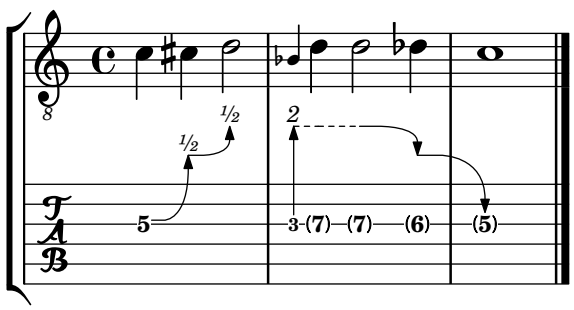
Las cuerdas al aire normalmente no admiten la curvatura de tono. Para que se puede hacer curvatura con una cuerda al aire, se fija la propiedad `bend-me` al valor `#t`. Para que otras notas queden excluidas de la curvatura, se fija al valor `#f`.

```
mus = {
  <>^"default"
  <a b f'>4\^
  <ais b fis'>\^
  <a b f'>2

  <>^"bend open strings"
  <a \tweak bend-me ##t b f'>4\^
  <ais \tweak bend-me ##t bis fis'>\^
  <a b f'>2

  <>^"exclude other strings"
  <g \tweak bend-me ##f b\3 d'>4\^
  <a e'\2 >\^
  <g \tweak bend-me ##f b\3 d'>2

  \bar "|."
}

\score {
  \new StaffGroup
  <<
    \new Staff {
      \override TextScript.font-size = -2
      \clef "G_8"
      \mus
    }
    \new TabStaff \mus
  >>
  \layout {
    \context {
      \Voice
      \omit StringNumber
    }
  }
}
```



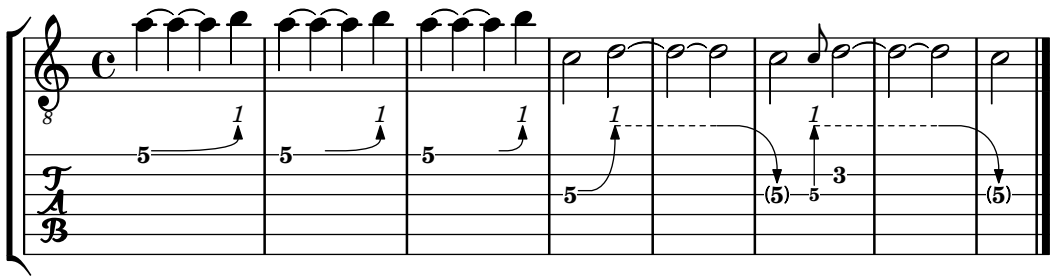
Para hacer varias curvaturas consecutivas, la primera puede necesitar un ajuste en `details.successive-level`. Por conveniencia, existe la función `bendStartLevel`, que admite un número entero.

```
printNext = -\tweak details.target-visibility ##t \etc

mus = {
  c'4\3\^ cis'\3 \^ d'2\3

  \grace bes4\3\preBendHold \bendStartLevel 2 \printNext \^
  d'4\3\bendHold \^ d'2\3\^ des'4\3 \^ c'1\3

  \bar "|."
}

\score {
  \new StaffGroup
  <<
    \new Staff {
      \override TextScript.font-size = -2
      \clef "G_8"
      \mus
    }
    \new TabStaff \mus
  >>
  \layout {
    \context {
      \Voice
      \omit StringNumber
    }
  }
}
```



Por omisión, el elemento de extensión de la curvatura BendSpanner termina en la siguiente nota o acorde incluso si está ligada a la nota o acorde inicial. Se puede saltar una sola columna NoteColumn usando \skipNC. Se puede saltar un grupo de columnas NoteColumn usando \skipNCs al principio y \endSkipNCs al final.

```
bends-with-ties-and-skips = {
  a'4~\^ \skipNC a'4~ \skipNC a'4 b'4
  a'4~ a'4~\^ \skipNC a'4 b'4
  a'4~ a'4~ a'4\^ b'4
  c'2\^ d'~ \bendHold \^ \skipNC d'~ d'\^ c'
  \grace { c'8-\preBendHold \^ }
  \skipNCs d'2~ d'2~ \endSkipNCs d'\^ c'2
  \bar "|."
}

\score {
  \new StaffGroup
  <<
    \new Staff {
      \clef "G_8"
      \bends-with-ties-and-skips
    }
    \new TabVoice \bends-with-ties-and-skips
  >>
  \layout {
    \context {
      \Voice
      \omit StringNumber
    }
    \context {
      \TabStaff
      minimumFret = 3
      restrainOpenStrings = ##t
    }
  }
}
```



Instrucciones predefinidas

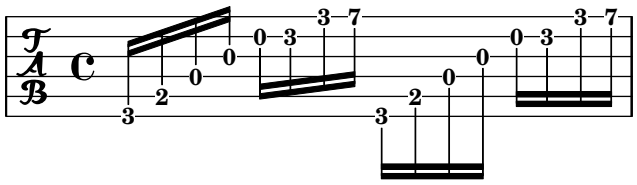
\skipNCs, \skipNC, \endSkipNCs.

Fragmentos de código seleccionados

Comportamiento de las plicas y las barras de corchea en tablaturas

La dirección de las plicas se controla de la misma forma en la tablatura que en la notación tradicional. Las barras se pueden poner horizontales, como se muestra en este ejemplo.

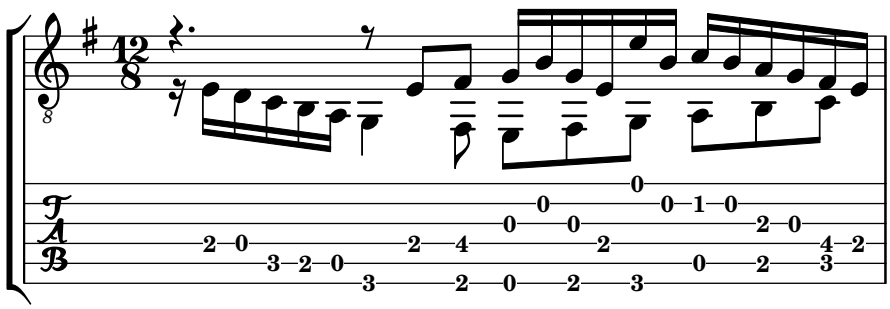
```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam.concaveness = 10000
    g,,16 b d g b d g b
  }
}
```



Polifonía en tablaturas

La polifonía se crea de la misma forma en un TabStaff que en una pauta normal.

```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  \new StaffGroup = "tab with traditional" <<
    \new Staff = "guitar traditional" <<
      \clef "treble_8"
      \new Voice = "upper" \upper
      \new Voice = "lower" \lower
    >>

    \new TabStaff = "guitar tab" <<
      \new TabVoice = "upper" \upper
      \new TabVoice = "lower" \lower
    >>
  >>
}
```

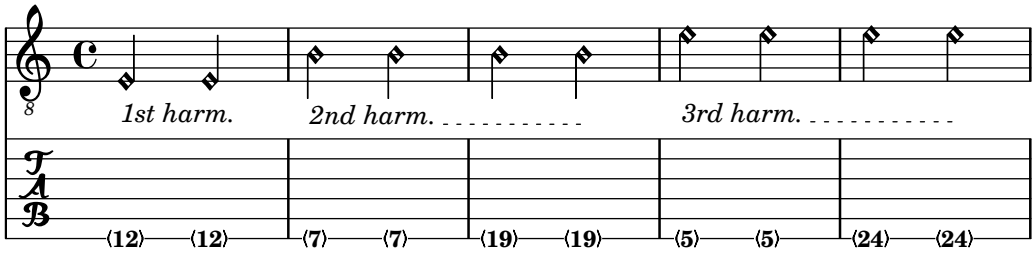


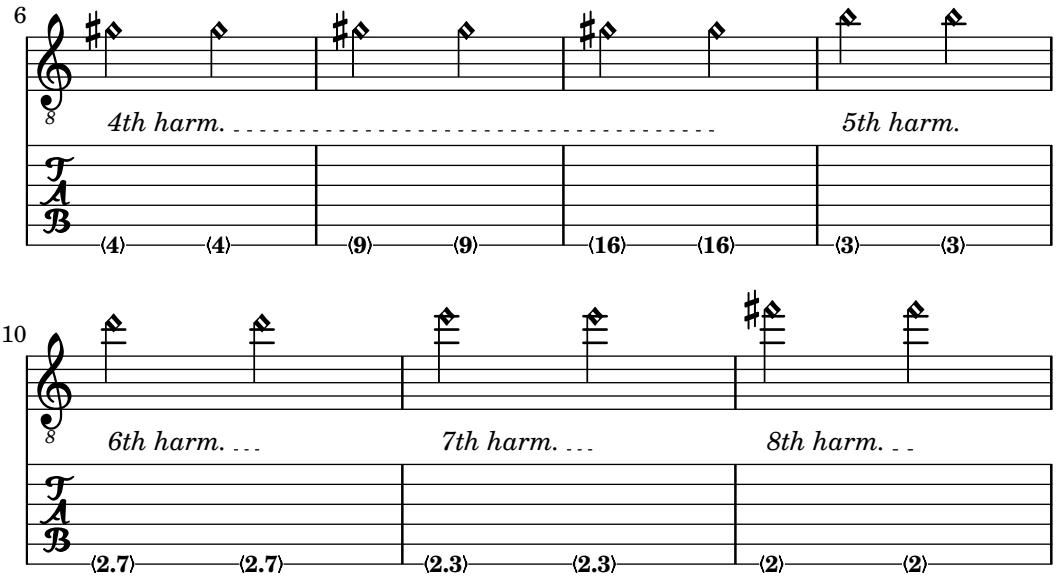
Referencia para armónicos sobre cuerdas al aire

Referencia para armónicos sobre cuerdas al aire (armónicos naturales).

```
openStringHarmonics = {
  \textSpannerDown
  \override TextSpanner.staff-padding = 3
  \override TextSpanner.dash-fraction = 0.3
  \override TextSpanner.dash-period = 1

  % first harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "1st harm. "
  \harmonicByFret 12 e,2\6\startTextSpan
  \harmonicByRatio #1/2 e,\6\stopTextSpan

  % second harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "2nd harm. "
  \harmonicByFret 7 e,\6\startTextSpan
  \harmonicByRatio #1/3 e,\6
  \harmonicByFret 19 e,\6
  \harmonicByRatio #2/3 e,\6\stopTextSpan
  %\harmonicByFret 19 < e,\6 a,\5 d\4 >
  %\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >

  % third harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "3rd harm. "
  \harmonicByFret 5 e,\6\startTextSpan
  \harmonicByRatio #1/4 e,\6
  \harmonicByFret 24 e,\6
  \harmonicByRatio #3/4 e,\6\stopTextSpan
  \break

  % fourth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "4th harm. "
  \harmonicByFret 4 e,\6\startTextSpan
  \harmonicByRatio #1/5 e,\6
  \harmonicByFret 9 e,\6
  \harmonicByRatio #2/5 e,\6
  \harmonicByFret 16 e,\6
  \harmonicByRatio #3/5 e,\6\stopTextSpan
```

```
  % fifth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "5th harm. "
  \harmonicByFret 3 e,\6\startTextSpan
  \harmonicByRatio #1/6 e,\6\stopTextSpan
  \break

  % sixth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "6th harm. "
  \harmonicByFret 2.7 e,\6\startTextSpan
  \harmonicByRatio #1/7 e,\6\stopTextSpan

  % seventh harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "7th harm. "
  \harmonicByFret 2.3 e,\6\startTextSpan
  \harmonicByRatio #1/8 e,\6\stopTextSpan

  % eighth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "8th harm. "
  \harmonicByFret 2 e,\6\startTextSpan
  \harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score {
  <<
    \new Staff \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \openStringHarmonics
      }
    }
  >>
}
```



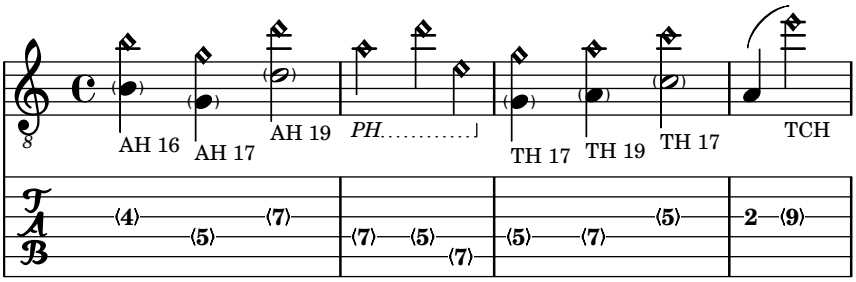


Armónicos sobre cuerdas pisadas en tablatura

El código que aparece a continuación muestra armónicos sobre cuerdas pisadas (armónicos artificiales) en una tablatura.

```
pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner.bound-details.left.text =
    \markup {\halign #-0.5 \teeny "PH" }
  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.dash-period = 0.6
  \override TextSpanner.bound-details.right.attach-dir = 1
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.padding = -0.5
}

harmonics = {
  % artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup { \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4_\markup { \teeny "AH 17" }
  <\parenthesize d' d''\harmonic>2_\markup { \teeny "AH 19" }

  % pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d''\harmonic>4
  <e'\harmonic>4\stopTextSpan

  % tapped harmonics (TH)
  <\parenthesize g\4 g'\harmonic>4_\markup { \teeny "TH 17" }
  <\parenthesize a\4 a'\harmonic>4_\markup { \teeny "TH 19" }
  <\parenthesize c'\3 c''\harmonic>2_\markup { \teeny "TH 17" }

  % touch harmonics (TCH)
```

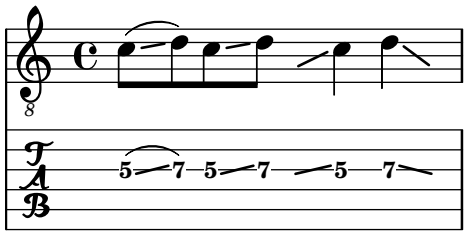
```
  a4( <e''\harmonic>2. )_\markup { \teeny "TCH" }
}

frettedStrings = {
  % artificial harmonics (AH)
  \harmonicByFret 4 g4\3
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 g2\3

  % pinched harmonics (PH)
  \harmonicByFret 7 d2\4
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 a4\5

  % tapped harmonics (TH)
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 d4\4
  \harmonicByFret 5 g2\3

  % touch harmonics (TCH)
  a4 \harmonicByFret 9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}
```



Deslizamientos en tablatura

Los deslizamientos se pueden componer tipográficamente tanto en los contextos de `Staff` como en los de `TabStaff`.

```
slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
  \stemDown \hideNotes
  g16 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>

  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = 4
      \override Glissando.springs-and-rods =
                  #ly:spanner::set-spacing-rods
      \override Glissando.thickness = 2
      \omit StringNumber
      % or:
      %\override StringNumber.stencil = ##f
    }
  }
}
```



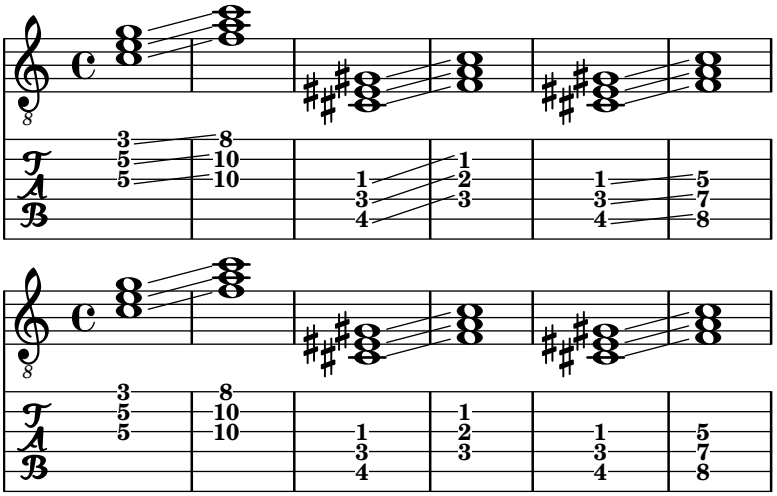
Glissando de acordes en tablatura

Los deslizamientos para acordes se indican por omisión tanto en el contexto `Staff` como en `TabStaff`.

Los números de cuerda son necesarios para TabStaff porque los cálculos de cuerda automáticos son diferentes para los acordes y para notas sueltas.

```
myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
  <cis, eis gis>1 \glissando <f a c>
  <cis eis gis>1 \glissando <f a c\3>
}
```

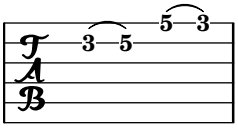
```
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \myMusic
  >>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \with { \override Glissando.style = #'none } {
      \myMusic
    }
  >>
}
```



Ligados ascendentes y descendentes (hammer-on y pull-off)

Se pueden obtener ligados ascendentes o de entrada y ligados descendentes o de salida (hammer-on y pull-off) usando ligaduras de expresión.

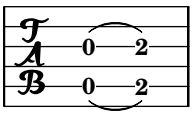
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Indicaciones de hammer-on y pull-off usando voces

El arco del ligado ascendente o de entrada (hammer-on) y descendente o de salida (pull-off) apunta hacia arriba en las voces uno y tres y hacia abajo en las voces dos y cuatro:

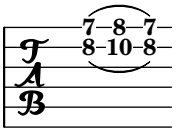
```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Indicaciones de ligado ascendente y descendente (hammer-on y pull-off) usando acordes

Cuando se usan ligados ascendentes o de entrada (*hammer-on* en inglés) o tirones de salida (*pull-off* en inglés) con notas en acorde, se dibuja un solo arco. Sin embargo son posibles "dobles arcos" si se fija la propiedad `doubleSlurs` a `#t`.

```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Véase también

Referencia de la notación: Sección 5.1.2 [Repetición de acordes], página 220, Sección 4.1.1 [Repeticiones explícitas], página 191, Sección 7.1.9 [Plicas], página 300, Sección 11.1.3 [Armónicos], página 430, Sección 3.3.1 [Glissando], página 178.

Fragmentos de código: Sección "Instrumentos de cuerda con trastes" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "TabNoteHead" en *Referencia de Funcionamiento Interno*, Sección "TabStaff" en *Referencia de Funcionamiento Interno*, Sección "TabVoice" en *Referencia de Funcionamiento Interno*, Sección "Beam" en *Referencia de Funcionamiento Interno*.

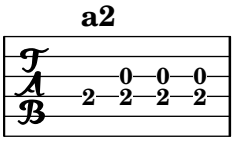
Advertencias y problemas conocidos

Los acordes no se tratan de una forma especial, y de aquí que el selector automático de la cuerda puede elegir fácilmente la misma cuerda para dos notas del acorde.

Para manejar `\partCombine`, es necesario que `TabStaff` utilice voces especialmente creadas:

```
melodia = \partCombine { e4 g g g } { e4 e e e }
<<
  \new TabStaff <<
```

```
    \new TabVoice = "one" s1
    \new TabVoice = "two" s1
    \new TabVoice = "shared" s1
    \new TabVoice = "solo" s1
    { \melodia }
  >>
>>
```



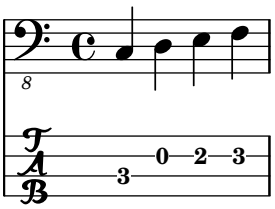
Los efectos especiales de guitarra se limitan a armónicos y *slides*.

12.1.4 Tablaturas personalizadas

La tablatura en LilyPond calcula automáticamente el traste para cada nota, basándose en la cuerda a que está asignada la nota. Para hacerlo, es necesario especificar la afinación de las cuerdas. La afinación de las cuerdas se da en la propiedad `stringTunings`.

LilyPond trae afinaciones predefinidas para el banjo, la mandolina, la guitarra, el bajo, el ukelele, el violín, la viola, el violoncello y el contrabajo. LilyPond establece automáticamente la transposición correcta para las afinaciones predefinidas. El ejemplo siguiente es para bajo, que suena una octava por debajo de lo escrito.

```
<<
  \new Voice \with {
    \omit StringNumber
  } {
    \clef "bass_8"
    \relative {
      c,4 d e f
    }
  }
  \new TabStaff \with {
    stringTunings = #bass-tuning
  } {
    \relative {
      c,4 d e f
    }
  }
>>
```

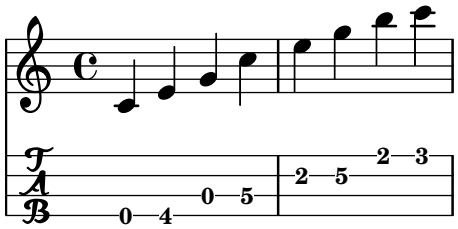


La afinación por omisión es `guitar-tuning` (la afinación estándar Mi-La-Re-Sol-Si-Mi). Otras afinaciones predefinidas son `guitar-open-g-tuning` (Sol Mayor al aire, Re-Sol-Re-Sol-Si-Re), `mandolin-tuning` (mandolina) y `banjo-open-g-tuning` (banjo con Sol Mayor al aire). Las afinaciones predefinidas están en `ly/string-tunings-init.ly`.

Es posible crear cualquier afinación de cuerdas que se desee. Se puede usar la función `\stringTuning` para definir una afinación de cuerdas y que puede usarse para establecerla como el valor de `stringTunings` para el contexto actual.

Su argumento es una construcción de acorde que define la altura de las notas de cada cuerda al aire en dicha afinación. La construcción de acorde debe estar en el modo de octava absoluto, véase Sección 1.1.1 [Escritura de octava absoluta], página 3. La cuerda con el número más alto (generalmente la más grave) debe ir primero en el acorde. Por ejemplo, podemos definir una afinación para un instrumento de cuatro cuerdas con las alturas `a''`, `d''`, `g'` y `c'`:

```
mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
  \new Staff {
    \clef treble
    \mynotes
  }
  \new TabStaff {
    \set Staff.stringTunings = \stringTuning <c' g' d'' a''>
    \mynotes
  }
>>
```



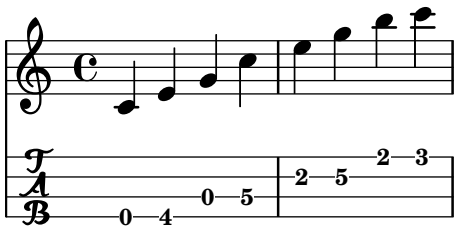
La propiedad `stringTunings` también se utiliza por parte de `FretBoards` para calcular los diagramas de posición automáticos.

Las afinaciones de cuerdas se utilizan como parte de la clave de asociación para los diagramas de posición predefinidos (véase Sección 12.1.6 [Diagramas predefinidos de trastes], página 469).

El ejemplo anterior también se podría haber escrito como sigue:

```
custom-tuning = \stringTuning <c' g' d'' a''>

mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
  \new Staff {
    \clef treble
    \mynotes
  }
  \new TabStaff {
    \set TabStaff.stringTunings = #custom-tuning
    \mynotes
```

```
  }
>>
```



Internamente, una afinación de las cuerdas es una lista de Scheme de alturas, una por cada cuerda, ordenadas de 1 a N, donde la cuerda 1 está arriba en la pauta de tablatura y la cuerda N está abajo. Esto resulta generalmente en un orden de la nota más aguda a la más grave, pero algunos instrumentos (p.ej. el ukelele) no tienen las cuerdas dispuestas en orden de altura.

La altura de una cuerda dentro de una lista de alturas de cuerdas es un objeto altura de LilyPond. Los objetos altura se crean con la función de Scheme `ly:make-pitch` (véase Sección "Funciones de Scheme" en *Referencia de Funcionamiento Interno*).

`\stringTuning` crea dicho objeto a partir de la entrada de acordes.

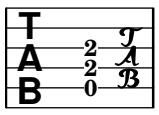
LilyPond calcula automáticamente el número de líneas dentro del `TabStaff` y el número de cuerdas dentro de un `FretBoard` calculado automáticamente como el número de elementos de `stringTunings`.

Para permitir a todos los contextos `TabStaff` usar la misma afinación personalizada por omisión, podemos usar

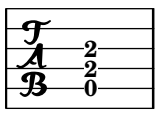
```
\layout {
  \context {
    \TabStaff
    stringTunings = \stringTuning <c' g' d'' a''>
  }
}
```

También se puede usar una clave moderna de tablatura.

```
\new TabStaff {
  \clef moderntab
  <a, e a>1
  \break
  \clef tab
  <a, e a>1
}
```



2

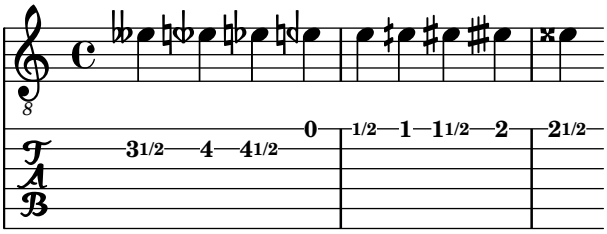


La clave moderna de tablatura contempla tablaturas de 4 a 7 cuerdas.

Ua pauta de tablatura `TabStaff` puede contener intervalos microtonales como cuartos de tono, que se pueden reproducir mediante eventos de la rueda de cambio de tono. Es necesario fijar el

valor supportNonIntegerFret = ##t dentro del contexto Score. Sin embargo, los microtonos no están contemplados en los diagramas FretBoards.

```
\layout {
  \context {
    \Score
    supportNonIntegerFret = ##t
  }
}

custom-tuning = \stringTuning <e, a, d ges beh eeh'>

mus = \relative {
  eeses'4
  eeseh
  ees
  eeh
  e
  eih
  eis
  eisih
  eisis
}

<<
  \new Staff << \clef "G_8" \mus >>
  \new TabStaff \with { stringTunings = \custom-tuning } \mus
>>
```



Véase también

Referencia de la notación: Sección 1.1.1 [Escritura de octava absoluta], página 3, Sección 12.1.6 [Diagramas predefinidos de trastes], página 469.

Archivos instalados: ly/string-tunings-init.ly, scm/tablature.scm.

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Tab_note_heads_engraver” en *Referencia de Funcionamiento Interno*, Sección “Funciones de Scheme” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

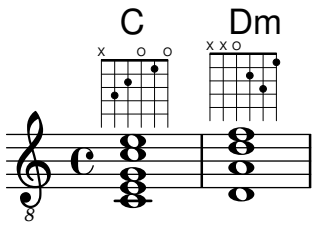
Los cálculos de tablatura automáticos no funcionan bien en la mayoría de los casos para instrumentos en que la afinación de las cuerdas no varía de forma monótona con el número de cuerda, como el ukelele.

12.1.5 Marcas de diagramas de trastes

Se pueden añadir diagramas de trastes a la música como elementos de marcado sobre la nota deseada. El marcado contiene información sobre el diagrama de trastes deseado. Existen tres interfaces distintos de marcado de diagramas de trastes: standard (estándar), terse (escueto) y verbose (prolijo). Los tres interfaces producen marcados equivalentes, pero tienen cantidades variables de información en la cadena de marcado. Hay más detalles sobre la sintaxis de las distintos cadenas de marcado que se usan para definir diagramas de posiciones en Sección A.1.6 [Instrument-specific markup], página 869.

La cadena de marcado de los diagramas estándar de trastes indica el número de la cuerda y el número de traste en que se coloca cada uno de los puntos sobre la cuerda. Además se pueden indicar cuerdas al aire y cuerdas mudas (que no se tocan).

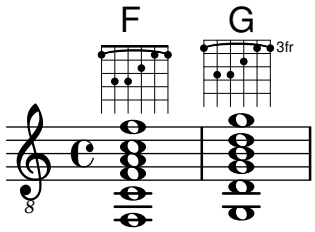
```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram "6-x;5-3;4-2;3-o;2-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram "6-x;5-x;4-o;3-2;2-3;1-1;"
    }
  }
>>
```



Las indicaciones de cejilla se pueden añadir al diagrama a partir de la cadena de marcado del diagrama de trastes.

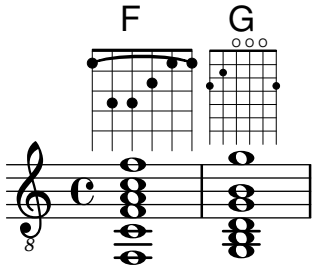
```
<<
  \new ChordNames {
     \chordmode {
       f1 g
     }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram "c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram "c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
    }
```

```
  }
>>
```



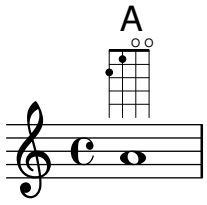
El tamaño del diagrama de trastes y el número de trastes del diagrama, se puede cambiar en la cadena de marcado `fret-diagram`.

```
<<
  \new ChordNames {
     \chordmode {
       f1 g
     }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram "s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram "h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
    }
  }
>>
```



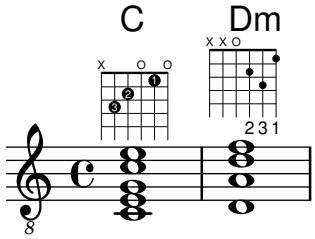
Se puede cambiar el número de cuerdas de un diagrama de trastes para que se adapte a distintos instrumentos como el bajo y el ukelele, con la cadena de marcado del diagrama.

```
<<
  \new ChordNames {
    \chordmode {
      a1
    }
  }
  \new Staff {
    % An 'A' chord for ukulele
    a'1^\markup {
      \fret-diagram "w:4;4-2-2;3-1-1;2-o;1-o;"
    }
  }
>>
```



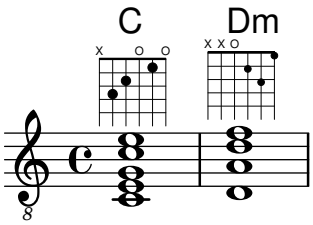
Se pueden añadir indicaciones de digitación, y la posición de las etiquetas de los dedos se puede controlar mediante la cadena de marcado del diagrama.

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram "f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram "f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
    }
  }
>>
```



Se puede controlar el radio y la posición de los puntos con la cadena de marcado `fret-diagram`.

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram "d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram "p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
    }
  }
>>
```



Se pueden imprimir diagramas de trastes de acordes invertidos para personas zurdas.

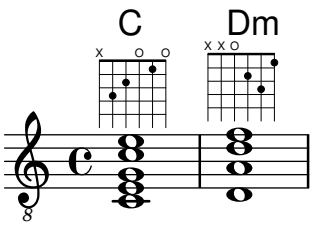
```
\markup
  \center-column {
    "C"
    "(left-handed)"
     \override #`(fret-diagram-details . ((handedness . ,LEFT)))
     \fret-diagram "6-x;5-3-3;4-2-2;3-o;2-1;1-o;"
  }
```

C
(left-handed)



La cadena de marcado de diagrama de trastes escueta, `\fret-diagram-terse`, omite los números de la cuerda; el número de cuerda viene implícito por la presencia del punto y coma. Hay un punto y coma por cada cuerda del diagrama. El primer punto y coma corresponde al número de cuerda más alto, y el último punto y coma corresponde a la primera cuerda. Se pueden indicad cuerdas mudas, al aire y números de traste.

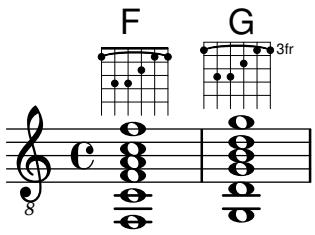
```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-terse "x;3;2;o;1;o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram-terse "x;x;o;2;3;1;"
    }
  }
>>
```



Se pueden incluir números de cejilla en la cadena de marcado escueta `\fret-diagram-terse`.

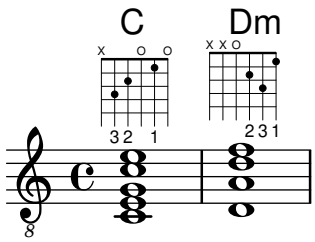
```
<<
```

```
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram-terse "1-(;3;3;2;1;1-);"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram-terse "3-(;5;5;4;3;3-);"
    }
  }
>>
```



Se pueden incluir indicaciones de digitación en la cadena de marcado escueta `\fret-diagram-terse`

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \override Voice.TextScript.fret-diagram-details.finger-code =
      #'below-string
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-terse "x;3-3;2-2;o;1-1;o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram-terse "x;x;o;2-2;3-3;1-1;"
    }
  }
>>
```

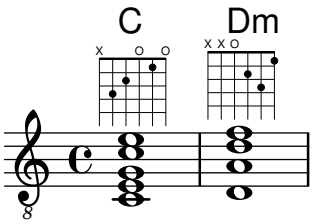


Otras propiedades de los diagramas de trastes se deben ajustar usando `\override` al utilizar el marcado escueto con `\fret-diagram-terse`.

Solo se puede incluir una indicación por cada cuerda en un marcado del tipo `\fret-diagram-terse`. Para tener más de una indicación por cuerda, utilice un marcado de diagrama normal o del tipo prolijo.

La cadena de marcado prolija `\fret-diagram-verbose` está en el formato de una lista de Scheme. Cada elemento de la lista indica la colocación un elemento sobre el diagrama de trastes.

```
<<
    \new ChordNames {
      \chordmode {
        c1 d:m
      }
    }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (place-fret 5 3)
        (place-fret 4 2)
        (open 3)
        (place-fret 2 1)
        (open 1)
      )
    }
    <d a d' f'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (mute 5)
        (open 4)
        (place-fret 3 2)
        (place-fret 2 3)
        (place-fret 1 1)
      )
    }
  }
>>
```



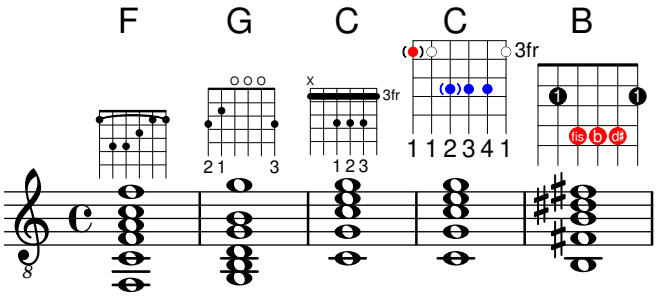
Se pueden incluir digitaciones y cejillas en una cadena de marcado prolija `\fret-diagram-verbose`. Es exclusiva del interfaz de `\fret-diagram-verbose` la indicación 'capo' que se puede colocar sobre el diagrama de posición. La indicación de capo es una barra gruesa que cubre todas las cuerdas. El traste que tiene el capo es el más bajo del diagrama de posición.

Los puntos que indican las digitaciones se pueden tanto colorear como imprimir entre paréntesis; asimismo, es posible alterar independientemente el color de los paréntesis.

También se pueden insertar elementos de marcado dentro de los puntos.

```
<<
    \new ChordNames {
      \chordmode {
        f1 g c c b
      }
    }
  \new Staff {
    \clef "treble_8"
    \override Voice.TextScript
      .fret-diagram-details.finger-code = #'below-string
    <f, c f a c' f'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 1)
        (place-fret 5 3)
        (place-fret 4 3)
        (place-fret 3 2)
        (place-fret 2 1)
        (place-fret 1 1)
        (barre 6 1 1)
      )
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 3 2)
        (place-fret 5 2 1)
        (open 4)
        (open 3)
        (open 2)
        (place-fret 1 3 3)
      )
    }
    <c g c' e' g'>1^\markup {
      \fret-diagram-verbose #'(
        (capo 3)
        (mute 6)
        (place-fret 4 5 1)
        (place-fret 3 5 2)
        (place-fret 2 5 3)
      )
    }
    \override Voice.TextScript.size = 1.4
    <c g c' e' g'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 3 1 red parenthesized default-paren-color)
        (place-fret 5 3 1 inverted)
        (place-fret 4 5 2 blue parenthesized)
        (place-fret 3 5 3 blue)
        (place-fret 2 5 4 blue)
        (place-fret 1 3 1 inverted)
      )
```

```
    }
    \override Voice.TextScript.size = 1.5
    <b, fis b dis' fis'>1^\markup
      \override #'(fret-diagram-details . ((finger-code . in-dot)))
      \fret-diagram-verbose #`(
        (place-fret 5 2 1)
        (place-fret 4 4 "fis" red)
        (place-fret 3 4 "b" red)
        (place-fret
          2 4
          ,#{ \markup
                \concat {
                  \vcenter "d"
                  \fontsize #-5
                  \musicglyph "accidentals.sharp"} #}
          red)
        (place-fret 1 2 1)
      )
  }
>>
```



Todas las otras propiedades de diagrama de trastes se deben ajustar utilizando `\override` cuando se usa el marcado prolijo con `\fret-diagram-verbose`.

El aspecto gráfico de un diagrama de trastes se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*; consulte Sección “Fret diagrams explained and developed” en *Fragmentos de código* para ver un ejemplo exhaustivo. Para un marcado de diagrama de trastes, las propiedades del interface pertenecen a `Voice.TextScript`.

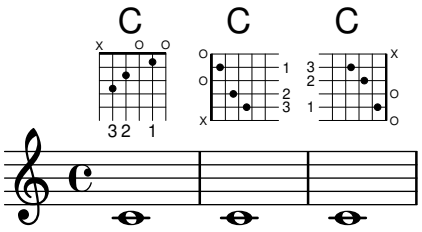
Fragmentos de código seleccionados

Modificar la orientación de los trastes

Los diagramas de posiciones de acordes se pueden orientar de tres formas. De manera predeterminada se alinena la cuerda o traste superior en las distintas alineaciones.

```
\include "predefined-guitar-fretboards.ly"

<<
  \chords {
    c1
    c1
    c1
  }
```

```
  \new FretBoards \chordmode {
    c1
    \override FretBoard.fret-diagram-details.orientation =
      #'landscape
    c1
    \override FretBoard.fret-diagram-details.orientation =
      #'opposing-landscape
    c1
  }
  \new Voice {
    c'1
    c'1
    c'
  }
>>
```

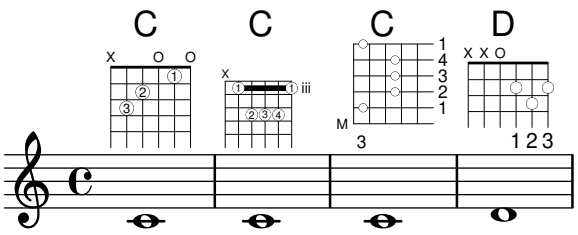


Personalizar diagramas de posiciones de marcado

Se pueden modificar las propiedades de los diagramas de posiciones estableciendo el valor de la propiedad `fret-diagram-details`. Para los diagramas de posiciones de marcado, se pueden aplicar overrides (sobreescrituras) al objeto `Voice.TextScript` o directamente al elemento de marcado.

```
<<
  \chords { c1 | c | c | d }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = 1.2
    \override TextScript.fret-diagram-details.finger-code = #'in-dot
    \override TextScript.fret-diagram-details.dot-color = #'white

    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

    %% C major for guitar, barred on third fret
       % verbose style
       % size 1.0
       % roman fret label, finger labels below string, straight barre
    c'1^\markup {
      % standard size
      \override #'(size . 1.0) {
        \override #'(fret-diagram-details . (
                     (number-type . roman-lower)
```

```
                    (finger-code . in-dot)
                    (barre-type . straight))) {
          \fret-diagram-verbose #'((mute 6)
                                   (place-fret 5 3 1)
                                   (place-fret 4 5 2)
                                   (place-fret 3 5 3)
                                   (place-fret 2 5 4)
                                   (place-fret 1 3 1)
                                   (barre 5 1 3))
        }
      }
    }

    %% C major for guitar, barred on third fret
       % verbose style
       % landscape orientation, arabic numbers, M for mute string
       % no barre, fret label down or left, small mute label font
    c'1^\markup {
      \override #'(fret-diagram-details . (
                  (finger-code . below-string)
                  (number-type . arabic)
                  (label-dir . -1)
                  (mute-string . "M")
                  (orientation . landscape)
                  (barre-type . none)
                  (xo-font-magnification . 0.4)
                  (xo-padding . 0.3))) {
        \fret-diagram-verbose #'((mute 6)
                                 (place-fret 5 3 1)
                                 (place-fret 4 5 2)
                                 (place-fret 3 5 3)
                                 (place-fret 2 5 4)
                                 (place-fret 1 3 1)
                                 (barre 5 1 3))
      }
    }

    %% simple D chord
       % terse style
       % larger dots, centered dots, fewer frets
       % label below string
    d'1^\markup {
      \override #'(fret-diagram-details . (
                  (finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



Véase también

Referencia de la notación: Sección A.1.6 [Instrument-specific markup], página 869.

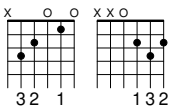
Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*.

12.1.6 Diagramas predefinidos de trastes

Se pueden imprimir los diagramas de trastes usando el contexto `FretBoards`. De forma predeterminada, el contexto `FretBoards` imprime diagramas de trastes que están almacenados en una tabla de búsqueda:

```
\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode {
    c1 d
  }
}
```

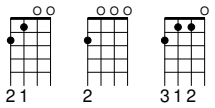


Los diagramas de traste predefinidos están en el archivo `predefined-guitar-fretboards.ly`. Los diagramas de trastes se almacenan tomando como base las notas de un acorde y el valor de `stringTunings` que se está utilizando en cada momento. `predefined-guitar-fretboards.ly` contiene diagramas de traste predefinidos solo para `guitar-tuning`. Los diagramas de traste predefinidos se pueden añadir para otros instrumentos u otras afinaciones siguiendo los ejemplos que aparecen en `predefined-guitar-fretboards.ly`.

Los diagramas de posiciones para el ukelele están en el archivo `predefined-ukulele-fretboards.ly`.

```
\include "predefined-ukulele-fretboards.ly"

myChords = \chordmode { a1 a:m a:aug }

\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #ukulele-tuning
  \myChords
}
```

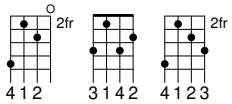




Los diagramas de posiciones para la mandolina están contenidos en el archivo `predefined-mandolin-fretboards.ly`.

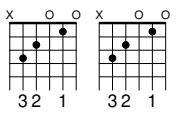
```
\include "predefined-mandolin-fretboards.ly"

myChords = \chordmode { c1 c:m7.5- c:aug }

\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #mandolin-tuning
  \myChords
}
```

C C^{ø} C+



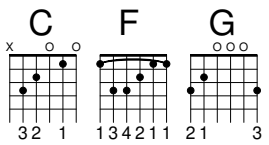
Las notas de los acordes se pueden introducir como música simultánea o bien usando el modo de acordes (véase Sección 15.1.1 [Panorámica del modo de acordes], página 510).

```
\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode { c1 }
  <c' e' g'>1
}
```



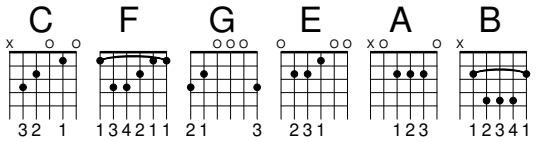
Es corriente que se impriman juntos los nombres de acorde y los diagramas de traste. Esto se puede hacer poniendo un contexto de nombres de acorde `ChordNames` en paralelo con un contexto de trastes `FretBoards` y dando a los dos contextos el mismo contenido musical.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode {
  c1 f g
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



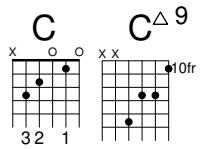
Los diagramas de traste predefinidos se pueden transportar, en la medida en que esté almacenado un diagrama para el acorde transportado, en la tabla de diagramas de traste.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

mychordlist = {
  \mychords
  \transpose c e { \mychords }
}
<<
  \new ChordNames {
    \mychordlist
  }
  \new FretBoards {
    \mychordlist
  }
>>
```



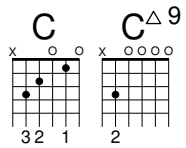
La tabla de diagramas de traste predefinidos para la guitarra contiene ocho acordes (mayor, menor, aumentado, disminuido, séptima dominante, séptima mayor, menor séptima y novena domniante) para 17 tonalidades distintas. La tabla de diagramas predefinidos para el ukelele contiene estos acordes y además otros tres (sexta mayor, segunda suspendida y cuarta suspendida). Véase Sección B.4 [Diagramas predefinidos de posiciones de trastes], página 899, para ver una lista completa de los diagramas de traste predefinidos. Si no hay una entrada en la tabla para un acorde, el grabador `FretBoards` calcula un cadena de diagrama fret-diagram usando la funcionalidad de diagramas automáticos, véase Sección 12.1.7 [Diagramas de traste automáticos], página 480.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode {
  c1 c:maj9
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Se pueden añadir diagramas de posiciones a la tabla de diagramas de posiciones. Para añadir un diagrama debemos especificar la tabla de correspondencias para el diagrama, el acorde del diagrama, la afinación utilizada y una definición del diagrama. Normalmente la tabla de correspondencias será *default-fret-table*. La definición del diagrama puede ser una cadena de definición escueta `\fret-diagram-terse` o una lista de marcados prolija `\fret-diagram-verbose`.

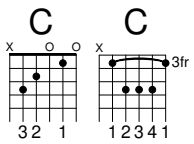
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
                        \chordmode { c:maj9 }
                        #guitar-tuning
                        "x;3-2;o;o;o;o;"

mychords = \chordmode {
  c1 c:maj9
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Se pueden almacenar distintos diagramas de traste para el mismo acorde usando distintas octavas para las notas. Las octavas de diferencia deben ser al menos dos por encima o por debajo de la octava predeterminada, porque las octavas inmediatas superior e inferior de la predeterminada se utilizan para los trasteros transpositores.

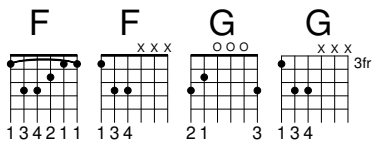
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
                        \chordmode { c'' }
                        #guitar-tuning
                        #(offset-fret 2
                           (chord-shape 'bes guitar-tuning))

mychords = \chordmode {
  c1 c''
}

<<
```

```
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Además de los diagramas de traste, LilyPond almacena una lista interna de formas de acorde. Las formas de acorde son diagramas de traste que se pueden desplazar por el mástil para dar acordes distintos. Se pueden añadir formas de acorde a la lista interna y luego usarlas para definir diagramas de posición predefinidos. Dado que se pueden mover a distintas posiciones dentro del mástil, las formas de acorde normalmente no contienen cuerdas al aire. Como los diagramas de posiciones, las formas de acorde se pueden introducir como cadenas escuetas `\fret-diagram-terse` o como listas de marcado prolijas `\fret-diagram-verbose`.

```
\include "predefined-guitar-fretboards.ly"

% Add a new chord shape

\addChordShape #'powerf #guitar-tuning "1-1;3-3;3-4;x;x;x;"

% add some new chords based on the power chord shape

\storePredefinedDiagram #default-fret-table
                        \chordmode { f'' }
                        #guitar-tuning
                        #(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram #default-fret-table
                        \chordmode { g'' }
                        #guitar-tuning
                        #(offset-fret 2
                           (chord-shape 'powerf guitar-tuning))

mychords = \chordmode {
  f1 f'' g g''
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



El aspecto gráfico de un diagrama de traste se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`; consulte Sección "Fret diagrams explained and developed" en *Fragmentos de código* para ver un ejemplo exhaustivo. Para un diagrama de trastes predefinido, las propiedades de interface pertenecen a `FretBoards.FretBoard`.

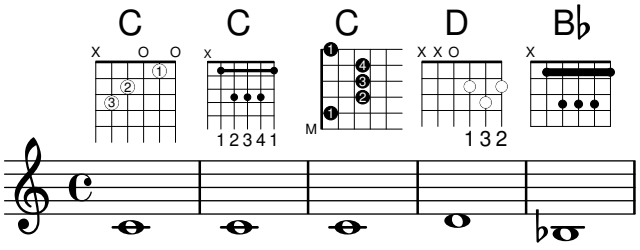
Fragmentos de código seleccionados

Personalizar los diagramas de posiciones

Se pueden modificar las propiedades de los diagramas de posiciones de acordes estableciendo la propiedad `fret-diagram-details`. Para los diagramas de posiciones de `FretBoard`, se aplican los overrides (sobreescrituras) al objeto `FretBoards.FretBoard`. Como `Voice`, `FretBoards` es un contexto del nivel inferior, y por tanto se puede omitir su nombre en la sobreescritura de propiedades.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        "x;1-1-(;3-2;3-3;3-4;1-1-);"

% shorthand
oo = #(define-music-function
       (grob-path value)
       (list? scheme?)
       #{ \once \override $grob-path = #value #})

<<
  \new ChordNames {
    \chordmode { c1 | c | c | d | bes }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = 1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \oo FretBoard.size #1.0
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      c'
      \oo FretBoard.fret-diagram-details.barre-type #'none
      \oo FretBoard.fret-diagram-details.number-type #'arabic
      \oo FretBoard.fret-diagram-details.orientation #'landscape
      \oo FretBoard.fret-diagram-details.mute-string "M"
      \oo FretBoard.fret-diagram-details.label-dir #LEFT
```

```
      \oo FretBoard.fret-diagram-details.dot-color #'black
      c'
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      \oo FretBoard.fret-diagram-details.dot-radius #0.35
      \oo FretBoard.fret-diagram-details.dot-position #0.5
      \oo FretBoard.fret-diagram-details.fret-count #3
      d
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.finger-code #'none
      \oo FretBoard.fret-diagram-details.dot-radius #0.25
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.string-overhang #0.
      \oo FretBoard.fret-diagram-details.barre-thickness #2.
      bes
    }
  }
  \new Voice {
    c'1 | c' | c' | d' | bes
  }
>>
```



Setting up predefined fretboards for other instruments

Predefined fret diagrams can be added for new instruments in addition to the standard diagrams used for guitar. This file shows how this is done by defining a new string tuning and a few predefined fretboards for the Venezuelan *cuatro*.

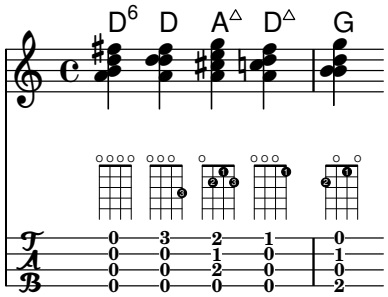
This file also shows how fingerings can be included in the chords used as reference points for the chord lookup, and displayed in the fret diagram and the `TabStaff`, but not the music.

These fretboards are not transposable because they contain string information. This is planned to be corrected in the future.

```
% Add fretboards for the cuatro.
%
%   Note: This section could be put into a separate file
%         `predefined-cuatro-fretboards.ly`
%         and be \included into each of your compositions.

cuatroTuning = #`(,(ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
```

```
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
                        "o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
                        "o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                        #cuatroTuning
                        "o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
                        #cuatroTuning
                        "o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
                        #cuatroTuning
                        "2-2;o;1-1;o;"

% End of potential include file `predefined-cuatro-fretboards.ly`.


#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
```

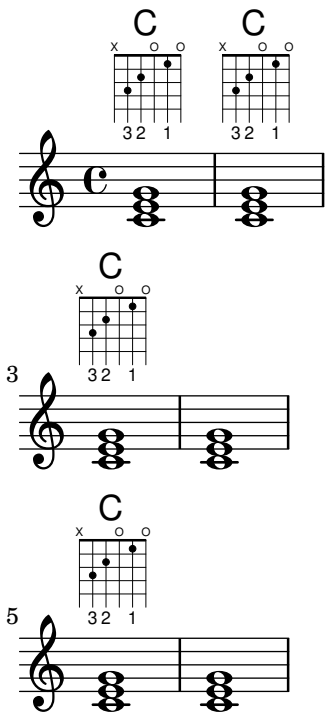
```
%       #'(fret-diagram-details string-count) = 4
      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }

  >>

  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration =
        \musicLength 16
    }
  }
  \midi { }
}
```



Chord changes for fretboards

Fretboards can be set to display only when the chord changes, or at the beginning of a new line.

```
\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

<<
  \new ChordNames { \myChords }
  \new FretBoards { \myChords }
  \new Staff { \myChords }
>>
```



Tablas alternativas de diagramas de posiciones

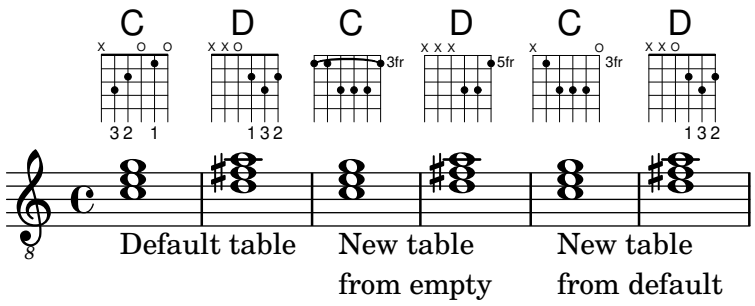
Se pueden crear tablas alternativas de diagramas de posiciones. Se utilizarían para tener diagramas altenativos para un acorde dado. Para usar una tabla alternativa de diagramas de posiciones, se debe crear la tabla primero. Después se añaden los diagramas a la tabla.

La tabla de diagramas de posiciones que se crea puede estar vacía o se puede copiar a partir de una tabla existente.

La tabla a usar en la impresión de los diagramas predefinidos se selecciona por medio de la propiedad \predefinedDiagramTable.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table.
#(define custom-fretboard-table-one
   (make-fretboard-table))

% Make a new fretboard table as a copy of `default-fret-table`.
#(define custom-fretboard-table-two
   (make-fretboard-table default-fret-table))

% Add a chord to `custom-fretboard-table-one`.
\storePredefinedDiagram #custom-fretboard-table-one
                        \chordmode {c}
                        #guitar-tuning
                        "3-(;3;5;5;5;3-);"

% Add a chord to `custom-fretboard-table-two`.
\storePredefinedDiagram #custom-fretboard-table-two
                        \chordmode {c}
                        #guitar-tuning
                        "x;3;5;5;5;o;"
```

```
<<
  \chords {
    c1 | d1 |
    c1 | d1 |
    c1 | d1 |
  }
  \new FretBoards {
    \chordmode {
      \set predefinedDiagramTable = #default-fret-table
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-one
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-two
      c1 | d1 |
    }
  }
  \new Staff {
    \clef "treble_8"
    <<
      \chordmode {
        c1 | d1 |
        c1 | d1 |
        c1 | d1 |
      }
      {
        s1_\markup "Default table" | s1 |
        s1_\markup \column { "New table" "from empty" } | s1 |
        s1_\markup \column { "New table" "from default" } | s1 |
      }
    >>
  }
>>
```



Véase también

Referencia de la notación: Sección 12.1.4 [Tablaturas personalizadas], página 455, Sección 12.1.7 [Diagramas de traste automáticos], página 480, Sección 15.1.1 [Panorámica del modo de acordes], página 510, Sección 12.1.6 [Diagramas predefinidos de trastes], página 469.

Archivos de inicio: `ly/predefined-guitar-fretboards.ly`, `ly/predefined-guitar-ninth-fretboards.ly`, `ly/predefined-ukulele-fretboards.ly`, `ly/predefined-mandolin-fretboards.ly`.

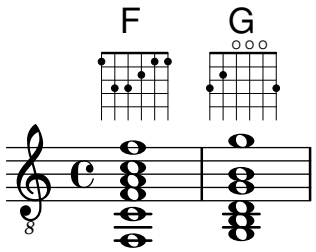
Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*.

12.1.7 Diagramas de traste automáticos

Se pueden crear diagramas de traste automáticamente a partir de las notas introducidas usando el contexto `FretBoards`. Si no está disponible ningún diagrama predefinido para las notas introducidas en la afinación `stringTunings` activa, este contexto calcula las cuerdas y los trastes que se pueden usar para tocar las notas.

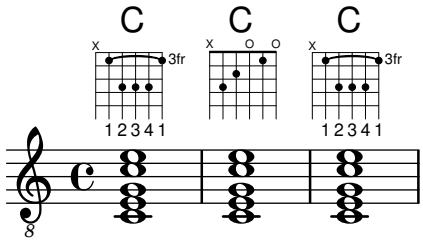
```
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new FretBoards {
    <f, c f a c' f'>1
    <g,\6 b, d g b g'>1
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1
    <g, b, d g b g'>1
  }
>>
```



Puesto que de forma predeterminada no se carga ningún diagrama predefinido, el comportamiento predeterminado es el cálculo automático de los diagramas de traste. Una vez que los diagramas predeterminados se han cargado, se puede habilitar e inhabilitar el cálculo automático con instrucciones predefinidas:

```
\storePredefinedDiagram #default-fret-table
                        <c e g c' e'>
                        #guitar-tuning
                        "x;3-1-(;5-2;5-3;5-4;3-1-1-);"
<<
  \new ChordNames {
    \chordmode {
      c1 c c
    }
  }
  \new FretBoards {
    <c e g c' e'>1
    \predefinedFretboardsOff
    <c e g c' e'>1
    \predefinedFretboardsOn
```

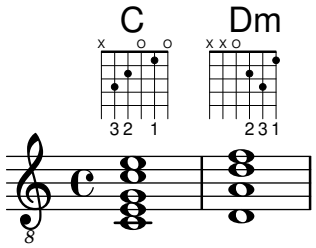
```
    <c e g c' e'>1
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <c e g c' e'>1
    <c e g c' e'>1
  }
>>
```



A veces el calculador de trastes es incapaz de encontrar un diagrama aceptable. Esto se puede remediar generalmente mediante la asignación manual de una nota a una cuerda. En muchos casos solo hay que colocar manualmente una nota sobre una cuerda; el resto de las notas se situará en el lugar adecuado por parte del contexto `FretBoards`.

Se pueden añadir digitaciones a los diagramas de traste del contexto `FretBoard`.

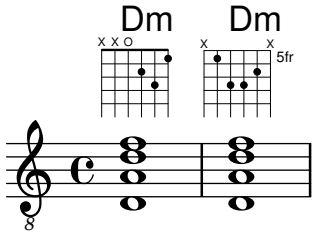
```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new FretBoards {
    <c-3 e-2 g c'-1 e'>1
    <d a-2 d'-3 f'-1>1
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <d a d' f'>1
  }
>>
```



El traste mínimo que usar en el cálculo de cuerdas y trastes para el contexto `FretBoard` se puede fijar con la propiedad `minimumFret`.

```
<<
  \new ChordNames {
    \chordmode {
```

```
      d1:m d:m
    }
  }
  \new FretBoards {
    <d a d' f'>1
    \set FretBoards.minimumFret = 5
    <d a d' f'>1
  }
  \new Staff {
    \clef "treble_8"
    <d a d' f'>1
    <d a d' f'>1
  }
>>
```



Las cuerdas y los trastes para el contexto `FretBoards` dependen de la propiedad `stringTunings`, que tiene el mismo significado que en el contexto de tablatura `TabStaff` context. Véase Sección 12.1.4 [Tablaturas personalizadas], página 455, para ver más información sobre la propiedad `stringTunings`.

El aspecto gráfico de un diagrama de trastes se puede personalizar de acuerdo a las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección "fret-diagram-interface" en *Referencia de Funcionamiento Interno*; consulte Sección "Fret diagrams explained and developed" en *Fragmentos de código* para ver un ejemplo exhaustivo. Para un diagrama de `FretBoards`, las propiedades del interface pertenecen a `FretBoards.FretBoard`.

Instrucciones predefinidas

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

Véase también

Referencia de la notación: Sección 12.1.4 [Tablaturas personalizadas], página 455.

Fragmentos de código: Sección "Instrumentos de cuerda con trastes" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "fret-diagram-interface" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

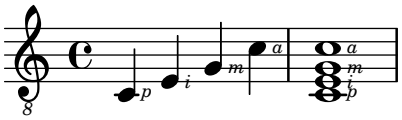
Los cálculos de tablatura automáticos no funcionan bien para instrumentos con afinaciones no monótonas.

12.1.8 Digitaciones de la mano derecha

Las digitaciones de la mano derecha *p-i-m-a* se deben introducir usando `\rightHandFinger` seguido de un número.

> **Nota:** Si el número se introduce en notación de Scheme, recuerde escribir al final un espacio seguido de un ángulo de cierre > o similar.

```
\clef "treble_8"
c4\rightHandFinger 1
e\rightHandFinger 2
g\rightHandFinger 3
c'\rightHandFinger 4
<c\rightHandFinger 1 e\rightHandFinger 2
 g\rightHandFinger 3 c'\rightHandFinger 4 >1
```



Por brevedad, se puede escribir `\rightHandFinger` como algo más corto, como por ejemplo `RH`:

```
RH = \rightHandFinger \etc
```

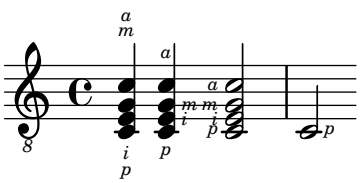
Casi todos los comprtamientos de las digitaciones de la mano derecha (concretamente el objeto `StrokeFinger`) se pueden establecer de la misma forma que las digitaciones ordinarias: véase Sección 7.1.2 [Indicaciones de digitación], página 284.

Fragmentos de código seleccionados

Posicionamiento de digitaciones de mano derecha

Es posible ejercer un mayor control sobre la colocación de las digitaciones de la mano derecha estableciendo el valor de una propiedad específica, como se muestra en el ejemplo siguiente.

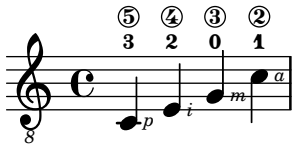
```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

  \set strokeFingerOrientations = #'(up down)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

  \set strokeFingerOrientations = #'(left)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >2

  \set strokeFingerOrientations = #'(right)
  c\RH 1
}
```



Digitaciones, indicación del número de cuerda y digitaciones de mano derecha

En este ejemplo se combinan las digitaciones de la mano izquierda, indicaciones del número de cuerda y digitaciones de la mano derecha.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"
  <c-3\5\RH 1 >4
  <e-2\4\RH 2 >4
  <g-0\3\RH 3 >4
  <c-1\2\RH 4 >4
}
```



Véase también

Referencia de la notación: Sección 7.1.2 [Indicaciones de digitación], página 284.

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StrokeFinger” en *Referencia de Funcionamiento Interno*.

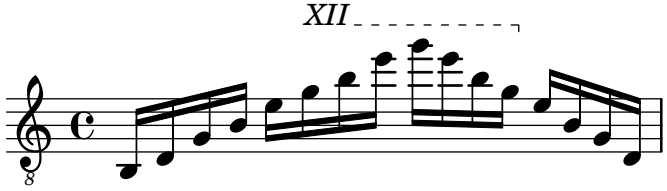
12.2 Guitarra

Casi todos los asuntos de notación relacionados con la música para guitarra están suficientemente cubiertos en la sección general sobre ajustes de instrumentos de trastes, pero hay algunos más de los que merece la pena hablar aquí. De forma ocasional, los usuarios quieren crear documentos del tipo de cancioneros que tengan solo la letra de las canciones con indicaciones de acordes sobre ella. Dado que LilyPond es un tipografiador de música, no se recomienda para documentos que no tienen notación musical. Una alternativa mejor es un procesador de texto, editor de texto o, para usuarios con experiencia, un tipografiador como GuitarTeX.

12.2.1 Indicar la posición y la cejilla

Este ejemplo muestra cómo incluir indicaciones de posición y cejilla de guitarra usando una línea de barré.

```
\relative {
  \clef "treble_8"
  b,16 d g b e
  \once \override TextSpanner.bound-details.left.text =
    \markup {"XII" \hspace #0.4 }
  \once \override TextSpanner.bound-details.right.text =
    \markup \draw-line #'(0 . -.5)
  \once \override TextSpanner.bound-details.right.padding = -0.65
  \once \override TextSpanner.dash-fraction = 0.4
  \once \override TextSpanner.dash-period = 1.3
  g16\startTextSpan
  b16 e g e b g\stopTextSpan
  e16 b g d
}
```



A more sophisticated solution can be found in LSR snippet 952 (`https://lsr.di.unimi.it/LSR/Item?id=952`).

Véase también

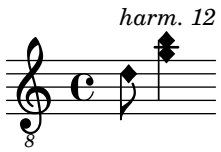
Referencia de la notación: Sección 8.1.3 [Objetos extensos de texto], página 315.

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*, Sección “Marcas de expresión” en *Fragmentos de código*.

12.2.2 Indicar armónicos y notas tapadas

Se pueden usar cabezas de nota especiales para indicar notas apagadas o armónicos. Los armónicos se suelen explicar de forma más completa con un marcado de texto.

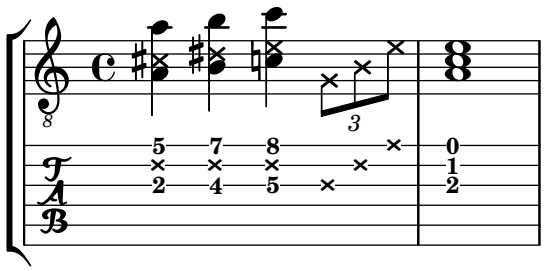
```
\relative {
  \clef "treble_8"
  \override NoteHead.style = #'harmonic-mixed
  d'8^\markup { \italic \fontsize #-2 "harm. 12" } <g b>4
}
```



Las notas tapadas (llamadas también *notas muertas*) están contempladas dentro de los pentagramas normales y de tablatura. En el ejemplo siguiente, estas notas se muestran con alturas en la pauta normal, indicando los trastes donde están tapadas. Una notación alternativa sería usar cuerdas vacías. Sin embargo, en la notación de tablatura, no hay ninguna diferencia.

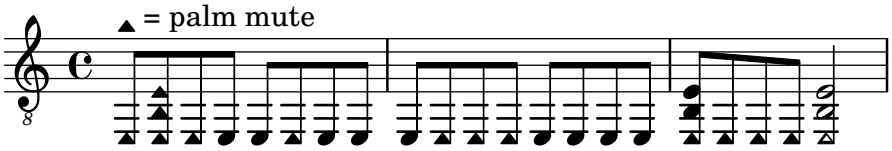
```
music = \relative c' {
  \omit StringNumber

  <a \deadNote cis a'>4
    <b\3 \deadNote dis b'>
    <c\3 \deadNote e\2 c'>
    \deadNotesOn
    \tuplet 3/2 { g8 b e }
    \deadNotesOff
  <a, c e>1
}

\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \music
  }
  \new TabStaff {
    \music
```

```
  }
>>
```



Otra téctnica de ejecución (utilizada especialmente en la guitarra eléctrica) se llama *tapado de palma*. La cuerda es parcialmente tapada por la palma de la mano que pulsa (de ahí el nombre). LilyPond contempla la notación de las notas de estilo de tapado de palma mediante el cambio de la forma de la cabeza de nota a un triángulo.

```
\relative c, {
  \clef "G_8"
  \palmMuteOn
    e8^\markup { \musicglyph "noteheads.s2do" = palm mute }
    <e b' e> e
    \palmMuteOff
    e e \palmMute e e e |
  e8 \palmMute { e e e } e e e e |
  <\palmMute e b' e>8 \palmMute { e e e } <\palmMute e b' e>2
}
```



Véase también

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

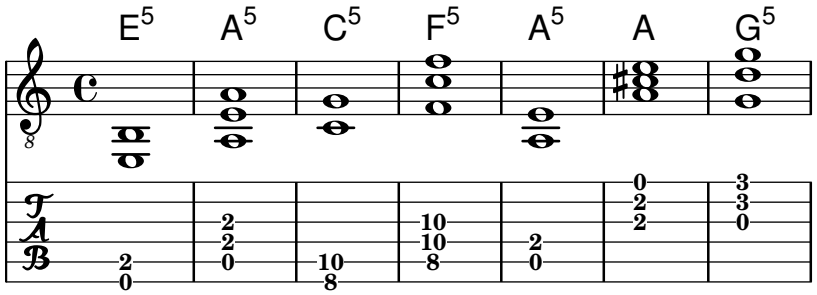
Referencia de la notación: Sección 1.4.1 [Cabezas de nota especiales], página 44, Sección B.9 [Estilos de cabezas de nota], página 927.

12.2.3 Indicación de acordes de potencia o de quinta vacía

Los acordes de quinta vacía o acordes de potencia (*power chords*) y sus símbolos se pueden tipografiar en el modo de acordes o como construcciones de acorde normales. Como excepción, se especifica la quinta en estos nombres de acorde, meintras que habitualmente se omite en otros acordes (p. ej. en las tríadas mayores o menores).

```
ChordsAndSymbols = {
  \chordmode {
    e,,1:5
    a,,:5.8
    \set TabStaff.restrainOpenStrings = ##t
    \set minimumFret = 8
    c,:5
    f,:5.8
  }
  \set minimumFret = 2
```

```
  \set restrainOpenStrings = ##f
  <a, e> <a cis' e'>
  <g d' g'>
}
\score {
  <<
    \new ChordNames {
    \ChordsAndSymbols
    }
    \new Staff {
      \clef "treble_8"
      \ChordsAndSymbols
    }
    \new TabStaff {
      \ChordsAndSymbols
    }
  >>
}
```



Véase también

Glosario musical: Sección “power chord” en *Glosario Musical*.

Referencia de la notación: Sección 15.1.3 [Acordes extendidos y alterados], página 512, Sección 15.2.1 [Impresión de los nombres de acorde], página 515.

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

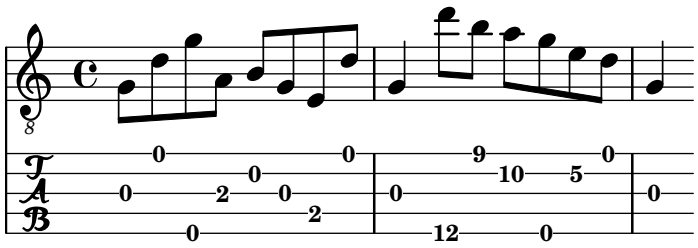
12.3 Banjo

12.3.1 Tablaturas de banjo

LilyPond contempla el banjo de cinco cuerdas de una manera básica. Cuando haga tablaturas para banjo de cinco cuerdas, utilice la función de formato de tablatura de banjo para obtener los números correctos de los trastes para la quinta cuerda:

```
music = {
  g8 d' g'\5 a b g e d' |
  g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
  g4
}

<<
  \new Staff \with { \omit StringNumber }
  { \clef "treble_8"  \music }
  \new TabStaff \with {
    tablatureFormat = #fret-number-tablature-format-banjo
```

```
    stringTunings = #banjo-open-g-tuning
  }
  { \music }
>>
```



Se encuentran predefinidas varias afinaciones comunes para el banjo de cinco cuerdas: `banjo-open-g-tuning` (sol-Re-Sol-Si-Re), `banjo-c-tuning` (sol-Do-Sol-Si-Re), `banjo-modal-tuning` (sol-Re-Sol-Do-Re), `banjo-open-d-tuning` (Re mayor al aire, la-Re-Fa#-La-Re), `banjo-open-dm-tuning` (Re menor al aire, la-Re-Fa-La-Re), `banjo-double-c-tuning` (doble Do, sol-Do-Sol-Do-Re) y `banjo-double-d-tuning` (doble Re, la-Re-Sol-Re-Mi).

Estas afinaciones se pueden convertir a afinaciones para banjo de cuatro cuerdas utilizando la función `four-string-banjo`:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

Véase también

Archivos de inicio: `ly/string-tunings-init.ly`

Fragmentos de código: Sección “Instrumentos de cuerda con trastes” en *Fragmentos de código*.

12.4 Laúd

12.4.1 Tablaturas de laúd

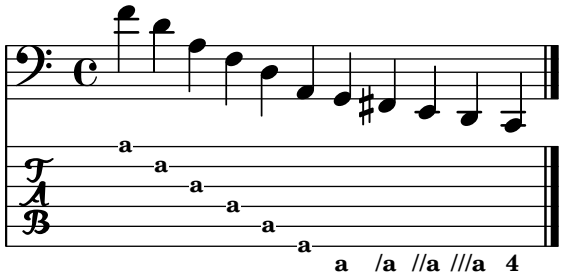
LilyPond contempla las tablaturas para laúd.

Para obtener las cuerdas graves adicionales, utilice `additionalBassStrings`, donde se determinan las notas correspondientes a dichas cuerdas. Se imprimen por debajo de la línea inferior como: a, /a, //a, ///a, 4, 5, etc.

Se debe usar el valor `fret-letter-tablature-format` para la propiedad de contexto `tablatureFormat` y probablemente también ajustar la propiedad `fretLabels` para una mayor personalización.

```
m = { f'4 d' a f d a, g, fis, e, d, c,  \bar "|." }

\score {
  <<
    \new Staff { \clef bass \cadenzaOn  \m }
    \new TabStaff \m
  >>
  \layout {
    \context {
      \Score
      tablatureFormat = #fret-letter-tablature-format
    }
    \context {
      \TabStaff
      stringTunings = \stringTuning <a, d f a d' f'>
```

```
      additionalBassStrings = \stringTuning <c, d, e, fis, g,>
      fretLabels = #'("a" "b" "r" "d" "e" "f" "g" "h" "i" "k")
    }
  }
}
```



Advertencias y problemas conocidos

No se contempla el uso de `FretBoards` con `additionalBassStrings`; de utilizarse, puede dar lugar a un resultado poco satisfactorio.

13 Percusión

13.1 Notación común para percusión

La notación rítmica se emplea primordialmente para la percusión y la batería, pero también se puede utilizar para mostrar los valores rítmicos una melodía.

13.1.1 Referencias para percusión

- La notación de algunas percusiones se puede hacer sobre una pauta de ritmo; esto se estudia en Sección 2.3.7 [Mostrar los ritmos de la melodía], página 98, y Sección 6.1.1 [Crear instancias de pentagramas nuevos], página 244.
- La salida MIDI se trata en una sección aparte; véase Capítulo 24 [Creación de salida MIDI], página 655.

Véase también

Referencia de la notación: Sección 2.3.7 [Mostrar los ritmos de la melodía], página 98, Sección 6.1.1 [Crear instancias de pentagramas nuevos], página 244, Capítulo 24 [Creación de salida MIDI], página 655.

Fragmentos de código: Sección “Percusión” en *Fragmentos de código*.

13.1.2 Notación básica de percusión

Las notas de percusión se pueden escribir en el modo `\drummode`, que es similar al modo estándar para introducir notas. La manera más fácil de escribir notas de percusión es utilizar la instrucción `\drums`, que crea el contexto y el modo de entrada apropiados para percusión:

```
\drums {
  hihat4 hh bassdrum bd
}
```



Esto es una abreviatura de

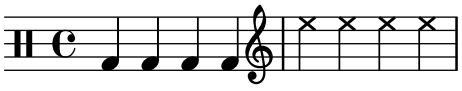
```
\new DrumStaff \drummode {
  hihat4 hh bassdrum bd
}
```



Cada elemento de un set de percusión lleva un nombre completo y un nombre abreviado, y los dos se pueden usar en la entrada. La lista completa de nombre de instrumentos de percusión se encuentra en Sección B.15 [Notas de percusión], página 937.

Observe que la notación normal de las notas con altura determinada (como `cis4`) en un contexto `DrumStaff` producen un mensaje de error. Las claves de percusión se añaden automáticamente al contexto `DrumStaff` pero también se pueden establecer explícitamente o utilizar otras claves.

```
\drums {
  \clef percussion
  bd4 4 4 4
  \clef treble
  hh4 4 4 4
}
```



Hay ciertos detalles respecto a cómo está contemplado el MIDI para los instrumentos de percusión; para ver los detalles consulte Capítulo 24 [Creación de salida MIDI], página 655.

Véase también

Referencia de la notación: Capítulo 24 [Creación de salida MIDI], página 655, Sección B.15 [Notas de percusión], página 937.

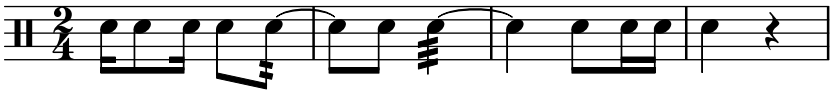
Archivos de inicio: `ly/drumpitch-init.ly`.

Fragmentos de código: Sección “Percusión” en *Fragmentos de código*.

13.1.3 Redobles

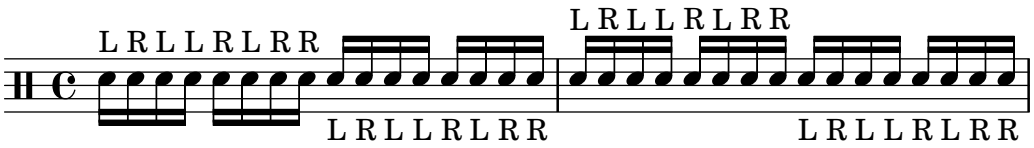
Los redobles se indican mediante tres barras cruzadas en la plica. Para las negras o notas más largas, las tres barras se muestran explícitamente, las corcheas se presentan con dos barras cruzadas (siendo la barra de corchea la tercera), y los redobles más breves que las corcheas tienen una barra cruzada para complementar las barras normales. Esto se consigue mediante la notación de trémolo, véase Sección 4.2.2 [Repeticiones de trémolo], página 214.

```
\drums {
  \time 2/4
  sn16 8 16 8 8:32 ~
  8 8 4:32 ~
  4 8 16 16
  4 r4
}
```



Los golpes de baqueta se pueden indicar mediante la colocación de los elementos de marcado `"R"` o `"L"` encima o debajo de las notas, como se estudia en Sección 36.1 [Dirección y posición], página 783. La propiedad `staff-padding` se puede sobreescribir para conseguir una línea de base satisfactoria.

```
\drums {
  \repeat unfold 2 {
    sn16^"L" 16^"R" 16^"L" 16^"L" 16^"R" 16^"L" 16^"R" 16^"R"
    \stemUp
    sn16_"L" 16_"R" 16_"L" 16_"L" 16_"R" 16_"L" 16_"R" 16_"R"
  }
}
```



Véase también

Referencia de la notación: Sección 4.2.2 [Repeticiones de trémolo], página 214.

Fragmentos de código: Sección “Percusión” en *Fragmentos de código*.

13.1.4 Percusión afinada

Ciertos instrumentos de percusión de altura determinada (p.ej.: el xilófono, el vibráfono y los timbales) se escriben usando pentagramas normales. Esto se estudia en otras secciones del manual.

Véase también

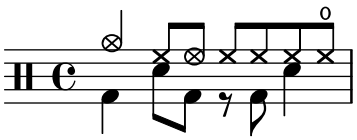
Referencia de la notación: Sección B.6 [Instrumentos MIDI], página 906.

Fragmentos de código: Sección "Percusión" en *Fragmentos de código*.

13.1.5 Pautas de percusión

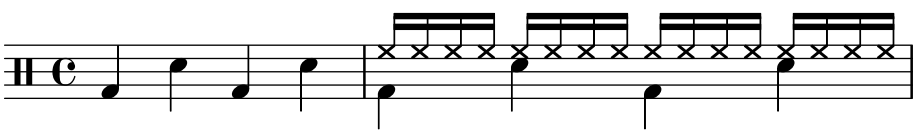
Una parte de percusión para más de un instrumento, normalmente utiliza una pauta de varias líneas donde cada posición dentro de la pauta se refiere a un elemento de percusión. Para tipografiar la música, se deben interpretar las notas dentro de los contextos Sección "DrumStaff" en *Referencia de Funcionamiento Interno* y Sección "DrumVoice" en *Referencia de Funcionamiento Interno*.

```
up = \drummode {
  crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat
}
down = \drummode {
  bassdrum4 snare8 bd r bd sn4
}
\new DrumStaff <<
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



El ejemplo anterior muestra una notación polifónica prolija. La notación polifónica abreviada, descrita en Sección "Oigo voces" en *Manual de Aprendizaje*, también se puede usar. Por ejemplo,

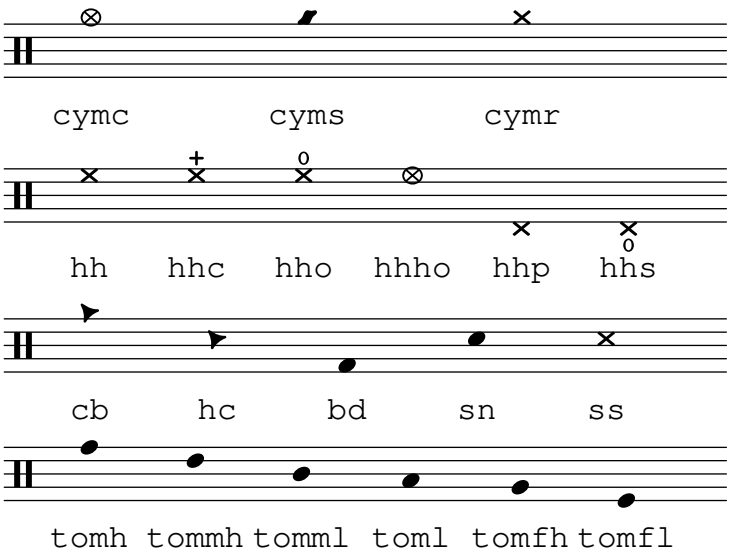
```
\new DrumStaff <<
  \drummode {
    bd4 sn4 bd4 sn4
    << {
      \repeat unfold 16 hh16
    } \\ {
      bd4 sn4 bd4 sn4
    } >>
  }
>>
```



También existen otras posibilidades en lo referente a la disposición. Para usarlas, establezca la propiedad `drumStyleTable` en el contexto Sección "DrumVoice" en *Referencia de Funcionamiento Interno*. Se han predefinido las siguientes variables:

`drums-style`

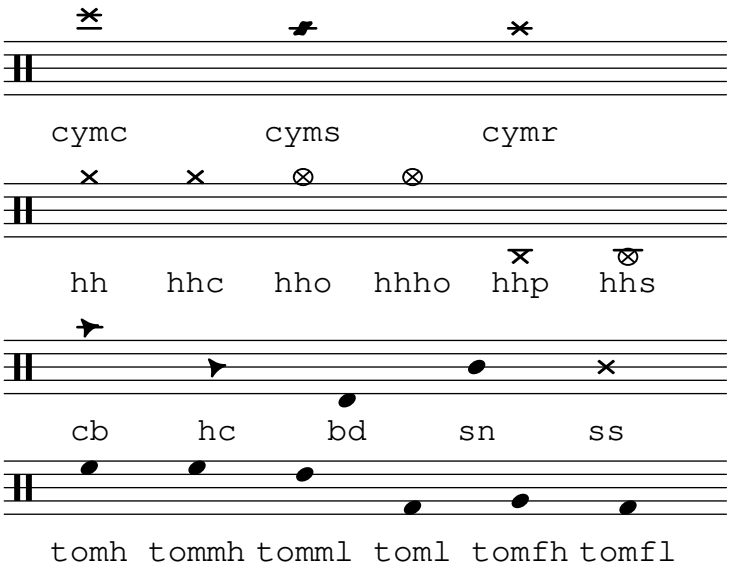
Es la opción por defecto. Se usa para tipografiar una batería típica sobre pentagrama:



El esquema de percusión contempla seis tambores graves (toms) distintos. Cuando haya menos toms, sencillamente seleccione aquellos que producen el resultado deseado, es decir, para tener toms en las tres líneas centrales utilizará `tommh`, `tomml` y `tomfh`.

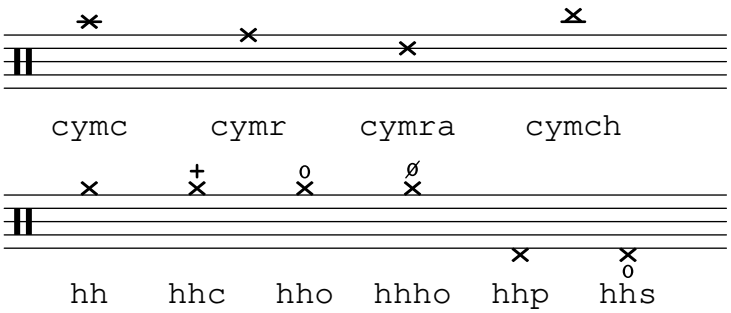
`agostini-drums-style`

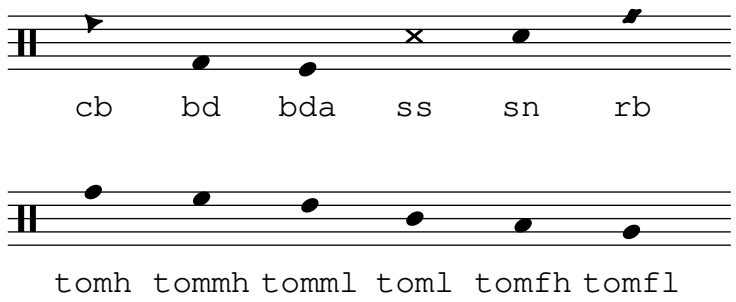
Inventado por el percusionista francés Dante Agostini en 1965, esta notación se emplea normalmente en Francia pero también en muchos otros lugares.



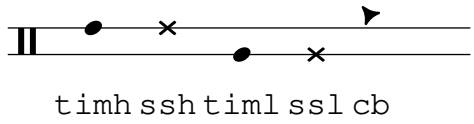
`weinberg-drums-style`

Basado en el trabajo de Norman Weinberg, publicado en sus *Guidelines for Drumset Notation* (Guía de la notación de batería).

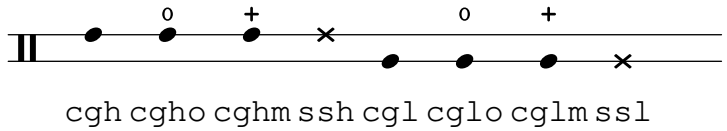




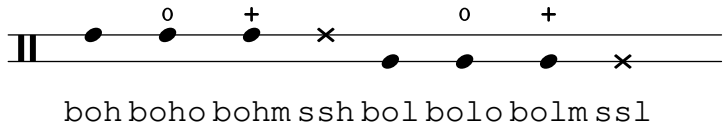
`timbales-style`
: Se usa para tipografiar timbales en una pauta de dos líneas:



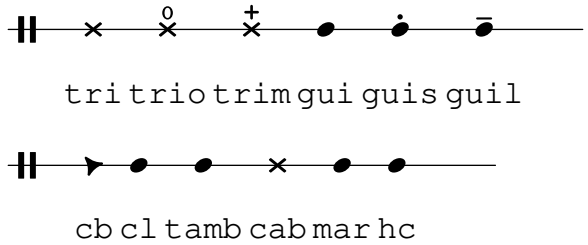
`congas-style`
: Se usa para tipografiar congas en una pauta de dos líneas:



`bongos-style`
: Se usa para bongos sobre una pauta de dos líneas:



`percussion-style`
: Para tipografiar toda clase de percusiones simples sobre pautas de una línea:



También se pueden definir estilos de percusión personalizados como se explica en Sección 13.1.6 [Pautas de percusión personalizadas], página 495.

Véase también

Manual de aprendizaje: Sección “Oigo voces” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 13.1.6 [Pautas de percusión personalizadas], página 495.

Archivos instalados: `ly/drumpitch-init.ly`.

Fragmentos de código: Sección “Percusión” en *Fragmentos de código*.

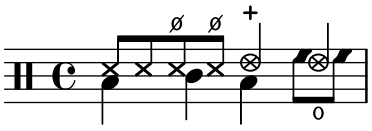
13.1.6 Pautas de percusión personalizadas

Se pueden definir estilos de percusión personalizados, a los que después puede establecerse la propiedad `drumStyleTable`. Las notaciones existentes se pueden redefinir como una lista de asociación en la que cada entrada debe estar compuesta de cuatro elementos: un nombre, el estilo de cabezas de nota (o '`()`' para indicar `default`), un símbolo de articulación si es necesario (o `#f` en caso contrario), y la posición de la cabeza sobre la pauta. Dicha lista se debe entonces convertir a una tabla de hash de Scheme usando la función `alist->hash-table`.

La entrada para una articulación puede ser el símbolo del nombre de una inscripción, o bien una pareja, donde el primer elemento es el símbolo del nombre de una inscripción, y el segundo un indicador de dirección forzada para la inscripción.

[En el ejemplo siguiente, observe el uso de la abreviatura de cuasi-notación (‘`’) al principio de la expresión de Scheme en lugar de la abreviatura estándar de comilla (‘'’), que habilita a la abreviatura de no-comilla (‘,’) a evaluar el elemento que le precede.]

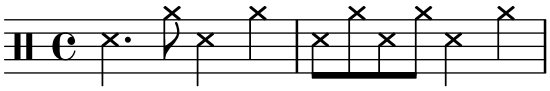
```
#(define mydrums `(
         (bassdrum        ()        #f              -1)
         (snare           ()        #f              0)
         (hihat           cross     #f              1)
         (halfopenhihat   cross     halfopen        1)
         (pedalhihat      xcircle   stopped         2)
         (splashhihat     xcircle   (open . ,DOWN)  2)
         (lowtom          diamond   #f              3)))

up = \drummode { hh8 hh hhho hhho hhp4 hhs }
down = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff \with { drumStyleTable = #(alist->hash-table mydrums) }
  <<
    \new DrumVoice { \voiceOne \up }
    \new DrumVoice { \voiceTwo \down }
  >>
```



También se pueden añadir nombres nuevos a dichas notaciones personalizadas mediante la variable `drumPitchNames`, que se puede redefinir como una lista de asociación (o aumentada añadiendo al final una lista nueva a su valor existente, como se muestra más abajo), pero también a través de sus entradas individuales. Esto también hace posible definir nombres alternativos o alias.

```
drumPitchNames =
  #(append
    '((leftsnap . sidestick)
      (rightsnap . ridecymbal))
    drumPitchNames)

drumPitchNames.ls = #'sidestick
drumPitchNames.rs = #'ridecymbal

\drums {
  leftsnap4. rightsnap8 leftsnap4 rightsnap
```

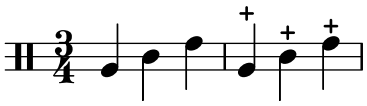
```
  ls8 rs ls rs ls4 rs
}
```



De manera similar, la propiedad `drumPitchTable` asocia una altura específica (con el significado de un sonido instrumental distinto, tal y como lo proveen las fuentes de sonido MIDI) a cada notación. Dicha propiedad necesita definirse como una tabla de hash, que es de nuevo convertida a partir de una lista de asociación (almacenada de forma predeterminada como la variable `midiDrumPitches`). La redefinición de estas asociaciones se consigue como se ha explicado más arriba, ya sea definiendo una lista de asociación completa o mediante las entradas individuales. El siguiente ejemplo muestra cómo crear un conjunto completo de notaciones con su propia sintaxis de entrada, su propia notación personalizada y la salida MIDI correspondiente.

```
drumPitchNames.dbass     = #'dbass
drumPitchNames.dba       = #'dbass  % 'db is in use already
drumPitchNames.dbassmute = #'dbassmute
drumPitchNames.dbm       = #'dbassmute
drumPitchNames.do        = #'dopen
drumPitchNames.dopenmute = #'dopenmute
drumPitchNames.dom       = #'dopenmute
drumPitchNames.dslap     = #'dslap
drumPitchNames.ds        = #'dslap
drumPitchNames.dslapmute = #'dslapmute
drumPitchNames.dsm       = #'dslapmute

#(define djembe-style
  '((dbass       ()   #f        -2)
    (dbassmute   ()   stopped   -2)
    (dopen       ()   #f        0)
    (dopenmute   ()   stopped   0)
    (dslap       ()   #f        2)
    (dslapmute   ()   stopped   2)))

midiDrumPitches.dbass     = g
midiDrumPitches.dbassmute = fis
midiDrumPitches.dopen     = a
midiDrumPitches.dopenmute = gis
midiDrumPitches.dslap     = b
midiDrumPitches.dslapmute = ais

test = \drummode { dba4 do ds dbm dom dsm }

\score {
  \new DrumStaff \with {
    \override StaffSymbol.line-count = 3
    instrumentName = "Djembé "
    drumStyleTable = #(alist->hash-table djembe-style)
    drumPitchTable = #(alist->hash-table midiDrumPitches)
  } {
    \time 3/4
    \test
```

```
  }
  \layout {}
  \midi {}
}
```



Véase también

Archivos instalados: `ly/drumpitch-init.ly`.

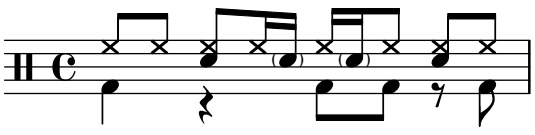
Fragmentos de código: Sección “Percusión” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “DrumStaff” en *Referencia de Funcionamiento Interno*, Sección “DrumVoice” en *Referencia de Funcionamiento Interno*.

13.1.7 Notas fantasma

Las notas fantasma para la batería e instrumentos de percusión, también conocidas como notas muertas, tapadas, silenciadas o falsas, se pueden crear utilizando la instrucción `\parenthesize`, véase Sección 7.1.8 [Paréntesis], página 298.

```
\new DrumStaff <<
  \new DrumVoice  = "1" { s1 }
  \new DrumVoice  = "2" { s1 }
  \drummode {
    <<
      {
        hh8[ 8] <hh sn> hh16
        \parenthesize sn hh
        \parenthesize sn hh8 <hh sn> hh
      } \\
      {
        bd4 r4 bd8 8 r8 bd
      }
    >>
  }
>>
```

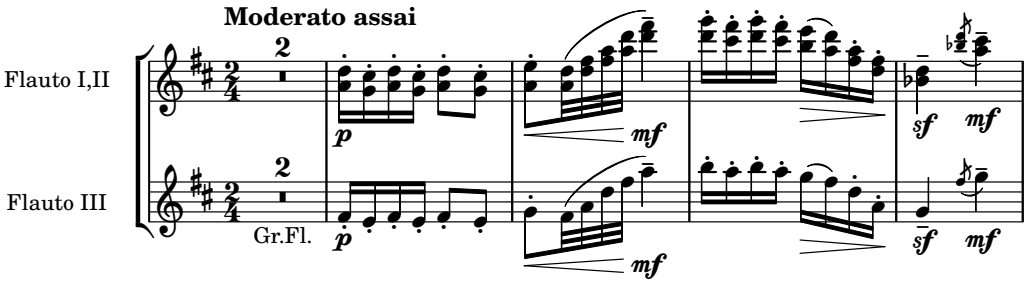


Véase también

Referencia de la notación: Sección 7.1.8 [Paréntesis], página 298.

Fragmentos de código: Sección “Percusión” en *Fragmentos de código*.

14 Instrumentos de viento



Esta sección incluye elementos de notación musical que afloran al escribir para instrumentos de viento.

14.1 Notación común para instrumentos de viento

Esta sección trata la notación común para casi todos los instrumentos de viento.

14.1.1 Referencias para instrumentos de viento

Muchas cuestiones de la notación para instrumentos de viento tienen que ver con las respiraciones y los golpes de lengua:

- Las respiraciones se pueden especificar mediante silencios o marcas de respiración, véase Sección 3.2.3 [Marcas de respiración], página 177.
- La ejecución ligada se indica mediante ligaduras de expresión, véase Sección 3.2.1 [Ligaduras de expresión], página 171.
- Los distintos golpes de lengua, desde legato hasta staccato pasando por non legato, se presentan por lo general mediante articulaciones, en ocasiones combinadas con ligaduras de expresión, véase Sección 3.1.1 [Articulaciones y ornamentos], página 156, y Sección B.13 [Lista de articulaciones], página 934.
- El frullato se indica generalmente mediante una indicación de trémolo y una marca textual sobre la nota. Véase Sección 4.2.2 [Repeticiones de trémolo], página 214.

También existen otros aspectos de la notación musical que son de aplicación a los instrumentos de viento:

- Muchos instrumentos de viento son transpositores, véase Sección 1.3.4 [Transposición de los instrumentos], página 30.
- El portamento es característico del trombón, pero otros instrumentos de viento pueden realizar glissandos con llaves o válvulas. Véase Sección 3.3.1 [Glissando], página 178.
- Los glissandos de serie armónica, que son posibles en todos los metales pero bastante específicos de las trompas, se escriben en general como notas de adorno, véase Sección 2.6.1 [Notas de adorno], página 148.
- Las inflexiones de tono al final de una nota se tratan en Sección 3.2.4 [Caídas y elevaciones], página 178.
- Los golpes de llave o de válvula se suelen indicar con el estilo `cross` (aspas) de cabezas de nota, véase Sección 1.4.1 [Cabezas de nota especiales], página 44.
- Los instrumentos de viento-madera pueden sobreinflar las notas graves para producir armónicos. Éstos se muestran por medio de la articulación `flageolet`. Véase Sección B.13 [Lista de articulaciones], página 934.
- El uso de sordinas para los metales se suele indicar mediante marcas de texto, pero en los lugares donde se suceden muchos cambios rápidos es mejor utilizar las articulaciones `stopped`

(tapado) y open (abierto). Véase Sección 3.1.1 [Articulaciones y ornamentos], página 156, y Sección B.13 [Lista de articulaciones], página 934.

- Las trompas tapadas se indican por medio de la articulación stopped. Véase Sección 3.1.1 [Articulaciones y ornamentos], página 156.

Véase también

Referencia de la notación: Sección 3.2.3 [Marcas de respiración], página 177, Sección 3.2.1 [Ligaduras de expresión], página 171, Sección 3.1.1 [Articulaciones y ornamentos], página 156, Sección B.13 [Lista de articulaciones], página 934, Sección 4.2.2 [Repeticiones de trémolo], página 214, Sección 1.3.4 [Transposición de los instrumentos], página 30, Sección 3.3.1 [Glissando], página 178, Sección 2.6.1 [Notas de adorno], página 148, Sección 3.2.4 [Caídas y elevaciones], página 178, Sección 1.4.1 [Cabezas de nota especiales], página 44.

Fragmentos de código: Sección "Instrumentos de viento" en *Fragmentos de código*.

14.1.2 Digitaciones

Todos los instrumentos de viento aparte del trombón requieren el uso de los distintos dedos para producir cada nota. En los fragmentos de código que aparecen a continuación pueden verse algunos ejemplos de digitaciones.

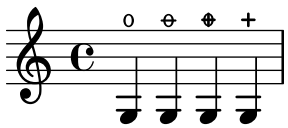
Se pueden producir diagramas de posiciones para instrumentos de viento madera,como se describe en Sección 14.3.1 [Diagramas de posiciones para viento madera], página 503.

Fragmentos de código seleccionados

Símbolos de digitación para instrumentos de viento

Se pueden conseguir símbolos especiales combinando glifos existentes, lo que es de utilidad para la notación de instrumentos de viento.

```
lineup =
  \tweak outside-staff-padding #0
  \tweak staff-padding #0
  \tweak padding #0.2
  \tweak parent-alignment-X #CENTER
  \tweak self-alignment-X #CENTER
  \etc

\relative c' {
  g\open
  g\lineup ^\markup \combine
              \musicglyph "scripts.open"
              \musicglyph "scripts.tenuto"
  g\lineup ^\markup \combine
              \musicglyph "scripts.open"
              \musicglyph "scripts.stopped"
  g\stopped
}
```

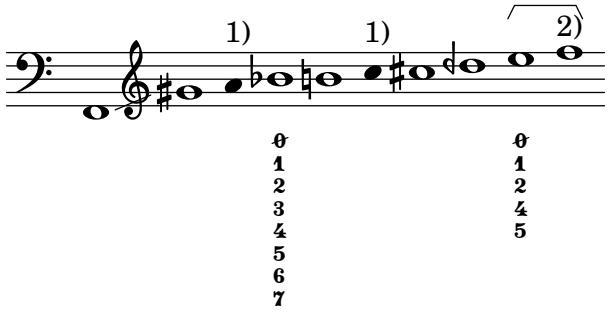


Diagramas de digitación para la flauta dulce

El ejemplo siguiente muestra cómo se pueden realizar diagramas de digitación para instrumentos de viento.

```
% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset = #(lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
       (ly:self-alignment-interface::x-aligned-on-self g)))
}

\new Staff \with {
  \remove "Time_signature_engraver"
  \omit Stem
  \omit Flag
  \consists "Horizontal_bracket_engraver"
} {
  \clef bass
  \set Score.timing = ##f

  f,1*1/4 \glissando

  \clef violin
  gis'1*1/4

  a'4^\markup "1)"

  \centermarkup
    \once \override TextScript.padding = 2
    bes'1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
      \fontsize #-5
      \slashed-digit #0 \finger 1 \finger 2
      \finger 3 \finger 4 \finger 5 \finger 6 \finger 7 }

  b'1*1/4

  c''4^\markup "1)"

  cis''1*1/4

  deh''1*1/4

  \centermarkup
    \once \override TextScript.padding = 2
    \once \override Staff.HorizontalBracket.direction = #UP
    e''1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
      \fontsize #-5
      \slashed-digit #0 \finger 1 \finger 2
      \finger 4 \finger 5 } \startGroup
```

```
  f''1*1/4^\markup "2)" \stopGroup
}
```



Véase también

Referencia de la notación: Sección 14.3.1 [Diagramas de posiciones para viento madera], página 503.

Fragmentos de código: Sección “Instrumentos de viento” en *Fragmentos de código*.

14.2 Gaita

Esta sección trata de la notación común para gaita.

14.2.1 Definiciones para la gaita

LilyPond contiene definiciones especiales para la música de gaita escocesa de las tierras altas; para usarlas, escriba

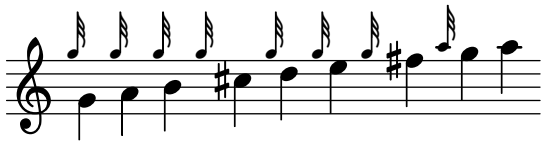
```
\include "bagpipe.ly"
```

al principio del archivo de entrada. De esta forma podrá escribir las notas de adorno especiales que son usuales en la música de gaita, mediante instrucciones breves. Por ejemplo, puede escribir `\taor` en lugar de

```
\grace { \small G32[ d G e] }
```

`bagpipe.ly` también contiene definiciones de alturas para las notas de la gaita en las octavas adecuadas, de forma que no se tenga que preocupar por `\relative` o por `\transpose`.

```
\include "bagpipe.ly"
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



La música de gaita utiliza siempre la tonalidad de Re mayor (aunque esto no es completamente cierto). Sin embargo, puesto que es la única tonalidad que se puede utilizar, normalmente la armadura no se escribe. Para configurar esto de forma correcta, comience la música siempre con `\hideKeySignature` (ocultar armadura). Si, por algún motivo, quiere mostrar la armadura, puede usar `\showKeySignature` en su lugar.

Alguna música moderna utiliza digitaciones cruzadas sobre el Do y el Fa para bemolizar estas notas. Se puede indicar esto mediante `c-flat` o `f-flat`. De forma parecida, el sol agudo *Piobaireachd* se puede escribir `g-flat` cuando aparece en música ligera.

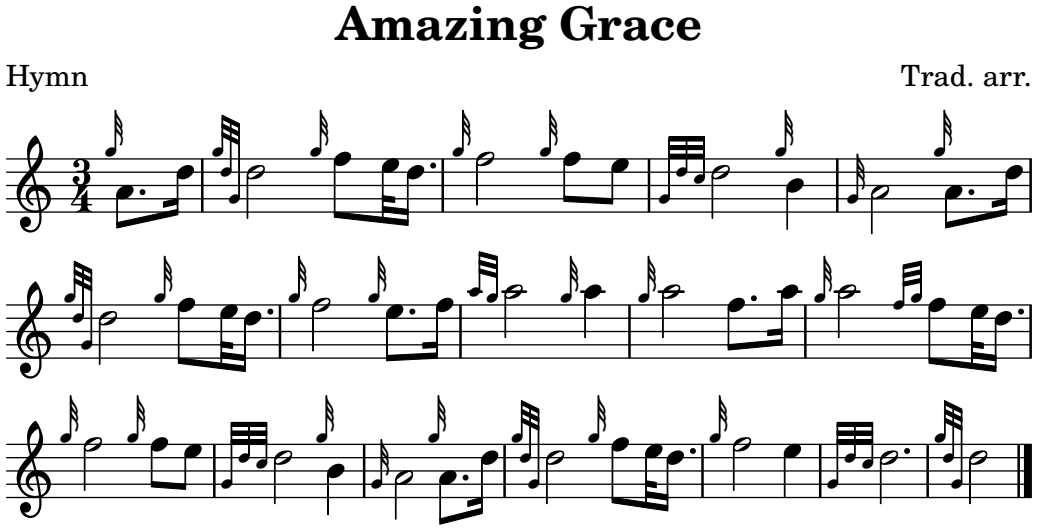
Véase también

Fragmentos de código: Sección “Instrumentos de viento” en *Fragmentos de código*.

14.2.2 Ejemplo de música de gaita

La conocida melodía Amazing Grace tiene este aspecto en notación de gaita.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove Bar_number_engraver }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdblf f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
  \bar "|."
}
```



Véase también

Fragmentos de código: Sección “Instrumentos de viento” en *Fragmentos de código*.

14.3 Instrumentos de viento madera

Esta sección estudia la notación específica para instrumentos de viento madera.

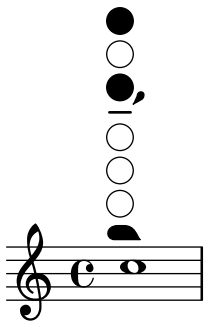
14.3.1 Diagramas de posiciones para viento madera

Se pueden utilizar diagramas de posiciones para los instrumentos de viento madera para indicar la digitación que usar al tocar ciertas notas, y están disponibles para los siguientes instrumentos:

- flautín
- flauta
- oboe
- clarinete
- clarinete bajo
- saxofón
- fagot
- contrafagot

Los diagramas se crean como elementos de marcado:

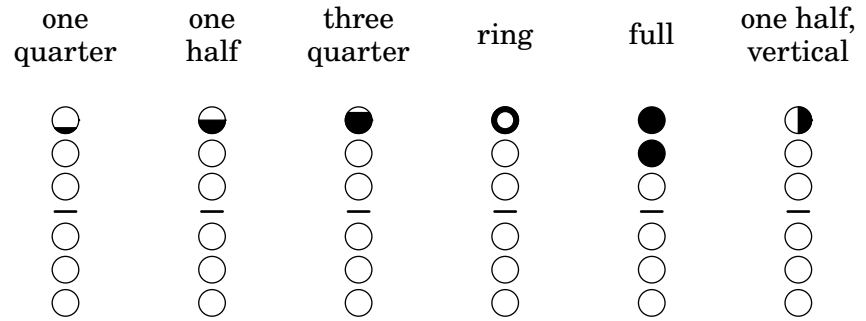
```
c''1^\markup {
  \woodwind-diagram #'piccolo #'((lh . (gis))
                                 (cc . (one three))
                                 (rh . (ees)))
}
```



Las llaves pueden estar abiertas, parcialmente cubiertas, con el anillo pulsado o totalmente apretadas. Se puede especificar el ángulo de las llaves parcialmante tapadas:

```
\markup \override #'(baseline-skip . 22) \column {
  \override #'(baseline-skip . 2.5) \fill-line {
    ""
    \raise #1.2 \center-column { "one" "quarter" }
    \raise #1.2 \center-column { "one" "half" }
    \raise #1.2 \center-column { "three" "quarter" }
    "ring"
    "full"
    \raise #1.2 \center-column { "one half," "vertical" }
    ""
  }
  \fill-line {
    ""
```

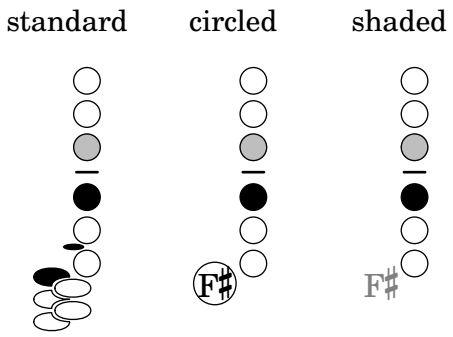
```
      \woodwind-diagram #'flute #'((cc . (one1q))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (one1h))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (one3q))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (oneR))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (oneF two))
                                   (lh . ())
                                   (rh . ()))
      \override #'(woodwind-diagram-details . ((fill-angle . 90)))
      \woodwind-diagram #'flute #'((cc . (one1h))
                                   (lh . ())
                                   (rh . ()))
      ""
    }
  }
```



Los trinos se indican mediante llaves sombreadas, o en modo no gráfico, bien como texto rodeado de un círculo (por omisión) o bien como texto sombreado:

```
\markup {
  \override #'(baseline-skip . 20)
  \override #'(line-width . 50)
  \column {
    \fill-line {
      ""
      "standard"
      "circled"
      "shaded"
      ""
    }
    \fill-line {
      ""
      \woodwind-diagram #'bass-clarinet
                        #'((cc . (threeT four))
                           (lh . ())
                           (rh . (b fis)))
```

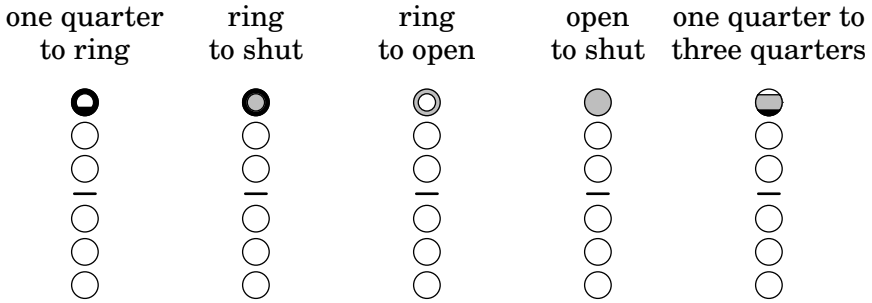
```
      \override #'(graphical . #f)
      \woodwind-diagram #'bass-clarinet
                        #'((cc . (threeT four))
                           (lh . ())
                           (rh . (b fisT)))
      \override #'(graphical . #f)
      \override #'(woodwind-diagram-details
                   . ((text-trill-circled . #f)))
      \woodwind-diagram #'bass-clarinet
                        #'((cc . (threeT four))
                           (lh . ())
                           (rh . (b fisT)))
      ""
    }
  }
}
```



Es posible imprimir una amplia variedad de trinos:

```
\markup {
  \override #'(baseline-skip . 22)
  \column {
    \override #'(baseline-skip . 2.5) \fill-line {
      ""
      \center-column { "one quarter" "to ring" }
      \center-column { "ring" "to shut" }
      \center-column { "ring" "to open" }
      \center-column { "open" "to shut" }
      \center-column { "one quarter to" "three quarters" }
      ""
    }
    \fill-line {
      ""
      \woodwind-diagram #'flute #'((cc . (one1qTR))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (oneTR))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (oneRT))
                                   (lh . ())
                                   (rh . ()))
      \woodwind-diagram #'flute #'((cc . (oneT))
```

```
                                    (lh . ())
                                    (rh . ()))
        \woodwind-diagram #'flute #'((cc . (one1qT3q))
                                    (lh . ())
                                    (rh . ()))
        ""
      }
    }
}
```



La lista de todas las tonalidades y ajustes posibles para un instrumento dado se puede imprimir en la consola usando `#(print-keys-verbose 'flute)` o en el archivo de registro usando `#(print-keys-verbose 'flute (current-error-port))`, aunque no se pueden mostrar en la salida de música impresa.

Es posible crear diagramas nuevos, aunque requiere conocimientos de Scheme y quizá no esté al alcance de todos los usuarios. Los patrones de los diagramas están en `scm/define-woodwind-diagrams.scm` y en `scm/display-woodwind-diagrams.scm`.

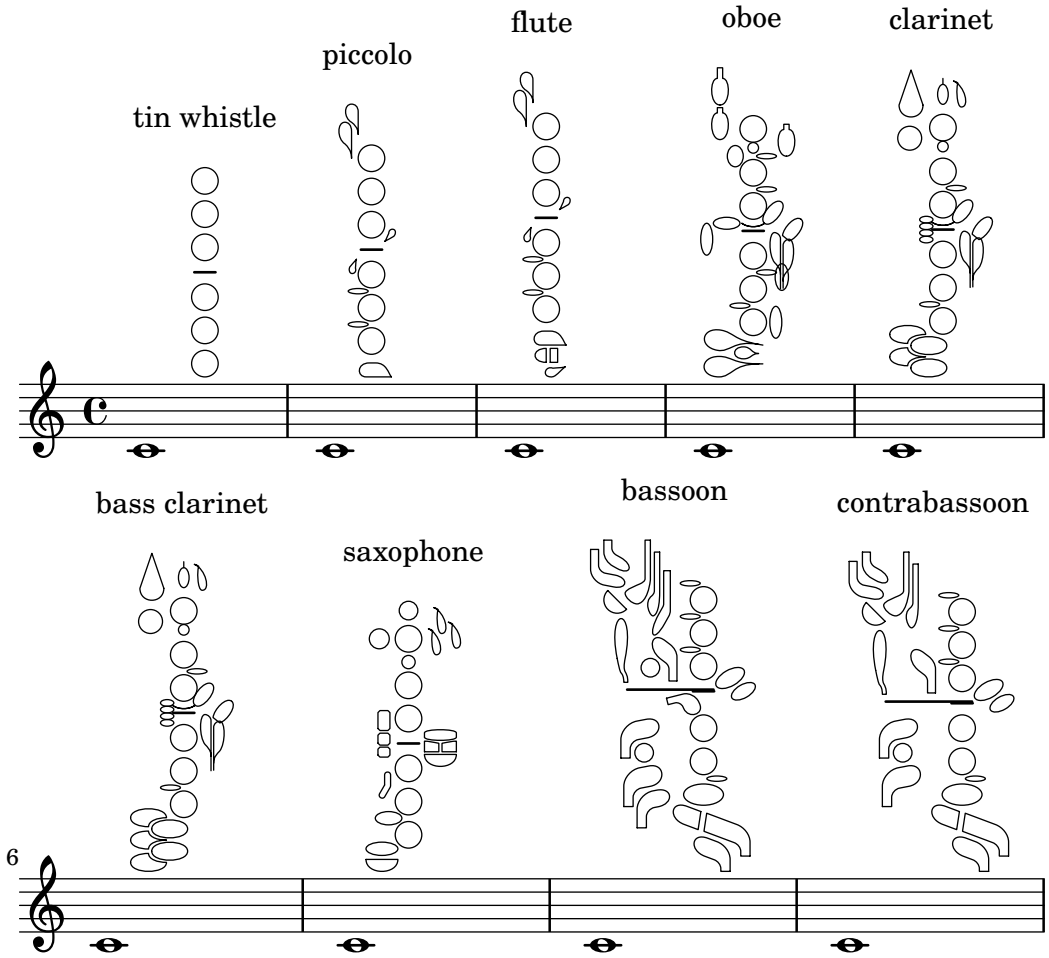
Fragmentos de código seleccionados

Listado de los diagramas para viento madera

El fragmento de música que aparece a continuación presenta todos los diagramas de viento madera que se encuentran definidos en LilyPond por el momento.

```
\relative c' {
  \textLengthOn
  c1^\markup \center-column { "tin whistle"
                              " "
                              \woodwind-diagram #'tin-whistle #'() }
  c1^\markup \center-column { "piccolo"
                              " "
                              \woodwind-diagram #'piccolo #'() }
  c1^\markup \center-column { "flute"
                              " "
                              \woodwind-diagram #'flute #'() }
  c1^\markup \center-column { "oboe"
                              " "
                              \woodwind-diagram #'oboe #'() }
  c1^\markup \center-column { "clarinet"
                              " "
                              \woodwind-diagram #'clarinet #'() }
  \break
```

```
  c1^\markup \center-column { "bass clarinet"
                              " "
                              \woodwind-diagram #'bass-clarinet #'() }
  c1^\markup \center-column { "saxophone"
                              " "
                              \woodwind-diagram #'saxophone #'() }
  c1^\markup \center-column { "bassoon"
                              " "
                              \woodwind-diagram #'bassoon #'() }
  c1^\markup \center-column { "contrabassoon"
                              " "
                              \woodwind-diagram #'contrabassoon #'() }
}

\paper {
  system-system-spacing.padding = 5
}
```

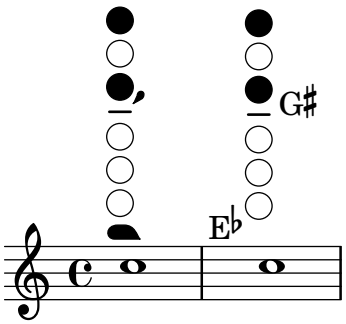


Diagramas para viento madera gráficos y textuales

En muchos casos, las llaves que no están en la columna central se pueden presentar por el nombre de la llave así como de forma gráfica.

```
\relative c'' {
  \textLengthOn
  c1^\markup
```

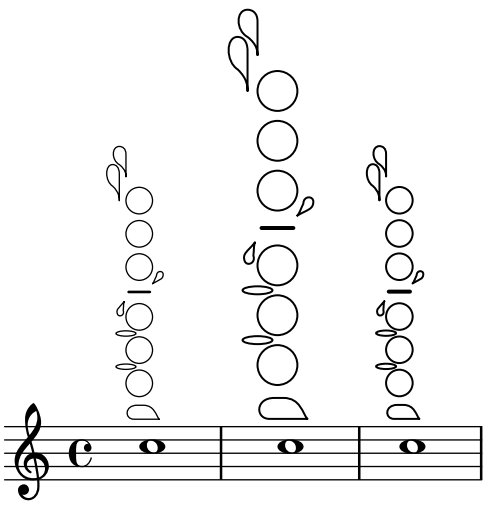
```
    \woodwind-diagram #'piccolo
                      #'((cc . (one three))
                         (lh . (gis))
                         (rh . (ees)))

  c^\markup
    \override #'(graphical . #f)
      \woodwind-diagram #'piccolo
                        #'((cc . (one three))
                           (lh . (gis))
                           (rh . (ees)))
}
```



Modificar el tamaño de los diagramas de viento madera

Se puede cambiar el tamaño y grosor de las líneas de los diagramas de posiciones para instrumentos de viento madera.

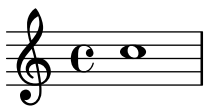
```
\relative c'' {
  \textLengthOn
  c1^\markup
       \woodwind-diagram #'piccolo #'()

  c^\markup \override #'(size . 1.5)
      \woodwind-diagram #'piccolo #'()
  c^\markup \override #'(thickness . 0.15)
      \woodwind-diagram #'piccolo #'()
}
```



Listas de llaves para los diagramas de viento madera

El fragmento de código que aparece a continuación produce una lista de todas las llaves y disposiciones de llave posibles para los diagramas de posiciones de instrumentos de viento madera, tal y como están definidos en `scm/define-woodwind-diagrams.scm`. La lista se muestra en la salida de error estándar stderr, pero no en la música. Si desea una salida por la consola a través de la salida estándar stdout, omita el código `(current-error-port)` de las instrucciones.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c''1}
```



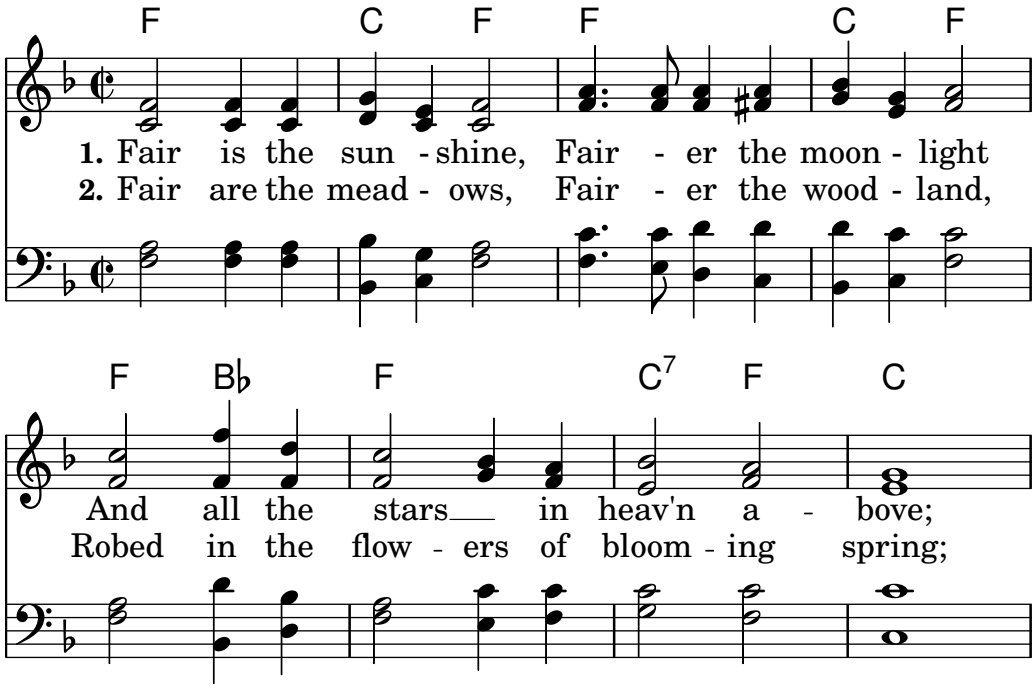
Véase también

Archivos de inicio: `scm/define-woodwind-diagrams.scm`, `scm/display-woodwind-diagrams.scm`.

Fragmentos de código: Sección “Instrumentos de viento” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*, Sección “instrument-specific-markup-interface” en *Referencia de Funcionamiento Interno*.

15 Notación de acordes



Los acordes se pueden escribir en el modo de acordes, que reconoce ciertas convenciones europeas tradicionales de nomenclatura de acordes. También se pueden imprimir los nombres de los acordes. Además se puede imprimir notación de bajo cifrado.

15.1 Modo de acordes

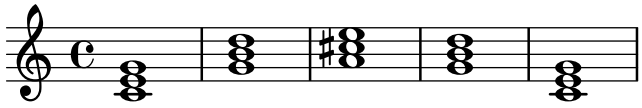
Para introducir acordes se utiliza el modo de acordes. Se usa un indicador de la estructura del acorde en lugar de la altura de sus notas.

15.1.1 Panorámica del modo de acordes

Los acordes se pueden escribir como música simultánea, como se explica en Sección 5.1.1 [Notas en acorde], página 218.

También se pueden escribir los acordes en el "modo de acordes", que es un modo de entrada que se centra en las estructuras de los acordes dentro de la música europea tradicional (donde se conoce como «cifrado americano») más que en las notas concretas. Esto es muy práctico para los que están familiarizados con la utilización de nombres para describir los acordes. Hay más información sobre los distintos modos de entrada en Capítulo 19 [Modos de entrada], página 587.

```
\chordmode { c1 g a g c }
```

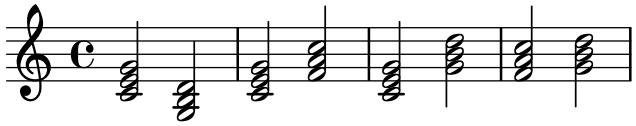


Los acordes escritos con el modo de acordes son elementos musicales, y se pueden transportar igual que los acordes escritos mediante construcciones de música simultánea. `\chordmode` es absoluto, pues `\relative` no tiene ningún efecto sobre los bloques `chordmode`. Sin embargo, dentro de `\chordmode` las alturas absolutas son una octava más agudas que en el modo de notas.

Se pueden mezclar los modos de acorde y de nota en la música secuencial:

```
\relative {
```

```
  <c' e g>2 <g b d>
  \chordmode { c2 f }
  <c e g>2 <g' b d>
  \chordmode { f2 g }
}
```



Véase también

Glosario musical: Sección “chord” en *Glosario Musical*.

Referencia de la notación: Sección 5.1.1 [Notas en acorde], página 218, Capítulo 19 [Modos de entrada], página 587.

Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

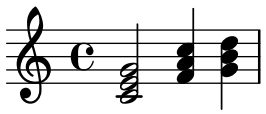
Advertencias y problemas conocidos

No se pueden usar las abreviaturas predefinidas de articulaciones y adornos sobre notas dentro del modo de acordes; véase Sección 3.1.1 [Articulaciones y ornamentos], página 156.

15.1.2 Acordes más usuales

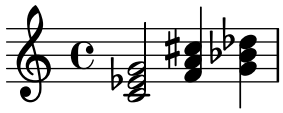
Las tríadas mayores se introducen escribiendo la fundamental y una duración opcional:

```
\chordmode { c2 f4 g }
```



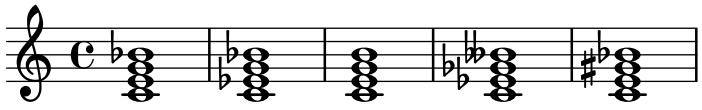
Las tríadas menores, aumentadas y disminuidas se escriben poniendo : y una cadena modificadora de variante después de la duración:

```
\chordmode { c2:m f4:aug g:dim }
```



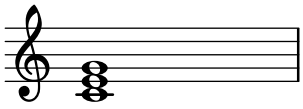
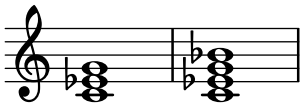
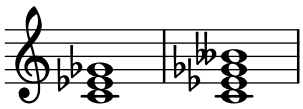
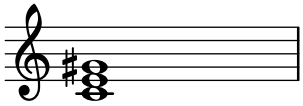
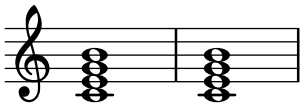
Se pueden crear acordes de séptima:

```
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
```



La tabla que aparece más abajo muestra el efecto de los modificadores de variante sobre los acordes de tríada y de séptima. La séptima añadida de forma predeterminada a los acordes es menor, lo que hace del acorde de séptima dominante el acorde básico de séptima. Todas las alteraciones se consideran en relación con la séptima dominante. Hay una tabla más completa donde aparecen los usos de los modificadores en Sección B.2 [Modificadores de acorde más usuales], página 895.

Modificador	**Acción**	**Ejemplo**

Ninguno	Acción predeterminada; produce una tríada mayor.	
m, m7	Acorde menor. Este modificador baja la tercera.	
dim, dim7	Acorde disminuido. Este modificador baja la tercera, la quinta y, si existe, la séptima.	
aug	Acorde aumentado. Este modificador eleva la quinta.	
maj, maj7	Acorde de séptima mayor. Este modificador añade una séptima elevada. El 7 que sigue a `maj` es opcional. NO utilice este modificador para crear una tríada mayor.	

Véase también

Referencia de la notación: Sección B.2 [Modificadores de acorde más usuales], página 895, Sección 15.1.3 [Acordes extendidos y alterados], página 512.

Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

Advertencias y problemas conocidos

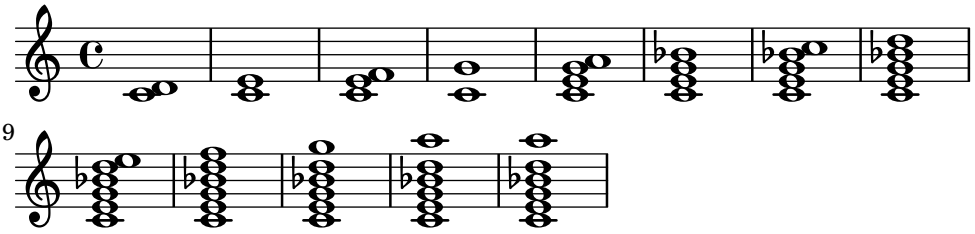
Solo se puede usar un modificador por cada acorde, normalmente sobre la nota más aguda del mismo. Los acordes con más de un modificador se analizan sin producir errores ni advertencias, pero el resultado es impredecible. Los acordes que no se pueden conseguir con un solo modificador se deben alterar en sus notas individuales como se describe en Sección 15.1.3 [Acordes extendidos y alterados], página 512.

15.1.3 Acordes extendidos y alterados

Se pueden crear estructuras de acorde de complejidad arbitraria dentro del modo de acordes. Se puede usar la cadena modificadora para extender un acorde, añadir o quitar notas, elevar o bajar notas del acorde y añadir un bajo distinto o crear una inversión.

El primer número que sigue al : se considera que es el ámbito del acorde. El acorde se construye secuencialmente añadiendo terceras a la fundamental hasta que se alcanza el número especificado. Observe que la séptima añadida como parte de un acorde extendido es la séptima menor, no mayor. Si el ámbito no es una tercera (p.ej. 6), se añaden terceras hasta la tercera más alta inferior al ámbito, y después se añade la nota del ámbito. El mayor valor posible para el ámbito es 13. Cualquier valor mayor se interpreta como 13.

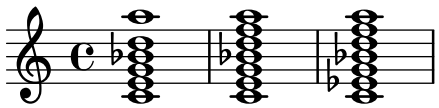
```
\chordmode {
  c1:2 c:3 c:4 c:5
  c1:6 c:7 c:8 c:9
  c1:10 c:11 c:12 c:13
  c1:14
}
```



Como excepción especial, `c:5` produce un 'power chord', que tiene solamente la fundamental y la quinta.

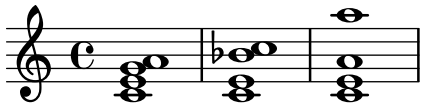
Puesto que una oncena sin alteración no suena bien cuando se combina con una trecena sin alteración, se elimina la oncena de los acordes de `:13` mayores (a no ser que se añada explícitamente).

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



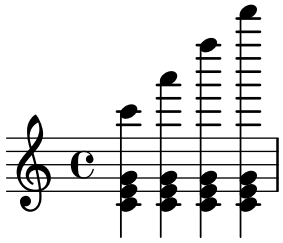
Se pueden añadir notas individuales a un acorde. Las adiciones siguen el ámbito y van prefijadas por un punto (`.`). La séptima normal que se añade a un acorde es la séptima menor, no mayor.

```
\chordmode {
  c1:3.5.6 c:3.7.8 c:3.6.13
}
```



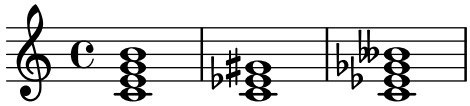
Las notas añadidas pueden ser tan agudas como se desee.

```
\chordmode {
  c4:3.5.15 c:3.5.20 c:3.5.25 c:3.5.30
}
```



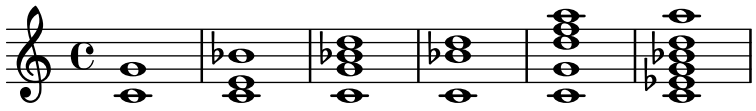
Las notas añadidas a los acordes se pueden alterar mediante la adición de los sufijos `-` o `+` al número. Para alterar una nota que se ha incluido automáticamente como parte de la estructura básica del acorde, añádala como una nota alterada.

```
\chordmode {
  c1:7+ c:5+.3- c:3-.5-.7-
}
```



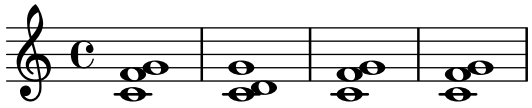
Una nota que se quiere suprimir de un acorde se indica dentro de la cadena modificadora prefijándola por un acento circunflejo ^. Solo se permite una supresión con ^ dentro de una cadena modificadora.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7 c:m13^11
}
```



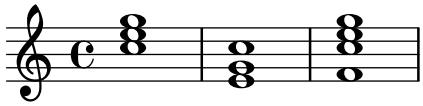
Se puede añadir el modificador sus a la cadena modificadora para crear acordes suspendidos. Esto elimina la tercera del acorde. Escriba bien 2 o bien 4 para añadir la segunda o la cuarta al acorde, respectivamente. Cuando sus va seguido de una segunda o una cuarta, es equivalente a ^3, y en caso contrario a sus4, o sea, 5.4.

```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4
}
```



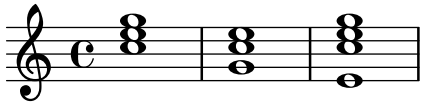
Se pueden especificar notas añadidas en el bajo (colocar en el bajo una nota distinta de la fundamental del acorde) mediante la adición de */nota* al acorde.

```
\chordmode {
  c'1 c'/e c'/f
}
```



Si la nota añadida ya forma parte del acorde, esto se puede aprovechar para imprimir inversiones de acorde, en cuyo caso la nota no se añade realmente sino que se desplaza a la base del acorde. Sin embargo, se puede tratar como una nota añadida (imprimiéndose así dos veces), mediante la sintaxis */+nota*.

```
\chordmode {
  c'1 c'/g c'/+e
}
```



Las inversiones y realizaciones de acorde automáticas se explican en Sección 15.1.4 [Inversiones de acorde y realizaciones especiales], página 515.

Los modificadores de acorde que se pueden utilizar para producir una amplia variedad de acordes estándar se muestran en Sección B.2 [Modificadores de acorde más usuales], página 895.

Véase también

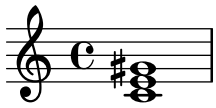
Referencia de la notación: Sección 15.1.4 [Inversiones de acorde y realizaciones especiales], página 515, Sección B.2 [Modificadores de acorde más usuales], página 895.

Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

Advertencias y problemas conocidos

Las apariciones posteriores de una nota del acorde sobreescriben a las apariciones anteriores. Lo que sigue produce simplemente el acorde aumentado, porque 5+ se interpreta en último lugar.

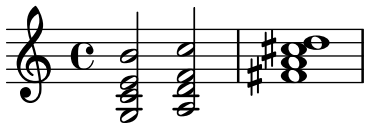
```
\chordmode { c1:3.5.5-.5+ }
```



15.1.4 Inversiones de acorde y realizaciones especiales

Además de los modificadores de acorde y las notas añadidas en el bajo, se pueden usar varias funciones para imprimir automáticamente acordes en una inversión o realización especial (por ejemplo la realización conocida como 'drop 2' que se usa habitualmente en el jazz).

```
\chordmode {
  \dropNote 2 {
    c2:maj7 d:m7
  }
  \invertChords 1 d1:maj7
}
```



A diferencia de las notas añadidas en el bajo que se ven en Sección 15.1.3 [Acordes extendidos y alterados], página 512, esto afecta solamente a la manera en que los acordes se imprimen en un pentagrama, y no a los nombres de acorde escritos con letras. Es más, estas funciones se pueden usar no solo en el modo de acordes sino también con la construcción de acorde habitual <. . .> que se explica en Sección 5.1.1 [Notas en acorde], página 218.

Véase también

Referencia de la notación: Sección 15.1.3 [Acordes extendidos y alterados], página 512, Sección 5.1.1 [Notas en acorde], página 218.

Fragmentos de código: Sección "Notación de los acordes" en *Fragmentos de código*.

15.2 Imprimir los acordes

Los acordes se pueden imprimir por su nombre, además de la impresión estándar como notas sobre un pentagrama.

15.2.1 Impresión de los nombres de acorde

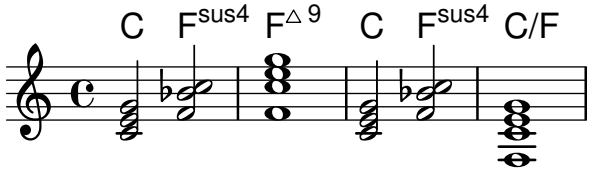
Los nombres de acorde se imprimen dentro del contexto `ChordNames`:

```
\new ChordNames {
  \chordmode {
    c2 f4. g8
  }
}
```

C F G

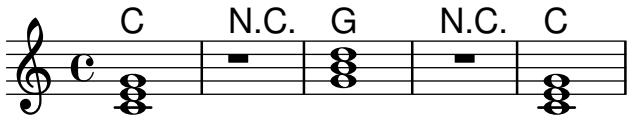
Los acordes se pueden escribir como notas simultáneas o a través del uso del modo de acordes. El nombre de acorde que se imprime es el mismo independientemente del modo de entrada, a no ser que existan inversiones o bajos añadidos:

```
chordmusic = \relative {
  <c' e g>2 <f bes c>
  <f c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
<<
  \new ChordNames {
    \chordmusic
  }
  {
    \chordmusic
  }
>>
```



Cuando se pasan a un contexto `ChordNames`, los silencios (incluidos los silencios multicompás) hacen que se imprima el texto `noChordSymbol` (símbolo de N.C., Sin Acorde).

```
myChords = \chordmode {
  c1
  r1
  g1
  R1
  c1
}

<<
  \new ChordNames \myChords
  \new Staff \myChords
>>
```



`\chords { ... }` es una forma abreviada de escribir `\new ChordNames \chordmode { ... }`.

```
\chords {
  c2 f4.:m g8:maj7
}
```



```
\new ChordNames {
  \chordmode {
```

```
    c2 f4.:m g8:maj7
  }
}
```

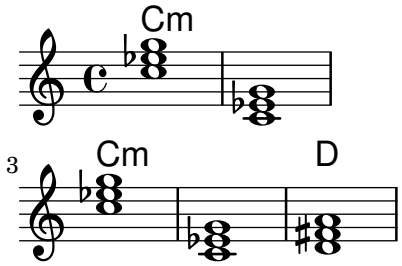


Fragmentos de código seleccionados

Imprimir los acordes cuando se produce un cambio

De forma predeterminada se imprimen todos los acordes que aparecen en el código de entrada. Se puede modificar este comportamiento de tal manera que los acordes se impriman exclusivamente al comienzo de las líneas o cuando cambia el acorde.

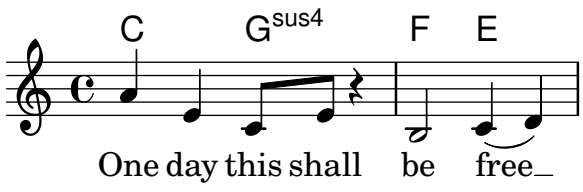
```
harmonies = \chordmode {
  c'1:m c:m \break
  c'1:m c:m d
}

<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies
  }
  \new Staff {
    \harmonies
  }
>>
```



Hoja guía de acordes o lead sheet sencilla

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet».

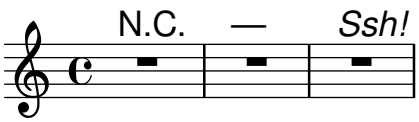
```
<<
  \chords { c2 g:sus4 f e }
  \new Staff \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Personalización del símbolo de Sin Acorde

Por omisión, los silencios en un contexto `ChordNames` producen la impresión del texto "N.C.". Este elemento de marcado se puede personalizar mediante el establecimiento de la propiedad de contexto `noChordSymbol`.

```
<<
  \chords {
    R1
    \set noChordSymbol = "---"
    R1
    \set noChordSymbol = \markup \italic "Ssh!"
    R1
  }
  {
    R1*3
  }
>>
```



Véase también

Glosario musical: Sección "chord" en *Glosario Musical*.

Referencia de la notación: Sección 5.2.6 [Escribir música en paralelo], página 241.

Fragmentos de código: Sección "Notación de los acordes" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "ChordNames" en *Referencia de Funcionamiento Interno*, Sección "ChordName" en *Referencia de Funcionamiento Interno*, Sección "Chord_name_engraver" en *Referencia de Funcionamiento Interno*, Sección "Volta_engraver" en *Referencia de Funcionamiento Interno*, Sección "Bar_engraver" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Los acordes que contienen inversiones o bajos alterados no reciben un nombre adecuado si se escriben usando música simultánea.

15.2.2 Nombres de acorde personalizados

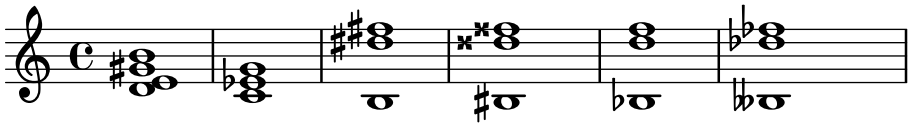
No hay un sistema único para nombrar acordes. Las distintas tradiciones musicales usan distintos nombres para el mismo conjunto de acordes. También hay distintos símbolos impresos para un nombre de acorde dado. Los nombres y símbolos que se imprimen para los nombres de acorde dados se pueden personalizar.

El esquema de nomenclatura de acordes predeterminado es un sistema para música de Jazz propuesto por Klaus Ignatzek (véase Sección "Lista de referencias bibliográficas" en *Ensayo*. Se pueden implementar otros sistemas de nomenclatura de acordes mediante funciones de Scheme, véase el ejemplo en Sección "Chord names alternative" en *Fragmentos de código*. Hay una lista de notaciones comunes para acordes de jazz en la tabla de la sección Sección B.1 [Cuadro de nombres de acordes], página 895.

Se puede trucar fácilmente el sistema de nomenclatura de acordes predeterminado. Para empezar, ciertas órdenes predefinidas permiten imprimir la nota fundamental en distintos idiomas. Entre ellas están `\germanChords`, `\italianChords`, `\frenchChords` y `\englishChords`.

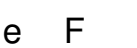
Un caso especial es `\semiGermanChords`, que está pensado para un uso didáctico en países de habla alemana al mismo tiempo que se dirigen a un público internacional. Imprime la nota alemana B (si bemol) con el símbolo de alteración del bemol; el resto es idéntico a `\germanChords`.

default	E/D	Cm	B/B	B♯/B♯	B♭/B♭	B𝄫/B𝄫
German	E/d	Cm	H/h	H♯/his	B/b	H𝄫/heses
semi-German	E/d	Cm	H/h	H♯/his	B♭/b	H𝄫/heses
Italian	Mi/Re	Do m	Si/Si	Si♯/Si♯	Si♭/Si♭	Si𝄫/Si𝄫
French	Mi/Ré	Do m	Si/Si	Si♯/Si♯	Si♭/Si♭	Si𝄫/Si𝄫
English	E/D	Cm	B/B	B♯/B♯	B♭/B♭	B𝄫/B𝄫



Algunos cancioneros alemanes indican los acordes menores como letras minúsculas, sin el sufijo 'm'. Esto se puede conseguir estableciendo la propiedad `chordNameLowercaseMinor`:

```
\chords {
  \set chordNameLowercaseMinor = ##t
  c2 d:m e:m f
}
```



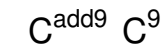
También se puede afinar la presentación de los nombres de acorde a través del ajuste de las siguientes propiedades.

`chordRootNamer`
: La configuración predeterminada es imprimir el nombre de acorde como una letra para la fundamental con una alteración opcional. La transformación de la nota en la letra se realiza por parte de esta función. Los nombres de nota especiales (por ejemplo, la 'H' alemana para un acorde de Si) se pueden producir almacenando una función diferente en esta propiedad.

`majorSevenSymbol`
: Esta propiedad contiene el objeto de marcado que identifica a un acorde de séptima mayor, que se imprimirá justo después de la nota fundamental del acorde. Los valores predeterminados son `whiteTriangleMarkup` (triángulo blanco, el valor por omisión) y `blackTriangleMarkup` (triángulo negro).

`additionalPitchPrefix`
: Si el nombre del acorde contiene notas añadidas, normalmente llevan el prefijo 'add', que es la opción predeterminada de LilyPond. Este prefijo se puede modificar o eliminar.

```
\new ChordNames {
  <c e g d'>    % add9
  \set additionalPitchPrefix = ""
  <c e g d'>    % add9
}
```

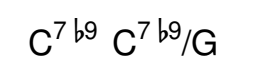


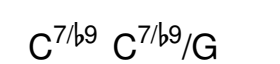
chordNoteNamer
: Si un nombre de acorde contiene notas añadidas distintas a la fundamental (p. ej. una nota añadida en el bajo), se usa la función asignada a esta propiedad para imprimir dichas notas. Si no está establecida o está fijada a `'()`, use en su lugar la función asignada a `chordRootNamer`.

  Como ejemplo, la función `chord-name:german-lowercase-name-markup` se use por parte de `\germanChords` para imprimir las notas del bajo en letras minúsculas.

chordNameSeparator
: Las diferentes partes del nombre de un acorde se separan normalmente mediante un pequeño espacio en blanco. Mediante el ajuste de `chordNameSeparator`, podemos usar cualquier marcado que deseemos como separador. Esto no afecta al separador entre un acorde y la nota del bajo: para personalizar eso, utilice `slashChordSeparator`.

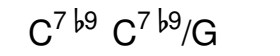
```
\chords {
  c4:7.9- c:7.9-/g
  \set chordNameSeparator = \markup { "/" }
  \break
  c4:7.9- c:7.9-/g
}
```

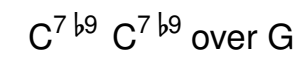




slashChordSeparator
: Los acordes se pueden tocar sobre una nota del bajo distinta a la fundamental. Se llaman “inversiones” o “acordes de barra inclinada” porque la manera predeterminada de efectuar su notación es con una barra inclinada entre el acorde y la nota del bajo. Por tanto, el valor de `slashChordSeparator` es, por omisión, una barra inclinada, pero podemos cambiarlo a cualquier otro elemento de marcado.

```
\chords {
  c4:7.9- c:7.9-/g
  \set slashChordSeparator = \markup { " over " }
  \break
  c4:7.9- c:7.9-/g
}
```



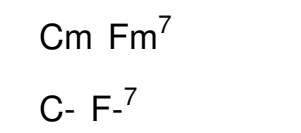


chordNameExceptions
: Esta propiedad es una lista de parejas. El primer elemento de cada pareja es un conjunto de notas utilizadas para identificar los elementos presentes en el acorde. El segundo elemento es un elemento de marcado que sigue a la salida de `chordRootNamer` para crear el nombre del acorde.

minorChordModifier
: La notación de los acordes menores se suele hacer por medio del sufijo ‘m’ a la derecha de la letra que indica la fundamental del acorde (es la opción predeterminada

de LilyPond). Sin embargo, otras convenciones dan preferencia a sufijos diferentes, como un signo menos.

```
\chords {
  c4:min f:min7
  \set minorChordModifier = \markup { "-" }
  \break
  c4:min f:min7
}
```



chordPrefixSpacer
: El modificador de los acordes menores tal y como viene determinado por minorChordModifier se imprime de manera predeterminada inmediatamente a la derecha de la fundamental. Se puede insertar un espacio entre la fundamental y el modificador estableciendo chordPrefixSpacer (utilizado, por ejemplo, en el caso de \italianChords). No se inserta ningún espacio si la fundamental es una nota alterada.

Instrucciones predefinidas

\whiteTriangleMarkup, \blackTriangleMarkup, \germanChords, \semiGermanChords, \italianChords, \frenchChords, \englishChords.

Fragmentos de código seleccionados

Excepciones para los nombres de acorde

La propiedad chordNameExceptions almacena una lista de excepciones para los nombres de acorde, para tratar casos que no están cubiertos, o bien que se tratan incorrectamente.

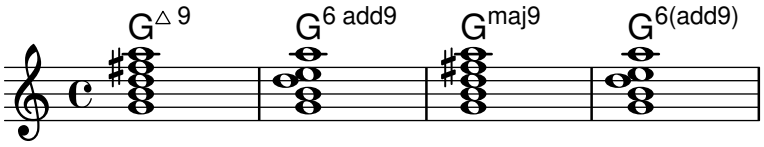
Los nombres de acorde predeterminados que se usan por LilyPond siguen las reglas dadas en el libro de Klaus Ignatzek's "Die Jazzmethode für Klavier 1"; el algoritmo para convertir acordes en nombres de acorde puede verse dentro del archivo scm/chord-ignatzek-names.scm. Las reglas adicionales se dan como excepciones de acordes y se almacenan dentro de la variable ignatzekExceptions, como está establecido en el archivo ly/chord-modifiers-init.ly.

Este fragmento de código modifica estas excepcioens en tres pasos.

1. Preparación de una música con acordes y elementos de marcado asociados. Por convención, la fundamental de cada acorde debe ser Do.
2. Llamar a la función de Scheme sequential-music-to-chord-exceptions para crear una lista de excepciones nueva, y después concatenarla con las existentes. Como ignatzekExceptions está preparado con el segundo parámetro de su función fijado al valor #t (para ignorar la fundamental de los acordes), tenemos que hacer lo mismo.
3. Registrar la nueva lista de excepciones.

```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% Step 2: Create extended exception list.
chExceptions =
```

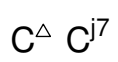
```
#(append (sequential-music-to-chord-exceptions chExceptionMusic #t)
         ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  % Step 3: Register extended exception list.
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

<<
  \new ChordNames \theMusic
  \new Voice \theMusic
>>

\layout {
  line-width = 10\cm
  ragged-right = ##f
}
```



Nombre de acorde maj7

La presentación del acorde de séptima mayor se puede ajustar mediante la propiedad de contexto `majorSevenSymbol`.

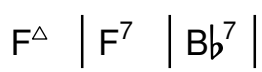
```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



Añadir barras de compás al contexto de nombres de acorde (ChordNames)

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde `ChordNames`, incluya el grabador `Bar_engraver`.

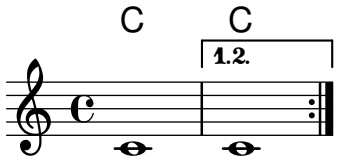
```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```



Corchetes de primera y segunda vez debajo de los acordes

Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



Modificación del separador de acordes

Se puede establecer el separador entre las distintas partes del nombre de un acorde para que sea cualquier elemento de marcado.

```
\chords {
  c:7sus4
  \set chordNameSeparator = \markup { \typewriter | }
  c:7sus4
}
```


Véase también

Referencia de la notación: Sección B.1 [Cuadro de nombres de acordes], página 895, Sección B.2 [Modificadores de acorde más usuales], página 895.

El ensayo sobre grabado musical automatizado: Sección “Lista de referencias bibliográficas” en *Ensayo*.

Archivos instalados: `scm/chords-ignatzek-names.scm`, `scm/chord-entry.scm`, `ly/chord-modifiers-init.ly`.

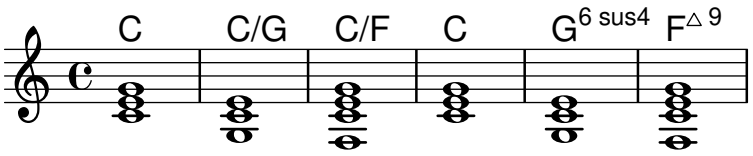
Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

Advertencias y problemas conocidos

Los nombres de acorde se determinan a partir de las notas que están presentes en el mismo y de la información acerca de la estructura del acorde que ha podido ser introducida en `\chordmode`. Si el método de las notas simultáneas se usa para introducir los acordes, pueden resultar nombres no deseados a partir de las inversiones o de las notas del bajo.

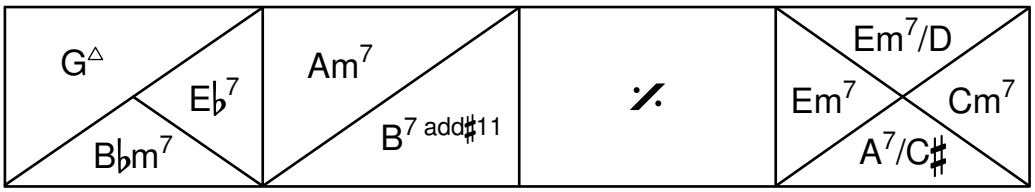
```
myChords = \relative c' {
```

```
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \myChords }
  \new Staff { \myChords }
>>
```



15.2.3 Rejillas de acordes

En algunos países europeos, sobre todo en Francia, los músicos de jazz usan lo que se conoce como 'rejillas de acordes', que imprimen los acordes visualmente poniéndolos dentro de rectángulos.



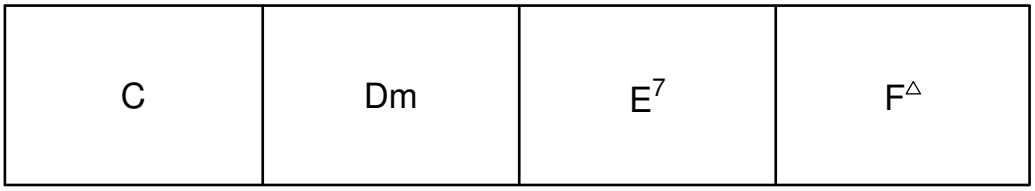
Aunque se omiten durante el resto de este apartado por brevedad, se recomienda usar los siguientes ajustes de \paper para las rejillas de acorde:

```
\paper {
  indent = 0
  ragged-right = ##f
}
```

El `indent = 0` garantiza que la primera línea no presenta un sangrado como lo haría normalmente (véase Sección 26.5.3 [Variables de \paper para desplazamientos y sangrados], página 681). `ragged-right = ##f` es necesario para rejillas de una sola línea para forzar que ocupe todo el ancho de la página; véase Sección 26.5.1 [Variables de \paper para la anchura y los márgenes], página 679.

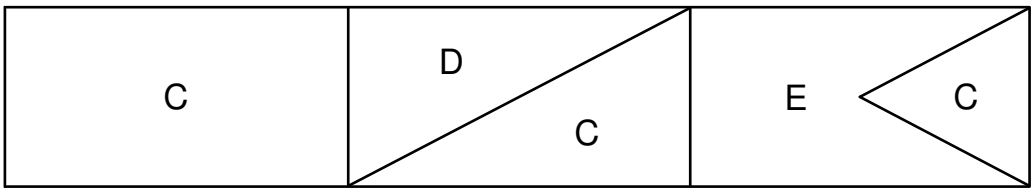
Para crear una rejilla de acordes, hay que crear una instancia del contexto `ChordGrid`.

```
\new ChordGrid \chordmode { c1 d1:m e1:7 f1:7+ }
```

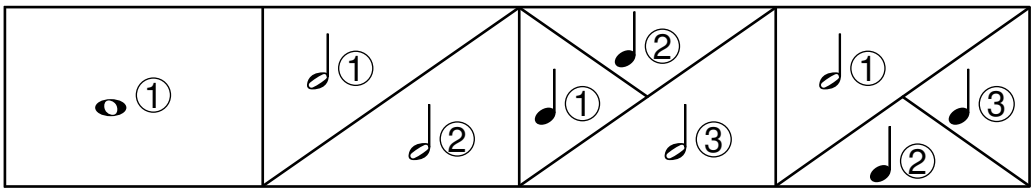


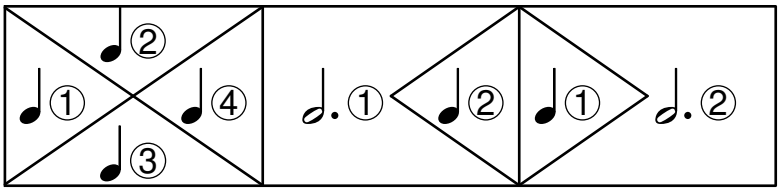
Cada cuadrado se subdivide automáticamente.

```
\new ChordGrid \chordmode {
  c1
  d2 c2
  e2. c4
}
```



Los acordes que se duran un compás entero se centran dentro de su cuadrado. Los acordes que duran medio compás ocupan medio cuadrado, y los que duran un cuarto de compás ocupan un cuarto de cuadrado. Esta imagen a modo de resumen presenta las reglas por omisión para la subdivisión del cuadrado:

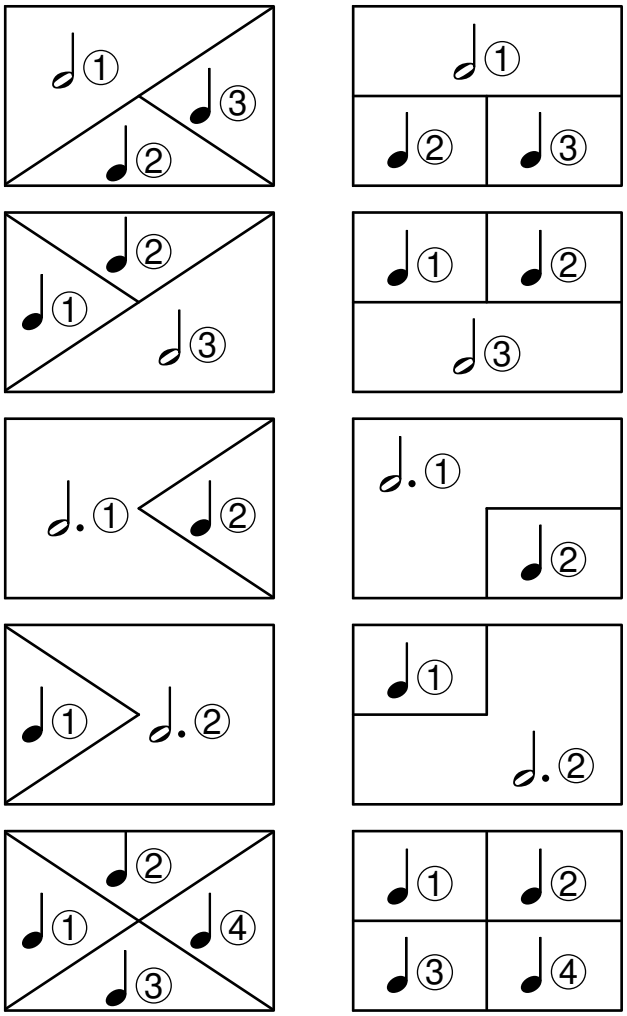




La instrucción `\medianChordGridStyle` modifica la impresión predeterminada de los cuadrados con divisiones del compás particulares, de forma que usen el estilo recomendado por Philippe Baudoin en su libro *Jazz, mode d'emploi* ("Jazz, instrucciones de uso").

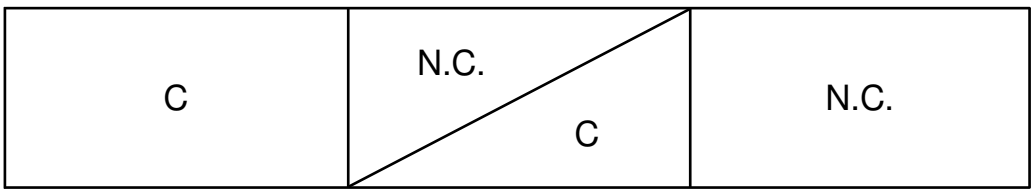
```
\layout {
  \context {
    \ChordGrid
    \medianChordGridStyle
  }
}
```

Default style `\medianChordGridStyle`



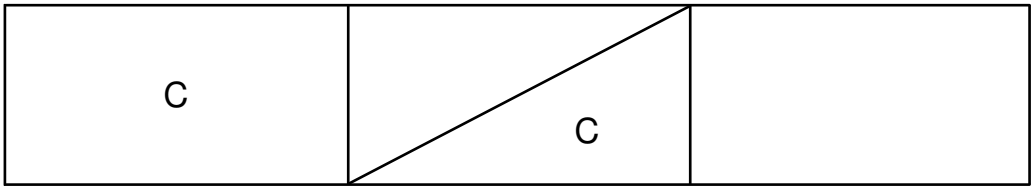
En las rejillas de acordes, los silencios hacen que se imprima el símbolo `noChordSymbol`, igual que en un contexto `ChordNames` normal (véase Sección 15.2.1 [Impresión de los nombres de acorde], página 515).

```
\new ChordGrid \chordmode { c1 r2 c2 R1 }
```



Los Skips o silencios de separación producen espacio vacío. TheyPueden ocupar parte de un cuadrado.

```
\new ChordGrid \chordmode { c1 s2 c2 s1 }
```



Fragmentos de código seleccionados

Personalización del estilo de las rejillas de acordes

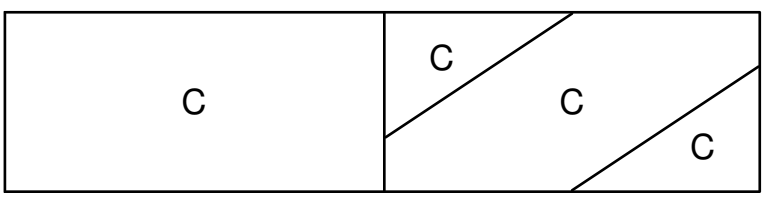
Se pueden definir divisiones personalizadas de los cuadrados de acordes a través de las propiedades `measure-division-lines-alist` y `measure-division-chord-placement-alist` de `ChordSquare`. Ambas son listas de asociación. Sus claves de asociación son divisiones de compás, concretamente listas que dan la fracción del compás que representa cada acorde (o silencio o separación). De forma más precisa, una lista-A de divisiones de compases consta de números positivos exactos que suman la unidad, por ejemplo `'(1/2 1/4 1/4)`. El requisito de la exactitud implica que p.ej. `1/2` es válido pero no `0.5`.

Los valores de `measure-division-lines-alist` son listas de líneas, que están representadas como `(x1 y1 x2 y2)`. La línea comienza en el punto `(x1 . y1)` y finaliza en `(x2 . y2)`. Las coordenadas se expresan en la escala [-1, 1] relativa a las dimensiones del cuadrado.

Los valores en `measure-division-chord-placement-alist` son listas de pares `(x . y)` que dan la ubicación de los compases respectivos.

Este ejemplo define un estilo de rejilla de acordes peculiar que tiene una regla para los compases divididos en tres partes iguales.

```
\paper {
  line-width = 10\cm
  ragged-right = ##f
}

\new ChordGrid \with {
  \override ChordSquare.measure-division-lines-alist =
    #'(((1) . ())
       ((1/3 1/3 1/3) . ((-1 -0.4 0 1) (0 -1 1 0.4))))
  \override ChordSquare.measure-division-chord-placement-alist =
    #'(((1) . ((0 . 0)))
       ((1/3 1/3 1/3) . ((-0.7 . 0.5) (0 . 0) (0.7 . -0.5))))
}
\chordmode {
  \time 3/4
  c2.
  c4 c4 c4
}
```

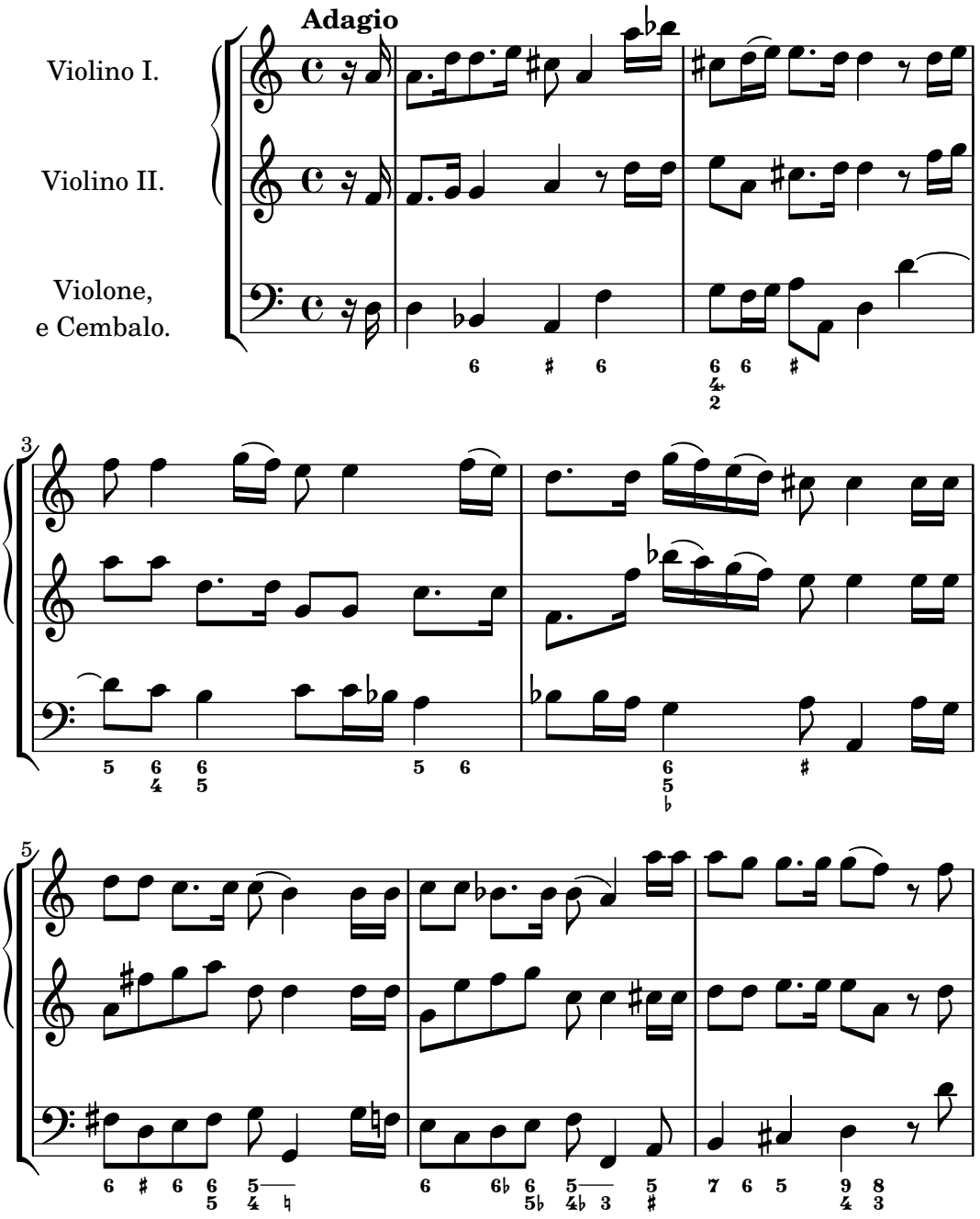


Véase también

Glosario musical: Sección “chord grid” en *Glosario Musical*.

Referencia de funcionamiento interno: Sección “ChordGrid” en *Referencia de Funcionamiento Interno*, Sección “ChordGridScore” en *Referencia de Funcionamiento Interno*, Sección “GridChordName” en *Referencia de Funcionamiento Interno*, Sección “ChordSquare” en *Referencia de Funcionamiento Interno*, Sección “Grid_chord_name_engraver” en *Referencia de Funcionamiento Interno*, Sección “Chord_square_engraver” en *Referencia de Funcionamiento Interno*.

15.3 Bajo cifrado



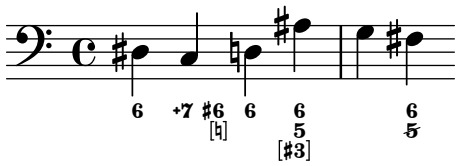
Se puede imprimir notación de bajo cifrado.

15.3.1 Introducción al bajo cifrado

LilyPond contempla el bajo cifrado, también conocido como *basso continuo*.

```
<<
```

```
  \new Voice { \clef bass dis4 c d ais g fis}
  \new FiguredBass {
    \figuremode {
      <6>4 <7\+>8 <6+ [_!]> <6>4 <6 5 [3+]> |
      <_>4 <6 5/>4
    }
  }
>>
```



El soporte para bajo cifrado se compone de dos partes: hay un modo de entrada, presentado por `\figuremode`, que admite la entrada de las cifras del bajo, y hay un contexto llamado `FiguredBass` que se ocupa de la impresión de los objetos del tipo `BassFigure`. El bajo cifrado también se puede imprimir dentro de contextos de `Staff`.

`\figures { ... }` es una notación abreviada de `\new FiguredBass \figuremode { ... }`.

Aunque el soporte para bajo cifrado puede parecerse superficialmente al de los acordes, realmente es mucho más simple. El modo `\figuremode` simplemente almacena las cifras y el contexto `FiguredBass` las imprime tal y como se han introducido. No hay ninguna conversión a notas.

Véase también

Glosario musical: Sección “figured bass” en *Glosario Musical*.

Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

15.3.2 Escritura del bajo cifrado

Se usa `\figuremode` para cambiar el modo de entrada al modo de cifras. Véase Capítulo 19 [Modos de entrada], página 587, para más información sobre los distintos modos de entrada.

En el modo de cifras, un grupo de cifras del bajo está delimitado mediante ángulos simples, ‘<’ y ‘>’. La duración se escribe después del ‘>’.

```
\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}
```



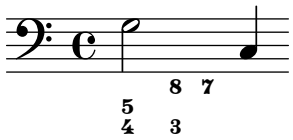
Se pueden usar alteraciones accidentales (incluso becuadros) para modificar grados de la escala. Se escriben añadiendo ‘+’ (para los sostenidos), ‘-’ (para los bemoles) o ‘!’ (para los becuadros) después del número. Para obtener alteraciones dobles, se aplica el modificador dos veces seguidas. Con frecuencia, para la modificación del tercer grado se omite el número, lo que puede obtenerse utilizando ‘_’ en lugar de un número.

```
\figures {
  <7! 6+ 4-> <5++> <3--> <_+> <7 _!>
}
```



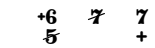
Si se usa sin alteración, '_' crea una cifra vacía que de todas formas ocupa cierto espacio. Se puede utilizar para controlar el apilamiento de cifras del bajo.

```
<<
  {
    \clef bass
    g2 c4
  }
  \figures {
    <_ 5 4>4 <8 _ 3>8 <7>
  }
>>
```



Se pueden indicar elementos del acorde aumentados y disminuidos:

```
\figures {
  <6\+ 5/> <7/> <7 _\+>
}
```

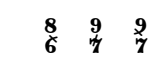


También está disponible una barra invertida que criza a una cifra.

```
\figures {
  <5> <5\\>
}
```

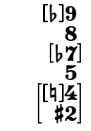

Para algunas cifras están disponibles glifos de barra invertida especiales.

```
\figures {
  <8 6\\> <9 7\\> <9\\ 7>
}
```



Se pueden añadir corchetes alrededor de las alteraciones, las cifras y los grupos de cifras consecutivas.

```
\figures {
  <9[-] 8 [7-] 5 [4[!] 2+]>
}
```



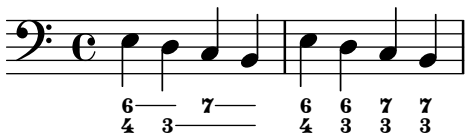
Se puede insertar cualquier elemento de marcado de texto como una cifra:

```
\figures {
  <\markup { \fontsize #-5 \number 6 \teeny \super (1) } 5>
}
```



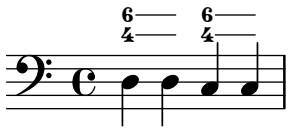
Se pueden usar líneas de continuación para indicar cifras que se repiten:

```
<<
  {
    \clef bass
    e4 d c b,
    e4 d c b,
  }
  \figures {
    \bassFigureExtendersOn
    <6 4>4 <6 3> <7 3> <7 3>
    \bassFigureExtendersOff
    <6 4>4 <6 3> <7 3> <7 3>
  }
>>
```



En este caso, las líneas de extensión sustituyen a las cifras existentes, a no ser que las líneas de continuación hayan sido terminadas explícitamente con `\!`.

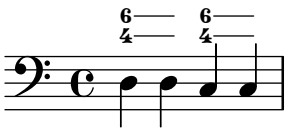
```
<<
  \figures {
    \bassFigureExtendersOn
    <6 4>4 <6 4> <6\! 4\!> <6 4>
  }
  {
    \clef bass
    d4 d c c
  }
>>
```



La siguiente tabla resume los modificadores de cifrado disponibles.

modificador	**propósito**	**ejemplo**
+, -, !	alteraciones	
\+, /	aumentaciones y disminuciones	
\\	elevado un semitono	

\! fin de línea de continuación



Instrucciones predefinidas

\bassFigureExtendersOn, \bassFigureExtendersOff.

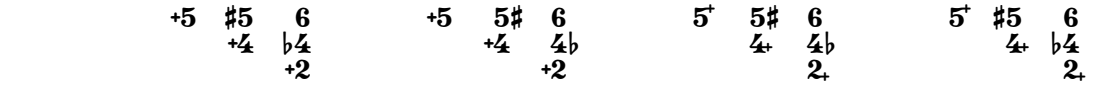
Fragmentos de código seleccionados

Cambiar las posiciones de las alteraciones del bajo cifrado

Las alteraciones y los signos «más» pueden aparecer antes o después de los números, según el valor de las propiedades figuredBassAlterationDirection y figuredBassPlusDirection.

Si un signo más aparece después del número, hay glifos diseñados especialmente para algunas cifras.

```
#(set-global-staff-size 26)

\figures {
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassPlusDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #LEFT
  <5\+> <5+ 4\+> <6 4- 2\+> r
}
```



Ajustar los glifos de las alteraciones en el bajo cifrado

De forma predeterminada, en el bajo cifrado se usan glifos diseñados espcialmente para 6\\, 7\\ y 9\\. De forma similar, se usan de forma predeterminada glifos diseñados especialmente para los símbolos 2\+, 4\+ y 5\+, si aparece un signo más después del número.

Para cambiarlo, pase una lista-A a figuredBassPlusStrokedAlist y establezca el glifo en cuestión a #f o no lo especifique en absoluto.

```
#(set-global-staff-size 26)

\figures {
  \set figuredBassPlusDirection = #RIGHT
  <6\\> <7\\> <9\\> r
  <2\+> <4\+> <5\+> r

  \set figuredBassPlusStrokedAlist =
    #'((2 . "figbass.twoplus")
       ;; (4 . "figbass.fourplus")
       ;; (5 . "figbass.fiveplus")
       (6 . "figbass.sixstroked")
       ;; (7 . "figbass.sevenstroked")
       ;; (9 . "figbass.ninestroked")
      )
  <6\\> <7\\> <9\\> r
```

```
  <2\+> <4\+> <5\+> r
}
```



Véase también

Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

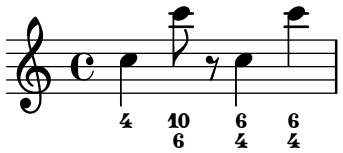
Referencia de funcionamiento interno: Sección “BassFigure” en *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” en *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” en *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” en *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” en *Referencia de Funcionamiento Interno*, Sección “FiguredBass” en *Referencia de Funcionamiento Interno*.

15.3.3 Imprimir el bajo cifrado

El bajo cifrado se puede imprimir usando el contexto `FiguredBass`, o dentro de la mayoría de los contextos de pentagrama.

Cuando se presenta dentro de un contexto de `FiguredBass`, la ubicación vertical de las cifras es independiente de las notas que están en el pentagrama.

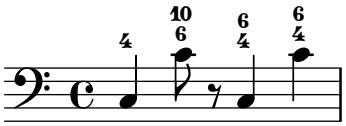
```
<<
  \relative {
    c''4 c'8 r8 c,4 c'
  }
  \new FiguredBass {
    \figuremode {
      <4>4 <10 6>8 s8
      <6 4>4 <6 4>
    }
  }
>>
```



En el ejemplo de arriba, el contexto `FiguredBass` se debe instanciar explícitamente para evitar crear un segundo pentagrama vacío.

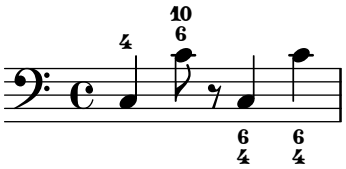
El bajo cifrado se puede añadir también a contextos de `Staff` directamente. En este caso, la posición vertical de las cifras se ajusta automáticamente.

```
<<
  \new Staff = "myStaff"
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
  %% Put notes on same Staff as figures
  \context Staff = "myStaff" {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



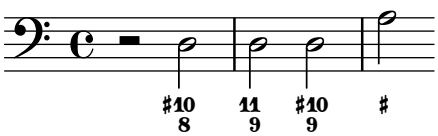
Cuando se escribe dentro de un contexto de pentagrama, el bajo cifrado se puede presentar encima o debajo del mismo.

```
<<
  \new Staff = "myStaff"
  \figuremode {
    <4>4 <10 6>8 s8
    \bassFigureStaffAlignmentDown
    <6 4>4 <6 4>
  }
  %% Put notes on same Staff as figures
  \context Staff = "myStaff" {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



La alineación horizontal de los números en una pila de cifras que tienen más de un dígito puede controlarse mediante la propiedad de contexto `figuredBassLargeNumberAlignment`.

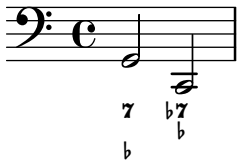
```
<<
  \new Voice {
    \clef bass
    r2 d | d d | a2
  }
  \new FiguredBass \figuremode {
    s2 <10+ 8> |
    \set figuredBassLargeNumberAlignment = #RIGHT
    <11 9>2
    \set figuredBassLargeNumberAlignment = #LEFT
    <10+ 9>2 |
    <_+>2
  }
>>
```



Se puede controlar la distancia vertical de los elementos del bajo cifrado con las subpropiedades `minimum-distance` y `padding` de `staff-staff-spacing`.

```
<<
  { \clef bass g,2 c, }
  \figures {
```

```
    \once \override BassFigureLine
                    .staff-staff-spacing.minimum-distance = 3
    <7 _-> <7- _->
  }
>>
```



Instrucciones predefinidas

`\bassFigureStaffAlignmentDown`, `\bassFigureStaffAlignmentUp`, `\bassFigureStaffAlignmentNeutral`.

Véase también

Fragmentos de código: Sección “Notación de los acordes” en *Fragmentos de código*.

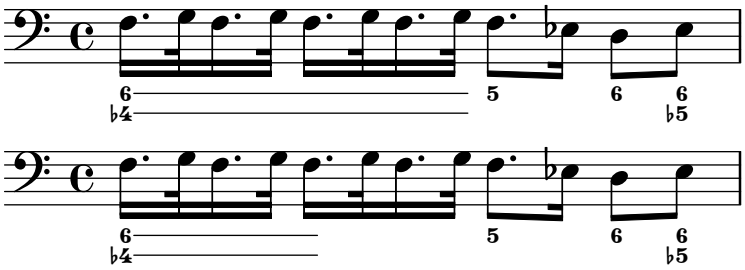
Referencia de funcionamiento interno: Sección “BassFigure” en *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” en *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” en *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” en *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” en *Referencia de Funcionamiento Interno*, Sección “FiguredBass” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Para asegurar que las líneas extensoras funcionan adecuadamente, lo más seguro es utilizar las mismas duraciones en la línea de cifras y en la línea del bajo.

```
<<
  {
    \clef bass
    \repeat unfold 4 { f16. g32 } f8. es16 d8 es
  }
  \figures {
    \bassFigureExtendersOn
    % The extenders are correct here,
    % with the same rhythm as the bass.
    \repeat unfold 4 { <6 4->16. <6 4->32 }
    <5>8. r16 <6>8 <6\! 5->
  }
>>
<<
  {
    \clef bass
    \repeat unfold 4 { f16. g32 } f8. es16 d8 es
  }
  \figures {
    \bassFigureExtendersOn
    % The extenders are incorrect here,
    % even though the timing is the same.
    <6 4->4 <6 4->4
    <5>8. r16 <6>8 <6\! 5->
  }
```

>>


6
♭4
5
6
6
♭5
6
♭4
5
6
6
♭5

16 Música contemporánea

A partir de comienzos del s.XX se ha producido una expansión masiva de los estilos y técnicas composicionales. Los nuevos desarrollos armónicos y rítmicos, una expansión del espectro de alturas y el desarrollo de un amplio abanico de nuevas técnicas instrumentales han venido acompañadas de una evolución y expansión paralelas de la notación musical. El objetivo de esta sección es proporcionar referencias e información relevante para el trabajo con estas nuevas técnicas notacionales.

16.1 Altura y armonía en la música contemporánea

Esta sección destaca los problemas de la notación de las alturas y la armonía en la música contemporánea.

16.1.1 Referencias para alturas y armonía en música contemporánea

- La notación estándar de cuartos de tono se estudia en Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10.
- Las armaduras de tonalidad no estándar se estudian en Sección 1.3.2 [Armadura de la tonalidad], página 25.
- Las prácticas contemporáneas en la presentación de las alteraciones accidentales se estudian en Sección 1.3.5 [Alteraciones accidentales automáticas], página 32.

16.1.2 Notación microtonal

16.1.3 Armonía y armaduras de tonalidad contemporáneas

16.2 Enfoques contemporáneos del ritmo

Esta sección destaca los problemas de la notación de las duraciones en la música contemporánea.

16.2.1 Referencias para enfoque contemporáneos del ritmo

- Los tipos de compás complejos o de amalgama se estudian en Sección 2.3.1 [Indicación de compás], página 79.
- La notación polimétrica básica se estudia en Sección 2.3.5 [Notación polimétrica], página 93.
- Las barras en ángulo progresivas se estudian en Sección 2.4.4 [Barras progresivas], página 118.
- Las líneas divisorias de estilo Mensurstriche (líneas que están solamente entre los pentagramas) se estudian en Sección 6.1.2 [Agrupar pentagramas], página 246.

16.2.2 Grupos de valoración especial en música contemporánea

16.2.3 Compases contemporáneos

16.2.4 Notación polimétrica extendida

16.2.5 Barrado en música contemporánea

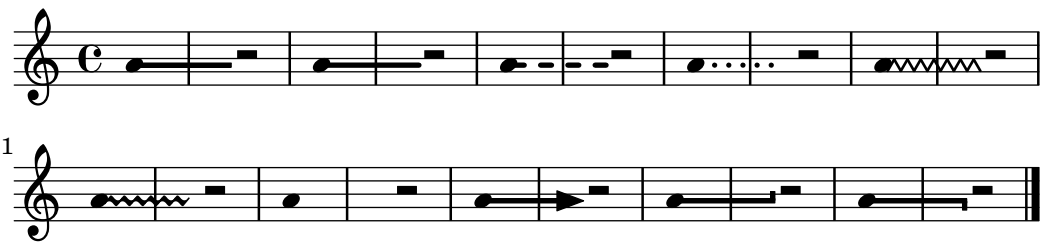
16.2.6 Líneas divisorias en música contemporánea

16.3 Notación gráfica

Los elementos rítmicos pueden continuarse con una línea de duración, que viene representada por un objeto gráfico `DurationLine`. Los estilos posibles son `'beam`, `'line`, `'dashed-line`,

'dotted-line, 'zigzag, 'trill y 'none. La línea de duración puede finalizar con un gancho (solo para el estilo beam) o una flecha.

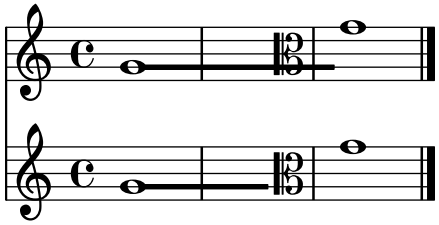
```
\layout {
  \context {
    \Voice
    \consists Duration_line_engraver
    \omit Stem
    \omit Flag
    \omit Beam
    \override NoteHead.duration-log = 2
  }
}

{
  a'1\- s2 r
  \once \override DurationLine.style = #'line
  a'1\- s2 r
  \once \override DurationLine.style = #'dashed-line
  \once \override DurationLine.dash-period = 2
  a'1\- s2 r
  \once \override DurationLine.style = #'dotted-line
  \once \override DurationLine.dash-period = 1
  \once \override DurationLine.bound-details.right.padding = 1
  a'1\- s2 r
  \once \override DurationLine.thickness = 2
  \once \override DurationLine.style = #'zigzag
  a'1\- s2 r
  \once \override DurationLine.style = #'trill
  a'1\- s2 r
  \once \override DurationLine.style = #'none
  a'1\- s2 r
  \once \override DurationLine.bound-details.right.end-style = #'arrow
  a'1\- s2 r
  \override DurationLine.bound-details.right.end-style = #'hook
  a'1\- s2 r
  \override DurationLine.details.hook-direction = #DOWN
  a'1\- s2 r
  \bar "|."
}
```



DurationLine puede evitar los elementos que están ecima de una línea a partir de BreakAlignGroup.

```
\layout {
  \context {
```

```
    \Voice
    \consists "Duration_line_engraver"
  }
}

<<
  \new Staff {
    g'1\- s \clef "alto" g'
  }
  \new Staff {
    \override DurationLine.bound-details
                          .right.end-on-break-align-group = ##t
    g'1\- s \clef "alto" g' \bar "|."
  }
>>
```



16.4 Técnicas de partitura contemporáneas

16.5 Nuevas técnicas instrumentales

16.6 Lecturas adicionales y partituras de interés

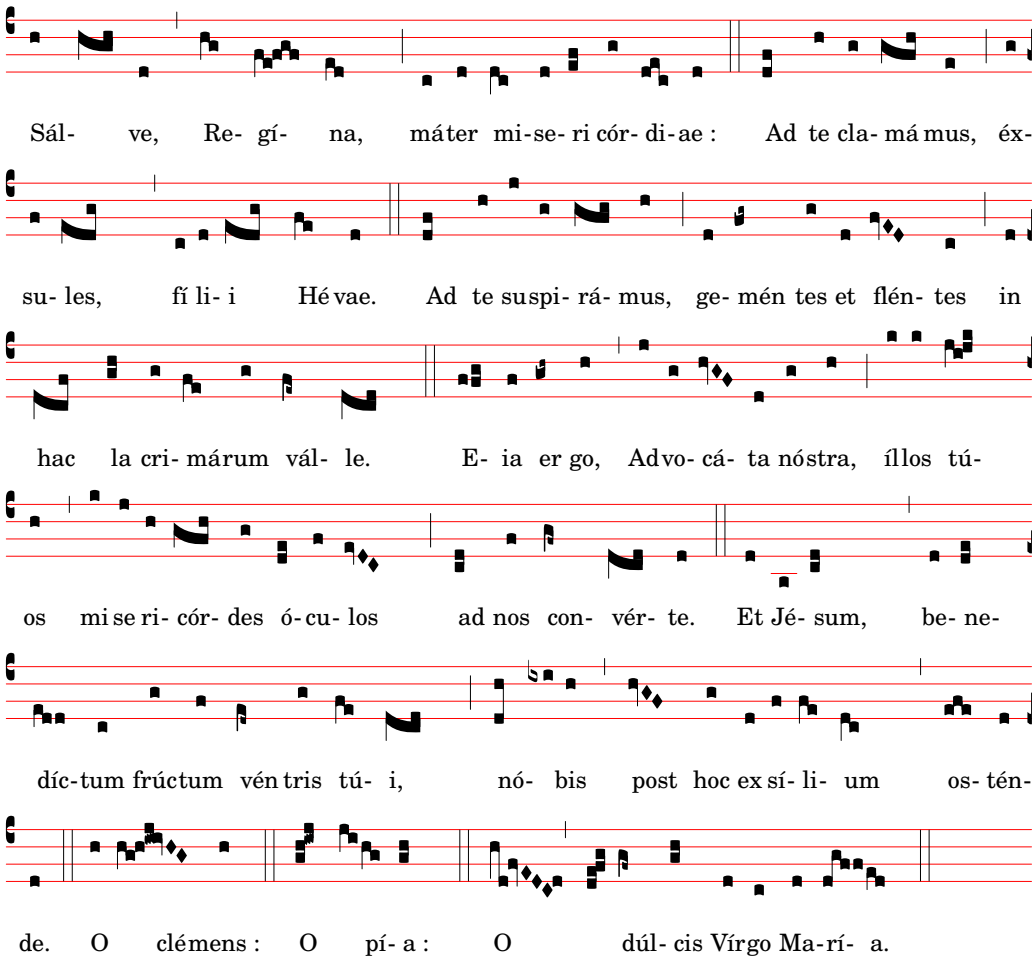
Esta sección sugiere libros, ejemplos musicales y otros recursos de utilidad en el estudio de la notación musical contemporánea.

16.6.1 Libros y artículos sobre notación musical contemporánea

- *Music Notation in the Twentieth Century: A Practical Guidebook* de Kurt Stone [W. W. Norton, 1980]
- *Music Notation: A Manual of Modern Practice* de Gardner Read [Taplinger, 1979]
- *Instrumentation and Orchestration* de Alfred Blatter [Schirmer, 2nd ed. 1997]

16.6.2 Partituras y ejemplos musicales

17 Notación antigua



Entre las formas en que está contemplada la notación antigua se encuentra la posibilidad de tipografiar notación mensural, canto gregoriano y canto kievano en notación cuadrada. Se puede acceder a estas funcionalidades mediante la modificación de propiedades de estilo de objetos gráficos como la cabeza de las notas o los silencios, o bien mediante la utilización de uno de los contextos predefinidos para estos estilos.

Muchos objetos gráficos, como la cabeza y el corchete de las figuras, las alteraciones, la indicación de compás y los silencios ofrecen una propiedad `style` (estilo), que se puede cambiar para emular varios estilos diferentes de notación antigua. Véase

- Sección 17.3.4 [Cabezas de nota de la música mensural], página 547,
- Sección 17.3.7 [Alteraciones y armaduras de la música mensural], página 549,
- Sección 17.3.6 [Silencios de la música mensural], página 548,
- Sección 17.3.2 [Claves de la música mensural], página 545,
- Sección 17.4.2 [Claves de canto gregoriano], página 553,
- Sección 17.3.5 [Corchetes de la música mensural], página 548,
- Sección 17.3.3 [Indicaciones de compás de la música mensural], página 546.

Ciertos conceptos de notación se incluyen específicamente para la notación antigua:

- Sección 17.2.3 [Custos], página 543,
- Sección 17.4.4 [Divisiones], página 554,
- Sección 17.2.2 [Ligaduras], página 542.

Véase también

Glosario musical: Sección “custos” en *Glosario Musical*, Sección “ligature” en *Glosario Musical*, Sección “mensural notation” en *Glosario Musical*.

Referencia de la notación: Sección 17.3.4 [Cabezas de nota de la música mensural], página 547, Sección 17.3.7 [Alteraciones y armaduras de la música mensural], página 549, Sección 17.3.6 [Silencios de la música mensural], página 548, Sección 17.4.2 [Claves de canto gregoriano], página 553, Sección 17.3.5 [Corchetes de la música mensural], página 548, Sección 17.3.3 [Indicaciones de compás de la música mensural], página 546, Sección 17.2.3 [Custos], página 543, Sección 17.4.4 [Divisiones], página 554, Sección 17.2.2 [Ligaduras], página 542.

17.1 Panorámica de los estilos contemplados

Están disponibles tres estilos para la composición tipográfica del canto gregoriano:

- *Editio Vaticana* es un estilo completo para canto gregoriano, que sigue la apariencia de las ediciones de Solesmes, los libros de canto oficiales del Vaticano desde 1904. LilyPond contempla todos los signos de notación que se utilizan en este estilo, entre ellos las ligaduras, *custodias* y símbolos especiales como el quilisma y el oriscus.
- El estilo *Editio Medicaea* ofrece ciertas funcionalidades que se usan en las ediciones Medicaea (o de Ratisbona) que se usaban con anterioridad a las ediciones de Solesmes. Las diferencias más significativas con respecto al estilo *Vaticana* son las claves, que tienen trazos descendentes, y la cabeza de las figuras, que son cuadradas y regulares.
- El estilo *Hufnagel* (en “uña de herradura”) o *gótico* reproduce el estilo de escritura en los manuscritos de canto medievales de Alemania y Centro Europa. Recibe el nombre de la forma de la nota básica (la *virga*), que parece una pequeña uña.

Tres estilos emulan la apariencia de los manuscritos y ediciones impresas de música mensural tardomedievales y renacentistas:

- El estilo *Mensural* se asemeja bastante al estilo de escritura utilizado en los manuscritos tardomedievales y renacentistas tempranos, con la cabeza de sus figuras en forma de rombo pequeñas y estrechas, y los silencios que imitan un estilo dibujado a mano.
- El estilo *Neomensural* es una versión modernizada y estilizada del anterior: la cabeza de las figuras es más ancha y los silencios están formados por trazos rectilíneos. Este estilo está particularmente indicado, p.ej., para los *incipits* de transcripciones de fragmentos de música mensural.
- El estilo *Petrucci* recibe su nombre de Ottaviano Petrucci (1466-1539), el primer impresor en utilizar tipos móviles para la música (en su *Harmonice musices odhecaton*, 1501). Este estilo utiliza para las figuras una cabeza más grande que los otros estilos mensurales.

Baroque y *classical* no son estilos completos, sino que difieren del estilo predeterminado solamente en ciertos detalles: algunas cabezas de nota (Baroque) y el silencio de negra (classical).

Solo el estilo mensural tiene alternativas para todos los aspectos de la notación. Así, no existen silencios ni corchetes en los estilos gregorianos, pues estos símbolos no se usan en la notación del canto llano, y el estilo Petrucci no posee corchetes ni alteraciones por sí mismo.

Cada elemento de la notación se puede cambiar independientemente de los otros, y así podemos usar corchetes mensurales, cabezas de Petrucci, silencios de classical y claves de Vaticana en la misma pieza, si queremos.

Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “flag” en *Glosario Musical*.

17.2 Notación antigua: funcionalidades comunes

17.2.1 Contextos predefinidos

Para el canto gregoriano y la notación mensural, están a nuestra disposición contextos predefinidos de voz y de pauta, que establecen todos los símbolos de notación a unos valores adecuados para estos estilos. Si nos satisfacen estos valores predeterminados, podemos a continuación introducir directamente las notas sin la necesidad de preocuparnos de los detalles sobre cómo personalizar un contexto. Véase uno de los contextos predefinidos `VaticanaScore`, `VaticanaVoice`, `VaticanaStaff`, `MensuralVoice`, `MensuralStaff`, `PetrucciStaff`, `PetrucciVoice`, `KievanVoice` y `KievanStaff`.

Véase también

Glosario musical: Sección "mensural notation" en *Glosario Musical*.

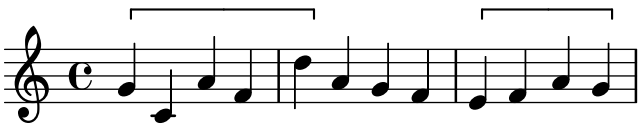
Referencia de la notación: Sección 17.4.1 [Contextos del canto gregoriano], página 552, Sección 17.3.1 [Contextos de la música mensural], página 544, Sección 17.5.1 [Contextos del canto kievano], página 562.

17.2.2 Ligaduras

Una ligadura es un símbolo gráfico que representa al menos dos notas distintas. Las ligaduras aparecieron originalmente en los manuscritos de la notación del canto gregoriano para denotar secuencias de notas ascendentes o descendentes en la misma sílaba. Se usan también en notación mensural.

Las ligaduras se escriben *encerrándolas* dentro de un par `\[` y `\]`. Ciertos estilos de ligadura podrían necesitar una sintaxis de entrada adicional específica para este tipo de ligadura en particular. De forma predeterminada, el grabador `LigatureBracket` se limita a colocar un corchete recto sobre la ligadura:

```
\relative {
  \[ g' c, a' f d' \]
  a g f
  \[ e f a g \]
}
```



Están disponibles otros tres estilos de ligaduras: el 'Vaticana' para canto gregoriano, el 'Mensural' para música mensural (solo están contempladas las ligaduras mensurales blancas para la música mensural, y con ciertas limitaciones), y 'Kievan' para los melismas del canto kievano. Para usar cualquiera de estos estilos, se debe sustituir el grabador predeterminado `Ligature_bracket_engraver` por uno de los grabadores especializados en ligaduras del contexto `Voice`. Para más información, véase Sección 17.3.9 [Ligaduras mensurales blancas], página 550, Sección 17.4.7 [Ligaduras de neumas cuadrados gregorianos], página 556, and Sección 17.5.6 [Melismas del canto kievano], página 564.

Véase también

Glosario musical: Sección "ligature" en *Glosario Musical*.

Referencia de la notación: Sección 17.3.9 [Ligaduras mensurales blancas], página 550, Sección 17.4.7 [Ligaduras de neumas cuadrados gregorianos], página 556.

Advertencias y problemas conocidos

El espacio que las ligaduras necesitan no está implementado actualmente, y como resultado, puede producirse una separación excesiva entre ellas. Los saltos de línea podrían también ser poco satisfactorios.

La estra podría no alinearse con la música de la forma esperada cuando se utilizan ligaduras.

No se deben imprimir alteraciones accidentales dentro de una ligadura, sino acumularse e imprimirse delante de ella.

La sintaxis aún utiliza el estilo obsoleto "infijo" `\[ expresión musical \]`. Por motivos de consistencia, se cambiará tarde o temprano al estilo "postfijo" `nota\[ ... nota\]`.

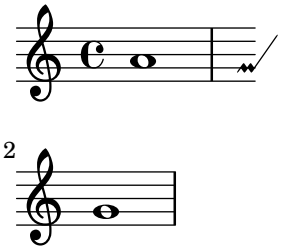
17.2.3 Custos

Un *custos* (en plural: *custodias*, que en latín significa "guarda") es un símbolo que aparece al final de una pauta. Anticipa la altura de la primera o primeras notas de la línea siguiente, ayudando al intérprete con los saltos de línea durante la ejecución.

Las custodias se utilizaron mucho en la notación musical hasta el siglo XVII. En nuestros días, sobreviven solo en algunas formas particulares de notación musical como las ediciones contemporáneas de canto gregoriano como la *Editio Vaticana*. Hay distintos glifos para el custos que se utilizan en las diferentes variaciones de estilo notacional.

Para tipografiar las custodias, simplemente coloque un grabador `Custos_engraver` dentro del contexto `Staff` (el contexto del pentagrama) al declarar el bloque de `\layout` (el bloque de disposición de la página), y modifique el estilo del custos con una instrucción `\override` si lo desea, como se muestra en el ejemplo siguiente:

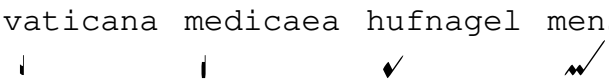
```
\score {
  \relative {
    a'1
    \break
    g
  }
  \layout {
    \context {
      \Staff
      \consists Custos_engraver
      \override Custos.style = #'mensural
    }
  }
}
```



El glifo del custos se selecciona mediante la propiedad `style`. Los estilos contemplados son `vaticana`, `medicaea`, `hufnagel` y `mensural`.

```
\new Lyrics \lyricmode {
  \markup { \column {
    \typewriter "vaticana "
```

```
    \line { " " \musicglyph "custodes.vaticana.u0" }
  } }
  \markup { \column {
    \typewriter "medicaea "
    \line { " " \musicglyph "custodes.medicaea.u0" }
  }}
  \markup { \column {
    \typewriter "hufnagel "
    \line { " " \musicglyph "custodes.hufnagel.u0" }
  }}
  \markup { \column {
    \typewriter "mensural "
    \line { " " \musicglyph "custodes.mensural.u0" }
  }}
}
```



Véase también

Glosario musical: Sección “custos” en *Glosario Musical*.

Fragmentos de código: Sección “Notación de la música antigua” en *Fragmentos de código*.

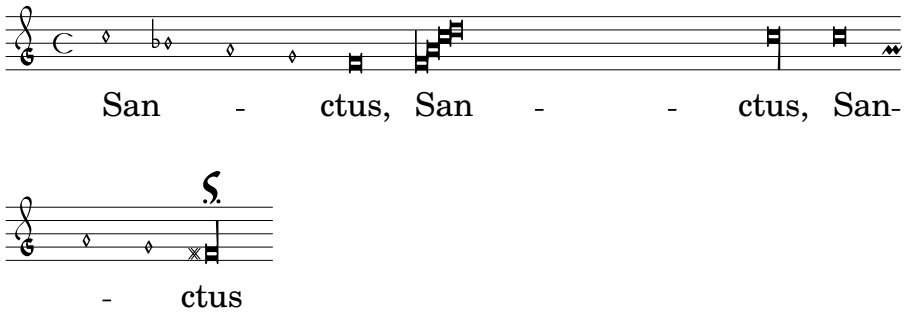
Referencia de funcionamiento interno: Sección “Custos” en *Referencia de Funcionamiento Interno*.

17.3 Tipografiar música mensural

17.3.1 Contextos de la música mensural

Los contextos predefinidos de voz `MensuralVoice` y de pauta `MensuralStaff` se pueden usar para grabar una pieza en estilo mensural. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de manera que pueda comenzar inmediatamente a escribir el canto, como lo demuestra el siguiente fragmento:

```
\score {
  <<
    \new MensuralVoice = "discantus" \relative {
      \hide Score.BarNumber {
        c''1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c\breve d\melismaEnd \]
        c\longa
        c\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



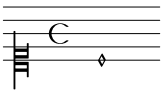
Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*.

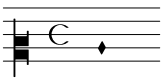
17.3.2 Claves de la música mensural

Están contempladas las claves mensurales usando la instrucción `\clef`. Algunas de las claves utilizan el mismo glifo, pero se diferencian solamente respecto a la línea en la que se imprimen. En tales casos, se utiliza un sufijo numérico en el nombre para denominar dichas claves, contando las líneas desde la inferior hasta la superior.

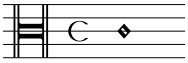
```
\new MensuralStaff {
  \clef "mensural-c1" c'1
}
```



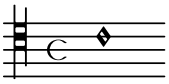
```
\new MensuralStaff {
  \override NoteHead.style = #'blackmensural
  \clef "blackmensural-c2" c'1
}
```



```
\new MensuralStaff {
  \override NoteHead.style = #'neomensural
  \clef "neomensural-c3" c'1
}
```



```
\new PetrucciStaff {
  \clef "petrucci-c4" c'1
}
```



Se puede forzar manualmente la composición tipográfica de un glifo de clave sobre cualquier línea seleccionada arbitrariamente; véase Sección 1.3.1 [Clave], página 20. Para ver el abanico completo de las claves posibles, véase Sección B.11 [Estilos de clave], página 928.

Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “clef” en *Glosario Musical*.

Referencia de la notación: Sección 17.4.2 [Claves de canto gregoriano], página 553, Sección 1.3.1 [Clave], página 20.

Archivos instalados: `scm/parser-clef.scm`.

Fragmentos de código: Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Clef_engraver” en *Referencia de Funcionamiento Interno*, Sección “Clef” en *Referencia de Funcionamiento Interno*, Sección “ClefModifier” en *Referencia de Funcionamiento Interno*, Sección “clef-interface” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

La clave de Sol Mensural está asignada a la clave de Sol de Petrucci.

17.3.3 Indicaciones de compás de la música mensural

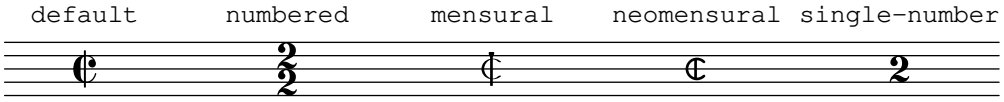
Los símbolos de mensuración (que son algo similar, pero no exactamente igual que las indicaciones de compás) están contemplados de forma limitada. Los glifos están configurados de forma fija para determinadas fracciones de tiempo. En otras palabras, para obtener un símbolo de mensuración determinado con la instrucción `\time` *n*`/`*m*, se tienen que escoger *n* y *m* según la siguiente tabla

`\time 4/4` `\time 2/2` `\time 6/4` `\time 6/8`

`\time 3/2` `\time 3/4` `\time 9/4` `\time 9/8`

`\time 4/8` `\time 2/4`

Utilice la propiedad `style` del elemento gráfico `TimeSignature` para seleccionar las indicaciones de compás de la música antigua. Los estilos contemplados son `neomensural` y `mensural`. La tabla anterior utiliza el estilo `neomensural`. Los ejemplos siguientes muestran las diferencias que existen entre los estilos:



Sección 2.3.1 [Indicación de compás], página 79, ofrece una introducción general al uso de las indicaciones de compás.

Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*.

Referencia de la notación: Sección 2.3.1 [Indicación de compás], página 79.

Advertencias y problemas conocidos

Las relaciones de duración de las notas no pueden cambiar con la indicación de compás, porque no son constantes. Por ejemplo, la razón de 1 breve = 3 semibreves (*tempus perfectum*) se puede hacer a mano mediante el establecimiento de

```
breveTP = #(ly:make-duration -1 0 3/2)
```

```
...
{ c\breveTP f1 }
```

Esto fija un valor para `breveTP` de 3/2 multiplicado por 2 = 3 veces una redonda.

Los símbolos `mensural68alt` y `neomensural68alt` (alternativos al símbolo de 6/8) no se puede seleccionar directamente con `\time`. En su lugar, utilice `\markup {\musicglyph "timesig.mensural68alt" }`.

17.3.4 Cabezas de nota de la música mensural

Para la notación antigua, se puede elegir un estilo de cabezas de notas distinto al `default` (estilo predeterminado). Esto se consigue estableciendo la propiedad `style` del objeto `NoteHead` a los valores `baroque`, `neomensural`, `mensural`, `petrucci`, `blackpetrucci` o `semipetrucci`.

El estilo `baroque` se diferencia del estilo predeterminado `default` en que:

- Proporciona una cabeza de nota para la `maxima`, y
- Usa una forma cuadrada para las cabezas de `\breve`.

Los estilos `neomensural`, `mensural` y `petrucci` se diferencian del estilo `baroque` en que:

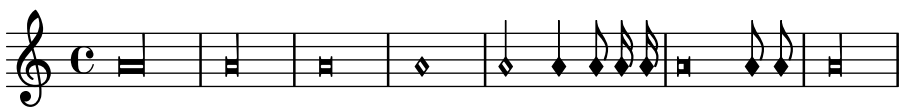
- Usan cabezas romboidales para las semibreves y todas las notas más cortas, y
- Centran las plicas sobre las cabezas.

El estilo `blackpetrucci` produce notas con una cabeza que puede usarse en notación mensural negra o en secciones de coloratio dentro de la notación mensural blanca. A causa de que el estilo de la cabeza de las notas no influye en el número de corchetes, en este estilo la notación de la semimínima debe hacer como `a8*2`, no como `a4`, pues de lo contrario tendrá el aspecto de una mínima. El multiplicador puede ser distinto si se usa la coloratio, es decir, para la notación de tresillos.

Utilice el estilo `semipetrucci` para dibujar notas con cabeza semi-rellena (breves, longas y máximas).

El ejemplo siguiente muestra el estilo `petrucci`:

```
\compressEmptyMeasures
\autoBeamOff
\override NoteHead.style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
\override NoteHead.style = #'semipetrucci
a'\breve*5/6
\override NoteHead.style = #'blackpetrucci
a'8*4/3 a'
\override NoteHead.style = #'petrucci
a'\longa
```



Sección B.9 [Estilos de cabezas de nota], página 927, ofrece una visión de conjunto sobre la totalidad de los estilos de cabeza disponibles.

Véase también

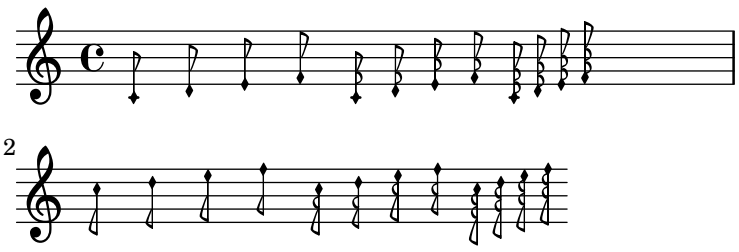
Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “note head” en *Glosario Musical*.

Referencia de la notación: Sección B.9 [Estilos de cabezas de nota], página 927.

17.3.5 Corchetes de la música mensural

Utilice la propiedad `style` del elemento gráfico `Flag` para seleccionar los corchetes de estilo antiguo. Aparte del estilo de corchete por defecto `default`, solo está contemplado el estilo `mensural`

```
\relative c' {
  \override Flag.style = #'mensural
  \override Stem.thickness = 1.0
  \override NoteHead.style = #'mensural
  \autoBeamOff
  c8 d e f c16 d e f c32 d e f s8
  c'8 d e f c16 d e f c32 d e f
}
```



Observe que la bandera más interna de cada uno de los corchetes mensurales se alinea verticalmente con una línea de la pauta.

No existe un estilo particular de corchete para la notación neo-mensural ni Petrucci. No existen corchetes en la notación del canto gregoriano.

Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “flag” en *Glosario Musical*.

Advertencias y problemas conocidos

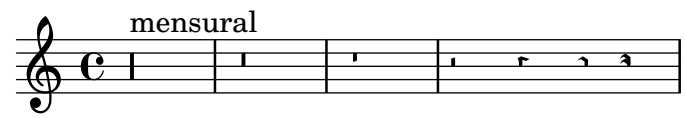
La alineación vertical de cada uno de los corchetes con una línea de la pauta da por supuesto que las plicas siempre terminan exactamente encima o bien exactamente en el medio de dos líneas de la pauta. Esto puede no ser siempre cierto cuando se utilizan las posibilidades de disposición avanzadas de la notación clásica (que, de todas formas, por lo común se encuentra fuera del ámbito de la notación mensural).

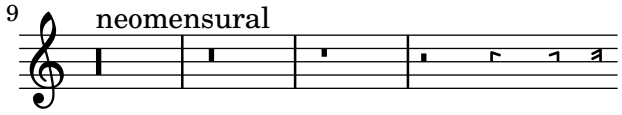
17.3.6 Silencios de la música mensural

Utilice la propiedad `style` del elemento gráfico `Rest` para seleccionar silencios antiguos. Los estilos de música antigua contemplados son `classical`, `neomensural` y `mensural`.

El ejemplo siguiente muestra estos estilos:

```
\compressEmptyMeasures
\override Rest.style = #'mensural
r\longa^"mensural" r\breve r1 r2 r4 r8 r16 s \break
\override Rest.style = #'neomensural
r\longa^"neomensural" r\breve r1 r2 r4 r8 r16
```





No existen silencios de fusa y semifusa específicos para los estilos mensural ni neo-mensural. Se usan los silencios del estilo predeterminado.

Véase también

Glosario musical: Sección "mensural notation" en *Glosario Musical*.

Referencia de la notación: Sección 2.2.1 [Silencios], página 68.

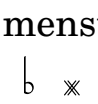
Fragmentos de código: Sección "Notación de la música antigua" en *Fragmentos de código*.

Advertencias y problemas conocidos

El glifo para el silencio de máxima en el estilo mensural es en realidad un silencio de longa perfecta; use dos (o tres) silencios de longa para imprimir un silencio de máxima. Los sliencions de longa no se agrupan automáticamente, por lo que debe hacerse manualmente utilizando silencios con altura determinada.

17.3.7 Alteraciones y armaduras de la música mensural

El estilo `mensural` ofrece unos símbolos de sostenido y bemol distintos de los del estilo predeterminado. La notación mensural rara vez utilizaba el becuadro: en su lugar, se usaba el sostenido o bemol correspondiente. Por ejemplo, un Si natural en la tonalidad de Fa mayor se indicaría con un sostenido. Sin embargo, si se solicita explícitamente, el símbolo de becuadro se extrae del estilo `vaticana`.



La manera de usar este estilo se explica en Sección 1.3.6 [Glifos de alteración alternativos], página 40. Se encuentra por omisión dentro del contexto `MensuralStaff`.

Véase también

Glosario musical: Sección "mensural notation" en *Glosario Musical*, Sección "Pitch names" en *Glosario Musical*, Sección "accidental" en *Glosario Musical*, Sección "key signature" en *Glosario Musical*.

Referencia de la notación: Capítulo 1 [Alturas], página 3, Sección 1.1.3 [Alteraciones accidentales], página 8, Sección 1.3.5 [Alteraciones accidentales automáticas], página 32, Sección 1.3.6 [Glifos de alteración alternativos], página 40, Sección B.10 [Conjuntos de glifos de alteraciones accidentales], página 927, Sección 1.3.2 [Armadura de la tonalidad], página 25.

Referencia de funcionamiento interno: Sección "KeySignature" en *Referencia de Funcionamiento Interno*.

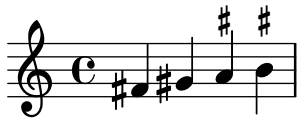
17.3.8 Alteraciones de anotación (*musica ficta*)

En la música europea de hasta aprox. 1600, se esperaba de los cantantes que alterasen cromáticamente ciertas notas según su propia iniciativa y de acuerdo a ciertas reglas. Esto recibe el nombre de *musica ficta*. En las transcripciones modernas, estas alteraciones accidentales se imprimen normalmente encima de la nota.

Están contempladas estas alteraciones sugeridas, y se pueden activar estableciendo `suggestAccidentals` al valor `#t` (verdadero).

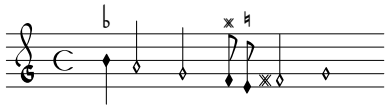
```
\relative {
```

```
  fis' gis
  \set suggestAccidentals = ##t
  ais bis
}
```



Esto hará que se traten *todas* las alteraciones siguientes como *musica ficta* hasta que se desactive con `\set suggestAccidentals = ##f`. Una forma más práctica es usar `\once \set suggestAccidentals = ##t`, que puede incluso definirse como una forma breve muy conveniente:

```
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative
  \new MensuralVoice  {
    \once \set suggestAccidentals = ##t
    bes'4 a2 g2 \ficta fis8 \ficta e! fis2 g1
  }
}
```



Véase también

Referencia de funcionamiento interno: Sección “Accidental_engraver” en *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” en *Referencia de Funcionamiento Interno*.

17.3.9 Ligaduras mensurales blancas

Las ligaduras mensurales blancas están contempladas de forma limitada.

Para grabar ligaduras mensurales blancas, reemplace el grabador `Ligature_bracket_engraver` por el grabador `Mensural_ligature_engraver` en el bloque layout del contexto `Voice`:

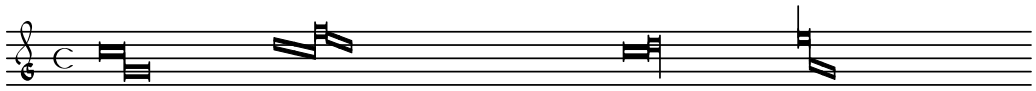
```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
}
```

En el ejemplo siguiente, usamos un contexto `PetrucciStaff`, que lleva a cabo esta sustitución, entre otras configuraciones, asemejándose a la tipografía mensural del *Harmonices Musices Odhecaton* de Ottaviano Petrucci (Venecia, 1501). El contexto de la voz que lo acompaña se llama `PetrucciVoice`.

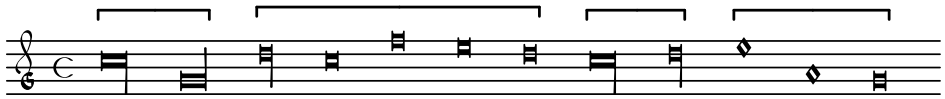
No existe ningún lenguaje de entrada adicional para describir la forma de una ligadura mensural blanca. Más bien la forma se determina exclusivamente a partir de las alturas y las duraciones de las notas comprendidas. Si bien este enfoque podría hacer que un usuario con poca experiencia tardase un tiempo en acostumbrarse a él, tiene la gran ventaja de que toda la información musical de la ligadura se conoce internamente. Esto no se requiere solamente para una salida MIDI correcta, sino que también permite la transcripción automática de las ligaduras.

En algunos lugares, dos notas consecutivas se pueden representar bien como dos cuadrados, bien como un paralelogramo oblicuo (forma de *flexa*). En tales casos la forma por omisión es dos cuadrados, pero puede requerirse una flexa fijando la propiedad `ligature-flexa` de la *segunda* cabeza. El ancho de una flexa se puede fijar mediante la propiedad `flexa-width` de la cabeza. Por ejemplo:

```
\new PetrucciStaff \relative {
  \[ c''\maxima g \]
  \[ d'\longa
     \tweak ligature-flexa ##t
     \tweak flexa-width #3.2 c\breve f e d \]
  \[ c\maxima d\longa \]
  \[ e1 a, g\breve \]
}
```



Si no se sustituye el grabador `Ligature_bracket_engraver` por el `Mensural_ligature_engraver`, la misma música presenta la siguiente apariencia:

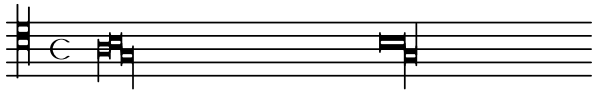


También hay casos en los que no se necesita una plica para registrar la duración de la nota de forma no ambigua, pero tampoco está prohibido:

- una breve inicial (con una altura más baja que la nota siguiente) puede tener, o no, una plica a la izquierda y hacia abajo;
- una maxima puede o no tener una plica a la derecha y hacia abajo;
- una longa final (con una altura más baja que la nota anterior) puede o no tener una plica a la derecha.

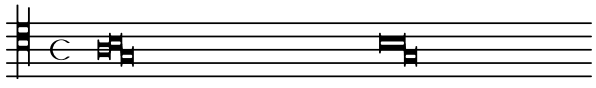
He aquí un ejemplo que muestra este trucaje.

```
\new PetrucciStaff \relative {
  \clef "petrucci-c4"
  \[ \tweak left-down-stem ##t a\breve b
     \tweak right-down-stem ##t g\longa \]
  \[ \tweak right-down-stem ##t b\maxima
     \tweak right-up-stem ##t g\longa \]
}
```



Without tweaking the same ligatures look as follows.

```
\new PetrucciStaff \relative {
  \clef "petrucci-c4"
  \[ a\breve b g\longa \]
  \[ b\maxima g\longa \]
}
```



Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: Sección 17.4.7 [Ligaduras de neumas cuadrados gregorianos], página 556, Sección 17.2.2 [Ligaduras], página 542.

Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras puede ser algo pobre. Las alteraciones accidentales pueden colisionar con las notas previas.

17.4 Tipografiado del canto gregoriano

Al componer tipográficamente una pieza en notación de canto gregoriano, el grabador `Vaticana_ligature_engraver` selecciona automáticamente la cabeza adecuada de las figuras, de manera que no es necesario establecer explícitamente el estilo de las cabezas. A pesar de ello se puede establecer el estilo de las cabezas, p.ej.: a `vaticana.punctum` para producir neumas de punctum. De forma similar, el grabador `Mensural_ligature_engraver` construye ligaduras mensurales automáticamente. Consulte Sección 17.2.2 [Ligaduras], página 542, para ver cómo funcionan los grabadores de ligaduras.

Véase también

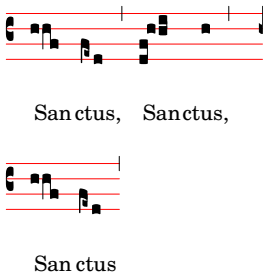
Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: Sección 17.3.9 [Ligaduras mensurales blancas], página 550, Sección 17.2.2 [Ligaduras], página 542.

17.4.1 Contextos del canto gregoriano

Los contextos predefinidos `VaticanaScore`, `VaticanaVoice`, `VaticanaStaff` y `VaticanaLyrics` se pueden utilizar para grabar una pieza de canto gregoriano en el estilo de la Editio Vaticana. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos. Puede inmediatamente comenzar a escribir el canto en sí, como lo demuestra el siguiente fragmento.

```
\new VaticanaScore {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
    \new VaticanaLyrics \lyricsto "cantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}

\layout {
  indent = 0
  ragged-last = ##t
}
```



17.4.2 Claves de canto gregoriano

La tabla siguiente muestra todas las claves gregorianas contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian solamente en la línea en que se imprimen. En estos casos, para enumerar estas claves se usa un número a continuación del nombre de la clave, contando desde la línea inferior. Aún se puede forzar manualmente la composición tipográfica de un glifo de clave sobre una línea arbitraria, como se describe en Sección 1.3.1 [Clave], página 20. La nota que se imprime a la derecha junto a cada una de las claves en la columna de los ejemplos, denota el Do central (`c'`) respecto a dicha clave.

Descripción	**Claves posibles**	**Ejemplo**
Clave de Do, estilo Editio Vaticana	`vaticana-do1`, `vaticana-do2`, `vaticana-do3`	
Clave de Fa, estilo Editio Vaticana	`vaticana-fa1`, `vaticana-fa2`	
Clave de Do, estilo Editio Medicaea	`medicaea-do1`, `medicaea-do2`, `medicaea-do3`	
Clave de Fa, estilo Editio Medicaea	`medicaea-fa1`, `medicaea-fa2`	
Clave de Do, estilo hufnagel	`hufnagel-do1`, `hufnagel-do2`, `hufnagel-do3`	
Clave de Fa, estilo hufnagel	`hufnagel-fa1`, `hufnagel-fa2`	
Clave combinada de Do y Fa, estilo hufnagel	`hufnagel-do-fa`	

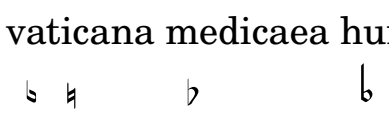
Véase también

Glosario musical: Sección “clef” en *Glosario Musical*.

Referencia de la notación: Sección 1.3.1 [Clave], página 20.

17.4.3 Alteraciones y armaduras de canto gregoriano

Están disponibles las alteraciones de los tres estilos gregorianos:



Como se ve en el ejemplo, cada estilo no contempla todas las alteraciones. Al intentar acceder a una alteración que no está contemplada, LilyPond cambia a un estilo diferente.

La forma de alternar entre los estilos se explica en Sección 1.3.6 [Glifos de alteración alternativos], página 40.

Véase también

Glosario musical: Sección “accidental” en *Glosario Musical*, Sección “key signature” en *Glosario Musical*.

Referencia de la notación: Capítulo 1 [Alturas], página 3, Sección 1.1.3 [Alteraciones accidentales], página 8, Sección 1.3.5 [Alteraciones accidentales automáticas], página 32, Sección 1.3.6 [Glifos de alteración alternativos], página 40, Sección 1.3.2 [Armadura de la tonalidad], página 25.

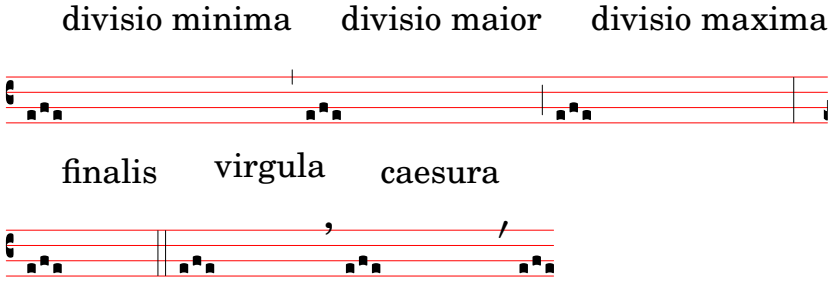
Referencia de funcionamiento interno: Sección “KeySignature” en *Referencia de Funcionamiento Interno*.

17.4.4 Divisiones

En la notación del canto gregoriano no existen silencios, sino Sección 17.4.4 [Divisiones], página 554.

Una *divisio* (en plural: *divisiones*, que en latín significa ‘división’) es un símbolo staff-context que indica la estructura de la música gregoriana en frases y secciones. El significado musical de *divisio minima*, *divisio maior* y *divisio maxima* se podrían caracterizar como una pausa corta, media y larga (respectivamente). El signo de *finalis* no solo marca el final de un canto, sino que se usa también con frecuencia dentro de un único canto antifonal o responsorial para marcar el final de cada una de las secciones.

Algunas ediciones usan *virgula* o *caesura* en vez de divisio minima. Por ello, `gregorian.ly` define también las instrucciones `\virgula` y `\caesura`.



Instrucciones predefinidas

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

Véase también

Glosario musical: Sección “caesura” en *Glosario Musical*, Sección “divisio” en *Glosario Musical*.

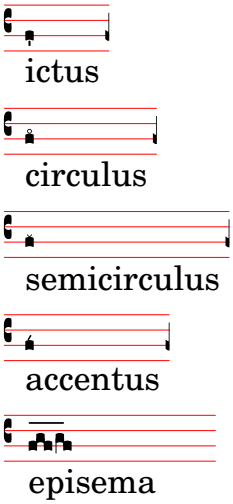
Referencia de la notación: Sección 3.2.3 [Marcas de respiración], página 177.

17.4.5 Articulaciones del canto gregoriano

Además de los signos de articulación estándar que se hallan descritos en la sección Sección 3.1.1 [Articulaciones y ornamentos], página 156, se proveen signos de articulación diseñados específicamente para utilizarlos con la notación en el estilo de la *Editio Vaticana*.

```
\new VaticanaScore {
  \new VaticanaVoice {
    \override Script.padding = -0.1
    a\ictus_"ictus " \break
    a\circulus_"circulus " \break
    a\semicirculus_"semicirculus " \break
```

```
    a\accentus_"accentus " \break
    \[ a_"episema" \episemInitium \pes b
       \flexa a b \episemFinis \flexa a \]
  }
}

\layout {
  indent = 0
  ragged-last = ##t
}
```



Véase también

Referencia de la notación: Sección 3.1.1 [Articulaciones y ornamentos], página 156.

Fragmentos de código: Sección “Notación de la música antigua” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Episema” en *Referencia de Funcionamiento Interno*, Sección “EpisemaEvent” en *Referencia de Funcionamiento Interno*, Sección “Episema_engraver” en *Referencia de Funcionamiento Interno*, Sección “Script” en *Referencia de Funcionamiento Interno*, Sección “ScriptEvent” en *Referencia de Funcionamiento Interno*, Sección “Script_engraver” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Ciertas articulaciones se colocan demasiado cerca verticalmente de las cabezas de nota correspondientes.

17.4.6 Puntos de aumentación (*morae*)

Los puntos de aumentación, llamados también *morae*, se escriben con la función musical `\augmentum`. Observe que `\augmentum` está programado como una función musical unaria y no como un prefijo de cabeza. Se aplica solamente a la expresión musical que sigue inmediatamente. Esto es, `\augmentum \virga c` no tendrá ningún efecto visible. En su lugar, escriba `\virga \augmentum c` o `\augmentum {\virga c}`. Tenga en cuenta también que puede escribir `\augmentum {a g}` como abreviatura de `\augmentum a \augmentum g`.

```
\new VaticanaScore {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
```

```
}
```



Véase también

Referencia de la notación: Sección 3.2.3 [Marcas de respiración], página 177.

Referencia de funcionamiento interno: Sección “Divisio” en *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Notación de la música antigua” en *Fragmentos de código*.

17.4.7 Ligaduras de neumas cuadrados gregorianos

La notación de neumas cuadrados gregorianos está contemplada de forma limitada (siguiendo el estilo de la Editio Vaticana). El núcleo principal del conjunto de las ligaduras ya se pueden tipografiar, pero aún faltan aspectos esenciales para una composición tipográfica seria, como (entre otros) la alineación horizontal de varias ligaduras, la alineación de la letra y un correcto manejo de las alteraciones accidentales.

Las cabezas de las notas se pueden *modificar* y/o *unir*.

- La forma de la cabeza se puede modificar *precediendo* el nombre de la nota con una cualquiera de las instrucciones siguientes: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.
- Las ligaduras hablando en propiedad, (es decir, notas unidas), se producen escribiendo una de las instrucciones de unión `\pes` o `\flexa`, para el movimiento ascendente y descendente, respectivamente, *intercaladas* entre las notas que se han de unir.

Una nota sin modificaciones produce un *punctum*. Todos los demás neumas, incluso los neumas de una nota con forma distinta como la *virga*, se consideran en principio como ligaduras y por ello se deben escribir dentro de `\[...\]`.

Neumas de una nota

- El *punctum* es la forma básica de nota (en el estilo *Vaticana*: un cuadrado con una ligera curvatura a modo de excelencia tipográfica). Además del *punctum* normal, están el *punctum inclinatum*, oblicuo, producido con el prefijo `\inclinatum`. El *punctum* normal se puede modificar con `\cavum`, que produce una nota vacía, y `\linea`, que traza líneas verticales a ambos lados de la nota.
- La *virga* tiene una plica descendente en el lado derecho. Se produce mediante el modificador `\virga`.

Ligaduras

A diferencia de casi todos los otros sistemas de notación de neumas, el aspecto tipográfico de las ligaduras no viene dictado directamente por las instrucciones de la entrada, sino que sigue unas convenciones que dependen del significado musical. Por ejemplo, una ligadura de tres notas con la forma musical bajo-alto-bajo, como `\[ a \pes b \flexa g \]`, produce un Torculus que consiste en tres cabezas de Punctum, mientras que la forma alto-bajo-alto, como `\[ a \flexa g \pes b \]`, produce un Porrectus con una forma de flexa curvada y una sola cabeza de Punctum. No existe ninguna instrucción para tipografiar explícitamente la forma de flexa curvada; la decisión de cuándo tipografiar una forma de flexa curvada está basada en la entrada musical. La idea de este enfoque es separar los aspectos musicales de la entrada, del estilo de notación de la salida. De esta forma, la misma entrada se puede reutilizar para tipografiar la misma música en un estilo diferente de notación de canto gregoriano.

Neumas licuescentes

Otra categoría fundamental de notas en el canto gregoriano es la de los llamados neumas licuescentes. Se utilizan bajo ciertas circunstancias al final de una sílaba que acaba en una letra 'licuescente', es decir, consonantes sonoras que pueden tener una altura tonal, (nasales, l, r, v, j, y sus diptongos equivalentes). Así, los neumas licuescentes nunca se utilizan aisladamente (aunque se pueden producir algunos de ellos), y siempre están al final de una ligadura.

Los neumas licuescentes se representan gráficamente de dos formas distintas más o menos intercambiables: con una nota pequeña o 'girando' la nota principal hacia arriba o hacia abajo. La primera forma se produce haciendo un `pes` o una `flexa` normales y modificando la forma de la segunda nota: `\[ a \pes \deminutum b \]` , mientras que la segunda se hace modificando la forma de un neuma de una nota con `\auctum` y uno de los generadores de dirección `\descendens` o `\ascendens`, p.ej. `\[ \auctum \descendens a \]` .

Símbolos especiales

Hay una tercera categoría de símbolos que se hace a partir de un pequeño número de símbolos que tienen un significado especial (que, por cierto, casi siempre se conocen solo vagamente): el *quilisma*, el *oriscus* y el *strophicus*. Todos ellos se producen anteponiendo al nombre de la nota el modificador correspondiente, `\quilisma`, `\oriscus` o `\stropha`.

Dentro de los delimitadores de ligadura `\[` y `\]`, se pueden acumular prácticamente cualquier cantidad de cabezas de nota para formar una sola ligadura, y los prefijos de cabeza como `\pes`, `\flexa`, `\virga`, `\inclinatum`, etc. se pueden mezclar libremente. El uso del conjunto de reglas que subyace a la construcción de ligaduras en la tabla anterior está consecuentemente extrapolada. De esta manera se pueden crear un número infinito de ligaduras distintas.

Observe que el uso de esos símbolos en la propia música sigue ciertas reglas que LilyPond no comprueba. P.ej., el *quilisma* siempre es la nota intermedia de una ligadura ascendente, y suele caer sobre un intervalo de semitono, pero es perfectamente posible, si bien incorrecto, hacer un quilisma de una nota.

Además de los símbolos de nota, LilyPond define también las instrucciones `\versus`, `\responsum`, `\ij`, `\iij`, `\IJ` y `\IIJ`, que producen los caracteres correspondientes, p.ej. para utilizarlos en la letra, como marcas de sección, etc. Estas instrucciones utilizan caracteres de Unicode especiales y solo funcionan si se usa una fuente tipográfica que los contemple.

La tabla siguiente muestra un conjunto limitado, pero representativo, de ligaduras gregorianas, junto a los fragmentos de código que las producen. La tabla está basada en la tabla de neumas extendidos del segundo volumen del Antiphonale Romanum (*Liber Hymnarius*), publicado en 1983 por los monjes de Solesmes. La primera columna da el nombre de la ligadura, con la forma principal en tipo negrita y las formas licuescentes en cursiva. La tercera forma muestra el fragmento de código que produce dicha ligadura, utilizando Sol, La y Si como alturas de ejemplo.

Neumas de una nota

Formas **Básica** y *Licuescente*	**Salida**	**Código de LilyPond**
Punctum		`\[ b \]`
		`\[ \cavum b \]`
		`\[ \linea b \]`
Punctum Auctum Ascendens		`\[ \auctum \ascendens b \]`
Punctum Auctum Descendens		`\[ \auctum \descendens b \]`
Punctum inclinatum		`\[ \inclinatum b \]`
Punctum Inclinatum Auctum		`\[ \inclinatum \auctum b \]`
Punctum Inclinatum Parvum		`\[ \inclinatum \deminutum b \]`
Virga		`\[ \virga b' \]`

Ligaduras de dos notas

Clivis vel Flexa		`\[ b \flexa g \]`
Clivis Aucta Descendens		`\[ b \flexa \auctum \descendens g \]`
Clivis Aucta Ascendens		`\[ b \flexa \auctum \ascendens g \]`
Cephalicus		`\[ b \flexa \deminutum g \]`
Podatus o Pes		`\[ g \pes b \]`
Pes Auctus Descendens		`\[ g \pes \auctum \descendens b \]`
Pes Auctus Ascendens		`\[ g \pes \auctum \ascendens b \]`
Epiphonus		`\[ g \pes \deminutum b \]`
Pes Initio Debilis		`\[ \deminutum g \pes b \]`
Pes Auctus Descendens Initio Debilis		`\[ \deminutum g \pes \auctum \descendens b \]`

Ligaduras de varias notas

Torculus		`\[ a \pes b \flexa g \]`
Torculus Auctus Descendens		`\[ a \pes b \flexa \auctum \descendens g \]`
Torculus Deminutus		`\[ a \pes b \flexa \deminutum g \]`
Torculus Initio Debilis		`\[ \deminutum a \pes b \flexa g \]`
Torculus Auctus Descendens Initio Debilis		`\[ \deminutum a \pes b \flexa \auctum \descendens g \]`
Torculus Deminutus Initio Debilis		`\[ \deminutum a \pes b \flexa \deminutum g \]`
Porrectus		`\[ a \flexa g \pes b \]`
Porrectus Auctus Descendens		`\[ a \flexa g \pes \auctum \descendens b \]`
Porrectus Deminutus		`\[ a \flexa g \pes \deminutum b \]`
Climacus		`\[ \virga b \inclinatum a \inclinatum g \]`
Climacus Auctus		`\[ \virga b \inclinatum a \inclinatum \auctum g \]`
Climacus Deminutus		`\[ \virga b \inclinatum a \inclinatum \deminutum g \]`
Scandicus		`\[ g \pes a \virga b \]`
Scandicus Auctus Descendens		`\[ g \pes a \pes \auctum \descendens b \]`

Scandicus Deminutus		`\[ g \pes a \pes \deminutum b \]`

Símbolos especiales

Quilisma		`\[ g \pes \quilisma a \pes b \]`
Quilisma Pes Auctus Descendens		`\[ g \quilisma a \pes \auctum \descendens b \]`
Oriscus		`\[ \oriscus b \]`
Pes Quassus		`\[ \oriscus g \pes \virga b \]`
Pes Quassus Auctus Descendens		`\[ \oriscus g \pes \auctum \descendens b \]`
Salicus		`\[ g \oriscus a \pes \virga b \]`
Salicus Auctus Descendens		`\[ g \oriscus a \pes \auctum \descendens b \]`
(Apo)stropha		`\[ \stropha b \]`
Stropha Aucta		`\[ \stropha \auctum b \]`
Bistropha		`\[ \stropha b \stropha b \]`
Tristropha		`\[ \stropha b \stropha b \stropha b \]`
Trigonus		`\[ \stropha b \stropha b \stropha a \]`

Instrucciones predefinidas

Están contemplados los siguientes prefijos de cabeza: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Se pueden acumular los prefijos de cabeza, aunque con ciertas restricciones. Por ejemplo, se pueden aplicar `\descendens` o `\ascendens` a una cabeza, pero no los dos a la misma cabeza.

Se pueden ligar dos cabezas adyacentes con las instrucciones infijas `\pes` y `\flexa` para una línea melódica ascendente y descendente, respectivamente.

Utilice la función musical unaria `\augmentum` para añadir puntos de augmentum.

Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: Sección 17.4.7 [Ligaduras de neumas cuadrados gregorianos], página 556, Sección 17.3.9 [Ligaduras mensurales blancas], página 550, Sección 17.2.2 [Ligaduras], página 542.

Advertencias y problemas conocidos

Cuando aparece un punto de `\augmentum` al final de la última pauta dentro de una ligadura, a veces se posiciona defectuosamente en el sentido vertical. Para rodear el problema, escriba una nota adicional de salto (p.ej. `s8`) como última nota de la pauta.

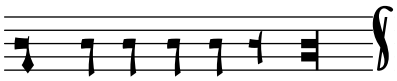
`\augmentum` debería estar implementado como un prefijo de cabeza en vez de una función musical unaria, de forma que `\augmentum` se pudiera entremezclar con los prefijos de cabeza en un orden arbitrario.

17.5 Tipografiado del canto kievano en notación cuadrada

17.5.1 Contextos del canto kievano

De la misma manera que con las notaciones mensural y gregoriana, pueden usarse los contextos predefinidos `KievanVoice` y `KievanStaff` para tipografiar una pieza en notación cuadrada. Estos contextos inician todas las propiedades de contexto necesarias a sus valores adecuados, de manera que pueda ponerse manos a la obra inmediatamente con la introducción del propio canto:

```
% Font settings for Cyrillic
\paper {
  property-defaults.fonts.serif = "Linux Libertine O,serif"
}

\score {
  <<
    \new KievanVoice = "melody" \relative c' {
      c4 c c c c2 b\longa \fine
    }
    \new Lyrics \lyricsto "melody" {
      Го -- спо -- ди по -- ми -- луй.
    }
  >>
}
```



Господи помилуй.

Véase también

Glosario musical: Sección “Kievan notation” en *Glosario Musical*.

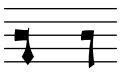
Advertencias y problemas conocidos

LilyPond contempla la notación del canto kievano en el estilo sinodal, tal y como se usa en los cantorales impresos por el Sacro Sínodo Ruso en la década de 1910 y recientemenre reimpreso por la Casa de Publicaciones del Patriarcado de Moscú. LilyPond no contempla las formas anteriores (menos corrientes) de notación kievana que se usaban en Galitzia para la notación del canto llano ruteno.

17.5.2 Claves del canto kievano

En la notación kievana solamente se utiliza una clave (la clave Tse-fa-ut). Se utiliza para indicar la posición del Do:

```
\clef "kievan-do"
\kievanOn
c'
```



Véase también

Glosario musical: Sección “Kievan notation” en *Glosario Musical*, Sección “clef” en *Glosario Musical*.

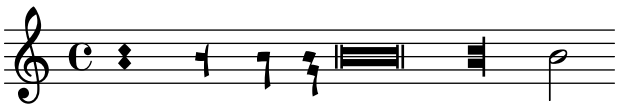
Referencia de la notación: Sección 1.3.1 [Clave], página 20.

17.5.3 Notas del canto kievano

Para la notación cuadrada del canto kievano, debe escogerse el estilo apropiado para la forma de la cabeza de las notas y se deben suprimir los corchetes y las plicas. Esto se consigue mediante una llamada a la función `\kievanOn`, que establece las propiedades adecuadas para la cabeza, plica y corchete de las notas. Una vez que las notas de estilo kievano ya no se necesiten, pueden revertirse estas propiedades llamando a la función `\kievanOff`.

La nota final del canto kievano, que suele ir al final de una pieza musical, puede seleccionarse estableciendo la duración al valor `\longa`. La marca kievana de recitativo, utilizada para indicar la entonación de varias sílabas sobre una sola nota, se puede seleccionar estableciendo la duración al valor `\breve`. El siguiente ejemplo muestra las diversas formas de la cabeza de las notas del canto kievano:

```
\cadenzaOn
\kievanOn
b'1 b'2 b'4 b'8 b'\breve b'\longa
\kievanOff
b'2
```



Véase también

Glosario musical: Sección “Kievan notation” en *Glosario Musical*, Sección “note head” en *Glosario Musical*.

Referencia de la notación: Sección B.9 [Estilos de cabezas de nota], página 927.

Advertencias y problemas conocidos

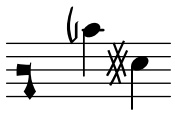
LilyPond determina de forma automática si debe trazarse la forma de una nota con la plica hacia arriba o hacia abajo. Sin embargo, cuando se está tipografiando un canto en notación cuadrada,

es preceptivo que las plicas estén orientadas en la misma dirección dentro de un melisma. Esto puede hacerse manualmente fijando la propiedad `direction` del objeto `Stem`.

17.5.4 Alteraciones accidentales del canto kievano

El estilo `kievan` para las alteraciones accidentales se selecciona con la propiedad `alteration-glyph-name-alist` del grob `Accidental`. El estilo `kievan` provee unos signos de sostenido y de bemol diferentes del estilo predeterminado. No esiste el becuadro en la notación kievana. El símbolo del sostenido no se utiliza en la música sinodal pero puede aparecer en manuscritos más antiguos. Se ha incluido principalmente a efectos de compatibilidad.

```
\clef "kievan-do"
\set Staff.alterationGlyphs =
  #alteration-kievan-glyph-name-alist
bes' dis'
```



Véase también

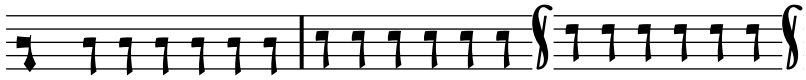
Glosario musical: Sección “Kievan notation” en *Glosario Musical*, Sección “accidental” en *Glosario Musical*.

Referencia de la notación: Sección 1.1.3 [Alteraciones accidentales], página 8, Sección 1.3.5 [Alteraciones accidentales automáticas], página 32, Sección 1.3.6 [Glifos de alteración alternativos], página 40, Sección B.8 [La tipografía Emmentaler], página 912.

17.5.5 Líneas divisorias del canto kievano

En una pauta del tipo `KievanStaff` no hay compases, pero la instrucción `\caesura` crea una barra de frase, y las instrucciones `\section` y `\fine` crean una barra especial de sección.

```
\new KievanStaff {
  c'4 4 4 4 4 4 \caesura  % \bar "."
  d'4 4 4 4 4 4 \section  % \bar "k"
  e'4 4 4 4 4 4 \fine
}
```



Véase también

Referencia de la notación: Sección 2.5 [Compases], página 120, Sección B.8 [La tipografía Emmentaler], página 912.

17.5.6 Melismas del canto kievano

Las notas que están dentro de un melisma del canto kievano se suelen colocar a poca distancia entre sí y cada melisma está separado de los demás por un espacio vacío. Esto se hace para permitir al cantor la identificación rápida de las estructuras melódicas del *canto Znamenny*. En LilyPond, los melismas se tratan como ligaduras y el especiado está implementado por el grabador `Kievan_ligature_engraver`.

Cuando se usan los contextos `KievanVoice` y `KievanStaff`, se habilita el grabador `Kievan_ligature_engraver` de forma predeterminada. En otros contextos, se puede invocar sustituyendo el grabador `Ligature_bracket_engraver` por `Kievan_ligature_engraver` en el bloque layout:

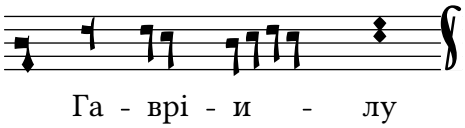
```
\layout {
```

```
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Kievan_ligature_engraver
  }
}
```

El espaciado entre las notas que están dentro de una misma ligadura kievana puede controlatse estableciendo la propiedad `padding` de `KievanLigature`.

El ejemplo siguiente muestra el uso de las ligaduras del canto kievano:

```
% Font settings for Cyrillic
\paper {
  property-defaults.fonts.serif = "Linux Libertine O,serif"
}

\score {
  <<
    \new KievanVoice = "melody" \relative c' {
      e2 \[ e4( d4 ) \] \[ c4( d e d) \] e1 \fine
    }
    \new Lyrics \lyricsto "melody" {
      Га -- врі -- и -- лу
    }
  >>
}
```



Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: Sección 17.3.9 [Ligaduras mensurales blancas], página 550, Sección 17.4.7 [Ligaduras de neumas cuadrados gregorianos], página 556, Sección 17.2.2 [Ligaduras], página 542.

Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras es mediocre.

17.6 Trabajar con música antigua: escenarios y soluciones

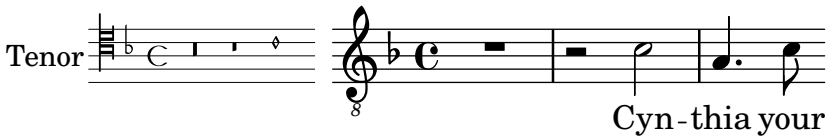
El trabajo con música antigua implica con frecuencia tareas adicionales que difieren considerablemente de la notación moderna para la que se diseñó LilyPond. En el resto de esta sección se perfilan unos cuantos escenarios típicos, con sugerencias de soluciones. Entre ellos están los siguientes:

- cómo hacer incipits (o sea, material a modo de prefacio que indica cómo era el original) a las transcripciones modernas de música mensural;
- cómo conseguir el diseño *Mensurstriche* que se usa con frecuencia para las transcripciones modernas de música polifónica;
- cómo transcribir canto gregoriano en notación moderna;
- cómo generar notación antigua y moderna a partir de la misma fuente.

17.6.1 Incipits

Al transcribir música mensural a la notación moderna, es costumbre poner una indicación de la forma en que aparecían en la versión original los silencios y la nota o notas iniciales (incluso con las claves originales). Esto recibe el nombre de *incipit*. La instrucción `\incipit` usa el valor de `indent` del pentagrama principal para fijar el espacio horizontal que ocupa el incipit junto al nombre del instrumento, y `incipit-width` para establecer la anchura del pentagrama del propio incipit. Un argumento opcional indica cuánto debe desplazarse el incipit hacia la izquierda.

```
\score {
  \new Staff <<
    \new Voice = Tenor {
      \set Staff.instrumentName = "Tenor"
      \override Staff.InstrumentName.self-alignment-X = #RIGHT
      \incipit #1 { \clef "mensural-c4" \key f \major r\breve r1 c'1 }
      \clef "treble_8"
      \key f \major
      R1 r2 c'2 |
      a4. c'8
    }
    \new Lyrics \lyricsto Tenor { Cyn -- thia your }
  >>
  \layout
  {
    indent = 5\cm
    incipit-width = 3\cm
  }
}
```



By default, LilyPond uses a 'MensuralStaff' context for typesetting an incipit. Other contexts can be used by directly writing `\incipit \new` *contexttype* `...`; in this case a 'MensuralStaff' wrapper is only used when it can contain the specified context.

Advertencias y problemas conocidos

Observe que se debe configurar el contenido de `instrumentName` dentro de la música, para que aparezca el incipit. Si no quiere que aparezca ningún nombre para el instrumento, utilice `\set Staff.instrumentName = ""`.

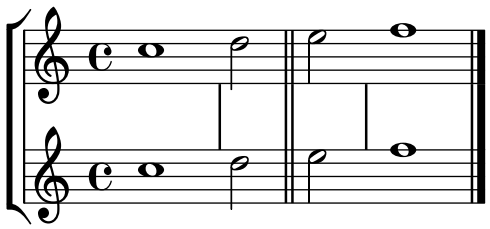
17.6.2 Disposición tipo «Mensurstriche»

Mensurstriche ('líneas de mensuración') es el término aceptado para las barras de compás que se trazan entre los pentagramas de un sistema pero no dentro de los propios pentagramas. Es una forma común de preservar el aspecto visual de las duraciones del original, es decir, sin tener que dividir las notas sincopadas en las barras de compás, y al mismo tiempo ofrecer la ayuda orientativa que proporcionan las barras de compás.

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir fijando `measureBarType` al valor `"-span|"` y usando un contexto de agrupamiento que admita barras extendidas, como `StaffGroup`.

```
\layout {
```

```
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



17.6.3 Transcripción de canto gregoriano

El canto gregoriano se puede transcribir a notación moderna con ciertos trucos sencillos.

Plicas. Todas las plicas se pueden eliminar de una vez retirando el grabador `Stem_engraver` del contexto de voz:

```
\layout {
  ...
  \context {
    \Voice
    \remove Stem_engraver
  }
}
```

Compás. Para el canto no medido, existen diversas alternativas.

Se puede eliminar el grabador Time_signature_engraver del contexto de Staff sin ningún efecto secundario negativo. La alternativa, que es hacer transparente la indicación de compás, deja un espacio en blanco en la partitura, pues la indicación invisible sigue ocupando un espacio.

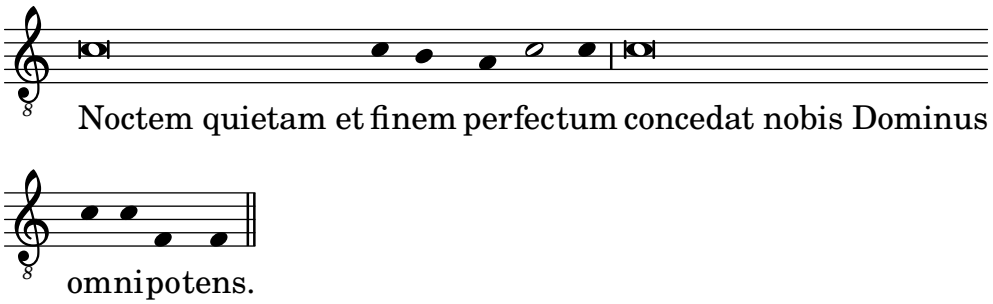
En muchos casos da un buen resultado `\set Score.timing = ##f`. Otra alternativa es utilizar `\cadenzaOn` y `\cadenzaOff`.

Para quitar las barras de compás, el enfoque radical consiste en retirar (mediante la instrucción `\remove`) el grabador Bar_engraver del contexto de pentagrama Staff. Una vez más, en vez de ello podemos usar `\hide BarLine` si se necesita ocasionalmente una barra de compás.

Los contextos de pauta predefinidos para música antigua permiten saltos de línea sin barras de compás. Se puede extender este comportamiento al resto de los contextos con `\set Score.forbidBreakBetweenBarLines = ##f` o a pautas específicas con `\set Staff .forbidBreakBetweenBarLines = ##f`.

Un tipo de transcripción muy común es el canto de recitativo, en el que la nota repetida se indica con una breve. Se puede tratar el texto del tono de recitativo de dos maneras. Podemos ponerlo como una sola sílaba alineada por la izquierda:

```
chant = \relative {
  \clef "G_8"
  c'\breve c4 b4 a c2 c4 \divisioMaior
  c\breve c4 c f, f \finalis
}

words = \lyricmode {
  \tweak self-alignment-X #LEFT "Noctem quietam et"
    fi -- nem per -- fec -- tum
  \tweak self-alignment-X #LEFT "concedat nobis Dominus"
    om -- ni -- po -- tens.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" {
      \chant
    }
    \new GregorianTranscriptionLyrics = "one" {
      \lyricsto "melody" \words
    }
  >>
}
```

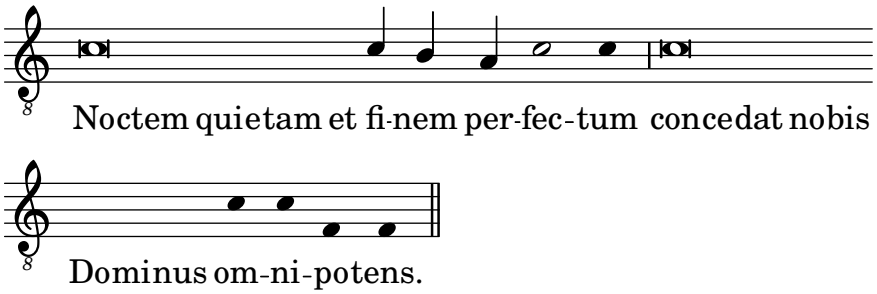


Esto funciona bien en la medida en que el texto no comprenda un salto de línea. Si es el caso, una alternativa sería añadir notas ocultas a la partitura, en combinación con cambios en la visibilidad de las plicas:

En ciertos estilos de transcripción, se usan plicas de forma ocasional, por ejemplo para indicar la transición a partir de un recitativo monotónico a un gesto melódico fijo. En estos casos podemos usar el grabador `Stem_engraver` y hacer manualmente `\omit Stem` y `\undo \omit Stem`.

```
chant = \relative {
  \clef "G_8"
  \omit Stem
  \omit Flag
  c'\breve*1/16 \hide NoteHead c8 c c c c
  \undo \hide NoteHead
  \undo \omit Stem \stemUp c4 b4 a
  \omit Stem c2 c4 \divisioMaior
  c\breve*1/16 \hide NoteHead c8 c c c c c c
```

```
  \undo \hide NoteHead c4 c f, f \finalis
}

verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" {
      \chant
    }
    \new GregorianTranscriptionLyrics = "one" {
      \lyricsto "melody" \verba
    }
  >>
  \layout {
    \context {
      \GregorianTranscriptionVoice
      \consists Stem_engraver
    }
  }
}
```

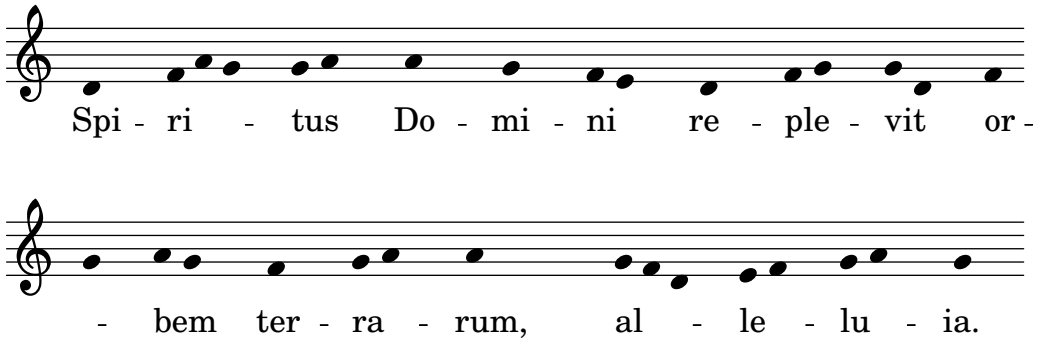


Otra situación común es la transcripción de cantos neumáticos o melismáticos, es decir, cantos con un número variable de notas en cada sílaba. En este caso, podemos establecer los grupos de sílabas claramente separados, normalmente también las subdivisiones de un melisma mayor. Una forma de conseguirlo es usar un `\time` fijo, p.ej. 1/4, y dejar que cada sílaba o grupo de notas llene uno de esos compases, con la ayuda de grupos de valoración especial o duraciones más breves. Si las líneas divisorias y todas las demás indicaciones rítmicas se hacen transparentes, y el espacio que rodea las líneas divisorias se aumenta, ello proporciona una aceptable representación del original, en notación moderna.

Para evitar que las sílabas de distinta anchura (como "-ri" y "-rum") dispongan los grupos de notas de las sílabas de forma irregular, se puede ajustar la propiedad `'X-extent` del objeto `LyricText` a un valor fijo. Otra forma más engorrosa sería añadir las sílabas como elementos de `\markup`. Si se necesitan más ajustes posteriormente, esto se puede hacer fácilmente con 'notas' s.

```
spiritus = \relative {
  \time 1/4
  d'4 \tuplet 3/2 { f8 a g } g a a4 g f8 e
  d4 f8 g g8 d f g a g f4 g8 a a4 s
  \tuplet 3/2 { g8 f d } e f g a g4
```

```
}

spirLyr = \lyricmode {
  \override Lyrics.LyricText.X-extent  = #'(0 . 3)
  Spi -- ri -- _ _ tus _ Do -- mi -- ni _
  re -- ple -- _ vit _ or -- _ bem _ ter -- ra -- _ rum,
  al -- _ _ le -- _ lu -- _ ia.
}
\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "chant" {
      \spiritus
    }
    \new GregorianTranscriptionLyrics = "one" {
      \lyricsto "chant" \spirLyr
    }
  >>
  \layout {
    \context {
      \GregorianTranscriptionStaff
      measureBarType = ""
      \override BarLine.X-extent = #'(-1 . 1)
      \hide TupletNumber
      \hide TupletBracket
    }
  }
}
```



17.6.4 Antigua y moderna desde la misma fuente

Uso de etiquetas para producir música mensural y moderna a partir del mismo código fuente

Usando etiquetas es posible producir las dos notaciones mensural y moderna a partir del mismo código fuente de la música. En este fragmento se introduce la función `\menrest`, que permite que los silencios mensurales estén a la altura de notas como en el original, pero con silencios modetnos en la posicion del pentagrama estándar.

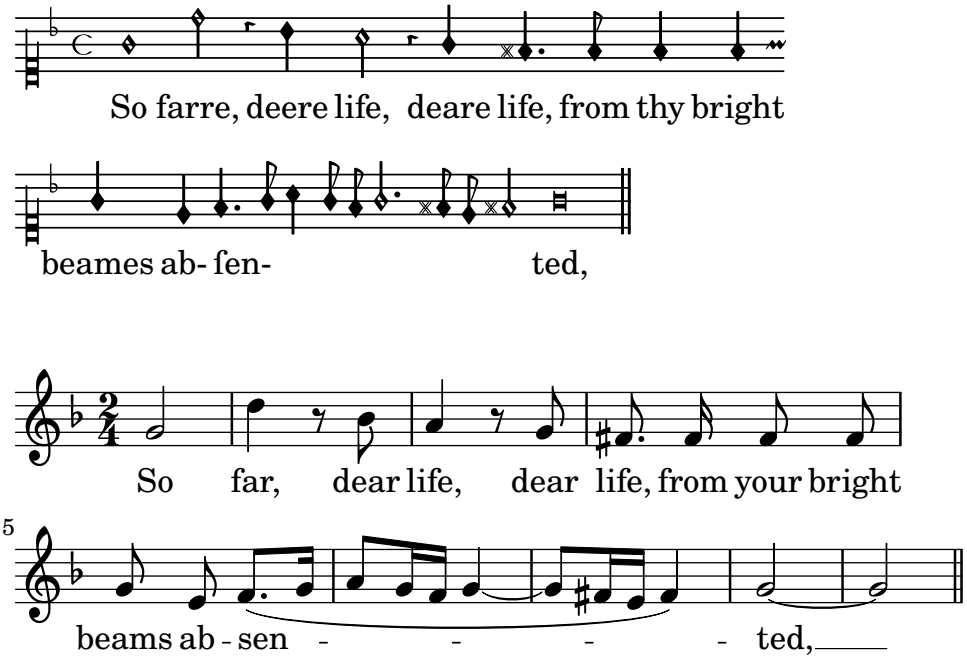
También se pueden usar etiquetas allí donde se necesiten otras diferencias: por ejemplo usando "silencios de compás completo" (`R1`, `R\breve`, etc.) en música moderna, pero silencios normales (`r1`, `r\breve`, etc.) en la versión mensural. La conversión de música mensural a su equivalente moderno se conoce normalmente como *transcription*.

La llamara `c4.\Be c8 c\Am` es lo mismo que `c4.[ c8 c]`. Sin embargo, suprime las advertencias sin se inicia en una nota que no puede llevar barra pero la necesita en todo caso debido al uso del grabador `Completion_heads_engraver`.

[La longitud ligeramente acortada de la línea en la pauta mensural evita el recorte del glifo de los custos cuando LilyPond genera imágenes recortadas muy estrechamente. El uso de `\with-true-dimensions`, tal y como aparece a continuación, evita esto.]

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
              #{
                \tag #'mens $(make-music 'RestEvent note)
                \tag #'mod $(make-music 'RestEvent note 'pitch '())
              #})
Be = \tag #'mod
       #(begin
          (ly:expect-warning (G_ "stem does not fit in beam"))
          (ly:expect-warning (G_ "beam was started here"))
          (make-span-event 'BeamEvent START))
Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

Music = \relative c'' {
  \key f \major
  g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
  g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}

MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, __
}

\score {
  \keepWithTag #'mens {
    <<
      \new PetrucciStaff {
        \new PetrucciVoice = "Cantus" {
```

```
          \clef "petrucci-c1" \time 4/4 \MenStyle \Music
        }
      }
      \new Lyrics \lyricsto "Cantus" \MenLyr
    >>
  }
  \layout {
    \context {
      \PetrucciVoice
      % No longer necessary starting with version 2.25.23.
      \override Flag.style = #'mensural
    }
  }
}

\markup\vspace #1

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff {
        \new Voice = "Sop" \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver"
        } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
      }
      \new Lyrics \lyricsto "Sop" \ModLyr
    >>
  }
}
```



18 Músicas del mundo

El propósito de esta sección es remarcar los problemas de notación que son de relevancia a las tradiciones musicales distintas a la occidental.

18.1 Notación común para músicas no occidentales

Esta sección estudia la forma de introducir e imprimir partituras musicales que no pertenecen a la tradición clásica occidental, también conocida como *Período de la práctica común*.

18.1.1 Extensión de la notación y los sistemas de afinación

La notación clásica estándar (también conocida como la notación del *Período de la práctica común*) se usa normalmente en todo tipo de música, sin limitación a la música occidental 'clásica'. Esta notación se estudia en Sección 1.1 [Escritura de notas], página 3, y los distintos nombres de las notas que se pueden utilizar están explicados en Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10.

Ciertos tipos de música no occidental, folk y tradicional con frecuencia emplean sistemas de afiniación altenativos o extendidos que no encajan fácilmente dentro de la notación clásica estándar.

Muchas veces se usa la notación estándar, pero con algunas diferencias de afinación implícitas. Por ejemplo, la notación de la *música árabe* se realiza con alteraciones accidentales de semitono y de cuarto de tono, estando determinadas por el contexto las exactas variaciones en la entonación. En el caso de la *música árabe*, el archivo de inicio `arabic.ly` proporciona un conjunto adecuado de macros y definiciones que extienden la notación estándar que emplea los nombres de nota italianos. Para ver más detalles, consulte Sección 18.2 [Música árabe], página 573.

Otros tipos de música requieren notaciones extendidas o singulares; por ejemplo, la *música clásica turca* o música clásica otomana, por ejemplo, emplea formas melódicas conocidas como *makamlar* cuyos intervalos están basados en divisiones de 1/9 de tono. Se siguen usando las notas del pentagrama occidental, pero con alteraciones accidentales que están definidas de forma inequívoca en el archivo `turkish-makam.ly`. Para ver más información sobre la música clásica turca y los *makamlar*, consulte Sección 18.3 [Música clásica de Turquía], página 580.

También están disponibles otros archivos de inicio relacionados, `hel-arabic.ly` and `makam.ly`.

Para localizar estos archivos de inicio en su sistema, consulte Sección "Otras fuentes de información" en *Manual de Aprendizaje*.

Véase también

Glosario musical: Sección "Common Practice Period" en *Glosario Musical*, Sección "makamlar" en *Glosario Musical*.

Manual de aprendizaje: Sección "Otras fuentes de información" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.1 [Escritura de notas], página 3, Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10, Sección 18.2 [Música árabe], página 573, Sección 18.3 [Música clásica de Turquía], página 580, Sección 18.4 [Música clásica persa], página 582.

18.2 Música árabe

Esta sección destaca asuntos relativos a la música árabe.

18.2.1 Referencias para música árabe

La música árabe ha sido principalmente una tradición oral hasta el momento. Cuando se transcribe la música, se suele hacer en la forma de un esbozo sobre el que los intérpretes deben improvisar

ampliamente. En una medida creciente, se está adoptando la notación occidental con ciertas variantes para preservar y transmitir la música árabe.

Algunos elementos de la notación musical occidental como la transcripción de acordes o las particellas independientes, no se necesitan para tipografiar las piezas árabes, más tradicionales. Sin embargo hay ciertos problemas, como la necesidad de indicar intervalos que tienen una medida intermedia entre el semitono y el tono, además de los intervalos mayores y menores de la música occidental. También existe la necesidad de agrupar e indicar los numerosos maqams (modos) que forman parte de la música árabe.

Por lo general la notación musical árabe no trata de indicar con precisión los elementos microtonales que se encuentran presentes en la práctica musical.

Son de relevancia para la música árabe ciertos asuntos que se encuentran cubiertos en otras partes del manual:

- Los nombres de las notas y sus alteraciones (entre ellos, cuartos de tono) se pueden realizar como se estudia en Sección 18.1 [Notación común para músicas no occidentales], página 573.
- Las armaduras adicionales también pueden realizarse como se describe en Sección 1.3.2 [Armadura de la tonalidad], página 25.
- Los compases complejos pueden requerir agrupaciones manuales de las notas tal y como se describe en Sección 2.4.3 [Barras manuales], página 114.
- Los *takasim* o improvisaciones libres desde el punto de vista rítmico se pueden escribir omitiendo las barras de compás como aparece en Sección 2.3.4 [Música sin compasear], página 91.

Véase también

Referencia de la notación: Sección 18.1 [Notación común para músicas no occidentales], página 573, Sección 1.3.2 [Armadura de la tonalidad], página 25, Sección 2.4.3 [Barras manuales], página 114.

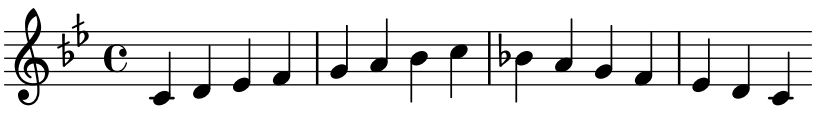
Fragmentos de código: Sección “Músicas del mundo” en *Fragmentos de código*.

18.2.2 Nombres de nota árabes

Los nombres tradicionales de las notas en árabe (como ‘rast’, ‘dukah’, ‘sikah’, etc.) pueden ser bastante largos y por ello no siempre son adecuados para el propósito de escribir música.

Incluya el archivo `arabic.ly` para escribir partituras de música árabe. El ejemplo siguiente muestra cómo escribir una escala ‘rast’:

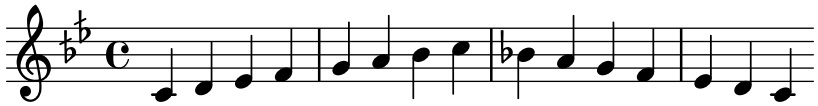
```
\include "arabic.ly"
\relative {
  \key do \rast
  do' re misb fa | sol la sisb do | sib la sol fa | misb re do
}
```



El archivo `arabic.ly` establece el idioma de las notas a italiano (o Solfege), ya que este es el estándar moderno en la música árabe y se encuentra ampliamente adoptado entre los músicos árabes. Si prefiere escribir las notas en otro idioma, cambie el idioma a su idioma preferido directamente después de la instrucción de inclusión del archivo `arabic.ly`. Esto es una escaoa ‘rast’ con nombres de notas en inglés:

```
\include "arabic.ly"
\language "english"
\relative {
```

```
  \key c \rast
  c' d eqf f | g a bqf c | bf a g f | eqf d c
}
```

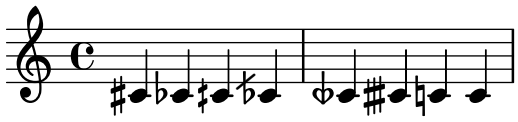


"Rast" es una escala heptatónica que usa intervalos en cuartos de tono y se considera la escala central y más importante del "Maqamat árabe". Para ver la lista completa de las escalas árabes contempladas, véase Sección 18.2.3 [Armaduras de tonalidad árabes], página 575.

El uso de notación occidental estándar para música no occidental se discute en Sección 18.1 [Notación común para músicas no occidentales], página 573. Véase también Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10.

El símbolo del semi-bemol no coincide con el símbolo que se utiliza en la notación árabe. Se puede usar el símbolo `\dwn` definido en `arabic.ly` precediendo a un bemol como vía alternativa si es importante usar el símbolo árabe específico del semi-bemol. El aspecto del símbolo del semi-bemol dentro de la armadura no se puede alterar mediante este método.

```
\include "arabic.ly"
\relative {
  \set Staff.extraNatural = ##f
  dod' dob dosd \dwn dob dobsb dodsd do do
}
```



Véase también

Referencia de la notación: Sección 1.1.5 [Nombres de las notas en otros idiomas], página 10, Sección 18.1 [Notación común para músicas no occidentales], página 573, Sección 22.1 [Inclusión de archivos de LilyPond], página 627.

Archivos instalados: `ly/arabic.ly`

Fragmentos de código: Sección "Músicas del mundo" en *Fragmentos de código*.

18.2.3 Armaduras de tonalidad árabes

Además de las armaduras de tonalidad mayores y menores, LilyPond proporciona las armaduras de tonalidad árabes más comunes en el archivo `arabic.ly`. Dicho esto, LilyPond no pretende dar soporte al conjunto completo de todos los maqams posibles. Más bien, define los más comunes que se usan con frecuencia y ofrece armaduras de tonalidad para formar grupos de maqams similares. En general, un maqam usa la armadura de su familia, o una familia vecina, y las alteraciones accidentales que varían se marcan a lo largo de la música. Cuando se forman armaduras de tonalidad, las familias de los maqam vecinos se agrupan. Por ejemplo, el maqam *saba* aparece rara vez fuera del contexto del maqam *bayati* y añade solo una alteración. Aunque los dos maqams provienen de distintas *familias de maqam* están dentro del mismo *grupo* de armaduras. Los maqams árabes admiten sólo modulaciones limitadas, a causa de la naturaleza de los instrumentos musicales árabes.

He aquí un ejemplo de armadura de tonalidad para una pieza del tipo "maqam muhayer":

```
\key re \bayati
```

Aquí, *re* es la altura predeterminada del maqam muhayer, y *bayati* es el nombre del maqam básico dentro del grupo.

Al tiempo que la armadura indica el grupo, el título suele indicar el maqam específico, y así en este ejemplo el nombre del "maqam muhayer" debe aparecer también en el título.

Se pueden indicar de la misma forma otros maqams dentro del mismo grupo *bayati*, como aparece en la tabla a continuación, (p. ej. *bayati*, *hussaini*, *saba* y *ushaq*). Todos ellos son variantes del maqam básico y más común dentro del grupo, que es *bayati*. Normalmente se diferencian del maqam básico en su tetracordo superior, o en ciertos detalles de discurso que no cambian su naturaleza fundamental como parientes.

El otro maqam del grupo (*nawa*) se relaciona con el *bayati* a través de una modulación y se indica en la tabla entre paréntesis para aquellos que son modulaciones de su maqam básico. El *Nawa*, por ejemplo, se puede indicar de la siguiente forma:

```
\key sol \bayati
```

En la música árabe, el mismo término (p.ej. *bayati*) que se utiliza para indicar una familia de maqam, es también un maqam que normalmente es el más importante de la familia, y podemos verlo como el *maqam básico*.

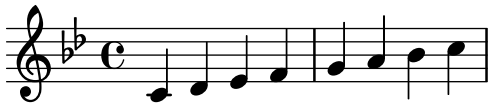
A continuación presentamos un agrupamiento que asigna los maqams más comunes a armaduras de tonalidad según se define en el archivo `arabic.ly`:

grupo de maqam	**tono**	**final**	**otros maqams del grupo (final)**
ajam	`major`	sib	jaharka (fa)
bayati	`bayati`	re	hussaini, muhayer, saba, ushaq, nawa (sol)
hijaz	`hijaz`	re	zanjaran (do)
hijaz kar	`hijaz_kar`	do	shahnaz, shad arban (sol)
huzam	`huzam`	misb	-
iraq	`iraq`	sisb	-
kurd	`kurd`	re	hijazkar kurd (do)
nahawand	`minor`	do	busalik (re), farah faza (sol)
nakriz	`nakriz`	do	nawa athar, hisar (re)
rast	`rast`	do	mahur, yakah (sol)
sikah	`sikah`	misb	-

En el caso de que echemos en falta un maqam específico, lo podemos definir por nosotros mismos en la partitura antes de usarlo. El ejemplo siguiente define, y luego utiliza, el maqam *zanjaran*.

```
\include "arabic.ly"

% For example on do: do reb mi fa sol la sib do
% reb and sib are FLAT
% You can also use SHARP, SEMI-FLAT, SEMI-SHARP
zanjaran = #`(
  (0 . ,NATURAL)
  (1 . ,FLAT)
  (2 . ,NATURAL)
  (3 . ,NATURAL)
  (4 . ,NATURAL)
  (5 . ,NATURAL)
  (6 . ,FLAT)
)

\relative {
  \key do \zanjaran
  do' reb mi fa sol la sib do
```

```
}
```



Para casos especiales, los maqam poco frecuentes están definidos en el archivo `hel-arabic.ly`. Diríjase al archivo incluido con LilyPond para ver una lista completa de los maqams que se proveen.

Fragmentos de código seleccionados

Armaduras de tonalidad no tradicionales

La muy utilizada instrucción `\key` establece la propiedad `keyAlterations` dentro del contexto `Staff`.

Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente. El formato de esta instrucción es una lista:

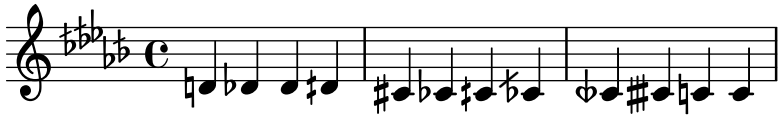
```
\set Staff.keyAlterations =
  #`(((octava . paso) . alteración) ((octava . paso) . alteración) ...)
```

donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es `SHARP`, `FLAT` o `DOUBLE-SHARP`, etc., precedido de una coma.

De forma alternativa, podemos usar el formato más conciso (*paso . alteración*) para cada elemento de la lista si se usa la misma alteración en todas las octavas.

Para escalas microtonales en las que un "sostenido" no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
  % \set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Véase también

Glosario musical: Sección "maqam" en *Glosario Musical*, Sección "bayati" en *Glosario Musical*, Sección "rast" en *Glosario Musical*, Sección "sikah" en *Glosario Musical*, Sección "iraq" en *Glosario Musical*, Sección "kurd" en *Glosario Musical*.

Manual de aprendizaje: Sección "Alturas y armaduras" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.3.2 [Armadura de la tonalidad], página 25.

Archivos instalados: `ly/arabic.ly` `ly/hel-arabic.ly`

Fragmentos de código: Sección “Músicas del mundo” en *Fragmentos de código*, Sección “Alturas” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “KeySignature” en *Referencia de Funcionamiento Interno*.

18.2.4 Indicaciones de compás árabes

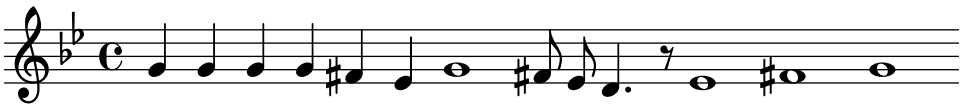
Ciertas formas clásicas de música árabe y turca como el *Semai* utilizan compases poco comunes como el 10/8. Ello puede derivar en una agrupación automática de las notas muy distinta a la de las partituras impresas existentes, donde las notas pueden no estar agrupadas sobre el pulso, sino de una forma difícil de emular ajustando el barrado automático. La alternativa sería desactivar el barrado automático y realizar un barrado manual de las notas. Incluso si no es necesaria una correspondencia con música escrita ya existente, puede seguir siendo deseable ajustar el comportamiento del barrado automático y/o utilizar indicaciones de compás complejas o de amalgama.

Fragmentos de código seleccionados

Improvisación de música árabe

Para las improvisaciones o *taqasim* que son libres durante unos momentos, se puede omitir la indicación de compás y se puede usar `\cadenzaOn`. Podría ser necesario ajustar el estilo de alteraciones accidentales, porque la ausencia de líneas divisorias hace que la alteración aparezca una sola vez. He aquí un ejemplo de cómo podría ser el comienzo de una improvisación *hijaz*.

```
\include "arabic.ly"

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```



Véase también

Glosario musical: Sección “semai” en *Glosario Musical*, Sección “taqasim” en *Glosario Musical*.

Referencia de la notación: Sección 2.4.3 [Barras manuales], página 114, Sección 2.4.1 [Barras automáticas], página 101, Sección 2.3.4 [Música sin compasear], página 91, Sección 1.3.5 [Alteraciones accidentales automáticas], página 32, Sección 2.4.2 [Establecer el comportamiento de las barras automáticas], página 104, Sección 2.3.1 [Indicación de compás], página 79.

Archivos instalados: `ly/arabic.ly`

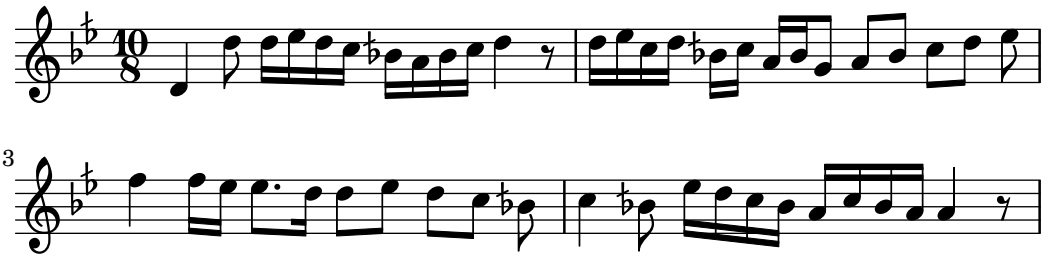
Fragmentos de código: Sección “Músicas del mundo” en *Fragmentos de código*.

18.2.5 Ejemplo de música árabe

A continuación aparece una plantilla que emplea también el comienzo de un *Semai* turco muy común en la educación musical árabe con el objeto de ilustrar algunas de las peculiaridades de la notación árabe, como los intervalos intermedios y los modos poco usuales que tratamos en esta sección.

```
\include "arabic.ly"
```

```
\score {
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
  \relative {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8

    re'4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re  [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
  }
}
```



Véase también

Archivos instalados: `ly/arabic.ly`

Fragmentos de código: Sección “Músicas del mundo” en *Fragmentos de código*.

18.2.6 Lecturas adicionales sobre música árabe

Existen ciertas variantes en los detalles sobre la forma en que se agrupan los maqams, a pesar del relativo acuerdo existente acerca de los criterios de agrupación de maqams en función de los tetracordos inferiores comunes, o por medio de la modulación. No es completamente consistente, incluso a veces dentro del mismo texto, la forma en que se deben especificar las armaduras para los maqams concretos. Sin embargo, es usual utilizar una armadura por ‘grupo’ de maqams, más que una armadura distinta para cada maqam concreto.

- *The music of the Arabs*, de Habib Hassan Touma [Amadeus Press, 1996], contiene un tratado de los maqams y sus formas de agrupación.

  También existen algunos sitios web que explican los maqams e incluso ofrecen ejemplos de audio:

  - `https://www.maqamworld.com/`
  - `https://www.turath.org/`
- Varios métodos de los siguientes autores para el *ud*, el laúd árabe, contienen ejemplos de composiciones turcas y árabes, principalmente.
  - Charbel Rouhana
  - George Farah
  - Ibrahim Ali Darwish Al-masri

18.3 Música clásica de Turquía

Esta sección destaca los problemas de relevancia para la notación de la música clásica de Turquía.

18.3.1 Referencias para música clásica de Turquía

La música clásica de Turquía se desarrolló en el imperio Otomano durante un período aproximadamente contemporáneo con la música clásica de Europa, y se ha prolongado hasta los ss.XX y XXI como una viva y característica tradición con sus propias formas composicionales, teorías y estilos de interpretación. Entre sus características más sobresalientes está el uso de intervalos microtonales basados en 'comas' de 1/9 de tono, a partir de los que se construyen las formas melódicas conocidas como *makam* (en plural *makamlar*).

Ciertos asuntos de relevancia respecto a la música clásica de Turquía se tratan en otras partes. Los nombres especiales de las notas y las alteraciones se dan en Sección 18.1 [Notación común para músicas no occidentales], página 573.

18.3.2 Nombres de las notas en turco

Las notas en la música clásica de Turquía tienen tradicionalmente nombres únicos, y la base de las alturas sobre divisiones de 1/9 de tono significan que los *makamlar* utilizan un conjunto de intervalos completamente diferente de las escalas y modos occidentales.

Desde el punto de vista de la notación moderna, es conveniente utilizar las notas occidentales (do, re, mi, o c, d, e, etc.) con alteraciones especiales que elevan o bajan las notas en intervalos de 1/9, 4/9, 5/9 y 8/9 de tono.

Estas alteraciones personalizadas están definidas en el archivo `makam.ly`.

Para ver una explicación más general sobre la notación musical no occidental, consulte Sección 18.1 [Notación común para músicas no occidentales], página 573.

Véase también

Glosario musical: Sección "makam" en *Glosario Musical*, Sección "makamlar" en *Glosario Musical*.

Referencia de la notación: Sección 18.1 [Notación común para músicas no occidentales], página 573.

18.3.3 Armaduras de tonalidad de la música turca

LilyPond da soporte a más de 200 definiciones de armaduras de *makam* (mucho más de las que se utilizan en la música clásica turca), teniendo cada *makam* su propia tónica o finalis (conocida en turco como 'karar').

Es importante tener en cuenta la nota finalis de cada maqam. He aquí un ejemplo en el que *g* es la tónica predeterminada y *rast* es el nombre del maqam.

```
\key g \rast
```

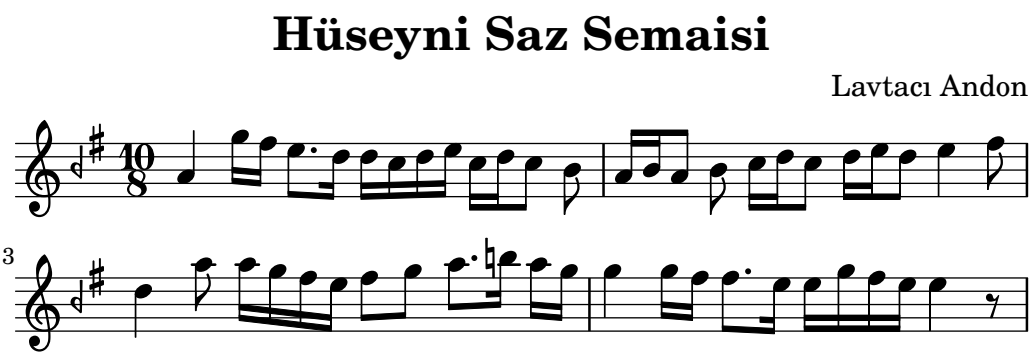
Las alteraciones correctas, bemol de coma (*b1*) y sostenido de coma (`f4`), (ambos en relación a la tónica *g*), se imprimen automáticamente.

Fragmentos de código seleccionados

Ejemplo de makam en la música turca

Esta plantilla usa el comienzo de un *Saz Semai* turco bien conocido que es familiar dentro del repertorio, para ilustrar algunos de los elementos de la notación musical de Turquía.

```
#(set-default-paper-size "a6" 'landscape)

\include "turkish-makam.ly"

\header {
  title = "Hüseyni Saz Semaisi"
  composer = "Lavtacı Andon"
  tagline = ##f
}

\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f

  \key a \huseyni
  \time 10/8

  a'4 g'16[ fb] e8.[ d16] d[ c d e] c[ d c8] bfc |
  a16[ bfc a8] bfc c16[ d c8] d16[ e d8] e4 fb8 |
  d4 a'8 a16[ g fb e] fb8[ g] a8.[ b16] a16[ g] |
  g4 g16[ fb] fb8.[ e16] e[ g fb e] e4 r8 |
}

\layout {
  indent = 0
}
```



18.3.4 Lecturas adicionales sobre música turca

- *Turk Musikisi Nazariyati ve Usulleri: Kudum Velveleleri* de Ismail Hakki Ozkan [(Kultur serisi, 41) (Turkish) Paperback – 1986]
  contiene información acerca de la teoría de los maqams y los usul.
- *Music of the Ottoman Court* de Walter Feldman [VWB Hardback – 1996]
  contiene información sobre la historia de la música cortesana otomana.
- *Turkish Music Makam Guide* de Murat Aydemir [Pan Paperback – 2010]
  contiene información en inglés concerniente al maqam turco, incluyendo dos CDs.

18.4 Música clásica persa

Esta sección destaca asuntos relevantes para la notación de la música clásica de persa.

18.4.1 Notación de la música persa

La notación de la música clásica persa usa normalmente dos alteraciones para los microtonos, *sori* y *koron*. Inventados por Ali-Naqi Vaziri alrededor de 1935, indican que la altura respectiva se eleva o desciende aproximadamente un cuarto de tono.

sori **koron**



El archivo `persian.ly`[1] ofrece soporte para koron y sori; se pueden obtener añadiendo 'k' (koron) y 'o' (sori) después de los nombres de las notas en inglés.

LilyPond contempla afinaciones para todos los principales modos persas en todas las tonalidades, suficientes para llevar a cabo la notación de los *gushehs* (melodías centrales) de todos los *dastgahs* (sistemas musicales modales).

La nota que sigue inmediatamente a un koron es a veces[2] bajada en unos 20 cents. Esto no se refleja en la notación pero se considera parte de la afinación. Sin embargo, para obtener un mejor soporte para el MIDI, podemos hacer un bemol añadiendo 'v' después del nombre de la nota ('*v*lat'). Esta nota debe tener también un intenso vibrato, y el vibrato y la afinación baja están integrados perceptualmente (*serialism*). Esto es solamente para el MIDI y no tiene ningún efecto en la notación en sí.

No hay más asuntos relacionados con la afinación en la música persa. Como la música es monofónica, la diferencia entre la entonación justa (por ejemplo) y el temperamento igual es meramente académica: no hay acordes en los que se pueda percibir algún intervalo desafinado.

Se proveen los siguientes sufijos para los nombres ingleses de las notas.

`ff`	doble bemol
`f`	bemol
`k`	koron (alrededor de un cuarto de bemol, -3/10 de tono, 60 cents)
`o`	sori (alrededor de un cuarto de sostenido, 2/10 de tono, 40 cents)
`s`	sostenido
`x`	doble sostenido
`v`	nota baja en 20 cents ("vlat", sin notación visible)
`fv`	bemol, afinado 20 cents hacia abajo (como un bemol normal en la notación)
`sv`	sostenido, afinado 20 cents hacia abajo (como un sostenido normal en la notación; nunca se encuentra en la música persa tradicional)

18.4.2 Afinaciones persas

Los dastgahs pueden tener muchas afinaciones; se ha convertido en algo común nombrar las afinaciones según dastgahs representativos que las usan. Se proveen las siguientes afinaciones persas para su uso como el segundo argumento de la instrucción `\key`. Observe que en la práctica no todas las afinaciones se usan en todas las tonalidades.

`\shur`	'Shur' gushehs con 5º grado natural

[1] Existe otro archivo más antiguo que da soporte a la música clásica persa, que también se llama `persian.ly` (escrito por Kees van den Doel) que ya no funciona con la versión actual de LilyPond; aunque los nombres de nota son compatibles, la selección de armaduras de tonalidad no lo es.

[2] Si el intervalo definido por la nota antes del koron y después del koron es una tercera menor. Lo mismo vale para la nota que está debajo de la finalis en el dastgah 'Esfahan' de acuerdo con algunos, aunque no todos, los músicos persas.

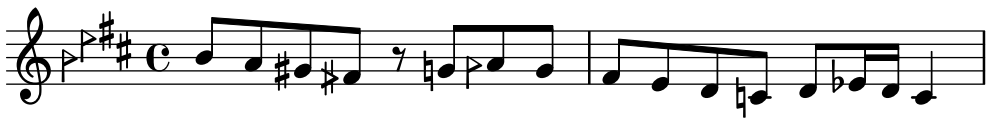
`\shurk`	'Shur' gushehs con 5º grado koron
`\esfahan`	
`\mokhalefsegah`	
`\chahargah`	
`\mahur`	igual que `\major` pero se usa en un contexto totalmente diferente
`\delkashMahur`	

En el dastgah 'Shur', el 5º grado puede ser natural o koron, así que se ofrecen dos afinaciones (`\shur` y `\shurk`). Los dastgahs secundarios 'Dashti', 'Abuata' y 'Bayat-e-tork' usan `\shur`, el dastgah derivado 'Afshari' usa `\shurk`. 'Nava' sobre G es como 'Shur' sobre D (a una 4ª de distancia). La afinación de 'Segah' es `\shurk`. Otras afinaciones son `\esfahan` ('Homayoun' sobre C es como 'Esfahan' sobre G, a una 5ª de distacnia), `\chahargah` y `\mokhalefsegah`. Para los dastgahs 'Mahur' y 'Rast-panjgah' se puede usar la misma afinación `\mahur`.

Todos las diversos gushehs modulatorios en todos los dastgahs se pueden tipografiar con estas afinaciones, con la excepción de 'Delkash' en 'Mahur', pero tradicionalmente las alteraciones peculiares se tipografían explícitamente (sin embargo, `persian.ly` también define una afinación `\delkashMahur`).

Como ejemplo, Chahargah en D se puede introducir de la siguiente forma.

```
\include "persian.ly"

\relative c' {
  \key d \chahargah
  bk'8 a gs fo r g ak g |
  fs ek d c d ef16 d c4 |
}
```

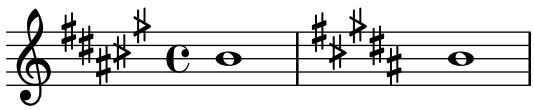


18.4.3 Armaduras de tonalidad persas

De forma predeterminada, el orden de las alteraciones en las armaduras de tonalidad persas es tal que los bemoles van seguidos de los korons, y después los sostenidos seguidos por los soris (después vendrían los dobles bemoles, y luego los dobles sostenidos).

Se puede seleccionar un orden alternativo, similar al que se usa en `turkish-makam.ly`. Aquí, el orden es bemoles o korons (o dobles bemoles), después sostenidos o soris (o dobles sostenidos). En otras palabras, los korons y los soris se tratan de igual manera que los bemoles y los sostenidos, respectivamente.

```
\include "persian.ly"

{
  \key b \chahargah b'1 |
  \set Staff.keyAlterationOrder = \persianAltKeyAlterationOrder
  \key b \chahargah b'1 |
}
```



18.4.4 Lecturas adicionales sobre música persa

- *Traditional Persian Art Music* de Dariush Tala'i [Bibliotheca Persica, Costa Mesa CA, 2000]

  Las afinaciones persas de LilyPond siguen estrechamente este libro.
- *The Dastgah Concept in Persian Music* de Hormoz Farhat [Cambridge University Press, Cambridge, 1990]
- *Le répertoire-modèle de la musique iranienne* de Jean During (en francés) [Sourush, Teherán, 1995]

  Este libro contiene medidas de los intervalos en la práctica real, que es consistente con las afinaciones de `persian.ly`.
- *Armoni-e Musiqi-e Iran* de Ali-Naqi Vaziri (en persa) [1935]
- *Scales and Modes Around the World* de Herman Rechberger [Fennica Gehrman, 2018, ISBN 978-952-5489-07-1]

Entrada y salida generales

19 Modos de entrada

La forma en que se interpreta la notación contenida dentro de un archivo de entrada, está determinada por el modo de entrada en curso. En general, existen dos maneras de especificar el modo: la forma larga, p.ej. `\chordmode`, y la forma corta, p.ej. `\chords`. La forma larga se utiliza normalmente cuando se aporta código de entrada a una variable o cuando se está escribiendo el código directamente dentro de un contexto creado explícitamente. La forma corta crea implícitamente un contexto del tipo adecuado para la entrada, y le pasa esta entrada directamente. Es útil en situaciones sencillas donde no se requiere crear explícitamente el contexto receptor.

Modo de acordes

Se activa con la instrucción `\chordmode` y produce que la entrada se interprete con al sintaxis de la notación de acordes, véase Capítulo 15 [Notación de acordes], página 510. La música en el modo de acordes se imprime como acordes sobre un pentagrama si se introduce en un contexto `Staff`, como nombres de acorde cuando se escribe dentro de un contexto `ChordNames` o como diagramas de posiciones de trastes si se escribe en un contexto `FretBoards`.

El modo de acordes se activa también con la instrucción `\chords`. Esto también produce que el código que sigue se interprete con la sintaxis de la notación de acordes pero además crea implícitamente un contexto `ChordNames` nuevo e imprime la entrada dentro de él como nombres de acordes, véase Sección 15.2.1 [Impresión de los nombres de acorde], página 515.

Modo de percusión

Se activa con la instrucción `\drummode` y produce que el código de entrada se interprete con la sintaxis de la notación de percusión, véase Sección 13.1.2 [Notación básica de percusión], página 490. La música en el modo de percusión se imprime como notas de percusión cuando se escribe dentro de un contexto `DrumStaff`.

El modo de percusión también se activa con la instrucción `\drums`. También hace que el código que sigue se interprete con la sintaxis de la notación de percusión pero además crea implícitamente un contexto `DrumStaff` nuevo e imprime la entrada dentro de él como notas de percusión, véase Sección 13.1.2 [Notación básica de percusión], página 490.

Modo de cifras

Se activa con la instrucción `\figuremode` y hace que el código de entrada se interprete con la sintaxis del bajo cifrado, véase Sección 15.3.2 [Escritura del bajo cifrado], página 529. La música dentro del modo de cifras se imprime como bajo cifrado cuando se escribe dentro de un contexto de `FiguredBass` o de `Staff`.

El modo de cifras también se activa con la instrucción `\figures`. También hace que el código que viene a continuación se interprete con la sintaxis del bajo cifrado pero además crea implícitamente un contexto `FiguredBass` nuevo e imprime la entrada dentro de él como bajo cifrado, véase Sección 15.3.1 [Introducción al bajo cifrado], página 528.

Modos de traste y tablatura

No existen modos de entrada especiales para introducir símbolos de trastes y de tablatura.

Para crear diagramas de trastes, escriba las notas o acordes en el modo de notas e imprímalos dentro de un contexto `TabStaff`, véase Sección 12.1.3 [Tablaturas predeterminadas], página 436.

Para crear diagramas de trastes encima de un pentagrama, escriba las notas o acordes en modo de notas o bien en modo de acordes, y desposítelos dentro de un contexto `FretBoards`, véase Sección 12.1.7 [Diagramas de traste automáticos], página 480. Como alternativa, los diagramas de trastes se pueden introducir como elementos de marcado encima de las notas utilizando la instrucción `\fret-diagram`, véase Sección 12.1.5 [Marcas de diagramas de trastes], página 459.

Modo de letra

Se activa con la instrucción `\lyricmode`, y hace que la entrada se interprete como sílabas de la letra de la canción con duraciones opcionales y modificadores de letra asociados, véase Capítulo 9 [Música vocal], página 349. La entrada en el modo de letra se imprime como sílabas de la letra si se escribe dentro de un contexto `Lyrics`.

El modo de letra también se habilita con la instrucción `\lyrics`. Esto también hace que la entrada siguiente se interprete como sílabas de letra, pero además crea un contexto `Lyrics` nuevo e imprime la entrada dentro de él como sílabas de letra.

Por último, el modo de letra también se puede activar con la instrucción `\addlyrics`. De esta forma también se crea un contexto `Lyrics` nuevo y además se añade una instrucción `\lyricsto` implícita que asocia la letra que viene a continuación con la música precedente, véase Sección 9.1.4 [Duración automática de las sílabas], página 353.

Modo de marcado

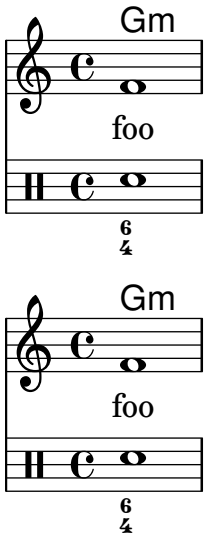
Se activa con la instrucción `\markup`, y hace que la entrada se interprete con la sintaxis del marcado, véase Sección A.1 [Instrucciones de marcado de texto], página 815.

Modo de notas

Es el modo predeterminado o se puede activar con la instrucción `\notemode`. La entrada se interpreta como alturas, duraciones, marcado, etc. y se imprime como notación musical sobre un pentagrama.

Normalmente no es necesario especificar el modo de notas de forma explícita, pero puede ser útil hacerlo en ciertas situaciones, por ejemplo si estamos en el modo de letra, en el modo de acordes o en otro modo y queremos insertar algo que solamente se puede hacer con la sintaxis del modo de notas.

```
% This ...

<<
  \chords { g1:m }
  { f'1 }
  \lyrics { foo1 }
  \drums { sn1 }
  \figures { <6 4>1 }
>>

% ... is equivalent to

<<
  \new ChordNames \chordmode { g1:m }
  \new Voice \notemode { f'1 }
  \new Lyrics \lyricmode { foo1 }
  \new DrumStaff \drummode { sn1 }
  \new FiguredBass \figuremode { <6 4>1 }
>>
```


Gm
foo
6
4
Gm
foo
6
4

20 Estructura del código de entrada

El formato principal de entrada para LilyPond son archivos de texto. Por convenio, el nombre de estos archivos termina en `.ly`.

20.1 Estructura de una partitura

Un bloque `\score` debe contener una expresión musical única delimitada por un par de llaves.[1]

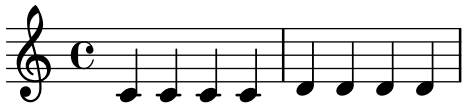
```
\score {
  ...
}
```

> **Nota:** Solo puede haber **una** expresión musical dentro del bloque `\score`, y **debe** estar encerrada entre llaves.

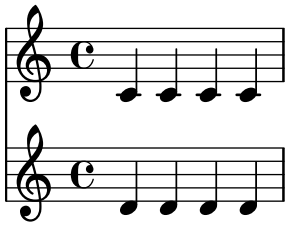
Esta expresión musical única puede tener cualquier tamaño, y puede contener otras expresiones musicales hasta cualquier grado de complejidad. Todos los siguientes ejemplos son expresiones musicales:

```
{ c'4 c' c' c' }
```

```
{
  { c'4 c' c' c' }
  { d'4 d' d' d' }
}
```



```
<<
  \new Staff { c'4 c' c' c' }
  \new Staff { d'4 d' d' d' }
>>
```



```
{
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { \flute }
      \new Staff { \oboe }
    >>
    \new StaffGroup <<
      \new Staff { \violinI }
      \new Staff { \violinII }
    >>
```

[1] Observe que también existe una instrucción de marcado llamada `\score`, véase [Scores within markup], página 865.

```
  >>
}
```

Los comentarios son una excepción a esta regla general (para ver otras excepciones, véase Sección 20.5 [Estructura del archivo], página 594). Tanto los comentarios de una línea como los que están delimitados por %{ ... %} se pueden escribir en cualquier lugar dentro del archivo de entrada. Se pueden escribir dentro o fuera del bloque `\score`, y dentro o fuera de la expresión musical única dentro del bloque `\score`.

Recuerde que incluso dentro de un archivo que contenga solamente un bloque `\score`, éste se encuentra incluido implícitamente dentro de un bloque `\book`. Un bloque `\book` dentro de un archivo fuente produce al menos un archivo de salida, y de forma predeterminada el nombre del archivo de salida que se produce está derivado del nombre del archivo de entrada, de forma que `fandangoforelephants.ly` da como resultado `fandangoforelephants.pdf`.

(Para ver más detalles acerca de los bloques `\book`, consulte Sección 20.2 [Varias partituras en un libro], página 591, Sección 20.3 [Varios archivos de salida a partir de uno de entrada], página 593, y Sección 20.5 [Estructura del archivo], página 594.)

Véase también

Manual de aprendizaje: Sección "Trabajar sobre los archivos de entrada" en *Manual de Aprendizaje*, Sección "Explicación de las expresiones musicales" en *Manual de Aprendizaje*, Sección "La partitura es una (única) expresión musical compuesta" en *Manual de Aprendizaje*.

20.2 Varias partituras en un libro

Un documento puede contener varias piezas de música y texto. Son ejemplos un libro de estudios o una parte orquestal con varios movimientos. Cada movimiento se introduce con un bloque `\score`:

```
\score {
  ...música...
}
```

y los textos se introducen con un bloque `\markup`:

```
\markup {
  ...texto...
}
```

Todos los movimientos y textos que aparecen en el mismo archivo `.ly` se tipografían normalmente en la forma de un solo archivo de salida.

```
\score {
  ...
}
\markup {
  ...
}
\score {
  ...
}
```

Una excepción importante se da dentro de los documentos de `lilypond-book`, en los que tenemos que escribir explícitamente un bloque `\book`, pues en caso contrario solo aparecerá en la salida el primer `\score` o `\markup`.

El encabezamiento de cada pieza se puede poner dentro del bloque `\score`. Antes de cada movimiento, se imprime el nombre de la pieza (`piece`) extraído del encabezamiento. El título

del libro entero se puede poner dentro del bloque \book, pero si no está presente se inserta el encabezamiento \header que está al principio del archivo.

```
\header {
  title = "Ocho miniaturas"
  composer = "Igor Stravinsky"
}
\score {
  \header { piece = "Romanze" }
  ...
}
\markup {
   ...texto de la segunda estrofa...
}
\markup {
   ...texto de la tercera estrofa...
}
\score {
  \header { piece = "Menuetto" }
  ...
}
```

Se pueden agrupar distintas piezas de música en partes de un libro utilizando bloques \bookpart. Las partes de libro se separan por medio de saltos de página, y pueden dar comienzo con un título, como el libro propiamente dicho, mediante la especificación del bloque \header.

```
\bookpart {
  \header {
    title = "Título del libro"
    subtitle = "Primera parte"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Segunda parte"
  }
  \score { ... }
  ...
}
```

Por principio de diseño, no se pueden definir variables dentro de un bloque \book o \bookpart (por cierto, lo mismo puede decirse de \score); esto es de especial importancia si queremos usar varios archivos para preparar la música con variables que deberían ser 'locales' a archivos individuales. Se puede usar la siguiente estructura en tales situaciones.

```
% movement1.ly
variableI = { ... }
bookpartI = \bookpart { \score { ... usar la \variableI ... } }

% movement2.ly
variableII = { ... }
bookpartII = \bookpart { \score { ... usar la \variableII ... } }

% main.ly
```

```
\include "movement1.ly"
\include "movement2.ly"
\book {
  \bookpart { \bookpartI }
  \bookpart { \bookpartII }
}
```

De forma similar, no podemos tener directamente un bloque `\layout` dentro de `\book` o de `\bookpart`. En su lugar, escríbalo dentro de un bloque `\score` que esté a su vez incluido dentro de bloques `\book` o `\bookpart`.

20.3 Varios archivos de salida a partir de uno de entrada

LilyPond crea un archivo de salida por cada bloque `\book`. Si no hay ningún bloque `\book` explícito dentro del archivo de entrada, LilyPond trata el archivo completo implícitamente como un único bloque `\book`, véase Sección 20.5 [Estructura del archivo], página 594.

De forma predeterminada, LilyPond da nombre al archivo de salida usando el nombre del archivo de entrada, y si es necesario, le añade un sufijo con un número creciente (esto es, si un archivo de salida con el mismo nombre ya se ha creado durante el ciclo de ejecución). El comportamiento predeterminado es añadir un sufijo con el número de versión por cada nombre que entre en conflicto, de forma que

```
\book {
  \score { ... }
  \paper { ... }
}
\book {
  \score { ... }
  \paper { ... }
}
\book {
  \score { ... }
  \paper { ... }
}
```

en el archivo fuente `eightminiatures.ly`, produce

```
eightminiatures.pdf
eightminiatures-1.pdf
eightminiatures-2.pdf
```

como archivos de salida.

20.4 Nombres de los archivos de salida

Es posible sobreescribir tanto el nombre del archivo de salida como el sufijo que se añade al nombre de archivo básico dentro de los bloques `\paper`.

```
\paper {
  output-filename = "mi_salida_especial"
}

\book {
  \paper {
    output-suffix = "menuetto"
  }
  ...
```

```
}
\book {
  \paper {
    output-suffix = "scherzo"
  }
  ...
}
```

El resultado son dos archivos de salida con los nombres `mi_salida_especial-menuetto.pdf` and `mi_salida_esecial-scherzo.pdf`. Tenga cuidado al elegir los valores de `output-filename` y de `output-suffix` de forma que sean válidos para los nombres de archivo en su sistema operativo.

Si este nombre de archivo de salida ya existe (puede ocurrir, por ejemplo, si tanto `output-filename` como `output-suffix` están configurados en el bloque `\paper` global y no dentro de algún bloque `\book`), LilyPond añade otro sufijo con un número creciente.

Observe que las variables predefinidas `\paper` (véase Sección 26.1 [El bloque \paper], página 673) se deben poner antes de las asignaciones de `output-filename` y `output-suffix`, por ejemplo

```
bigMargin = \paper { top-margin = 10\cm }

\book {
  \paper {
    \bigMargin % tiene que ir primero
    output-filename = "fulanito"
  }
}
```

20.5 Estructura del archivo

Un archivo `.ly` puede contener cualquier número de expresiones del nivel superior, donde una expresión del nivel superior es una de las siguientes posibilidades:

- Una definición de salida, como `\paper`, `\midi` y `\layout`. Tal definición en el nivel más alto cambia los ajustes predeterminados para todo el libro. Si se introduce más de una de estas definiciones del mismo tipo en el nivel más alto, las definiciones se combinan, pero en situaciones de conflicto las definiciones más tardías tienen preferencia. Para ver los detalles sobre cómo afecta al bloque `\layout`, consulte Sección 27.1 [El bloque \layout], página 686.
- Una expresión directa de Scheme, como `#(set-default-paper-size "a7" 'landscape)` o `#(ly:set-option 'point-and-click #f)`.
- Un bloque `\header`. Esto establece el bloque de encabezamiento global (es decir, al principio del archivo). Es el bloque que contiene los ajustes predeterminados de los campos de encabezamientos como el compositor, título, etc. para todos los libros que están dentro del archivo (véase Sección 21.1.1 [Explicación de los títulos], página 598).
- Un bloque `\score`. Esta partitura se reúne con otras partituras del nivel superior, y se combina con ellas formando un `\book` (libro) único. Este comportamiento se puede cambiar estableciendo el valor de la variable `toplevel-score-handler` en el nivel superior (el *handler* predeterminado se define en el archivo `../scm/lily-library.scm` y su valor se establece en el archivo `../ly/declarations-init.ly`.)
- Un bloque `\book` combina lógicamente varios movimientos (es decir, varios bloques `\score`) en un solo documento. Si hay varias partituras `\score`, se crea un solo archivo de salida para cada bloque `\book`, en el que se concatenan todos los movimientos correspondientes. La única razón para especificar explícitamente bloques `\book` en un archivo de entrada

`.ly` es si queremos crear varios archivos de salida a partir de un único archivo de entrada. Una excepción se da dentro de los documentos de lilypond-book, en los que tenemos que escribir explícitamente un bloque `\book` si queremos tener más de una única partitura `\score` o marcado `\markup` en el mismo ejemplo. Se puede cambiar este comportamiento estableciendo el valor de la variable `toplevel-book-handler` en el nivel superior. El *handler* predeterminado se define en el archivo de inicio `../scm/lily.scm`.

- Un bloque `\bookpart`. Un libro puede dividirse en varias partes, utilizando bloques `\bookpart`, con objeto de facilitar la inserción de saltos de página o usar distintos ajustes de `\paper` en las distintas partes.
- Una expresión musical compuesta, como

```
{ c'4 d' e'2 }
```

Esto añade la pieza al `\score` y la conforma dentro de un libro único junto a todas las demás partituras `\score` y expresiones musicales del nivel superior. En otras palabras, un archivo que contiene solo la expresión musical anterior se traduce a

```
\book {
  \score {
    \new Staff {
      \new Voice {
        { c'4 d' e'2 }
      }
    }
    \layout { }
  }
  \paper { }
  \header { }
}
```

Se puede cambiar este comportamiento estableciendo el valor de la variable `toplevel-music-handler` en el nivel más alto. El *handler* predeterminado se define en el archivo de inicio `../scm/lily.scm`.

- Un texto de marcado, por ejemplo una estrofa

```
\markup {
   2.  Estrofa dos, línea uno
}
```

Los textos de marcado se colocan encima, en medio o debajo de las partituras o expresiones musicales, según donde aparezcan escritos.

- Una variable o una estructura anidada de listas de asociación de Scheme (listas-A)

```
fulano = { c4 d e d }    % variable normal
"Horn 3" = { c4 d e d }  % variable entre comillas
verse.1 = { c4 d e d }   % variable anidada
```

Las variables se pueden usar más tarde en el archivo anteponiéndolas por una barra invertida.

```
\foo
\"Horn 3"
\verse.1
```

El nombre de una variable no puede contener números (ASCII), más de una barra baja seguida, varios signos menos seguidos, ni espacios. Todos los demás caracteres que ofrece Unicode están permitidos, por ejemplo latín, griego, chino o cirílico. También se permiten varias barras bajas únicas y signos menos no seguidos. En otras palabras, funcionan los nombres de variable como `TrompaIII` o `Скрипка-II`.

Se permite cualquier combinación de caracteres si el nombre de la variable está entre dobles comillas (aunque no es recomendable).[2] Ejemplos: `"fulano mengano"`, `"a-b-c"`, `"Trompa 3"`.

Las listas-A anidadas ofrecen un medio para rodear parcialmente la restricción mencionada anteriormente de nombres de variable que no contengan un número. Una lista-A anidada es un nombre de variable válido seguido por uno o más conjuntos de un punto y una clave. En `mus.violin.1 = { a1 }` tenemos una variable llamada `mus`, que tiene una lista de asociación con una clave llamada `violin`, que a su vez tiene una clave llamada `1`, a la cual se asigna finalmente el valor `{ a1 }`. En `"1.2"."3.4".5` el nombre de la variable es `"1.2"` y las claves son `"3.4"` y `5`. Estas construcciones son bastante ilegibles pero pueden ser útiles para generar nombres de variables programáticamente.

Ni un nombre de variable ni una clave dentro de una lista anidada pueden tener el nombre de una nota. Si es necesario, enciérrelas entre comillas dobles, o mejor aún cambie la clave o el nombre conflictivos.

Por razones técnicas, acceder a listas-A anidadas puede fallar bajo ciertas circunstancias. Concretamente, no se puede acceder a ellas en el nivel jerárquico superior (esto es, fuera de un bloque de música). Si la orden `lilypond` abrta con el error 'bad expression type', sustituya la lista-A anidada con una variable normal o entrecomillada.

El ejemplo siguiente muestra tres cosas que se pueden introducir en el nivel superior:

```
\layout {
  % No justificar la salida
  ragged-right = ##t
}

\header {
   title = "Do-re-mi"
}

{ c'4 d' e2 }
```

En cualquier punto del archivo se pueden introducir cualquiera de las instrucciones léxicas siguientes:

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Un comentario de una línea, determinado por un símbolo % al principio.
-
- Un comentario de varias líneas, delimitado por `%{ ... %}`.

Por lo general se ignoran los espacios entre elementos del flujo de entrada, y se pueden omitir o aumentar tranquilamente para mejorar la legibilidad. Sin embargo, los espacios se deben utilizar siempre para evitar errores, en las siguientes situaciones:

- Alrededor de todas y cada una de las llaves curvas de apertura y cierre.
- A continuación de las instrucciones y variables, es decir, todos los elementos que comienzan con un signo \ de barra invertida.
- A continuación de todo elemento que se deba interpretar como expresión de Scheme, es decir, todo elemento que comience por un signo '#'.

[2] En este caso se aplican las reglas estándar para las cadenas de caracteres: las barras invertidas y las comillas dobles dentro de comillas dobles se tienen que escapar con barras invertidas.

- Para separar los elementos de una expresión de Scheme.
- Dentro de `lyricmode` antes y después de las instrucciones `\set` y `\override`.

Véase también

Manual de aprendizaje: Sección “Cómo funcionan los archivos de entrada de LilyPond” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 21.1.1 [Explicación de los títulos], página 598, Sección 27.1 [El bloque `\layout`], página 686.

21 Títulos y encabezamientos

Casi toda la música impresa tiene un título y el nombre del compositor; algunas piezas tienen mucha más información.

21.1 Crear títulos, encabezamientos y pies de página

21.1.1 Explicación de los títulos

Cada bloque `\book` de un solo archivo de entrada produce un archivo de salida distinto, véase Sección 20.5 [Estructura del archivo], página 594. Dentro de cada archivo de salida están disponibles tres tipos de zonas para títulos: *Títulos de libro* al principio de cada libro, *Títulos de parte de libro* al principio de cada parte de un libro y *Títulos de partitura* al comienzo de una partitura.

Los valores de los campos para encabezamientos tales como `title` (para el título) y `composer` (para el autor) se establecen en bloques `\header` (para ver la sintaxis de los bloques `\header` y una lista completa de los campos disponibles de forma predeterminada, véase Sección 21.1.2 [Presentación predeterminada de los títulos de partes de libro y partitura], página 601). Tanto los títulos de libro como los títulos de parte de libro y los títulos de partitura pueden contener los mismos campos, aunque de forma predeterminada los campos de los títulos de partitura se limitan a `piece` (pieza) y `opus`.

Los bloques `\header` se pueden situar en cuatro lugares distintos para formar una jerarquía descendente:

- Al principio del archivo de entrada, antes de cualquier otro bloque `\book`, `\bookpart` o `\score`.
- Dentro de un bloque `\book` pero fuera de los bloques `\bookpart` y `\score` que están dentro de ese libro.
- Dentro de un bloque `\bookpart` pero fuera de cualquier bloque `\score` dentro de dicha parte de libro.
- Dentro de un bloque `\score`.

Los valores de los campos se filtran a través de esta jerarquía, persistiendo aquellos que se establecen más arriba a no ser que se sobreescriban por un valor establecido más abajo en la jerarquía.

- El título de un libro deriva de los campos establecidos al principio del archivo de entrada, modificado por campos establecidos en el bloque `\book`. Los campos resultantes se usan para imprimir el título de libro para dicho libro, siempre y cuando haya otro material que genere una página al comienzo del libro, antes de la primera parte de libro. Es suficiente una sola instrucción `\pageBreak`.
- Un título de parte de libro se deriva de los campos fijados al principio del archivo de entrada, modificado por los campos establecidos dentro del bloque `\book`, y posteriormente modificado por los campos que se fijan dentro del bloque `\bookpart`. Los valores resultantes se usan para imprimir el título de parte de libro para esa parte.
- Un título de partitura se deriva de los campos que se fijan al principio del archivo de entrada, modificados por los campos fijados dentro del bloque `\book`, posteriormente modificados por los campos que se fijan dentro del bloque `\bookpart` y finalmente modificados por los campos establecidos dentro del bloque `\score`. Los valores resultantes se utilizan para imprimir el título de partitura. Observe, sin embargo, que de forma predeterminada solamente se imprimen los campos `piece` y `opus` en los títulos de partitura a no ser que la variable de `\paper`, `print-all-headers`, tenga el valor `#t`.

No es necesario aportar bloques \header en los cuatro lugares: cualquiera de ellos o incluso todos pueden omitirse. De forma similar, los archivos de entrada sencillos pueden omitir los bloques \book y \bookpart, dejando que se creen de forma implícita.

Si el libro solamente tiene una partitura, el bloque \header debiera situarse normalmente al principio del archivo de forma que solamente se produce una zona de título de parte de libro, haciendo que estén disponibles para su uso todos los títulos de encabezamientos.

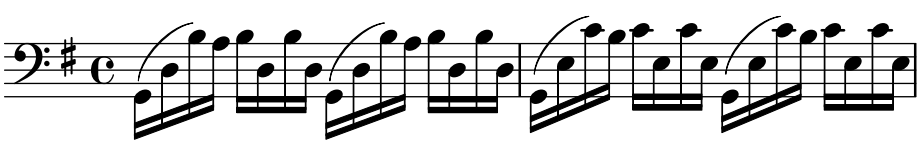
Si el libro tiene varias partituras, son posibles distintas combinaciones de bloques \header, según los distintos tipos de publicaciones musicales. Por ejemplo, si la publicación contiene varias piezas del mismo autor, un bloque \header colocado al principio del archivo que especifique el título del libro y el autor, con bloques \header en cada bloque \score que especifique la piece y/o el opus sería lo más adecuado, como aquí:

```
\header {
  title = "SUITE I."
  composer = "J. S. Bach."
}

\score {
  \header {
    piece = "Prélude."
  }
  \new Staff \relative {
    \clef bass
    \key g \major
    \repeat unfold 2 { g,16( d' b') a b d, b' d, } |
    \repeat unfold 2 { g,16( e' c') b c e, c' e, } |
  }
}

\score {
  \header {
    piece = "Allemande."
  }
  \new Staff \relative {
    \clef bass
    \key g \major
    \partial 16 b16 |
    <g, d' b'~>4 b'16 a( g fis) g( d e fis) g( a b c) |
    d16( b g fis) g( e d c) b(c d e) fis( g a b) |
  }
}
```

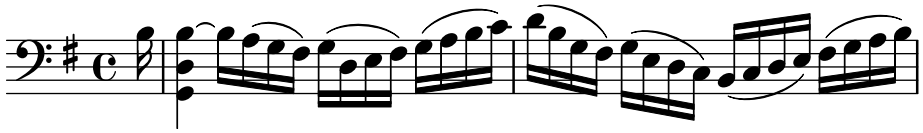
SUITE I.

J. S. Bach.

Prélude.



Allemande.



Son posibles disposiciones más complejas. Por ejemplo, los campos de texto extraídos del bloque \header de un libro se pueden imprimir en todos los títulos de partitura, con ciertos campos sobreescritos y otros suprimidos manualmente:

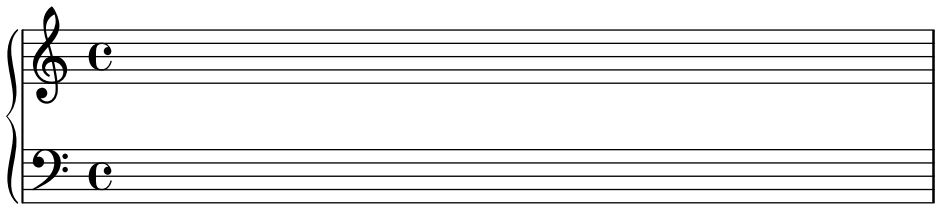
```
\book {
  \paper {
    print-all-headers = ##t
  }
  \header {
    title = "DAS WOHLTEMPERIRTE CLAVIER"
    subtitle = "TEIL I"
    % Do not display the default LilyPond footer for this book
    tagline = ##f
  }
  \markup { \vspace #1 }
  \score {
    \header {
      title = "PRAELUDIUM I"
      opus = "BWV 846"
      % Do not display the subtitle for this score
      subtitle = ##f
    }
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
  }
  \score {
    \header {
      title = "FUGA I"
      subsubtitle = "A 4 VOCI"
      opus = "BWV 846"
      % Do not display the subtitle for this score
      subtitle = ##f
    }
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
  }
}
```

DAS WOHLTEMPERIRTE CLAVIER

TEIL I

PRAELUDIUM I

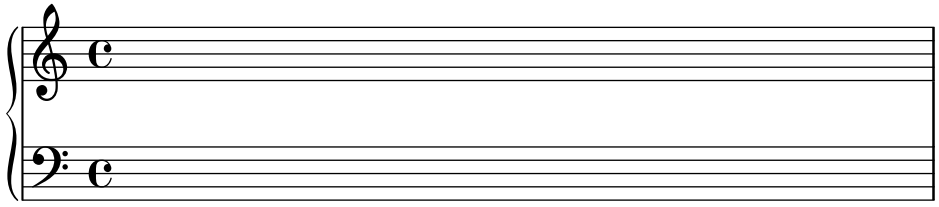
BWV 846



FUGA I

A 4 VOCI

BWV 846



Véase también

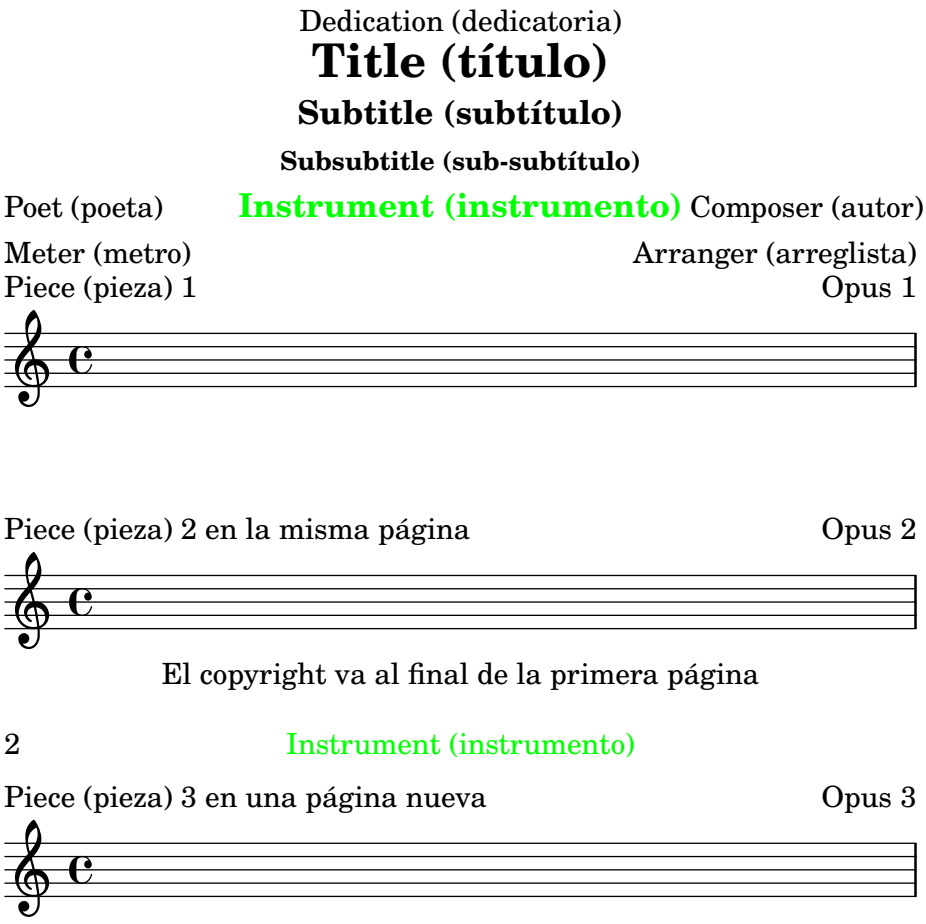
Referencia de la notación: Sección 20.5 [Estructura del archivo], página 594, Sección 21.1.2 [Presentación predeterminada de los títulos de partes de libro y partitura], página 601, Sección 21.2.2 [Personalización de los títulos], página 606.

21.1.2 Presentación predeterminada de los títulos de partes de libro y partitura

El siguiente ejemplo muestra todas las variables impresas de `\header`. Observe que el espaciado vertica entre los distintos elementos de la cabecera en la disposición predeterminada está optimizado para elementos de una sola línea. Si necesita elementos de varias líneas, por ejemplo el nombre del compositor, intente añadir `\vspace` al campo si es necesario para ajustar el espaciado vertical. Una alternativa sería definir su propia disposición personalizada, véase Sección 21.2.2 [Personalización de los títulos], página 606.

```
\book {
  \header {
      % Los siguientes campos están centrados
    dedication = "Dedication (dedicatoria)"
    title = "Title (título)"
    subtitle = "Subtitle (subtítulo)"
    subsubtitle = "Subsubtitle (sub-subtítulo)"

    % Los siguientes campos se distribuyen regularmente
    % sobre una línea
    % el campo "instrument" también aparece
```

```
    % en las siguientes páginas
  instrument = \markup \with-color #green "Instrument (instrumento)"
  poet = "Poet (poeta)"
  composer = "Composer (autor)"

    % Los siguientes campos se sitúan en los
    % extremos opuestos de la misma línea
  meter = "Meter (metro)"
  arranger = "Arranger (arreglista)"

    % Los siguientes campos están centrados en la parte inferior
  tagline = "La tagline o línea de etiqueta va al final de la última página"
  copyright = "El copyright va al final de la primera página"

}
\score {
  \header {
      % Los siguientes campos se colocan en los extremos de una línea
    piece = "Piece (pieza) 1"
    opus = "Opus 1"
  }
  { s1 }
}
\score {
  \header {
    % Los siguientes campos se sitúan en los extremos
    % de una línea
    piece = "Piece (pieza) 2 en la misma página"
    opus = "Opus 2"
  }
  { s1 }
}
\pageBreak
\score {
  \header {
    % Los siguientes campos se sitúan en los extremos
    % de una línea
    piece = "Piece (pieza) 3 en una página nueva"
    opus = "Opus 3"
  }
  { s1 }
}
}
```



La tagline o línea de etiqueta va al final de la última página

Observe que

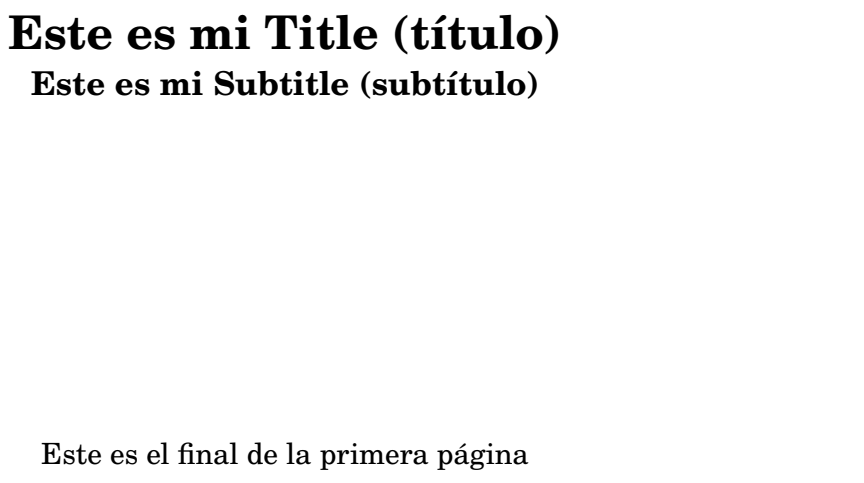
- el nombre de instrumento se repite en todas las páginas,
- solamente `piece` (la pieza) y `opus` se imprimen en una `\score` cuando la variable de papel `print-all-headers` se establece a `##f` (el valor predeterminado),
- los campos de texto que se dejan sin establecer dentro de un bloque `\header` no producen ninguna salida de manera que no ocupen espacio innecesariamente,
- los ajustes predeterminados para `scoreTitleMarkup` sitúan los campos de texto `piece` y `opus` en extremos opuestos de la misma línea.

Para cambiar la disposición predeterminada, véase Sección 21.2.2 [Personalización de los títulos], página 606.

Si un bloque `\book` comienza inmediatamente con un bloque `\bookpart`, no se imprime ningún título de libro, pues no existe ninguna página en que imprimirlo. Si es necesario un título de libro, inicie el bloque `\book` con algún material de marcado o con una instrucción `\pageBreak`.

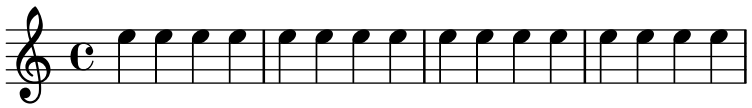
Utilice la variable `breakbefore` dentro de un bloque `\header` que está propiamente dentro de un bloque `\score` para hacer que los títulos del bloque `\header` del nivel superior aparezcan ocupando toda la primera página, empezando la música (definida en el bloque `\score`) en la página siguiente.

```
\book {
  \header {
    title = "Este es mi Title (título)"
    subtitle = "Este es mi Subtitle (subtítulo)"
    copyright = "Este es el final de la primera página"
  }
  \score {
    \header {
      piece = "Esto es la música"
      breakbefore = ##t
    }
    \repeat unfold 4 { e'' e'' e'' e'' }
  }
}
```



2

Esto es la música



LilyPond v2.26.0

Véase también

Manual de aprendizaje: Sección “Cómo funcionan los archivos de entrada de LilyPond” en *Manual de Aprendizaje*

Referencia de la notación: Sección 21.2.2 [Personalización de los títulos], página 606, Sección 20.5 [Estructura del archivo], página 594.

Archivos de inicio: `ly/titling-init.ly`.

21.1.3 Disposición predeterminada de las cabeceras y pies de página

Las *cabeceras* y los *pies* de página son líneas de texto que aparecen en la parte más alta y en la más baja de las páginas separadas del texto principal de un libro. Se controlan mediante las siguientes variables de `\paper`:

- `oddHeaderMarkup` (marcado de cabecera impar)
- `evenHeaderMarkup` (marcado de cabecera par)
- `oddFooterMarkup` (marcado de pie impar)
- `evenFooterMarkup` (marcado de pie par)

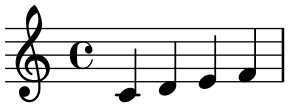
Estas variables de marcado se definen en el archivo `ly/titling-init.ly`. De forma predeterminada hacen lo siguiente.

- los números de página se sitúan automáticamente en el extremo superior izquierdo (si es par) o superior derecho (si es impar), a partir de la segunda página.
- el campo de cabecera `instrument` se sitúa en el centro en cada página, a partir de la segunda página.
- el campo de cabecera `copyright` se centra en la parte baja de la primera página.
- el campo de cabecera `tagline` se centra al final de la última página, y debajo del texto de `copyright` si solo hay una página.

El ejemplo siguiente muestra cómo cambiar la línea inferior de etiqueta.

```
\book {
  \header {
    tagline = "... music notation for Everyone"
  }
  \score {
    \relative {
```

```
      c'4 d e f
    }
  }
}
```



... music notation for Everyone

Para eliminar la línea de pie de página, establezca el valor de `tagline` a `##f`.

Véase también

Referencia de la notación: Sección 21.2.3 [Disposición personalizada de cabeceras y pies de página], página 610.

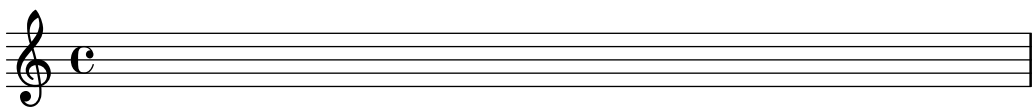
21.2 Títulos, encabezamientos y pies de página personalizados

21.2.1 Formateo personalizado del texto de los bloques de título

Se pueden usar instrucciones `\markup` estándar para personalizar el texto de cualquier cabecera, pie o título dentro del bloque `\header`.

```
\score {
  \header {
    piece = \markup { \fontsize #4 \bold "PRAELUDIUM I" }
    opus = \markup { \italic "BWV 846" }
  }
  { s1 }
}
```

PRAELUDIUM I *BWV 846*



Véase también

Referencia de la notación: Sección 8.2 [Formatear el texto], página 323.

21.2.2 Personalización de los títulos

Las instrucciones `\markup` dentro del bloque `\header` son útiles para dar un formato simple al texto, pero no permiten un control preciso sobre la colocación de los títulos. Para personalizar la colocación de los campos de texto, cambie una o las dos variables de `\paper` siguientes:

- `bookTitleMarkup` (marcado de título del libro)
- `scoreTitleMarkup` (marcado de título de la partitura)

Véase Sección 21.1.2 [Presentación predeterminada de los títulos de partes de libro y partitura], página 601, para la colocación de los títulos cuando se utilizan los valores predeterminados de estas variables de \markup.

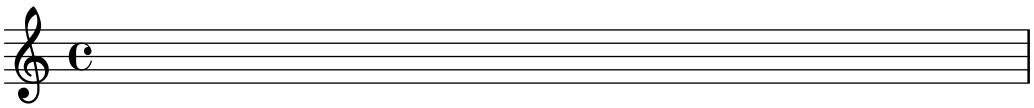
Los ajustes predeterminados para scoreTitleMarkup tal y como están definidos en el archivo ly/titling-init.ly son los siguientes.

```
scoreTitleMarkup = \markup \column {
  \if \should-print-all-headers { \bookTitleMarkup \hspace #1 }
  \fill-line {
    \fromproperty #'header:piece
    \fromproperty #'header:opus
  }
}
```

Esto sitúa los campos de texto piece y opus en extremos opuestos de la misma línea:

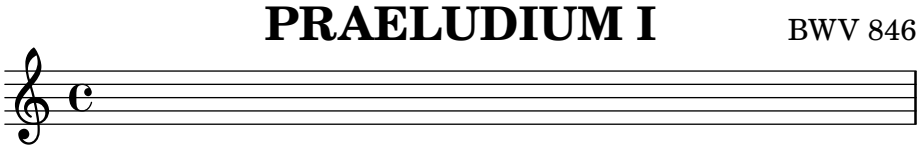
```
\score {
  \header {
    piece = "PRAELUDIUM I"
    opus = "BWV 846"
  }
  { s1 }
}
```

PRAELUDIUM I BWV 846



El ejemplo siguiente redefine scoreTitleMarkup de manera que el campo de texto piece aparece centrado y en un tipo de letra grande y en negrita.

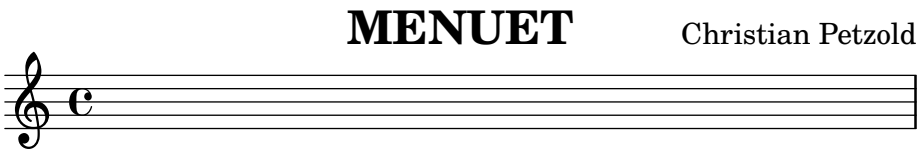
```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:opus
      }
    }
  }
  \header { tagline = ##f }
  \score {
    \header {
      piece = "PRAELUDIUM I"
      opus = "BWV 846"
    }
    { s1 }
  }
}
```

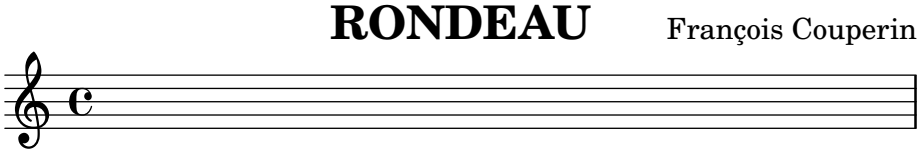


Los campos de texto que normalmente no son efectivos dentro de los bloques `\header` de una partitura se pueden imprimir en la zona del Título de la partitura si se coloca `print-all-headers` dentro del bloque `\paper`. Una desventaja de la utilización de este método es que los campos de texto que están orientados específicamente para la zona del título de parte de libro han de suprimirse manualmente en cada uno de los bloques `\score`. Véase Sección 21.1.1 [Explicación de los títulos], página 598.

Para evitarlo, añada el campo de texto deseado a la definición de `scoreTitleMarkup`. En el ejemplo siguiente, el campo de texto `composer` (asociado normalmente con `bookTitleMarkup`) se añade a `scoreTitleMarkup`, permitiendo que cada partitura muestre un compositor diferente.

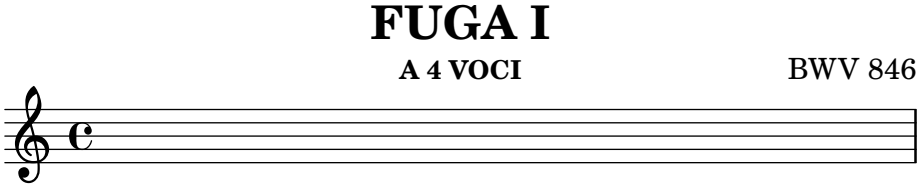
```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
  }
  \header { tagline = ##f }
  \score {
    \header {
      piece = "MENUET"
      composer = "Christian Petzold"
    }
    { s1 }
  }
  \score {
    \header {
      piece = "RONDEAU"
      composer = "François Couperin"
    }
    { s1 }
  }
}
```





También podemos crear nuestros propios campos de texto personalizados, y referirnos a ellos en la definición del elemento de marcado.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \override #`(direction . ,UP)
        \dir-column {
          \center-align \fontsize #-1 \bold
            \fromproperty #'header:mycustomtext %% User-defined field
          \center-align \fontsize #4 \bold
            \fromproperty #'header:piece
        }
        \fromproperty #'header:opus
      }
    }
  }
  \header { tagline = ##f }
  \score {
    \header {
      piece = "FUGA I"
      mycustomtext = "A 4 VOCI" %% User-defined field
      opus = "BWV 846"
    }
    { s1 }
  }
}
```



Véase también

Referencia de la notación: Sección 21.1.1 [Explicación de los títulos], página 598.

21.2.3 Disposición personalizada de cabeceras y pies de página

Las instrucciones `\markup` dentro del bloque `\header` son de utilidad para dar formato al texto de una manera sencilla, pero no permiten un control preciso sobre la colocación de las cabeceras y los pies de página. Para personalizar la colocación de los campos de texto, use una o más de las siguientes variables de `\paper`.

- `oddHeaderMarkup` (marcado de encabezamiento impar)
- `evenHeaderMarkup` (marcado de encabezamiento par)
- `oddFooterMarkup` (marcado de pie de página impar)
- `evenFooterMarkup` (marcado de pie de página par)

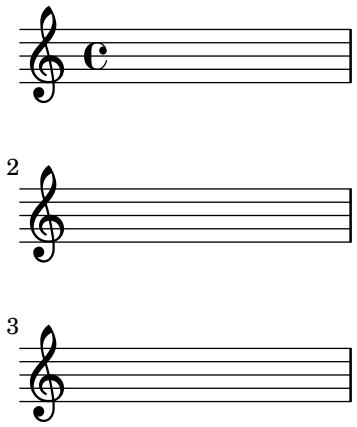
La instrucción de marcado `\if` se puede utilizar para añadir elementos de forma condicional al texto de encabezamiento y pie de página definido dentro del bloque `\paper`, usando la sintaxis siguiente dentro de `\markup`:

```
\if condición argumento
```

La *condición* se comprueba cada vez que se interpreta el elemento de marcado. El marcado resultante es *argumento* si se cumple la condición, y vacío si no se cumple. Entre las condiciones más comunes encontramos pruebas de números de página (si es la primera, la última, una página específica, ...). Para hacer la comprobación de si la condición es falsa, sustituya `\if` por `\unless`.

El ejemplo siguiente centra los números de página en la parte baja de las páginas. En primer lugar, los ajustes predeterminados para `oddHeaderMarkup` y `evenHeaderMarkup` se eliminan definiéndolos como un marcado *nulo*. Después de esto, se redefine `oddFooterMarkup` con el número de página centrado. Finalmente, `evenFooterMarkup` recibe la misma disposición definiéndola como `\oddFooterMarkup`:

```
\book {
  \paper {
    print-page-number = ##t
    print-first-page-number = ##t
    oddHeaderMarkup = ##f
    evenHeaderMarkup = ##f
    oddFooterMarkup = \markup {
      \fill-line {
        \if \should-print-page-number
          \fromproperty #'page:page-number-string
      }
    }
    evenFooterMarkup = \oddFooterMarkup
  }
  \score {
    \new Staff { s1 \break s1 \break s1 }
  }
}
```



1

Aquí aparece una lista de todos los procedimientos predefinidos disponibles para su uso con `\if` y `\unless`.

Sintaxis	**Condición que se comprueba**
`\on-first-page`	¿Es la primera página del libro?
`\on-last-page`	¿Es la última página del libro?
`\on-first-page-of-part`	¿Es la primera página en la parte de libro?
`\on-last-page-of-part`	¿Es la última página en la parte de libro?
`\on-page` *n*	¿Está en la página número *n*?
`\single-page`	¿El libro entero cabe en una sola página?
`\should-print-page-numbers-global`	¿Imprimir números de página en el libro?[1]
`\should-print-page-number`	¿Imprimir el número de la página actual?
`\should-print-all-headers`	¿Es `print-all-headers` verdadero?

Véase también

Referencia de la notación: Sección 21.1.1 [Explicación de los títulos], página 598, Sección 21.1.2 [Presentación predeterminada de los títulos de partes de libro y partitura], página 601, Sección A.1.5 [Conditional markup], página 868.

Archivos instalados: `../ly/titling-init.ly`.

21.3 Crear metadatos en los archivos de salida

Además de mostrarse en la salida impresa, las variables de encabezamiento de `\header` se usan para establecer los metadatos para los archivos de salida. Por ejemplo, con los archivos PDF, estos metadatos pueden mostrarse por parte de los lectores de documentos como `propiedades` del PDF. Para cada tipo de archivo de salida, solamente se consultan las definiciones de `\header` de los bloques que definen archivos separados de ese tipo, y los bloques más altos en la jerarquía de bloques. Por tanto, para los archivos PDF, solo afectan a los metadatos de PDF de cada documento las definiciones de `\header` de los niveles de `\book` y de nivel superior, mientras que para archivos MIDI, se usan todas las cabeceras que están por encima o al nivel de `\score`.

Por ejemplo, si se establece la propiedad `title` del bloque `header` como 'Sinfonía núm. 1', el documento PDF llevará también este título, y lo usará como el nombre de la secuencia del archivo MIDI.

[1] `\should-print-page-numbers-global` puede ser diferente de `\should-print-page-number` para la primera página dentro del libro, dependiento del valor de `print-first-page-number` del bloque `\paper`.

```
\header {
  title = "Sinfonía núm. 1"
}
```

Si quiere fijar el título de la salida impresa con un texto pero quiere que la propiedad de título del PDF tenga un texto diferente, puede usar `pdftitle` como se muestra a continuación.

```
\header {
  title = "Sinfonía núm. 1"
  pdftitle = "Sinfonía núm. 1 de Beethoven"
}
```

Todas las variables `title`, `subject`, `keywords`, `subtitle`, `composer`, `arranger`, `poet`, `author` y `copyright` establecen propiedades del PDF y pueden hacerse preceder de 'pdf' para fijar una propiedad del archivo PDF a un valor distinto del de la salida impresa.

La propiedad de PDF `Creator` se fija automáticamente a 'LilyPond' más la versión actual de LilyPond, y las variables `CreationDate` y `ModDate` se fijan a la fecha y hora actuales. Se puede sobreescribir `ModDate` mediante el establecimiento de un valor para la variable de cabecera `moddate` (o `pdfmoddate`) a una cadena de fecha de PDF válida.

La variable `title` establece también el nombre de la secuencia para el archivo MIDI. Se puede usar la variable `miditítle` para fijar el nombre de la secuencia independientemente del valor usado para la salida impresa.

21.4 Crear notas al pie

Las notas al pie se pueden usar en muchas situaciones diferentes. En todos los casos, se sitúa una 'footnote mark' como referencia dentro del texto o la música, y el 'texto de la nota al pie' correspondiente aparece en la parte inferior de la misma página, separado de la música por una línea horizontal. La apariencia visual de este spearadoe se puede modificar estableciendo la variable `footnote-separator-markup` del bloque `paper`, véase Sección 26.6.4 [Variables de \paper relativas a los encabezamientos y marcados], página 684.

Las notas al pie dentro de las expresiones musicales y las notas al pie dentro de elementos de texto autocontenidos fuera de las expresiones musicales, se crean de formas diferentes.

21.4.1 Notas al pie dentro de expresiones musicales

Panorámica de las notas al pie

Las notas al pie dentro de expresiones musicales se agrupan en dos categorías:

Notas al pie basadas en eventos
: Se adjuntan a un evento en particular. Son ejemplos de tales eventos las notas sueltas, las articulaciones (como indicaciones de digitación, acentos, indicaciones dinámicas) y los post-eventos (como las ligaduras de expresión y las barras de corchea manuales). La forma general de las notas al pie basadas en eventos es como sigue:

```
[dirección] \footnote [marca] desplazamiento texto-de-la-nota música
```

Notas al pie basadas en tiempo
: Se ligan a un punto temporal determinado, dentro de un contexto musical. Algunas instrucciones como `\time` y `\clef` en realidad no usan eventos para crear objetos como la indicación de compás y la clave. Un acorde tampoco crea un evento por sí mismo: su plica o corchete se crea al final de un paso de tiempo (nominalmente, a través de uno de los eventos de nota que contiene). Una nota al pie basada en tiempo permite anotar tales objetos de presentación sin referirse a ningún evento.

Una nota al pie basada en tiempo permite que tales objetos de presentación se puedan anotar sin hacer referencia a un evento. La forma general para las notas al pie basadas en tiempo es:

```
\footnote [marca] desplazamiento texto-de-la-nota [Contexto].Nombre-del-Grob
```

Los elementos para las dos formas son como sigue:

dirección Si (y solo si) el `\footnote` se está aplicando a un post-evento o articulación, debe ir precedida de un indicador de dirección (‘`-`’, ‘`_`’, ‘`^`’) con el objeto de adjuntar la *música* (con una marca de nota al pie) a la nota o silencio precedente.

marca es un elemento de marcado o cadena de caracteres que especifica la marca de nota al pie que se usa para marcar tanto el punto de referencia como la propia nota al pie en la parte inferior de la página. Se puede omitir (o, de forma equivalente, sustituirse por `\default`) en cuyo caso se genera un número secuencialmente de forma automática. Por omisión, tales secuencias numéricas se reinician en cada página que contiene una nota al pie. Las notas al pie se pueden numerar en orden creciente a través de los saltos de página estableciendo la variable `reset-footnotes-on-new-page` a `#f`, véase Sección 26.6.4 [Variables de `\paper` relativas a los encabezamientos y marcados], página 684.

desplazamiento
es una pareja de números tal como ‘`#'(2 . 1)`’ que especifica los desplazamientos en X y en Y en unidades de espacios de pentagrama a partir del límite del objeto en que se desea situar la marca. Los valores de desplazamiento positivos se toman a partir del borde superior derecho, los valores negativos a partir del borde inferior izquierdo y cero implica que la marca se centra sobre el borde.

Contexto es el contexto en que se crea el grob que recibe la nota al pie. Se puede omitir si el grob está en un contexto del nivel inferior, p. ej. un contexto `Voice`.

Nombre-del-Grob
especifica un tipo de grob que marcar (como ‘`Flag`’ para el corchete). Si se da, la nota al pie no se adjunta a una expresión musical en particular, sino a todos los grobs del tipo especificado que ocurren en dicho momento de tiempo musical.

texto-de-la-nota
es el elemento de marcado o cadena de caracteres que especifica el texto de la nota al pie que utilizar en la parte baja de la página.

música es el evento musical, post-evento o articulación que se está anotando.

Notas al pie basadas en eventos

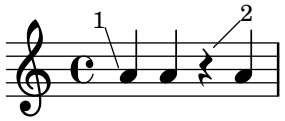
Una nota al pie se adjunta a un objeto de presentación causado directamente por el evento que corresponde a *música*, con la sintaxis siguiente:

```
\footnote [marca] desplazamiento texto-de-la-nota música
```

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en evento"
  \markup \null
  \relative c'' {
    \footnote #'(-1 . 3) "Una nota." a4
    a4
    \footnote #'(2 . 2) "Un silencio." r4
    a4
```

```
  }
}
```

notas al pie basadas en evento

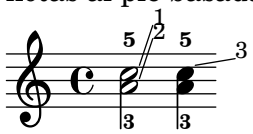


[1]Una nota.
[2]Un silencio.

Si un acorde está marcado con una nota al pie basada en evento, cada nota del acorde recibe una nota al pie idéntica pero independiente, algo que normalmente no es lo que se desea hacer. Sin embargo, es posible crear notas al pie para las notas individuales dentro de un acorde.

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en evento"
  \markup \null
  \relative c'' {
    \footnote #'(1 . 3) "Un acorde." <a-3 c-5>2
    <a-3 \footnote #'(3 . 0.5) "Una nota de un acorde." c-5>4
  }
}
```

notas al pie basadas en evento



[1]Un acorde.
[2]Un acorde.
[3]Una nota de un acorde.

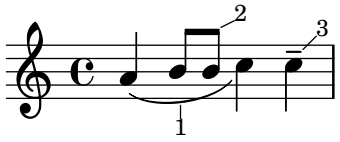
Si se desea que la nota al pie se adjunte a un post-evento o articulación, la instrucción `\footnote` *debe* ir precedida por un indicador de dirección, (‘-’, ‘_’, ‘^’), y seguida por el post-evento o articulación que se desea anotar como argumento *música*. En esta forma, la `\footnote` se puede considerar como una simple copia de su último argumento con una marca de nota al pie adjunta. La sintaxis es:

dirección \footnote [*marca*] *desplazamiento texto-de-la-nota música*

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en evento"
  \markup \null
  \relative {
    a'4_\footnote #'(0 . -1) "Ligadura forzada hacia abajo" (
    b8^\footnote #'(1 . 0.5) "Barra manual forzada hacia arriba" [
    b8 ]
    c4 )
```

```
    c-\footnote #'(1 . 1) "Tenuto." --
  }
}
```

notas al pie basadas en evento



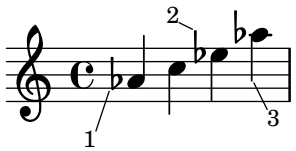
[1]Ligadura forzada hacia abajo
[2]Barra manual forzada hacia arriba
[3]Tenuto.

Notas al pie basadas en tiempo

Si el objeto de presentación que recibe la nota al pie está causado *indirectamente* por un evento (como un objeto `Accidental`, alteración, o `Stem`, plica, causados por un evento `NoteHead`, cabeza de nota), el nombre del grob *GrobName* del objeto de presentación es necesario después del texto de la nota al pie, en lugar de *música*:

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en tiempo"
  \markup \null
  \relative c'' {
    \footnote #'(-1 . -3) "Un bemol." Accidental
    aes4 c
    \footnote #'(-1 . 0.5) "Otro bemol." Accidental
    ees
    \footnote #'(1 . -2) "Una plica." Stem
    aes
  }
}
```

notas al pie basadas en tiempo



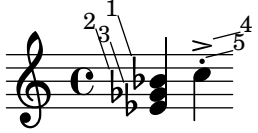
[1]Un bemol.
[2]Otro bemol.
[3]Una plica.

Sin embargo, observe que cuando se especifica un *GrobName* (el nombre de un grob), se adjunta una nota al pie a todos los grobs de ese tipo dentro del instante de tiempo actual:

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en tiempo"
  \markup \null
  \markup \null
  \relative c' {
```

```
    \footnote #'(-1 . 3) "Un bemol." Accidental
    <ees ges bes>4
    \footnote #'(2 . 0.5) "Articulación." Script
    c'->-.
  }
}
```

notas al pie basadas en tiempo

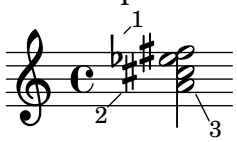


[1]Un bemol.
[2]Un bemol.
[3]Un bemol.
[4]Articulación.
[5]Articulación.

Una nota dentro de un acorde puede recibir una nota al pie individual, basada en evento. El único grob causado directamente por una nota de un acorde es 'NoteHead', por lo que la instrucción de nota al pie basada en evento *solamente* es apta para añadir una nota al pie a la 'NoteHead' dentro de un acorde. Todos los demás grobs de notas de acorde están causados indirectamente. La propia instrucción \footnote no ofrece ninguna sintaxis para especificar al mismo tiempo un tipo concreto de grob *y también* un evento concreto al que adjuntar la nota. Sin embargo, se puede usar una instrucción \footnote basada en tiempo para especificar el tipo de grob, y luego añadir la instrucción \single como prefijo, con el objeto de que se aplique solamente al evento siguiente:

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en tiempo"
  \markup \null
  \relative c'' {
    < \footnote #'(1 . -2) "Un La." a
      \single \footnote #'(-1 . -1) "Un sostenido." Accidental
      cis
      \single \footnote #'(0.5 . 0.5) "Un bemol." Accidental
      ees fis
    >2
  }
}
```

notas al pie basadas en tiempo



[1]Un bemol.
[2]Un sostenido.
[3]Un La.

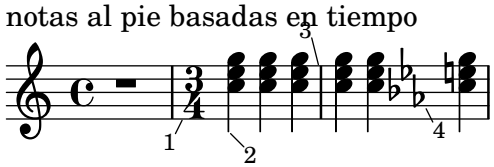
Nota: Cuando las notas al pie se adjuntan a varios elementos musicales que están dentro del mismo momento musical, como ocurre en el ejemplo anterior, las notas al pie se numeran desde los elementos más altos a los más bajos tal y como aparecen en la salida impresa, no en el orden en que se escriben en el flujo de entrada.

Los objetos de presentación como las claves y las armaduras de cambio de tonalidad están, más bien, causadas por propiedades que cambian, y no por eventos. Otros, como las líneas divisorias y los números de compás, son una consecuencia directa del paso del tiempo. Por esta razón, las notas al pie sobre tales objetos tienen que basarse en su tiempo musical. Las notas al pie basadas en tiempo so preferibles también cuando se marcan cosas como las plicas y las barras de corchea en *acordes*: dado que estas funcionalidades de cada acorde individual están asignadas normalmente a *un solo* evento dentro del acorde, sería imprudente descansar en una elección particular.

En las notas al pie basadas en tiempo, siempre se debe especificar explícitamente el objeto de presentación en cuestión, y si el grob se crea dentro de un contexto distinto al de nivel más bajo, debe especificarse el contexto apropiado.

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie basadas en tiempo"
  \relative c'' {
    r1 |
    \footnote #'(-0.5 . -1) "Cambio de compás." Staff.TimeSignature
    \time 3/4
    \footnote #'(1 . -1) "Plica de acorde." Stem
    <c e g>4 q q
    \footnote #'(-0.5 . 2) "Línea divisoria." Staff.BarLine
    q q
    \footnote #'(0.5 . -1) "Cambio de armadura." Staff.KeySignature
    \key c \minor
    q
  }
}
```

notas al pie basadas en tiempo



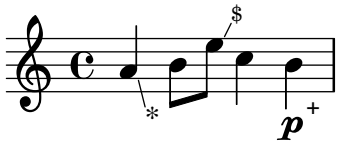
[1]Cambio de compás.
[2]Plica de acorde.
[3]Línea divisoria.
[4]Cambio de armadura.

Se pueden usar marcas personalizadas como alternativas a las marcas numéricas, y se puede suprimir la línea de anotación que une el objeto marcado con su marca:

```
\book {
  \header { tagline = ##f }
  \markup "notas al pie con marcas personalizadas"
  \markup \null
```

```
  \relative c' {
    \footnote "*" #'(0.5 . -2) \markup { \italic "* La primera nota" }
    a'4 b8
    \footnote \markup { \super "$" } #'(0.5 . 1)
      \markup { \super "$" \italic " La segunda nota." } e
    c4
    \once \override Score.Footnote.annotation-line = ##f
    b-\footnote \markup \tiny "+" #'(0.1 . 0.1)
      \markup { \super "+" \italic " Editorial." } \p
  }
}
```

notas al pie con marcas personalizadas



* *La primera nota*
$ *La segunda nota.*
\+ *Editorial.*

Se muestran más ejemplos de marcas personalizadas en Sección 21.4.2 [Notas al pie en texto independiente], página 618.

21.4.2 Notas al pie en texto independiente

Se utilizan dentro de un elemento de marcado fuera de cualquier expresión musical. No se traza una línea hasta el punto de referencia: la marca simplemente sigue al elemento de marcado que se referencia. Se pueden insertar las marcas automáticamente, en cuyo caso son numéricas. De forma alternativa, se pueden aportar manualmente marcas personalizadas.

Las notas al pie para texto independiente con marcas automáticas y personalizadas se crean de distintas formas.

Notas al pie en texto independiente con marcas automáticas

La sintaxis de una nota al pie sobre texto independiente con marcas automáticas es

```
\markup { ... \auto-footnote texto nota ... }
```

Los elementos son:

texto es el elemento de marcado o cadena de caracteres que se ha de marcar,

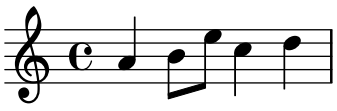
nota es el elemento de marcado o cadena que especifica el texto de la nota al pie a utilizar en la parte baja de la página.

Por ejemplo:

```
\book {
  \header { tagline = ##f }
  \markup {
    "Una sencilla"
    \auto-footnote "melodía" \italic " Hecha por mí."
    "aparece a continuación.  Es una composición"
    \auto-footnote "reciente" \italic " Agosto de 2012."
  }
```

```
  \relative {
    a'4 b8 e c4 d
  }
}
```

Una sencilla melodía[1] aparece a continuación. Es una composición reciente[2]



[1] *Hecha por mí.*
[2] *Agosto de 2012.*

Notas al pie sobre texto independiente con marcas personalizadas

La sintaxis de una nota al pie sobre texto independiente con marcas personalizadas es

```
\markup { ... \footnote marca nota ... }
```

Los elementos son:

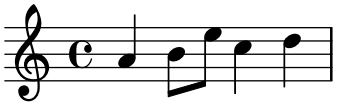
marca es un elemento de marcado o cadena de caracteres que especifica la marca que se usa para señalar el punto de referencia. Observe que esta marca *no* se inserta automáticamente antes de la propia nota al pie.

nota es el elemento de marcado o cadena de caracteres que especifica el texto de la nota que aparece en la parte baja de la página, precedida por la *marca* especificada.

Se puede usar como marca un carácter fácil de teclear, como '*' ó '+', según se muestra en Sección 21.4.1 [Notas al pie dentro de expresiones musicales], página 612. De forma alternativa, se pueden usar alias en ASCII (véase Sección 22.4.3 [Alias de ASCII], página 648):

```
\book {
  \paper { #(include-special-characters) }
  \header { tagline = ##f }
  \markup {
    "Una sencilla melodía"
    \footnote "*" \italic "* Hecha por mí."
    "aparece a continuación.  Es una composición reciente"
    \footnote \super &dagger; \concat {
      \super &dagger; \italic " Agosto de 2012."
    }
  }
  \relative {
    a'4 b8 e c4 d
  }
}
```

Una sencilla melodía * aparece a continuación. Es una composición reciente †

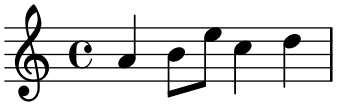


* *Hecha por mí.*
† *Agosto de 2012.*

También pueden utilizarse códigos de caracteres Unicode para especificar las marcas (véase Sección 22.4.2 [Unicode], página 647):

```
\book {
  \header { tagline = ##f }
  \markup {
    "Una sencilla melodía"
    \footnote \super \char##x00a7 \concat {
      \super \char##x00a7 \italic " Hecha por mí."
    }
    "aparece a continuación.  Es una composición reciente"
    \footnote \super \char##x00b6 \concat {
      \super \char##x00b6 \italic " Agosto de 2012."
    }
  }
  \relative {
    a'4 b8 e c4 d
  }
}
```

Una sencilla melodía § aparece a continuación. Es una composición reciente ¶



§ *Hecha por mí.*
¶ *Agosto de 2012.*

Véase también

Manual de aprendizaje: Sección “Objetos e interfaces” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 22.4.3 [Alias de ASCII], página 648, Sección 7.2.2 [Globos de ayuda], página 303, Sección B.12 [Lista de caracteres especiales], página 932, Sección 8.1.5 [Marcas de texto], página 317, Sección 8.1.2 [Inscripciones textuales], página 314, Sección 22.4.2 [Unicode], página 647.

Referencia de funcionamiento interno: Sección "FootnoteEvent" en *Referencia de Funcionamiento Interno*, Sección "Footnote" en *Referencia de Funcionamiento Interno*, Sección "Footnote_engraver" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Varias notas al pie en la misma página solo pueden ir apiladas unas sobre otras; no se pueden imprimir en la misma línea.

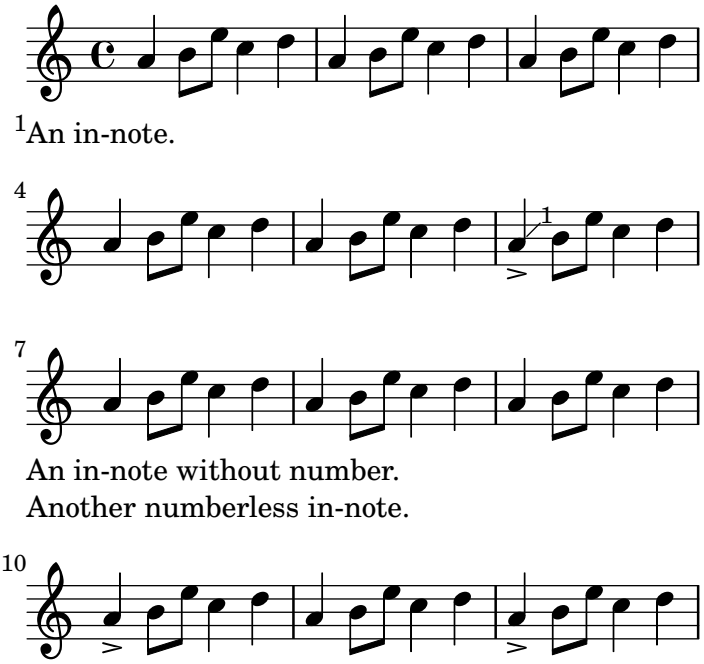
Las notas al pie pueden colisionar con las pautas, objetos `\markup`, otras marcas de notas al pie y líneas de anotación.

21.5 Creación de notas interiores

Las notas interiores funcionan como las notas al pie en el sentido de que sirven para anotar la música, pero se diferencian en que se tipografían encima o debajo del sistema a que pertenece el objeto gráfico que se está anotando.

Para crear una nota interior, fije la propiedad `footnote` del objeto gráfico `Footnote` al valor `#f`. La distancia entre dos notas interiores se puede controlar con la variable `in-note-padding` del bloque `paper`, y la distancia entre la nota interior y su sistema asociado mediante `in-note-system-padding`. Si quiere que las notas interiores estén posicionadas debajo de su sistema asociado, establezca la variable `in-note-direction` del bloque `paper` al valor `DOWN`.

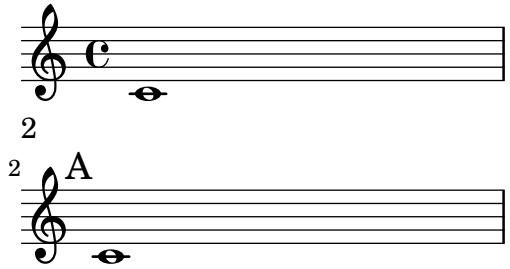
```
music = { a4 b8 e c4 d }

\book {
  \relative c'' {
    \override Score.Footnote.footnote = ##f

    \repeat unfold 5 \music
    \footnote #'(1 . 1) "An in-note." NoteHead
    <>-> \repeat unfold 4 \music
    \footnote "" #'(0 . 0) "An in-note without number." NoteHead
    <>-> \repeat unfold 2 \music
    \footnote "" #'(0 . 0) "Another numberless in-note." NoteHead
    <>-> \music
  }

  \paper {
    in-note-system-padding = 5
    in-note-padding = 2
    tagline = ##f
  }
}
```



21.6 Referencia a números de página

Un lugar determinado de una partitura puede señalizarse usando la instrucción `\label` (etiqueta), ya sea en el nivel superior o dentro de la música. Podemos referirnos a esta etiqueta más tarde dentro de un elemento de marcado, para obtener el número de la página en la que se sitúa el punto señalizado, usando la instrucción de marcado `\page-ref`.

```
\header { tagline = ##f }
\book {
  \label #'firstScore
  \score {
    {
      c'1
      \pageBreak \mark A \label #'markA
      c'1
    }
  }
  \markup { La primera partitura comienza en la página \page-ref #'firstScore "0" "?" }
  \markup { La marca A está en la página \page-ref #'markA "0" "?" }
}
```



La primera partitura comienza en la página 1
La marca A está en la página 2

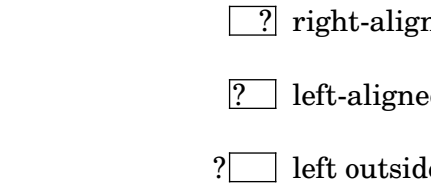
La instrucción de marcado \page-ref toma tres argumentos:

1. la etiqueta, un símbolo de Scheme, p.ej. #'firstScore;
2. un elemento de marcado que se usará como medidor para estimar las dimensiones del marcado;
3. un elemento de marcado que se utilizará en sustitución del número de página si la etiqueta es desconocida.

El motivo de que se necesite un medidor es que en el momento en que se están interpretando los marcados, los saltos de página aún no se han producido y por tanto los números de página no se conocen todavía. Para sortear este inconveniente, la interpretación real del marcado se retrasa hasta un momento posterior; sin embargo, las dimensiones del marcado se tienen que conocer de antemano, así que se usa el medidor para decidir estas dimensiones. Si el libro tiene entre 10 y 99 páginas, el medidor puede ser "00", es decir, un número de dos dígitos.

Si el tamaño del texto final es diferente del medidor, puede ser útil configurar la alineación horizontal relativa al medidor con la propiedad x-align. Por omisión está alineado a la derecha.

```
\markup {
  \box
    \page-ref #'foo "???" "?" " right-aligned (default)"
}
\markup {
  \box
    \override #`(x-align . ,LEFT)
    \page-ref #'foo "???" "?" " left-aligned"
}
\markup {
  \box
    \override #'(x-align . -2.5)
    \page-ref #'foo "???" "?" " left outside"
}
```



En el ejemplo, el medidor '???' es más ancho que el texto de sustitución '?'. La sustitución se usa porque la etiqueta #'foo no existe. La propiedad x-align se puede establecer con números cualesquiera. También se pueden usar los símbolos predefinidos LEFT, CENTER o RIGHT para fijar la alineación a la izquierda, centrada o a la derecha en relación al medidor.

Instrucciones predefinidas

\label, \page-ref.

21.7 Tabla de contenidos

Se puede insertar un índice general o tabla de contenidos utilizando la instrucción \markuplist \table-of-contents. Los elementos que deben aparecer en la tabla de contenidos se introducen con la instrucción \tocItem, que se puede usar en el nivel más alto de la jerarquía del código, o dentro de una expresión musical.

```
\markuplist \table-of-contents
\pageBreak
```

```
\tocItem \markup "Primera partitura"
\score {
  {
    c'4  % ...
    \tocItem \markup "Un punto concreto dentro de la primera partitura"
    d'4  % ...
  }
}

\tocItem \markup "Segunda partitura"
\score {
  {
    e'4 % ...
    \tocItem actI \markup "Act I"
    f'4 % ...
    \tocItem actI.sceneI \markup "Scene 1"
    g'4 % ...
    \tocItem actI.sceneI.recitativo \markup "Recit."
    a'4 % ...
  }
}
```

De manera opcional se puede asociar una etiqueta con un elemento en particular o con una lista jerárquica de etiquetas existentes, terminando con la etiqueta de ese elemento. Esta última posibilidad nos permite marcar el elemento como un 'hijo' de los elementos etiquetados precedentes, haciendo así aparente la estructura de la partitura en el índice general.

Los elementos de marcado que se usan para dar formato al índice general o tabla de contenidos se encuentran definidos dentro del bloque `\paper`. Hay tres elementos de marcado 'predefinidos' disponibles:

- `tocTitleMarkup`

  Utilizado para dar formato al título del índice general.

  ```
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Índice general" \null }
    \null
  }
  ```

- `tocItemMarkup`

  Usado para dar formato a los elementos del índice general.

  ```
  tocItemMarkup = \markup \fill-line {
    \fromproperty #'toc:text \fromproperty #'toc:page
  }
  ```

- `tocFormatMarkup`

  Cómo se formatean las entradas del nivel superior de la tabla (si existen varios niveles jerárquicos). Esto es un procedimiento en realidad, como se explica en Sección "Construcción de elementos de marcado en Scheme" en *Extender*.

  ```
  tocFormatMarkup = #make-bold-markup
  ```

- `tocIndentMarkup`

  Usado para definir cómo la jerarquía del esquema general se hace aparente. Este elemento de marcado se imprime ninguna, una o varias veces dependiendo del nivel de cada entrada.

  ```
  tocIndentMarkup = \markup \hspace #4
  ```

Cualquiera de estas variables puede modificarse.

He aquí un ejemplo que traduce el título del índice general al francés:

```
\paper {
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Table des matières" \null }
    \hspace #1
  }
```

A continuación aparece un ejemplo que modifica el tamaño de fuente tipográfica de los elementos del índice general:

```
tocItemMarkup = \markup \large \fill-line {
  \fromproperty #'toc:text \fromproperty #'toc:page
}
```

Observe la forma en que nos referimos al texto y al número de página del elemento de índice, dentro de la definición `tocItemMarkup`.

La instrucción `\tocItemWithDotsMarkup` se puede incluir dentro de `tocItemMarkup` para llenar la línea con puntos entre un elemento del índice general y su número de página correspondiente:

```
\header { tagline = ##f }
\paper {
  tocItemMarkup = \tocItemWithDotsMarkup
}

\book {
  \markuplist \table-of-contents
  \tocItem \markup { Allegro }
  \tocItem \markup { Largo }
  \markup \null
}
```

Table of Contents

Allegro 1
Largo 1

Además del mecanismo incorporado de construcción del índice general, las instrucciones predefinidas pueden definirse también para que construyan un índice general más personalizado con diferentes elementos de marcado. En el ejemplo siguiente se define un estilo nuevo para introducir los actos y escenas de una ópera en el índice general:

Una nueva variable de marcado (llamada `tocActMarkup`) se define en el bloque `\paper`:

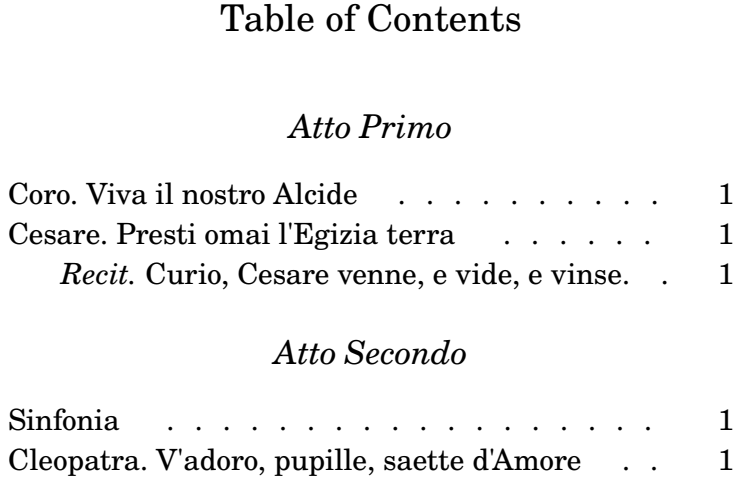
```
\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}
```

A continuación se crea una función musical personalizada (`tocAct`), que usa la nueva definición de marcado `tocActMarkup`, y permite especificar una etiqueta para cada acto.

```
tocAct =
```

```
#(define-music-function (label text) (symbol? markup?)
   (add-toc-item! 'tocActMarkup text label))
```

Usando estas definiciones personalizadas y modificando algunas de las definiciones existentes, el archivo fuente se podría escribir entonces de la manera siguiente:



El ejemplo anterior muestra también cómo usar la instrucción de marcado `\fill-with-pattern` dentro del contexto de un índice general o tabla de contenidos.

Véase también

Archivos de inicio: `ly/toc-init.ly`.

Instrucciones predefinidas

`\table-of-contents`, `\tocItem`, `tocItemMarkup`, `tocTitleMarkup`, `tocFormatMarkup`, `tocIndentMarkup`.

22 Trabajar sobre los archivos de entrada

22.1 Inclusión de archivos de LilyPond

Un proyecto grande se puede dividir en varios archivos. Para referirse a otro archivo, utilice

```
\include "otroarchivo.ly"
```

La línea `\include "otroarchivo.ly"` equivale a pegar todo el contenido de `otroarchivo.ly` en el archivo actual en el lugar en que aparece el `\include`. Por ejemplo, en un proyecto grande podríamos querer archivos distintos para cada parte instrumental y crear un archivo de "partitura completa" que reúne los archivos individuales de los instrumentos. Normalmente el archivo incluido define un cierto número de variables que a partir de entonces quedan disponibles para poderlas utilizar en el archivo de la partitura completa. En los archivos incluidos se pueden marcar las secciones etiquetadas para ayudar a hacerlas utilizables en distintos lugares de la partitura, véase Sección 22.2 [Distintas ediciones a partir de una misma fuente], página 628.

Los archivos que están en el directorio de trabajo actual se pueden referenciar simplemente especificando el nombre después de la instrucción `\include`. Los archivos en otros lugares se pueden incluir proporcionando una referencia de ruta completa o una ruta relativa (pero utilice la barra inclinada normal del UNIX, /, no la barra invertida de DOS/Windows, \, como separador de directorio). Por ejemplo, si `material.ly` está situado un directorio por encima del directorio de trabajo actual, utilice

```
\include "../material.ly"
```

o si los archivos de las partes orquestales incluidas están todos situados en un subdirectorio llamado `partes` dentro del directorio actual, use

```
\include "partes/VI.ly"
\include "partes/VII.ly"
... etc.
```

Los archivos de inclusión también pueden contener enunciados `\include` a su vez. Estos enunciados `\include` de segundo nivel se interpretan de manera relativa a la ruta del archivo que contiene dicha instrucción, lo que es conveniente para varios archivos ubicados dentro del mismo subdirectorio. Por ejemplo, una biblioteca general, bibA, puede a su vez usar sub-archivos que son incluidos por el archivo principal de dicha biblioteca, así:

```
bibA/
  bibA.ly
  A1.ly
  A2.ly
  ...
```

entonces el archivo de entrada, `bibA.ly`, contiene

```
\include "A1.ly"
\include "A2.ly"
...
```

Cualquier archivo `.ly` puede entonces incluir la biblioteca completa, simplemente con

```
\include "~/bibA/bibA.ly"
```

Sin embargo, este comportamiento se puede modificar de forma global pasando la opción `-drelative-includes=#f` en la línea de órdenes (o añadiendo `#(ly:set-option 'relative-includes #f)` al principio del archivo principal de entrada). En tal caso, cada archivo se incuirá relativamente a la ubicación del archivo principal, sin importar dónde está situado su instrucción `\include`. Estructuras de archivo complejas que requieren realizar la inclusión tanto de archivos relativos al directorio principal como de archivos relativos a algún

otro directorio, pueden incluso configurarse estableciendo `relative-includes` a los valores `#f` o `#t` en los lugares adecuados dentro de dichos archivos.

También se pueden incluir archivos de un directorio que está en una ruta de búsqueda especificada como opción al invocar a LilyPond desde la línea de órdenes. Los archivos incluidos se especifican usando solamente su nombre de archivo. Por ejemplo, para compilar mediante este método un archivo `principal.ly` que incluye archivos situados en un subdirectorio llamado `partes`, cambie al directorio que contiene `principal.ly` e introduzca

```
lilypond --include=partes principal.ly
```

y en `principal.ly` escriba

```
\include "VI.ly"
\include "VII.ly"
... etc.
```

Los archivos que se incluyen en muchas partituras se pueden poner en el directorio de LilyPond `../ly` (la localización de este directorio depende de la instalación: véase Sección "Otras fuentes de información" en *Manual de Aprendizaje*). Estos archivos se pueden incluir sencillamente por su nombre en un enunciado `\include`. De esta forma se incluyen los archivos dependientes del idioma como `espanol.ly`.

LilyPond incluye de forma predeterminada ciertos archivos cuando se inicia el programa. Estas inclusiones no son visibles para el usuario, pero los archivos se pueden identificar ejecutando `lilypond --verbose` desde la línea de órdenes. Esto presentará una lista de rutas y archivos que utiliza LilyPond, junto a muchas otras indicaciones. De forma alternativa, pueden verse explicaciones sobre los archivos más importantes de este conjunto en Sección "Otras fuentes de información" en *Manual de Aprendizaje*. Estos archivos se pueden editar, pero los cambios realizados sobre ellos se perderán al instalar una nueva versión de LilyPond.

Pueden verse ejemplos sencillos sobre el uso de `\include` en Sección "Partituras y particellas" en *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Otras fuentes de información" en *Manual de Aprendizaje*, Sección "Partituras y particellas" en *Manual de Aprendizaje*.

Advertencias y problemas conocidos

Si un archivo incluido recibe un nombre igual al de uno de los archivos de instalación de LilyPond, el archivo que tiene prioridad es el de los archivos de instalación de LilyPond.

22.2 Distintas ediciones a partir de una misma fuente

Se pueden usar varios métodos para generar la producción de distintas versiones de una partitura a partir de la misma fuente musical. Las variables son quizá el más útil para combinar secciones extensas de música y/o anotaciones. Las etiquetas son más útiles para seleccionar una sección de entre varias secciones de música alternativas, más cortas, y se pueden usar también para dividir piezas de música en distintos trozos.

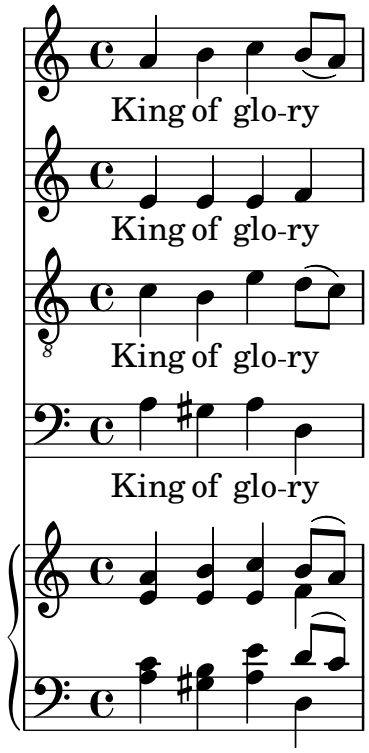
Cualquiera que sea el método utilizado, la separación de la notación de la estructura de la partitura hará que sea más fácil cambiar la estructura dejando intacta la notación.

22.2.1 Uso de las variables

Si las secciones de la música se definen dentro de variables, se pueden reutilizar en distintas partes de la partitura, véase Sección "Organizar las piezas mediante variables" en *Manual de Aprendizaje*. Por ejemplo, una partitura vocal *a cappella* con frecuencia incluye una reducción de piano de las partes para ensayar que es idéntica a la música vocal, por lo que ésta solo se tiene

que escribir una vez. Se puede combinar sobre un solo pentagrama la música de dos variables, véase Sección 5.2.5 [Combinación automática de las partes], página 235. He aquí un ejemplo:

```
sopranoMusic = \relative { a'4 b c b8( a) }
altoMusic = \relative { e'4 e e f }
tenorMusic = \relative { c'4 b e d8( c) }
bassMusic = \relative { a4 gis a d, }
allLyrics = \lyricmode { King of glo -- ry }
<<
  \new Staff = "Soprano" \sopranoMusic
  \new Lyrics \allLyrics
  \new Staff = "Alto" \altoMusic
  \new Lyrics \allLyrics
  \new Staff = "Tenor" {
    \clef "treble_8"
    \tenorMusic
  }
  \new Lyrics \allLyrics
  \new Staff = "Bass" {
    \clef "bass"
    \bassMusic
  }
  \new Lyrics \allLyrics
  \new PianoStaff <<
    \new Staff = "RH" {
      \partCombine \sopranoMusic \altoMusic
    }
    \new Staff = "LH" {
      \clef "bass"
      \partCombine \tenorMusic \bassMusic
    }
  >>
>>
```



Se pueden producir partituras distintas que presenten solo las partes vocales o solo la parte de piano, cambiando solamente los enunciados estructurales, sin modificar la notación musical.

Para partituras extensas, las definiciones de variable se pueden poner en archivos separados que se incluyen más tarde, véase Sección 22.1 [Inclusión de archivos de LilyPond], página 627.

22.2.2 Uso de etiquetas

La instrucción `\tag` marca una expresión musical con un nombre.

```
\tag #'fulano { ... }
```

Las expresiones etiquetadas de esta manera se pueden tratar de una forma muy conveniente como un todo.

Mantenimiento y eliminación de música etiquetada

La instrucción `\keepWithTag` selecciona música etiquetada. En cambio, `\removeWithTag` excluye la música etiquetada.

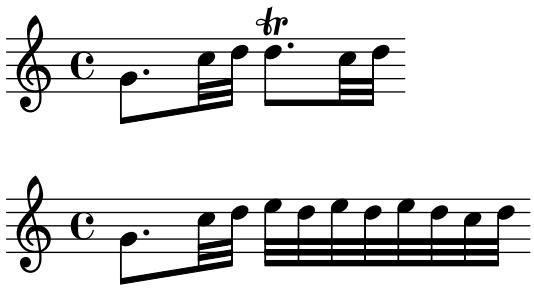
Filtro	**Resultado**
Música etiquetada precedida de `\keepWithTag #'`*nombre* o de `\keepWithTag #'(`*nombre1 nombre2...*`)` `\keepWithTag #'(`*nombre1 nombre2...*`)`	Se incluye la música no etiquetada y la música etiquetada con el nombre o nombres de etiqueta dados; se excluye la música etiquetada con cualquier otro nombre de etiqueta.
Música etiquetada precedida de `\removeWithTag #'`*nombre* o de `\removeWithTag #'(`*nombre1 nombre2...*`)`	Se incluye la música no etiquetada y la música que no está etiquetada con ninguno de los nombres de etiqueta dados; se excluye la música etiquetada con los nombres de etiqueta dados.

Música etiquetada no precedida de `\keepWithTag` ni de `\removeWithTag`	Se incluye toda la música etiquetada y no etiquetada.

Los argumentos de las instrucciones `\tag`, `\keepWithTag` y `\removeWithTag` deben ser un símbolo o una lista de símbolos (como `#'score` o `#'(violinI violinII)`, seguida de una expresión musical. Si *y solo si* los símbolos son identificadores de LilyPond válidos (caracteres alfabéticos solamente, sin números, guiones bajos ni guiones normales y que no puedan confundirse con notas), el prefijo `#'` se puede omitir y, como abreviatura, una lista de símbolos puede usar el separador de coma. Por ejemplo, `\tag #'(violinI violinII)` se puede escribir como `\tag violinI,violinII`. Lo mismo vale para los argumentos de `\keepWithTag` y `\removeWithTag`. Las instrucciones de etiquetado son funciones musicales, por lo que no se pueden usar para filtrar elementos que no son expresiones musicales, tales como bloques `\book` o `\score`.

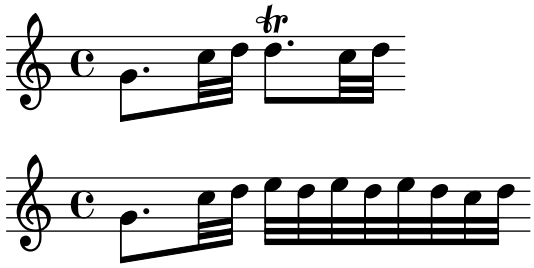
En el ejemplo siguiente vemos dos versiones de una pieza musical, una que muestra trinos con la notación usual y otra con los trinos expandidos explícitamente.

```
music = \relative {
  g'8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand { \repeat unfold 3 { e32 d } }
  c32 d
 }

\score {
  \keepWithTag #'trills \music
}
\score {
  \keepWithTag #'expand \music
}
```



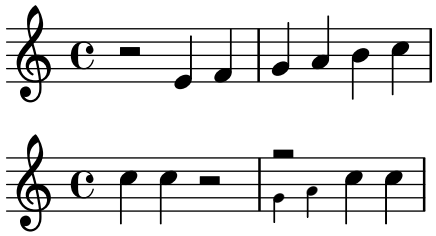
De forma alternativa, a veces es más fácil excluir secciones de música.

```
music = \relative {
  g'8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand { \repeat unfold 3 { e32 d } }
  c32 d
 }

\score {
  \removeWithTag #'expand \music
}
\score {
  \removeWithTag #'trills \music
}
```



Si las etiquetas marcan alternativas que tienen una duración no nula, las alternativas son normalmente simultáneas como concepto. En este caso es mejor poner las alternativas dentro de una expresión musical simultánea de manera que la expresión musical tenga la misma duración sin importar qué etiquetas se están reteniendo. Esto es importante especialmente si está usando etiquetas en combinación con instrucciones como \cueDuring.

```
outputTypeTag = "isScore"

firstInstrument = \relative c' {
  <<
    \tag #'isPart {
      \cueDuring "quoteSecondInstrument" #UP { r2 } }
    \tag #'isScore { r2 }
  >>
  e4 f |
  g4 a b c |
}

secondInstrument= \relative c'' {
  c4 c r2 |
  \cueDuring "quoteFirstInstrument" #DOWN { r2 }
  c4 c |
}

\addQuote quoteFirstInstrument \firstInstrument
\addQuote quoteSecondInstrument \secondInstrument

\new Staff {
  \keepWithTag \outputTypeTag \firstInstrument
}

\new Staff {
  \keepWithTag \outputTypeTag \secondInstrument
}
```



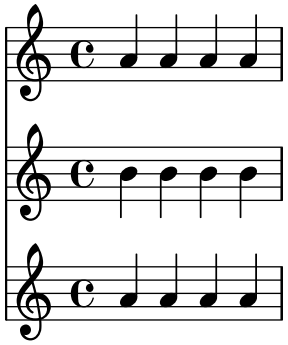
El fitrado mediante etiquetas se puede aplicar a articulaciones, textos, etc., anteponiendo `-\tag #'mi-etiqueta`

a una articulación. Por ejemplo, lo siguiente define una nota con una indicación opcional de digitación y una nota con una anotación condicional.

```
c1-\tag #'finger ^4
c1-\tag #'warn ^"¡Ojo!"
```

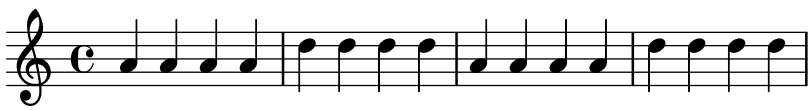
Se pueden poner varias etiquetas sobre expresiones con varias entradas de \tag, o combinando varias etiquetas en una lista de símbolos.

```
music = \relative c'' {
  \tag AA \tag AB { a4 a a a }
  \tag BB,both { b4 b b b }
}

<<
  \keepWithTag AA \music
  \keepWithTag BB \music
  \keepWithTag AB \music
>>
```



Se pueden aplicar varios filtros \removeWithTag a una sola expresión musical para quitar varias secciones etiquetadas con nombres distintos. De forma alternativa, podemos usar una sola instrucción \removeWithTag con una lista de etiquetas.

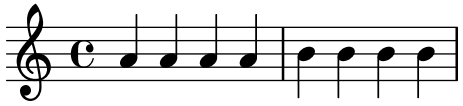
```
music = \relative c'' {
  \tag AA { a4 a a a }
  \tag BB { b4 b b b }
  \tag CC { c4 c c c }
  \tag DD { d4 d d d }
}

\new Voice {
  \removeWithTag BB \removeWithTag CC \music
  \removeWithTag BB,CC \music
}
```



El uso de dos o más filtros \keepWithTag sobre una sola expresión musical produce la eliminación de *todas* las secciones etiquetadas, que no es lo que queremos: el primer filtro quita todas excepto la que se nombra, y cualquier filtro posterior elimina el resto. En su lugar, use una sola instrucción \keepWithTag con una lista de varias etiquetas para suprimir todas las secciones

etiquetadas que no aparezcan especificadas en la lista. Como ejemplo, el código siguiente imprime música etiquetada con *violinI* y *violinII* pero no la etiquetada con *viola* ni *cello*.

```
music = \relative c'' {
  \tag violinI { a4 a a a }
  \tag violinII { b4 b b b }
  \tag viola { c4 c c c }
  \tag cello { d4 d d d }
}

\new Staff {
  \keepWithTag violinI,violinII \music
}
```



Uso de los grupos de etiquetas

Aunque `\keepWithTag` es útil para el manejo de *un solo* conjunto de alternativas, la eliminación de música etiquetada con otras etiquetas no relacionadas es problemática si se utilizan para más de un propósito. En ese caso las etiquetas se pueden agrupar con `\tagGroup`. El código siguiente

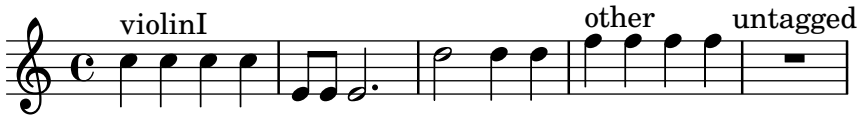
```
\tagGroup violinI,violinII,viola,cello
```

sitúa las cuatro etiquetas en un solo grupo de etiquetas. Como resultado, un código como el siguiente

```
\keepWithTag violinI ...
```

ahora mostrará solamente la música etiquetada a partir del grupo de etiquetas `violinI` y cualquier fragmento musical etiquetado con una de las otras etiquetas resultará suprimido, como muestra el ejemplo siguiente.

```
\tagGroup violinI,violinII
\tagGroup original,arranged

music = \relative {
  \tag violinI { c''4^"violinI" c c c }
  \tag violinII { a2 a }
  \tag original { e8 e e2. }
  \tag arranged { d'2 d4 d }
  \tag other { f^"other" f f f }
  R1^"untagged"
}

\new Voice {
  \keepWithTag violinI \music
}
```



Observe que las etiquetas individuales no pueden ser miembros de más de un grupo de etiquetas.

Existe la posibilidad de modificar los grupos de etiquetas. Hay una forma alternativa de definirlos mediante el uso de `\tagGroupRef` para así poder hacer referencia al grupo más tarde para su modificación. La llamada

```
tg = \tagGroupRef violinI,violinII
```

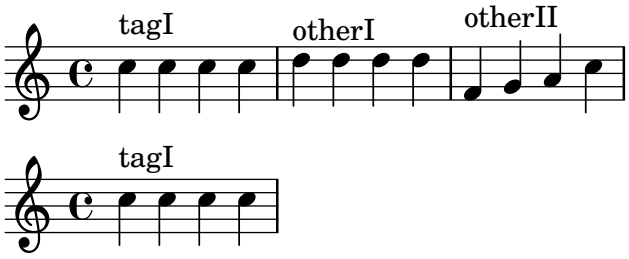
es, en esencia, lo mismo que

```
tg = #'(violinI violinII)
\tagGroup \tg
```

Hay varias posibilidades diferentes para modificar grupos de etiquetas existentes.

- Para deshacer los efectos de cada llamada de `\tagGroup` o de `\tagGroupRef` podemos simlemente usar `\resetTagGroups`.

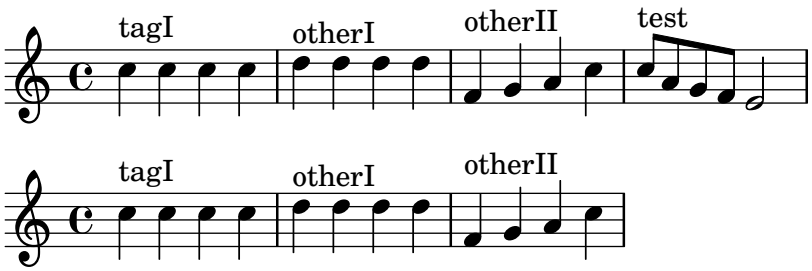
```
\tagGroup tagI,tagII

music = \relative {
  \tag tagI { c''4^"tagI" c c c }
  \tag tagII { a2^"tagII" a }
  \tag otherI { d4^"otherI" d d d }
  \tag otherII { f,4^"otherII" g a c }
}

\new Voice {
  \keepWithTag tagI \music
}

\resetTagGroups

\new Voice {
  \keepWithTag tagI \music
}
```



- Si queremos reiniciar un grupo de etiquetas específico, tenemos que usar `\resetTagGroup` con un grupo de etiquetas dado, especificado a través de todas sus etiquetas. Es cómodo usar `\tagGroupRef` para la definición del grupo de etiquetas.

```
tgI = \tagGroupRef tagI,tagII
tgII = \tagGroupRef otherI,otherII

music = \relative {
  \tag tagI { c''4^"tagI" c c c }
  \tag tagII { a2^"tagII" a }
  \tag otherI { d4^"otherI" d d d }
  \tag otherII { f,4^"otherII" g a c }
  \tag test { c8^"test" a g f e2 }
}
```

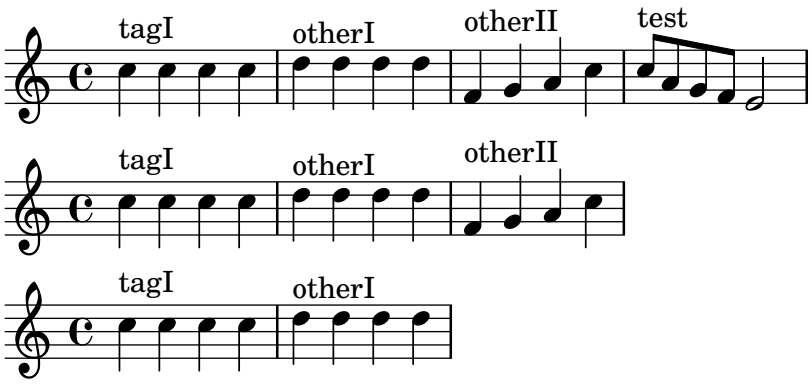
```
% 'otherI', 'otherII' and 'test' are not in the same group
% as 'tagI' so they stay untouched.
\new Voice {
  \keepWithTag tagI \music
}

\resetTagGroup \tgI

% now only 'otherI' and 'otherII' are in a tag group,
% so '\keepWithTag tagI' filters 'tagII' and 'test'.
\new Voice {
  \keepWithTag tagI \music
}
```



- Otra manera de modificar un grupo de etiquetas existente es añadir o eliminar etiquetas sencillas con las instrucciones \addToTagGroup y \removeFromTagGroup.

```
tgI = \tagGroupRef tagI,tagII
tgII = \tagGroupRef otherI,otherII

music = \relative {
  \tag tagI { c''4^"tagI" c c c }
  \tag tagII { a2^"tagII" a }
  \tag otherI { d4^"otherI" d d d }
  \tag otherII { f,4^"otherII" g a c }
  \tag test { c8^"test" a g f e2 }
}

% 'otherI', 'otherII' and 'test' are not in the same group
% as 'tagI' so they stay untouched.
\new Voice {
  \keepWithTag tagI \music
}

\addToTagGroup \tgI test

% now 'tagI', 'tagII' and 'test' are in the same group
% so 'tagII' and 'test' are removed.
\new Voice {
  \keepWithTag tagI \music
}

\removeFromTagGroup \tgII otherII
\addToTagGroup \tgI otherII
```

```
% now 'otherII' is removed from 'tgII' and
% added to 'tgI' so only 'otherI' is in its
% own tag group and is not filtered.
\new Voice {
  \keepWithTag tagI \music
}
```



Añadir contenido antes y después de la música etiquetada

A veces podemos desear dividir la música en un determinado lugar dentro de una expresión musical existente. Podemos usar \pushToTag y \appendToTag para añadir material delante o al final de los `elements` de diversas construcciones musicales. Los lugares contemplados son

música secuencial y simultánea
: Si etiqueta una construcción completa {...} o <<...>>, puede añadir expresiones musicales por delante o por detrás.

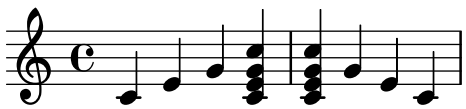
acordes
: Si etiqueta un acorde <...>, puede añadir notas por delante o bien por detrás, o articulaciones para el acorde completo.

notas y silencios
: Si etiqueta una nota (también dentro de un acorde) o un silencio, puede añadir articulaciones por delante o por detrás de las articulaciones existentes. Observe que para añadir otras notas, en cambio, tiene que poner la nota dentro de un acorde y etiquetar el *acorde*.

  Observe también que no se puede etiquetar una sola articulación y añadirla porque no es inherentemente una lista. En lugar de ello, etiquete la nota.

Las dos instrucciones toman una etiqueta, el material en el que separar en cada aparición de la etiqueta, y la expresión etiquetada.

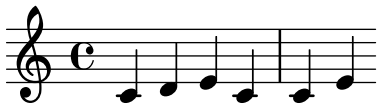
```
music = { \tag #'here { \tag #'here <<c''>> } }

{
  \pushToTag #'here c'
  \pushToTag #'here e'
  \pushToTag #'here g' \music
  \appendToTag #'here c'
  \appendToTag #'here e'
  \appendToTag #'here g' \music
}
```



Sustituir música etiquetada

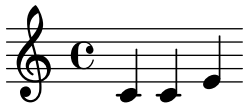
Se puede sustituir música etiquetada con otra música.

```
music = { c' \tag #'here { d' } e' }

{
  \music
  \replaceWithTag #'here c' \music
}
```



No olvide que la etiqueta desaparece después de la sustitución, de manera que solo podemos sustituirla una vez. Asimismo, no se puede realizar un filtrado después de la susticuión.

```
music = { c' \tag #'here { d' } e' }

{
  \removeWithTag #'here % does nothing because tag is replaced
  \replaceWithTag #'here c'
  \music
}
```



Etiquetado dentro de elementos de marcado

La instrucción \tag también se puede usar dentro de entornos \markup. LilyPond aporta las instrucciones de marcado \keep-with-tag, \remove-with-tag, \push-to-tag, \append-to-tag y \replace-with-tag de forma análoga a las instrucciones correspondientes para expresiones musicales.

```
test = \markup {
  \tag #'AA a
  \tag #'BB b
  \tag #'CC c
}

\markup { \keep-with-tag #'BB \test }
\markup { \remove-with-tag #'BB \test }
\markup { \push-to-tag #'CC pre \test }
\markup { \append-to-tag #'CC post \test }
\markup { \replace-with-tag #'CC sub \test }
```

b

a c

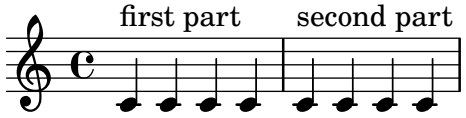
a b pre c

a b c post

a b sub

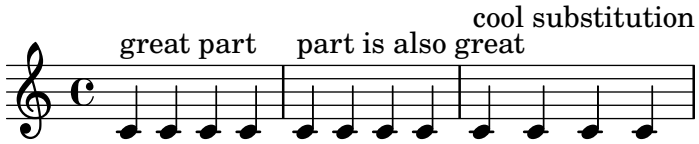
Las instrucciones musicales como \keepWithTag y \removeWithTag filtran etiquetes dentro de las partes \markup dentro de la música relacionada, también.

```
music = \relative {
  c'4^\markup { \tag #'one first \tag #'two second part } c c c
}

{
  \keepWithTag #'one \music
  \removeWithTag #'one \music
}
```



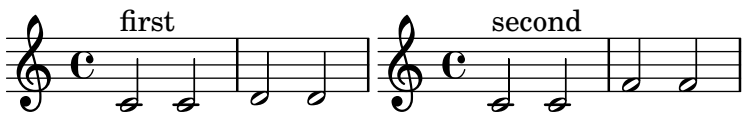
También es posible añadir algo al principio y al final del \markup de objetos musicales. Sin embargo, no se pueden usar \pushToTag, \appendToTag ni \replaceWithTag porque solo insertan música, así que necesitamos instrucciones como \pushToTagMarkup, \appendToTagMarkup o \replaceWithTagMarkup para insertar elementos de marcado.

```
music = \relative {
  c'4^\markup { \tag #'part part } c c c
}

{
  \pushToTagMarkup #'part "great" \music
  \appendToTagMarkup #'part \markup { is also great } \music
  \replaceWithTagMarkup #'part \markup { cool substitution } \music
}
```



El filtrado de etiquetas también funciona para música empotrada dentro de bloques \score en instrucciones de marcado.

```
music = \relative {
  c'2^\markup { \tag #'first first \tag #'second second } c
  \tag #'first { d d }
  \tag #'second { f f }
}

\markup {
  \keep-with-tag #'first \score { \music }
  \remove-with-tag #'first \score { \music }
}
```



El uso de etiquetas con listas de marcado requiere un cuidado especial. Aunque las funciones de filtrado funcionan como se espera, no es así con instrucciones como \push-to-tag, \append-to-tag y \replace-with-tag.

```
\markup {
  \remove-with-tag #'test { a \tag #'test { b c } d }
}
\markup {
  \push-to-tag #'test "twice" { a \tag #'test { b c } d }
}
\markup {
  \replace-with-tag #'test "twice" { a \tag #'test { b c } d }
}
```

a d

a twice b twice c d

a twice twice d

El motivo de este comportamiento es que LilyPond resuelve

```
\tag #'test { b c }
```

internamente a

```
\tag #'test b
\tag #'test c
```

y por ello el texto dado aparece insertado dos veces. Para etiquetar una lista para insertar o añadir algo antes o después de la lista completa, se debe usar la instrucción \tag-list.

```
\markup {
  \push-to-tag #'test "once" { a \tag-list #'test { b c } d }
}
```

a once b c d

Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2.5 [Combinación automática de las partes], página 235, Sección 22.1 [Inclusión de archivos de LilyPond], página 627.

Advertencias y problemas conocidos

Al llamar a \relative sobre una expresión musical obtenida mediante el filtrado de música con las instrucciones \keepWithTag o \removeWithTag, podría ocurrir que cambiaran las relaciones de octava, dado que solo se tienen en cuenta las alturas de las notas que efectivamente permanecen en la expresión filtrada. La aplicación de \relative en primer lugar, antes de \keepWithTag o de \removeWithTag, evita este peligro porque en este caso \relative actúa sobre todas las notas tal y como se introducen.

Incluso si se usan grupos de etiquetas, las instrucciones como \keepWithTag no son conmutativas. Un código como el siguiente

```
\keepWithTag violinI,original \someMusic
```

no produce la misma salida que

```
\keepWithTag violinI \keepWithTag original \someMusic
```

22.2.3 Uso de ajustes globales

Se pueden incluir ajustes globales a partir de un archivo distinto:

```
lilypond -dinclude-settings=MIS_AJUSTES.ly MI_PARTITURA.ly
```

En archivos diferentes se pueden almacenar grupos de ajustes como el tamaño de la página o las fuentes tipográficas, y estos grupos de ajustes se pueden cargar con varias opciones de `-dinclude-settings`. Ello permite hacer diferentes ediciones de la misma partitura así como aplicar ajustes estándar a muchas partituras, simplemente por medio de la especificación del archivo de ajustes adecuado.

Esta técnica también funciona bien con el uso de hojas de estilo, como se estudia en Sección "Hojas de estilo" en *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección "Organizar las piezas mediante variables" en *Manual de Aprendizaje*, Sección "Hojas de estilo" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 22.1 [Inclusión de archivos de LilyPond], página 627.

22.3 Uso de las funciones musicales

En los lugares donde se necesita reutilizar los trucos con diferentes expresiones musicales, suele ser conveniente hacer que el truco forme parte de una *función_musical*. En esta sección vamos a tratar solo las funciones de *sustitución*, donde el objeto sustituye a una variable dentro del código de entrada de LilyPond. Otra funciones más complejas aparecen descritas en Sección "Music functions" en *Extender*.

22.3.1 Sintaxis de las funciones de sustitución

Es fácil hacer una función que sustituya a una variable en código de LilyPond. La forma general de estas funciones es

```
function =
#(define-music-function
     (arg1 arg2 ...)
     (tipo1? tipo2? ...)
   #{
     ...música...
   #})
```

donde

argN — El *n*-ésimo argumento.

tipoN? — Un *predicado_de_tipo* de Scheme para el que *argN* debe devolver `#t`.

...música... — Código normal de entrada de LilyPond, que usa '`$`' (en aquellos lugares donde solo se admiten construcciones de LilyPond) o '`#`' (para usarlo como un valor de Scheme o argumento de función musical o música dentro de listas de música) para referenciar argumentos (p.ej., '`#arg1`').

La lista de predicados de tipo es obligatoria. Algunos de los predicados de tipo más comunes que se usan en las funciones musicales son:

boolean?

cheap-list? *(usar en lugar de* 'list?' *para un procesamiento más rápido)*

ly:duration?
ly:music?
ly:pitch?
markup?
number?
pair?
string?
symbol?

Para ver una lista de los predicados de tipo disponibles, véase Sección B.25 [Predicados de tipo predefinidos], página 978. También se permiten predicados de tipo definidos por el usuario.

Véase también

Referencia de la notación: Sección B.25 [Predicados de tipo predefinidos], página 978.

Manual de extensión de LilyPond: Sección “Music functions” en *Extender*.

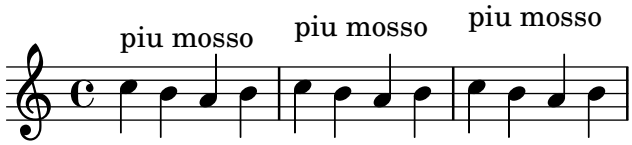
Archivos instalados: `lily/music-scheme.cc`, `scm/c++.scm`, `scm/lily.scm`.

22.3.2 Ejemplos de funciones de sustitución

Esta sección introduce algunos ejemplos de funciones de sustitución. No pretenden ser exhaustivos, sino más bien mostrar algunas de las posibilidades de las funciones de sustitución sencillas.

En el primer ejemplo, se define una función que simplifica el establecimiento de un espacio de relleno para un objeto gráfico `TextScript`:

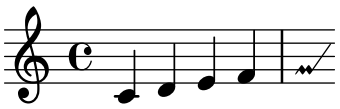
```
padText =
#(define-music-function
     (padding)
     (number?)
   #{
     \once \override TextScript.padding = #padding
   #})

\relative {
  c''4^"piu mosso" b a b
  \padText 1.8
  c4^"piu mosso" b a b
  \padText 2.6
  c4^"piu mosso" b a b
}
```



Además de números, se pueden usar expresiones como notas en los argumentos de las funciones musicales:

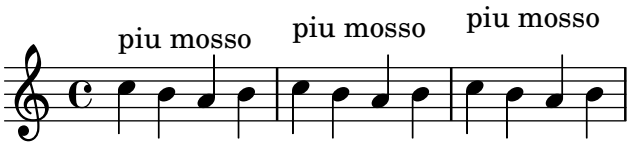
```
custosNote =
#(define-music-function
     (note)
     (ly:music?)
   #{
```

```
    \tweak NoteHead.stencil #ly:text-interface::print
    \tweak NoteHead.text
       \markup \musicglyph "custodes.mensural.u0"
    \tweak Stem.stencil ##f
    #note
  #})

\relative { c'4 d e f \custosNote g }
```

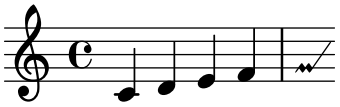


Ambas funciones son expresiones únicas sencillas en las que solo falta el último elemento de una llamada de función o una sobreescritura. Para esas definiciones de función en particular existe una sintaxis alternativa más simple, que consiste en escribir solo la parte constante de la expresión y sustituir el elemento final que falta con `\etc`:

```
padText =
  \once \override TextScript.padding = \etc

\relative {
  c''4^"piu mosso" b a b
  \padText 1.8
  c4^"piu mosso" b a b
  \padText 2.6
  c4^"piu mosso" b a b
}
```



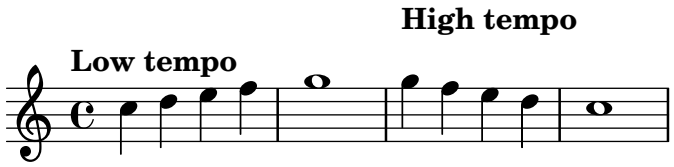

```
custosNote =
  \tweak NoteHead.stencil #ly:text-interface::print
  \tweak NoteHead.text
     \markup \musicglyph "custodes.mensural.u0"
  \tweak Stem.stencil ##f
  \etc

\relative { c'4 d e f \custosNote g }
```



Se pueden definir funciones de sustitución con más de un argumento:

```
tempoPadded =
#(define-music-function
     (padding tempotext)
     (number? markup?)
   #{
     \once \override Score.MetronomeMark.padding = #padding
     \tempo \markup { \bold #tempotext }
```

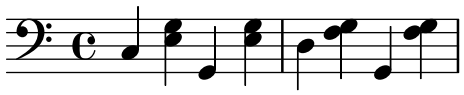
```
   #})

\relative {
  \tempo \markup { "Low tempo" }
  c''4 d e f g1
  \tempoPadded 4.0 "High tempo"
  g4 f e d c1
}
```



22.3.3 Cómo evitar la compartición de expresiones musicales

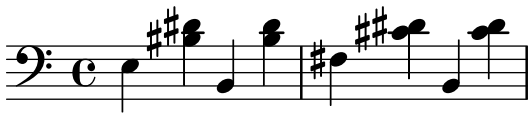
Al escribir funciones musicales, es importante tener en cuenta una regla: las mismas expresiones musicales no se deben compartir en más de un lugar. Como ejemplo, he aquí una función problemática:

```
simpleAccompaniment =
#(define-music-function
   (bass-1 bass-2 chord) (ly:music? ly:music? ly:music?)
   #{
     #bass-1 #chord #bass-2 #chord
   #})

{
  \clef bass
  \simpleAccompaniment c g, <e g>
  \simpleAccompaniment d g, <f g>
}
```



El problema con esta función queda claro si se transporta el resultado:

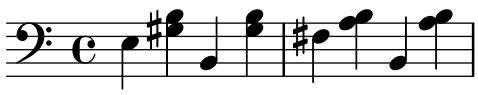
```
simpleAccompaniment =
#(define-music-function
   (bass-1 bass-2 chord) (ly:music? ly:music? ly:music?)
   #{
     #bass-1 #chord #bass-2 #chord
   #})

\transpose c e {
  \clef bass
  \simpleAccompaniment c g, <e g>
  \simpleAccompaniment d g, <f g>
}
```



Aunque las notas del bajo son correctas, el acorde no se transporta adecuadamente: de hecho, se transporta dos veces. El motivo de esto es que la expresión musical *chord* se ha usado dos veces en el resultado de la función, sin copiarlo. Las funciones como `\transpose` modifican el objeto musical directamente (en el caso de `\transpose`, cambia la altura de las notas). Si se reutiliza el mismo objeto musical, las modificaciones efectuadas en un lugar donde se usa afecan a los dos lugares, porque contienen el mismo objeto. En este caso, `\transpose` encuentra el objeto dos veces y por tanto lo transporta dos veces también.

Una forma de arreglar esta función es usar '$' en lugar de '#' para referenciar las variables, lo cual hace una copia. La diferencia entre '#' y '$' se detalla en Sección "LilyPond Scheme syntax" en *Extender*.

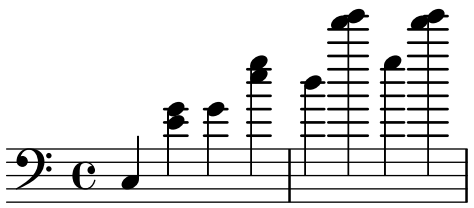
```
simpleAccompaniment =
#(define-music-function
   (bass-1 bass-2 chord) (ly:music? ly:music? ly:music?)
   #{
     $bass-1 $chord $bass-2 $chord
   #})

\transpose c e {
  \clef bass
  \simpleAccompaniment c g, <e g>
  \simpleAccompaniment d g, <f g>
}
```



22.3.4 Funciones de sustitución y escritura de octava relativa

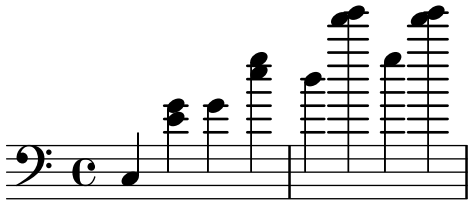
Cuando se aplica `\relative` a una expresión musical, atraviesa su busqueda de notas con su altura, y modifica las alturas en el orden en que las va encontrando, cambiando la octava de cada altura de acuerdo con sus marcas de octava ('' y ',') y la altura de la nota anterior. Cuando se escribem funciones de sustitución, esto puede dar lugar a una situación en que una expresión musical resulta 'relativizada' de una forma inesperada porque la salida de la función usa los parámetros varias veces y/o en un orden distinto. Considere esta función y cómo su salida reacciona al `\relative`:

```
simpleAccompaniment =
#(define-music-function
   (bass-1 bass-2 chord) (ly:music? ly:music? ly:music?)
   #{
     $bass-1 $chord $bass-2 $chord
   #})

\relative {
  \clef bass
  \simpleAccompaniment c g <e' g>
  \simpleAccompaniment d g, <f' g>
}
```



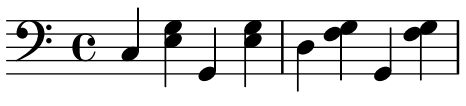
En este ejemplo, la salida es la misma que la de

```
\relative {
  \clef bass
  c <e' g> g <e' g>
  d <f' g>  g, <f' g>
}
```



Sin embargo, este no es el resultado deseado cuando se usa la función `\simpleAccompaniment`. La nota `g,` es relativa a la primera nota del acorde que la precede, `<e' g>`, aunque aparece después de `c` en la entrada. Claramente las alturas tendrían que ser relativas de acuerdo al orden en que se escriben al usar la función, no en el orden en que aparecen en la salida de la función. Esto se puede conseguir usando el macro de Scheme `make-relative`. Sus argumentos son: una lista de variables, una expresión de referencia, y una expresión musical principal. La expresión de referencia tiene el propósito de servir como maqueta de cómo las variables se escribieron en la entrada. Casi siempre puede ser una expresión simple hecha con `#{ ... #}` conteniendo las variables en orden. Tenga cuidado de *no* hacer copias en la expresión de referencia (concretamente, use '`#`' y no '`$`'). El ejemplo anterior se puede arreclar usando `make-relative` de esta forma:

```
simpleAccompaniment =
#(define-music-function
   (bass-1 bass-2 chord) (ly:music? ly:music? ly:music?)
   (make-relative
    (bass-1 bass-2 chord)
    #{ #bass-1 #bass-2 #chord #}
    #{ $bass-1 $chord $bass-2 $chord #}))

\relative {
  \clef bass
  \simpleAccompaniment c g <e' g>
  \simpleAccompaniment d g, <f' g>
}
```



22.4 Caracteres especiales

22.4.1 Codificación del texto

LilyPond usa el conjunto de caracteres definido por el consorcio Unicode y la norma ISO/IEC 10646. Define un nombre único y un código para los conjuntos de caracteres que se utilizan en prácticamente todos los idiomas modernos y también en muchos otros. Unicode se puede implementar utilizando varios esquemas de codificación distintos. LilyPond usa la codificación UTF-8 (UTF son las siglas de *Unicode Transformation Format*, o formato de transformación de Unicode) que representa todos los caracteres comunes de la codificación Latin en un solo byte, y representa otros caracteres usando un formato de longitud variable de hasta cuatro bytes.

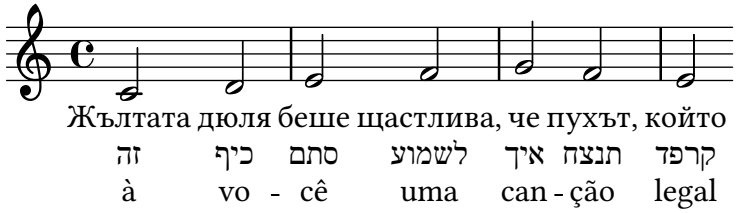
El aspecto visual real de los caracteres viene determinado por los glifos que se definen en las fuentes tipográficas concretas que se tengan disponibles: una fuente tipográfica define la asignación de un subconjunto de los códigos de Unicode a glifos. LilyPond usa la biblioteca Pango para representar y disponer tipográficamente textos multilingües.

LilyPond no realiza ninguna conversión en la codificación de la entrada. Esto significa que cualquier text, ya sea el título, la letra de la canción o una instrucción musical que contenga caracteres distintos a los del conjunto ASCII, se deben codificar en UTF-8. La forma más fácil de escribir dicho texto es utilizar un editor preparado para Unicode y guardar el archivo con la codificación UTF-8. Casi todos los editores modernos populares contemplan el UTF-8, por ejemplo lo hacen vim, Emacs, jEdit y Gedit. Todos los sistemas MS Windows posteriores a NT usan Unicode como codificación de caracteres nativa, de manera que incluso el accesorio Bloc de Notas (Notepad) puede editar y guardar un archivo en el formato UTF-8. Una alternativa más funcional para Windows es BabelPad.

Si un archivo de entrada que contiene un carácter que no es ASCII, no se guarda en el formato UTF-8, se genera el mensaje de error

```
FT_Get_Glyph_Name () error: argumento inválido
```

He aquí un ejemplo que muestra texto cirílico, hebreo y portugués:



22.4.2 Unicode

Para introducir un carácter aislado para el que se conoce el punto de código Unicode pero no está disponible en el editor que se está usando, use `\char ##xhhhh` o bien `\char #dddd` dentro de un bloque `\markup`, donde `hhhh` es el código hexadecimal del carácter en cuestión y `dddd` es su valor decimal correspondiente. Pueden omitirse los ceros iniciales, pero es costumbre indicar los cuatro caracteres en la representación hexadecimal (observe que *no* debe utilizarse la codificación UTF-8 del punto de código Unicode después de `\char`, pues la codificación UTF-8 contiene bits adicionales que indican el número de octetos). Hay tablas de códigos Unicode y un índice de nombres de caracteres que da el punto de código en hexadecimal para cualquier carácter en el portal del Consorcio Unicode, `https://www.unicode.org/`.

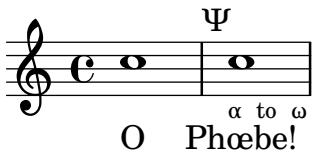
Por ejemplo, tanto `\char ##x03BE` como `\char #958` insertan el carácter Unicode U+03BE, que tiene el nombre Unicode “Letra griega Xi pequeña”.

Se puede escribir de esta forma cualquier punto de código Unicode, y si todos los caracteres especiales se escriben en este formato no es necesario guardar el archivo de entrada en formato

UTF-8. Por supuesto, debe estar instalada y estar disponible para LilyPond una fuente tipográfica que contenga codificados todos estos caracteres.

El ejemplo siguiente muestra valores hexadecimales Unicode que se usan en cuatro lugares: en una llamada de ensayo, como texto de articulación, en la letra y como texto independiente bajo la partitura:

```
\score {
  \relative {
    c''1
    \textMark \markup { \char ##x03A8 }
    c1_\markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } }
  }
  \addlyrics { O \markup { \concat { Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008--2026" \char ##x00A9 }
```



Copyright 2008--2026 ©

Para escribir el símbolo de copyright en la nota de créditos, utilice:

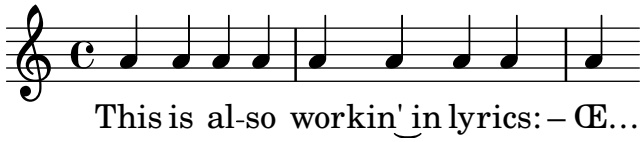
```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

22.4.3 Alias de ASCII

Se puede incluir una lista de alias de ASCII para caracteres especiales:

```
\paper {
  #(include-special-characters)
}

\markup "&flqq; &ndash; &OE;uvre incomplète&hellip; &frqq;"

\score {
  \new Staff { \repeat unfold 9 a'4 }
  \addlyrics {
    This is al -- so wor -- kin'~in ly -- rics: &ndash;_&OE;&hellip;
  }
}

\markup \column {
  "The replacement can be disabled:"
  "&ndash; &OE; &hellip;"
  \override #'(replacement-alist . ()) "&ndash; &OE; &hellip;"
}
```

« – Œuvre incomplète… »



The replacement can be disabled:
– Œ …
– &OE; …

También podemos hacer nuestros propios alias, ya sea de forma global,

```
\paper {
  #(add-text-replacements!
    '(("100" . "hundred")
      ("dpi" . "dots per inch")))
}
\markup "A 100 dpi."
```

A hundred dots per inch.

o local:

```
\markup \replace #'(("100" . "hundred")
                    ("dpi" . "dots per inch")) "A 100 dpi."
```

A hundred dots per inch.

La sustitución no es necesariamente una cadena de caracteres; puede ser un elemento de marcado arbitrario. A nivel de sintaxis, esto requiere usar la sintaxis de cuasi-cita de Scheme, con un apóstrofo invertido ‘`’ en lugar de un apóstrofo simple normal ‘'’ para escribir la lista-A.

```
\markup \replace
  #`(("2nd" . ,#{ \markup \concat { 2 \super nd } #})) "2nd time"
```

2nd time

Los propios alias no se siguen procesando para posibles sustituciones.

Véase también

Referencia de la notación: Sección B.12 [Lista de caracteres especiales], página 932.

Archivos instalados: `ly/text-replacements.ly`.

23 Controlar la salida

23.1 Extracción de fragmentos de música

Es posible producir en la salida uno o más fragmentos de una partitura. Para hacerlo, use la variable `clip-regions` dentro del bloque `\layout` o `\paper` para definir una lista de ubicaciones explícitas de la música que se quiere extraer, y a continuación ejecute `lilypond` con la opción `-dclip-systems`.

```
\layout {
  clip-regions
  = #(list (cons (make-rhythmic-location 5 1 2)
                 (make-rhythmic-location 7 3 4)))
}
```

Este ejemplo extrae un solo fragmento del archivo de entrada comenzando después de la duración de una blanca en el quinto compás (`5 1 2`) y terminando después de la tercera negra del compás séptimo (`7 3 4`).

Se pueden extraer fragmentos adicionales añadiendo más pares de entradas de `make-rhythmic-location` a la lista `clip-regions` dentro del bloque `\layout`.

Si se incluyen los comienzos y finales de los sistemas, incluyen las dimensiones del objeto gráfico `System`, p.ej. los nombres de instrumento.

No se incluyen las notas de adorno en el punto de final de la región.

Cada fragmento de música se escribe como un archivo distinto. La música extraída se imprime como si hubiera sido literalmente 'recortada' a partir de la partitura original impresa, de manera que si un fragmento se extiende sobre una o más líneas, se genera un archivo de salida diferente por cada línea. Suponiendo que el ejemplo de arriba ocupa dos líneas en la salida PDF del archivo de entrada `foo.ly`, los archivos de salida para los fragmentos musicales se llamarían `foo-from-5.1.2-to-7.3.4-clip.pdf` y `foo-from-5.1.2-to-7.3.4-clip-1.pdf`.

Véase también

Referencia de la notación: Sección 27.1 [El bloque `\layout`], página 686.

Manual de utilización: Sección "Utilización desde la línea de órdenes" en *Utilización del Programa*.

23.2 Saltar la música corregida

Al escribir o copiar música, normalmente solo es interesante de ver y corregir la música cercana al final (donde estamos añadiendo las notas). Para acelerar este proceso de corrección, es posible saltar la composición tipográfica de todos excepto unos pocos de los últimos compases. Esto se consigue definiendo una variable especial al principio del archivo fuente, como sigue:

```
showLastLength = R1*5
\score { ... }
```

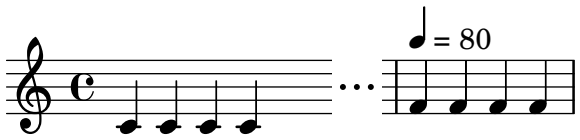
En este ejemplo no se dibujará nada excepto los últimos cinco compases (suponiendo un compás de 4/4) de cada una de las partituras `\score` del archivo de entrada. Para piezas largas, el tipografiado de únicamente una parte pequeña es con frecuencia un orden de magnitud más rápido que el de la obra completa. Si estamos trabajando sobre el principio de una partitura que ya hemos tipografiado (p.ej., para añadir una parte instrumental nueva), también puede ser útil la propiedad `showFirstLength`.

Este mecanismo de pasar por alto partes de un partitura se puede controlar con un grano más fino mediante la propiedad `Score.skipTypesetting`. Cuando su valor está establecido, no

se lleva a cabo ningún tipografiado en absoluto. Como propiedad del contexto `Score`, afecta a todas las voces y pentagramas; véase Sección 33.1.2 [`Score`: el contexto maestro], página 742.

Esta propiedad se usa también para controlar la salida hacia el archivo MIDI. Si algún evento dentro de la sección que se salta altera alguna de sus propiedades de contexto, por ejemplo una indicación de tempo o un cambio de instrumento, entonces el ajuste tendrá efecto solamente en el punto temporal en que `skipTypesetting` se desactiva de nuevo:

```
\relative c' {
  c4 c c c
  \set Score.skipTypesetting = ##t
  d4 d d d
  \tempo 4 = 80
  e4 e e e
  \set Score.skipTypesetting = ##f
  f4 f f f
}
```



Instrucciones predefinidas

`showLastLength`, `showFirstLength`.

Véase también

Referencia de la notación: Capítulo 33 [Contextos de interpretación], página 742, Sección 33.1.2 [`Score`: el contexto maestro], página 742.

Referencia de funcionamiento interno: Sección “Todas las propiedades de contexto” en *Referencia de Funcionamiento Interno*.

23.3 Formatos de salida alternativos

Los formatos de salida predeterminados para la partitura impresa son el formato de documento portátil (PDF) y PostScript (PS). Los formatos de salida Gráficos de red portátiles (PNG), Gráficos de vector escalables (SVG) y PostScript encapsulado (EPS) también están disponibles a través de opciones de la línea de órdenes, véase Sección “Opciones básicas de la línea de órdenes para LilyPond” en *Utilización del Programa*.

23.3.1 Salida de SVG

La salida de SVG puede contener, opcionalmente, metadatos para objetos gráficos (grobs) como cabezas de nota, silencios, etc. Estos metadatos pueden ser atributos SVG estándar como `id` y `class`, o atributos personalizados no estándar. Especifique los atributos y sus values mediante la sobreescritura de la propiedad de grob `output-attributes` con una lista de asociación de Scheme (alist). Los valores pueden ser números, cadenas de caracteres o símbolos. Por ejemplo:

```
{
  \once \override NoteHead.output-attributes =
  #'((id . 123)
     (class . "esto aquello")
     (data-lo-que-sea . algo))
  c
}
```

El código de entrada anterior produce la siguiente etiqueta <g> (grupo) dentro del archivo SVG:

```
<g id="123" class="esto aquello" data-lo-que-sea="algo">
  ...elementos del SVG de objetos gráficos como cabezas de nota...
</g>
```

La etiqueta <g> contiene todos los elementos de SVG para un grob dado (ciertos grobs generan más de un elemento del SVG). En la sintaxis del SVG el prfijo `data-` se usa para atributos de medios personalizados no estándar.

23.4 Empotrar archivos dentro de la salida PDF

La opción de la línea de órdenes `-dembed-source-code` hace que LilyPond *empotre* o inserte todos los archivos fuente (del usuario) necesarios para la compilación, dentro del archivo final PDF de salida (véase Sección “Advanced command-line options for LilyPond” en *Utilización del Programa*); un visor de PDF puede extraer después esos contenidos adjuntos para su uso posterior.

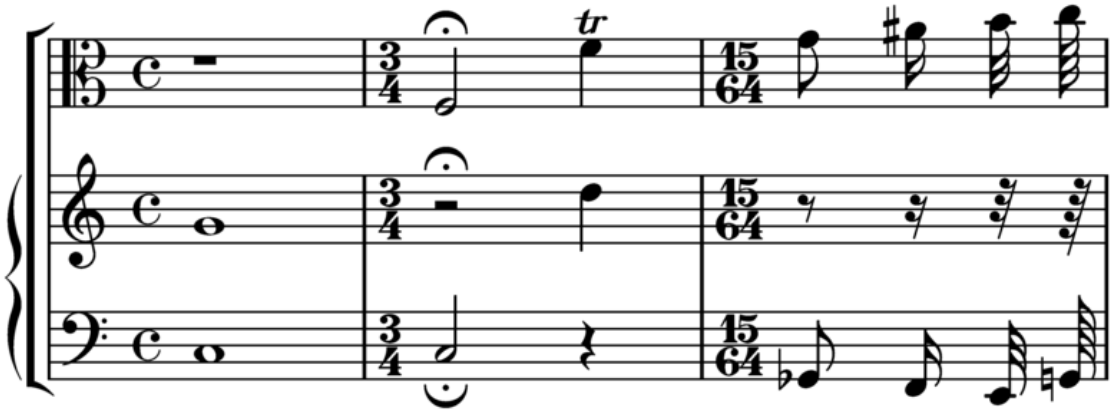
De forma parecida, es posible incrustar archivos arbitrarios dentro de la salida PDF con la función `ly:note-extra-source-file` (véase Sección “Scheme functions” en *Referencia de Funcionamiento Interno*).

23.5 Sustituir la tipografía de la notación

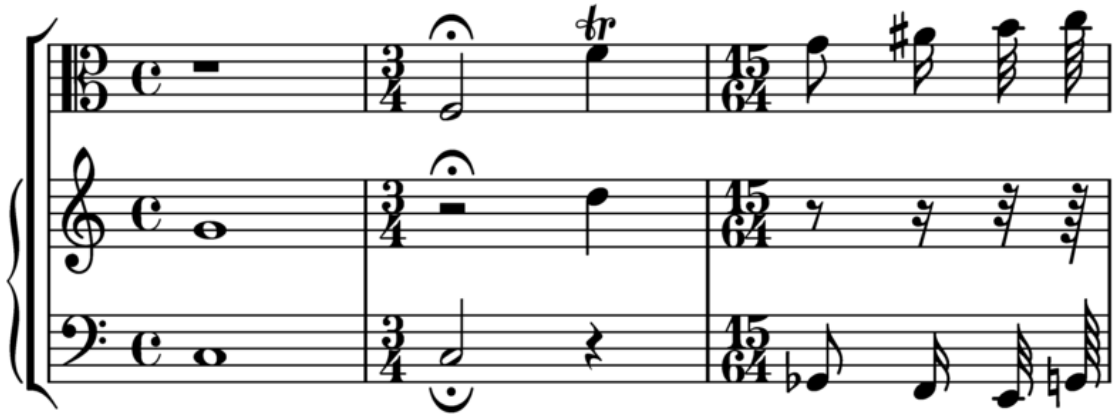
Gonville es un conjunto de glifos alternativo a la tipografía Feta (que es parte de la tipografía Emmentaler que se usa en LilyPond). Se puede descargar de:

`http://www.chiark.greenend.org.uk/~sgtatham/gonville/` (`http://www.chiark.greenend.org.uk/~sgtatham/gonville/`)

Aquí presentamos algunos compases de muestra tipografiados con la fuente Gonville:



Aquí hay unos compases de muestra tipografiados con los glifos Feta de LilyPond:



Instrucciones de instalación

- Descargue y extraiga los archivos de fuente tipográfica.
- Copie[1] los archivos

```
gonville-11.otf
gonville-13.otf
gonville-14.otf
gonville-16.otf
gonville-18.otf
gonville-20.otf
gonville-23.otf
gonville-26.otf
gonville-brace.otf
```

  al directorio `.../share/lilypond/X.Y.Z/fonts/otf`.
- Si tiene archivos `gonville-*.svg`, cópielos al directorio `.../share/lilypond/X.Y.Z/fonts/svg`.

Para ver más información, consulte Sección "Other sources of information" en *Manual de Aprendizaje*.

Nota: los archivos `gonville-*.otf` son para el back-end `ps` y `cairo` (para las salidas PDF y PostScript, así como para todos los formatos de salida cuando se usa el back-end Cairo). Los archivos `gonville-*.svg` son para el back-end `svg`. Para más información, véase Sección "Advanced command-line options for LilyPond" en *Utilización del Programa*.

El código siguiente modifica la fuente de notación a Gonville.

```
\paper {
  property-defaults.fonts.music = "gonville"
}
```

Para más información, véase Sección 8.4 [Cambiar las fuentes tipográficas], página 345.

Véase también

Manual de aprendizaje: Sección "Otras fuentes de información" en *Manual de Aprendizaje*.

Referencia de la notación: Sección B.8 [La tipografía Emmentaler], página 912, Sección 8.4 [Cambiar las fuentes tipográficas], página 345.

[1] Por el momento es necesario repetir estos pasos después de instalar cada nueva versión de LilyPond. Si está ejecutando el archivo binario `lilypond` directamente a partir del directorio de compilación, consulte Sección "Replacing the notation fonts in development versions" en *Guía del colaborador* para ver más información.

Advertencias y problemas conocidos

Gonville no se puede usar para tipografiar notación de 'Música Antigua' y es probable que cualquier nuevo glifo en versiones posteriores de LilyPond no existan en la familia tipográfica Gonville. Diríjase a la página web del autor para obtener más información sobre éste y otros asuntos, entre ellos el régimen de licencias de Gonville.

Otras fuentes de notación

Si tiene otras fuentes de notación como *fontname*-*.otf y *fontname*-*.svg, puede usarlas de la misma manera que Gonville.

Esto es, copie los archivos *fontname*-*.otf a la carpeta .../share/lilypond/*X.Y.Z*/fonts/otf. Si tiene los archivos *fontname*-*.svg, cópielos a .../share/lilypond/current/fonts/svg o a .../share/lilypond/*X.Y.Z*/fonts/svg.

Nota: por el momento, LilyPond espera que los nombres de archivo tengan los siguientes sufijos, todos los cuales deben estar presentes en las carpetas de instalación arriba mencionadas para que funcionen adecuadamente: -11, -13, -14, -16, -18, -20, -23, -26, -brace. Por ejemplo, emmentaler-11.otf, emmentaler-20.svg, etc.

El código siguiente cambia la fuente tipográfica de la notación a la fuente *tipoletra*.

```
\paper {
  % nombre del archivo de fuente sin el sufijo ni la extensión
  property-defaults.fonts.music = "tipoletra"
}
```

24 Creación de salida MIDI

LilyPond puede producir archivos que siguen el estándar MIDI (Musical Instrument Digital Interface; interfaz digital para instrumentos musicales) y permitir así comprobar la música por el oído (con la ayuda de alguna aplicación o dispositivo que entienda el MIDI). La escucha de la salida MIDI también puede ser de ayuda en la localización de errores tales como notas que se han introducido incorrectamente, alteraciones omitidas y otros casos por el estilo.

Los archivos MIDI no contienen sonido (como los archivos AAC, MP3 o Vorbis) pero requieren software adicional para producir el sonido a partir de ellos.

24.1 Notación contemplada por el MIDI

Se puede usar la notación musical siguiente con las posibilidades predeterminadas de LilyPond para producir salida MIDI:

- Marcas de respiración
- Acordes introducidos como nombres de acorde
- Crescendos y decrescendos sobre más de una nota. El volumen se altera de forma lineal entre los dos extremos
- Indicadores dinámicos desde `ppppp` hasta `fffff`, incluyendo `mp`, `mf` y `sf`
- Letra de las canciones
- Marcadores: llamadas de ensayo, segni, codas y etiquetas de sección
- Microtonos, pero *no* acordes microtonales. Se necesita también un reproductor de MIDI capaz de ejecutar eventos de rueda de tono.
- Altura de las notas
- Ritmo introducido como duraciones de las notas, incluidos los grupos de valoración especial como los tresillos
- Articulaciones 'sencillas'; picado, staccato, acento, marcato y portato
- Cambios de tempo introducidos con la función `\tempo`, including for fractional metronome values
- Ligaduras de unión
- Trémolos que *no* se hayan introducido como un valor del tipo ':[*número*]'

También se pueden controlar el panorama, el balance, la expresión y los efectos de reverberación y chorus mediante el establecimiento de propiedades de contexto, véase Sección 24.8 [Propiedades de contexto para efectos MIDI], página 665.

Si se combina con el script `articulate`, es posible producir la salida por el MIDI de los siguientes elementos adicionales de notación musical:

- Appoggiaturas (apoyaturas). Se les da la mitad de la duración de la nota siguiente (sin tener en cuenta los puntillos). Por ejemplo:

  ```
  \appoggiatura c8 d2.
  ```

  El Do toma la duración de una negra.
- Notas y grupos de adorno (mordentes de una nota, trinos, grupetos circulares, etc.)
- Rallentando, accelerando, ritardando y a tempo
- Ligaduras de expresión y de fraseo
- Tenuto

Véase Sección 24.9 [Enriquecimiento de la salida MIDI], página 666.

24.2 Notación no contemplada en el MIDI

Los siguientes elementos de notación musical no se pueden hacer salir por el MIDI:

- Articulaciones distintas al picado, staccato, acento, marcato y portato
- Crescendos y decrescendos sobre una *sola* nota
- Calderón
- Bajo cifrado
- Glissandos
- Caídas y elevaciones
- Acordes microtonales
- Duraciones introducidas como anotaciones, p.ej. el swing
- Cambios de tempo sin `\tempo` (p.ej., introducidas como anotaciones)
- Trémolos que *se introducen* con la notación de la forma ‘:[*número*]’

24.3 El bloque MIDI

Para crear un archivo de salida MIDI a partir de un archivo de entrada de LilyPond, introduzca un bloque `\midi`, que puede estar vacío, dentro del bloque `\score`.[1]

```
\score {
  ... música ...
  \layout { }
  \midi { }
}
```

> **Nota:** Un bloque `\score` que, además de la música, contiene solamente un bloque `\midi` (o sea, *sin* el bloque `\layout`), solo produce archivos de salida MIDI; no se imprime ninguna notación musical.
>
> Se puede usar un bloque `\midi` en el nivel jerárquico superior para cambiar los ajustes de MIDI de forma global; sin embargo, la generación de un verdadero archivo MIDI solo se produce cuando un bloque `\midi` forma parte de un bloque `\score`.
>
> De manera parecida, un bloque `\layout` en el nivel jerárquico superior afecta a los ajustes de disposición de forma global pero no influye en si se produce una salida impresa o no.

La extensión predeterminada para los archivos de salida (`.midi`) se puede cambiar usando la opción `-dmidi-extension` con la instrucción `lilypond`:

```
lilypond -dmidi-extension=mid Archivo.ly
```

Como alternativa, puede añadirse la siguiente expresión de Scheme antes del comienzo de uno de los bloques `\book`, `\bookpart` o `\score`. Véase Sección 20.5 [Estructura del archivo], página 594.

```
#(ly:set-option 'midi-extension "mid")
```

Véase también

Referencia de la notación: Sección 20.5 [Estructura del archivo], página 594, Sección 21.3 [Crear metadatos en los archivos de salida], página 611.

Archivos instalados: `scm/midi.scm`.

[1] Observe que también existe una instrucción de marcado llamada `\score` que no produce ninguna salida MIDI, incluso si está presente un bloque `\midi`. Véase [Scores within markup], página 865.

Advertencias y problemas conocidos

Están disponibles 15 canales MIDI y un canal adicional (el número 10) para la percusión. Los pentagramas se asignan a los canales por orden, de forma que una partitura que contenga más de 15 pentagramas da como resultado que los pentagramas adicionales comparten (pero no sobreescriben) el mismo canal MIDI. Esto puede ser problemático si los pentagramas que comparten canal tienen establecidas propiedades MIDI, basadas en canal, en mutuo conflicto (como distintos instrumentos MIDI).

24.4 Control de las dinámicas del MIDI

Es posible controlar el volumen MIDI general, el volumen relativo de las indicaciones de matiz dinámico y el volumen relativo de los distintos instrumentos.

Las indicaciones matiz dinámico se traducen automáticamente a niveles de volumen dentro del rango de volúmenes MIDI disponible mientras que los crescendos y decrescendos varían el volumen linealmente entre sus dos extremos. Es posible controlar el volumen relativo de las indicaciones de matiz dinámico, y el volumen general de los diferentes instrumentos.

24.4.1 Indicaciones de matiz dinámico en el MIDI

Solo las marcas de dinámica desde `ppppp` hasta `fffff`, incluidos `mp`, `mf` y `sf` tienen valores asignados. Este valor se aplica a continuación al valor del rango de volumen MIDI general para obtener el volumen final que se incluye en la salida MIDI para esa indicación de matiz dinámico en particular. Las fracciones predeterminadas varían desde 0.25 para el *ppppp* hasta 0.95 para el *fffff*. El conjunto completo de las indicaciones de matiz dinámico pueden verse en el archivo `ly/midi-init.ly`.

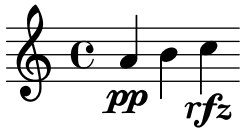
Fragmentos de código seleccionados

Creacaión de matices dinámicos personalizados en el MIDI

El ejemplo siguiente muestra cómo crear una marda de matiz dinámico, no incluida en la lista predeterminada, y asignarle un valor específico de forma que afecte a la salida MIDI.

El matiz `\rfz` conlleva el valor `0.9`.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = "cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Archivos instalados: `ly/script-init.ly` `scm/midi.scm`.

Fragmentos de código: Sección "MIDI" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Dynamic_performer" en *Referencia de Funcionamiento Interno*.

24.4.2 Establecimiento del volumen MIDI

Los volúmenes generales mínimo y máximo de las indicaciones de matiz dinámico MIDI se controlan estableciendo las propiedades `midiMinimumVolume` y `midiMaximumVolume` en el nivel del bloque `Score`. Estas propiedades tienen efecto solamente al comienzo de una vos y sobre las indicaciones dinámicas. La fracción que corresponde a cada indicación dinámica se modifica con esta fórmula:

```
midiMinimumVolume + (midiMaximumVolume - midiMinimumVolume) * fracción
```

En el ejemplo siguiente, el rango dinámico del volumen MIDI general se limita al intervalo entre `0.2` y `0.5`.

```
\score {
  <<
    \new Staff {
      \set Staff.midiInstrument = "flute"
      ... music ...
    }
    \new Staff {
      \set Staff.midiInstrument = "clarinet"
      ... music ...
    }
  >>
  \midi {
    \context {
      \Score
      midiMinimumVolume = 0.2
      midiMaximumVolume = 0.5
    }
  }
}
```

Se puede obtener una ecualización sencilla de instrumentos MIDI estableciendo las propiedades `midiMinimumVolume` y `midiMaximumVolume` dentro del contexto `Staff`.

```
\score {
  \new Staff {
    \set Staff.midiInstrument = "flute"
    \set Staff.midiMinimumVolume = 0.7
    \set Staff.midiMaximumVolume = 0.9
    ... music ...
  }
  \midi { }
}
```

Para partituras con más de un pentagrama y más de un instrumento MIDI, se pueden fijar individualmente los volúmenes relativos de cada instrumento:

```
\score {
  <<
    \new Staff {
      \set Staff.midiInstrument = "flute"
      \set Staff.midiMinimumVolume = 0.7
      \set Staff.midiMaximumVolume = 0.9
      ... music ...
    }
    \new Staff {
      \set Staff.midiInstrument = "clarinet"
      \set Staff.midiMinimumVolume = 0.3
      \set Staff.midiMaximumVolume = 0.6
      ... music ...
    }
  >>
  \midi { }
}
```

En este ejemplo el volumen del clarinete se reduce de forma relativa al volumen de la flauta.

Si no se establecen estas propiedades de volumen, LilyPond aún aplica un 'pequeño grado' de ecualización a ciertos instrumentos. Véase `scm/midi.scm`.

Archivos instalados: `scm/midi.scm`.

Véase también

Referencia de la notación: Capítulo 27 [Disposición de la partitura], página 686.

Referencia de funcionamiento interno: Sección "Dynamic_performer" en *Referencia de Funcionamiento Interno*.

Fragmentos de código seleccionados

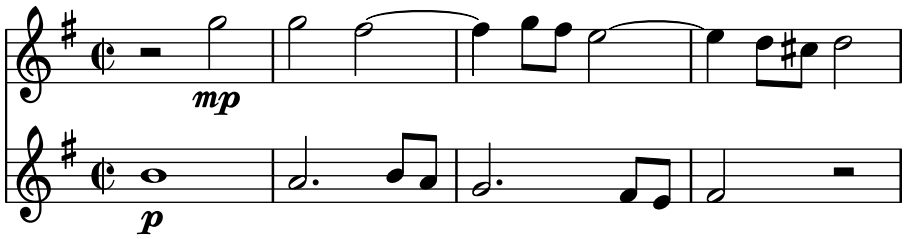
Reemplazo de la ecualización predeterminada de instrumentos MIDI

El ecualizador predeterminado de MIDI se puede sustituir fijando la propiedad `instrumentEqualizer` en el contexto `Score` a un procedimiento de Scheme definido por el usuario que usa un nombre de instrumento MIDI como argumento junto a un par de fracciones que indican los vloúmenes mínimo y máximo, respectivamente, que aplicar a ese instrumento.

El ejemplo siguiente fija los volúmenes mínimo y máximo de la flauta y el clarinete.

```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
```

```
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = "flute"
      \new Voice \relative {
        r2 g''\mp g fis~
        4 g8 fis e2~
        4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = "clarinet"
      \new Voice \relative {
        b'1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {  }
}
```



Advertencias y problemas conocidos

Los cambios en el volumen del MIDI solamente tienen lugar en el comienzo de una nota, por lo que los crescendos y decrescendos no afectan al volumen de una sola nota.

24.4.3 Establecimiento de las propiedades del bloque MIDI

El bloque `\midi` puede contener distintas disposiciones de los contextos, definiciones de contexto nuevas o código que fija los valores de ciertas propiedades.

```
\score {
  ... music ...
  \midi {
    \tempo 4 = 72
  }
}
```

Aquí, el tempo se establece a 72 pulsos de negra por minuto. La indicación de tempo dentro del bloque `\midi` no aparece en la partitura impresa. Aunque cualquier otra indicación de `\tempo` que se especifique dentro del bloque `\score` también se reflejará en la salida MIDI.

Dentro de un bloque `\midi`, la instrucción `\tempo` está estableciendo propiedades durante la interpretación de la música y en el contexto de las definiciones de salida; así, se interpreta *como si* fuera una modificación de contesto.

Las definiciones de contexto siguen la misma sintaxis que las del bloque `\layout`:

```
\score {
  ... music ...
  \midi {
    \context {
      \Voice
      \remove Dynamic_performer
    }
  }
}
```

Este ejemplo suprime el efecto de las indicaciones dinámicas sobre la salida MIDI. Nota: los módulos de traducción de LilyPond usados para el sonido se llaman 'performers'.

Véase también

Manual de aprendizaje: Sección "Otras fuentes de información" en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 3 [Expresiones], página 156, Capítulo 27 [Disposición de la partitura], página 686.

Archivos instalados: `ly/performer-init.ly`.

Fragmentos de código: Sección "MIDI" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "Dynamic_performer" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Ciertos reproductores de MIDI no siempre maneja correctamente los cambios de tempo en la salida MIDI.

Los cambios efectuados al `midiInstrument`, así como ciertas opciones MIDI, al *comienzo* de un pentagrama pueden aparecer dos veces en la salida MIDI.

24.5 Uso de los instrumentos MIDI

Los instrumentos MIDI se establecen usando la propiedad `midiInstrument` dentro de un contexto `Staff`.

```
\score {
  \new Staff {
    \set Staff.midiInstrument = "glockenspiel"
    ... music ...
  }
  \midi { }
}
```

o

```
\score {
  \new Staff \with {midiInstrument = "cello"} {
    ... music ...
  }
  \midi { }
}
```

Si el instrumento elegido no coincide exactamente con uno de los instrumentos de la lista de la sección 'instrumentos MIDI', se usará el instrumento Piano de Cola (`"acoustic grand"`). Véase Sección B.6 [Instrumentos MIDI], página 906.

Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Sección B.6 [Instrumentos MIDI], página 906, Capítulo 27 [Disposición de la partitura], página 686.

Archivos instalados: `scm/midi.scm`.

Advertencias y problemas conocidos

Los instrumentos de percusión cuya notación está en un contexto `DrumStaff` se encontrará a la salida, correctamente, sobre el canal MIDI número 10 pero ciertos instrumentos de percusión afinada como el xilófono, el vibráfono o los timbales, se tratan como instrumentos “normales”, de forma que la música de éstos se debe introducir en un contexto `Staff` (no `DrumStaff`) para obtener una salida MIDI correcta. Hay una lista completa de entradas de los `sets de percusión del canal 10` en el archivo `scm/midi.scm`. Véase Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

24.6 Uso de las repeticiones con el MIDI

Las repeticiones se pueden representar en la salida MIDI aplicando la instrucción `\unfoldRepeats`.

```
\score {
  \unfoldRepeats {
    \repeat tremolo 8 { c'32 e' }
    \repeat percent 2 { c''8 d'' }
    \repeat volta 2 { c'4 d' e' f' }
    \alternative {
      \volta 1 { g' a' a' g' }
      \volta 2 { f' e' d' c' }
    }
  }
  \midi { }
}
```

Para restringir el efecto de `\unfoldRepeats` a la salida MIDI únicamente, y al mismo tiempo genera partituras imprimibles, es necesario hacer *dos* bloques `\score`: uno para el MIDI (con las repeticiones desplegadas) y otro para la notación (con repeticiones de primera y segunda vez, trémolos y del tipo porcentaje);

```
\score {
  ... music ...
  \layout { }
}
\score {
  \unfoldRepeats {
    ... music ...
  }
  \midi { }
}
```

Si se usan varias voces, cada una de ellas debe contener las repeticiones completamente desplegadas para poder obtener una salida MIDI correcta.

Véase también

Referencia de la notación: Capítulo 4 [Repeticiones], página 190.

24.7 Asignación de canales MIDI

Cuando se está generando un archivo MIDI de una partitura, LilyPond asigna cada nota de la partitura automáticamente a un canal MIDI, aquel en que se tocaría cuando se envía hacia un dispositivo MIDI. Un canal MIDI tiene una cierta cantidad de controles disponibles para seleccionar, por ejemplo, el instrumento que usar para reproducir las notas en ese canal, u ordenar al dispositivo MIDI que aplique diversos efectos al sonido producido sobre el canal. En todo momento, cada control de un canal MIDI solo puede tener asignado un valor único (que sin embargo se puede modificar, por ejemplo, para cambiar a otro instrumento en mitad de una partitura).

El estándar MIDI contempla solamente 16 canales por dispositivo MIDI. Este límite en el número de canales limita también el número de instrumentos diferentes que pueden estar tocando al mismo tiempo.

LilyPond crea pistas MIDI separadas para cada pentagrama (o instrumento o voz, según el valor de `Score.midiChannelMapping`), y también para cada contexto de letra. No hay un límite para el número de pistas.

Para solventar el número limitado de canales MIDI, LilyPond contempla algunos modos diferentes para la asignación de canales MIDI, que se seleccionan usando la propiedad de contexto `Score.midiChannelMapping`. En cada caso, si se necesitan más canales MIDI que el límite, los números de canal asignados vuelven a contar desde cero, causando posiblemente una asignación incorrecta de instrumento a algunas notas. Esta propiedad de contexto se puede fijar a uno de los siguientes valores:

`'staff`

Reservar un canal MIDI distinto para cada uno de los pentagramas de la partitura (es el ajuste predeterminado). Todas las notas de todas las voces comparten el canal MIDI del pentagrama que las encierra, y todas se codifican dentro de la misma pista MIDI.

El límite de 16 canales se aplica al número total de contextos de pentagrama y de letra, incluso aunque las letras del MIDI no consumen un canal MIDI.

`'instrument`

Reservar un canal MIDI distinto a cada uno de los instrumentos diferentes especificados en la partitura. Esto significa que todas las notas reproducidas con el mismo instrumento MIDI comparten el mismo canal MIDI (y la misma pista), incluso aunque las notas provengan de diferentes voces o pentagramas.

En este caso los contextos de letra no cuentan para el límite de 16 canales MIDI (ya que no se asignan a un instrumento MIDI), de forma que este ajuste puede permitir una mejor distribución de los canales MIDI cuando el número de contextos de pentagrama y de letra en la partitura excede la cantidad de 16.

`'voice`

Reservar un canal MIDI distinto a cada voz de la partitura que tenga un nombre único entre las voces de su pentagrama circundante. Las voces de los distintos pentagramas siempre se asignan a distintos canales MIDI pero dos voces cualesquiera contenidas dentro del mismo pentagrama comparten el mismo canal MIDI sin tienen el mismo nombre. A causa de que `midiInstrument` y los diversos controles MIDI para los efectos son propiedades del contexto de pentagrama, no se pueden fijar independientemente para cada voz. La primera voz se reproduce con el instrumento y los efectos especificados para el pentagrama, y las voces con un nombre distinto de la primera recibirán el instrumento y los efectos predeterminados.

Nota: se pueden asignar distintos instrumentos y/o efectos a varias voces dentro del mismo pentagrama moviendo el `Staff_performer` del contexto `Staff` al contexto

`Voice`, y dejando que `midiChannelMapping` conserve su valor predeterminado de `'staff` o se fije al valor `'instrument`; véase el fragmento de código siguiente.

Por ejemplo, la asignación predeterminada de canal MIDI de una partitura puede cambiarse al ajuste `'instrument` como se muestra a continuación:

```
\score {
  ...music...
  \midi {
    \context {
      \Score
      midiChannelMapping = #'instrument
    }
  }
}
```

Fragmentos de código seleccionados

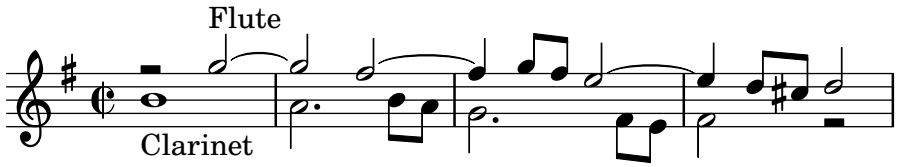
Modificar la salida MIDI para que tenga un canal por cada voz

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por cada puerto MIDI, y la mayoría de los dispositivos solo tiene un puerto.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
```

```
      }
      \context {
        \Voice
        \consists "Staff_performer"
      }
      \tempo 2 = 72
    }
  }
```



24.8 Propiedades de contexto para efectos MIDI

Se pueden usar las siguientes propiedades de contexto para aplicar diversos efectos MIDI a las notas que se reproducen sobre el canal MIDI asociado con el pentagrama, instrumento MIDI o voz actual (dependiendo del valor de la propiedad de contexto `Score.midiChannelMapping` y del contexto en que está situado el `Staff_performer`; véase Sección 24.7 [Asignación de canales MIDI], página 663).

Los cambios en estas propiedades de contexto afectan a todas las notas que se reproducen sobre el canal después del cambio, aunque algunos de los efectos se pueden incluso aplicar también a notas que están ya sonando en ese momento (dependiendo de la implementación del dispositivo de salida MIDI).

Se contemplan las siguientes propiedades de contexto:

`Staff.midiPanPosition`
: La posición de panorama controla cómo se distribuye el sonido de un canal MIDI entre las salidas estéreo izquierda y derecha. La propiedad de contexto acepta un número entre -1.0 (`#LEFT`) y 1.0 (`#RIGHT`); el valor -1.0 pone toda la potencia de sonido en la salida estéreo izquierda (dejando la salida derecha en completo silencio), el valor 0.0 (`#CENTER`) distribuye el sonido por igual entre las salidas estéreo izquierda y derecha, y el valor 1.0 mueve todo el sonido a la salida estéreo derecha. Los valores entre -1.0 y 1.0 se pueden usar para obtener distribuciones mezcladas entre las salidas estéreo izquierda y derecha.

`Staff.midiBalance`
: Equilibrio estéreo de un canal MIDI. De forma similar a la posición de panorama esta propiedad de contexto acepta un número entre -1.0 (`#LEFT`) y 1.0 (`#RIGHT`). Varía el volumen relativo que se envía a los dos altavoces estéreo sin afectar a la distribución de las propias señales estéreo.

`Staff.midiExpression`
: Nivel de expresión (como una fracción del nivel máximo disponible) que se aplica a un canal MIDI. Un dispositivo MIDI combina el nivel de expresión del canal MIDI con el nivel actual de dinámica de una voz (controlado mediante las instrucciones de dinámica como `\p` o `\ff`) para obtener el volumen total de cada nota dentro de la voz. El control de expresión se puede usar, por ejemplo, para implementar efectos de crescendo y decrescendo sobre notas mantenidas (algo que LilyPond no contempla de forma automática).

  El rango de los niveles de expresión se extiende desde 0.0 (ninguna expresión, lo que significa un volumen nulo) hasta 1.0 (expresión al máximo).

`Staff.midiReverbLevel`
: El nivel de reverberación (cono una fracción del máximo nivel disponible) que se aplica a un canal MIDI. Esta propiedad acepta números entre 0.0 (sin reverberación) y 1.0 (efecto máximo).

`Staff.midiChorusLevel`
: Nivel del efecto Chorus (como una fracción del máximo nivel disponible) que se aplica a un canal MIDI. Esta propiedad acepta números entre 0.0 (sin efecto de chorus) y 1.0 (efecto al máximo).

Advertencias y problemas conocidos

Dado que los archivos MIDI no contienen en realidad ninguna información sonora, los cambios en estas propiedades de contexto solamente se convierten en peticiones para que cambien los controles de canal MIDI dentro de los archivos MIDI emitidos. Si un dispositivo MIDI determinado (como un reproductor MIDI por software) puede de hecho manejar cualquiera de estas solicitudes de un archivo MIDI, es algo que depende totalmente de la implementación del mismo: un dispositivo podría optar por ignorar algunas o todas estas solicitudes. Asimismo, la forma en que un dispositivo MIDI interpreta los diferentes valores de estos controles (generalmente, el estándar MIDI determina el comportamiento solamente en los puntos extremos del rango de valores disponibles para cada control), y el hecho de si un cambio en el valor de un control afecta a las notas que ya se están reproduciendo sobre ese canal MIDI o no, también son aspectos específicos de la implementación del dispositivo MIDI.

Al generar archivos MIDI, LilyPond sencillamente convierte de forma lineal los valores fraccionarios dentro de cada rango (7 bits, o 14 bits para los controles de canal MIDI que admiten resolución fina) en valores enteros dentro del rango correspondiente (de 0 a 127 o de 0 a 16383, respectivamente), redondeando los valores fraccionarios hacia el entero más próximo en dirección opuesta al cero. Los valores enteros convertidos se almacenan tal cual en el archivo MIDI generado. Consulte la documentación de su dispositivo MIDI para ver información acerca de cómo interpreta estos valores.

24.9 Enriquecimiento de la salida MIDI

La salida MIDI predeterminada es muy básica, pero se puede mejorar mediante el establecimiento de instrumentos MIDI y varias propiedades del bloque `\midi`.

Algunos guiones adicionales permiten el ajuste fino de la forma en que se trasladan al MIDI los matices dinámicos, las articulaciones y el ritmo: el guion `articulate` y el guion `swing`.

24.9.1 El script `articulate`

Para utilizar el script Articulate, tenemos que incluirlo al principio de nuestro archivo de entrada:

```
\include "articulate.ly"
```

El script crea una salida MIDI en que las notas tienen una duración ajustada en el tiempo de forma que se correspondan con una serie de indicaciones de articulación y de tempo. Sin embargo, la salida tipografiada también se altera para reflejar literalmente el contenido de la salida MIDI.

```
\score {
  \articulate <<
    ... music ...
  >>
  \midi { }
}
```

La instrucción `\articulate` habilita el procesado de las abreviaturas como trinos y mordentes de varias notas. En el propio script puede verse una lista completa de los elementos que están previstos. Véase `ly/articulate.ly`.

Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 27 [Disposición de la partitura], página 686.

Archivos instalados: `ly/articulate.ly`.

> **Nota:** El script `articulate` puede acortar los acordes, lo que podría no ser adecuado para algunos tipos de instrumento, como el órgano. Las notas que no tienen ninguna articulación también se pueden acortar; así, para contener el alcance de este efecto limite el uso de de la función `\articulate` a segmentos de música más breves o modifique los valores de las variables definidas dentro del script `articulate` para compensar el efecto de acortamiento de las notas.

24.9.2 El script `swing`

El script o guion `swing` aporta funciones adicionales que permiten tocar duraciones regulares con un ritmo desigual. El ejemplo más obvio es la interpretación con ‘swing’ que podemos encontrar habitualmente en la música de jazz donde las corcheas de subdivisión binaria se tocan de forma ternaria; sin embargo tmbién se contemplan otros tipos de interpretación.

Este guion se tiene que incluir con la instrucción `\include` al comienzo del archivo de entrada:

```
\include "swing.ly"
```

Se proporcionan tres instrucciones:

- `\tripletFeel` crea un swing de tipo tresillo. Toma dos argumentos: las duraciones que deben quedar afectadas por él (normalmente `8` para las corcheas), y después la expresión musical a la que se debe aplicar.
- `\applySwing` toma un argumento adicional previo a la expresión musical: una ‘lista de pesos’ de *n* relaciones numéricas que expresan la forma en que se deben tocar las notas normales: por ejemplo, `#'(2 1)` indica que las notas alternas se deben tocar el doble de largas que la nota siguiente (de hecho, `\tripletFeel` *duration* {*music*} es realmente una forma abreviada de escribir `\applySwing` *duration* `#'(2 1)` {*music*}). Se pueden obtener corcheas con un swing más suave con una lista de pesos de `#'(3 2)`, u otros valores según el gusto de cada uno.

  Esta lista puede incluir más de dos valores, lo que permite patrones de alteración del ritmo más largos y sofisticados; por ejemplo, un ritmo tipo samba para las semicorcheas se puede obtener como sigue:

```
\score {
  \applySwing 16 #'(3 2 2 3) {
    ... music ...
  }
  \midi { }
}
```

- `\applySwingWithOffset` añade un argumento más entre la ‘lista de pesos’ y la expresión musical: una longitud de desplazamiento. Esta instrucción se debe usar cuando la expresión musical tiene que comenzar fuera del pulso, con un ciclo de swing parcial.

Nota: Igual que con el script `articulate`, todas las instrucciones de swing se trasladan también a la salida impresa, lo que da como resultado un espaciado irregular de las notas. Esto se puede evitar usándolas solamente dentro de un bloque `\score` dedicado a la salida MIDI, en lugar de a la música impresa.

Hay más ayuda e información incluida en el propio archivo del guion: véase `ly/swing.ly`.

Véase también

Manual de aprendizaje: Sección "Other sources of information" en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 2 [Duraciones], página 53.

Archivos instalados: `ly/swing.ly`.

Advertencias y problemas conocidos

- Las construcciones `\repeat` dentro de la música (incluso `\repeat unfold`) no se tienen en cuenta para la determinación de los tiempos de las notas. Esto da lugar a problemas a no ser que las duraciones de todas las partes repetidas sean múltiplos enteros de la duración del ciclo de swing.
- Estas funciones ignoran por completo las indicaciones de compás y los compases. Es la razón de que se tengan que aportar desplazamientos suando `\applySwingWithOffset` si la música comienza fuera del pulso.
- Las notas de adorno se ignoran y simplemente se dejan sin afectar; lo mismo para los grupos de valoración especial como los tresillos.

25 Extraer información musical

Además de crear un resultado visual y MIDI, LilyPond es capaz de presentar información musical en forma de texto.

25.1 Mostrar la notación de LilyPond

La impresión textual de una expresión musical en notación de LilyPond puede hacerse con la función musical `\displayLilyMusic`. Para ver el resultado, lo más usual es llamar a LilyPond mediante la línea de órdenes. Por ejemplo,

```
{
  \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

imprime lo siguiente:

```
{ a,4 cis4 e4 fis4 g4 }
```

De forma predeterminada, LilyPond imprime estos mensajes en la consola junto al resto de los mensajes de la compilación de LilyPond. Para discernir entre estos mensajes y guardar el resultado de `\displayLilyMusic`, redireccione la salida hacia un archivo.

```
lilypond archivo.ly >resultado.txt
```

Observe que LilyPond no se limita a mostrar la expresión musical, sino que también la interpreta (porque `\displayLilyMusic` la devuelve además de mostrarla). Simplemente inserte `\displayLilyMusic` dentro de la música existente con el objeto de obtener información sobre ella.

Para interpretar y mostrar una sección de música en la consola, y al mismo tiempo suprimirla del archivo de salida, use la instrucción `\void`.

```
{
  \void \displayLilyMusic \transpose c a, { c4 e g a bes }
  c1
}
```

25.2 Impresión de las expresiones musicales de Scheme

Véase Sección “Presentación de las expresiones musicales” en *Extender*.

25.3 Guardar los eventos musicales en un archivo

Los eventos musicales se pueden guardar en un archivo pentagrama a pentagrama mediante la inclusión de un archivo en nuestra partitura principal.

```
\include "event-listener.ly"
```

De esta forma se crean uno o más archivos llamados `FILENAME-STAFFNAME.notes` ó `FILENAME-unnamed-staff.notes` para cada pentagrama. Observe que si tiene más de un pentagrama sin nombre, los eventos de todos los pentagramas se mezclarán entre sí dentro del mismo archivo. El resultado tiene el aspecto siguiente:

```
0.000   note     57       4   p-c 2 12
0.000   dynamic  f
0.250   note     62       4   p-c 7 12
0.500   note     66       8   p-c 9 12
0.625   note     69       8   p-c 14 12
0.750   rest     4
0.750   breathe
```

La sintaxis consiste en una línea delimitada por caracteres de tabulación, con dos campos fijos en cada línea seguidos de parámetros opcionales.

```
tiempo tipo ...parámetros...
```

Esta información se puede leer fácilmente por parte de otros programas como guiones de Python, y pueden ser muy útiles para aquellos investigadores que desean realizar un análisis musical o hacer experimentos de reproducción con LilyPond.

Advertencias y problemas conocidos

No todos los eventos musicales de lilypond están contemplados por `event-listener.ly`. Se pretende que sea una "prueba de concepto" bien realizada. Si algunos eventos que quiere ver no aparecen incluidos, copie `event-listener.ly` en su carpeta de lilypond y modifique el archivo de forma que produzca la información que desea.

Problemas de espaciado

26 Disposición de la página

La disposición global sobre el papel viene determinada por tres factores: el diseño de página, los saltos de línea y el espaciado. Estos factores influyen entre sí. La elección de uno u otro espaciado determina qué densidad termina teniendo cada sistema de música. Esto influye en el lugar en que se eligen los saltos de línea y, en último término, en el número de páginas que tiene una partitura.

Considerado globalmente, este proceso se produce en cuatro fases: en primer lugar se eligen unas distancias elásticas o 'muelles', basados en la duración de las figuras. Se prueban todas las combinaciones de saltos de línea posibles, y para cada una de ellas se calcula una puntuación de 'maldad'. Después se hace una estimación de la altura de cada uno de los posibles sistemas. Finalmente se selecciona una combinación de saltos de página y de línea de tal forma que ni el espaciado horizontal ni el vertical estén demasiado estrechos ni estirados.

Hay dos tipos de bloques que pueden contener ajustes de disposición: `\paper {...}` y `\layout {...}`. El bloque `\paper` contiene ajustes de disposición de la página que se espera sean los mismos para todas las partituras de un libro o parte de libro, como la altura de la hoja o si se imprimen los números de página, etc. Véase Capítulo 26 [Disposición de la página], página 673. El bloque `\layout` contiene ajustes de disposición de la partitura, como el número de sistemas que utilizar, o la separación entre grupos de pentagramas, etc. Véase Capítulo 27 [Disposición de la partitura], página 686.

26.1 El bloque \paper

Los bloques `\paper` pueden aparecer en tres lugares diferentes para formar una jerarquía descendente de bloques `\paper`:

- Al principio del archivo de entrada, antes de cualquier bloque `\book`, `\bookpart`, o `\score`.
- Dentro de un bloque `\book` pero fuera de cualquier bloque `\bookpart` o `\score` que estén dentro de ese libro.
- Dentro de un bloque `\bookpart` pero fuera de cualquier bloque `\score` que esté dentro de esa parte de libro.

No se puede poner un bloque `\paper` dentro de un bloque `\score`.

Los valores de los campos filtran esta jerarquía, persistiendo los valores establecidos más arriba en la jerarquía a no ser que se sobreescriban por algún valor establecido más abajo en la jerarquía.

Pueden aparecer varios bloques `\paper` en cada uno de los niveles, por ejemplo como parte de distintos archivos incluidos con `\include`. Si es el caso, los campos en cada nivel se combinan, tomando precedencia los campos que aparecen más tarde si se producen campos duplicados.

Entre los ajustes que pueden aparecer dentro de un bloque `\paper` están los siguientes:

- la función de Scheme `set-paper-size`,
- variables de `\paper` utilizadas para personalizar la disposición de la página, y
- definiciones de marcado utilizadas para personalizar la disposición de los encabezamientos, pies y títulos.

La función `set-paper-size` se estudia en la sección siguiente, Sección 26.2 [Tamaño del papel y escalado automático], página 674. Las variables de `\paper` que se ocupan de la disposición de la página se estudian en secciones posteriores. Las definiciones de elementos de marcado que se encargan de los encabezamientos, pies y títulos se estudian en Sección 21.2 [Títulos, encabezamientos y pies de página personalizados], página 606.

Casi todas las variables de `\paper` funcionan solamente dentro de un bloque `\paper`. Las pocas que funcionan también en un bloque `\layout` están relacionadas en Sección 27.1 [El bloque `\layout`], página 686.

Excepto cuando se especifica lo contrario, todas las variables de \paper que corresponden a distancias sobre la página se miden en milímetros, a no ser que el usuario especifique unas unidades diferentes. Por ejemplo, la declaración siguiente establece el margen superior top-margin a diez milímetros:

```
\paper {
  top-margin = 10
}
```

Para fijarlo en 0.5 pulgadas, utilice el sufijo de unidades \in:

```
\paper {
  top-margin = 0.5\in
}
```

Los sufijos de unidades que están disponibles son \mm, \cm, \in \pt y \bp. Estas unidades son valores simples para convertir a partir de milímetros; están definidos en ly/paper-defaults-init.ly. Por claridad, al usar milímetros, se suele escribir el sufijo \mm aunque no es necesario técnicamente.

También es posible definir valores de \paper utilizando Scheme. El equivalente de Scheme del ejemplo anterior es:

```
\paper {
  #(define top-margin (* 0.5 in))
}
```

Por último, también se pueden predefinir variables del papel.

```
bigMargin = \paper { top-margin = 10\cm }

\paper {
  \bigMargin
  indent = 0\mm
}
```

Véase también

Referencia de la notación: Sección 26.2 [Tamaño del papel y escalado automático], página 674, Sección 21.2 [Títulos, encabezamientos y pies de página personalizados], página 606, Sección 27.1 [El bloque \layout], página 686.

Archivos instalados: ly/paper-defaults-init.ly.

26.2 Tamaño del papel y escalado automático

26.2.1 Fijar el tamaño del papel

'A4' es el valor predeterminado si no se establece ningún tamaño de página explícitamente. Sin embargo, existen dos funciones que se pueden usar para modificarlo:

```
set-default-paper-size
               #(set-default-paper-size "quarto")
```

que siempre se debe colocar en el ámbito superior, y

```
set-paper-size
               \paper {
                 #(set-paper-size "tabloid")
               }
```

que se debe colocar siempre dentro de un bloque \paper.

Si se usa la función `set-default-paper-size` en el ámbito superior jerárquico, debe ir antes de cualquier bloque `\paper`. `set-default-paper-size` establece el tamaño de todas las páginas, mientras que `set-paper-size` establece solo el tamaño de las páginas a las que se aplica el bloque `\paper`. Por ejemplo, si el bloque `\paper` está al principio del archivo, aplica el tamaño de papel a todas las páginas. Si el bloque `\paper` está dentro de un bloque `\book`, entonces el tamaño del papel solo se aplica a ese libro.

Cuando se usa la función `set-paper-size`, se debe colocar *antes* de cualquier otra función que se utilice dentro del mismo bloque `\paper`. Véase Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

Los tamaños de página están definidos en el archivo `scm/paper.scm`; véase Sección B.5 [Tamaños de página predefinidos], página 904, para ver la lista de todos los tamaños de papel disponibles.

Tanto `set-default-paper-size` como `set-paper-size` también aceptan una pareja de números con comilla como argumento para fijar un tamaño del papel personalizado. Por ejemplo

```
#(set-default-paper-size '(cons (* 100 mm) (* 50 mm)))
```

establece la anchura y altura del papel a 100 mm y 50 mm, respectivamente.

Las unidades posibles son `in` (pulgadas), `cm` (centímetros), `mm` (milímetros), `pt` (puntos) y `bp` (puntos grandes).

Si se añade el símbolo `'landscape` como segundo argumento a la función del tamaño de la página, el papel se gira 90 grados y se adapta la anchura de las líneas de acuerdo a ello.

```
#(set-default-paper-size "a6" 'landscape)
```

Se pueden intercambiar las dimensiones del papel *sin que la impresión resulte girada* (como cuando se imprime en tamaño tarjeta postal, o al crear gráficos para su inclusión en lugar de un documento independiente) añadiendo '`landscape`' después del nombre del propio tamaño del papel:

```
#(set-default-paper-size "a6landscape")
```

Cuando el tamaño del papel termina de forma explícita con '`landscape`' o con '`portrait`', la presencia de un símbolo `'landscape` *solamente* afecta a la orientación de la impresión, no a las dimensiones del papel que se usan para la disposición.

Véase también

Referencia de la notación: Sección 26.2.2 [Escalado automático al tamaño del papel], página 675, Sección B.5 [Tamaños de página predefinidos], página 904.

Archivos instalados: `scm/paper.scm`.

26.2.2 Escalado automático al tamaño del papel

Si se cambia el tamaño del papel con una de las funciones de Scheme (`set-default-paper-size` o `set-paper-size`), los valores de algunas variables de `\paper` se escalan automáticamente al nuevo tamaño. Para sortear el escalado automático para una variable determinada, fije el valor de la variable después de establecer el tamaño del papel. Observe que el escalado automático no se activa mediante el ajuste de las variables `paper-height` o de `paper-width`, incluso aunque `paper-width` puede influir sobre otros valores (esto es distinto al escalado y se estudia más abajo). Las funciones `set-default-paper-size` y `set-paper-size` se describen en Sección 26.2.1 [Fijar el tamaño del papel], página 674.

Las dimensiones verticales afectadas por el escalado automático son `top-margin` y `bottom-margin` (véase Sección 26.3 [Variables de espaciado de `\paper` verticales fijas], página 676). Las dimensiones horizontales afectadas por el escalado automático son `left-margin`, `right-margin`, `inner-margin`, `outer-margin`, `binding-offset`, `indent` y `short-indent` (véase Sección 26.5 [Variables de espaciado de `\paper` horizontales], página 679).

Los valores predeterminados para estas dimensiones están fijados en el archivo `ly/paper-defaults-init.ly`, usando variables internas llamadas `top-margin-default`, `bottom-margin-default`, etc. Estos son los valores que resultan del tamaño predeterminado del papel `a4`. Como referencia, con el papel `a4` la altura `paper-height` es `297\mm` y la anchura `paper-width` es `210\mm`.

Véase también

Referencia de la notación: Sección 26.3 [Variables de espaciado de \paper verticales fijas], página 676, Sección 26.5 [Variables de espaciado de \paper horizontales], página 679.

Archivos de inicio: `ly/paper-defaults-init.ly`, `scm/paper.scm`.

26.3 Variables de espaciado de \paper verticales fijas

> **Nota:** Algunas dimensiones de \paper se escalan automáticamente al tamaño del papel, lo que puede llevar a un comportamiento no esperado. Véase Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

Los valores predeterminados (previos al escalado) están definidos en el archivo `ly/paper-defaults-init.ly`.

`paper-height (altura del papel)`
: Altura de la página, no fijada de forma predeterminada. Observe que el escalado automático de algunas dimensiones verticales no resulta afectado por esto.

`top-margin (margen superior)`
: Margen entre el extremo superior de la página y la parte superior de la zona imprimible. Si se modifica el tamaño del papel, el valor predeterminado de esta dimensión se escala de acuerdo con ello.

`bottom-margin (margen inferior)`
: Margen entre la parte inferior de la zona imprimible y el extremo inferior de la página. Si se modifica el tamaño del papel, se escala de acuerdo con ello el valor predeterminado de esta dimensión.

`ragged-bottom (sin justificar por abajo)`
: Si esto se fija a `#t`, los sistemas se disponen según su espaciado natural, ni comprimido ni ampliado verticalmente para rellenar la página.

`ragged-last-bottom (sin justificar la última por abajo)`
: Si se establece a `#f`, entonces la última página (tanto del documento como de cada sección que se hubiera creado con un bloque `\bookpart`) se justifica verticalmente de la misma forma que las páginas anteriores.

Véase también

Referencia de la notación: Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

Archivos de inicio: `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Advertencias y problemas conocidos

Los títulos (extraídos del bloque `\header`) se tratan como sistemas, así pues `ragged-bottom` y `ragged-last-bottom` aumentan la separación entre los títulos y el primer sistema de la partitura.

Los tamaños de papel definidos explícitamente sobreescriben cualquier ajuste de los márgenes superior o inferior definidos por el usuario.

26.4 Variables de espaciado de \paper verticales flexibles

En casi todos los casos es preferible que las distancias verticales entre ciertos elementos (como márgenes, títulos, sistemas y las distintas partituras) sean flexibles, de manera que se amplíen y compriman adecuadamente dependiendo de la situación. Están disponibles un cierto número de variables del bloque \paper (relacionadas más abajo) para realizar un ajuste fino del comportamiento de estas dimensiones frente a su ampliación o compresión.

Observe que las variables de \paper que se estudian en esta sección no controlan el espaciado de los pentagramas que están dentro de los sistemas individuales. El espaciado dentro de los sistemas se controla por medio de propiedades de grob, con ajustes que se escriben normalmente dentro de un bloque \score o \layout, y no dentro del bloque \paper. Véase Sección 29.1 [Espaciado vertical flexible dentro de los sistemas], página 701.

26.4.1 Estructura de las listas-A de espaciado vertical flexible

Cada una de las variables de espaciado de \paper flexibles es una lista-A (lista asociativa) que contiene cuatro *claves*:

- `basic-distance` (distancia básica): distancia vertical, medida en espacios de pentagrama, entre los *puntos de referencia* de los dos elementos, cuando no resulte ninguna colisión, y no se produzca ninguna ampliación o compresión. El punto de referencia de un elemento de marcado (de título o del nivel jerárquico superior) es su punto más alto, y el punto de referencia de un sistema es el centro vertical del `StaffSymbol` más cercano (incluso si se trata de una línea que no es un pentagrama, como un contexto `Lyrics`). Los valores de `distancia básica` menores de `padding` o de `minimum-distance` no son significativos, porque la distancia resultante nunca será menor que `padding` o que `minimum-distance`.
- `minimum-distance` (distancia mínima): la distancia vertical mínima permitida, medida en espacios de pentagrama, entre los puntos de referencia de los dos elementos, cuando se produce una compresión. Los valores de `minimum-distance` menores de `padding` no son significativos, porque la distancia resultante nunca será menor de `padding`.
- `padding` (relleno): mínimo espacio vertical vacío necesario entre los rectángulos circundantes (o las líneas de horizonte) de los dos elementos, medido en espacios de pentagrama.
- `stretchability` (ampliabilidad): medida sin unidades de la propensión relativa de esta dimensión a ampliarse. Si es cero, la distancia no se ampliará (a no ser que resultasen colisiones). Si es positiva, la significación del valor de ampliabilidad de una dimensión concreta depende solamente de su relación con los valores de `stretchability` de las otras dimensiones. Por ejemplo, si una dimensión tiene el doble de `stretchability` que otra, se ampliará con el doble de facilidad. Los valores deben ser no negativos y finitos. El valor `+inf.0` desencadena un error de programación y se ignora, pero se puede usar `1.0e7` para un muelle casi infinitamente ampliable. Si no se fija un valor, el valor predeterminado se establece a `basic-distance`. Observe que la propensión de la dimensión a *comprimirse* no se puede establecer directamente por el usuario y es igual a (`basic-distance` − `minimum-distance`).

Si una página tiene un margen inferior no justificado, la distancia resultante es la mayor de:

- `basic-distance`,
- `minimum-distance`, y
- `padding` más la menor distancia necesaria para eliminar las colisiones.

Para partituras de varias páginas con la parte inferior no justificada en la última página, ésta usará el mismo espaciado que la página anterior siempre y cuando haya espacio suficiente para ello.

Los métodos específicos para la modificación de las listas-A se estudian en Sección 35.8 [Modificación de las listas-A], página 780. El siguiente ejemplo muestra las dos formas en que pueden modificarse estas listas-A. La primera declaración actualiza un par clave-valor individualmente, y la segunda redefine la variable completamente:

```
\paper {
  system-system-spacing.basic-distance = 8
  score-system-spacing =
    #'((basic-distance . 12)
       (minimum-distance . 6)
       (padding . 1)
       (stretchability . 12))
}
```

26.4.2 Lista de variables de espaciado de \paper verticales flexibles

Los nombres de estas variables siguen el formato *superior*`-`*inferior*`-spacing`, donde *superior* e *inferior* son los elementos que van a ser espaciados. Cada distancia se mide entre los puntos de referencia de los dos elementos (véase más arriba la descripción de la estructura de la lista-A). Observe que en estos nombres de variable, el término 'markup' se refiere tanto a *elementos de marcado de título* (`bookTitleMarkup` o `scoreTitleMarkup`) como a *elementos de marcado del nivel superior* (véase Sección 20.5 [Estructura del archivo], página 594). Todas las distancias se miden en espacios de pentagrama.

Los ajustes predeterminados están definidos en el archivo `ly/paper-defaults-init.ly`.

`markup-system-spacing`
: distancia entre un elemento de marcado (de título o del nivel superior) y el sistema que le sigue.

`score-markup-spacing`
: distancia entre el último sistema de una partitura y el elemento de marcado (de título o del nivel superior) que le sigue.

`score-system-spacing`
: distancia entre el último sistema de una partitura y el primer sistema de la partitura que le sigue, cuando no existe ningún elemento de marcado (de título o del nivel superior) entre ellos.

`system-system-spacing`
: distancia entre dos sistemas dentro de la misma partitura.

`markup-markup-spacing`
: distancia entre dos elementos de marcado (de título o del nivel superior).

`last-bottom-spacing`
: distancia desde el último sistema o elemento de marcado del nivel superior en una página, hasta la parte inferior de la zona imprimible (es decir, el extremo superior del margen inferior).

`top-system-spacing`
: distancia desde la parte superior de la zona imprimible (es decir, el extremo inferior del margen superior) hasta el primer sistema de una página, cuando no hay ningún elemento de marcado (de título o del nivel superior) entre los dos.

`top-markup-spacing`
: distancia desde el extremo superior de la zona imprimible (es decir, el extremo inferior del margen superior) hasta el primer elemento de marcado (de título o del nivel superior) sobre una página, cuando no hay ningún sistema entre los dos.

Véase también

Referencia de la notación: Sección 29.1 [Espaciado vertical flexible dentro de los sistemas], página 701.

Archivos de inicio: `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

26.5 Variables de espaciado de \paper horizontales

> **Nota:** Algunas dimensiones de `\paper` se escalan automáticamente al tamaño del papel, lo que puede dar lugar a un comportamiento distinto al esperado. Véase Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

26.5.1 Variables de \paper para la anchura y los márgenes

Los valores predeterminados (antes del escalado) que no están relacionados aquí se encuentran definidos en el archivo `ly/paper-defaults-init.ly`.

`paper-width (anchura del papel)`
: Anchura de la página, sin fijar de forma predeterminada. Aunque `paper-width` no tiene ningún efecto sobre el escalado automático de algunas dimensiones horizontales, sí influye sobre la variable `line-width`. Si están establecidas las dos variables `paper-width` y `line-width`, entonces también se actualizan `left-margin` y `right-margin`. Véase también `check-consistency`.

`line-width (longitud de la línea)`
: Cuando se especifica dentro de un bloque `\paper`, define las dimensiones horizontales del espacio disponible para las líneas del pentagrama en los sistemas sin sangrado por la izquierda. Si se deja sin especificar, la longitud `line-width` del papel se determina a partir de `(paper-width − left-margin − right-margin)`. Si se especifica la longitud `line-width` del papel, y no se especifican el margen izquierdo `left-margin` ni el margen derecho `right-margin`, entonces los márgenes se actualizan de forma que los sistemas se centran automáticamente sobre la página. Véase también `check-consistency`.

  Los valores de `line-width` para las partituras individuales se pueden especificar dentro de los bloques `\layout` de las partituras. Estos valores controlan la longitud de las líneas producidas partitura a partitura. Si la longitud `line-width` no se especifica para una partitura, su valor predeterminado es el `line-width` del bloque paper. El hecho de fijar la longitud `line-width` de una partitura no tiene efecto sobre los márgenes del papel. Las líneas del pentagrama, de una longitud determinada por el valor `line-width` de la partitura, se alinean por la izquierda dentro del área definida por la longitud `line-width` del papel. Si la longitud `line-width` del papel y de la partitura son iguales, las líneas del pentagrama se extienden exactamente desde el margen izquierdo hasta el margen derecho, pero si la longitud `line-width` de la partitura es mayor que la `line-width` del papel, las líneas del pentagrama se imprimirán encima del espacio reservado al margen derecho.

`left-margin (margen izquierdo)`
: Margen entre el extremo izquierdo de la página y el comienzo de las líneas del pentagrama en los sistemas sin sangrado. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Si no se fija el valor de `left-margin`, y tanto

line-width como right-margin están fijados, entonces se establece el valor de left-margin a (paper-width − line-width − right-margin). Si solamente está fijado line-width, entonces los dos márgenes se establecen a ((paper-width − line-width) / 2), y como consecuencia los sistemas se centran sobre la página. Véase también check-consistency.

right-margin (margen derecho)
: Margen entre el extremo derecho de la página y el final de las líneas del pentagrama en sistemas justificados. Si el tamaño de la página se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Si right-margin está sin fijar, y tanto line-width como left-margin están fijados, el valor de right-margin se establece a (paper-width − line-width − left-margin). Si solamente está fijado line-width, los dos márgenes se establecen a ((paper-width − line-width) / 2), y como consecuencia los sistemas aparecen centrados sobre la página. Véase también check-consistency.

check-consistency (comprobar la consistencia)
: Si esto es verdadero (su valor predeterminado), se imprime una advertencia en caso de que la suma de left-margin, line-width y right-margin no coincida exactamente con el valor de paper-width, y se sustituye cada uno de estos valores (excepto paper-width) con su valor predeterminado (escalado al tamaño del papel si es necesario). Si está fijado a #f, se ignora cualquier inconsistencia y se permite que los sistemas se salgan del borde del papel.

ragged-right (no justificar por la derecha)
: Si está fijado a #t, los sistemas no cubren el ancho de la línea. En lugar de ello, los sistemas finalizan en su longitud horizontal natural. Valor predeterminado: #t para las partituras con un solo sistema, y #f para partituras que tienen dos o más sistemas. Esta variable también se puede establecer dentro de un bloque \layout.

ragged-last (no justificar el último)
: Si está fijado a #t, el último sistema de la partitura no llena toda la anchura de la línea. En lugar de ello, el último sistema finaliza en su longitud horizontal natural. Valor predeterminado: #f. Esta variable también puede ajustarse dentro de un bloque \layout.

Véase también

Referencia de la notación: Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

Archivos de inicio: ly/paper-defaults-init.ly.

Advertencias y problemas conocidos

Los tamaños de papel definidos explícitamente sobreescriben cualquier ajuste de los márgenes izquierdo o derecho definidos por el usuario.

26.5.2 Variables de \paper para el modo de doble cara

Los valores predeterminados (antes del escalado) se definen en ly/paper-defaults-init.ly.

two-sided (dos caras)
: Si tiene el valor #t, utilizar inner-margin, outer-margin y binding-offset para determinar los márgenes dependiendo de si el número de la página es par o impar. Esto sobreescribe a left-margin y a right-margin.

inner-margin (margen interno)
: Margen que todas las páginas tienen en el lado interno si forman parte de un libro. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con two-sided fijado a #t.

`outer-margin (margen externo)`
: Margen que todas las páginas tienen en el lado externo si forman parte de un libro. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado a `#t`.

`binding-offset (margen de encuadernación)`
: Medida en que se aumenta `inner-margin` para asegurar que no se oculta nada a causa de la encuadernación. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado a `#t`.

Véase también

Referencia de la notación: Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

Archivos de inicio: `ly/paper-defaults-init.ly`.

26.5.3 Variables de \paper para desplazamientos y sangrados

Los valores predeterminados (antes del escalado) que no se relacionan aquí están definidos en el archivo `ly/paper-defaults-init.ly`.

`horizontal-shift (desplazamiento horizontal)`
: Medida en que todos los sistemas (incluidos los títulos de cabecera y los separadores de sistemas) se desplazan a la derecha. Predeterminado: `0.0`.

`indent (sangrado)`
: Nivel de sangrado para el primer sistema de una partitura. Si se modifica el tamaño del papel, el valor predeterminado de esta dimensión se escala de acuerdo con ello. El espacio dentro de `line-width` que está disponible para el primer sistema se reduce en esta cantidad. `indent` se puede también especificar dentro de los bloques `\layout` para ajustar los sangrados partitura a partitura.

`short-indent (sangrado reducido)`
: Nivel de sangrado para todos los sistemas de una partitura excepto el primero. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. El espacio dentro de `line-width` que está disponible para los sistemas que no son el primero, se reduce en esta cantidad. También se puede especificar `short-indent` dentro de los bloques `\layout` para ajustar los sangrados reducidos partitura a partitura.

Véase también

Referencia de la notación: Sección 26.2.2 [Escalado automático al tamaño del papel], página 675.

Archivos de inicio: `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

26.6 Otras variables de \paper

26.6.1 Variables de \paper para los saltos de línea

`max-systems-per-page`
: Número máximo de sistemas que se colocan sobre una página. Solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

`min-systems-per-page`
: Número mínimo de sistemas que se colocan sobre una página. Puede causar que las páginas aparezcan atiborradas si el valor es demasiado grande En el momento actual

solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

`systems-per-page`
: Número de sistemas que se deben colocar en cada página. En el momento actual solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

`system-count`
: Número de sistemas que utilizar para una partitura. Valor predeterminado: no fijado. Esta variables también se puede ajustar dentro de un bloque `\layout`.

Véase también

Referencia de la notación: Sección 28.1 [Saltos de línea], página 692.

26.6.2 Variables de \paper para los saltos de página

Los valores predeterminados que no están relacionados aquí, se definen en el archivo `ly/paper-defaults-init.ly`

`page-breaking`
: Algoritmo de saltos de página que utilizar. Se puede elegir entre `ly:minimal-breaking`, `ly:page-turn-breaking`, `ly:one-page-breaking`, `ly:one-line-breaking`, `ly:one-line-auto-height-breaking` y `ly:optimal-breaking`. El predeterminado es `ly:optimal-breaking`.

`page-breaking-system-system-spacing`
: Engaña a la parte del programa encargada de los saltos de página para que piense que `system-system-spacing` está establecido a un valor distinto al que tiene realmente. Por ejemplo, si `page-breaking-system-system-spacing.padding` está establecido a algo que es mucho mayor que `system-system-spacing.padding`, entonces el divisor de páginas pone menos sistemas en cada página. Valor predeterminado: sin ajustar.

`page-count`
: Número de páginas que utilizar en una partitura. Por omisión: no establecido.

`page-spacing-weight`
: Cuando se usa el algoritmo `ly:optimal-breaking` para los saltos de página, LilyPond tiene que hacer varias negociaciones entre los estiramientos horizontal y vertical de forma que el espaciado sea lo más aceptable posible en términos generales. Este parámetro controla la importancia relativa del espacio (vertical) de las páginas y el espaciado (horizontal) de las líneas. Los valores altos hacen que el espaciado de la página tenga más importancia. Predeterminado: `10`.

Las siguientes variables son efectivas solamente cuando `page-breaking` está sijado a `ly:page-turn-breaking`. Los saltos de página se seleccionan entonces de forma que se minimice el número de pasos de página. Dado que los pasos de página son necesarios cuando se pasa de una página impar a una página par, normalmente resultará favorecida aquella disposición en que la última página sea impar. Los lugares en que se prefieran pasos de página pueden indicarse manualmente insertando `\allowPageTurn` o automáticamente incluyendo el grabador `Page_turn_engraver` (véase Sección 28.2.7 [Paso de página óptimo], página 698).

Si no hay un número suficiente de posibilidades para escoger a la hora de insertar unos pasos de página adecuados, LilyPond puede insertar una página vacía dentro de una partitura, entre partituras (si hay dos o más partituras) o finalizando una partitura en una página de numeración par. Los valores de las tres siguientes variables pueden incrementarse para que estas acciones se vuelvan menos probables.

Los valores son penalizaciones, es decir, cuando más alto es el valor menos probable será la acción asociada relativa a otras posibilidades.

`blank-page-penalty (penalización por página vacía)`
: Penalización por tener una página vacía en mitad de una partitura. Si `blank-page-penalty` es grande y está seleccionado `ly:page-turn-breaking`, entonces será menos probable que LilyPond inserte una página en medio de una partitura. En lugar de ello, intercala espacio en la música hasta que se rellene la página vacía y la siguiente. Predeterminado: 5.

`blank-last-page-penalty (penalización por última página vacía)`
: Penalización por terminar la partitura sobre una página par. Si `blank-last-page-penalty` es grande y está seleccionado `ly:page-turn-breaking`, entonces será menos probable que LilyPond produzca una partitura en que la última página tenga numeración par. En su lugar, ajustará el espaciado de manera que use una página más o una página menos. Predeterminado: 0.

`blank-after-score-page-penalty (penalización por página vacía después de una partitura)`
: Penalización por tener una página vacía después del final de una partitura y antes de la siguiente. De forma predeterminada, esta penalización es más pequeña que `blank-page-penalty`, de forma que se insertan páginas vacías después de las partituras como preferencia sobre la inserción de páginas vacías dentro de una partitura. Predeterminado: 2.

Véase también

Referencia de la notación: Sección 28.2 [Saltos de página], página 696, Sección 28.2.2 [Saltos de página óptimos], página 697, Sección 28.2.7 [Paso de página óptimo], página 698, Sección 28.2.3 [Saltos de página mínimos], página 697, Sección 28.2.4 [Saltos de página del tipo página única], página 698, Sección 28.2.5 [Saltos de página de línea única], página 698, Sección 28.2.6 [Saltos de página del tipo línea única con altura automática], página 698.

Archivos de inicio: `ly/paper-defaults-init.ly`.

26.6.3 Variables de \paper para la numeración de las páginas

Los valores predeterminados que no se relacionan aquí están definidos en el archivo `ly/paper-defaults-init.ly`

`auto-first-page-number`
: El algoritmo de división de páginas está afectado por el hecho de que el número de la primera página sea par o impar. Si está establecido a `#t`, el algoritmo de división de páginas decide si comenzar con un número par o impar. Esto hace que el número de la primera página se quede como está, o que se aumente en una unidad. Predeterminado: `#f`.

`first-page-number`
: Valor del número de página en la primera página.

`print-first-page-number`
: Si está fijado a `#t`, se imprime un número de página en la primera página.

`print-page-number`
: Si está fijado a `#f`, no se imprimen los números de página.

`page-number-type`
: Tipo de numeración para los números de las páginas. Se puede elegir entre `'arabic`, `'roman-ij-lower`, `'roman-ij-upper`, `'roman-lower` y `'roman-upper`. Por omisión: `'arabic`.

bookpart-level-page-numbering
: Si está fijado a #t, cada parte de libro tiene su propia secuencia independiente de números de página, empezando por first-page-number (por omisión 1).

  Esto también se puede usar para una sola parte de libro. El escenario típico es numerar las páginas de la primera parte de libro independientemente y en números romanos, como se haría en la introducción analítica de la obra que se está publicando.

```
\book {
  \bookpart {
    \paper {
      bookpart-level-page-numbering = ##t
      page-number-type = #'roman-lower
    }
    \markuplist \wordwrap-lines {
      Lorem ipsum dolor sit amet.
    }
  }
  \bookpart {
    ...
  }
}
```

Véase también

Archivos de inicio: ly/paper-defaults-init.ly.

Advertencias y problemas conocidos

Los números de página impares siempre están a la derecha. Si quiere que la música comience en la página 1 debe haber una página en blanco en el reverso de la página de cubierta de manera que la página 1 esté en el lado derecho.

26.6.4 Variables de \paper relativas a los encabezamientos y marcados

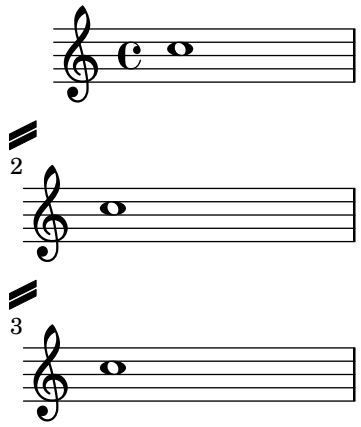
print-all-headers
: Si está fijado a #t, imprime todos los encabezamientos para cada una de las partituras \score de la salida. Normalmente solo se imprimen las variables de encabezamiento piece y opus. Para ver casos de uso consulte Capítulo 21 [Títulos y encabezamientos], página 598. Valor predeterminado: #f.

reset-footnotes-on-new-page
: Si está establecido a #t, los números de las notas al pie se reinician después de cada salto de página. Si queremos notas al pie numeradas consecutivamente por encima de los saltos de página, hay que establecerlo a #f. Valor por omisión: #t.

system-separator-markup
: Objeto de marcado que se inserta entre los sistemas. Se suele usar para partituras orquestales. Predeterminado: sin establecer. El elemento de marcado \slashSeparator, definido en el archivo ly/titling-init.ly, se aporta como un valor predeterminado adecuado, por ejemplo:

```
#(set-default-paper-size "a8")

\book {
  \paper {
    system-separator-markup = \slashSeparator
  }
```

```
  \header {
    tagline = ##f
  }
  \score {
    \relative { c''1 \break c1 \break c1 }
  }
}
```



`footnote-separator-markup`
: Un objeto de marcado que se inserta por encima de los textos de la nota al pie en la parte inferior de la página. Por omisión: una línea horizontal centrada, definida en `ly/paper-defaults-init.ly`.

Véase también

Archivos de inicio: `ly/titling-init.ly`, `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Advertencias y problemas conocidos

El encabezamiento de página predeterminado pone el número de página y el campo `instrument` del bloque `\header` sobre la misma línea.

26.6.5 Variables de \paper para la depuración

Las variables `debug-beam-scoring`, `debug-slur-scoring` y `debug-tie-scoring` permiten imprimir información de depuración para las puntuaciones de las barras y las ligaduras de unión y de expresión. Consulte Sección “Debugging scoring algorithms” en *Guía del colaborador* para ver una explicación detallada de qué hacen esas variables.

27 Disposición de la partitura

Esta sección trata de las opciones de disposición de la partitura para el bloque `\layout`.

27.1 El bloque `\layout`

Así como el bloque `\paper` contiene ajustes relativos al formateo de la página para todo el documento, el bloque `\layout` contiene ajustes para la disposición específica de las partituras. Para fijar las opciones de disposición de partituras de forma global, escríbalas en un bloque `\layout` del nivel jerárquico superior. Para fijar opciones de disposición para una partitura individual escríbalas dentro de un bloque `\layout` dentro del bloque `\score`, después de la música. Entre los ajustes que pueden aparecer en un bloque `\layout` se encuentran los siguientes:

- la función de Scheme `layout-set-staff-size`,
- las modificaciones de contexto de los bloques `\context`, y
- las variables de `\paper` que afectan a la disposición de la partitura.

La función `layout-set-staff-size` se estudia en la sección siguiente, Sección 27.2 [Establecer el tamaño del pentagrama], página 688. Las modificaciones de contexto se estudian en un capítulo distinto; véase Sección 33.4 [Modificar los complementos (plug-ins) de contexto], página 752, y Sección 33.5 [Cambiar los valores por omisión de los contextos], página 753.

Las variables de `\paper` que pueden aparecer dentro de un bloque `\layout`, son como sigue.

- `line-width`, `ragged-right` y `ragged-last` (véase Sección 26.5.1 [Variables de \paper para la anchura y los márgenes], página 679)
- `indent` y `short-indent` (véase Sección 26.5.3 [Variables de \paper para desplazamientos y sangrados], página 681)
- `system-count` (véase Sección 26.6.1 [Variables de \paper para los saltos de línea], página 681)

Los valores predeterminados para las variables anteriores se toman del bloque `\paper`.

He aquí un ejemplo de un bloque `\layout`:

```
\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGrouper.staff-staff-spacing.basic-distance = 8
  }
  \context {
    \Voice
    \override TextScript.padding = 1
    \override Glissando.thickness = 3
  }
}
```

Se pueden escribir varios bloques `\layout` como expresiones del nivel superior. Esto puede ser útil, por ejemplo, si hay distintos ajustes almacenados en diversos archivos y se incluyen opcionalmente. Internamente, cuando se evalúa un bloque `\layout`, se hace una copia de la configuración de `\layout` actual, y después se aplican los cambios definidos en el bloque; el resultado se guarda como la nueva configuración actual. Desde el punto de vista del usuario, los bloques `\layout` se combinan, pero en situaciones de conflicto (cuando se cambia la misma propiedad en diferentes bloques) las definiciones más tardías toman precedencia.

Por ejemplo: si este bloque:

```
\layout {
  \context {
```

```
    \Voice
    \override TextScript.color = #magenta
    \override Glissando.thickness = 1.5
  }
}
```

se coloca después del del ejemplo anterior, las sobreescrituras de `padding` y de `color` para `TextScript` se combinan, pero la última sobreescritura `thickness` para `Glissando` sustituye (u oculta) a la anterior.

Los bloques `\layout` se pueden asignar a variables para una reutilización posterior, pero la forma en que esto funciona es ligera y significativamente diferente del hecho de escribirlas literalmente.

Si se define una variable de la siguiente manera

```
variable_de_Layout = \layout {
  \context {
    \Voice
    \override NoteHead.font-size = 4
  }
}
```

guarda la configuración actual de `\layout` con la sobreescritura añadida de `NoteHead.font-size`, pero esta combinación *no* se guarda como la nueva configuración actual. Tenga en cuenta que la 'configuración actual' se lee cuando la variable se define y no cuando se usa, de manera que el contenido de la variable depende de su posición dentro del código fuente.

La variable se puede entonces usar dentro de otro bloque `\layout`, por ejemplo:

```
\layout {
  \variable_de_layout
  \context {
    \Voice
    \override NoteHead.color = #red
  }
}
```

Un bloque `\layout` que contiene una variable, como en el ejemplo anterior, *no* copia la configuración actual, sino que en lugar de ello usa el contenido de `\variable_de_layout` como la configuración base para adiciones posteriores. Esto significa que cualquier cambio definido entre la definición y el uso de la variable, se pierde.

Si la `variable_de_layout` se define (o se incluye con `\include`) inmediatamente antes de que se utilice, su contenido es sencillamente la configuración en curso más las sobreescrituras que se hayan definido dentro de él. Así pues, en el ejemplo anterior que muestra el uso de `\variable_de_layout` el último bloque `\layout` consistiría en

```
TextScript.padding = 1
TextScript.color = #magenta
Glissando.thickness = 1.5
NoteHead.font-size = 4
NoteHead.color = #red
```

más las sobreescrituras de `indent` y de `StaffGrouper`.

Pero si la variable ya hubiera sido definida antes del primer bloque `\layout` la configuración actual contendría ahora solamente

```
  NoteHead.font-size = 4  % escrita en la definición de la variable
  NoteHead.color = #red   % añadida después del uso de la variable
```

Si se planifica cuidadosamente, las variables de `\layout` pueden ser una herramienta valiosa para estructurar el diseño de la disposición de las fuentes, y también para reiniciar la configuración de `\layout` a un estado conocido.

Véase también

Referencia de la notación: Sección 33.5 [Cambiar los valores por omisión de los contextos], página 753.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

27.2 Establecer el tamaño del pentagrama

El *tamaño de pentagrama* predeterminado es de 20 puntos, que corresponde a una altura del pentagrama de 7.03 mm (un punto es 100/7227 de pulgada, o 2540/7227 mm). El tamaño del pentagrama se puede modificar de tres formas:

1. Para establecer globalmente el tamaño del pentagrama para todas las partituras de un archivo, utilice `set-global-staff-size`:

   ```
   #(set-global-staff-size 14)
   ```

   El ejemplo de arriba establece el tamaño global predeterminado del pentagrama a 14 pt (4.92 mm) y escala todas las tipografías según corresponda.

   La función también se puede usar para fijar diferentes tamaños del pentagrama para cada bloque `\book`:

   ```
   #(set-global-staff-size 30)
   \book {
     { c' }
   }

   #(set-global-staff-size 10)
   \book {
     { c' }
   }
   ```

2. Para fijar el tamaño del pentagrama para una sola partitura dentro de un libro, use `layout-set-staff-size` dentro del bloque `\layout` de dicha partitura:

   ```
   \score {
     ...
     \layout {
       #(layout-set-staff-size 14)
     }
   }
   ```

3. Para fijar el tamaño del pentagrama para una sola pauta dentro de un sistema, use la instrucción `\magnifyStaff`. Por ejemplo, las partituras de música de cámara editadas de la forma tradicional, con piano, usan habitualmente pautas de 7 mm para el piano, mientras que los otros pentagramas suelen estar entre 3/5 y 5/7 de reducción (entre el 60% y el 71%). Para obtener la proporción de 5/7, escriba:

   ```
   \score {
     <<
       \new Staff \with {
         \magnifyStaff #5/7
       } { ... }
       \new PianoStaff { ... }
     >>
   ```

```
}
```

En caso de que usted sepa qué tamaño de `fontSize` quiere usar, puede emplear la forma siguiente:

```
\score {
  <<
    \new Staff \with {
      \magnifyStaff #(magstep -3)
    } { ... }
    \new PianoStaff { ... }
  >>
}
```

Para imitar la apariencia de las ediciones tradicionales, es mejor evitar reducir el grosor de las líneas del pentagrama.

Peso automático de la tipografía a distintos tamaños

La tipografía Emmentaler ofrece el conjunto de glifos musicales *Feta* en ocho tamaños distintos, cada uno de ellos ajustado a un tamaño de pentagrama. Cuanto más pequeño es el tamaño de los glifos, más "pesados" se vuelven, de manera que armonicen con las líneas del pentagrama, que son también relativamente más gruesas. Los tamaños de glifo recomendados se relacionan en la tabla siguiente:

nombre de la fuente	**altura del pentagrama (pt)**	**altura del pentagrama (mm)**	**usos**
feta11	11.22	3.9	partituras de bolsillo
feta13	12.60	4.4	
feta14	14.14	5.0	
feta16	15.87	5.6	
feta18	17.82	6.3	cancioneros
feta20	20	7.0	particellas estándar
feta23	22.45	7.9	
feta26	25.2	8.9	

Véase también

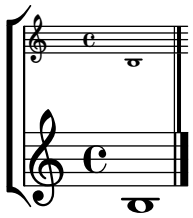
Referencia de la notación: Sección 7.1.1 [Seleccionar el tamaño de la tipografía para la notación], página 280, Sección B.8 [La tipografía Emmentaler], página 912.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

Advertencias y problemas conocidos

Si se usa `\magnifyStaff` solo para algunos pentagramas de un `StaffGroup`, los objetos gráficos `BarLine` ya no se alinean, a causa de las propiedades de `BarLine` alteradas `thick-thickness`, `hair-thickness` y `kern`.

```
\new StaffGroup
  <<
    \new Staff \with { \magnifyStaff #1/2 } { b1 \bar "|." }
    \new Staff { b }
  >>
```

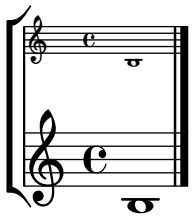


Podemos cancelar el agrandamiento de los objetos gráficos BarLine, imitarlos en otros pentagramas o aplicar valores intermedios para cada Staff.

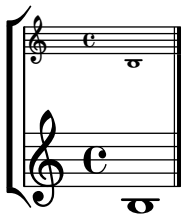
```
#(define bar-line-props
  '((BarLine thick-thickness)
    (BarLine hair-thickness)
    (BarLine kern)))

mus = { b1 \bar "|."}

\markup "Cancel \\magnifyStaff for bar lines:"
\new StaffGroup
  <<
    \new Staff
      \with {
        \magnifyStaff #1/2
        #(revert-props 'magnifyStaff 0 bar-line-props)
      }
      \mus
    \new Staff
      \mus
  >>

\markup "Mimic \\magnifyStaff on other staves:"
\new StaffGroup
  <<
    \new Staff
      \with { \magnifyStaff #1/2 }
      \mus
    \new Staff
      \with {
        #(scale-props 'magnifyStaff 1/2 #t bar-line-props)
      }
      \mus
  >>

\markup "Apply an intermediate value to all staves:"
\new StaffGroup
  <<
    \new Staff
      \with {
        \magnifyStaff #1/2
        #(scale-props 'magnifyStaff 3/2 #t bar-line-props)
      }
      \mus
    \new Staff
```

```
    \with {
      #(scale-props 'magnifyStaff 3/4 #t bar-line-props)
    }
    \mus
>>
```

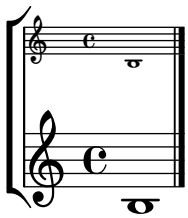
Cancel \magnifyStaff for bar lines:



Mimic \magnifyStaff on other staves:



Apply an intermediate value to all staves:



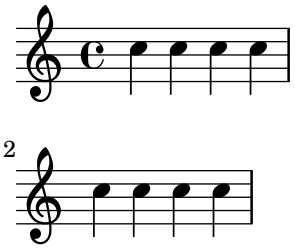
28 Saltos

28.1 Saltos de línea

Normalmente los saltos de línea se determinan automáticamente. Se eligen de forma que las líneas no aparezcan demasiado apretadas ni demasiado sueltas, y que las líneas consecutivas tengan una densidad similar.

Para forzar manualmente un salto de línea sobre la línea divisoria, utilice la instrucción `\break`:

```
\relative c'' {
  c4 c c c | \break
  c4 c c c |
}
```

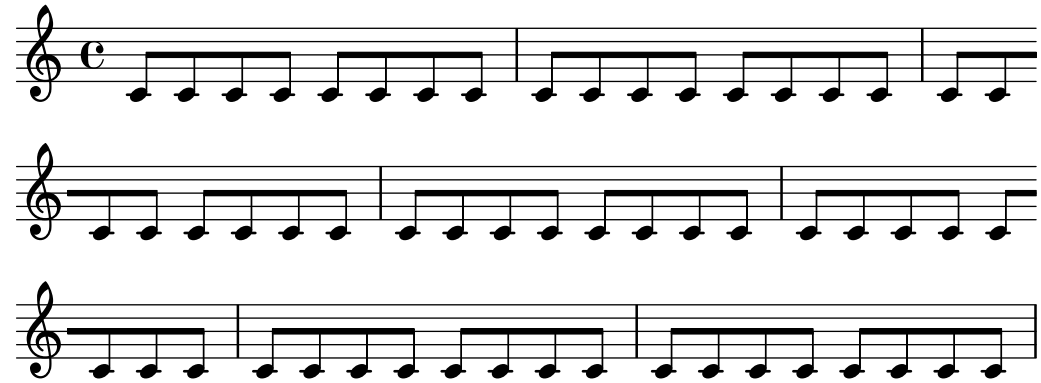


De manera predeterminada, solo se permiten los saltos en las líneas divisorias. También hay otros factores que pueden impedir el salto en una determinada línea divisoria:

- una cabeza de nota o un silencio que continúa más allá de la línea divisoria,
- la presencia de un objeto de extensión 'indivisible', como una barra o un glissando, que cruza la línea divisoria.

La instrucción `\break` fuerza un salto en todo caso, sin que importe si hay una línea divisoria o cualquier otro factor. También es posible puentear todos estos factores usando la instrucción `\allowBreak`. En el ejemplo siguiente, los saltos están permitidos en cualquier sitio, incluso en mitad de un compás, y a pesar de la presencia de barras de corchea.

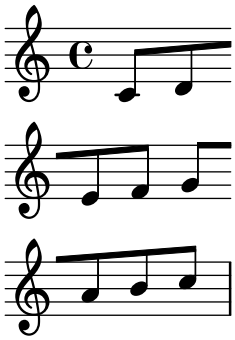
```
\repeat unfold 56 { c'8 \allowBreak }

\paper {
  indent = 0
  line-width = 135\mm
}
```



Si vemos que estamos usando muchas veces `\allowBreak`, podemos impedir que algunos de los factores que hemos mencionado bloqueen la posibilidad de saltos.

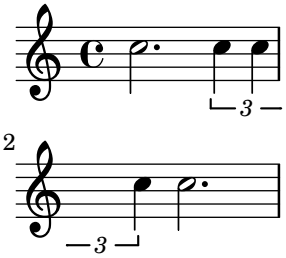
- `Bar_engraver` prohibe los saltos entre barras de compás cuando `forbidBreakBetweenBarLines` está establecido al valor `#t`. Para inhibir esto, fijamos la propiedad al valor `#f`.

```
\fixed c' {
  c8 d e f g a b c'
}

\layout {
  \context {
    \Score
    forbidBreakBetweenBarLines = ##f
  }
}

\paper {
  indent = 0
  line-width = 30\mm
}
```



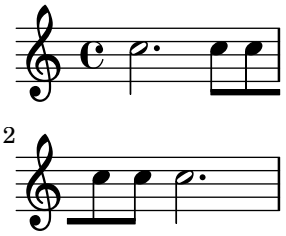
- Se puede hacer que las notas y silencios que se extienden por encima de las líneas divisorias no supriman los saltos, quitando el grabador `Forbid_line_break_engraver` del contexto `Voice`.

```
\new Voice \with {
  \remove Forbid_line_break_engraver
} \relative {
  c''2. \tuplet 3/2 { c4 c c } c2.
}

\paper {
  indent = 0
  line-width = 35\mm
}
```



- La presencia de barras de corchea y otros objetos de extensión indivisibles se ignora si su propiedad `breakable` está estblecida al valor `#t`.

```
\relative c'' {
  \override Beam.breakable = ##t
  c2. c8[ c |
  c8 c] c2. |
}

\paper {
  indent = 0
  line-width = 35\mm
}
```



La instrucción `\noBreak`, evita que se produzca un salto de línea en la barra divisoria en que se inserta.

Dentro de una partitura, se evitan los saltos de línea automáticos dento de la música que se encuentra entre las instrucciones `\autoLineBreaksOff` y `\autoLineBreaksOn`. Si también se quieren evitar los saltos de página automáticos, deben usarse las instrucciones `\autoBreaksOff` y `\autoBreaksOn`. Los saltos manuales no están afectados por estas instrucciones. Observe que el hecho de inhibir los saltos de línea automáticos puede hacer que la música se extienda sobre el margen derecho si no cabe completamente dentro de una sola línea.

Los saltos automáticos de línea (no de página) se pueden habilitar sobre líneas divisorias individuales mediante el uso de `\once \autoLineBreaksOn` en una barra. La instrucción identifica un salto de línea permitido, en lugar de uno prohibido.

Los ajustes más básicos que influyen sobre el espaciado de las líneas son `indent` y `line-width`. Se establecen dentro del bloque `\layout`. Controlan el sangrado de la primera línea de música, y la longitud de las líneas.

Si se establece `ragged-right` a `#t` en el bloque `\layout`, los sistemas terminan en su longitud horizontal natural, en lugar de distribuirse horizontalmente para llenar toda la línea. Esto es de utilidad para fragmentos cortos, y para efectuar una comprobación de lo apretado que es el espaciado natural.

La opción `ragged-last` es similar a `ragged-right`, pero afecta solo a la última línea de la pieza.

```
\layout {
  indent = 0\mm
  line-width = 150\mm
  ragged-last = ##t
}
```

Para saltos de línea a intervalos regulares utilice `\break` separado mediante desplazamientos con `\skip` y repetidos con `\repeat`. Por ejemplo, esto haría que los 28 compases siguientes (suponiendo un compás de 4/4) se dividan cada cuatro compases, y solo en dichos lugares:

```
<<
  \repeat unfold 7 {
```

```
    s1 \noBreak s1 \noBreak
    s1 \noBreak s1 \break
  }
  { la música real... }
>>
```

Instrucciones predefinidas

\break, \allowBreak, \noBreak, \autoBreaksOff, \autoBreaksOn, \autoLineBreaksOff, \autoLineBreaksOn.

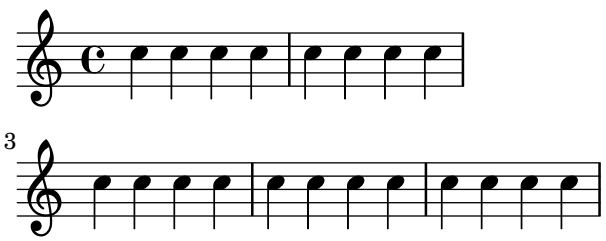
Fragmentos de código seleccionados

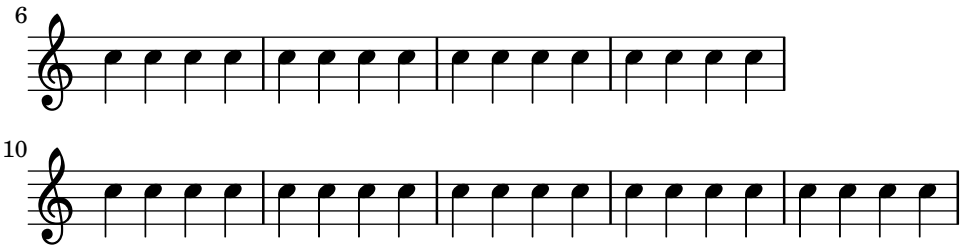
Uso de una voz adicional para los saltos de línea y de página

Con frecuencia es más fácil gestionar la información de los saltos de línea y de página manteniéndola separada de la música, por medio de la introducción de una vos adicional que contiene solamente silencios de separacion junto a las instrucciones \break, \pageBreak y otras informaciones de disposición.

Este patrón se hace especialmente útil cuando se está sobreescribiendo la propiedad line-break-system-details y las otras útiles pero largas propiedades del objeto gráfico NonMusicalPaperColumn.

```
music = \relative c'' { c4 c c c }

\score {
  \new Staff <<
    \new Voice {
      s1*2 \break
      s1*3 \break
      s1*4 \break
      s1*5 \break
    }
    \new Voice {
      \repeat unfold 2 { \music }
      \repeat unfold 3 { \music }
      \repeat unfold 4 { \music }
      \repeat unfold 5 { \music }
    }
  >>
}

\paper {
  indent = 0
  line-width = 140\mm
  ragged-right = ##t
}
```





Véase también

Referencia de la notación: Sección 26.6.1 [Variables de \paper para los saltos de línea], página 681, Sección 27.1 [El bloque \layout], página 686.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "LineBreakEvent" en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

Si se escribe \autoLineBreaksOff o \autoBreaksOff antes de cualquier contenido musical, aparecerán mensajes de error. Escriba siempre estas instrucciones después de algo de música.

28.2 Saltos de página

Esta sección describe los distintos métodos para crear saltos de página, y cómo modificarlos.

28.2.1 Saltos de página manuales

Se puede sobreescribir el mecanismo predeterminado de salto de página insertando instrucciones \pageBreak o \noPageBreak. Estas instrucciones son análogas a \break y \noBreak. Se deben insertar en una línea divisoria. Estas instrucciones fuerzan y prohíben, respectivamente, la eventualidad de un salto de página en dicha línea divisoria. Por supuesto, la instrucción \pageBreak también fuerza un salto de línea.

Las instrucciones \pageBreak y \noPageBreak también se pueden insertar en el nivel más alto, entre las partituras y los elementos de marcado situados en el nivel superior.

Dentro de una partitura, no se producen saltos de página automáticos en el interior de la música que se encuentra entre las instrucciones \autoPageBreaksOff y \autoPageBreaksOn. Los saltos de página manuales no reultan afectados por estas instrucciones.

Hay ajustes análogos a ragged-right y ragged-last que tienen el mismo efecto sobre el espaciado vertical. Si ragged-bottom se establece a #t los sistemas no se justifican verticalmente. Cuando el valor de ragged-last-bottom se fija a #t, como lo está de forma predeterminada, se permite una separación al final de la última página (o la última página de cada uno de las partes de libro \bookpart). Véase Sección 26.3 [Variables de espaciado de \paper verticales fijas], página 676.

Los saltos de página se calculan por medio de la función page-breaking. LilyPond ofrece varios algoritmos para el cómputo de los saltos de página, entre ellos ly:optimal-breaking, ly:page-turn-breaking y ly:minimal-breaking. El predeterminado es ly:optimal-breaking, pero el valor se puede cambiar en el bloque \paper:

```
\paper {
  page-breaking = #ly:page-turn-breaking
}
```

Cuando un libro tiene muchas partituras y páginas, puede ser difícil resolver el problema de los saltos de página, necesitando mucha memoria y prolongados tiempos de procesamiento. Para facilitar el proceso de división en páginas, se usan los bloques \bookpart para dividir el libro en

varias partes: los saltos de página se producen de manera independiente en cada parte. También se pueden usar diferentes funciones de división en páginas para las distintas partes del libro.

```
\bookpart {
  \header {
    subtitle = "Prefacio"
  }
  \paper {
     %% En una parte que consiste en texto principalmente,
     %% puede ser preferible ly:minimal-breaking
     page-breaking = #ly:minimal-breaking
  }
  \markup { ... }
  ...
}
\bookpart {
  %% En esta parte, consistente en música, se usa la función
  %% óptima predeterminada de saltos de página.
  \header {
    subtitle = "Primer movimiento"
  }
  \score { ... }
  ...
}
```

Instrucciones predefinidas

`\pageBreak`, `\noPageBreak`, `\autoPageBreaksOn`, `\autoPageBreaksOff`.

Véase también

Referencia de la notación: Sección 26.6.2 [Variables de `\paper` para los saltos de página], página 682.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Advertencias y problemas conocidos

El prefijo `\once` no es efectivo con `\autoPageBreaksOn` y `\autoPageBreaksOff`. Si se ha desactivado la división automática de páginas y después se vuelve a activar para permitir un salto, debe continuar activada durante algunos compases (cuyo número exacto dependerá de la partitura) antes de volverse a desactivar, pues en caso contrario podría perderse la oportunidad de que el salto se produzca.

28.2.2 Saltos de página óptimos

La función `ly:optimal-breaking` es el método predeterminado de LilyPond para determinar los saltos de página. Intenta hallar una división de páginas que haga mínimos el apretujamiento y la distensión, tanto horizontal como verticalmente. A diferencia de `ly:page-turn-breaking`, no tiene un concepto de los pasos de página.

Véase también

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

28.2.3 Saltos de página mínimos

La función `ly:minimal-breaking` efectúa unos cálculos mínimos para determinar los saltos de página: completa una página con tantos sistemas como sea posible antes de continuar con la

siguiente. Así, puede preferirse para partituras con muchas páginas, donde las otras funciones de salto de página pueden resultar demasiado lentas o ávidas de memoria, o con una gran cantidad de textos. Se habilita utilizando:

```
\paper {
  page-breaking = #ly:minimal-breaking
}
```

Véase también

Snippets: Sección "Espaciado" en *Fragmentos de código*.

28.2.4 Saltos de página del tipo página única

La función `ly:one-page-breaking` (saltos de página única) es un algoritmo de saltos de página de proósito especial que ajusta automáticamente la altura de la página para que quepa toda la música en una sola página. La variable `paper-height` del bloque `paper` se ignora, pero los otros ajustes funcionan de la manera acostumbrada. Concretamente, el espaciado entre el último sistema (o elemento de marcado de nivel superior) y el pie de página se puede personalizar con `last-bottom-spacing` dentro del bloque `paper`. La anchura de la página se queda sin alterar por omisión, pero se puede fijar con la variable `paper-width` del bloque `paper`.

Advertencias y problemas conocidos

`ly:one-page-breaking` no es compatible, de momento, con `\bookpart`.

28.2.5 Saltos de página de línea única

La función `ly:one-line-breaking` es un algoritmo de saltos de página para uso especial que pone cada partitura en una página y en una sola línea. Esta función de salto de página no compone los títulos ni los márgenes; solamente se imprime la partitura.

El ancho de la página se ajusta de forma que la partitura más larga quepa en una línea. Concretamente, las variables `paper-width`, `line-width` y `indent` del bloque `\paper` se ignoran, aunque `left-margin` y `right-margin` aún se respetan. La altura de la página se deja sin cambios.

28.2.6 Saltos de página del tipo línea única con altura automática

La función `ly:one-line-auto-height-breaking` funciona igual que `ly:one-line-breaking` excepto que la altura de la página se modifica automáticamente para hacerse corresponder con la altura de la música. De forma específica, la variable `paper-height` del bloque `\paper` se ajusta de tal manera que abarca los extremos superior e inferior de la partitura más alta más los márgenes superior e inferior `top-margin` y `bottom-margin`.

Observe que el ajuste `top-system-spacing` afecta a la posición vertical de la música. Fíjelo al valor `#f` dentro de un bloque `paper` para disponer la música sencillamente entre los márgenes superior e inferior.

28.2.7 Paso de página óptimo

Con frecuencia es necesario encontrar una configuración de división de páginas de manera que haya un silencio al final de una página de cada dos. De esta forma, el músico puede pasar la página sin perder notas. La función `ly:page-turn-breaking` trata de encontrar una división de páginas que haga mínimos el apretujamiento y el estiramiento, pero con la restricción añadida de que solo se permite introducir vueltas de página en los lugares especificados.

Hay dos etapas en el uso de esta función de división de páginas. En primer lugar debemos habilitarlo en el bloque `\paper`, como se explicó en Sección 28.2 [Saltos de página], página 696. Entonces debemos decirle a la función dónde nos gustaría permitir los saltos de página.

Hay dos formas de conseguir la segunda tarea. Primero, podemos especificar manualmente cada uno de los pasos de página potenciales, insertando `\allowPageTurn` en nuestro archivo de entrada en los lugares adecuados.

Si esto es demasiado tedioso, podemos añadir un grabador `Page_turn_engraver` a un contexto Staff o Voice. El grabador `Page_turn_engraver` analizará el contexto en busca de secciones sin notas (observe que no busca silencios, sino la ausencia de notas. Se hace así para que la polifonía en un solo pentagrama con silencios en una de las voces no arruine la labor del grabador `Page_turn_engraver`). Cuando encuentra una sección sin notas suficientemente larga, el grabador `Page_turn_engraver` inserta un `\allowPageTurn` en la última barra de compás de dicha sección, a no ser que haya una barra 'especial' de compás (como una doble barra), en cuyo caso se insertará el `\allowPageTurn` en la última barra 'especial' de compás de la sección.

El grabador `Page_turn_engraver` lee la propiedad de contexto `pageTurnMinimumRestLength` para determinar qué longitud debe tener una sección sin notas antes de que se considere la posibilidad de un paso de página. El valor predeterminado para `pageTurnMinimumRestLength` es `1`. Si quiere inhabilitar las vueltas de página, fíjelo a algún valor 'muy grande'.

```
\new Staff \with { \consists Page_turn_engraver }
{
  a4 b c d |
  R1 | % aquí se permite un salto de página
  a4 b c d |
  \set Staff.pageTurnMinimumRestLength = #5/2
  R1 | % aquí no se permite un salto de página
  a4 b r2 |
  R1*2 | % aquí se permite un salto de página
  a1
}
```

Al usar repeticiones de primera y segunda vez, el grabador `Page_turn_engraver` solo permite un pase de página durante la repetición si hay suficiente tiempo al principio y al final de la repetición para volver a pasar la página hacia atrás. Si la repetición es demasiado corta, se puede usar el grabador `Page_turn_engraver` para *inhabilitar* los pasos de página mediante el establecimiento de un valor adecuado para la propiedad de contexto `pageTurnMinimumRepeatLength`. En este caso el grabador `Page_turn_engraver` solo permitirá los pases de página en las repeticiones cuya duración sea mayor que el valor especificado.

Las instrucciones de paso de página, `\pageTurn`, `\noPageTurn` y `\allowPageTurn`, se pueden usar también en el nivel más elevado del código, dentro de elementos de marcado del nivel superior y entre las partituras.

Instrucciones predefinidas

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

Véase también

Referencia de la notación: Sección 26.6.1 [Variables de `\paper` para los saltos de línea], página 681.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

Advertencias y problemas conocidos

Utilice un solo grabador `Page_turn_engraver` en cada partitura. Si hay más de uno, se interferirán entre sí.

Véase también

Notation Reference: Capítulo 29 [Espaciado vertical], página 701.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

29 Espaciado vertical

El espaciado vertical está controlado por tres factores: la cantidad de espacio disponible (es decir, el tamaño del papel y los márgenes), la separación entre los sistemas, y la separación entre los pentagramas dentro de un sistema.

29.1 Espaciado vertical flexible dentro de los sistemas

Tres mecanismos distintos controlan el espaciado vertical flexible dentro de los sistemas, uno por cada una de las categorías siguientes:

- *pautas sin agrupar*,
- *pautas agrupadas* (pautas dentro de un grupo de pentagramas como `ChoirStaff`, etc.), y
- *líneas que no son pautas* (como `Lyrics`, `ChordNames`, etc.).

La altura de cada sistema se determina en dos fases. Primero, todos los pentagramas se disponen según la cantidad de espacio vacío disponible. Después, las líneas que no son pautas se distribuyen entre las pautas.

Observe que los mecanismos de espaciado estudiados en esta sección solamente controlan el espaciado vertical de las pautas y líneas que no son pautas dentro de los sistemas individuales. El espaciado vertical entre distintos sistemas, partituras, marcados y márgenes se controla mediante variables de `\paper` que se estudian en Sección 26.4 [Variables de espaciado de \paper verticales flexibles], página 677.

29.1.1 Propiedades de espaciado dentro de los sistemas

Los mecanismos de espaciado vertical dentro del sistema están controlados por dos conjuntos de propiedades de grob. El primer conjunto está asociado con el grob `VerticalAxisGroup`, que se crea por parte de todas las pautas y líneas que no son pautas. El segundo conjunto está asociado con el grob `StaffGrouper`, que puede crearse por parte de los grupos de pentagramas, pero solamente si se le llama explícitamente. Estas propiedades se describen individualmente al final de la presente sección.

Los nombres de estas propiedades (excepto para `staff-affinity`) siguen el formato *`elemento1`*`-`*`elemento2`*`-spacing`, donde *`elemento1`* y *`elemento2`* son los elementos que se van a espaciar. Observe que *`elemento2`* no está necesariamente por debajo de *`elemento1`*; por ejemplo, `nonstaff-relatedstaff-spacing` mide hacia arriba a partir de la línea del tipo no-pauta si `staff-affinity` es `UP`.

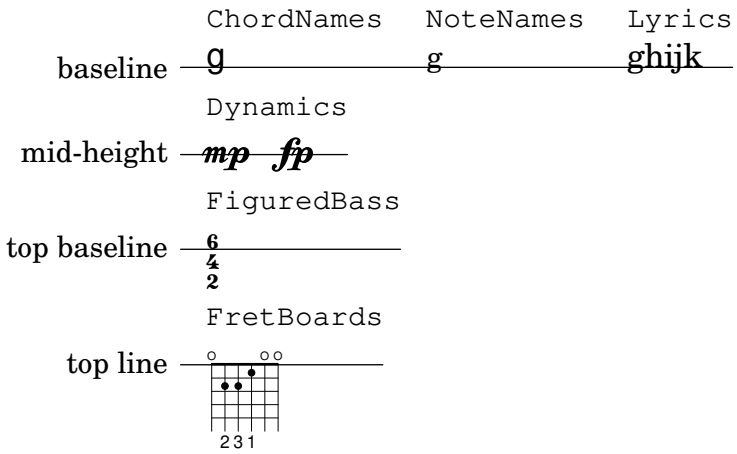
Cada distancia se mide entre los *puntos de referencia*[1] de los dos elementos. El *punto de referencia* para un pentagrama y otro tipo de pauta es el centro vertical de su `StaffSymbol` (es decir, la línea central si `line-count` es un número impar; el espacio central si `line-count` es par). Los puntos de referencia para las líneas individuales que no son pautas aparecen en la tabla siguiente:

Línea que no es una pauta	**Punto de referencia**
`ChordNames`	línea de base
`NoteNames`	línea de base
`Lyrics`	línea de base
`Dynamics`	la mitad de la altura de la 'm'
`FiguredBass`	línea base del elemento más alto

[1] Esta redacción es un poco imprecisa. Para los contextos, las posiciones de los objetos gráficos a lo largo del eje x se calculan por parte del algoritmo de espaciado de LilyPond (usando objetos `PaperColumn` y otros parecidos para la alineación); esto significa que la única información relevante aquí es la posición vertical. En otras palabras, un 'punto' de referencia de un contexto es la coordenada y al que otros objetos gráficos se alinean.

`FretBoards` línea superior

En la siguiente imagen, las líneas horizontales indican las posiciones de estos puntos de referencia:



Cada una de las propiedades de grob del espaciado vertical (excepto `staff-affinity`) usa la misma estructura de lista-A que las variables de espaciado del bloque `\paper` que se han estudiado en Sección 26.4 [Variables de espaciado de `\paper` verticales flexibles], página 677. En Sección 35.8 [Modificación de las listas-A], página 780, se estudian métodos específicos para la modificación de las listas-A. Las propiedades de los grobs deben ajustarse con un `\override` dentro de un bloque `\score` o `\layout`, y no dentro de un bloque `\paper`.

El ejemplo que sigue muestra las dos maneras en que pueden modificarse estas listas-A. La primera declaración actualiza un par clave-valor de forma individual, y la segunda redefine la propiedad completamente:

```
\new Staff \with {
  \override VerticalAxisGroup
            .default-staff-staff-spacing.basic-distance = 10
} { ... }

\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 10)
       (minimum-distance . 9)
       (padding . 1)
       (stretchability . 10))
} { ... }
```

Para modificar globalmente cualquiera de los ajustes de espaciado, póngalos dentro del bloque `\layout`:

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup
              .default-staff-staff-spacing
              .basic-distance = 10
  }
}
```

Los ajustes estándar para las propiedades de grob del espaciado vertical están relacionadas en Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno* y en Sección “Staff-Grouper” en *Referencia de Funcionamiento Interno*. Las sobreescrituras predeterminadas para los

tipos específicos de líneas que no son pautas están relacionadas en las descripciones de contexto correspondientes en Sección “Contexts” en *Referencia de Funcionamiento Interno*.

Propiedades del grob `VerticalAxisGroup`

Las propiedades de `VerticalAxisGroup` se suelen ajustar con una instrucción `\override` en el nivel de `Staff` (o equivalente).

`staff-staff-spacing`
: Se utiliza para determinar la distancia entre la pauta actual y la pauta que está justo por debajo de ella dentro del mismo sistema, incluso si hay más líneas que no son pautas (tales como `Lyrics`) entre las dos pautas. No se aplica a la pauta inferior de un sistema.

  Inicialmente, el valor de espaciado `staff-staff-spacing` de un grupo `VerticalAxisGroup` es una función de Scheme que aplica las propiedades del `StaffGrouper` si la pauta forma parte de un grupo, o el `default-staff-staff-spacing` de la pauta en caso contrario. Esto hace posible que las pautas estén espaciadas de forma diferente cuando están agrupadas. Para conseguir un espaciado regular sin importar el agrupamiento, esta función se puede sustituir por una lista-A de espaciado flexible, utilizando la forma de sobreescritura de redefinición completa que se ha mostrado más arriba. Si solo se especifican algunos valores en una operación de sobreescritura, los valores que faltan se toman de `default-staff-staff-spacing` (si tiene valores para ellos).

`default-staff-staff-spacing`
: Una lista-A de espaciado flexible que define el espacio `staff-staff-spacing` que se usará para las pautas sin agrupar, a no ser que se haya sobreescrito explícitamente `staff-staff-spacing` con una instrucción `\override`.

`staff-affinity`
: Dirección de la pauta que utilizar para el espaciado de la línea actual que no es pauta. Se puede elegir entre `UP` (arriba), `DOWN` (abajo) y `CENTER` (centro). Si es `CENTER`, la línea del tipo no-pauta se coloca equidistante entre las dos pautas contiguas a ambos lados, a no ser que lo impidan colisiones u otras restricciones de espaciado. Las líneas del tipo no-pauta adyacentes han de llevar valores de `staff-affinity` no crecientes desde la parte superior hasta la inferior, p.ej. una línea del tipo no-pauta establecida a `UP` no debería seguir inmediatamente a otra que esté establecida a `DOWN`. Las líneas del tipo no-pauta en la parte superior de un sistema deben usar `DOWN`; las de la parte inferior deben usar `UP`. El ajuste de `staff-affinity` para una pauta hace que se le trate como una línea que no es una pauta. El establecimiento de `staff-affinity` a `#f` para una línea que no es una pauta hace que se trate como una pauta. Establecer `staff-affinity` a `UP`, `CENTER`, o `DOWN` hace que el pentagrama resulte espaciado como si fuera una línea que no es una pauta.

`nonstaff-relatedstaff-spacing`
: Distancia entre la línea del tipo no-pauta actual y la pauta más próxima en la dirección de `staff-affinity`, si no hay ninguna línea que no sea una pauta entre las dos, y `staff-affinity` es `UP` o `DOWN`. Si `staff-affinity` es `CENTER`, entonces se usa `nonstaff-relatedstaff-spacing` para las pautas más próximas *a los dos lados*, incluso aunque aparezcan otras líneas del tipo no-pauta entre la pauta actual y una de las otras. Esto significa que la colocación de una línea que no es una pauta depende tanto de las pautas que la rodean como de las líneas que no son pautas y que la rodean. Establecer la ampliabilidad `stretchability` de uno de estos tipos de espaciado a un valor más pequeño provoca que este espaciado domine.

El establecimiento de `stretchability` a un valor mayor hace que dicho espaciado tenga un menor efecto.

`nonstaff-nonstaff-spacing`
: Distancia entre la línea actual del tipo no-pauta y la siguiente línea del tipo no-pauta en la dirección de `staff-affinity`, si las dos están al mismo lado de la pauta relacionada, y `staff-affinity` es `UP` o `DOWN`.

`nonstaff-unrelatedstaff-spacing`
: Distancia entre la línea actual del tipo no-pauta y la pauta en la dirección opuesta de `staff-affinity`, si no hay ninguna otra línea del tipo no-pauta entre las dos, y `staff-affinity` es `UP` o `DOWN`. Se puede usar, por ejemplo, para requerir una cantidad de relleno mínima entre una línea de `Lyrics` y la pauta a la que no pertenece.

Propiedades del grob `StaffGrouper`

Las propiedades de `StaffGrouper` se suelen ajustar con una instrucción `\override` en el nivel de `StaffGroup` (o equivalente).

`staff-staff-spacing`
: Distancia entre pautas consecutivas dentro del grupo de pautas actual. La propiedad `staff-staff-spacing` del grob `VerticalAxisGroup` de una pauta individual se puede sobreescribir con distintos ajustes de espaciado para dicha pauta.

`staffgroup-staff-spacing`
: Distancia entre la última pauta del grupo en curso y la pauta que está justo por debajo de ella dentro del mismo sistema, incluso si existen una o más líneas que no son pautas (tales como `Lyrics`) entre las dos pautas. No se aplica al pentagrama o pauta inferior de un sistema. La propiedad `staff-staff-spacing` del grob `VerticalAxisGroup` de una pauta individual se puede sobreescribir con distintos ajustes de espaciado para dicha pauta.

Véase también

Referencia de la notación: Sección 26.4 [Variables de espaciado de `\paper` verticales flexibles], página 677, Sección 35.8 [Modificación de las listas-A], página 780.

Archivos de inicio: `ly/engraver-init.ly`, `scm/define-grobs.scm`.

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*, Sección “StaffGrouper” en *Referencia de Funcionamiento Interno*.

29.1.2 Espaciado de pautas no agrupadas

Las *pautas*, tales como los pentagramas (`Staff`), pautas de percusión (`DrumStaff`) o de tablatura (`TabStaff`), etc., son contextos que pueden contener uno o más contextos de voz, pero no pueden contener otras pautas.

Las siguientes propiedades afectan al espaciado de las pautas *no agrupadas*:

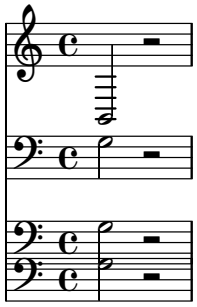
- Propiedades de `VerticalAxisGroup`:
  - `default-staff-staff-spacing`
  - `staff-staff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase Sección 29.1.1 [Propiedades de espaciado dentro de los sistemas], página 701.

Están implicadas ciertas propiedades adicionales para las pautas que son parte de un grupo; véase Sección 29.1.3 [Espaciado de pautas agrupadas], página 706.

El ejemplo siguiente muestra cómo la propiedad `default-staff-staff-spacing` puede afectar al espaciado de pautas no agrupadas. Las mismas sobreescrituras aplicadas a `staff-staff-spacing` tendrían el mismo efecto, pero también se aplicaría en caso de que las pautas estuvieran combinadas en un grupo o grupos.

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing =
      #'((basic-distance . 8)
         (minimum-distance . 7)
         (padding . 1))
  }
}

<<
  % The very low note here needs more room than 'basic-distance
  % can provide, so the distance between this staff and the next
  % is determined by 'padding.
  \new Staff { b,2 r | }

  % Here, 'basic-distance provides enough room, and there is no
  % need to compress the space (towards 'minimum-distance) to make
  % room for anything else on the page, so the distance between
  % this staff and the next is determined by 'basic-distance.
  \new Staff { \clef bass g2 r | }

  % By setting 'padding to a negative value, staves can be made to
  % collide.  The lowest acceptable value for 'basic-distance is 0.
  \new Staff \with {
    \override VerticalAxisGroup.default-staff-staff-spacing =
      #'((basic-distance . 3.5)
         (padding . -10))
  } { \clef bass g2 r | }
  \new Staff { \clef bass g2 r | }
>>
```



Véase también

Archivos de inicio: `scm/define-grobs.scm`.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*.

29.1.3 Espaciado de pautas agrupadas

En partituras grandes como las orquestales, es común colocar los pentagramas en grupos. El espacio entre los grupos suele ser mayor que el espacio que hay entre los pentagramas dentro del mismo grupo.

Los grupos de pautas, *Staff-groups*, tales como `StaffGroup`, `ChoirStaff`, etc., son contextos que pueden contener al mismo tiempo uno o más pentagramas o pautas.

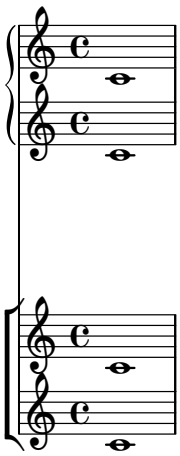
Las siguientes propiedades afectan al espaciado de las pautas dentro de los grupos:

- Propiedades de `VerticalAxisGroup`:
  - `staff-staff-spacing`
- Propiedades de `StaffGrouper`:
  - `staff-staff-spacing`
  - `staffgroup-staff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase Sección 29.1.1 [Propiedades de espaciado dentro de los sistemas], página 701.

El ejemplo siguiente muestra cómo pueden afectar las propiedades del grob `StaffGrouper` al espaciado de las pautas agrupadas:

```
\layout {
  \context {
    \Score
    \override StaffGrouper.staff-staff-spacing.padding = 0
    \override StaffGrouper.staff-staff-spacing.basic-distance = 1
  }
}

<<
  \new PianoStaff \with {
    \override StaffGrouper
              .staffgroup-staff-spacing
              .basic-distance = 20
  } <<
    \new Staff { c'1 }
    \new Staff { c'1 }
  >>

  \new StaffGroup <<
    \new Staff { c'1 }
    \new Staff { c'1 }
  >>
>>
```



Véase también

Archivos de inicio: `scm/define-grobs.scm`.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección "VerticalAxisGroup" en *Referencia de Funcionamiento Interno*, Sección "StaffGrouper" en *Referencia de Funcionamiento Interno*.

29.1.4 Espaciado de las líneas que no son pautas

Las *líneas que no son pautas*, tales como `Lyrics`, `ChordNames`, etc., son contextos cuyos objetos de presentación se imprimen como pentagramas (es decir, en líneas horizontales dentro de los sistemas). Específicamente, las líneas que no son pautas son contextos del tipo no-pauta que contienen el grabador Sección "Axis_group_engraver" en *Referencia de Funcionamiento Interno*.

Las siguientes propiedades afectan al espaciado de las líneas que no son pautas:

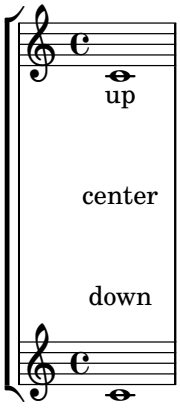
- Propiedades de `VerticalAxisGroup`:
  - `staff-affinity`
  - `nonstaff-relatedstaff-spacing`
  - `nonstaff-nonstaff-spacing`
  - `nonstaff-unrelatedstaff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase Sección 29.1.1 [Propiedades de espaciado dentro de los sistemas], página 701.

El ejemplo siguiente muestra cómo la propiedad `nonstaff-nonstaff-spacing` puede afectar el espaciado de líneas consecutivas que no son pautas. Aquí, mediante el establecimiento de la clave de ampliabilidad `stretchability` a un valor muy grande, la línea de letra es capaz de ampliarse mucho más de lo que es usual:

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
              .nonstaff-nonstaff-spacing
              .stretchability = 1000
  }
}

\new StaffGroup
<<
  \new Staff \with {
```

```
    \override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 30))
  } { c'1 }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #UP
  } \lyricmode { up }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #CENTER
  } \lyricmode { center }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #DOWN
  } \lyricmode { down }
  \new Staff { c'1 }
>>
```



Véase también

Archivos de inicio: `ly/engraver-init.ly`, `scm/define-grobs.scm`.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*.

29.2 Posicionamiento explícito de los pentagramas y los sistemas

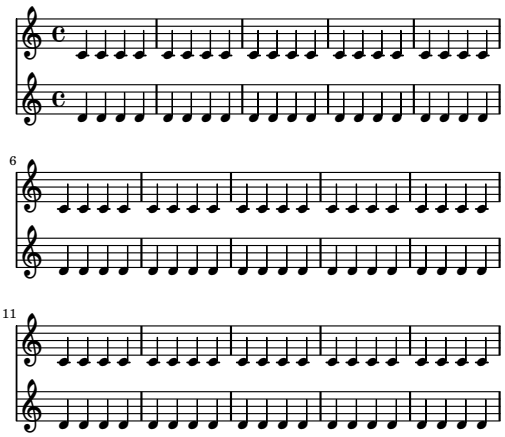
Es posible enfrentarse al espaciado vertical de una forma distinta (comparado con lo que se ha visto en las secciones previas) utilizando la propiedad `line-break-system-details` del objeto gráfico `NonMusicalPaperColumn`. Mientras que los mecanismos del espaciado vertical flexible especifican relleno vertical, `line-break-system-details` puede especificar posiciones verticales exactas sobre la página.

Esta propiedad contiene cinco subpropiedades diferentes: `X-offset` (desplazamiento en X) `Y-offset` (desplazamiento en Y) `extra-offset` (desplazamiento adicional) `alignment-distances` (distancias de alineación) y `bottom-padding` (relleno por debajo).

Para entender cómo funciona cada uno de los distintos ajustes, empezamos observando un ejemplo que no incluye absolutamente ninguna sobreescritura. Mantenemos aisladas tanto la información de los saltos de línea como los de página en una voz dedicada a ello, dividiendo la música en líneas de cinco compases cada una. Esta técnica de crear una voz de saltos ayuda a mantener la disposición separada de la escritura de notas conforme el ejemplo se vuelve más complicado. Consulte Capítulo 28 [Saltos], página 692.

```
\book {
  \score {
```

```
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          s1*5 \break }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
    >>
  }
}
```

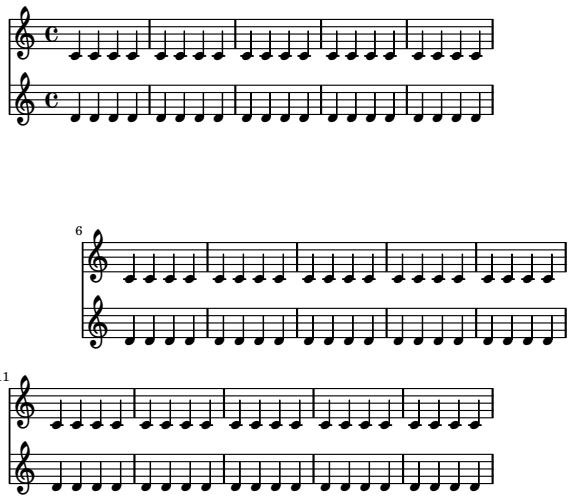


LilyPond v2.26.0

A continuación fijamos los puntos de inicio vertical y horizontal de cada sistema explícitamente usando las subpropiedades `Y-offset` y `X-offset` de `line-break-system-details`.

```
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \once \override
            Score.NonMusicalPaperColumn
                 .line-break-system-details.Y-offset = 0
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn
                 .line-break-system-details.Y-offset = 30
          \once \override
            Score.NonMusicalPaperColumn
                 .line-break-system-details.X-offset = 10
          s1*5 \break
```

```
            \once \override
              Score.NonMusicalPaperColumn
                   .line-break-system-details.Y-offset = 50
            s1*5 \break }
          \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
        >>
        \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
      >>
    }
}
```

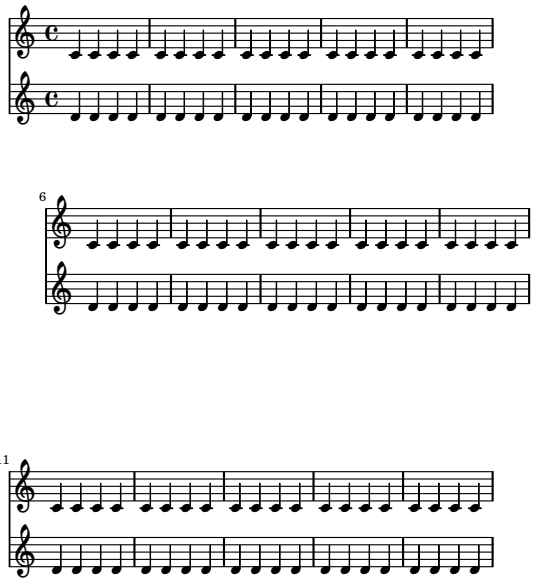


LilyPond v2.26.0

En contraste con el posicionado absoluto que está disponible a través de `Y-offset` y de `X-offset`, es posible un posicionamiento relativo con la subpropiedad `extra-offset`. La colocación es relativa a la disposición predeterminada o al posicionamiento absoluto creado mediante el establecimiento de los valores de `X-offset` y de `Y-offset`. La subpropiedad `extra-offset` acepta una pareja de números (un `pair`) que consiste en desplazamientos a lo largo de los ejes X e Y.

```
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn
              .line-break-system-details.extra-offset = #'(5 . 5)
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn
              .line-break-system-details.extra-offset = #'(0 . 20)
```

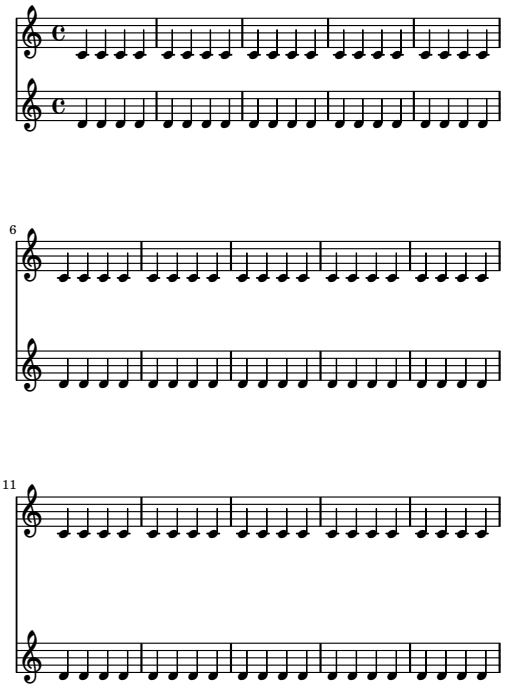
```
          s1*5 \break }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
    >>
  }
}
```



Ahora que hemos establecido explícitamente el punto de origen vertical de cada sistema, podemos también establecer manualmente las distancias verticales entre los pentagramas dentro de cada sistema. Lo hacemos usando la subpropiedad `alignment-distances`. Como puede verse, usamos una lista-A para fijar dos subpropiedades `line-break-system-details` diferentes al mismo tiempo.

```
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \once \override
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((Y-offset . 0)
                 (alignment-distances . (10)))
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((Y-offset . 30)
                 (alignment-distances . (15)))
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((Y-offset . 65)
```

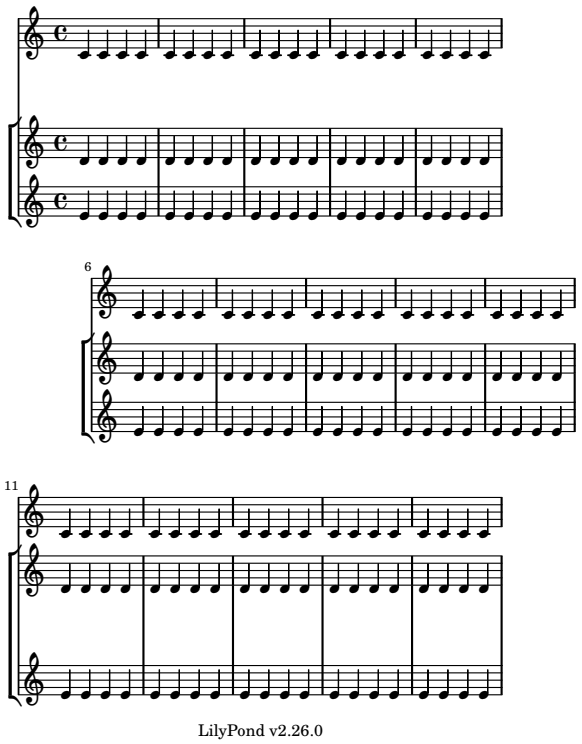
```
                (alignment-distances . (20)))
            s1*5 \break }
          \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
        >>
        \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
      >>
    }
}
```



El ejemplo siguiente muestra cómo fijar las posiciones verticales de varios pentagramas dentro de un sistema. Observe que `alignment-distances` se aplica directamente a los pentagramas, sin preocuparse de las agrupaciones hechas con `StaffGroup` o similares. El valor `#f` indica que se debe usar el valor predeterminado.

```
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \once \override
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((Y-offset . 0)
                 (alignment-distances . (15 8)))
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((Y-offset . 35)
                 (X-offset . 10)
                 (alignment-distances . (#f 8)))
          s1*5 \break
          \once \override
```

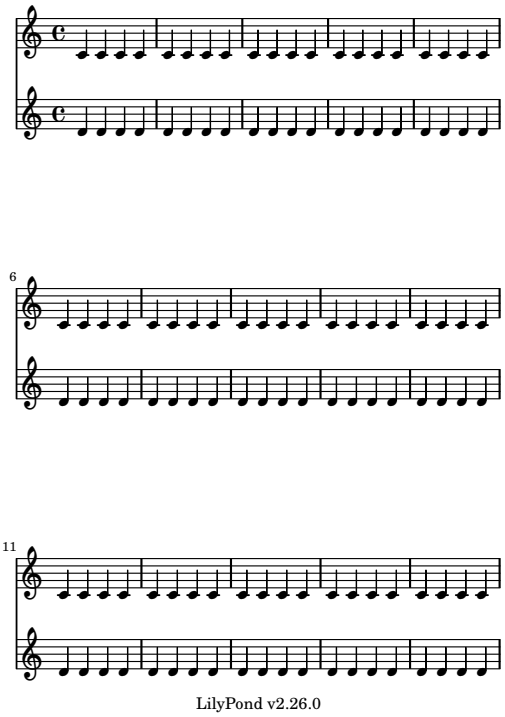
```
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((Y-offset . 65)
                 (alignment-distances . (8 15)))
          s1*5 \break }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new StaffGroup <<
        \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
        \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
      >>
    >>
  }
}
```



La última subpropiedad de `line-break-system-details`, `bottom-padding`, tiene un efecto similar al de `last-bottom-spacing`. Sin embargo, observe que no guarda una dimensión de espaciado que se pueda estirar o comprimir, sino un valor de relleno rígido y fijo a partir del extremo inferior de la página hasta el pentagrama inferior. Aquí aparece el fragmento de prueba con la parte inferior justificada, mostrando la distancia por omisión.

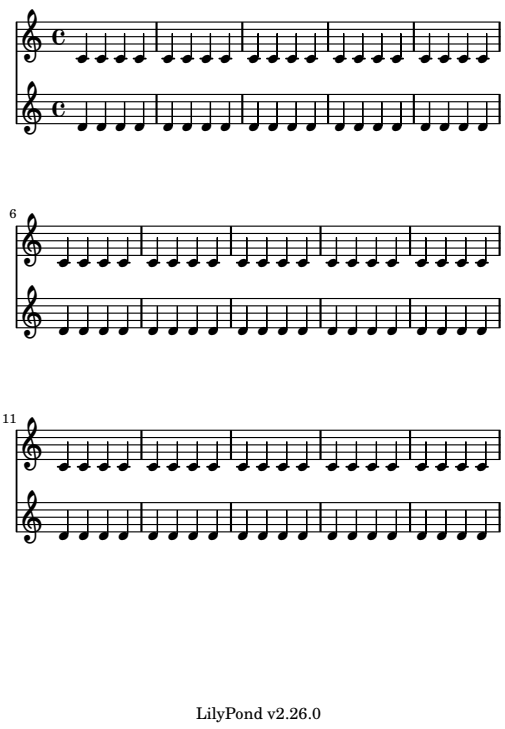
```
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          s1*5 \break }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff { \repeat unfold 15 { d'4 d' d' d' }
```

```
      }
    >>
  }
  \paper { ragged-last-bottom = ##f }
}
```



Al fijar un valor para `bottom-padding` se obtiene el resultado siguiente.

```
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          \once \override
            Score.NonMusicalPaperColumn.line-break-system-details =
              #'((bottom-padding . 20))
          s1*5 \break }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
    >>
  }
  \paper { ragged-last-bottom = ##f }
}
```



Algunos puntos que tener en cuenta:

- Al usar `alignment-distances`, la letra y otras líneas que no son pautas, no cuentan como un pentagrama.
- Las unidades de los números que se pasan a las subpropiedades `X-offset`, `Y-offset`, `extra-offset` y `alignment-distances` se interpretan como múltiplos de la distancia entre líneas de pentagrama adyacentes. Los valores positivos mueven a los pentagramas y a la letra hacia arriba, los valores negativos mueven los pentagramas y la letra hacia abajo.
- A causa de que los ajustes a `NonMusicalPaperColumn.line-break-system-details` dados aquí permiten el posicionamiento de pentagramas y sistemas en cualquier lugar de la página, es posible violar los márgenes o los límites del papel, o incluso imprimir pentagramas o sistemas unos encima de otros. Para evitarlo, pase unos valores razonables a las subpropiedades.

Véase también

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

29.3 Evitar las colisiones verticales

Podemos decir intuitivamente que algunos objetos de la notación musical pertenecen al pentagrama y otros se sitúan fuera del pentagrama. Entre los objetos que pertenecen al exterior del pentagrama están las marcas de ensayo, las marcas textuales y las indicaciones de dinámica (en lo sucesivo, les llamaremos objetos fuera del pentagrama). La regla de LilyPond para la colocación vertical de los objetos fuera del pentagrama es ponerlos tan cerca del pentagrama como sea posible pero no tan próximos como para que choquen con otro objeto.

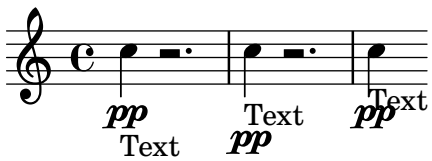
LilyPond utiliza la propiedad `outside-staff-priority` para determinar si un grob es un objeto fuera del pentagrama: si `outside-staff-priority` es un número, el grob es un objeto fuera del pentagrama. Además, `outside-staff-priority` informa a LilyPond en qué orden se debe situar los objetos.

En primer lugar, LilyPond sitúa todos los objetos que no pertenecen al exterior del pentagrama. Después ordena los objetos fuera del pentagrama de acuerdo a su prioridad

outside-staff-priority (en orden creciente). LilyPond toma los objetos fuera del pentagrama uno a uno y los coloca de forma que no choquen con ningún objeto que ya haya sido colocado. Esto es, si dos grobs fuera del pentagrama compiten por el mismo espacio, el que tiene la prioridad outside-staff-priority más baja se colocará más próximo al pentagrama.

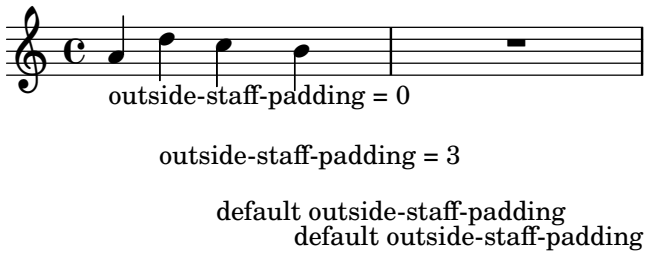
Hay un listado completo de los valores predeterminados de outside-staff-priority en Sección B.17 [Valores por omisión para outside-staff-priority], página 943.

```
\relative c'' {
  c4_"Text"\pp
  r2.
  \once \override TextScript.outside-staff-priority = 1
  c4_"Text"\pp % this time the text will be closer to the staff
  r2.
  % by setting outside-staff-priority to a non-number,
  % we disable the automatic collision avoidance
  \once \override TextScript.outside-staff-priority = ##f
  \once \override DynamicLineSpanner.outside-staff-priority = ##f
  c4_"Text"\pp % now they will collide
}
```



El relleno vertical que rodea a los objetos fuera del pentagrama se puede controlar con la propiedad outside-staff-padding.

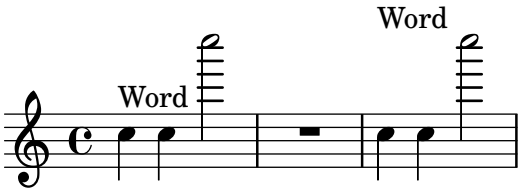
```
\relative {
  \once \override TextScript.outside-staff-padding = 0
  a'4-"outside-staff-padding = 0"
  \once \override TextScript.outside-staff-padding = 3
  d-"outside-staff-padding = 3"
  c-"default outside-staff-padding"
  b-"default outside-staff-padding"
  R1
}
```



De forma predeterminada, los objetos fuera del pentagrama se colocan de forma que eviten una colisión horizontal con los grobs posicionados previamente. Esto puede llevar a situaciones en las que los objetos se colocan muy próximos entre sí en el sentido horizontal. Como se muestra en el ejemplo que aparece más abajo, al establecer outside-staff-horizontal-padding se incrementa el espacio horicontal necesario, y en este caso mueve el texto hacia arriba para evitar que quede demasiado cerca de las líneas adicionales.

```
\relative {
  c''4^"Word" c c''2
```

```
  R1
  \once \override TextScript.outside-staff-horizontal-padding = 1
  c,,4^"Word" c c''2
}
```



Véase también

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

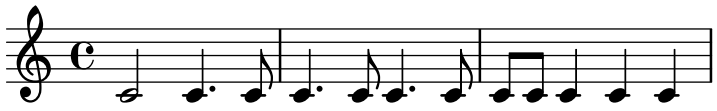
30 Espaciado horizontal

30.1 Panorámica del espaciado horizontal

El motor de espaciado traduce las diferencias en las duraciones a distancias ampliables ('muelles') de distintas longitudes. Las duraciones más largas reciben un espacio mayor y las duraciones más cortas reciben menos. Las duraciones más breves reciben un espacio de tamaño fijo (que se controla mediante `shortest-duration-space` en el objeto Sección "SpacingSpanner" en *Referencia de Funcionamiento Interno*). Cuanto más larga es la duración, más espacio recibe: al doblar una duración se suma `spacing-increment` al espacio que ocupa la nota.

Por ejemplo, la pieza siguiente contiene numerosas blancas, negras y corcheas; la corchea va seguida de un ancho de cabeza de blanca (ACB). La negra va seguida de 2 ACB, la blanca por 3 ACB, etc.

```
\relative c' {
  c2 c4. c8
  c4. c8 c4. c8
  c8 c c4 c c
}
```



Normalmente, el valor de `spacing-increment` está establecido en 1.2 espacios de pentagrama, lo que es aproximadamente el ancho de una cabeza, y `shortest-duration-space` está establecido en 2.0, lo que significa que la nota más breve recibe 2.4 espacios de pentagrama (2.0 multiplicado por el `spacing-increment`) de espacio horizontal. Este espacio se cuenta a partir del borde izquierdo del símbolo, de manera que las notas más breves van seguidas generalmente por un espacio de 1 ACB.

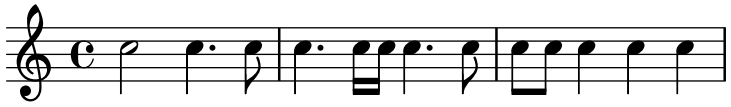
Si siguiésemos el procedimiento anterior exactamente, entonces la adición de una sola fusa a una partitura que usa corcheas y semicorcheas, aumentaría enormemente la anchura de la partitura completa. La nota más breve ya no es la semicorchea, sino la fusa, añadiendo así 1 ACB a cada una de las notas. Para evitarlo, la duración más breve a efectos de espaciado no es la nota más breve de la partitura, sino la que ocurre con más frecuencia.

La duración más breve que es más común se determina de la siguiente manera: en cada compás se determina la duración más breve. La menor duración más común se toma como base para el espaciado, habiendo estipulado que esta duración menor siempre debe ser igual o menor que una corchea. La duración más breve se imprime cuando se ejecuta `lilypond` con la opción `--verbose`.

Estas duraciones también se pueden personalizar. Si establecemos la `common-shortest-duration` en Sección "SpacingSpanner" en *Referencia de Funcionamiento Interno*, entonces éste establece la duración base para el espaciado. La duración máxima para esta base (normalmente una corchea), se fija a través de `base-shortest-duration`.

Las notas que son aún más breves que la nota común más breve van seguidas por un espacio proporcional a su duración en relación con la nota común más breve. Así pues, si fuésemos a añadir tan sólo algunas semicorcheas al ejemplo anterior, irían seguidas por medio ACB:

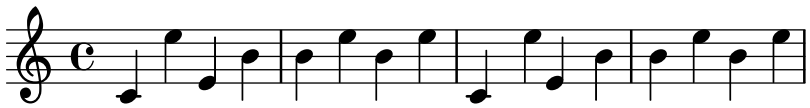
```
\relative { c''2 c4. c8 | c4. c16[ c] c4. c8 | c8 c c4 c c }
```



Como se explicó en el *Ensayo sobre grabado musical automatizado*, las direcciones de las plicas influyen en el espaciado (véase Sección “Espaciado óptico” en *Ensayo*) y se puede ajustar con la propiedad `stem-spacing-correction` del objeto Sección “NoteSpacing” en *Referencia de Funcionamiento Interno*. Estos se generan para cada uno de los contextos de Sección “Voice” en *Referencia de Funcionamiento Interno*.

El objeto `StaffSpacing` (generado en el contexto de Sección “Staff” en *Referencia de Funcionamiento Interno*) contiene la misma propiedad para controlar el espaciado de las líneas de plica o divisorias.

El ejemplo siguiente muestra estas correcciones, una vez con los valores predeterminados y otra con un ajuste exagerado:



Está contemplada la notación proporcional; consulte Sección 30.6 [Notación proporcional], página 728.

Véase también

Ensayo sobre grabado musical automatizado: Sección “Espaciado óptico” en *Ensayo*.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SpacingSpanner” en *Referencia de Funcionamiento Interno*, Sección “NoteSpacing” en *Referencia de Funcionamiento Interno*, Sección “StaffSpacing” en *Referencia de Funcionamiento Interno*, Sección “NonMusicalPaperColumn” en *Referencia de Funcionamiento Interno*.

Advertencias y problemas conocidos

No existe ningún mecanismo conveniente para sobreescribir el espaciado manualmente. Se puede usar el siguiente rodeo para insertar espacio adicional en una partitura, ajustando el valor del relleno en la medida necesaria.

```
\override Score.NonMusicalPaperColumn.padding = 10
```

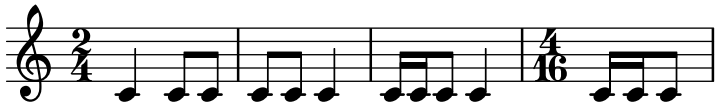
No existe ningún rodeo para disminuir la magnitud de la separación.

30.2 Sección de espaciado nueva

Se pueden iniciar secciones nuevas con diferentes parámetros de espaciado, con la instrucción `newSpacingSection`. Esto es útil para las secciones que tienen distinta noción de las notas ‘largas’ y ‘cortas’. La instrucción `\newSpacingSection` crea un nuevo objeto `SpacingSpanner` en ese momento musical.

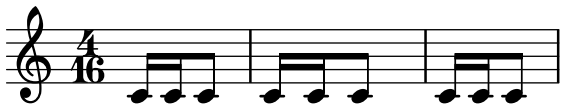
En el ejemplo siguiente, el cambio de compás introduce una sección nueva, y por ello las semicorcheas se separan automáticamente de manera ligeramente más amplia.

```
\relative c' {
  \time 2/4
  c4 c8 c
  c8 c c4 c16[ c c8] c4
  \newSpacingSection
  \time 4/16
  c16[ c c8]
}
```



Si los ajustes de espaciado automáticos no dan el espaciado deseado, pueden aplicarse sobreescrituras manuales a sus propiedades con `\override`. Las sobreescrituras se deben aplicar en el mismo momento musical que la propia instrucción `\newSpacingSection` y afectarán entonces al espaciado de toda la música que venga a continuación hasta que las propiedades se cambien en una nueva sección de espaciado, por ejemplo:

```
\relative c' {
  \time 4/16
  c16[ c c8]
  \newSpacingSection
  \override Score.SpacingSpanner.spacing-increment = 2
  c16[ c c8]
  \newSpacingSection
  \revert Score.SpacingSpanner.spacing-increment
  c16[ c c8]
}
```



Véase también

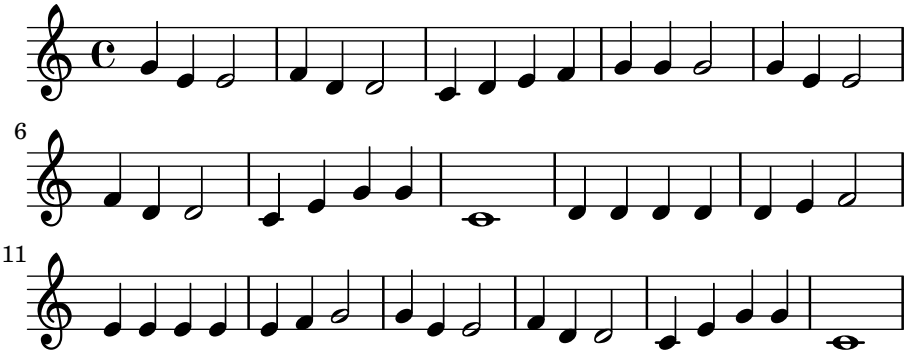
Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SpacingSpanner” en *Referencia de Funcionamiento Interno*.

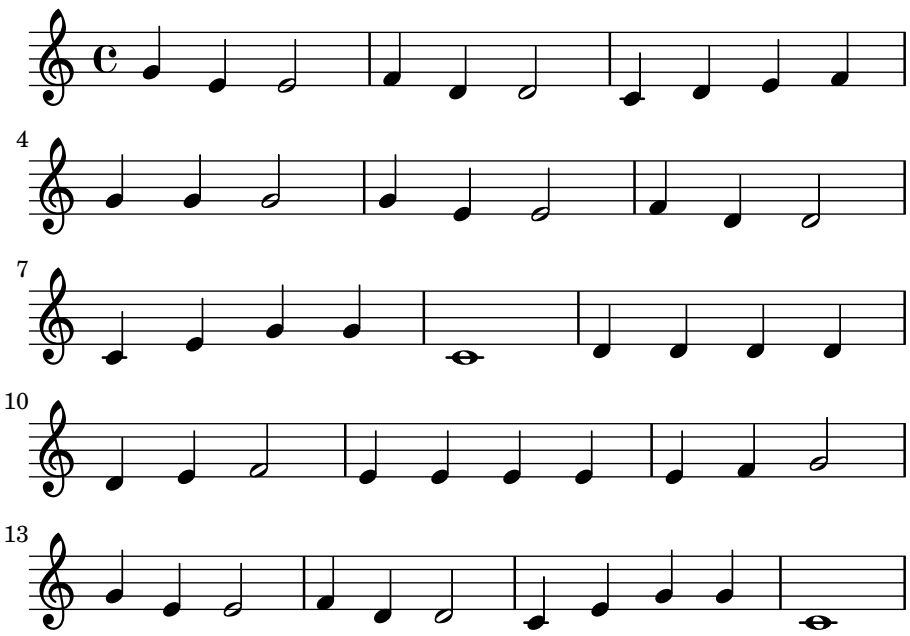
30.3 Cambiar el espaciado horizontal globalmente

Se puede alterar el espaciado horizontal con la propiedad `base-shortest-duration`. Aquí compararemos la misma música, una vez sin alterar la propiedad, y luego alterándola. Los valores mayores producen música más apretada.

```
\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}
```



```
\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration = \musicLength 16
    }
  }
}
```

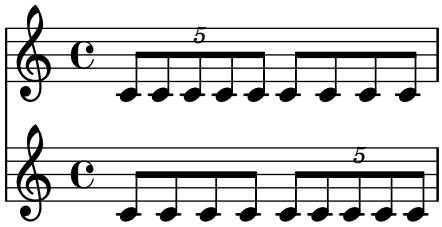


30.3.1 Estiramiento uniforme de los grupos especiales

De forma predeterminada, el espaciado en los grupos de valoración especial depende de varios factores ajenos a la duración (como alteraciones, cambios de clave, etc.). Para pasar por alto estos símbolos y forzar un espaciado de duraciones iguales uniforme, use `Score.SpacingSpanner.uniform-stretching`. Esta propiedad solo puede cambiarse al principio de la partitura:

```
\score {
  <<
    \new Staff \relative c' {
      \tuplet 5/4 { c8 c c c c } c8 c c c
    }
    \new Staff \relative c' {
      c8 c c c \tuplet 5/4 { c8 c c c c }
    }
  >>
  \layout {
    \context {
      \Score
```

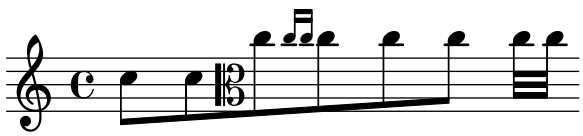
```
        \override SpacingSpanner.uniform-stretching = ##t
      }
    }
}
```



30.3.2 Espaciado estricto de las notas

Cuando se establece `strict-note-spacing`, las notas se separan sin tener en cuenta las claves, líneas divisorias ni notas de adorno:

```
\override Score.SpacingSpanner.strict-note-spacing = ##t
\new Staff \relative {
  c''8[ c \clef alto c \grace { c16 c } c8 c c]  c32[ c] }
```



Véase también

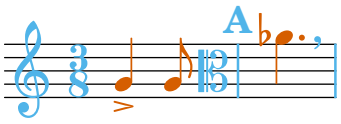
Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

30.4 Ajuste del espaciado horizontal para objetos de disposición específicos

Además de los parámetros de uso general del algoritmo predeterminado de espaciamiento que se aplican a todos los elementos de la partitura o de la sección de espaciado, existen varias propiedades útiles para hacer ajustes sobre los objetos, tipo po tipo. Entre los ejemplos se encuentran el ajuste de la distancia entre la clave y la indicación de compás, pero no entre la clave y las notas cuando no hay ninguna indicación de compás, o la distensión entre las notas en presencia de un texto impreso de tal forma que no se solape con la nota siguiente. Estos procedimientos de trucaje requieren que primero se identifique el tipo de especiado que está en juego.

30.4.1 Panorámica de los trucajes de espaciado horizontal específicos de cada objeto

Los objetos de disposición que tienen una posición fija horizontal se llaman *items* (elementos) por oposición a los *spanners* (objetos de extensión) y respecto al espaciado horizontal se agrupan en *columnas*. Las cabezas de nota y los silencios, que forman el material musica principal junto con esos objetos que están enlazados a ellos lógicamente (alteraciones accidentales, articulaciones, plicas, puntillos, etc.) son todos parte de ‘columnas musicales’ (representadas por objetos gráficos del tipo `NoteColumn`). El *material introductorio*, como las claves, indicaciones de compás y líneas divisorias, se agrupa en ‘columnas no musicales’ (representadas por objetos gráficos del tipo `NonMusicalPaperColumn`). En el ejemplo siguiente, los elementos musicales están en color rojo, y los no musicales en azul.



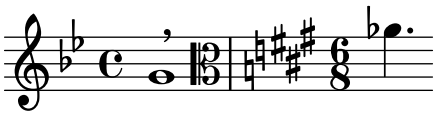
Este ejemplo muestra que existe una alternancia entre columnas musicales y no musicales. La primera columna no musical contiene una clave y una indicación de compás. La primera columna musical tiene una cabeza de nota con su plica y una articulación. La segunda columna no musical está vacía y por tanto se elimina durante el proceso de disposición. La segunda columna musical tiene, de nuevo, una nota. La tercera columna no musical contiene una clave, una barra de compás y una llamada de ensayo, y así sucesivamente.

Dentro de cada columna, el espaciado es fijo. Por otro lado, el espacio de separación entre columnas consecutivas es flexible. Como veremos, los métodos para ajustar el espaciado dentro de una columna y entre distintas columnas, son diferentes.

30.4.2 Espaciado entre elementos no musicales adyacentes

Dentro de una columna no musical, los elementos se van disponiendo en un orden específico. Por ejemplo, con el conjunto de elementos de la imagen que aparece a continuación, el orden predeterminado sitúa primero el signo de respiración, después la clave, luego la barra de compás, la armadura de cancelación, la armadura normal, y por último la indicación de compás (esto se controla por parte de la propiedad `BreakAlignment.break-align-orders`, que se puede cambiar de forma conveniente usando `\breakAlignInsert`).

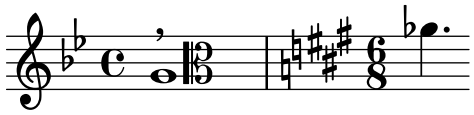
```
\relative {
  \key g \minor
  g'1
  \breathe
  \clef alto
  \time 6/8
  \key a \major
  aes4.
}
```



La distancia entre dos elementos adyacentes de la misma columna no musical viene controlada por el valor de la propiedad `space-alist` (lista de asociación de espaciado) de la que está más a la izquierda de las dos. `space-alist` tiene la forma de una lista asociativa que asigna símbolos de break-align a pares (*`spacing-style . value`*). El símbolo break-align de un elemento divisible viene dado por el valor de su propiedad `break-align-symbol`; las opciones estándar están relacionadas en Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno*. Los estilos de espaciado se relacionan en Sección “break-aligned-interface” en *Referencia de Funcionamiento Interno*. Entre las opciones que están disponibles, solo `extra-space` y `minimum-space` son relevantes para el trucaje del espacio entre elementos no musicales. La diferencia es que `extra-space` mide el espacio de relleno a partir de la derecha del primer objetohasta la izquierda del segundo objeto, mientras que `minimum-space` cuenta a partir de la izquierda del primer objeto. De esta forma, una manera de desplazar la línea divisoria, alejándola de la clave, sería:

```
\relative {
  \key g \minor
  g'1
```

```
  \override Staff.Clef.space-alist.staff-bar = #'(extra-space . 4)
  \breathe
  \clef alto
  \time 6/8
  \key a \major
  aes4.
}
```



Los ajustes de space-alist, que no se limitan a los dos estilos de espaciado que se han descrito anteriormente, son posibles también para sobreescribir el espaciado entre diferentes columnas. Sin embargo, este tipo de espaciado es flexible, y no depende meramente de los tipos de objeto implicados, sino también de sus formas. Los métodos específicos para ello se documentan en el apartado siguiente.

Fragmentos de código seleccionados

Separar las cancelaciones de tonalidad de los cambios de armadura

De forma predeterminada, las alteraciones accidentales que se usan para las cancelaciones en las armaduras se colocan adyacentes a las que se usan para los cambios de tonalidad. Este comportamiento se puede cambiar sobreescribiendo la propiedad break-align-orders del objeto gráfico BreakAlignment.

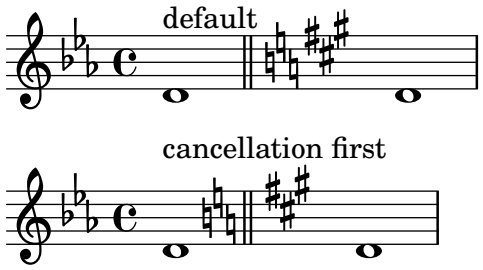
Si busca la definición de break-alignment-interface en la Referencia de funcionamiento interno (https://lilypond.org/doc/v2.24/Documentation/internals/break_002dalignment_002dinterface) de LilyPond, obtendrá la siguiente lista de los símbolos de break-align disponibles:

```
ambitus
breathing-sign
clef
cue-clef
cue-end-clef
custos
key-cancellation
key-signature
left-edge
signum-repetitionis
staff-bar
staff-ellipsis
time-signature
```

A partir de esta lista, vemos que lo que queremos hacer es trasladar la key-cancellation de forma que esté antes del staff-bar. Esto se hace con la función \breakAlignInsert.

```
music = { \key es \major d'1 \bar "||"
          \key a \major d'1 }

{ <>^\markup "default"
  \music }

{ <>^\markup "cancellation first"
```

```
\breakAlignInsert key-cancellation before staff-bar
\music }
```



Véase también

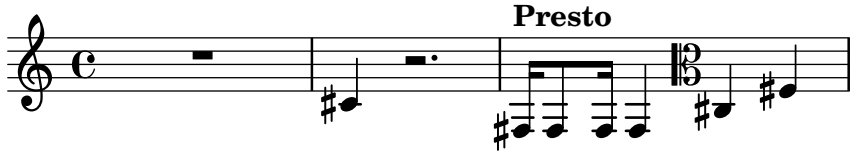
Referencia de la notación: Sección 36.9.4 [Uso del `break-alignable-interface`], página 804.

Manual de extensión de LilyPond: Sección “Association lists (alists)” en *Extender*.

Referencia de funcionamiento interno: Sección “Break_align_engraver” en *Referencia de Funcionamiento Interno*, Sección “BreakAlignGroup” en *Referencia de Funcionamiento Interno*, Sección “BreakAlignment” en *Referencia de Funcionamiento Interno*, Sección “break-alignable-interface” en *Referencia de Funcionamiento Interno*, Sección “break-aligned-interface” en *Referencia de Funcionamiento Interno*, Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno*.

30.4.3 Espaciado entre columnas adyacentes

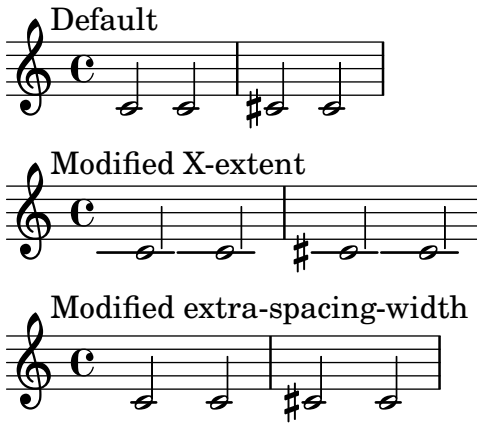
A diferencia del espaciado dentro de una columna, el espaciado entre columnas adyacentes es flexible y se amplía o se comprime de acuerdo a la densidad de la música sobre la línea. De manera predeterminada, las columnas pueden incluso solaparse en determinadas situaciones. El ejemplo siguiente muestra tres casos. La segunda alteración accidental se desliza por detrás de la barra de compás, mientras que la tercera se solapa con la clave. Asimismo, la indicación de tempo *Presto* se extiende sobre varias columnas. Fíjese en cómo la primera alteración, que permanece dentro de los límites verticales de la barra de compás por el lado izquierdo, se desplaza alejándose.



Estas reglas de espaciado se pueden configurar. Esto se hace modificando la anchura y altura que ocupa un objero en el espaciado horizontal. Las propiedades de relevancia son `extra-spacing-width` y `extra-spacing-height`. Cuando no están establecidas, un objeto ocupa tanto espacio en horizontal como le permitan sus propiedades `X-extent` y `Y-extent`. Estos son valres precisos de sus dimensiones. Las propiedades `extra-spacing-width` y `extra-spacing-height` hacen que un objeto se haga más grande o más pequeño a los efectos del cálculo del espaciado horizontal entre columnas solamente, pero preservan sus dimensiones para otros tipos de espaciado.

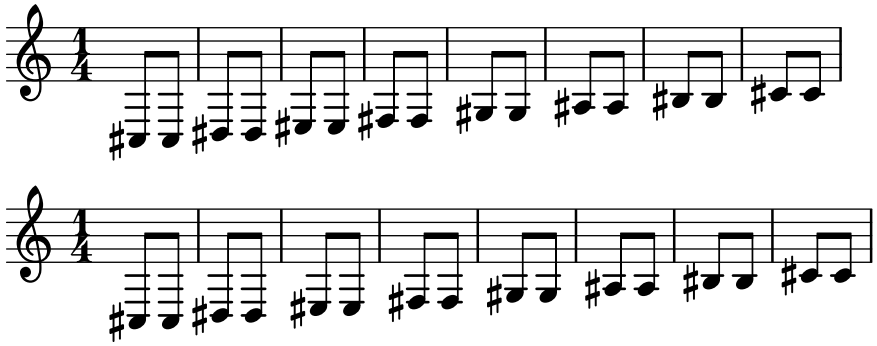
```
{
  \textMark "Default"
  c'2 2 cis'2 2
}

{
  \textMark "Modified X-extent"
  \override NoteHead.X-extent = #'(-2 . 2)
```

```
  c'2 2 cis'2 2
}

{
  \textMark "Modified extra-spacing-width"
  \override NoteHead.extra-spacing-width = #'(-2 . 2)
  c'2 2 cis'2 2
}
```



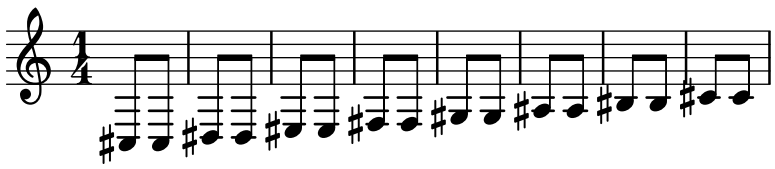
`extra-spacing-width` y `extra-spacing-height` son pares de números, que se añaden a las dimensiones sobre cada uno de los ejes. Por ejemplo, establecer `extra-spacing-height` a `'(-2 . 3)` hace que el objeto sea tres unidades mayor en la parte superior, y dos unidades mayor en la parte de abajo (el límite reducido en 2). El ejemplo siguiente muestra cómo usar `extra-spacing-height` para modificar el límite después del cual las alteraciones accidentales ya no se solapan con las barras de compás.

```
music = \relative {
  \time 1/4
  cis8 8 | dis8 8 | eis8 8 | fis8 8 |
  gis8 8 | ais8 8 | bis8 8 | cis8 8 |
}

{
  \music
}

{
  \override Accidental.extra-spacing-height = #'(0 . 1.0)
  \music
}
```



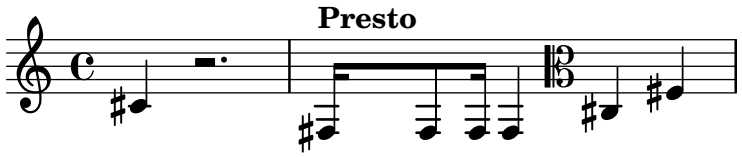
El valor `'(+inf.0 . -inf.0)` para `extra-spacing-width` o para `extra-spacing-height` suprime la presencia del objeto.

```
music = \relative {
  \time 1/4
  cis8 8 | dis8 8 | eis8 8 | fis8 8 |
  gis8 8 | ais8 8 | bis8 8 | cis8 8 |
}

{
  \override Accidental.extra-spacing-height = #'(+inf.0 . -inf.0)
  \music
}
```



De forma recíproca, un valor para `extra-spacing-height` de `'(-inf.0 . +inf.0)` hace al objeto infinitamente alto, impidiendo completamente la superposición con otra columna. El ejemplo que aparece a continuación muestra esta técnica sobre `Accidental` y `MetronomeMark`. En el caso de `MetronomeMark`, es necesario ajustar el valor de `extra-spacing-width` a `'(0 . 0)` porque el valor predeterminado es `'(+inf.0 . -inf.0)`, e incluso un objeto infinitamente alto no ocupa ningún espacio si no tiene anchura.

```
{
  \override Score.MetronomeMark.extra-spacing-width =
    #'(0 . 0)
  \override Score.MetronomeMark.extra-spacing-height =
    #'(-inf.0 . +inf.0)
  \override Accidental.extra-spacing-height =
    #'(-inf.0 . +inf.0)
  cis'4 r2.
  \tempo Presto
  fis16 8 16 4 \clef alto cis4 fis4
}
```



Véase también

Referencia de funcionamiento interno: Sección “item-interface” en *Referencia de Funcionamiento Interno*, Sección “separation-item-interface” en *Referencia de Funcionamiento Interno*.

30.5 Longitud de la línea

Los ajustes más básicos que influyen en el espaciado son `indent` y `line-width`. Se definen en el bloque `\layout`. Controlan el sangrado de la primera línea de música y la longitud de las líneas.

Si se fija el valor `#t` para `ragged-right` en el bloque `\layout`, entonces los sistemas terminan en su longitud horizontal natural, en lugar de repartirse horizontalmente hasta llenar toda la

línea. Esto es útil para fragmentos cortos, y para comprobar qué tan apretado es el espaciado natural. El valor normal predeterminado es falso, pero si la partitura tiene un solo sistema, el valor predeterminado es verdadero.

La opción `ragged-last` es similar a `ragged-right`, pero afecta solo a la última línea de la pieza. No se efectúa ninguna restricción sobre dicha línea. El resultado es similar al formateo de los párrafos de texto. En un párrafo, la última línea sencillamente ocupa su longitud horizontal natural.

```
\layout {
  indent = 0
  line-width = 150
  ragged-last = ##t
}
```

Véase también

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

30.6 Notación proporcional

LilyPond contempla la notación proporcional, un tipo de espaciado horizontal en el que cada nota consume una medida horizontal que equivale exactamente a su duración rítmica. Este tipo de espaciado proporcional es comparable al espaciado horizontal hecho sobre un papel milimetrado. Ciertas partituras del finales del s.XX y principios del s.XXI utilizan notación proporcional para clarificar relaciones rítmicas complejas o para facilitar la colocación de líneas cronométricas u otros gráficos directamente en la partitura.

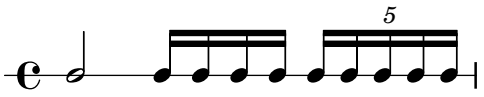
Están contemplados los siguiente ajustes para la notación proporcional, que se pueden usar solos o combinados:

- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove Separating_line_group_engraver`

En los ejemplos que siguen, exploramos el uso de los distintos ajustes de la notación proporcional y examinamos la forma en que interactúan.

Comenzamos con el siguiente ejemplo de un solo compás, que usa un espaciado clásico sin justificación por la derecha.

```
\new RhythmicStaff {
  c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
}
```



Observe que la blanca que inicia el compás ocupa mucho menos de la mitad de todo el espacio horizontal del compás. De forma similar, las semicorcheas y el cinquillo de semicorcheas con que finaliza el compás ocupan en conjunto mucho más de la mitad de todo el espacio horizontal del compás.

En el grabado clásico, este espaciado puede ser exactamente el que deseamos porque podemos tomar prestado el espacio horizontal de la blanca y conservar el espacio horizontal a lo largo del compás como un todo.

Por otro lado, si queremos insertar una línea de tiempo graduada o algún otro gráfico encima o debajo de la partitura, necesitamos la notación proporcional. Se activa la notación proporcional con el ajuste `proportionalNotationDuration`.

```
\new RhythmicStaff {
  c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
}
\layout {
  \context {
    \Score
    proportionalNotationDuration = #1/20
  }
}
```



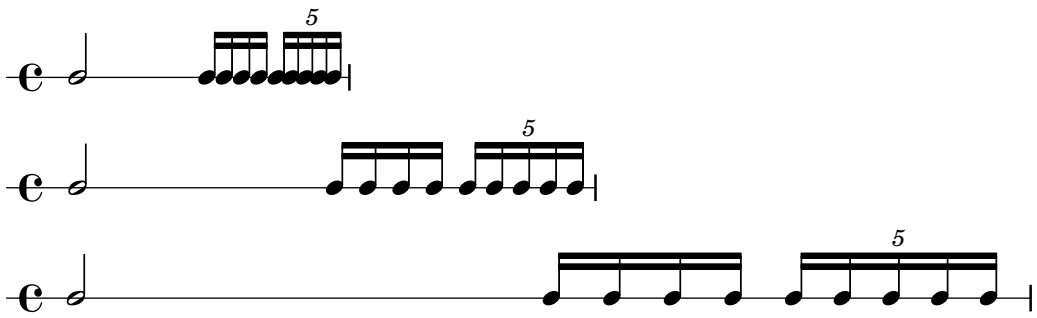
La blanca al principio del compás y las notas rápidas de la segunda mitad del compás ocupan ahora cantidades iguales de espacio horizontal. Podríamos colocar una línea de tiempo graduada o un gráfico encima o debajo de este ejemplo.

El ajuste `proportionalNotationDuration` es un ajuste de contexto que reside en `Score`. Recordemos que los ajustes de contexto aparecen en uno de tres posibles lugares del archivo de entrada: en un bloque `\with`, en un bloque `\context`, o directamente entre la música precedido por la instrucción `\set`. Como con todos los ajustes de contexto, el usuario puede elegir en cuál de los tres lugares diferentes prefiere establecer el valor de `proportionalNotationDuration`.

El ajuste `proportionalNotationDuration` acepta un solo argumento, que es la duración de referencia para espaciar toda la música. El valor `#1/20` especifica un veinteavo de redonda.

¿Cómo seleccionamos la duración correcta de referencia para pasarla a `proportionalNotationDuration`? Normalmente mediante un proceso de ensayo y error, comenzando con una duración cercana a la más rápida (o más breve) de la pieza. Las duraciones de referencia más pequeñas aplican un espaciado más suelto; las duraciones de referencia más largas aplican un espaciado más apretado.

```
rhythm = { c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 } }

\new RhythmicStaff {
  \set Score.proportionalNotationDuration = #1/8
  % Allow overlapping of note heads.
  \override NoteHead.extra-spacing-width = #'(+inf.0 . -inf.0)
  \rhythm
}

\new RhythmicStaff {
  \set Score.proportionalNotationDuration = #1/16
  \rhythm
}

\new RhythmicStaff {
  \set Score.proportionalNotationDuration = #1/32
  \rhythm
}
```

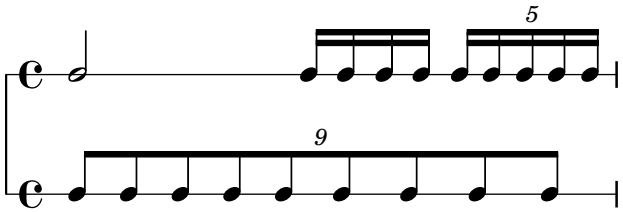


Observe que una duración de referencia demasiado grande (como la corchea, en el ejemplo de arriba) produce un espaciado excesivamente apretado y puede ser causa de colisiones entre las cabezas de nota. En general, la notación proporcional ocupa más espacio horizontal que el espaciado clásico. El espaciado proporcional aporta claridad rítmica a expensas del espacio horizontal.

Ahora veremos cómo espaciar de forma óptima grupos de valoración especial que se superponen.

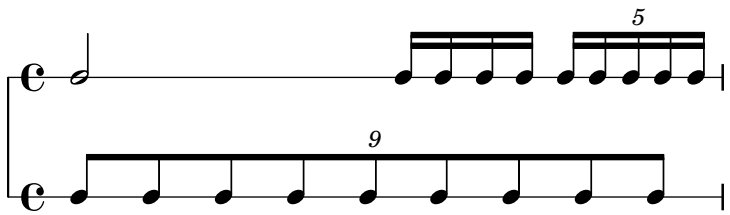
Empezamos por examinar qué le ocurre a nuestro ejemplo original, con espaciado clásico, cuando añadimos un segundo pentagrama con un tipo diferente de grupo especial.

```
<<
  \new RhythmicStaff {
    c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
  }
  \new RhythmicStaff {
    \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
  }
>>
```



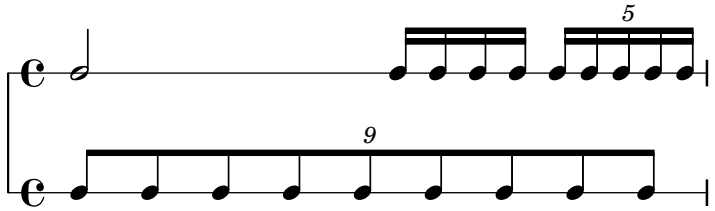
El espaciado es defectuoso porque las notas regularmente espaciadas del pentagrama inferior no se amplían de manera uniforme. Los grabados clásicos incluyen muy pocos tresillos complejos y así las reglas del grabado clásico pueden generar este tipo de resultado. El establecimiento de `proportionalNotationDuration` lo soluciona.

```
<<
  \new RhythmicStaff {
    c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
  }
  \new RhythmicStaff {
    \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
  }
>>
\layout {
  \context {
    \Score
    proportionalNotationDuration = #1/20
  }
}
```



Pero si observamos con mucho cuidado podremos ver que las notas de la segunda mitad del 9-illo están espaciadas de forma ligeramente más ancha que las de la primera mitad del 9-illo. Para asegurar una ampliación uniforme, activamos `uniform-stretching`, que es una propiedad de `SpacingSpanner`.

```
<<
  \new RhythmicStaff {
    c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
  }
  \new RhythmicStaff {
    \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
  }
>>
\layout {
  \context {
    \Score
    proportionalNotationDuration = #1/20
    \override SpacingSpanner.uniform-stretching = ##t
  }
}
```



Nuestro ejemplo de dos pentagramas ahora está espaciado exactamente, nuestras relaciones rítmicas son visualmente claras, y podemos incluir una línea de tiempo graduada o un gráfico, si queremos.

Siempre se recomienda fijar el valor de la propiedad `uniform-stretching` de `SpacingSpanner` a `#t` si estamos usando `proportionalNotationDuration`. Por ejemplo, omitirlo hace que los silencios de separación consuman una cantidad incorrecta de espacio horizontal.

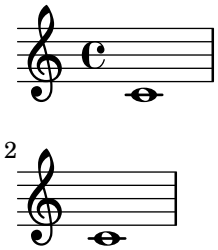
El `SpacingSpanner` es un grob abstracto que reside en el contexto de `Score`. Como con nuestros ajustes de `proportionalNotationDuration`, las sobreescrituras al `SpacingSpanner` solo pueden ocurrir en uno de estos tres lugares dentro del archivo de entrada: en el bloque `\with` de `Score`, en el bloque `\context` de `Score` o directamente dentro de la escritura de notas.

De manera predeterminada, solo hay un `SpacingSpanner` por `Score`. Esto supone que `uniform-stretching` está activado para la partitura completa o desactivado para la partitura completa. Sin embargo, podemos sobreescribir este comportamiento y activar distintas posibilidades de espaciado en distintos lugares de la partitura usando la instrucción `\newSpacingSection`. Véase Sección 30.2 [Sección de espaciado nueva], página 719, para más información.

A continuación examinamos los efectos del grabador `Separating_line_group_engraver` y veremos por qué las partituras proporcionales con frecuencia eliminan este grabador. El ejemplo

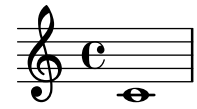
siguiente muestra que hay una pequeña cantidad de espacio “preliminar” justo antes de la primera nota de cada sistema.

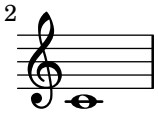
```
\paper {
  indent = 0
}

\new Staff {
  c'1 \break
  c'1
}
```



Esta cantidad de espacio preliminar es la misma sin importar que después haya una indicación de compás, una armadura o una clave. El grabador `Separating_line_group_engraver` es responsable de este espacio, y su elminiación reduce este espacio a cero.

```
\paper {
  indent = 0
}

\new Staff \with {
  \remove Separating_line_group_engraver
} {
  c'1 \break
  c'1
}
```



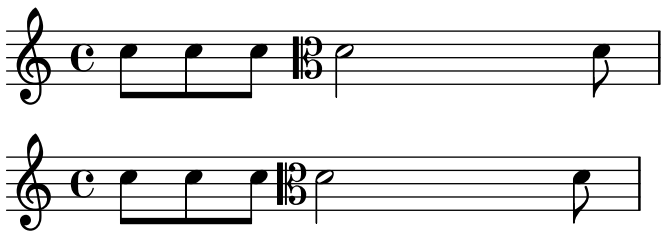


Los elementos no musicales como la indicación de compás, la armadura, la clave y las alteraciones son problemáticos en notación proporcional. Ninguno de estos elementos tiene duración rítmica, pero todos ellos consumen espacio horizontal. Las distintas partituras proporcionales abordan este problema de distinta manera.

Sería posible evitar los problemas de espaciado con las armaduras, simplemente evitando tenerlas. Esta es una opción válida pues casi todas las partituras proporcionales son música contemporánea. Lo mismo puede valer para las indicaciones de compás, especialmente para las partituras que incluyen una línea de tiempo graduada u otro gráfico. Sin embargo, estas partituras son excepcionales y casi todas las partituras proporcionales incluyen al menos unas pocas indicaciones de compás. Las claves y las alteraciones son aún más esenciales.

Así pues ¿qué estrategias existen para el espaciado de los elementos no musicales en un contexto proporcional? Una buena opción es la propiedad `strict-note-spacing` de `SpacingSpanner`. Compare las dos partituras siguientes:

```
{
  \set Score.proportionalNotationDuration = #1/16
  c''8 8 8 \clef alto d'2 d'8
}

{
  \set Score.proportionalNotationDuration = #1/16
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  c''8 8 8 \clef alto d'2 d'8
}
```



Las dos partituras son proporcionales, pero el espaciado de la primera es muy suelto a causa del cambio de clave. Sin embargo, el espaciado de la segunda partitura se mantiene estricto, porque `strict-note-spacing` está activado. La activación de esta propiedad hace que el ancho de las indicaciones de compás, armaduras, cambios de clave y alteraciones no tomen parte en el algoritmo de espaciado.

Además de los ajustes dados aquí, hay otros que aparecen con frecuencia en las partituras proporcionales.

`\override SpacingSpanner.strict-grace-spacing = ##t`
: espaciar las notas de adorno de forma estricta (véase [Positioning grace notes with floating space], página 151)

`\set tupletFullLength = ##t`
: extender los corchetes de tresillo para que marquen los dos puntos rítmicos, el de comienzo y el de final

`\override Beam.breakable = ##t`
: permitir barras divididas (véase [Beams across line breaks], página 102)

`\override Glissando.breakable = ##t`
: permitir glissandos divididos (véase [Making glissandi breakable], página 181)

`\remove Forbid_line_break_engraver`
: permitir saltos de línea incluso si un elemento musical sigue activo (véase Sección 28.1 [Saltos de línea], página 692)

Véase también

Referencia de la notación: Sección 30.2 [Sección de espaciado nueva], página 719.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

31 Encajar la música en menos páginas

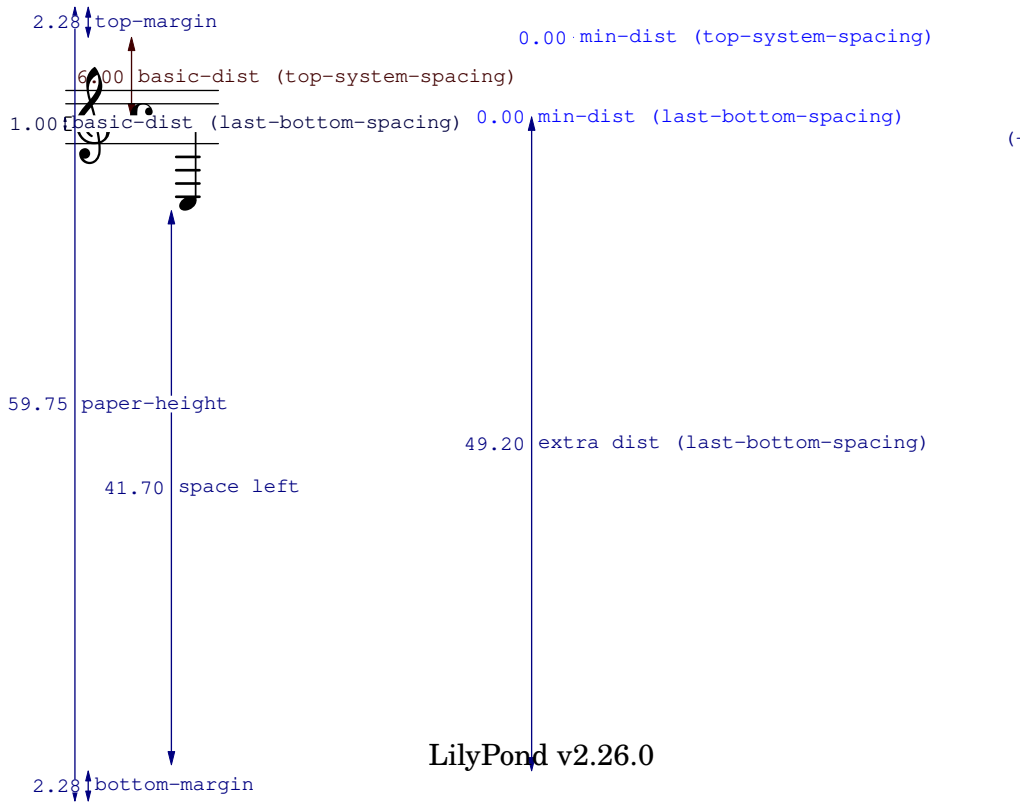
En ocasiones, podemos terminar con uno o dos pentagramas en una segunda página (o tercera, o cuarta...). Es fastidioso, especialmente cuando vemos que las páginas anteriores parecen tener espacio de sobra.

Al investigar los problemas de disposición, una herramienta de valor incalculable es `annotate-spacing`. Esta instrucción imprime los valores de un cierto número de variables de espaciado; para ver más detalles consulte la sección siguiente, Sección 31.1 [Mostrar el espaciado], página 734.

31.1 Mostrar el espaciado

Para presentar gráficamente las dimensiones de las variables de disposición vertical que pueden verse alteradas por el formato de la página, establezca `annotate-spacing` en el bloque `\paper`:

```
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
}
```



Todas las dimensiones de disposición se muestran en espacios de pentagrama, independientemente de las unidades especificadas en los bloques `\paper` o `\layout`. En el ejemplo anterior, `paper-height` tiene un valor de 59.75 espacios de pentagrama, y el `staff-size` (tamaño del pentagrama) es de 20 puntos, el valor predeterminado. Observe que:

1 punto = (25.4/72.27) mm

1 espacio de pentagrama = (`staff-size`)/4 pts

= (staff-size)/4 * (25.4/72.27) mm

En este caso, un espacio de pentagrama equivale aproximadamente a 1.757 mm. Así, la medida paper-height de 59.75 espacios de pentagrama equivale a 105 milímetros, la altura de una hoja a6 en orientación apaisada. Las parejas (*a*,*b*) son intervalos, donde *a* es el límite inferior y *b* es el límite superior del intervalo.

Véase también

Referencia de la notación: Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

Fragmentos de código: Sección "Espaciado" en *Fragmentos de código*.

31.2 Cambiar el espaciado

La salida de annotate-spacing revela las dimensiones verticales con gran detalle. Para ver más detalles acerca de la modificación de los márgenes y otras variables de diseño de la página, consulte Capítulo 26 [Disposición de la página], página 673.

Aparte de los márgenes, hay algunas otras opciones para ahorrar espacio:

- Forzar los sistemas para que se junten lo más posible entre sí (de forma que quepan la mayor cantidad de ellos en la página) y al mismo tiempo estén espaciados de forma que no exista un espacio vacío en la parte baja del papel.

```
\paper {
  system-system-spacing = #'((basic-distance . 0.1) (padding . 0))
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forzar el número de sistemas. Puede ser de utilidad de dos formas. Si tan solo se establece un valor, incluso el mismo valor que el número de sistemas que se están trazando de forma predeterminada, ello ocasionará a veces que se traten de introducir más sistemas en cada página, pues en este caso se omite un paso en que se produce una estimación, dando así un encaje mejor en cada página. Asimismo, el forzar una reducción neta en el número de sistemas puede producir el ahorro de una página adicional. Por ejemplo, si la disposición predeterminada tiene 11 sistemas, la asignación siguiente fuerza una disposición con 10 sistemas.

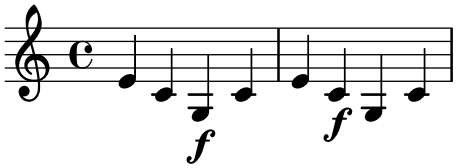
```
\paper {
  system-count = 10
}
```

- Forzar el número de páginas. Por ejemplo, la asignación siguiente fuerza una disposición con dos páginas.

```
\paper {
  page-count = 2
}
```

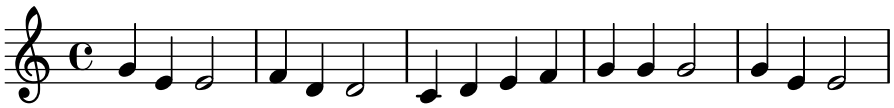
- Evitar (o reducir en número) los objetos que aumentan el tamaño vertical de un sistema. Por ejemplo, los corchetes de las repeticiones de primera y segunda vez (o repeticiones con finales alternativos) necesitan espacio adicional. Si estas casillas se reparten a lo largo de dos sistemas, ocupan más espacio que si estuvieran sobre el mismo sistema. Como otro ejemplo, las indicaciones dinámicas que se 'salen' de un sistema se pueden acercar al pentagrama:

```
\relative e' {
  e4 c g\f c
  e4 c g-\tweak X-offset -2.7 \f c
}
```



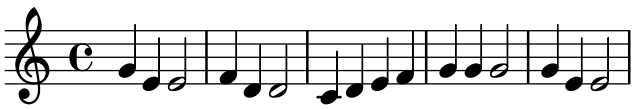
- Alterar el espaciado horizontal por medio de `SpacingSpanner`. Para ver más detalles, consulte Sección 30.3 [Cambiar el espaciado horizontal globalmente], página 720. El ejemplo siguiente ilustra el espaciado predeterminado:

```
\score {
  \relative {
    g'4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```



El ejemplo siguiente modifica `common-shortest-duration` de un valor de `1/4` (negra) a `1/2` (blanca). La negra es la nota más común y la más breve del ejemplo, por lo que al agrandar esta duración se produce un efecto de 'apretujamiento':

```
\score {
  \relative {
    g'4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.common-shortest-duration =
        \musicLength 2
    }
  }
}
```



La propiedad `common-shortest-duration` no se puede modificar de manera dinámica, por lo que se debe situar siempre dentro de un bloque `\context` de forma que se aplique a la partitura completa.

Véase también

Referencia de la notación: Capítulo 26 [Disposición de la página], página 673, Sección 30.3 [Cambiar el espaciado horizontal globalmente], página 720.

Fragmentos de código: Sección “Espaciado” en *Fragmentos de código*.

Cambiar los valores por omisión

32 Ajuste fino de la salida

El objetivo del diseño de LilyPond es proporcionar la más alta calidad de los resultados, de forma predeterminada. A pesar de ello, podría tener que cambiar este resultado predeterminado. La disposición sobre el papel se controla a través de un amplio número de 'botones e interruptores' llamados en su conjunto *propiedades*. En el Manual de aprendizaje podemos encontrar una introducción en forma de tutorial al acceso y modificación de estas propiedades, véase Sección "Trucar la salida" en *Manual de Aprendizaje*. Éste debería leerse en primer lugar. Este capítulo cubre un terreno similar, pero con un estilo más adecuado para un manual de referencia.

La descripción definitiva de los controles que están disponibles para su ajuste fino están en un documento aparte: la *Referencia de Funcionamiento Interno*. Dicho manual relaciona todas las variables, funciones y opciones que se encuentran disponibles en LilyPond.

Internamente, LilyPond utiliza el lenguaje Scheme (un dialecto de Lisp) para aportar la infraestructura. La sobreescritura de las decisiones de disposición da acceso efectivo a las interioridades del programa, lo que requiere código de Scheme como entrada. Los elementos de Scheme se inician dentro de un archivo `.ly` con el símbolo de cuadradillo '`#`'.[1]

[1] Sección "Tutorial de Scheme" en *Extender* contiene un breve tutorial sobre la introducción de números, listas, cadenas y símbolos en Scheme.

33 Contextos de interpretación

Esta sección explica qué son los contextos y cómo modificarlos.

Véase también

Manual de aprendizaje: Sección “Contextos y grabadores” en *Manual de Aprendizaje*.

Archivos de inicio: `ly/engraver-init.ly`, `ly/performer-init.ly`.

Fragmentos de código: Sección “Contextos y grabadores” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “Engravers and Performers” en *Referencia de Funcionamiento Interno*.

33.1 Explicación de los contextos

Los contextos se disponen de forma jerárquica.

33.1.1 Definiciones de salida - estructura de los contextos

Esta sección explica la relevancia de las definiciones de salida cuando se trabaja con los *contextos*. Más adelante ofrecemos ejemplos de definiciones de salida reales (véase Sección 33.5.1 [Cambiar todos los contextos del mismo tipo], página 753).

Aunque la música escrita en un archivo haga referencia a tipos y nombres de contexto, los contextos se crean solamente cuando la música se está realmente interpretando. LilyPond interpreta la música bajo el control de una *definición de salida* y puede hacerlo así para varias definiciones de salida distintas, dando como resultado distintas salidas también. La definicion de salida que corresponde a la impresión de la música se especifica mediante `\layout`.

Una definición de salida mucho más sencilla que se usa para la producción de MIDI se especifica mediante `\midi`. Internamente, LilyPond utiliza varias otras definiciones de salida, como cuando se usa el combinador de particellas (Sección 5.2.5 [Combinación automática de las partes], página 235) o se crean fragmentos de notas guía citadas (Sección 6.3.2 [Citar otras voces], página 268).

Las definiciones de salida establecen la relación entre los contextos así como sus respectivos valores predeterminados. Aunque la mayor parte de los cambios se suelen hacer dentro de un bloque `\layout`, los ajustes de valores relacionados con el MIDI solamente tienen efecto cuando se hacen dentro de un bloque `\midi` block.

Algunos ajustes afectan a varias salidas: por ejemplo, si se desactiva el barrado automático, `autoBeaming`, dentro de algún contexto, las barras cuentan como melismas en lo que respecta a la correspondencia entre la música y la letra, véase Sección 9.1.4 [Duración automática de las sílabas], página 353. Esta correspondencia se hace tanto para la salida impresa como para el MIDI. Si los cambios hechos sobre el `autoBeaming` dentro de la definición de contexto de un bloque `\layout` no se repiten dentro del correspondiente bloque `\midi`, la letra y la música dejarán de estar sincronizadas dentro del MIDI.

Véase también

Archivos instalados: `ly/engraver-init.ly`. `ly/performer-init.ly`.

33.1.2 `Score`: el contexto maestro

Este es el contexto de notación del nivel más alto que hay.[1] Ningún otro contexto puede contener a un contexto `Score`. Este contexto se encarga de gestionar las indicaciones de compás. También

[1] Existe un contexto llamado `Global` que está incluso a un nivel más alto que `Score`, y que es el punto de entrada fijo programado para LilyPond. Sin embargo, se supone que este contexto nunca se modifica; es necesario solamente si vamos a implementar un grabador en Scheme y en caso contrario se puede simplemente ignorar.

garantiza que elementos como las claves, el compás y las armaduras de tonalidad están alineados entre los distintos pentagramas.

No podemos instanciar un contexto `Score` explícitamente (ya que no está contenido dentro de ningún otro contexto). Se crea una instancia automáticamente cuando se procesa una definición de salida (un bloque `\score` o `\layout`).

Se establece por parte de `Timing_translator` un alias llamado `Timing` dentro de cualquier contexto en que se inicializa, y las variables de la cuenta de tiempo se copian a partir del lugar en que se haya establecido `Timing` previamente. El alias en el nivel de `Score` ofrece una diana para inicializar las variables de `Timing` dentro de las definiciones de presentación antes de que se haya ejecutado cualquier `Timing_translator`.

33.1.3 Contextos del nivel superior: contenedores de pentagramas

`StaffGroup (grupo de pentagramas)`
: Agrupa pentagramas y añade un corchete en la parte izquierda, formando un grupo. Además, las líneas divisorias de los pentagramas contenidos se conectan verticalmente.

`ChoirStaff (sistema de coro)`
: Idéntico a `StaffGroup` excepto que las barras de compás de los pentagramas contenidos no se conectan verticalmente.

`GrandStaff (sistema de piano)`
: Conectar las pautas verticalmente mediante una llave en el lado izquierdo. Las barras de compás de las pautas contenidas también se conectan verticalmente.

`PianoStaff (sistema de piano)`
: Igual que `GrandStaff`, pero los pentagramas se eliminan siempre juntos, nunca por separado.

`OneStaff (una pauta)`
: Ofrece un eje común para las pautas contenidas, haciendo que aparezcan dentro del mismo espacio vertical. Esto puede ser útil para tipografiar pautas de distinto tipo sucesivamente, o para cambiar temporalmente el carácter de una pauta, o para superponerle otra distinta. Los mejores resultados se obtienen frecuentemente cuando se usa con `\stopStaff` y `\startStaff`.

`VaticanaScore (partitura de Vaticana)`
: Contexto del nivel superior que suple a `Score` para canto gregoriano en el estilo de la edición vaticana. Comparado con `Score`, cambia el color de las líneas de la pauta a rojo, usa un espaciado compacto y suprime los números de compás.

`ChordGridScore (rejilla de acordes)`
: Contexto del nivel superior que suple a `Score` en notación de rejilla de acordes. Comparado con `Score`, usa notación proporcional, y tiene otros ajustes como la eliminación de los números de compás, por ejemplo.

`StandaloneRhythmScore (partitura rítimica autocontenida)`
: Un contexto del mismo nivel que `Score` para su uso por parte de `\markup \rhythm`.

33.1.4 Contextos de nivel intermedio. Pentagramas

`Staff (pauta o pentagrama)`
: Maneja claves, barras de compás, tonalidades y alteraciones accidentales. Puede contener contextos de `Voice`.

`RhythmicStaff` (pauta rítmica)
: Como `Staff`, pero para imprimir ritmos. Al imprimir se ignoran las alturas de las notas; las notas se imprimen sobre una línea. La salida de MIDI conserva las alturas sin modificación.

`TabStaff` (pauta de tablatura)
: Contexto para generar tablaturas. Solo acepta contextos `TabVoice` y maneja adecuadamente el espaciado de líneas, la clave de tablatura, etc.

`DrumStaff` (pauta de percusión)
: Maneja el tipografiado para instrumentos de percusión. Puede contener contextos `DrumVoice`.

`VaticanaStaff` (pauta de Vaticana)
: Un tipo de `Staff` para la tipografía de canto gragoriano en un estilo de notación que se asemeja al de la *Editio Vaticana*.

`MensuralStaff` (pauta de notación mensural)
: Igual que `Staff`, excepto que está diseñado para tipografiar piezas en estilo mensural.

`PetrucciStaff` (pauta de Petrucci)
: Una variante de `Staff` que se asemeja al tipografiado mensural del *Harmonices Musices Odhecaton* (Venecia, 1501) de Ottaviano Petrucci.

`KievanStaff` (pauta de canto kievano)
: Igual que el contexto `Staff` pero adaptado para la tipografía de piezas en el estilo del canto kievano.

`GregorianTranscriptionStaff` (pauta de transcripción de canto gregoriano)
: Pauta para la notación de canto gregoriano en estilo moderno.

`ChordGrid` (rejilla de acordes)
: Crea notación de rejilla de acordes (hojas guía) en cifrado americano. Este contexto siempre forma parte de un contexto `ChordGridScore`.

`StandaloneRhythmStaff` (pauta rítmica autocontenida)
: Un contexto del mismo nivel que `Staff`, para su uso por parte de `\markup \rhythm`.

`FretBoards` (diagramas de trastes)
: Contexto para la impresión de diagramas de trastes.

`Devnull` (contexto nulo)
: Descartar discretamente cualquier información musical que se le dé a este contexto.

33.1.5 Contextos del nivel más bajo. Voces

Los contextos del mismo nivel que `Voice` dan un valor inicial a ciertas propiedades e inician los grabadores correspondientes. Un contexto del nivel más bajo es aquel que no tiene un contexto descendiente predeterminado `\defaultchild`. Aunque es posible hacer que pueda aceptar o contener subcontextos, éstos solo se pueden crear e introducir de forma explícita.

`Voice`
: Corresponde a una voz sobre un pentagrama. este contexto maneja la conversión de las indicaciones dinámicas, plicas, barras, subíndices y superíndices, ligaduras de expresión y de unión, y silencios. Tenemos que crear instancias explícitas de este contexto si necesitamos varias voces en el mismo pentagrama.

`VaticanaVoice` (voz de Vaticana)
: Una variante de `Voice` para el tipografiado de canto gregoriano en un estilo de notación que se parece a la *Editio Vaticana*.

`MensuralVoice (voz mensural)`
: Lo mismo que `Voice`, con modificaciones para el tipografiado de piezas en estilo mensural.

`PetrucciVoice (voz de Petrucci)`
: Una vairante de `Voice` que se parece a la tipografía mensural del *Harmonices Musices Odhecaton* (Venecia, 1501) de Ottaviano Petrucci.

`KievanVoice (voz de kievano)`
: Igual que el contexto `Voice` pero adaptado a la tipografía de piezas en el estilo del canto kievano.

`GregorianTranscriptionVoice (voz de transcripción gregoriana)`
: Contexto de voz para la notación de canto gregoriano en estilo moderno.

`Lyrics (letra)`
: Corresponde a una voz con letra. Maneja la impresión de una sola línea de letra.

`VaticanaLyrics (letra de Vaticana)`
: Igual que el contexto `Lyrics`, excepto que provee un estilo de enguionado en el estilo de la *Editio Vaticana*: un guion único alineado a la izquierda entre cada dos sílabas.

`GregorianTranscriptionLyrics (letra de transcripción gregoriana)`
: Un contexto de letra para la notación de canto gregoriano en estilo moderno.

`DrumVoice (voz de percusión)`
: Una voz sobre una pauta de percusión.

`FiguredBass (bajo cifrado)`
: El contexto en que los objetos `BassFigure` se crean a partir de la entrada escrita en el modo `\figuremode`.

`TabVoice (voz de tablatura)`
: El contexto de voz utilizado dentro de un contexto `TabStaff`. Se suele dejar que se cree implícitamente.

`CueVoice (voz de notas guía)`
: El contexto de voz que se utiliza para dibujar notas de tamaño reducido, con el principal objetivo de añadir notas guía de un pentagrama a otro, véase Sección 6.3.3 [Formateo de las notas guía], página 272. Normalmente se deja que se cree implícitamente.

`ChordNames (nombres de acorde)`
: Tipografía nombres de acordes.

`NoteNames (nombres de nota)`
: Tipografía nombres de nota como su nombre indica.

`NullVoice (voz nula)`
: Para la alineación de letras sin imprimir las notas.

`Devnull (contexto nulo)`
: Descartar discretamente cualquier información musical que se le dé a este contexto.

`Dynamics (matices dinámicos)`
: Alberga una línea única de indicaciones de matiz dinámico, centrada entre las pautas que rodean al propio contexto.

`StandaloneRhythmVoice (voz rítmica autocontenida)`
: Un contexto en el mismo nivel que `Voice` para su uso por parte de `\markup \rhythm`.

`Bottom (inferior)`
: Es un contexto genérico del nivel inferior, aceptado por todos los contextos de nivel intermedio. Se puede usar para situaciones en que la misma música tenga que aparecer, por ejemplo, en un contexto de `Staff` y en otro de `TabStaff`.

33.2 Crear y referenciar contextos

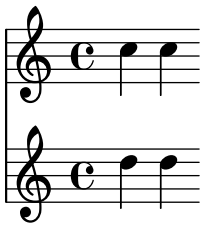
LilyPond crea automáticamente contextos de nifel inferior si se encuentra una expresión musical antes de que exista un contexto adecuado, pero normalmente esto sólo funciona bien para partituras sencillas o fragmentos musicales como los que aparecen en la documentación. Para partituras más complejas, se recomienda especificar explícitamente todos los contextos con las instrucciones `\new` o `\context`. La sintaxis de estas dos instrucciones es muy similar:

```
[\new | \context] Contexto [= nombre] [expresión_musical]
```

donde se puede especificar `\new` o `\context`. *Contexto* es el tipo de contexto que se desea crear, *nombre* es un nombre opcional que se da al contexto concreto que se está creando, y *expresión_musical* es una sola expresión musical que será interpretada por los complementos grabadores y reproductores dentro de este contexto.

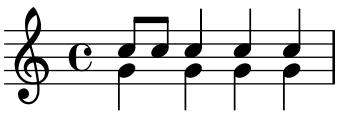
El prefijo `\new` sin ningún nombre se usa con frecuencia para crear partituras con muchos pentagramas:

```
<<
  \new Staff \relative {
    % leave the Voice context to be created implicitly
    c''4 c
  }
  \new Staff \relative {
    d''4 d
  }
>>
```



y para introducir varias voces dentro de un solo pentagrama:

```
\new Staff <<
  \new Voice \relative {
    \voiceOne c''8 c c4 c c
  }
  \new Voice \relative {
    \voiceTwo g'4 g g g
  }
>>
```

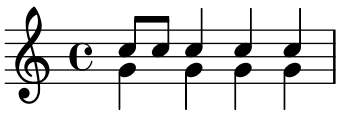


`\new` debería usarse siempre para especificar contextos sin nombre.

La diferencia entre \new y \context se encuentra en la acción que se realiza:

- \new con un nombre o sin él, siempre crea un contexto nuevo y distinto, incluso si ya existe un contexto con el mismo nombre:

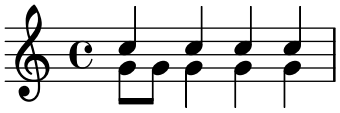
```
\new Staff <<
  \new Voice = "A" \relative {
    \voiceOne c''8 c c4 c c
  }
  \new Voice = "A" \relative {
    \voiceTwo g'4 g g g
  }
>>
```



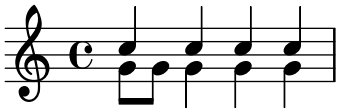
- \context con un nombre especificado, crea un contexto nuevo solamente si no existe ya un contexto del mismo tipo y con el mismo nombre, dentro de la misma jerarquía de contextos. En caso contrario, se toma como referencia a dicho contexto creado previamente, y su expresión musical se pasa a este contexto para su interpretación.

  Los contextos con nombre pueden ser útiles en casos especiales como la letra de las canciones o los bajos cifrados, véase Sección 9.2.1 [Trabajar con letra y variables], página 362, y Sección "Plantillas de conjuntos vocales" en *Manual de Aprendizaje* para el primer caso, y Sección 15.3.3 [Imprimir el bajo cifrado], página 533, para el segundo. De manera más general, una aplicación de los contextos con nombre es la separación entre la disposición de la partitura y el contenido musical. Son válidas culaquiera de las dos formas siguientes:

```
\score {
  <<
    % score layout
    \new Staff <<
      \new Voice = "one" {
        \voiceOne
      }
      \new Voice = "two" {
        \voiceTwo
      }
    >>

    % musical content
    \context Voice = "one" {
      \relative {
        c''4 c c c
      }
    }
    \context Voice = "two" {
      \relative {
        g'8 g g4 g g
      }
    }
  >>
}
```



```
\score {
  <<
    % score layout
    \new Staff <<
      \context Voice = "one" {
        \voiceOne
      }
      \context Voice = "two" {
        \voiceTwo
      }
    >>

    % musical content
    \context Voice = "one" {
      \relative {
        c''4 c c c
      }
    }
    \context Voice = "two" {
      \relative {
        g'8 g g4 g g
      }
    }
  >>
}
```



De manera alternativa, se pueden utilizar variables con un efecto similar. Véase Sección "Organizar las piezas mediante variables" en *Manual de Aprendizaje*.

- `\context` sin ningún nombre corresponderá con el primer contexto que se encuentre entre los creados previamente que sean del mismo tipo dentro de la misma jerarquía de contextos, incluso si tiene nombre, y su expresión musical se pasará a dicho contexto para su interpretación. Esta forma rara vez es útil. Sin embargo, `\context` sin nombre y sin expresión musical se usa para establecer el contexto en que se ejecuta un procedimiento de Scheme especificado con `\applyContext`:

```
\new Staff \relative {
  c'1
  \context Timing
  \applyContext #(lambda (ctx)
                   (newline)
                   (display (ly:context-current-moment ctx)))
  c1
}
```

Un contexto debe tener un nombre si se va a hacer referencia a él más tarde, por ejemplo cuando se asocia la letra con la música:

```
\new Voice = "tenor" música
...
\new Lyrics \lyricsto "tenor" letra
```

Para ver más detalles sobre la asociación de letra y música, véase Sección 9.1.4 [Duración automática de las sílabas], página 353.

Las propiedades de todos los contextos de un tipo en particular se pueden modificar dentro de un bloque `\layout` (con una sintaxis diferente), véase Sección 33.5.1 [Cambiar todos los contextos del mismo tipo], página 753. Esta construcción también ofrece una forma de mantener las instrucciones de disposición separadas del contenido musical. Si se va a modificar un solo contexto, debe usarse un bloque `\with`, véase Sección 33.5.2 [Cambiar solamente un contexto determinado], página 756.

Véase también

Manual de aprendizaje: Sección "Organizar las piezas mediante variables" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 33.5.2 [Cambiar solamente un contexto determinado], página 756, Sección 9.1.4 [Duración automática de las sílabas], página 353.

33.3 Mantener vivos los contextos

Normalmente los contextos finalizan en el primer momento musical en que no tienen nada que hacer. Así, los contextos de `Voice` mueren tan pronto como ya no contienen ningún evento, los contextos de `Staff` mueren tan pronto como todos los contextos de `Voice` que contenían ya no contengan ningún evento, etc. Esto puede ocasionar dificultades si se tiene que hacer referencia a contextos anteriores que ya han desaparecido, por ejemplo, al cambiar de pentagramas con instrucciones `\change`, asociar letra con una voz mediante instrucciones `\lyricsto`, o cuando se añaden eventos musicales adicionales a un contexto anterior.

Existe una excepción a esta regla general: dentro de una construcción `{...}` (música secuencial), la noción que la construcción tiene del "contexto actual" descenderá un nivel cada vez que un elemento de la secuencia finaliza en un subcontexto del contexto anterior. Esto evita la creación espúrea de contextos implícitos en ciertas situaciones, pero significa que el primer contexto dentro del que se desciende, se va a mantener vivo hasta el final de la expresión.

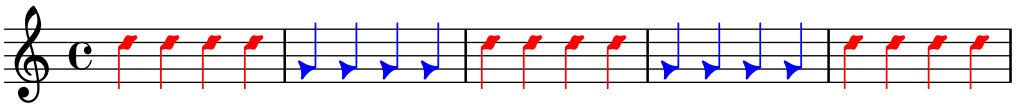
Como contraste, los contextos de una expresión hecha con la construcción `<<...>>` (música simultánea) no se prolongan, por lo que si una instrucción que crea un contexto se encierra en otro par de `<<...>>`, se impedirá que el contexto persista durante toda la duración de la secuencia `{...}` que lo contiene.

Cualquier contexto se puede mantener vivo si nos aseguramos de que tiene algo que hacer en cualquier momento musical dado. Los contextos de `Staff` se mantienen con vida si nos aseguramos de que una de sus voces se mantiene viva. Una manera de hacerlo es añadir silencios de separación a una voz en paralelo con la música real. Éstos deben añadirse a todos y cada uno de los contextos de `Voice` que se hayan de mantener vivos. Si se van a usar esporádicamente varias voces, es más seguro mantenerlas todas vivas en lugar de tratar de confiar en las excepciones que hemos mencionado arriba.

En el ejemplo siguiente, tanto la voz A como la voz B se mantienen vivas de esta manera durante la duración de la pieza:

```
musicA = \relative { d''4 d d d }
musicB = \relative { g'4 g g g }
keepVoicesAlive = {
  <<
    \new Voice = "A" { s1*5 }  % keep voice 'A' alive for 5 bars
```

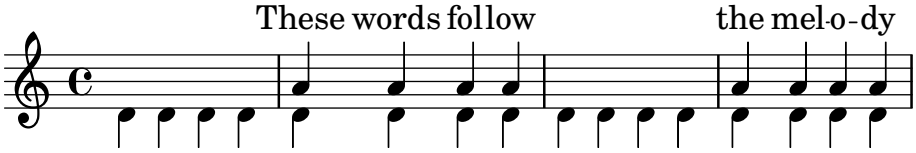
```
    \new Voice = "B" { s1*5 }  % keep voice 'B' alive for 5 bars
  >>
}

music = {
  \context Voice = "A" {
    \voiceOneStyle \musicA
  }
  \context Voice = "B" {
    \voiceTwoStyle \musicB
  }
  \context Voice = "A" { \musicA }
  \context Voice = "B" { \musicB }
  \context Voice = "A" { \musicA }
}

\score {
  \new Staff <<
    \keepVoicesAlive
    \music
  >>
}
```



El ejemplo siguiente muestra cómo se puede escribir una línea melódica esporádica con letra utilizando este enfoque. Por supuesto, en una situación real la melodía y el acompañamiento consistirían en varias secciones diferentes.

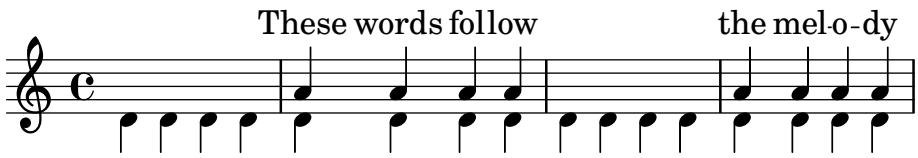
```
melody = \relative { a'4 a a a }
accompaniment = \relative { d'4 d d d }
words = \lyricmode { These words fol -- low the mel -- o -- dy }

\score {
  <<
    \new Staff = "music" {
      <<
        \new Voice = "melody" {
          \voiceOne s1*4  % keep voice 'melody' alive for 4 bars
        }
        {
          \new Voice = "accompaniment" {
            \voiceTwo \accompaniment
          }
          <<
            \context Voice = "melody" { \melody }
            \context Voice = "accompaniment" { \accompaniment }
          >>
          \context Voice = "accompaniment" { \accompaniment }
          <<
            \context Voice = "melody" { \melody }
```

```
            \context Voice = "accompaniment" { \accompaniment }
          >>
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = "music" }
    \lyricsto "melody" { \words }
  >>
}
```



Una forma alternativa, que podría resultar mejor en muchas situaciones, es mantener con vida la línea melódica simplemente incluyendo notas espaciadoras para alinearla correctamente con el acompañamiento:

```
melody = \relative {
  s1  % skip a bar
  a'4 a a a
  s1  % skip a bar
  a4 a a a
}
accompaniment = \relative {
  d'4 d d d
  d4 d d d
  d4 d d d
  d4 d d d
}
words = \lyricmode { These words fol -- low the mel -- o -- dy }

\score {
  <<
    \new Staff = "music" {
      <<
        \new Voice = "melody" {
          \voiceOne \melody
        }
        \new Voice = "accompaniment" {
          \voiceTwo \accompaniment
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = "music" }
    \lyricsto "melody" { \words }
  >>
}
```



33.4 Modificar los complementos (plug-ins) de contexto

Los contextos de notación (como `Score` y `Staff`) no sólo almacenan propiedades, también contienen «plug-ins» o complementos llamados *grabadores* que crean elementos de notación. Por ejemplo, el contexto `Voice` contiene un grabador `Note_heads_engraver` que crea las cabezas de nota y el contexto `Staff` contiene un grabador `Key_engraver` que crea la armadura.

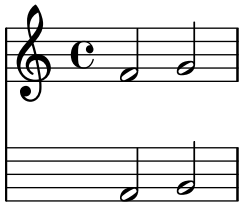
Para ver una descripción completa de todos y cada uno de los complementos, consulte Sección "Engravers and Performers" en *Referencia de Funcionamiento Interno*. Cada contexto que se describe en Sección "Contexts" en *Referencia de Funcionamiento Interno* relaciona los grabadores que se usan para ese contexto.

Puede ser de utilidad jugar un poco con estos complementos. Se hace iniciando un contexto nuevo con `\new` o `\context` y modificándolo:

```
\new contexto \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ...música...
}
```

donde cada uno de los '...' deben ser el nombre de un grabador. Aquí tenemos un ejemplo sencillo que suprime los grabadores `Time_signature_engraver` y `Clef_engraver` de un contexto `Staff`.

```
<<
  \new Staff \relative {
    f'2 g
  }
  \new Staff \with {
     \remove Time_signature_engraver
     \remove Clef_engraver
  } \relative {
    f'2 g2
  }
>>
```



En el segundo pentagrama no hay indicación de compás ni clave. Éste es un método bastante rudimentario de hacer que desaparezcan los objetos porque afecta a todo el pentagrama. Este

método también afecta al espaciado, lo que puede ser deseable o no serlo. Se muestran métodos más sofisticados para quitar objetos en Sección "Visibilidad y color de los objetos" en *Manual de Aprendizaje*.

Advertencias y problemas conocidos

El orden en que los grabadores se especifican es el orden en que se llaman para realizar su tarea de procesamiento. Normalmente, el orden en que se especifican los grabadores no tiene importancia, pero en algunos casos especiales sí la tiene, por ejemplo donde un grabador escribe una propiedad y otro la lee, o donde un grabador crea un groby otro debe procesarlo.

Las siguientes ordenaciones son importantes:

- el grabador de compases `Bar_engraver` debe ir normalmente en primer lugar,
- el grabador de digitaciones `New_fingering_engraver` debe ir antes del grabador `Script_column_engraver` de columnas de inscripciones,
- el `Timing_translator` debe ir antes del grabador de números de compás `Bar_number_engraver`.

Véase también

Archivos instalados: `ly/engraver-init.ly`.

33.5 Cambiar los valores por omisión de los contextos

Se pueden cambiar las propiedades de contexto y de grob con las instrucciones `\set` y `\override`, véase Capítulo 35 [Modificar las propiedades], página 768. Estas instrucciones crean eventos musicales, haciendo que los cambios tengan efecto en el punto temporal en que la música se está procesando.

Por contra, esta sección explica la forma de cambiarlos valores *predeterminados* de las propiedades de contexto y de grob en el momento en que se crea el contexto. Existen dos formas de hacerlo. Una modifica los valores predeterminados en todos los contextos de un tipo dado, y el otro modifica los valores predeterminados solamente en una instancia concreta de un contexto.

33.5.1 Cambiar todos los contextos del mismo tipo

Los ajustes de contexto predeterminados que se han de usar para la composición tipográfica dentro de `Score`, `Staff`, `Voice` y otros contextos se pueden especificar en un bloque `\context` dentro de cualquier bloque `\layout`.

Los ajustes para la salida MIDI, al contrario que para la composición tipográfica, se tendrán que especificar aparte en bloques `\midi` (véase Sección 33.1.1 [Definiciones de salida - estructura de los contextos], página 742).

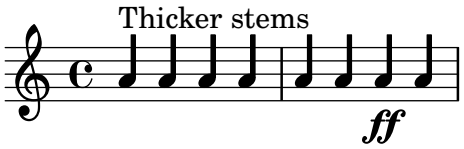
El bloque `\layout` se debe colocar dentro del bloque `\score` al que se aplica, después de la música.

```
\layout {
  \context {
    \Voice
    [ajustes de contexto para todos los contextos Voice]
  }
  \context {
    \Staff
    [ajustes de contexto para todos los contextos Staff]
  }
}
```

Se pueden especificar los siguientes tipos de ajustes:

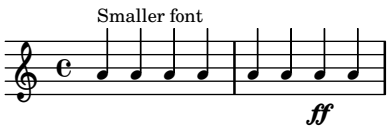
- Una instrucción `\override`, pero omitiendo el nombre del contexto

```
\score {
  \relative {
    a'4^"Thicker stems" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      \override Stem.thickness = 4.0
    }
  }
}
```



- Estableciendo una propiedad de contexto directamente

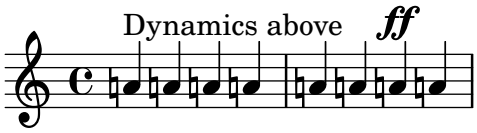
```
\score {
  \relative {
    a'4^"Smaller font" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      fontSize = -4
    }
  }
}
```



- Una instrucción predefinida tal como `\dynamicUp` o una expresión musical como `\accidentalStyle dodecaphonic`.

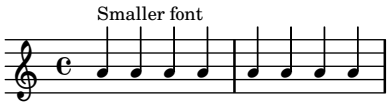
```
\score {
  \relative {
    a'4^"Dynamics above" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Voice
      \dynamicUp
    }
    \context {
      \Staff
```

```
      \accidentalStyle dodecaphonic
    }
  }
}
```



- Una variable definida por el usuario que contenga un bloque `\with`; para ver detalles acerca del bloque `\with`, véase Sección 33.5.2 [Cambiar solamente un contexto determinado], página 756.

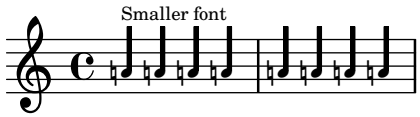
```
StaffDefaults = \with {
  fontSize = -4
}

\score {
  \new Staff {
    \relative {
      a'4^"Smaller font" a a a
      a4 a a a
    }
  }
  \layout {
    \context {
      \Staff
      \StaffDefaults
    }
  }
}
```



Las instrucciones de ajuste de propiedades se pueden disponer dentro de un bloque `\layout` sin que estén encerradas en un bloque `\context`. Tales ajustes son equivalentes a incluir las mismas instrucciones de ajuste de propiedades al comienzo de cada uno de los contextos del tipo especificado. Si no se especifica ningún contexto, *todos y cada uno* de los contextos del nivel inferior quedan afectados (véase Sección 33.1.5 [Contextos del nivel más bajo. Voces], página 744). La sintaxis de una instrucción de ajuste de propiedades dentro de un bloque `\layout` es la misma que si la misma instrucción estuviera escrita en el propio flujo musical.

```
\score {
  \new Staff {
    \relative {
      a'4^"Smaller font" a a a
      a4 a a a
    }
  }
  \layout {
    \accidentalStyle dodecaphonic
    \set fontSize = -4
```

```
    \override Voice.Stem.thickness = 4.0
  }
}
```



33.5.2 Cambiar solamente un contexto determinado

Las propiedades de contexto de una única instancia de contexto pueden cambiarse dentro de un bloque `\with`. Todas las demás instancias de contexto del mismo tipo retienen los ajustes predeterminados que LilyPond tiene programados y que se modifican por parte de cualquier bloque `\layout` que se encuentre dentro del ámbito. El bloque `\with` se debe situar inmediatamente después de las instrucciones `\new` *context-type*:

```
\new Staff \with { [ajustes de contexto para esta instancia de contexto solamente] }
{
  ...
}
```

De forma alternativa, si la música se escribe usando la forma corta de las instrucciones de especificación de modo, como por ejemplo `\chords` en lugar de `\chordmode`, la instrucción `\with` se debe colocar inmediatamente después de la instrucción que especifica el modo:

```
\chords \with {
  [ajustes de contexto para esta instancia de contexto (implícita) solamente] } {
  ...
}
```

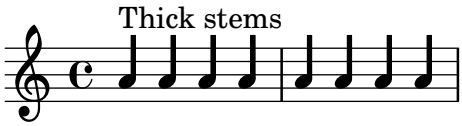
dado que es el contexto implícito creado por estas formas breves el que se quiere modificar. Se aplica la misma consideración a las otras formas cortas de especificar el modo de entrada (`\drums`, `\figures`), véase Capítulo 19 [Modos de entrada], página 587.

Dado las modificaciones de contexto especificadas en bloques `\with` están dentro de la música, afectarán a *todas* las salidas (tipografía *y también* el MIDI), a diferencia de los cambios que se hacen dentro de una definición de salida.

Se pueden especificar los siguientes tipos de ajustes:

- Una instrucción `\override`, pero omitiendo el nombre del contexto

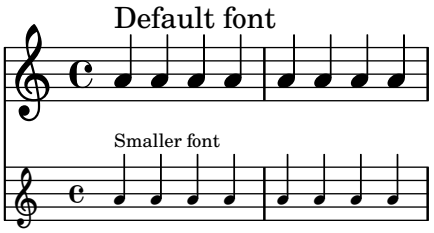
```
\score {
  \new Staff {
    \new Voice \with { \override Stem.thickness = 4.0 } {
      \relative {
        a'4^"Thick stems" a a a
        a4 a a a
      }
    }
  }
}
```



- Estableciendo una propiedad de contexto directamente

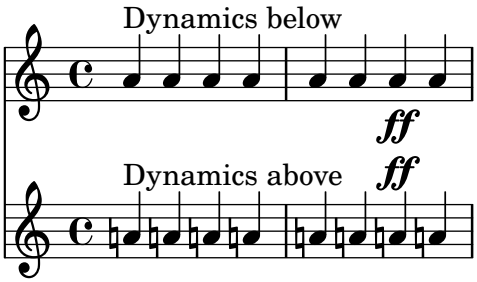
```
\score {
```

```
  <<
    \new Staff {
      \relative {
        a'4^"Default font" a a a
        a4 a a a
      }
    }
    \new Staff \with { fontSize = -4 } {
      \relative {
        a'4^"Smaller font" a a a
        a4 a a a
      }
    }
  >>
}
```



- Una instrucción predefinida tal como \dynamicUp.

```
\score {
  <<
    \new Staff {
      \new Voice {
        \relative {
          a'4^"Dynamics below" a a a
          a4 a a\ff a
        }
      }
    }
    \new Staff \with { \accidentalStyle dodecaphonic } {
      \new Voice \with { \dynamicUp } {
        \relative {
          a'4^"Dynamics above" a a a
          a4 a a\ff a
        }
      }
    }
  >>
}
```



Véase también

Referencia de la notación: Capítulo 19 [Modos de entrada], página 587,

33.5.3 Orden de precedencia

El valor de una propiedad que se aplica en un momento determinado se determina de la siguiente forma:

- si está en efecto una instrucción `\override` o `\set` dentro del flujo musical, se usa dicho valor,
- de lo contrario, se usa el valor predeterminado tomado de un enunciado `\with` en las instrucciones de inicio del contexto,
- de lo contrario, se usa el valor tomado del bloque `\context` más reciente que corresponda dentro de los bloques `\layout` o `\midi`,
- de lo contrario se usa el valor predeterminado que LilyPond lleva preprogramado.

Véase también

Manual de aprendizaje: Sección "Modificar las propiedades de los contextos" en *Manual de Aprendizaje*.

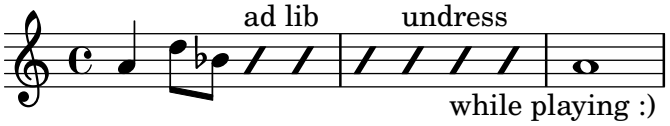
Referencia de la notación: Sección 33.1 [Explicación de los contextos], página 742, Sección 33.1.5 [Contextos del nivel más bajo. Voces], página 744, Sección 35.2 [Las instrucciones `\set` y `\unset`], página 768, Sección 35.3 [Las instrucciones `\override` y `\revert`], página 770, Sección 27.1 [El bloque `\layout`], página 686.

33.6 Definir contextos nuevos

Panorámica de la definición de contextos

Los contextos específicos, como `Staff` y `Voice`, están construidos a partir de bloques sencillos. Es posible crear nuevos tipos de contextos con combinaciones distintas de añadidos grabadores.

El siguiente ejemplo muestra cómo construir un tipo diferente de contexto de `Voice` partiendo de cero. Será parecido a `Voice`, pero imprime solamente cabezas centradas en forma de barra inclinada. Se puede usar para indicar improvisación en piezas de jazz,



Estos ajustes se definen dentro de un bloque `\context` que a su vez está dentro de un bloque `\layout`.

```
\layout {
  \context {
    ...
  }
}
```

En el siguiente análisis, la entrada de ejemplo que se muestra debe ir en el lugar de los puntos suspensivos `...` del fragmento anterior.

En primer lugar es necesario definir un nombre para el nuevo contexto:

```
\name ImproVoice
```

Debido a que es parecido al contexto `Voice`, queremos órdenes que funcionen dentro de los contextos `Voice` (existentes) para que siga funcionando. Esto se consigue dando al contexto nuevo un alias de `Voice`,

```
\alias Voice
```

El contexto imprimirá notas y textos explicativos, por ello tenemos que añadir los grabadores que aportan esta funcionalidad, y además el grabador que agrupa las notas, plicas y silencios que están en el mismo momento musical en forma de columnas,

```
\consists Note_heads_engraver
\consists Text_engraver
\consists Rhythmic_column_engraver
```

Las cabezas de todas las notas se deben situar sobre la línea central.

```
\consists Pitch_squash_engraver
squashedPosition = 0
```

El grabador `Pitch_squash_engraver` modifica las cabezas de nota (creadas por el grabador `Note_heads_engraver`) y establece sus posiciones verticales al valor de `squashedPosition`, en este caso `0`, la línea central.

Las notas deberían parecer barras inclinadas sin plica.

```
\override NoteHead.style = #'slash
\hide Stem
```

Todos estos complementos o plug-ins tienen que comunicarse bajo el control del contexto. Los mecanismos con el que se comunican los contextos se establecen mediante la declaración del `\type` (tipo) del contexto. Dentro de un bloque `\layout`, casi todos los contextos serán del tipo `Engraver_group`. Algunos contextos especiales y los contextos de los bloques `\midi` usan otros tipos. La copia y la modificación de una definición de contexto existente también cumplimentan el tipo. Como este ejemplo crea una definición partiendo de cero, tiene que ser especificada explícitamente.

```
\type Engraver_group
```

Al juntarlo todo, obtenemos

```
\context {
  \name ImproVoice
  \type Engraver_group
  \consists Note_heads_engraver
  \consists Text_engraver
  \consists Rhythmic_column_engraver
  \consists Pitch_squash_engraver
  squashedPosition = 0
  \override NoteHead.style = #'slash
  \hide Stem
  \alias Voice
}
```

Los contextos dan lugar a jerarquías. Queremos poner el contexto `ImproVoice` dentro del contexto `Staff`, igual que los contextos de voz normales. Por tanto, modificamos la definición de `Staff` con la instrucción `\accepts` (acepta).

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Con frecuencia, al reutilizar una definición de contexto existente, el contexto resultante se puede usar en cualquier lugar donde el contexto original hubiera podido hacerlo. Si hacemos

```
\layout {
  ...
  \inherit-acceptability hacia desde
}
```

conseguimos tener contextos del tipo *to* aceptados por todos los contextos que asimismo aceptan *from*. Por ejemplo, usar

```
\layout {
  ...
  \inherit-acceptability ImproVoice Voice
}
```

añade un `\accepts` para `ImproVoice` tanto a la definición de `Staff` como a la de `RhythmicStaff`.

Lo opuesto a `\accepts` (acepta) es `\denies` (deniega), lo que a veces se necesita cuando se están reutilizando definiciones de contexto existentes.

Organizar las piezas requeridas dentro de un bloque `\layout` nos deja con

```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \inherit-acceptability ImproVoice Voice
}
```

Así pues, la salida que aparece al comienzo de esta sub-sección se puede escribir como

```
\relative {
  a'4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"desvístete"
    c c_"mientras tocas :)"
  }
  a1
}
```

Véase también

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “Engravers and Performers” en *Referencia de Funcionamiento Interno*.

Nuevos contextos en el MIDI

En la salida MIDI, la sintaxis para definir nuevos tipos de contexto es la misma, excepto que el bloque `\context` se debe situar dentro de un bloque `\midi`, y el `\type` tendría que ser normalmente `Performer_group` en lugar de `Engraver_group`. El término *engraver* (grabador) se refiere a un plug-in o complemento de contexto que crea una cierta salida visual. En cambio, un *performer* o intérprete es de relevancia solamente en la salida MIDI. Cuando los plug-ins o complementos llevan “translator” como parte del nombre en lugar de “engraver” o “performer”, son relevantes tanto para la salida gráfica como para la interpretación MIDI. Así, cuando estamos adaptando una definición de contexto para el bloque `\midi`, tenemos que

- copiarlo en un bloque `\midi`,
- sustituir `Engraver_group` por `Performer_group`,
- eliminar `\consists` para los engravers o grabadores (ya que no son relevantes), y posiblemente añadir `\consists` para los performers o intérpretes.

Tenga muy en cuenta que, para mantener una interpretación coherente entre las salidas gráfica y de MIDI, se recomienda copiar cualquier definición de contexto personalizada dentro de un bloque `\midi`. Como mínimo, tendría que incluir las instrucciones que especifican la jerarquía de

contextos, como \accepts, \defaultchild y \inherit-acceptability. Se recomienda también copiar los alias.

De manera que, para completar el ejemplo anterior, se puede añadir lo siguiente:

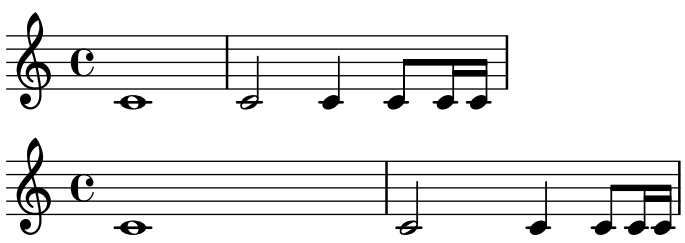
```
\midi {
  \context {
    \name ImproVoice
    \type Performer_group
    \alias Voice
    \consists Note_performer
    \consists Beam_performer
    \consists Dynamic_performer
    \consists Tie_performer
    \consists Slur_performer
  }
  \context {
    \Staff
    \accepts ImproVoice
  }
}
```

Esto hace que el contexto ImproVoice funcione también en la salida MIDI.

Replacing the Score context

Para escribir un contexto MyScore que funcione como el contexto de nivel más alto igual que lo hace normalmente el contexto Score, use \inherit-acceptability MyScore Score. El ejemplo siguiente define un contexto ProportionalScore en el que está activada la notación proporcional (véase Sección 30.6 [Notación proporcional], página 728).

```
\layout {
  \context {
    \Score
    \name ProportionalScore
    \alias Score
    proportionalNotationDuration = #1/8
  }
  \inherit-acceptability ProportionalScore Score
}

music = { c'1 2 4 8 16 16 }

\new Score \music
\new ProportionalScore \music
```



Dado que el contexto del nivel superior debe poder contener un cierto número de grabadores fundamentales,, la herencia de los ajustes con \Score es lo más sencillo en casi todos los casos. Si, de todas formas, definimos un contexto en el nivel de Score partiendo de cero y sin heredar

la definición de Score, el argumento de \type debe ser Score_engraver (o Score_performer dentro de \midi) en lugar de Engraver_group. Además está muy recomendado dar al contexto de nivel superior el alias Score, porque varios grabadores necesitan poder acceder al contexto superior usando su alias.

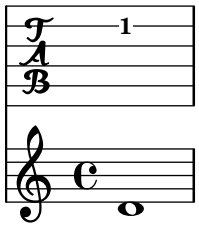
33.7 Orden de disposición de los contextos

Los contextos se disponen en un sistema normalmente desde arriba hacia abajo en el orden en que se encuentran en el archivo de entrada. Cuando los contextos se anidan unos dentro de otros, el contexto exterior incluye a los contextos anidados tal y como se especifica en el archivo de entrada, siempre y cuando los contextos interiores estén incluidos en la lista "accepts" del contexto externo. Los contextos anidados que no están incluidos en la lista "accepts" del contexto externo se recolocan debajo del contexto externo en lugar de anidarse dentro de él.

La lista "accepts" de un contexto se puede cambiar con las instrucciones \accepts (acepta) o \denies (niega). \accepts añade un contexto a la lista "accepts" y \denies elimina un contexto de la lista.

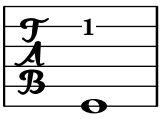
Por ejemplo, un contexto TabStaff (pauta de tablatura) de manera predeterminada \accepts (acepta) contextos TabVoice (contextos de voz de tablatura) y \denies (deniega) contextos Voice. Si un contexto Voice se escribiera dentro del TabStaff, se crearía sobre una pauta separada.

```
\score {
  \new TabStaff <<
    \new TabVoice { c'1 }
    \new Voice { d'1 }
  >>
}
```



Sin embargo, usando la instrucción \accepts, se puede forzar a Voice a que pueda estar dentro del contexto TabStaff.

```
\score {
  \new TabStaff <<
    \new TabVoice { c'1 }
    \new Voice { d'1 }
  >>

\layout {
    \context {
      \TabStaff
      \accepts Voice
    }
  }
}
```



`\denies` se usa principalmente cuando un contexto nuevo se está basando en otro, pero los anidamientos requeridos difieren. Por ejemplo, el contexto `VaticanaStaff` está basado en el contexto `Staff`, pero con el contexto `VaticanaVoice` sustituido por el contexto `Voice` en la lista "accepts".

Observe que discretamente se crea un contexto de forma implícita si se encuentra una instrucción donde no hay un contexto apropiado para contenerlo.

Dentro de una definición de contexto, el tipo de un subcontexto que se va a crear implícitamente se especifica usando `\defaultchild` (hijo predeterminado). Algunos eventos musicales requieren un contexto del nivel más bajo: cuando se encuentra este evento, se crean subcontextos de forma recursiva hasta que se alcanza un contexto que no tiene establecido el `defaultchild`.

La creación implícita de contextos puede dar lugar a pentagramas o partituras nuevos no esperados. La utilización de `\new` para crear contextos explícitamente evita esos problemas.

En ocasiones se necesita que un contexto exista durante un breve intervalo de tiempo, siendo un buen ejemplo el contexto de pentagrama de un ossia. Esto se consigue normalmente mediante la introducción de la definición del contexto en el lugar apropiado en paralelo con la sección correspondiente de la música principal. De forma predeterminada, el contexto tempral se coloca debajo de todos los contextos existentes. Para reposicionarlo por encima del contexto que tenga el nombre "principal", debería definirse de esta forma:

```
\new Staff \with { alignAboveContext = "principal" }
```

Se presenta una situación similar cuando se posiciona un contexto temporal de letra de una canción dentro de una disposición de varios pentagramas tal como `ChoirStaff`, por ejemplo, cuando se añade una segunda estrofa a una sección que se repite. De forma predeterminada, el contexto temporal de letra se coloca debajo de los pentagramas inferiores. Mediante la definición del contexto temporal de letra con `alignBelowContext` se puede posicionar correctamente debajo del contexto de letra con nombre que contiene el texto de la primera estrofa.

En diversos lugares pueden verse ejemplos que muestran esta recolocación de contextos temporales: véase Sección "Anidado de expresiones musicales" en *Manual de Aprendizaje*, Sección 6.2 [Modificación de pentagramas sueltos], página 252, y Sección 9.2 [Técnicas específicas para la letra], página 362.

Véase también

Manual de aprendizaje: Sección "Anidado de expresiones musicales" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 6.2 [Modificación de pentagramas sueltos], página 252, Sección 9.2 [Técnicas específicas para la letra], página 362.

Manual de utilización del programa: Sección "Aparece un pentagrama de más" en *Utilización del Programa*.

Archivos instalados: `ly/engraver-init.ly`.

34 Explicación del Manual de referencia de funcionamiento interno

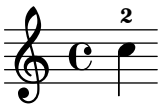
Véase también

Referencia de la Notación: Sección B.24 [Convenciones de nomenclatura], página 978.

34.1 Navegar por la referencia del programa

Supongamos que queremos mover la indicación de digitación del fragmento siguiente:

```
c''-2
```



Si hace una visita a la documentación en busca de instrucciones de digitación (en Sección 7.1.2 [Indicaciones de digitación], página 284), encontrará:

> **Véase también**
>
> Referencia de funcionamiento interno: Sección “Fingering” en *Referencia de Funcionamiento Interno*.

La referencia del programador se encuentra disponible en forma de documento HTML. Se recomienda mucho que lo lea en la forma HTML, bien en línea o bien descargando los archivos de la documentación HTML. Esta sección sería mucho más difícil de entender si está utilizando el manual en formato PDF.

Siga el enlace que lleva a Sección “Fingering” en *Referencia de Funcionamiento Interno*. Al principio de la página, puede ver

> Los objetos de digitación se crean por parte de los siguientes grabadores: Sección “Fingering_engraver” en *Referencia de Funcionamiento Interno* y Sección “New_fingering_engraver” en *Referencia de Funcionamiento Interno*.

Siguiendo los enlaces relacionados dentro de la referencia del programa, podemos seguir el flujo de información dentro del programa:

- Sección “Fingering” en *Referencia de Funcionamiento Interno*: los objetos Sección “Fingering” en *Referencia de Funcionamiento Interno* se crean por parte de los siguientes grabadores: Sección “Fingering_engraver” en *Referencia de Funcionamiento Interno*
- Sección “Fingering_engraver” en *Referencia de Funcionamiento Interno*: Tipos de música aceptados: Sección “fingering-event” en *Referencia de Funcionamiento Interno*
- Sección “fingering-event” en *Referencia de Funcionamiento Interno*: El tipo de evento musical `fingering-event` está descrito en Expresiones musicales con el nombre de Sección “FingeringEvent” en *Referencia de Funcionamiento Interno*

Este camino se recorre en contra de la corriente de información del programa: comienza por la salida y acaba en el evento de entrada. También podríamos haber empezado por un evento de la entrada, y leído siguiendo el flujo de información terminando en su caso en el objeto (u objetos) de la salida.

La referencia del programa también se puede examinar como un documento normal. Contiene capítulos que tratan de `Music definitions` de la Sección “Translation” en *Referencia de Funcionamiento Interno*, y del Sección “Backend” en *Referencia de Funcionamiento Interno*. Cada uno de los capítulos relaciona todas las definiciones utilizadas y todas las propiedades que se pueden ajustar.

34.2 Interfaces de la presentación

La página HTML que pudimos ver en la sección anterior describe el objeto de presentación llamado Sección “Fingering” en *Referencia de Funcionamiento Interno*. Dicho objeto es un símbolo dentro de la partitura. Tiene propiedades que guardan números (como grosores y direcciones), pero también punteros a objetos relacionados. Un objeto de presentación también se llama un *Grob*, que es una abreviatura de Graphical Object (objeto gráfico). Para ver más detalles acerca de los objetos gráficos o Grobs, consulte Sección “grob-interface” en *Referencia de Funcionamiento Interno*.

La página dedicada a `Fingering` relaciona las definiciones del objeto `Fingering`. Por ejemplo, la página dice

> `padding` (dimensión, en espacios de pentagrama):
> `0.5`

lo que significa que el número se mantendrá a una distancia de al menos 0.5 de la cabeza de la nota.

Cada objeto de presentación puede tener varias funciones como elemento notacional o tipográfico. Por ejemplo, el objeto de digitación Fingering tiene los siguientes aspectos

- Su tamaño es independiente del espaciado horizontal, al contrario de las ligaduras o las barras de las figuras.
- Es un elemento de texto. Casi seguro que es un texto muy corto.
- este elemento de texto se tipografía con un tipo de letra, no como las ligaduras o las barras de las figuras.
- Horizontalmente, el centro del símbolo se debe alinear con el centro de la cabeza de la nota.
- Verticalmente, el símbolo se coloca cerca de la nota y del pentagrama.
- La posición vertical también está coordinada con otros símbolos de superíndice y de subíndice.

Cada uno de estos aspectos se capta en lo que se llaman *interface*s, que se relacionan al final de la página dedicada a Sección “Fingering” en *Referencia de Funcionamiento Interno*

> Este objeto contempla los siguientes interfaces: Sección “item-interface” en *Referencia de Funcionamiento Interno*, Sección “self-alignment-interface” en *Referencia de Funcionamiento Interno*, Sección “side-position-interface” en *Referencia de Funcionamiento Interno*, Sección “text-interface” en *Referencia de Funcionamiento Interno*, Sección “text-script-interface” en *Referencia de Funcionamiento Interno*, Sección “font-interface” en *Referencia de Funcionamiento Interno*, Sección “finger-interface” en *Referencia de Funcionamiento Interno* y Sección “grob-interface” en *Referencia de Funcionamiento Interno*.

Al pulsar sobre cualquiera de los enlaces nos desplazaremos a la página del respectivo interfaz del objeto. Cada interfaz tiene un cierto número de propiedades. Algunas de ellas no son para que el usuario las pueda ajustar (‘Propiedades internas’), pero otras sí se pueden modificar.

Hemos estado hablando de *el* objeto `Fingering`, pero realmente esto no significa mucho. El archivo de inicialización (véase Sección “Otras fuentes de información” en *Manual de Aprendizaje*) `scm/define-grobs.scm` muestra el alma del ‘objeto’,

```
(Fingering
  . ((padding . 0.5)
     (avoid-slur . around)
     (slur-padding . 0.2)
     (staff-padding . 0.5)
     (self-alignment-X . 0)
     (self-alignment-Y . 0)
     (script-priority . 100)
```

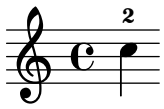
```
(stencil . ,ly:text-interface::print)
(direction . ,ly:script-interface::calc-direction)
(font-encoding . fetaText)
(font-size . -5)           ; no superponerse cuando está junto a una cabeza.
(meta . ((class . Item)
(interfaces . (finger-interface
               font-interface
               text-script-interface
               text-interface
               side-position-interface
               self-alignment-interface
               item-interface))))))
```

Como podemos ver, el objeto `Fingering` no es más que un montón de valores de variables, y la página web de la Referencia de funcionamiento interno se genera directamente a partir de esta definición.

34.3 Determinar la propiedad del grob

Recordemos que queríamos cambiar la posición del **2** en

```
c''-2
```



Puesto que el **2** se encuentra colocado verticalmente sobre su nota, tenemos que negociar con el interfaz asociado con esta colocación. Esto se hace usando `side-position-interface`. La página que describe este interface dice:

> `side-position-interface`
>
> Colocar un objeto víctima (este mismo) junto a otros objetos (el soporte). La propiedad `direction` significa dónde poner el objeto víctima con relación al soporte (¿a la izquierda o a la derecha, encima o debajo?)

Debajo de esta descripción, la variable `padding` (relleno) se describe como

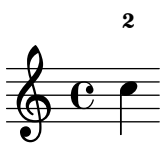
> `padding` (dimensión, en espacios de pentagrama)
>
> Añadir esta cantidad de espacio adicional entre objetos que están unos junto a otros.

Aumentando el valor de `padding`, podemos alejar la cifra de digitación de la cabeza de la nota. La siguiente orden inserta un espacio en blanco de tres "espacios de pentagrama" entre la nota y la digitación:

```
\once \override Voice.Fingering.padding = 3
```

Al insertar este relleno antes de que se haya creado el objeto de digitación, llegamos al siguiente resultado:

```
\once \override Voice.Fingering.padding = 3
c''-2
```



En este caso, el contexto de este truco es `Voice`. Véase el apartado dedicado al añadido Sección “Fingering_engraver” en *Referencia de Funcionamiento Interno*, que dice:

> El grabador Fingering_engraver forma parte de los contextos: . . . Sección “Voice” en *Referencia de Funcionamiento Interno*

35 Modificar las propiedades

35.1 Panorámica de la modificación de las propiedades

Dentro de cada contexto hay dos tipos diferentes de propiedades: *propiedades de contexto* y *propiedades de grob*. Las propiedades de contexto se aplican a un contexto como un todo, mientras que las propiedades de grob se usan para inicializar objetos gráficos (grobs) impresos desde dentro de un contexto.

Las propiedades de contexto controlan la traducción de la música a la notación. Por ejemplo, `localAlterations` se usa para determinar si hay que imprimir las alteraciones accidentales; o `currentBarNumber` para determinar qué número de compás se imprime. También pueden cambar de valor con el tiempo durante la interpretación de una pieza de música; `currentBarNumber` (el número del compás actual) es un ejemplo obvio.

La instrucción `\set` (y su contrapartida `\unset`) se utiliza para alterar los valores de las propiedades de contexto, mientras que la instrucción `\override` (y su contrapartida `\revert`) se usa para cambiar valores de las propiedades de un grob.

35.2 Las instrucciones `\set` y `\unset`

La sintaxis de la instrucción `\set` es

```
\set contexto.propiedad = valor
```

donde *valor* debe ir precedido del carácter de almohadilla '`#`' si es un objeto de Scheme.

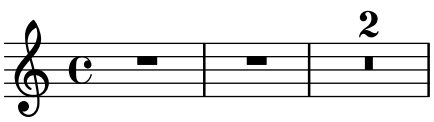
La sintaxis de su contrapartida, la instrucción `\unset`, es

```
\unset contexto.propiedad
```

Esto elimina una definición anteriormente establecida de la *propiedad* del *contexto*. Cualquier propiedad que haya sido establecida dentro del *contexto* incluido no resulta alterada por un `\unset` en el mismo contexto incluido.

Por ejemplo, los silencios multicompás se combinan en un solo compás (como se explica en Sección 6.3.4 [Compresión de los compases vacíos], página 277) si la propiedad de contexto `skipBars` se ha establecido a `#t`:

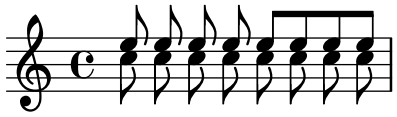
```
R1*2
\set Score.skipBars = ##t
R1*2
```



Si el argumento *contexto* se deja sin especificar, entonces la propiedad se establece en el contexto 'de nivel inferior' actual (normalmente `ChordNames`, `Voice`, `TabVoice`, o `Lyrics`).

```
<<
  \set Score.autoBeaming = ##f
  \relative {
    e''8 e e e
    \set autoBeaming = ##t
    e8 e e e
  } \\
  \relative {
    c''8 c c c c8 c c c
  }
```

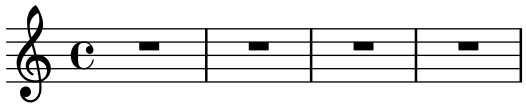
```
>>
```



El cambio se aplica 'al vuelo', mientras dura la música, de forma que el ajuste sólo afecta al segundo grupo de corcheas.

Observe que el contexto del nivel más bajo no siempre contiene un *engraver* que usa la propiedad que querríamos modificar: por ejemplo, intentar ajustar el valor de la propiedad `skipBars` del contexto predeterminado del nivel más bajo, no tendrá ningún efecto, porque skipBars es una propiedad del contexto `Score`, no de `Voice`.

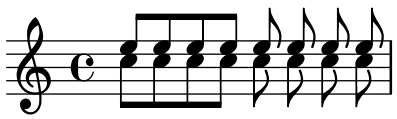
```
R1*2
\set skipBars = ##t
R1*2
```



Los contextos son jerárquicos; cualquier cambio especificado para un contexto que encierra a otros (p.ej. `Staff`) se aplica también a todas las `Voices` dentro de ese contexto `Staff` en curso (suponiendo, por descontado, que el propio contexto `Voice` no tiene una sobreescritura).

Todos los contextos heredan los ajustes que estén establecidos en el contexto de nivel más alto `Global` (a través de `\grobdescriptions`), aunque algunos de esos valores predeterminados resultan sobreescritos dentro de sus propias definiciones de contexto.

```
<<
  \set Score.autoBeaming = ##t
  \relative {
    \unset autoBeaming
    e''8 e e e
    \unset Score.autoBeaming
    e8 e e e
  } \\
  \relative {
    c''8 c c c c8 c c c
  }
>>
```



Como `\set`, el argumento *contexto* no se tiene que especificar para un contexto del nivel más bajo, por lo que los dos enunciados

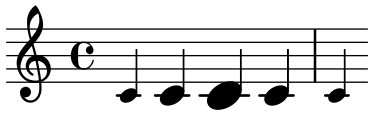
```
\set Voice.autoBeaming = ##t
\set autoBeaming = ##t
```

son equivalentes si el contexto inferior en curso es `Voice`.

Tal y como se ha descrito anteriormente, `\unset` restaura el valor predeterminado de una propiedad de contexto. Sin embarco, a veces es útil cambiar un valor durante un tiempo, y después devolverlo al valor utilizado anteriormente. Para este propósito existen las dos instrucciones

\pushContextProperty y \popContextProperty: la primera inserta el valor actual en una pila, mientras que la segunda retira un valor a partir de la pila y restaura la propiedad a ese valor.

```
{
  c'
  \pushContextProperty Staff.fontSize
  \set Staff.fontSize = 3
  c'
  \pushContextProperty Staff.fontSize
  \set Staff.fontSize = 6
  c'
  \popContextProperty Staff.fontSize
  c'
  \popContextProperty Staff.fontSize
  c'
}
```



35.3 Las instrucciones \override y \revert

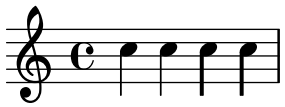
Existe un tipo especial de propiedad de contexto: la descripción de los grobs. Las decscripciones de los grobs reciben un nombre en MayúsculasDeCamello (empezando en mayúscula). Contienen los 'ajustes predeterminados' para un tipo particular de grob, en forma de lista asociativa. Consulte scm/define-grobs.scm para ver los ajustes de cada descripción de grob. Las descripciones de grob se modifican con \override.

La sintaxis de la instrucción \override es

```
\override [contexto.]NombreDelGrob.propiedad = #valor
```

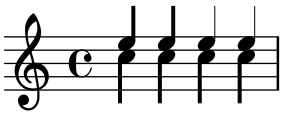
Por ejemplo, podemos aumentar el grosor de la plica de una figura sobreescribiendo la propiedad thickness (grosor) del objeto Stem (plica):

```
c''4 c''
\override Voice.Stem.thickness = 3.0
c''4 c''
```



Si no se ha especificado ningún contexto en la instrucción \override, se utiliza el contexto del nivel inferior:

```
\override Staff.Stem.thickness = 3.0
<<
  \relative {
    e''4 e
    \override Stem.thickness = 0.5
    e4 e
  } \\
  \relative {
    c''4 c c c
  }
>>
```



Algunas opciones susceptibles de trucaje, se llaman 'subpropiedades' y residen dentro de las propiedades. Para efectura trujajes sobre ellas, utilice instrucciones de la forma

```
\override Stem.details.beamed-lengths = #'(4 4 3)
```

o para modificar los extremos de los objetos extensos, utilice una forma como las siguientes:

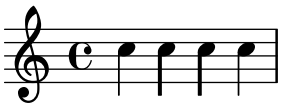
```
\override TextSpanner.bound-details.left.text = "left text"
\override TextSpanner.bound-details.right.text = "right text"
```

El efecto de una instrucción de sobreescritura `\override` se puede deshacer con `\revert`; su sintaxis es

```
\revert [Contexto.]NombreDelGrob.propiedad
```

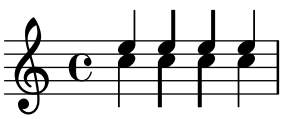
Por ejemplo,

```
\relative {
  c''4
  \override Voice.Stem.thickness = 3.0
  c4 c
  \revert Voice.Stem.thickness
  c4
}
```



Los efectos de `\override` y `\revert` se aplican a todos los grobs del contexto afectado partiendo del momento actual y hacia adelante:

```
<<
  \relative {
    e''4
    \override Staff.Stem.thickness = 3.0
    e4 e e
  } \\
  \relative {
    c''4 c c
    \revert Staff.Stem.thickness
    c4
  }
>>
```

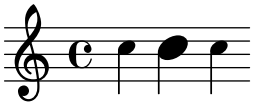


35.4 La instrucción \once

`\once` se usa junto a las instrucciones `\set` o `\override` para afectar solamente al instante de tiempo actual.

```
c''4
\once \set fontSize = 4.7
```

```
c''4
c''4
```



Véase también

Referencia de funcionamiento interno: Sección “Backend” en *Referencia de Funcionamiento Interno*.

35.5 `\set` frente a `\override`

Las instrucciones `\set` y `\override` manipulan propeidades asociadas a contextos. Las propiedades existen dentro de una jerarquía de contextos en la que cada uno de ellos contiene por debajo otross cero o más contextos. Una propiedad que no se ha establecido en un contexto específico presenta el valor que hereda del contexto que la encierra más próximo en que se ha establecido.

EL tiempo de vida y el valor de una propiedad de contexto son dinámicos y solo están disponibles mientras la música se está interpretando (esto es: ‘iterando’). En el momento temporal de la creación del contexto, las propiedades se incializan a partir de sus definiciones correspondientes (junto a cualesquiera otras modificaciones) de ese contexto. Cualquier cambio posterior se consigue con las instrucciones de ‘establecimiento de propiedades’ que están dentro de la propia música.

Las definiciones de Objetos Gráficos (o “grobs”) son una categoría *especial* de propiedades de contexto, ya que su uso y estructura es diferente de las propiedades de contexto normales. A diferencia de las propiedades de contexto normales, las definiciones de grob se subdividen en *propiedades de grob*.

Es más, a diferencia de las propiedades de contexto normales, las definiciones de grob tienen su propio protocolo de mantenimiento interno que se usa para seguir la pista de sus propieas propiedades de grob y cualquier sub-propiedad. Esto significa que es posible definir aquellas partes dentro de distintos contextos y a pesar de ello tener la definición de grob general en el momento de la creación del grob a partir de todas las piezas aportadas entre los contextos actual y circundantes.

Un grob se crea normalmente por parte de un grabador en el momento de la interpretación de una expresión musical, y recibe sus propiedades iniciales de la definición de grob en curso del contexto del grabador. El grabador (u otras partes de ‘backend’ LilyPond) pueden entonces cambiar (o añadirse a) las porpieades iniciales del grob. Sin embargo, esto no afecta a la propia definición de grob del contexto.

Lo que LilyPond llama *propiedades de grob* en el contexto de los trucos en el ‘nivel del usuario’ son realmente las propiedades de la propia definición de grob del *contexto*.

Se accede a las definiciones de grob con un conjunto de instrucciones diferentes y se manipulan usando `\override` y `\revert`, y tienen un nombre con la primera letra en mayúscula (p.ej. ‘`NoteHead`’); mientras que las propiedades de contexto normales se manipulan usando `\set` y `\unset` y su nombre empieza en letra minúscula.

Las instrucciones `\tweak` y `\overrideProperty` cambian las propiedades de grob puenteando completamente todas las propiedades de contexto, y captan los grobs conforme se van creando, fijando propieades sobre ellos para un evento de música (`\tweak`) o, en el caso de `\overrideProperty`, para una sobreescritura específica.

35.6 Las instrucciones \tweak y \single

Cuando varios grobs tienen lugar en el mismo instante musical, no puede usarse la instrucción \override para modificar una sola de ellas. En este caso se usa la instrucción \tweak.

La instrucción \tweak tiene la siguiente sintaxis:

```
\tweak [objeto-de-presentación.]propiedad-del-grob valor
```

y se aplica a la expresión musical que sigue inmediatamente a *valor* dentro del flujo de la música. Es necesario especificar *objeto-de-presentación* para resolver la ambigüedad si la expresión musical causa la creación indirecta de grobs de distintos tipos (por ejemplo, NoteHead produce Stem).

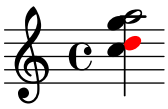
Para ver una introducción a la sintaxis y los usos de la instrucción tweak, consulte Sección "Métodos de trucaje" en *Manual de Aprendizaje*.

Entre los elementos que pueden aparecer más de una vez en el mismo momento musical se encuentran los siguientes:

- las cabezas de las notas de un acorde
- signos de articulación sobre la misma nota
- ligaduras de unión entre notas de un acorde
- corchetes de grupos especials que comienzan en el mismo momento

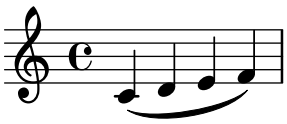
En este ejemplo se modifican el color de una cabeza y el tipo de otra, dentro del mismo acorde:

```
< c''
  \tweak color #red
  d''
  g''
  \tweak duration-log 1
  a''
> 4
```



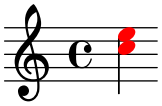
\tweak se puede usar también para modificar ligaduras de expresión:

```
\relative { c'-\tweak thickness 5 ( d e f) }
```



El truco efectuado sobre un acorde se aplica a todas las notas que contiene:

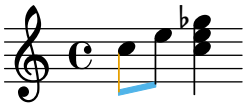
```
{ \tweak color #red <c'' e''>4 }
```



Como se mencionó anteriormente, la instrucción \tweak sencilla no se puede usar para modificar ningún objeto que no se haya creado directamente a partir de la entrada. Concretamente, no afecta a las plicas, barras automáticas ni alteraciones, porque éstos se generan posteriormente por parte de objetos de presentación NoteHead más que por elementos musicales del flujo de entrada.

Tales objetos de presentación creados indirectamente se pueden trucar usando la forma de la instrucción \tweak en que el nombre del grob se especifica de forma explícita:

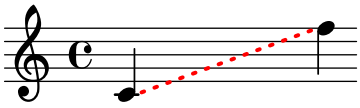
```
\tweak Stem.color #(universal-color 'orange)
\tweak Beam.color #(universal-color 'skyblue) c''8 e''
<c'' e'' \tweak Accidental.font-size -3 ges''>4
```



No se puede usar \tweak para modificar las claves o las indicaciones de compás, porque éstos se separan de cualquier instrucción \tweak precedente dentro del flujo de entrada merced a la inserción automática de elementos adicionales que se requieren para especificar el contexto.

Se pueden colocar varias instrucciones \tweak antes de un elemento de notación; todos le afectan:

```
c'
  -\tweak springs-and-rods #ly:spanner::set-spacing-rods
  -\tweak minimum-length 15
  -\tweak style #'dashed-line
  -\tweak dash-fraction 0.2
  -\tweak thickness 3
  -\tweak color #red
  \glissando
f''
```

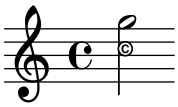


El flujo musical que se genera a partir de una sección de un archivo de entrada (incluido cualquier elemento insertado automáticamente) puede examinarse, véase Sección “Presentación de las expresiones musicales” en *Extender*. Esto puede ser de utilidad en la determinación de lo que puede modificarse por medio de una instrucción \tweak.

La instrucción \single toma una o más instrucciones \override (que se pretende que tengan efecto en un momento musical dado o posterior) y las convierte efectivamente en *un solo* ‘tweak’ que ahora se aplica a los grobs específicos que se han creado.

El archivo ly/property-init.ly contiene muchas definiciones de varias instrucciones \override y pueden así ser usadas en conjunción con la instrucción \single. Por ejemplo, se puede usar la función \easyHeadsOn con \single para que afecte solo a una nota de un acorde;

```
\relative c' {
  <\single \easyHeadsOn c' g'>2
}
```



Véase también

Manual de aprendizaje: Sección “Métodos de trucaje” en *Manual de Aprendizaje*.

Manual de extensión: Sección “Presentación de las expresiones musicales” en *Extender*.

Advertencias y problemas conocidos

No se puede usar la instrucción `\tweak` para modificar los puntos de control de una sola de varias ligaduras de unión dentro de un acorde, aparte de la primera que se encuentre en el código de entrada.

35.7 La instrucción `\offset`

Aunque es posible fijar propiedades de grob a valores nuevos con las instrucciones `\override`, `\tweak` y `\overrideProperty`, muchas veces conviene más modificar dichas propiedades en una cantidad relativa al valor predeterminado. La instrucción `\offset` se encuentra disponible para este fin.

La sintaxis de `\offset` es

```
[-]\offset propiedad incrementos elemento
```

La instrucción funciona añadiendo el contenido de *incrementos* al ajuste predeterminado de la *propiedad* del grob indicado por *elemento*.

Dependiendo de la formulación de la instrucción, `\offset` puede actuar como un `\tweak` (trucaje) o como un `\override` (sobreescritura). Las variaciones en su uso se explicarán después de haber examinado las propiedades de grob que se pueden usar con `\offset`.

Propiedades que admiten el desplazamiento

Muchas, pero no todas, las propiedades de grob admiten la aplicación de un desplazamiento. Si *propiedad* no se puede desplazar, el objeto permanece sin cambios y se emite una advertencia. En estos casos se deben usar en su lugar `\override` o `\tweak` para modificar el objeto.

Se puede trabajar por ensayo y error y dejar que las advertencias nos sirvan de guía para saber lo que puede o no puede ser desplazado. Sin embargo, es posible un enfoque más sistemático.

Los criterios siguientes determinan si una propiedad se puede modificar con `\offset`:

- La propiedad tiene un 'ajuste predeterminado' en la descripción del grob. Tales propiedades aparecen listadas para cada grob en Sección "All layout objects" en *Referencia de Funcionamiento Interno* (y también están en `scm/define-grobs.scm`).
- La propiedad admite un valor numérico. Entre los valores numéricos se encuentran `number`, una lista de `numbers`, `number-pair`, y `number-pair-list`. Las páginas de Sección "All layout objects" en *Referencia de Funcionamiento Interno* contienen una lista de los tipos de datos característicos de cada propiedad. No importa si el ajuste predeterminado es una función.
- La propiedad no puede ser una 'subpropiedad' (una propiedad que reside dentro de otra propiedad).
- Las propiedades establecidas a valores infinitos no se pueden desplazar. No hay ninguna forma sensata de incrementar los valores infinitos positivos o negativos.

Los ejemplos siguientes tratan diversas propiedades de grob frente a los criterios que se han expuesto.

- Propiedades que se pueden desplazar

  `Hairpin.height`
  : Esta propiedad no es una subpropiedad, y está en la lista que aparece en Sección "Hairpin" en *Referencia de Funcionamiento Interno*. En cuanto a su valor, admite 'dimension, in staff space' (dimensión, en espacios de pentagrama) fijado a `0.6666` (que claramente es un `number` no infinito).

  `Arpeggio.positions`
  : En la lista de la página que describe Sección "Arpeggio" en *Referencia de Funcionamiento Interno* aparece una propiedad `positions` que admite una

'pareja de números'. Su valor predeterminado es `ly:arpeggio::positions`, que es un 'callback' que se evalúa durante la fase de tipografiado para dar como resultado una pareja de números para cualquier objeto `Arpeggio` dado.

- Propiedades que no admiten el desplazamiento

`Hairpin.color`
: No hay un listado para `color` en Sección "Hairpin" en *Referencia de Funcionamiento Interno*.

`Hairpin.circled-tip`
: La lista de `Hairpin.circled-tip` en la página de Sección "Hairpin" en *Referencia de Funcionamiento Interno* muestra que admite un valor `boolean`. Los valores booleanos so no numéricos.

`Stem.details.lengths`
: Aunque está listado en Sección "Stem" en *Referencia de Funcionamiento Interno* y su valor por omisión es una lista de `numbers`, es una 'subpropiedad'. Por el momento no se contemplan las 'propiedades anidadas'.

`\offset` como sobreescritura

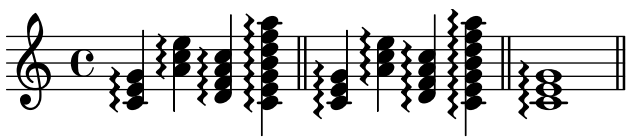
Si *elemento* es un nombre de grob como `Arpeggio` o `Staff.OttavaBracket`, el resultado es un `\override` o sobreescritura de el tipo de grob especificado.

`\offset` *`propiedad incremento`* `[`*`contexto`*`.]`*`NombreDeGrob`*

Observe que el guión precedente *nunca* se utiliza con la forma 'override', de igual manera que nunca se usa con la propia instrucción `\override`.

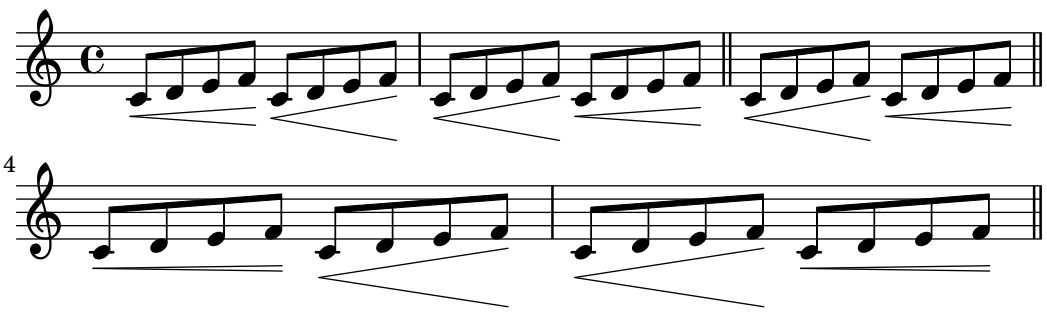
El ejemplo siguiente usa la forma 'override', de sobreescritura, para alargar los arpegios predeterminados que aparecen en el primer compás para que cubran todo el tamaño de los acordes. Los arpegios se amplían en medio espacio de pentagrama hacia arriba y hacia abajo. También se muestra la misma operación efectuada sobre el primer acorde con una sobreescritura normal de la propiedad `positions`. Este método no es en absoluto expresivo de la acción de 'ampliar en medio espacio de pentagrama', porque los extremos se tienen que especificar mediante coordenadas absolutas, no relativas. Además, se necesitarían sobreescrituras individuales para los otros acordes, ya que varían en tamaño y posición.

```
arpeggioMusic = {
  <c' e' g'>\arpeggio <a' c'' e''>\arpeggio
  <d' f' a' c''>\arpeggio <c' e' g' b' d'' f'' a''>\arpeggio
}

{
  \arpeggioMusic
  \bar "||"
  \offset positions #'(-0.5 . 0.5) Arpeggio
  \arpeggioMusic
  \bar "||"
  \once \override Arpeggio.positions = #'(-3.5 . -0.5)
  <c' e' g'>1\arpeggio
  \bar "||"
}
```



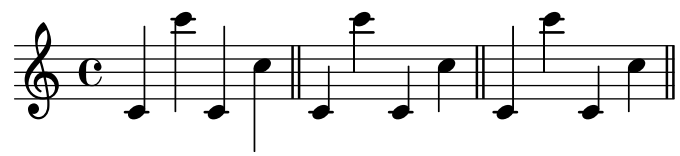
En su uso como 'override', \offset se puede hacer preceder por \once o por \temporary y revertirse mediante \revert con la *propiedad*. (véase Sección "Funciones de sustitución intermedias" en *Extender*). Esto se sigue a partir del hecho de que \offset en realidad crea un \override para la *propiedad*.

```
music = { c'8\< d' e' f'\! }

{
  \music
  \offset height 1 Hairpin
  \music
  \music
  \revert Hairpin.height
  \music
  \bar "||"
  \once \offset height 1 Hairpin
  \music \music
  \bar "||"
  \override Hairpin.height = 0.2
  \music
  \temporary \offset height 2 Hairpin
  \music
  \music
  \revert Hairpin.height
  \music
  \bar "||"
}
```



Así como \override, la forma 'override' de \offset se puede usar con \undo y con \single.

```
longStem = \offset length 6 Stem

{
 \longStem c'4 c''' c' c''
 \bar "||"
 \undo \longStem c'4 c''' c' c''
 \bar "||"
 \single \longStem c'4 c''' c' c''
 \bar "||"
}
```



\offset como trucaje

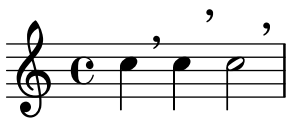
Si *elemento* es una expresión musical como (o \arpeggio, el resultado es la misma expresión musical con un trucaje aplicado.

[-]\offset [*NombreDeGrob.*]*propiedad incrementos expresión-musical*

La sintaxis de \offset en forma de 'tweak' es análoga a la propia instrucción \tweak, tanto en orden como en cuanto a la presencia o ausencia del guión inicial.

El ejemplo siguiente usa la forma 'tweak' para ajustar la posición vertical del objeto BreathingSign. Compárelo con la instrucción \tweak normal que también aparece. La sintaxis es equivalente; sin embargo, la salida de \tweak es menos intuitiva, ya que BreathingSign.Y-offset se calcula a partir de la tercera línea del pentagrama. No es necesario saber cómo se calcula Y-offset, al usar \offset.

```
{
  c''4
  \breathe
  c''4
  \offset Y-offset 2 \breathe
  c''2
  \tweak Y-offset 3 \breathe
}
```

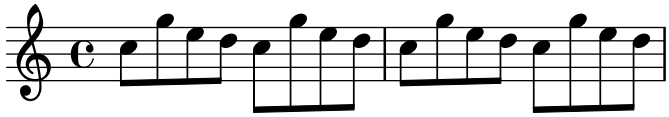


En el ejemplo anterior, los objetos trucados se crearon directamente a partir de la entrada del usuario: la instrucción \breathe era una instrucción explícita para devolver un objeto BreathingSign. Dado que el objeto de la instrucción no es ambiguo, no había necesidad de especificar el nombre del objeto. Sin embargo, cuando un objeto se crea *indirectamente*, es necesario incluir el nombre del grob. Es lo mismo que para la instrucción \tweak.

En el ejemplo siguiente, el objeto Beam se baja en dos espacios de pentagrama aplicando \offset a la propiedad positions.

La primera aplicación de \offset exige que se incluya el nombre del grob, porque nada en el código de entrada crea explícitamente la barra. En la segunda aplicación, la barra se crea manualmente con la expresión musical [; por tanto, el nombre del grob no es necesario (también aparece en la ilustración una abreviatura: un solo número se aplica a los dos miembros de una pareja).

```
{
  c''8 g'' e'' d''
  \offset Beam.positions #'(-2 . -2)
  c''8 g'' e'' d''
  c''8 g'' e'' d''
  c''8-\offset positions -2 [ g'' e'' d'']
}
```



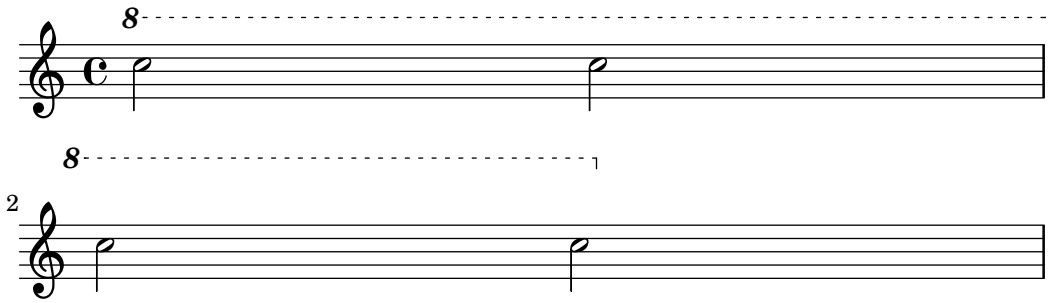
\offset con objetos extensos divididos

También es posible modificar segmentos de un objeto que se extiende sobre un salto o saltos de línea. En este caso, *offsets* admite una lista de valores del tipo de datos requerido por la propiedad.

La instrucción \offset usada de esta forma es similar a la instrucción \alterBroken (véase Sección 36.4.1 [Modificar objetos de extensión fragmentados], página 786). Sin embargo, a diferencia de \alterBroken, los valores que se le dan a \offset son relativos, no absolutos.

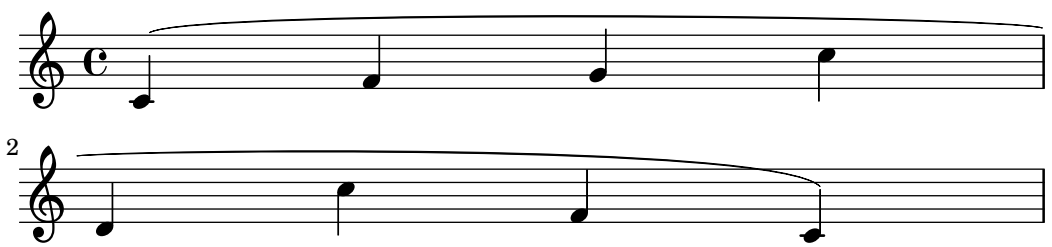
El ejemplo siguiente desplaza el objeto 'dividido' OttavaBracket a través de su propiedad staff-padding. Dado que la propiedad toma un number, se le proporciona a *incrementos* una lista de números para aplicarlos a los dos segmentos creados por el salto de línea. La parte de corchete de la primera línea queda, en realidad, inalterada, ya que se añade 0 a su valor por omisión de staff-padding. El segmento que está en la segunda línea se eleva tres espacios de pentagrama a partir de su altura por omisión. La altura predeterminada resulta ser 2, aunque no es necesario saberlo para conseguir la posición deseada.

```
{
  \offset staff-padding #'(0 3) Staff.OttavaBracket
  \ottava 1
  c'''2 c'''
  \break
  c'''2 c'''
}
```



El ejemplo siguiente copia el efecto de la instrucción \shape mediante el incremento de la propiedad control-points del objeto Slur. Aquí, *incrementos* es una lista de parejas de números, una por cada segmento de la ligadura. Este ejemplo produce un resultado idéntico a la ilustración correspondiente que aparece en Sección 36.11 [Modificación de las formas], página 808.

```
{
  c'4-\offset control-points #'(
              ((0 . 0) (0 . 0) (0 . 0) (0 . 1))
              ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
             ) ( f'4 g' c''
  \break
  d'4 c'' f' c')
}
```



Consulte Sección “Adjusting slur positions vertically” en *Fragmentos de código* para ver una función de Scheme `\offsetPositions` que facilita el desplazamiento vertical de las ligaduras.

35.8 Modificación de las listas-A

Ciertas propiedades configurables por parte del usuario se representan internamente como *listas-A* (listas asociativas), que almacenan duplas de *claves* y *valores*. La estructura de una lista-A es la siguiente:

```
'((clave1 . valor1)
  (clave2 . valor2)
  (clave3 . valor3)
  ...)
```

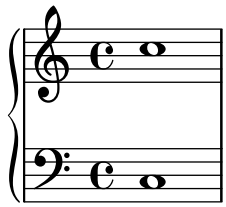
Si una lista-A es una propiedad de un grob o una variable de `\paper`, sus claves se pueden modificar individualmente sin que afecte a las otras claves.

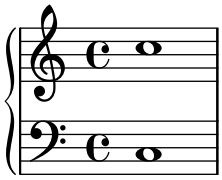
Por ejemplo, para reducir el espacio entre pentagramas adyacentes dentro de un grupo, use la propiedad `staff-staff-spacing` del grob `StaffGrouper`. La propiedad es una lista-A con cuatro claves: `basic-distance` (distancia básica), `minimum-distance` (distancia mínima), `padding` (relleno) y `stretchability` (ampliabilidad). Los ajustes estándar para esta propiedad se relacionan en la sección “Backend” de la Referencia de Funcionamiento Interno (véase Sección “StaffGrouper” en *Referencia de Funcionamiento Interno*):

```
'((basic-distance . 9)
  (minimum-distance . 7)
  (padding . 1)
  (stretchability . 5))
```

Una forma de acercar los pentagramas entre sí es reducir el valor de la clave `basic-distance` (9) para que se corresponda con el valor de `minimum-distance` (7). Para modificar una única clave de forma individual, utilice una *declaración anidada*:

```
% default space between staves
\new PianoStaff <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>

% reduced space between staves
\new PianoStaff \with {
  % this is the nested declaration
  \override StaffGrouper.staff-staff-spacing.basic-distance = 7
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>
```

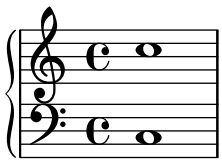




La utilización de una declaración anidada actualiza la clave especificada (como `basic-distance` en el ejemplo anterior) sin alterar ninguna de las otras claves que ya se habían establecido para la misma propiedad.

Ahora, supongamos que deseamos que los pentagramas estén tan próximos como sea posible sin que se superpongan. La manera más sencilla de hacerlo es establecer las cuatro claves de la lista-A a cero. Sin embargo, no es necesario escribir cuatro declaraciones anidadas, una por cada clave. En lugar de eso, se puede redefinir completamente la propiedad con una sola declaración, como una lista-A:

```
\new PianoStaff \with {
  \override StaffGrouper.staff-staff-spacing =
    #'((basic-distance . 0)
       (minimum-distance . 0)
       (padding . 0)
       (stretchability . 0))
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>
```



Observe que cualquier clave que no haya sido relacionada explícitamente en la definición de la lista-A, será reiniciada a sus valores *predeterminados si no se han fijado*. En el caso de `staff-staff-spacing`, el valor de cualquier clave no fijada se reiniciaría a cero (excepto `stretchability`, que toma el valor de `basic-distance` si no se fija). Así, las dos declaraciones siguientes son equivalentes:

```
\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7))

\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7)
     (minimum-distance . 0)
     (padding . 0)
     (stretchability . 7))
```

Una consecuencia de esto (posiblemente no intencionada) es la eliminación de cualquier valor estándar que se establezca en un archivo de inicio y que se carga cada vez que se compila un archivo de entrada. En el ejemplo anterior, los ajustes estándar para `padding` y `minimum-distance` (definidos en `scm/define-grobs.scm`) se reinician a sus valores predeterminados si no se han fijado (cero para las dos claves). La definición de una propiedad o variable como una lista-A (de cualquier tamaño) siempre reinicia todos los valores de clave no establecidos a sus valores predeterminados si no se han fijado. A no ser que este sea el resultado deseado, es más seguro actualizar los valores de clave individualmente con una declaración anidada.

Nota: Las declaraciones anidadas no funcionan para las listas-A de propiedades de contexto (como `beamExceptions`, `keyAlterations`, `timeSignatureSettings`, etc.). Estas propiedades sólo se pueden modificar redefiniéndolas completamente como listas-A.

36 Conceptos y propiedades útiles

Esta sección se ocupa de varios problemas comunes de disposición y los métodos de trucaje relacionados.

Véase también

Manual de aprendizaje: Sección “Tweaking output” en *Manual de Aprendizaje*, Sección “Other sources of information” en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 34 [Explicación del Manual de referencia de funcionamiento interno], página 764, Capítulo 35 [Modificar las propiedades], página 768.

Manual de extensión del programa: Sección “Interfaces for programmers” en *Extender*.

Archivos instalados: `scm/define-grobs.scm`.

Fragmentos de código: Sección “Trucos y sobreescrituras” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “All layout objects” en *Referencia de Funcionamiento Interno*.

36.1 Dirección y posición

Al tipografiar música, la dirección y colocación de muchos elementos es cuestión de elección. Por ejemplo, las plicas de las notas se pueden dirigir hacia arriba o hacia abajo; la letra, las indicaciones dinámicas y otras marcas expresivas se pueden colocar encima o debajo del pentagrama; el texto se pude alinear a la izquierda, a la derecha o centrado; etc. La mayoría de estas elecciones pueden dejarse que LilyPond las determine automáticamente, pero en ciertos casos puede ser deseable forzar una dirección o colocación concreta.

36.1.1 Indicadores de dirección de las articulaciones

De forma predeterminada algunas direcciones siempre son hacia arriba o siempre hacia abajo (p. ej. los matices o el calderón), mientras que otras cosas pueden alternar entre arriba y abajo en función de la dirección de las plicas (como las ligaduras o los acentos).

Se puede sobreescribir la acción predeterminada mediante el prefijado de la articulación por un *indicacor de dirección*. Están disponibles tres indicadores de dirección: `^` (que significa “arriba”), `_` (que significa “abajo”) o `-` (que significa “usar la dirección predeterminada”). El indicador de dirección se puede normalmente omitir, en cuyo caso se supone el indicador predeterminado `-`, pero se necesita un indicador de dirección **siempre** antes de:

- las instrucciones `\tweak`
- las instrucciones `\markup`
- las instrucciones `\tag`
- los marcados de cadena, p.ej. `-"cadena"`
- las instrucciones de digitación, p.ej. `-1`
- las abreviaturas de articulación, p.ej. `-.`, `->`, `--`

Estas indicaciones afectan sólo a la nota siguiente.

```
\relative {
  c''2( c)
  c2_( c)
  c2( c)
  c2^( c)
}
```



36.1.2 La propiedad de dirección

La posición o dirección de muchos objetos de presentación está controlada por la propiedad `direction`.

El valor de la propiedad `direction` se puede establecer al valor `1`, con el significado de "hacia arriba" o "encima", o a `-1`, con el significado de "hacia abajo" o "debajo". Se pueden usar los símbolos `UP` y `DOWN` en sustitución de `1` y `-1` respectivamente. La dirección predeterminada se puede especificar estableciendo `direction` a `0` ó a `CENTER`. De forma alternativa, en muchos casos existen instrucciones predefinidas para especificar la dirección. Todas ellas son de la forma:

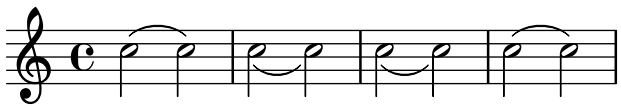
`\xxxUp, \xxxDown o \xxxNeutral`

donde `\xxxNeutral` significa "utilizar la dirección predeterminada". Véase Sección "Objetos interiores al pentagrama" en *Manual de Aprendizaje*.

En alguna que otra ocasión como en el arpegio, el valor de la propiedad `direction` puede especificar si el objeto se debe colocar a la izquierda o a la derecha del objeto padre. En este caso `-1` ó `LEFT` significan "a la izquierda" y `1` ó `RIGHT` significan "a la derecha". `0` ó `CENTER` significan "utilizar la dirección predeterminada".

Estas indicaciones afectan a todas las notas hasta que son canceladas.

```
\relative {
  c''2( c)
  \slurDown
  c2( c)
  c2( c)
  \slurNeutral
  c2( c)
}
```



En música polifónica, en general es mejor especificar una voz explícita que cambiar la dirección de un objeto. Para ver más información, véase Sección 5.2 [Varias voces], página 224.

Véase también

Manual de aprendizaje: Sección "Objetos interiores al pentagrama" en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2 [Varias voces], página 224.

36.2 Distancias y medidas

Las distancias en LilyPond son de dos tipos: absolutas y escaladas.

Las distancias absolutas se usan para especificar márgenes, sangrados y otros detalles de la disposición de la página, y por defecto se especifican en milímetros. En general, las unidades de distancia se pueden especificar añadiendo `\mm` (milímetros), `\cm` (centímetros), `\in` (pulgadas), `\pt` (puntos) o `\bp` (puntos grandes).

	mm	**cm**	**in**	**pt**	**bp**
mm	1	0.1	0.0394	2.8453	2.8346
cm	10	1	0.3937	28.4528	28.3465

in	25.4	2.54	1	72.27	72
pt	0.3515	0.0351	0.0138	1	0.996
bp	0.3538	0.0354	0.0139	1.0038	1

Las distancias de maquetación también se pueden especificar en unidades escalables (véase el párrafo siguiente) añadiendo `\staff-space` (espacios de pentagrama) a la cantidad. Véase Capítulo 26 [Disposición de la página], página 673, para una descripción detallada de la maquetación o disposición de página de LilyPond.

Las distancias escaladas siempre se especifican en unidades de un espacio del pentagrama o, más raramente, medio espacio del pentagrama. El espacio de pentagrama es la distancia entre dos líneas del pentagrama adyacentes. El valor predeterminado se puede cambiar globalmente fijando el tamaño global del pentagrama, o se puede sobreescribir localmente cambiando la propiedad `staff-space` del objeto `StaffSymbol`. Las distancias escaladas se escalan automáticamente con cualquier cambio al tamaño global del pentagrama o a la propiedad `staff-space` del objeto `StaffSymbol`, pero las fuentes tipográficas se escalan solamente con los cambios efectuados al tamaño global del pentagrama. Así, el tamaño global del pentagrama posibilita la fácil variación del tamaño general de una partitura impresa. Para ver los métodos de establecimiento del tamaño global del pentagrama, véase Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

Si se necesita dibujar sólo una sección de una partitura a una escala distinta, por ejemplo una sección ossia o una nota al pie, no se puede simplemente cambiar el tamaño global del pentagrama porque esto afectaría a toda la partitura. En tales casos, el cambio de tamaño se hace sobreescribiendo tanto la propiedad `staff-space` de `StaffSymbol` como el tamaño de las fuentes tipográficas. Está a nuestra disposición una función de Scheme, `magstep`, para convertir de un cambio en el tamaño de la fuente al cambio equivalente en `staff-space`. Para ver una explicación y un ejemplo de su utilización, consulte Sección “Longitud y grosor de los objetos” en *Manual de Aprendizaje*.

Véase también

Manual de aprendizaje: Sección “Longitud y grosor de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Capítulo 26 [Disposición de la página], página 673, Sección 27.2 [Establecer el tamaño del pentagrama], página 688.

36.3 Dimensiones

Las dimensiones de un objeto gráfico especifican las posiciones de los límites izquierdo y derecho, así como los límites inferior y superior de las cajas limítrofes de los objetos como distancias a partir del punto de referencia del objeto en unidades de espacios de pentagrama. Estas posiciones se codifican frecuentemente como dos parejas de valores de Scheme. Por ejemplo, la instrucción de marcado de texto `\with-dimensions` toma tres argumentos, de los cuales los dos primeros son una pareja de Scheme que da las posiiciones de los bordes izquierdo y derecho y otra pareja de Scheme que da las posiciones de los límites inferior y superior:

```
\with-dimensions #'(-5 . 10) #'(-3 . 15) arg
```

Esto especifica una caja circundante para *arg* con su límite izquierdo en -5, su límite derecho en 10, el inferior en -3 y el superior en 15, todos ellos medidos a partir del punto de referencia del objeto en unidades de espacios de pentagrama.

Para más información sobre la forma en que se definen estos rectángulos, incluida la definición formal del espacio horizontal y vertical según se establece por las instrucciones de marcado `\hspace` y `\vspace`, véase Sección “El modelo de caja de LilyPond” en *Extender*.

Véase también

Referencia de la notación: Sección A.1.8 [Other markup commands], página 879, (documentación para instrucciones similares a `\with-dimensions`, como `\with-dimension` o `\with-dimension-from`), Sección 36.2 [Distancias y medidas], página 784.

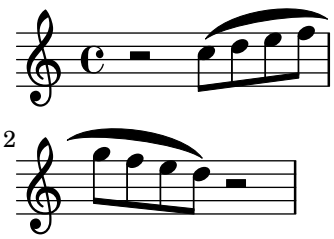
36.4 Objetos de extensión (spanners)

Muchos objetos de la notación musical se extienden a lo largo de varias notas o incluso varios compases. Son ejemplos las ligaduras de expresión, las barras de corchea, los corchetes de grupo especial, los corchetes de repetición de primera y segunda vez, crescendi, trinos y glissandi. Tales objetos reciben colectivamente el nombre de "spanners" u objetos de extensión, y tienen propiedades especiales que controlan su apariencia y comportamiento, así como métodos especiales de trucaje relacionados con el hecho de que pueden verse fragmentados a lo largo de varios sistemas.

36.4.1 Modificar objetos de extensión fragmentados

Cuando un objeto de extensión cruza un salto o saltos de línea, cada fragmento hereda los atributos del objeto de extensión original. Así, el trucaje ordinario de un objeto de extensión fragmentado aplica las mismas modificaciones a cada uno de sus segmentos. En el ejemplo que aparece a continuación, la sobreescritura de la propiedad `thickness` afecta a la ligadura de expresión a ambos lados del salto de línea.

```
\relative c'' {
  r2
  \once\override Slur.thickness = 10
  c8( d e f
  \break
  g8 f e d) r2
}
```



Es posible modificar independientemente la apariencia de los fragmentos individuales de un elemento de extenssión fragmentado con la instrucción `\alterBroken`. Esta instrucción puede producir o bien un `\override` (sobreescritura) o un `\tweak` (trucaje) de una propiedad del elemento extenso.

La sintaxis de la instrucción `\alterBroken` es

```
[-]\alterBroken propiedad valores objetivo
```

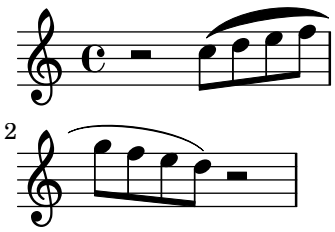
El argumento *valores* es una lista de valores, uno por cada fragmento. Si *objetivo* es el nombre de un grob como `Slur` o `Staff.PianoPedalBracket`, el resultado es un `\override` del tipo de grob especificado. Si *objetivo* es una expresión musical como '`(`' o '`[`', el resultado es la misma expresión musical con el trucaje arpopiado aplicado.

Se debe usar el guion inicial con la forma `\tweak`. No lo escriba cuando use `\alterBroken` como un `\override`.

En su uso como `\override`, `\alterBroken` may be prefaced by `\once` or `\temporary` and reverted by using `\revert` with *property* (see Sección "Intermediate substitution functions" en *Extender*).

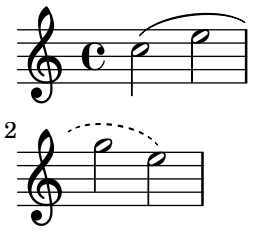
El código siguiente aplica una sobreescritura `\override` independiente a cada uno de los segmentos de ligadura de expresión del ejemplo anterior:

```
\relative c'' {
  r2
  \alterBroken thickness #'(10 1) Slur
  c8( d e f
  \break
  g8 f e d) r2
}
```



Se puede usar la instrucción `\alterBroken` con cualquier objeto de extensión, incluyendo `Tie` (ligadura de unión), `PhrasingSlur` (ligadura de fraseo), `Beam` (barra de corchea) y `TextSpanner` (elemento extenso de texto). Por ejemplo, un editor que se encuentra preparando una edición crítica podría desear indicar la ausencia de parte de una ligadura de fraseo en una fuente haciendo discontinuo solamente el segmenteo que se ha añadido. El ejemplo siguiente ilustra cómo se puede conesguir, en este caso usando la forma `\tweak` de la instrucción:

```
% The empty list is conveniently used below, because it is the
% default setting of dash-definition, resulting in a solid curve.
\relative {
  c''2-\alterBroken dash-definition #'(() ((0 1.0 0.4 0.75))) \(e
  \break
  g2 e\)
}
```



Es importante comprender que `\alterBroken` establece cada fragmento de un objeto de extensión dividido, al valor correspondiente dentro de *valores*. Cuando hay menos valores que fragmentos, se asigna una lista vacía a cualquier fragmento adicional. Esto puede conducir a resultados no deseados si la propiedad de presentación no está establecida a la lista vacía de forma predeterminada. En estos casos, se debería asignar un valor apropiado a cada segmento.

Pueden producirse saltos de lína en diferentes lugares como resultado de los cambios en la disposición. Los ajustes elegidos para `\alterBroken` podrían no ser adecuados para un objeto de extensión que ya no se encuentra dividido, o uno que está dividido en más segmentos que antes. Un uso explícito de `\break` puede prevenir esta situación.

La instrucción `\alterBroken` no es efectiva para propiedades de objetos de extensión accedidos antes del salto de línea, como `direction`.

Véase también

Manual de extensión de LilyPond: Sección “Difficult tweaks” en *Extender*.

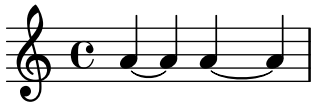
36.4.2 Establecimiento de longitudes mínimas para los objetos de extensión

La interfaz `spanner-interface` provee tres propiedades que se aplican a varios objetos de extensión.

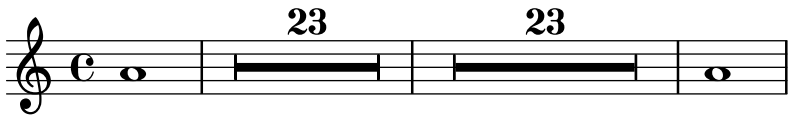
La propiedad `minimum-length` (longitud mínima)

La longitud mínima del objeto de extensión se especifica a través de la propiedad `minimum-length`. Su aumento suele producir el efecto necesario de aumentar el espaciado de las notas entre los dos puntos extremos. Sin embargo, esta sobreescritura no tiene ningún efecto sobre muchos extensores, pues su longitud está determinada por otras consideraciones. Más abajo se muestran algunos ejemplos de dónde es efectiva.

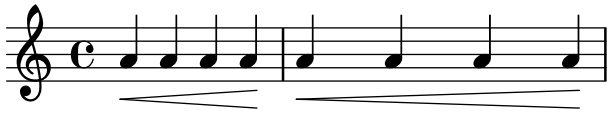
```
a'~ a'
a'
% increase the length of the tie
-\tweak minimum-length 5
~ a'
```



```
\relative \compressMMRests {
  a'1
  R1*23
  % increase the length of the rest bar
  \once \override MultiMeasureRest.minimum-length = 20
  R1*23
  a1
}
```




```
\relative {
  a' \< a a a \!
  % increase the length of the hairpin
  \override Hairpin.minimum-length = 20
  a \< a a a \!
}
```



Esta sobreescritura se puede usar también para aumentar la longitud de las ligaduras de expresión y de fraseo:

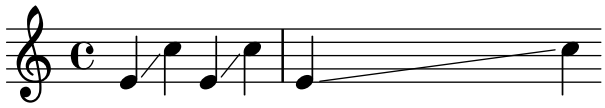
```
\relative {
  a'( g)
  a
  -\tweak minimum-length 5
  ( g)

  a\( g\)
```

```
  a
  -\tweak minimum-length 5
  \( g\)
}
```



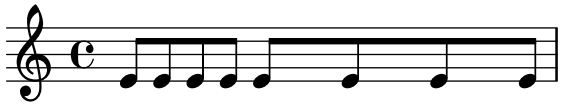
Para algunos objetos de presentación, la propiedad `minimum-length` es efectiva sólo si se llama explícitamente al procedimiento `set-spacing-rods`. Para hacerlo, se debe fijar la propiedad `springs-and-rods` al valor `ly:spanner::set-spacing-rods`. Por ejemplo, la longitud mínima de un glissando no tiene efecto a no ser que se establezca la propiedad `springs-and-rods`:

```
% default
e' \glissando c''

% not effective alone
\once \override Glissando.minimum-length = 20
e' \glissando c''

% effective only when both overrides are present
\once \override Glissando.minimum-length = 20
\once \override Glissando.springs-and-rods =
                  #ly:spanner::set-spacing-rods
e' \glissando c''
```



Lo mismo se puede decir del objeto `Beam`:

```
% not effective alone
\once \override Beam.minimum-length = 20
e'8 e' e' e'

% effective only when both overrides are present
\once \override Beam.minimum-length = 20
\once \override Beam.springs-and-rods =
                  #ly:spanner::set-spacing-rods
e'8 e' e' e'
```

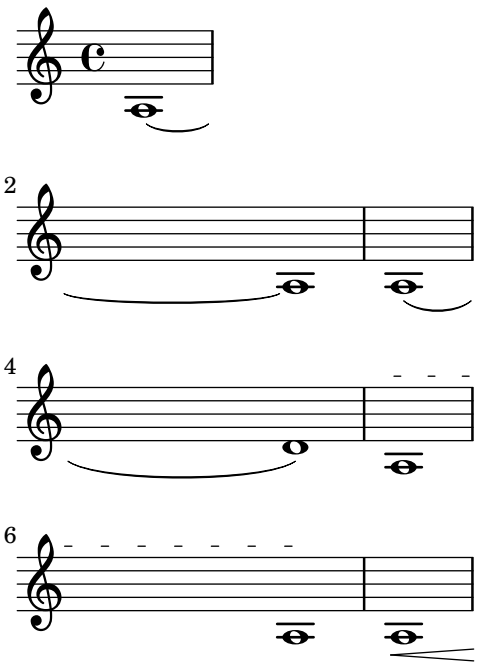


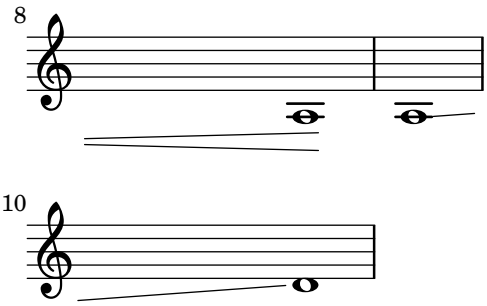
La propiedad `minimum-length-after-break`

Se puede usar la propiedad `minimum-length-after-break` para ampliar objetos de extensión fragmentados que comienzan después de un salto de línea. Respecto a la propiedad `minimum-length`, es necesaria con frecuencia para establecer la propiedad `springs-and-rods` al valor `ly:spanner::set-spacing-rods`.

```
{
  \once \override Tie.minimum-length-after-break = 20
  a1~
```

```
  \break
  a1

  \once \override Slur.minimum-length-after-break = 20
  a1(
  \break
  d'1)

  \once \override TextSpanner.springs-and-rods =
    #ly:spanner::set-spacing-rods
  \once \override TextSpanner.minimum-length-after-break = 20
  a1\startTextSpan
  \break
  a1\stopTextSpan

  \once \override Hairpin.after-line-breaking = ##t
  \once \override Hairpin.to-barline = ##f
  \once \override Hairpin.minimum-length-after-break = 20
  a1\<
  \break
  a1\!

  \once \override Glissando.springs-and-rods =
    #ly:spanner::set-spacing-rods
  % for completeness; not necessary for manual breaks
  \once \override Glissando.breakable = ##t
  \once \override Glissando.after-line-breaking = ##t
  \once \override Glissando.minimum-length-after-break = 20
  a1\glissando
  \break
  d'1
}
```

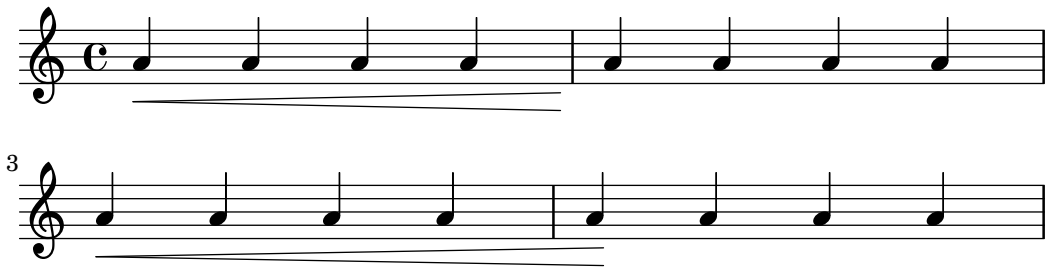




36.4.3 Control de los puntos extremos de un objeto de extensión

La propiedad `to-barline` (hasta la línea divisoria) de la interfaz `spanner-interface`, con un valor predeterminado que normalmente es `#t`, hace que los reguladores y otros objetos de extensión finalicen sobre la primera nota de un compás en lugar de hacerlo sobre la línea divisoria inmediatamente anterior. Si se establece al valor `#f`, el objeto de extensión se expande más allá de la barra de compás y termina sobre la propia nota:

```
\relative {
  a' \< a a a a \! a a a \break
  \override Hairpin.to-barline = ##f
  a \< a a a a \! a a a
}
```

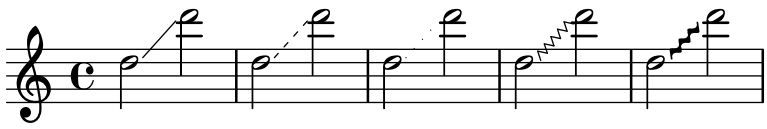


Esta propiedad no es efectiva para todos los extensores. Por ejemplo, su establecimiento a `#t` no tienen ningún efecto sobre las ligaduras de expresión o de fraseo, o sobre otros extensores para los que terminar en la barra de compás no tendría ningún significado.

36.5 Estilos de línea

La interfaz Sección “line-interface” en *Referencia de Funcionamiento Interno* engloba a todos los objetos que imprimen líneas. Todos los objetos que contemplan estas interfaces se pueden imprimir usando distintos estilos de línea. He aquí un ejemplo que muestra los valores que están disponibles.

```
\relative {
  d''2 \glissando d'2
  \once \override Glissando.style = #'dashed-line
  d,2 \glissando d'2
  \override Glissando.style = #'dotted-line
  d,2 \glissando d'2
  \override Glissando.style = #'zigzag
  d,2 \glissando d'2
  \override Glissando.style = #'trill
  d,2 \glissando d'2
}
```



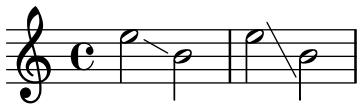
Algunos objetos pueden contemplar estilos específicos adicionales.

36.6 Líneas de extensión

Ciertas indicaciones de ejecución, p.ej. *rallentando*, *accelerando* y los trinos, se escriben como texto y se extienden sobre muchos compases mediante líneas, en ocasiones discontinuas u onduladas.

Las posiciones de los dos puntos extremos del objeto de extensión se calculan sobre la marcha, pero es posible sobreescribir sus coordenadas Y. Las propiedades que hay que especificar están anidadas dos niveles más abajo dentro de la jerarquía de propiedades, pero la sintaxis de la instrucción `\override` es bastante sencilla:

```
e''2 \glissando b'
\once \override Glissando.bound-details.left.Y = 3
\once \override Glissando.bound-details.right.Y = -2
e''2 \glissando b'
```

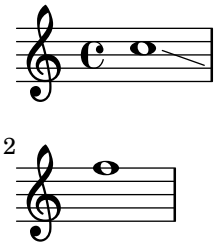


Las unidades para la propiedad `Y` son `staff-spaces`, siendo el punto del cero la línea central del pentagrama. Para el glissando, esto es el valor de `Y` en la coordenada X que corresponde al punto central de cada cabeza de nota si nos imaginamos que la línea se extiende hasta allí.

Si no está fijado `Y`, su valor se calcula a partir de la posición vertical del punto de anclaje correspondiente del extensor.

En caso de salto de línea, los valores para los puntos extremos se especifican por las sub-listas `left-broken` y `right-broken` de `bound-details`. Por ejemplo:

```
\override Glissando.bound-details.right-broken.Y = -3
c''1 \glissando \break
f''1
```



Un número de propiedades adicionales de las sub-listas `left` y `right` de la propiedad `bound-details` se pueden especificar de la misma forma que `Y`:

`Y` Establece la coordenada Y del punto extremo, medida en desplazamientos de `staff-spaces` desde la línea central del pentagrama. De forma predeterminada es el centro del objeto ancla, y así un glissando apunta al centro vertical de la cabeza de la nota.

Para extensores horizontales como los extensores de texto y los trinos, está inamoviblemente codificado como 0.

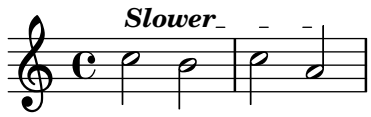
`attach-dir` (dirección de anclaje)
: Determina dónde comienza y termina la línea en la dirección X, con relación al objeto ancla. Sí, un valor de `-1` (o `LEFT`, izquierda) hace que la línea comience o termine en el lado izquierdo de la cabeza de la nota a la que está anclado.

`X`
: Es la coordenada X absoluta del punto extremo. Se suele calcular al vuelo, y su sobreescritura no tiene un efecto útil.

`stencil` (sello)
: Los extensores de línea pueden tener símbolos al comienzo o al final, lo que está contenido en esta sub-propiedad. Esto es para uso interno; se recomienda en su lugar el uso de `text`.

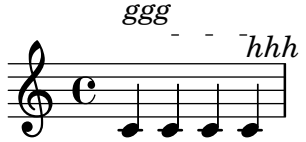
`text` (texto)
: Es un elemento de marcado que se evalúa para dar lugar al sello. Se usa para escribir *cresc.*, *tr* y otros textos sobre los objetos de extensión horizontales.

```
\override TextSpanner.bound-details.left.text
   = \markup { \small \bold Slower }
\relative { c''2\startTextSpan b c a\stopTextSpan }
```



`stencil-align-dir-y` (alineación del sello en y)
`stencil-offset` (desplazamiento del sello)
: Si no se modifican uno u otro, el sello se coloca sencillamente en el punto extremo, centrado sobre la línea, como viene definido por las subpropiedades `X` e `Y`. Si se fijan `stencil-align-dir-y` o `stencil-offset` se moverá el símbolo del borde verticalmente con relación al extremo de la línea:

```
\override TextSpanner.bound-details
          .left.stencil-align-dir-y = -2
\override TextSpanner.bound-details
          .right.stencil-align-dir-y = #UP

\override TextSpanner.bound-details.left.text = "ggg"
\override TextSpanner.bound-details.right.text = "hhh"

\relative { c'4^\startTextSpan c c c \stopTextSpan }
```



Observe que los valores negativos mueven el texto *hacia arriba*, al contrario de lo que podría esperarse, pues el valor de `-1` o `DOWN` (abajo) significa alinear el borde *inferior* del texto con la línea de extensión. Un valor de `1` o `UP` (arriba) alinea el borde superior del texto con la línea extensora.

`arrow` (flecha)
: Al establecer esta sub-propiedad a `#t` se produce una punta de flecha en el extremo de la línea.

`padding (relleno)`
: Esta sub-propiedad controla el espacio entre el punto extremo especificado de la línea y el extremo real. Sin relleno, un glissando empezaría y terminaría en el centro de la cabeza de las notas.

La función musical `\endSpanners` da por terminados prematuramente todos los objetos de extensión que están en el argumento, obedeciendo la propiedad `to-barline` (hasta la barra), si está establecida.

```
\relative c'' {
  \endSpanners c1 \> c
  \endSpanners { r4 c2.\< c1\startTextSpan } c1 c
}
```



Si se usa `\endSpanners` no es necesario cerrar \startTextSpan con \stopTextSpan, ni cerrar los reguladores con `\!`.

Véase también

Referencia de funcionamiento interno: Sección “TextSpanner” en *Referencia de Funcionamiento Interno*, Sección “Glissando” en *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” en *Referencia de Funcionamiento Interno*, Sección “TrillSpanner” en *Referencia de Funcionamiento Interno*, Sección “line-spanner-interface” en *Referencia de Funcionamiento Interno*.

36.7 Visibilidad de los objetos

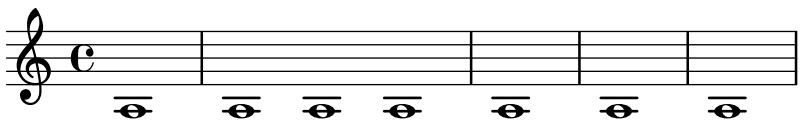
Hay cuatro formas principales en que se puede controlar la visibilidad de los objetos de presentación: se puede eliminar su sello, se pueden volver transparentes, se pueden pintar de blanco, o se puede sobreescribir su propiedad `break-visibility`. Las tres primeras se aplican a todos los objetos de presentación; la última sólo a unos pocos: los objetos *divisibles*. El Manual de aprendizaje introduce estas cuatro técnicas, véase Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Hay también algunas otras técnicas que son específicas de ciertos objetos de presentación. Se estudian bajo Consideraciones especiales.

36.7.1 Eliminar el sello

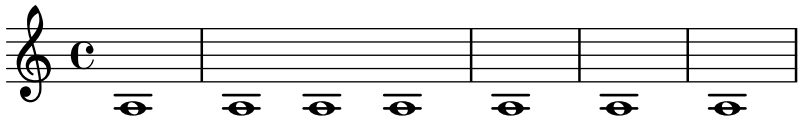
Todo objeto de presentación tiene una propiedad stencil (sello). De forma predeterminada está establecida a la función específica que dibuja ese objeto. Si se sobreescribe esta propiedad a `#f` no se llama a ninguna función y el objeto no se dibuja. La acción predeterminada se puede recuperar con `\revert`.

```
a1 a
\override Score.BarLine.stencil = ##f
a a
\revert Score.BarLine.stencil
a a a
```



Esta operación, bastante común, tiene la forma abreviada `\omit` como atajo:

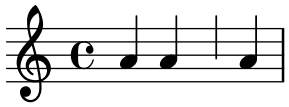
```
a1 a
\omit Score.BarLine
a a
\undo \omit Score.BarLine
a a a
```



36.7.2 Hacer transparentes los objetos

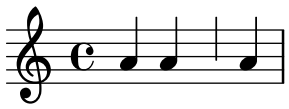
Todo objeto de presentación tiene una propiedad transparent (transparente) que de forma predeterminada está establecida a `#f`. Si se fija a `#t` el objeto aún ocupa espacio pero es invisible.

```
a'4 a'
\once \override NoteHead.transparent = ##t
a' a'
```



Esta operación, bastante común, tiene la forma abreviada `\hide` como atajo:

```
a'4 a'
\once \hide NoteHead
a' a'
```



36.7.3 Pintar los objetos de blanco

Todo objeto de presentación tiene una propiedad de color que de forma predeterminada está establecida a `black` (negro). Si se sobreescribe a `white` (blanco) el objeto será indistinguible del fondo blanco. Sin embargo, si el objeto cruza a otros objetos, el color de los puntos de cruce queda determinado por el orden en que se dibujan estos objetos, lo que puede dejar una imagen fantasma del objeto blanco, como puede verse aquí:

```
\override Staff.Clef.color = #white
a'1
```



Se puede evitar esto cambiando el orden de impresión de los objetos. Todos los objetos de presentación tienen una propiedad `layer` (capa) que se debe establecer a un valor entero. Los objetos con el valor de `layer` más bajo se dibujan primero, después se dibujan los objetos con valores progresivamente mayores, y así los objetos con valores más altos se dibujan encima de los que tienen valores más bajos. De forma predeterminada, casi todos los objetos tienen asignado un valor `layer` de `1`, aunque algunos objetos, entre ellos el pentagrama y las líneas divisorias, `StaffSymbol` y `BarLine`, tienen asignado un calor de `0`. El orden de impresión de los objetos con el mismo valor de `layer` es indeterminado.

En el ejemplo de arriba, la clave blanca, con un valor `layer` predeterminado de `1`, se dibuja después de las líneas del pentagrama (valor `layer` predeterminado de `0`), sobreimpresionándolas. Para cambiarlo, se debe dar al objeto `Clef` un valor de `layer` más bajo, digamos `-1`, para que se dibuje antes:

```
\override Staff.Clef.color = #white
\override Staff.Clef.layer = -1
a'1
```



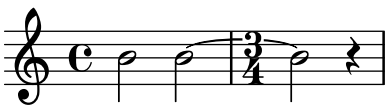
Fragmentos de código seleccionados

Uso de la propiedad `whiteout`

Se puede imprimir cualquier objeto gráfico sobre un fondo blanco para enmascarar parte de los objetos que están por debajo. Esto puede ser útil para mejorar el aspecto de las colisiones en situaciones complejas cuando no es práctico reposicionar los objetos. Es necesario establecer explícitamente la propiedad de capa (`layer`) para controlar qué objetos resultan enmascarados por el fondo blanco.

En este ejemplo, la colisión de la ligadura de unión con la indicación de compás resulta mejorada enmascarando la parte de la ligadura que cruza a la indicación de compás estableciendo la propiedad `whiteout` de `TimeSignature`. Para hacer esto, se mueve `TimeSignature` a una capa por encima de `Tie`, que se deja en la capa predeterminada 1, y `StaffSymbol` se mueve a una capa por encima de `TimeSignature` de manera que no resulte enmascarada.

```
{
  \override Score.StaffSymbol.layer = 4
  \override Staff.TimeSignature.layer = 3
  b'2 b'~
  \once \override Staff.TimeSignature.whiteout = ##t
  \time 3/4
  b' r4
}
```



36.7.4 Uso de break-visibility (visibilidad en el salto)

Casi todos los objetos de presentación se imprimen una sola vez, pero algunos como las líneas divisorias, claves, indicaciones de compás y armaduras de tonalidad, se pueden tener que imprimir dos veces cuando se produce un salto de línea : una vez al final de la línea y otra al comienzo de la siguiente. Estos objetos reciben el nombre de *divisibles*, y tienen una propiedad, `break-visibility` (visibilidad en el salto), para controlar su visibilidad en las tres posiciones en que pueden aparecer: al comienzo de una línea, dentro de la línea si se produce un cambio, y al final de la línea si el cambio se produce en ese lugar.

Por ejemplo, la indicación de compás se imprime de forma predeterminada al comienzo de la primera línea y en ningún otro lugar a no ser que cambie, en cuyo caso se imprime en el punto en que se produce el cambio. Si este cambio se produce al final de una línea, la nueva indicación de compás se imprime al principio de la línea siguiente y también al final de la línea anterior como indicación de precaución.

Este comportamiento se controla por medio de la propiedad `break-visibility`, que se explica en Sección "Visibilidad y color de los objetos" en *Manual de Aprendizaje*. Esta propiedad toma un vector de tres valores booleanos que, por orden, determinan si el objeto se imprime al final, dentro, o al principio de la línea. O, para ser más exactos: antes del salto de línea, si no hay salto, o después del salto.

Como alternativa se puede especificar cualquiera de las ocho combinaciones mediante funciones predefinidas cuya definición está en `scm/output-lib.scm`, donde las tres últimas columnas indican si los objetos de presentación serán visibles en las posiciones que se muestran en el encabezamiento de cada columna:

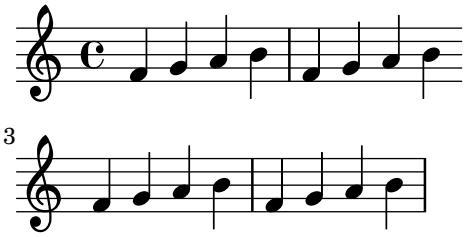
Forma de función	**Forma de vector**	**Antes del salto**	**Si no hay salto**	**Después del salto**
`all-visible`	`#(#t #t #t)`	sí	sí	sí
`begin-of-line-visible`	`#(#f #f #t)`	no	no	sí
`center-visible`	`#(#f #t #f)`	no	sí	no
`end-of-line-visible`	`#(#t #f #f)`	sí	no	no
`begin-of-line-invisible`	`#(#t #t #f)`	sí	sí	no
`center-invisible`	`#(#t #f #t)`	sí	no	sí
`end-of-line-invisible`	`#(#f #t #t)`	no	sí	sí
`all-invisible`	`#(#f #f #f)`	no	no	no

Los ajustes predeterminados de `break-visibility` dependen del objeto de presentación. La tabla siguiente muestra todos los objetos de presentación de interés que resultan afectados por `break-visibility` y el ajuste predeterminado de esta propiedad:

Objeto	**Contexto usual**	**Valor predet.**
`BarLine`	`Score`	calculado
`BarNumber`	`Score`	`begin-of-line-visible`
`BreathingSign`	`Voice`	`begin-of-line-invisible`
`Clef`	`Staff`	`begin-of-line-visible`
`Custos`	`Staff`	`end-of-line-visible`
`Divisio`	`Staff`	`begin-of-line-invisible`
`DoublePercentRepeat`	`Voice`	`begin-of-line-invisible`
`KeyCancellation`	`Staff`	`begin-of-line-invisible`
`KeySignature`	`Staff`	`begin-of-line-visible`
`ClefModifier`	`Staff`	`begin-of-line-visible`
`RehearsalMark`	`Score`	`end-of-line-invisible`
`TimeSignature`	`Staff`	`all-visible`

El ejemplo de abajo muestra el uso de la forma de vector para controlar la visibilidad de las líneas divisorias:

```
\relative {
  f'4 g a b
  f4 g a b
  % Remove bar line at the end of the current line
  \once \override Score.BarLine.break-visibility = ##(#f #t #t)
  \break
  f4 g a b
  f4 g a b
}
```



Aunque los tres componentes del vector utilizado para sobreescribir `break-visibility` deben estar presentes, no todos son efectivos para todos los objetos de presentación, y algunas combinaciones pueden incluso dar errores. Son de aplicación las siguientes limitaciones:

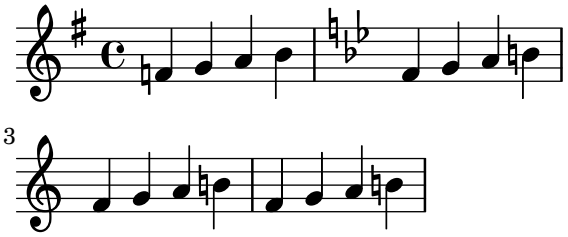
- Las líneas divisorias no se pueden imprimir al principio de la línea.
- No se puede imprimir el número de compás al principio de la *primera* línea a no ser que su valor establecido sea distinto de `1`.
- Clave: véase la sección siguiente.
- Las repeticiones dobles de tipo porcentaje se imprimen todas siempre o bien no se imprimen nunca. Utilice `begin-of-line-invisible` para imprimirlas y `all-invisible` para suprimirlas.
- Armadura: véase la sección siguiente.
- ClefModifier: véase la sección siguiente.

36.7.5 Consideraciones especiales

Visibilidad después de un cambio explícito

La propiedad `break-visibility` controla la visibilidad de las armaduras y cambios de clave sólo al principio de las líneas, es decir, después de un salto. No tiene ningún efecto sobre la visibilidad de la armadura o la clave después de un cambio explícito de tonalidad o de clave dentro o al final de una línea. En el ejemplo siguiente la armadura que sigue al cambio explícito de tonalidad a Si bemol mayor es visible incluso con `all-invisible` establecido.

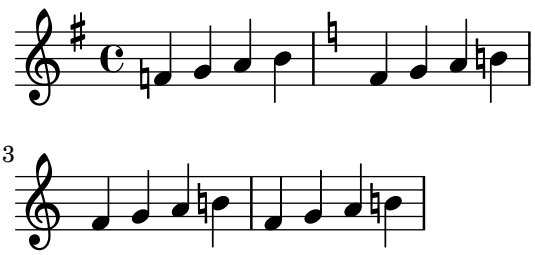
```
\relative {
  \key g \major
  f'4 g a b
  % Try to remove all key signatures
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b
  \break
  f4 g a b
  f4 g a b
}
```



La visibilidad de estos cambios explícitos de tonalidad y de clave se controla por medio de las propiedades `explicitKeySignatureVisibility` y `explicitClefVisibility`. Son los equivalentes de la propiedad `break-visibility` y las dos toman un vector de tres valores booleanos o

las funciones predefinidas relacionadas anteriormente, exactamente igual que `break-visibility`. Las dos son propiedades del contexto Staff, no de los propios objetos de presentación, y por tanto se establecen utilizando la instrucción `\set`. Las dos están establecidas de forma predeterminada al valor `all-visible`. Estas propiedades controlan sólo la visibilidad de las armaduras y las claves que resultan de los cambios explícitos y no afectan a las armaduras y tonalidades que están al principio de las líneas; para quitarlas, aún se debe sobreescribir la propiedad `break-visibility` en el objeto correspondiente.

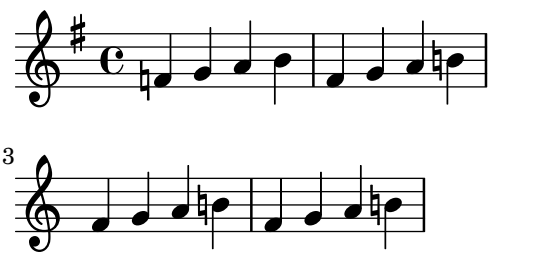
```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Visibilidad de las alteraciones de precaución

Para eliminar las alteraciones de cancelación que se imprimen en un cambio de tonalidad explícito, establezca la propiedad `printKeyCancellation` del contexto Staff a `#f`:

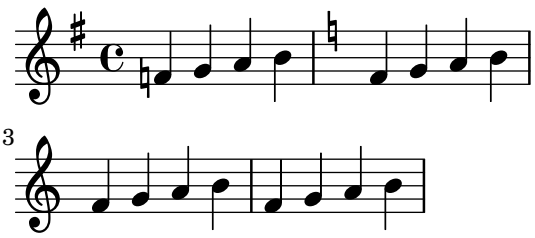
```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \set Staff.printKeyCancellation = ##f
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Con estas sobreescrituras solamente permanecen las alteraciones accidentales delante de las notas para indicar el cambio de tonalidad.

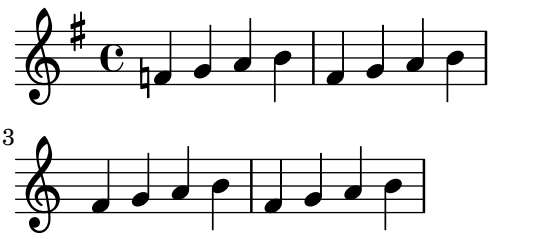
Observe que al modificar la tonalidad a Do mayor o La menor, las alteraciones de cancelación serían *la única* indicación del cambio de armadura. En este caso, el establecimiento de `printKeyCancellation` al valor `#f` no tiene ningún efecto:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \set Staff.printKeyCancellation = ##f
  \key c \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



En lugar de ello, para suprimir las alteraciones de cancelación aún cuando la armadura cambia a Do mayor o a La menor, sobreescriba la visibilidad del grob `KeyCancellation`:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \override Staff.KeyCancellation.break-visibility = #all-invisible
  \key c \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Claves transportadas

El pequeño símbolo de transposición sobre una clave se produce por parte del objeto `ClefModifier`. Su visibilidad se hereda automáticamente de la del objeto `Clef`, así que no es necesario aplicar las sobreescrituras correspondientes `break-visibility` a los objetos `ClefModifier` para suprimir los símbolos de transposición u octavación para las claves invisibles.

Para los cambios de clave explícitos, la propiedad `explicitClefVisibility` controla tanto el símbolo de clave como el símbolo de transposición o de octava asociado.

Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*

36.8 Rotación de objetos

Tanto los objetos de presentación como los elementos de texto de marcado se pueden girar cualquier ángulo respecto a cualquier punto, pero difiere el método de hacerlo.

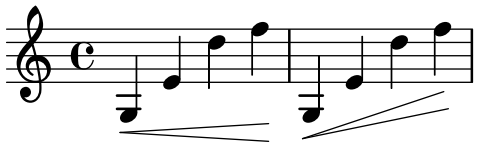
36.8.1 Rotación de objetos de presentación

Todos los objetos de presentación que contemplan el interface `grob-interface` se pueden rotar estableciendo su propiedad `rotation`. Acepta una lista de tres elementos: el ángulo de la rotación en sentido antihorario, y las coordenadas x e y del punto con relación al punto de referencia del objeto, alrededor del que se va a realizar la rotación. El ángulo de rotación se especifica en grados y las coordenadas en espacios de pentagrama.

El ángulo de rotación y las coordenadas del punto de rotación se deben determinar por ensayo y error.

Solamente en ciertas ocasiones es útil la rotación de objetos de presentación; el ejemplo siguiente muestra una situación en que puede serlo:

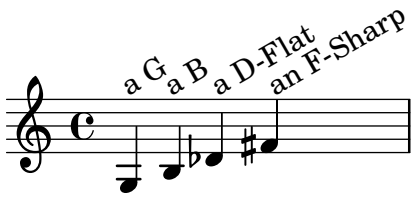
```
g4\< e' d'' f''\!
\override Hairpin.rotation = #'(15 -1 0)
g4\< e' d'' f''\!
```



36.8.2 Rotación de elementos de marcado

Todos los textos de marcado se pueden rotar para que se dispongan en cualquier ángulo precediéndolos de la instrucción `\rotate`. La instrucción acepta dos argumentos: el ángulo de rotación en grados en sentido antihorario, y el texto que rotar. Los límites que ocupa el texto no se rotan: toman su valor a partir de los extremos de las coordenadas x e y del texto rotado. En el ejemplo siguiente la propiedad `outside-staff-priority` del texto se establece a `#f` para desactivar la evitación automática de colisiones, lo que empuja al texto a una posición muy alta.

```
\override TextScript.outside-staff-priority = ##f
g4^\markup { \rotate #30 "a G" }
b^\markup { \rotate #30 "a B" }
des'^\markup { \rotate #30 "a D-Flat" }
fis'^\markup { \rotate #30 "an F-Sharp" }
```



36.9 Alineación de objetos

Los objetos gráficos que soportan el interface `self-alignment-interface` y/o el `side-position-interface` se pueden alinear contra un objeto colocado previamente, de diversas maneras. Para ver una lista de estos objetos, consulte Sección “self-alignment-interface” en *Referencia de Funcionamiento Interno* y Sección “side-position-interface” en *Referencia de Funcionamiento Interno*.

Todos los objetos gráficos tienen un punto de referencia, una extensión horizontal y una extensión vertical. La extensión horizontal es una pareja de números que dan los desplazamientos a partir del punto de referencia de los bordes izquierdo y derecho, siendo negativos los desplazamientos hacia la izquierda. La extensión vertical es una pareja de números que dan el desplazamiento a partir del punto de referencia hasta los bordes inferior y superior, siendo negativos los desplazamientos hacia abajo.

La posición de un objeto sobre el pentagrama viene dada por los valores de las propiedades `X-offset` e `Y-offset`. El valor de `X-offset` da el desplazamiento desde la coordenada X del punto de referencia del objeto padre, y el valor de `Y-offset` da el desplazamiento a partir de la línea central del pentagrama. Los valores de `X-offset` y `Y-offset` se pueden establecer directamente o se puede dejar que se calculen por parte de procedimientos para conseguir una alineación con el objeto padre.

> **Nota:** Muchos objetos tienen consideraciones de posicionamiento especiales que hacen que se ignore o se modifique cualquier ajuste realizado a `X-offset` o a `Y-offset`, a pesar de que el objeto contemple el interface `self-alignment-interface`. La sobreescritura de las propiedades `X-offset` ó `Y-offset` a un valor fijo hace que se descarte la respectiva propiedad `self-alignment`.

Por ejemplo, una alteración accidental se puede reposicionar verticalmente estableciendo `Y-offset` pero los cambios a `X-offset` no tienen ningún efecto.

Las letras de ensayo se pueden alinear con objetos divisibles como líneas divisorias, claves, armaduras e indicaciones de compás. Hay propiedades especiales que se encuentran en `break-aligned-interface` para el posicionamiento de las letras de ensayo sobre dichos objetos.

Véase también

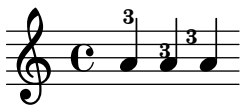
Referencia de la notación: Sección 36.9.4 [Uso del `break-alignable-interface`], página 804.

Manual de extensión: Sección “Funciones de callback” en *Extender*.

36.9.1 Establecer directamente `X-offset` y `Y-offset`

Se pueden dar valores numéricos a las propiedades `X-offset` y `Y-offset` de muchos objetos. El ejemplo siguiente muestra tres notas con una digitación en su posición predeterminada y con los valores `X-offset` y `Y-offset` modificados.

```
a'-3
a'
-\tweak X-offset 0
-\tweak Y-offset 0
-3
a'
-\tweak X-offset -1
-\tweak Y-offset 1
-3
```



36.9.2 Uso del side-position-interface

Un objeto que contempla el side-position-interface se puede colocar junto a su objeto padre de forma que los bordes especificados de los dos objetos se toquen. El objeto se puede situar encima, debajo, a la derecha o a la izquierda del objeto padre. El padre no se puede especificar; viene determinado por el orden de los elementos en el flujo de entrada. Casi todos los objetos tienen la cabeza de la nota asociada como padre.

Los valores de las propiedades side-axis y direction determinan dónde colocar el objeto, como sigue:

propiedad side-axis	**propiedad** direction	**colocación**
0	-1	izquierda
0	1	derecha
1	-1	debajo
1	1	encima

Si side-axis es 0, X-offset se debe establecer al procedimiento ly:side-position-interface::x-aligned-side. Este procedimiento devuelve el calor correcto de X-offset para situar el objeto al lado izquierdo o derecho del padre de acuerdo con el valor de direction.

Si side-axis es 1, Y-offset se debe establecer al procedimiento ly:side-position-interface::y-aligned-side. Este procedimiento devuelve el valor correcto de Y-offset para situar el objeto encima o debajo del padre de acuerdo con el valor de direction.

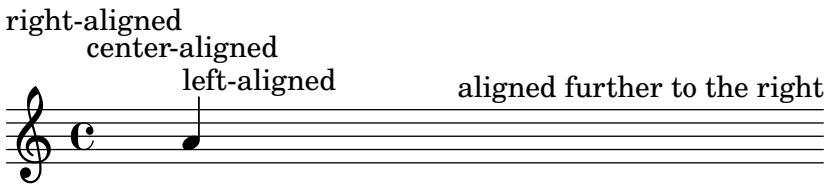
36.9.3 Uso del self-alignment-interface

Auto-alineación horizontal de los objetos

La alineación horizontal de un objeto que contempla el interface self-alignment-interface está controlada por el valor de la propiedad self-alignment-X, siempre y cuando la propiedad X-offset de este objeto esté establecida a ly:self-alignment-interface::x-aligned-on-self. Se le puede dar a self-alignment-X cualquier valor real, en unidades de la mitad de la extensión X total del objeto. Los valores negativos mueven el objeto a la derecha, los positivos hacia la izquierda. Un valor de 0 centra el objeto sobre el punto de referencia de su padre, un valor de -1 alinea el borde izquierdo del objeto sobre el punto de referencia de su padre, y un valor de 1 alinea el borde derecho del objeto sobre el punto de referencia de su padre. Se pueden usar los símbolos LEFT, CENTER y RIGHT en sustitución de los valores -1, 0 y 1, respectivamente.

Normalmente se usaría la instrucción \override para modificar el valor de self-alignment-X, pero se puede usar la instrucción \tweak para alinear varias anotaciones por separado sobre una sola nota:

```
a'
-\tweak self-alignment-X -1
-"left-aligned"
-\tweak self-alignment-X 0
-"center-aligned"
-\tweak self-alignment-X #RIGHT
-"right-aligned"
-\tweak self-alignment-X -2.5
-"aligned further to the right"
```



Auto-alineación vertical de los objetos

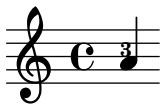
Los objetos se pueden alinear verticalmente en una forma análoga a la alineación horizontal si la propiedad `Y-offset` está establecida a `ly:self-alignment-interface::y-aligned-on-self`. Sin embargo, a menudo se encuentran implicados otros mecanismos en la alineación vertical: el valor de `Y-offset` es tan sólo una variable que se tiene en cuenta. Esto puede hacer que ajustar el valor de ciertos objetos sea una tarea dificultosa. Las unidades son sólo la mitad de las dimensiones verticales del objeto, que suele ser bastante pequeño, por lo que pueden requerirse números bastante grandes. Un valor de `-1` alinea el borde inferior del objeto con el punto de referencia del objeto padre, un valor de `0` alinea el centro del objeto con el punto de referencia del padre, y un valor de `1` alinea el borde superior del objeto con el punto de referencia del padre. Se pueden usar los símbolos `DOWN`, `CENTER`, `UP` en sustitución de `-1`, `0` y `1`, respectivamente.

Auto-alineación de objetos en las dos direcciones

Estableciendo tanto `X-offset` como `Y-offset`, se puede alinear un objeto en las dos direcciones simultáneamente.

El ejemplo siguiente muestra cómo ajustar una digitación de forma que se acerque a la cabeza dela nota.

```
a'
-\tweak self-alignment-X 0.5  % move horizontally left
-\tweak Y-offset #ly:self-alignment-interface::y-aligned-on-self
-\tweak self-alignment-Y -1  % move vertically up
-3  % third finger
```



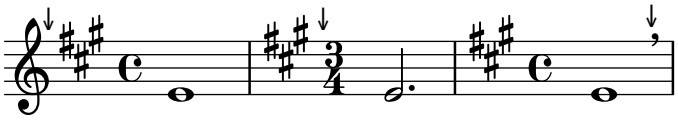
36.9.4 Uso del `break-alignable-interface`

Las llamadas de ensayo y de texto, los números de compás, y de forma más general todos los objetos con la interfaz Sección “break-alignable-interface” en *Referencia de Funcionamiento Interno*, a los cuales nos referimos colectivamente como “objetos alineables en el salto de línea”, se pueden alinear con objetos de notación tales como líneas divisorias, armaduras de tonalidad, indicaciones de compás y de forma general con cualquier objeto que tenga la interfaz Sección “break-aligned-interface” en *Referencia de Funcionamiento Interno*. Para ser más exactos, los elementos break-aligned (alineados con el salto) tienen una propiedad `break-align-symbol`, que aporta símbolos que se pueden usar como ‘puntos de anclaje’ para otros objetos por medio de la propiedad `break-align-symbols`. Consulte Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno* para ver una lista completa de los símbolos disponibles.

Cada tipo de objeto tiene su propio punto de referencia predeterminado, con el que se alinean las marcas de ensayo:

```
\override Score.TextMark.self-alignment-X = #CENTER
% The text mark will be aligned
% to the right edge of the clef
\override Score.TextMark.break-align-symbols =
            #'(clef)
```

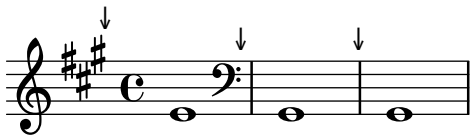
```
\key a \major
\clef treble
\textMark "↓"
e'1
% The text mark will be aligned
% to the left edge of the time signature
\override Score.TextMark.break-align-symbols =
           #'(time-signature)
\key a \major
\clef treble
\time 3/4
\textMark "↓"
e'2.
% The text mark will be centered
% above the breathing mark
\override Score.TextMark.break-align-symbols =
           #'(breathing-sign)
\key a \major
\clef treble
\time 4/4
e'1
\breathe
\textEndMark "↓"
```



Se puede especificar una lista de posibles objetos para la alineación. Si algunos de los objetos son invisibles en ese punto debido al valor de `break-visibility` o a valores de visibilidad explícitos para las armaduras y las claves, la letra de ensayo o número de compás se alinean con el primer objeto de la lista que sea visible. Si ningún objeto de la lista es visible, el objeto se alinea con la línea divisoria. Si la línea divisoria es invisible, el objeto se alinea con el punto en el que se encontraría la línea divisoria.

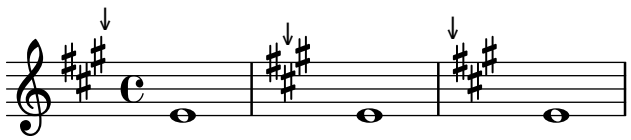
```
\override Score.TextMark.self-alignment-X = #CENTER
% The text mark will be aligned
% to the right edge of the key signature
\override Score.TextMark.break-align-symbols =
           #'(key-signature clef)
\key a \major
\clef treble
\textMark "↓"
e'1
% The text mark will be aligned
% to the right edge of the clef
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.TextMark.break-align-symbols =
           #'(key-signature clef)
\key a \major
\clef bass
\textMark "↓"
gis,1
```

```
% The text mark will be centered
% above the bar line
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.explicitClefVisibility = #all-invisible
\override Score.TextMark.break-align-symbols =
           #'(key-signature clef)
\key a \major
\clef treble
\textMark "↓"
e'1
```



La alineación del elemento break-alignable con relación al objeto de notación se puede cambiar, como se ve en el ejemplo siguiente. En una partitura con varios pentagramas, este ajuste se debe hacer para todos los pentagramas.

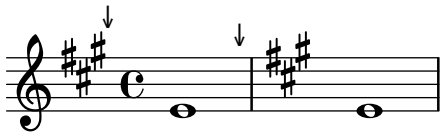
```
\override Score.TextMark.self-alignment-X = #CENTER
% The text mark will be aligned
% with the right edge of the key signature
\override Score.TextMark.break-align-symbols =
           #'(key-signature)
\key a \major
\clef treble
\time 4/4
\textMark "↓"
e'1
% The text mark will be centered
% above the key signature
\once \override Score.KeySignature.break-align-anchor-alignment =
                 #CENTER
\textMark "↓"
\key a \major
e'1
% The text mark will be aligned
% with the left edge of the key signature
\once \override Score.KeySignature.break-align-anchor-alignment =
                 #LEFT
\key a \major
\textMark "↓"
e'1
```



El elemento break-alignable también se puede desplazar al borde derecho o izquierdo en una medida arbitraria. Las unidades son espacios de pentagrama:

```
\override Score.TextMark.self-alignment-X = #CENTER
% The text mark will be aligned
```

```
% with the left edge of the key signature
% and then shifted right by 3.5 staff spaces
\override Score.TextMark.break-align-symbols =
           #'(key-signature)
\once \override Score.KeySignature.break-align-anchor = 3.5
\key a \major
\textMark "↓"
e'1
% The text mark will be aligned
% with the left edge of the key signature
% and then shifted left by 2 staff spaces
\once \override Score.KeySignature.break-align-anchor = -2
\key a \major
\textMark "↓"
e'1
```



36.10 Modificación de los sellos

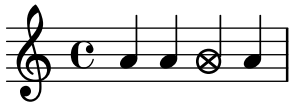
Todos los objetos de presentación tienen una propiedad `stencil` que es parte del `grob-interface`. De forma predeterminada, esta propiedad suele estar establecida a una función específica del objeto que está hecha a medida para disponer el símbolo que lo representa en la salida. Por ejemplo, el ajuste estándar para la propiedad `stencil` del objeto `MultiMeasureRest` es `ly:multi-measure-rest::print`.

El símbolo estándar de cualquier objeto se puede sustituir modificando la propiedad `stencil` para que haga referencia a un procedimiento diferente escrito especialmente. Esto requiere un alto nivel de conocimiento del funcionamiento interno de LilyPond, pero hay una forma más fácil que a menudo puede dar resultados adecuados.

El procedimiento es establecer la propiedad `stencil` al procedimiento que imprime texto (`ly:text-interface::print`) y añadir una propiedad `text` al objeto ajustada para que contenga el texto de marcado que produce el símbolo requerido. Debido a la flexibilidad del marcado, se pueden conseguir muchas cosas; en particular, consulte Sección 8.2.4 [Notación gráfica dentro de elementos de marcado], página 336.

El ejemplo siguiente muestra esto cambiando el símbolo de la cabeza de la nota a unas aspas dentro de una circunferencia.

```
XinO = {
  \once \override NoteHead.stencil = #ly:text-interface::print
  \once \override NoteHead.text = \markup {
    \combine
      \halign #-0.7 \draw-circle #0.85 #0.2 ##f
      \musicglyph "noteheads.s2cross"
  }
}
\relative {
  a' a \XinO a a
}
```



Cualquiera de los glifos de *Feta* usados en la fuente tipográfica Emmentaler se puede pasar a la instrucción de marcado `\musicglyph`: véase Sección B.8 [La tipografía Emmentaler], página 912.

Se pueden insertar en línea tanto imágenes en formato `EPS` como instrucciones de PostScript usando las instrucciones de marcado `\epsfile` y `\postscript` respectivamente; véase Sección A.1.3 [Graphical markup], página 845.

Véase también

Referencia de la notación: Sección 8.2.4 [Notación gráfica dentro de elementos de marcado], página 336, Sección 8.2 [Formatear el texto], página 323, Sección A.1 [Instrucciones de marcado de texto], página 815, Sección B.8 [La tipografía Emmentaler], página 912, Sección A.1.3 [Graphical markup], página 845.

36.11 Modificación de las formas

36.11.1 Modificación de ligaduras de unión y de expresión

Las ligaduras de unión (`Ties`), de expresión (`Slurs`), de fraseo (`PhrasingSlurs`), de dejar vibrar (`LaissezVibrerTies`) y de repetición (`RepeatTies`) se trazan como curvas de Bézier de tercer orden. Si la forma de la ligadura calculada automáticamente no resulta óptima, se puede modificar su forma manualmente de dos formas:

a. mediante la especificación de los desplazamientos que se quieren realizar sobre los puntos de control de la curva de Bézier calculada automáticamente, o bien
b. mediante la especificación explícita de las posiciones de los cuatro puntos de control que se requieren para definir la curva deseada.

A continuación se explican ambos métodos. El primer método es más adecuado cuando solo se requieren ligeros ajustes de la curva; el segundo puede ser mejor para crear curvas que tienen relación con una única nota.

Curvas de Bézier cúbicas

Las curvas Bézier de tercer orden o cúbicas están definidas por cuatro puntos de control. El primer y cuarto puntos de control son exactamente los puntos extremos de comienzo y de final de la curva. Los dos puntos de control intermedios definen la forma. Se pueden encontrar en la web animaciones que muestran cómo se traza la curva, pero la descripción y la imagen siguiente puede ser de ayuda. La curva comienza a partir del primer punto de control dirigiéndose directamente hacia el segundo, curvándose progresivamente para dirigirse hacia el tercero y continuando la curva hacia el cuarto, llegando a éste en viaje directo desde el tercer punto de control. La curva está contenida enteramente dentro del cuadrilátero definido por los cuatro puntos de control.



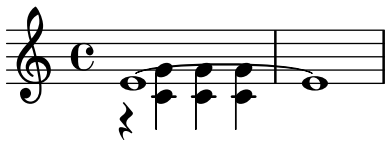
Todas las traslaciones, rotaciones y escalado de los puntos de control producen exactamente las mismas operaciones sobre la curva.

Especificar desplazamientos a partir de los puntos de control actuales

En este ejemplo, la colocación automática de la ligadura no resulta óptima, y `\tieDown` no es la solución.

```
<<
```

```
  { e'1~ 1 }
\\
  \relative { r4 <g' c,> <g c,> <g c,> }
>>
```



El ajuste de los puntos de control de la ligadura mediante la instrucción \shape permite evitar las colisiones.

La sintaxis de \shape es:

[-]\shape *desplazamientos elemento*

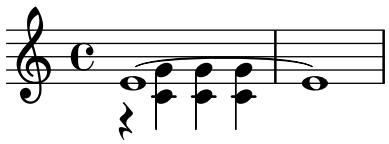
Esta instrucción recoloca los puntos de control de *elemento* en las cantidades dadas por *desplazamientos*. El argumento *desplazamientos* es una lista de parejas de números o una lista de tales listas. Cada elemento de una pareja representa el desplazamiento de una de las coordenadas de un punto de control. Si el *elemento* es una cadena de caracteres, el resultado es \once\override para el tipo de grob especificado. Si *elemento* es una expresión musical, el resultado es la misma expresión musical con la consiguiente modificación aplicada.

En otras palabras, la función \shape puede actuar como una instrucción \once\override o como una instrucción \tweak dependiendo de si el argumento *elemento* es el nombre de un grob, como "Slur", o una expresión musical, como "(". El argumento *desplazamientos* especifica los desplazamientos de los cuatro puntos de control como una lista de cuatro parejas de valores (dx . dy) en unidades de espacios de pentagrama (o una lista de tales listas si la curva tiene más de un segmento).

El guión precedente se exige si, y sólo si, se usa la forma \tweak.

Así, usando el mismo ejemplo de arriba y la forma \once\override de la instrucción \shape, lo siguiente tiene el efecto de elevar la ligadura de unión en la mitad de un espacio de pentagrama:

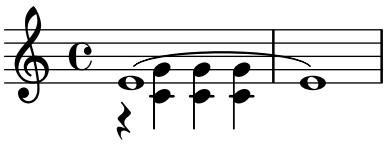
```
<<
  {
    \shape #'((0 . 0.5) (0 . 0.5) (0 . 0.5) (0 . 0.5)) Tie
    e'1~ 1
  }
\\
  \relative { r4 <g' c,> <g c,> <g c,> }
>>
```



Este posicionamiento de la ligadura de unión es mejor, pero quizá debería elevarse más en la parte central. El ejemplo siguiente hace esto, esta vez usando la forma alternativa \tweak:

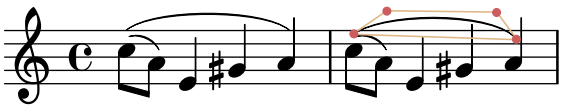
```
<<
  {
    e'1-\shape #'((0 . 0.5) (0 . 1) (0 . 1) (0 . 0.5)) ~ e'
  }
\\
  \relative { r4 <g' c,> <g c,> <g c,> }
```

```
>>
```



Como ayuda para el proceso de trucaje, se ofrece la función \vshape. Su nombre significa *forma visual*: actúa exactamente igual que \shape, excepto que los puntos y el polígono de control se imprimen por añadidiura.

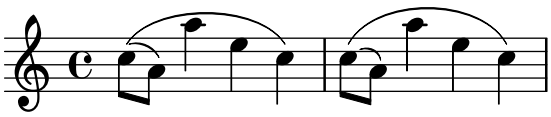
```
\relative {
  c''8\( a) e4 gis a\)
  \vshape #'((0 . -0.3) (0.5 . -0.2)
             (0.5 . -0.3) (0 . -0.7)) PhrasingSlur
  c8\( a) e4 gis a\)
}
```



Se recomienda empezar con \vshape e ir ajustando hasta que se consigue una curva satisfactoria, para después sencillamente borrar la letra "v" de la instrucción \vshape.

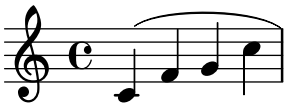
También es posible dar forma a dos curvas diferentes que dan comienzo en el mismo momento musical:

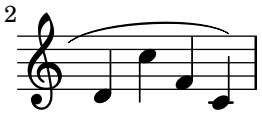
```
\relative {
  c''8\( a) a'4 e c\)
  \shape #'((0.7 . -0.4) (0.5 . -0.4)
            (0.3 . -0.3) (0 . -0.2)) Slur
  \shape #'((0 . 0) (0 . 0.5)
            (0 . 0.5) (0 . 0)) PhrasingSlur
  c8\( a) a'4 e c\)
}
```



La función \shape también puede desplazar los puntos de control de las curvas que se extienden atravesando saltos de línea. Cada pieza de la curva dividida puede recibir su propia lista de desplazamientos. Si no se necesita hacer cambios a uno de los segmentos en particular, se puede usar una lista vacía como contenedor. En este ejemplo, el salto de línea hace que la única ligadura de expresión aparezca como si fueran dos:

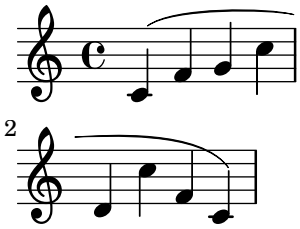
```
\relative {
  c'4( f g c
  \break
  d,4 c' f, c)
}
```





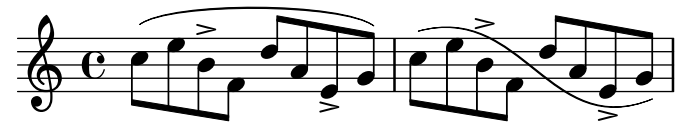
Los cambios en la forma de las dos mitades de la ligadura de expresión deja más claro que la ligadura continúa más allá del salto de línea:

```
% () may be used as a shorthand for ((0 . 0) (0 . 0) (0 . 0) (0 . 0))
% if any of the segments does not need to be changed
\relative c' {
  \shape #'(
             (( 0 . 0) (0 . 0) (0 . 0) (0 . 1))
             ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
           ) Slur
  c4( f g c
  \break
  d,4 c' f, c)
}
```



En una curva en forma de «S» siempre es necesario ajustar manualmente los puntos de control: LilyPond nunca escoge tales formas automáticamente.

```
\relative c'' {
  c8( e b-> f d' a e-> g)
  \shape #'((0 . -1) (5.5 . -0.5) (-5.5 . -10.5) (0 . -5.5))
         PhrasingSlur
  c8\( e b-> f d' a e-> g\)
}
```



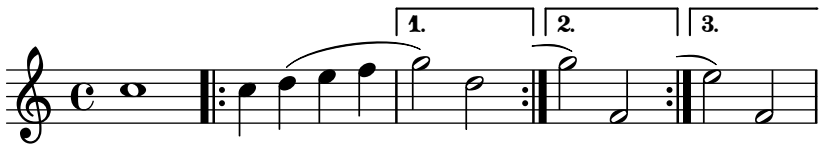
Especificar los puntos de control explícitamente

Las coordenadas de los puntos de contro de Bézier se especifican en unidades de espacios de pentagrama. La coordenada X es relativa al punto de referencia al que se adjunta la ligadura, y la coordenada Y es relativa a la línea central del pentagrama. Las coordenadas se especifican como una lista de cuatro parejas de números decimales (reales). Un enfoque consiste en estimar las coordenadas de los dos extremos, y después adivinar los dos puntos intermedios. Los valores óptimos se encuentran después por ensayo y error. Tenga en cuenta que estos valores podrían requerir un ajuste manual si se hace posteriormente cualquier cambio manual a la música o a la disposición.

Una situación en la que es preferible especificar los puntos de control explícitamente a especificar los desplazamientos es cuando se necesita especificarlos de forma relativa a una nota única. A continuación presentamos un ejemplo de esto. Muestra una forma de indicar una ligadura que se extiende hacia el interior de las casillas de primera y segunda vez de una repetición.

```
\relative {
```

```
  c''1
  \repeat volta 3 { c4 d( e f }
  \alternative {
    \volta 1 { g2) d }
    \volta 2 {
      g2
        % create a slur and move it to a new position
        % the <> is just an empty chord to carry the slur termination
        -\tweak control-points
                  #'((-2 . 3.8) (-1 . 3.9) (0 . 4) (1 . 3.4)) ( <> )
      f,
    }
    \volta 3 {
      e'2
        % create a slur and move it to a new position
        -\tweak control-points
                  #'((-2 . 3) (-1 . 3.1) (0 . 3.2) (1 . 2.4)) ( <> )
      f,
    }
  }
}
```



Advertencias y problemas conocidos

No es posible modificar la forma de las ligaduras de unión o de expresión cambiando la propiedad `control-points` si hay más de una en el mismo momento musical, ni siquiera usando la instrucción `\tweak`. Sin embargo, se puede sobreescribir la propiedad `tie-configuration` de `TieColumn` para fijar la línea de inicio y la dirección según se requiera.

Véase también

Referencia de funcionamiento interno: Sección “TieColumn” en *Referencia de Funcionamiento Interno*.

Apéndices

Apéndice A Instrucciones de marcado

A.1 Instrucciones de marcado de texto

The following commands can all be used inside `\markup { }`.

A.1.1 Font markup

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size (in points) to display *arg*.

This function adjusts the `baseline-skip` and `word-space` properties accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```



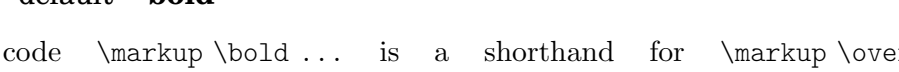
Used properties:

- `baseline-skip` (3)
- `word-space` (0.6)

`\bold` *arg* (markup)

Print *arg* with a bold face.

```
\markup {
  default
  \hspace #2
  \bold bold
}
```



The code `\markup \bold ...` is a shorthand for `\markup \override #'(font-series . bold) ...` – using the more verbose form, it is possible to obtain nuances such as semi-bold, if the text font has such variants. Refer to the documentation for the `font-series` properties (Sección "User backend properties" en *Referencia de Funcionamiento Interno*).

`\box` *arg* (markup)

Draw a box around *arg*.

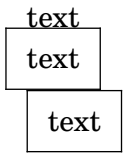
This function looks at the `thickness`, `box-padding`, and `font-size` properties to determine the line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box \line { V. S. }
}
```



Note that the box does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

```
\markup {
  \override #'(box-padding . 1.5)
  \column {
    "text"
    \box "text"
    \left-align \box "text"
  }
}
```



Used properties:

- `box-padding` (`0.2`)
- `font-size` (`0`)
- `thickness` (`1`)

`\caps` *arg* (markup)

Print *arg* in (fake) small caps.

This function is a copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

Use `\fontCaps` for real small caps (if the font provides it).

`\dynamic` *arg* (markup)

Print *arg* using the (music) font for dynamics.

This font only contains letters **f**, **m**, **n**, **p**, **r**, **s**, and **z**. When producing phrases like 'più **f**', the normal words (like 'più') should be done in a different font. The recommended font for this is bold and italic.

```
\markup {
  \dynamic {
    sfzp
  }
}
```

sfzp

`\figured-bass` *arg* (markup)

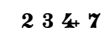
Set *arg* as small numbers for figured bass.

Specially slashed digits can be achieved with a trailing backslash (for numbers 6, 7, and 9) or a trailing plus (for numbers 2, 4, and 5).[1]

[1] Internally, this works by activating the 'dlig' OpenType feature of the Emmentaler font.

The use of a backslash is in analogy to \figuremode (véase Sección 15.3.2 [Escritura del bajo cifrado], página 529). Note that to get a backslash character in markup it must be escaped by doubling it. Additionally, it must be put into double quotes.

```
\markup {
  \figured-bass {
    2 3 4+ 7 "9\\"
  }
}
```



\finger *arg* (markup)
: Set *arg* as small numbers for fingering instructions.

```
\markup {
  \finger {
    1 2 3 4 5
  }
}
```



\fontCaps *arg* (markup)
: Print *arg* in small caps.

  This command sets the font-variant property to small-caps.

  Unlike \smallCaps, which merely uses capital letters at a smaller font size, this uses the actual variant of the font for small caps. (As a consequence, if you configure a custom text font, this command has no effect if that font does not have a small caps variant.)

```
\markup \fontCaps "Small caps"
```



\fontsize *increment* (number) *arg* (markup)
: Increase current font size by *increment* to print *arg*.

  This function adjusts the baseline-skip and word-space properties accordingly.

```
\markup {
  default
  \hspace #2
  \fontsize #-1.5 smaller
}
```



  Used properties:

  - baseline-skip (2)
  - word-space (1)
  - font-size (0)

\huge *arg* (markup)
: Set font size to value 2 to print *arg*.

```
\markup {
  default
  \hspace #2
```

```
  \huge huge
}
```

default **huge**

`\italic` *arg* (markup)
: Print *arg* in italics.

  This command sets the `font-shape` property to `italic`.

```
\markup {
  default
  \hspace #2
  \italic italic
}
```

default *italic*

`\large` *arg* (markup)
: Set font size to value 1 to print *arg*.

```
\markup {
  default
  \hspace #2
  \large large
}
```

default large

`\larger` *arg* (markup)
: Increase current font size by 1 to print *arg*.

  This function adjusts the `baseline-skip` and `word-space` properties accordingly.

```
\markup {
  default
  \hspace #2
  \larger larger
}
```

default larger

`\magnify` *sz* (number) *arg* (markup)
: Magnify current font by factor *sz* to print *arg*.

```
\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}
```

default 50% larger

  Note that magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

\normal-size-sub *arg* (markup)
Set *arg* in subscript with a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

default $_{\text{subscript in standard size}}$

Used properties:

- font-size (0)

\normal-size-super *arg* (markup)
Set *arg* in superscript with a normal font size.

```
\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}
```

default $^{\text{superscript in standard size}}$

Used properties:

- font-size (0)

\normal-text *arg* (markup)
Print *arg* with default text font.

This resets all font-related properties (except the size), no matter what font was used earlier.

```
\markup {
  \huge \bold \sans \fontCaps {
    huge bold sans caps
    \hspace #2
    \normal-text {
      huge normal
    }
    \hspace #2
    as before
  }
}
```

HUGE BOLD SANS CAPS huge normal **AS BEFORE**

\normal-weight *arg* (markup)
Switch to normal weight (in contrast to bold) to print *arg*.

This command sets the font-series property to normal.

```
\markup {
  \bold {
    some bold text
```

```
    \hspace #2
    \normal-weight {
      normal font series
    }
    \hspace #2
    bold again
  }
}
```

some bold text normal font series **bold again**

\normalsize *arg* (markup)
: Set font size to default (i.e., to value 0) to print *arg*.

```
\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}
```

this is very small normal size teeny again

\number *arg* (markup)
: Print *arg* using the (music) font for numbers.

  This font also contains symbols for figured bass, some punctuation, spaces of various widths, some letters and text variants of accidentals. Use \dynamic to access the (very small number of) letters. For accidentals you might use \number in combination with Unicode characters to access the text representation forms of accidental glyphs, as the following table shows.

Unicode value	Unicode character
U+266D	♭
U+266E	♮
U+266F	♯
U+1D12A	𝄪
U+1D12B	𝄫

Examples:

```
\number ♭ → ♭
\number { \char ##x266F } → ♯
```

To get accidentals protected against overrides of `font-name` it is preferable to use `\text-doubleflat`, `\text-flat`, `\text-natural`, `\text-sharp`, `\text-doublesharp` or the general `\text-accidental` for the text variants of accidentals.

The appearance of digits in the Emmentaler font can be controlled with four OpenType features: 'tnum', 'cv47', 'ss01', and 'kern', which can be arbitrarily combined.

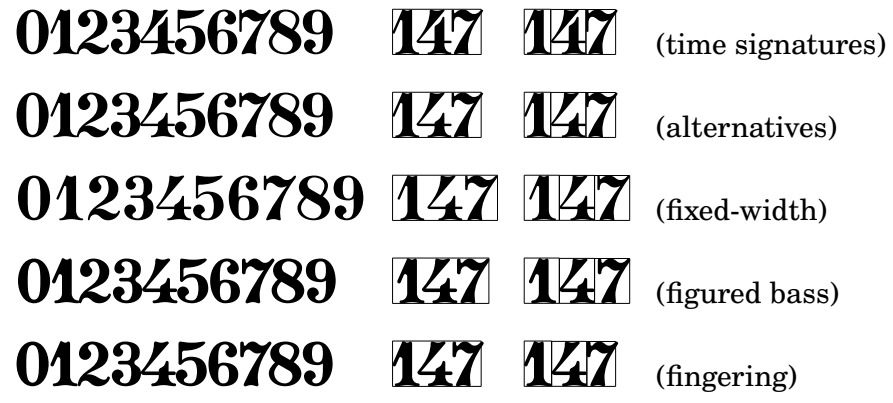
tnum — If off (which is the default), glyphs 'zero' to 'nine' have no left and right side bearings. If on, the glyphs all have the same advance width by making the bearings non-zero.

cv47 — If on, glyphs 'four' and 'seven' have shorter vertical strokes. Default is off.

ss01 — If on, glyphs 'zero' to 'nine' have a fatter design, making them more readable at small sizes. Default is off.

kern — If on (which is the default), provide pairwise kerning between (most) glyphs.

```
\markuplist
  \number
  \fontsize #4.5
  \override #'((padding . 2)
               (baseline-skip . 4)
               (box-padding . 0)
               (thickness . 0.1))
  \table #'(-1 -1 -1 -1) {
      0123456789 \box 147 \concat { \box 1 \box 4 \box 7 }
    \normal-text \normalsize "(time signatures)"
    \override #'(font-features .("cv47")) {
      0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
    \normal-text \normalsize "(alternatives)"
    \override #'(font-features .("tnum" "cv47" "-kern")) {
      0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
    \normal-text \normalsize "(fixed-width)"
    \override #'(font-features . ("tnum" "cv47" "ss01")) {
      0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
    \normal-text \normalsize "(figured bass)"
    \override #'(font-features . ("cv47" "ss01")) {
      0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
    \normal-text \normalsize "(fingering)"
  }
```



See also the markup commands `\figured-bass` and `\finger`, which set the font features accordingly.

`\overtie` *arg* (markup)
Print a tie over *arg*.

```
\markup \line {
  \overtie "overtied"
  \override #'((offset . 5) (thickness . 1))
  \overtie "overtied"
  \override #'((offset . 1) (thickness . 5))
  \overtie "overtied"
}
```



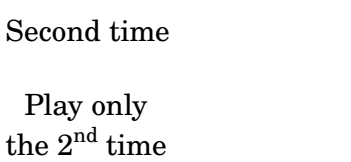
Used properties:

- `shorten-pair` (`(0 . 0)`)
- `height-limit` (`0.7`)
- `direction` (`1`)
- `offset` (`2`)
- `thickness` (`1`)

`\replace` *replacements* (list) *arg* (markup)
Use *replacements* to replace strings in *arg*.

Each (*key* . *value*) pair of the *replacements* alist specifies what should be replaced; *key* gets replaced by *value*. Note the quasi-quoting syntax with a backquote in the second example.

```
\markup \replace #'(("2nd" . "Second"))
  "2nd time"
\markup \replace
  #`(("2nd" . ,#{ \markup \concat { 2 \super nd } #}))
  \center-column {
    \line { Play only }
    \line { the 2nd time }
  }
```



Used properties:

- `replacement-alist`

`\sans` *arg* (markup)
Print *arg* with a sans-serif font.

This command sets the `font-family` property to `sans`.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
```

```
  }
}
```

default sans serif

\serif *arg* (markup)
Print *arg* with a serif font.

This command sets the `font-family` property to `serif`.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \serif {
      text in serif font
    }
    \hspace #2
    return to sans
  }
}
```

sans serif, bold text in serif font return to sans

\simple *str* (string)
Print string *str*.

`\markup \simple "x"` is equivalent to `\markup "x"`. This command was previously used internally, but no longer is, and is being kept for backward compatibility only.

\small *arg* (markup)
Set font size to value -1 to print *arg*.

```
\markup {
  default
  \hspace #2
  \small small
}
```

default small

\smallCaps *arg* (markup)
Print *arg* in (fake) small caps.

Unlike `\fontCaps`, which uses the actual small caps variant of the current font, this fakes small caps by using capital letters at a smaller font size. It can thus be used for fonts that don't have a small caps variant.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

\smaller *arg* (markup)
Decrease current font size by 1 to print *arg*.

This function adjusts the `baseline-skip` and `word-space` properties accordingly.

```
\markup {
  \fontsize #3.5 {
    large text
    \hspace #2
    \smaller { smaller text }
    \hspace #2
    large text
  }
}
```

large text smaller text large text

\sub *arg* (markup)

Set *arg* in subscript.

```
\markup { \concat { H \sub 2 O } }
```

H_2O

See also \super.

Used properties:

- `font-size` (0)

\super *arg* (markup)

Set *arg* in superscript.

```
\markup { E = \concat { mc \super 2 } }
```

$E = mc^2$

See also \sub.

Used properties:

- `font-size` (0)

\teeny *arg* (markup)

Set font size to value -3 to print *arg*.

```
\markup {
  default
  \hspace #2
  \teeny teeny
}
```

default teeny

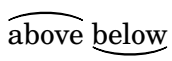
\tie *arg* (markup)

Add a horizontal bow at the bottom or top of *arg*.

This function uses `make-tie-stencil` to create the bow; it looks at the `thickness` and `offset` properties to determine the line thickness and vertical offset, respectively. The added bow fits the extent of *arg*; use the `shorten-pair` property to modify this. The `direction` property may be set explicitly using `override` or direction modifiers, or implicitly by using `voiceOne`, etc.

```
\markup {
  \override #'(direction . 1)
  \tie "above"
```

```
  \override #'(direction . -1)
  \tie "below"
}
```



See also \undertie and \overtie, which are shorthands for this function.

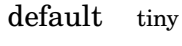
Used properties:

- shorten-pair ((0 . 0))
- height-limit (0.7)
- direction (1)
- offset (2)
- thickness (1)

\tiny *arg* (markup)

Set font size to value -2 to print *arg*.

```
\markup {
  default
  \hspace #2
  \tiny tiny
}
```

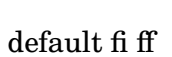


\typewriter *arg* (markup)

Print *arg* in typewriter.

This command sets the font-family property to typewriter, also switching off the 'liga' OpenType feature to disable ligatures like 'ff' or 'fi'.

```
\markup {
  "default fi ff"
  \hspace #2
  \typewriter "typewriter fi ff"
}
```



\underline *arg* (markup)

Underline *arg*.

This function looks at the property thickness to determine the line thickness, at offset to determine the line's vertical offset from *arg*, and at underline-skip to determine the distance of additional lines from the others.

The underline-shift property is used to make subsequent calls work correctly. Overriding it makes little sense since it would end up adding the provided value to the one of offset.

```
\markup \justify-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
```

```
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(underline-skip . 4)
  \underline \underline \underline "underlined thrice"
}
```

underlined underlined underlined underlined thrice

Used properties:

- `underline-skip` (2)
- `underline-shift` (0)
- `offset` (2)
- `thickness` (1)

`\undertie` *arg* (markup)
: Print a tie under *arg*.

```
\markup \line {
  \undertie "undertied"
  \override #'((offset . 5) (thickness . 1))
  \undertie "undertied"
  \override #'((offset . 1) (thickness . 5))
  \undertie "undertied"
}
```

undertied undertied undertied

Used properties:

- `shorten-pair` (`(0 . 0)`)
- `height-limit` (0.7)
- `direction` (1)
- `offset` (2)
- `thickness` (1)

`\upright` *arg* (markup)
: Print *arg* upright.

  This command is the opposite of `\italic`; it sets the `font-shape` property to `upright`.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}
```

italic text upright text *italic again*

`\volta-number` *arg* (markup)
Set *arg* in a font appropriate for volta numbers.

```
\markup \volta-number "4."
```

4.

`\with-string-transformer` *transformer* (procedure) *arg* (markup)
Apply string transformer function *transformer* to *arg*.

Whenever a string is interpreted inside *arg*, function *transformer* is called first, and its result is then interpreted. The arguments passed to *transformer* are the output definition, the property alist chain, and the markup *arg*. See Sección "New markup command definition" en *Extender* about the two first arguments.

```
\markup \with-string-transformer
  #(lambda (layout props str)
     (string-upcase str))
  \concat { "abc" \larger "def" }
```

ABCDEF

A.1.2 Markup for text alignment

`\abs-hspace` *amount* (number)
Create an invisible object taking up absolute horizontal space of *amount* points.

```
\markup {
  one
  \abs-hspace #20
  two
  \abs-hspace #40
  three
}
```

one two three

See also `\hspace`.

`\abs-vspace` *amount* (number)
Create an invisible object taking up absolute vertical space of *amount* points.

```
\markup {
    \center-column {
    one
    \abs-vspace #20
    two
    \abs-vspace #40
    three
  }
}
```

one

two

three

See also \vspace.

\align-on-other *axis* (non-negative, exact integer) *other-dir* (boolean-or-number) *other* (markup) *self-dir* (boolean-or-number) *self* (markup)

Align markup *self* on markup *other* along *axis*.

This function uses *self-dir* and *other-dir* for mutual alignment of *self* and *other*, respectively, translating *self* as requested relative to its surroundings. *other* is not printed.

If *self-dir* or *other-dir* is #f, use the reference point of *self* or *other*, respectively.

```
\markup \column {
  12
  \align-on-other #X #RIGHT 12
                     #LEFT 12345
  \align-on-other #X #RIGHT 123
                     #LEFT \fermata
  123
  \align-on-other #X #RIGHT 123
                     ##f \fermata
}
```



\center-align *arg* (markup)

Align *arg* to its X center.

```
\markup {
  \column {
    one
    \center-align two
    three
  }
}
```

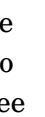


\center-column *args* (markup list)

Put *args* into a centered column.

See also \column.

```
\markup {
  \center-column {
    one
    two
    three
  }
}
```



Used properties:

- `baseline-skip`

`\column` *args* (markup list)

Stack the markups in *args* vertically.

The property `baseline-skip` determines the space between markups in *args* (to be more precise, the space between the baselines of the markups).

```
\markup {
  \column {
    one
    two
    three
  }
}
```

one
two
three

The baseline of the output of `\column` is the baseline of its first line.

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)

Print *arg1*, then print *arg2* on top of it.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; for this purpose use `\overlay` instead.

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
}
```



`\concat` *args* (markup list)

Concatenate *args* in a horizontal line, without spaces in between.

Strings are concatenated on the input level, allowing ligatures. For example, `\concat { "f" "i" }` is equivalent to `"fi"`.

```
\markup {
  \concat {
    one two three
  }
}
```

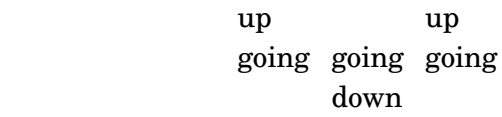
onetwothree

`\dir-column` *args* (markup list)

Make a column of *args*.

Depending on the setting of the `direction` layout property, the arguments are stacked upwards or downwards.

```
\markup {
  \override #`(direction . ,UP)
  \dir-column {
    going up
  }
  \hspace #1
  \dir-column {
    going down
  }
  \hspace #1
  \override #'(direction . 1)
  \dir-column {
    going up
  }
}
```



The baseline of the output of `\dir-column` is the baseline of its first line.

Used properties:

- `baseline-skip`
- `direction`

`\fill-line` *args* (markup list)

Put markups *args* into a horizontal line at fixed positions.

If there is a single argument, it gets centered. If there are two arguments, they get aligned to the left and right, respectively. Otherwise, if there are *n* arguments, the markups are aligned to *n* equally spaced positions, with the first markup left-aligned, the last markup right-aligned, and the remaining markups centered at the respective positions. If there are no arguments, return an empty stencil.

The width of the horizontal line can be modified by overriding the `line-width` property. The space between arguments is at least as large as the value of the `word-space` property, at the cost of lengthening the line if necessary.

```
\markup {
  \column {
    \fill-line { Words positioned evenly across a line }
    \fill-line { | | | | | | }
    \null
    \fill-line {
      \line { Text markups }
      \line { \italic { positioned evenly } }
      \line { across a line }
    }
    \null
    \override #'(line-width . 50)
    \fill-line { Width explicitly specified }
  }
}
```

Words positioned evenly across a line
| | | | | |

Text markups *positioned evenly* across a line

Width explicitly specified

See also `\justify-line`.

Used properties:

- `line-width` (`#f`)
- `word-space` (`0.6`)
- `text-direction` (`1`)

`\fill-with-pattern` *space* (number) *dir* (direction) *pattern* (markup) *left* (markup) *right* (markup)

Put *left* and *right* at the start and end of a line, respectively, and fill the space in between with repeated *pattern* markups.

Patterns are spaced apart by *space* and aligned to direction *dir*. The width of the line is given by the `line-width` property.

```
\markup \column {
  "right-aligned:"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned:"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned:"
  \override #'(line-width . 50) {
    \fill-with-pattern #2 #LEFT : left first
    \fill-with-pattern #2 #LEFT : left second
  }
}
```

right-aligned:
first . . . . . right
second . . . . . right

center-aligned:
left - right

left-aligned:
left : : : : : : : : : : : : : : : : first
left : : : : : : : : : : : : : : : : second

Used properties:

- `line-width`
- `word-space`

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)

Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT two
    three
    \null
    one
    \general-align #X #CENTER two
    three
    \null
    \line {
      one
      \general-align #Y #UP two
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2 two
      three
    }
  }
}
```

one
two
three

one
two
three

one two three

one three
two

\halign *dir* (number) *arg* (markup)

Print *arg* with horizontal alignment set to *dir*.

If *dir* is -1, *arg* is left-aligned, while +1 makes it right-aligned. Values in between interpolate the alignment accordingly.

```
\markup {
  \column {
    one
    \halign #LEFT two
    three
    \null
    one
    \halign #CENTER two
    three
    \null
```

```
    one
    \halign #RIGHT two
    three
    \null
    one
    \halign #-5 two
    three
  }
}
```

one
two
three

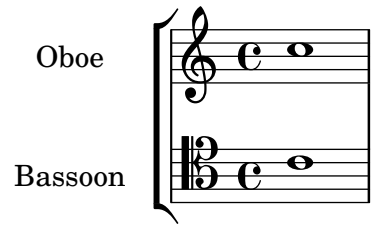
one
two
three

one
two
three

one
two
three

\hcenter-in *length* (number) *arg* (markup)
Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12 Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12 Bassoon
    }
    \clef tenor
    c'1
  }
>>
```



`\hspace` *amount* (number)

Create an invisible object taking up *amount* horizontal space.

```
\markup {
  one
  \hspace #2
  two
  \hspace #8
  three
}
```



amount can be also a negative value, which can be best visualized as if the current drawing point gets moved to the left.

```
\markup \concat {
  \hspace #4
  \column {
    \box \concat { AAAA \hspace #4 }
    \box \concat { AAAA \hspace #-4 }
    \box \concat { \hspace #4 AAAA }
    \box \concat { \hspace #-4 AAAA }
  }
}
```



See also `\abs-hspace`.

`\justify` *args* (markup list)

Print *args* as lines aligned both at the left and the right.

Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; X is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua.  Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

The baseline of the output of `\justify` is the baseline of its first line.

Used properties:

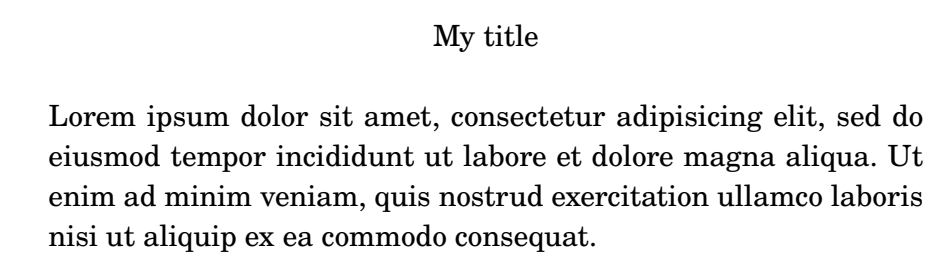
- `text-direction` (1)

- word-space
- line-width (#f)
- baseline-skip

\justify-field *symbol* (symbol)

Justify the data that has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt
    ut labore et dolore magna aliqua.  Ut enim ad minim
    veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:myText
    }
  }
}

\markup {
  \null
}
```



\justify-line *args* (markup list)

Put markups *args* into a horizontal line, equally spaced.

If there is a single argument, it gets centered. If there are two arguments, they get aligned to the left and right, respectively. Otherwise, the markups are spread to fill the entire line, separated by equally large spaces. If there are no arguments, return an empty stencil.

The width of the horizontal line can be modified by overriding the line-width property. The space between arguments is at least as large as the value of the word-space property, at the cost of lengthening the line if necessary.

```
\markup {
  \justify-line {
    Constant space between neighboring words
  }
}
```

Constant space between neighboring words

See also `\fill-line`.

Used properties:

- `line-width` (`#f`)
- `word-space` (`0.6`)
- `text-direction` (`1`)

`\justify-string` *arg* (string)

Print string *arg* as lines aligned both at the left and the right.

Paragraphs are indicated by double newlines. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \override #'(line-width . 40)
  \justify-string "Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut
      labore et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation
      ullamco laboris nisi ut aliquip ex ea commodo
      consequat.


      Excepteur sint occaecat cupidatat non proident, sunt
      in culpa qui officia deserunt mollit anim id est
      laborum"
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

The baseline of the output of `\justify-string` is the baseline of its first line.

Used properties:

- `text-direction` (`1`)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
```

```
\column {
  one
  \left-align two
  three
}
}
```

one
two
three

\left-column *args* (markup list)
: Put *args* into a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

\line *args* (markup list)
: Put *args* into a horizontal line.

The property `word-space` determines the space between markups in *args*. For right-to-left scripts like Hebrew, `text-direction` should be set to -1.

```
\markup
  \override #'(word-space . 3)
  \column {
    \line { "A B" "C D" "E F" }
    \override #'(text-direction . -1)
    \line { "A B" "C D" "E F" }
  }
```

A B C D E F
E F C D A B

Used properties:

- `text-direction` (1)
- `word-space`

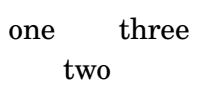
\lower *amount* (number) *arg* (markup)
: Lower *arg* by the distance *amount*.

A negative *amount* indicates raising; see also `\raise`.

The argument to `\lower` is the vertical displacement amount, measured in (global) staff spaces, which is independent of the markup's current font size. If you need vertical movement that takes the font size into account, use `\translate-scaled` instead.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as `padding`, `Y-offset`, or `extra-offset`, or spacing variables such as `markup-system-spacing`.

```
\markup {
  one
  \lower #3 two
  three
}
```



\overlay *args* (markup list)

Take a list of markups *args* and combine them.

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \overlay {
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
    \translate #'(0 . 4) \arrow-head #Y #UP ##f
  }
}
```

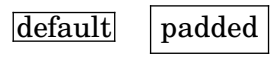


\pad-around *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

Identical to function \pad-markup.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}
```



\pad-markup *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

Identical to function \pad-around.

```
\markup {
  \box {
    default
```

```
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
    }
  }
}
```

default padded

\pad-to-box *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)
Make *arg* take at least *x-ext*, *y-ext* space.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}
```

default padded

\pad-x *amount* (number) *arg* (markup)
Add padding *amount* around *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x #2 {
      padded
    }
  }
}
```

default padded

\pad-x-left *amount* (number) *arg* (markup)
Add padding *amount* to the left of *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
```

```
    \pad-x-left #2 {
      padded
    }
  }
}
```

default padded

\pad-x-right *amount* (number) *arg* (markup)
: Add padding *amount* to the right of *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x-right #2 {
      padded
    }
  }
}
```

default padded

\put-adjacent *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)
: Put *arg2* next to *arg1* along *axis* to the *dir* side, without moving *arg1*.

```
\markup \column {
  text
  \put-adjacent #X #LEFT text *
  text
}
```

text
*text
text

\raise *amount* (number) *arg* (markup)
: Raise *arg* by the distance *amount*.

  A negative *amount* indicates lowering, see also \lower.

  The argument to \raise is the vertical displacement amount, measured in (global) staff spaces, which is independent of the markup's current font size. If you need vertical movement that takes the font size into account, use \translate-scaled instead.

  This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as padding, Y-offset, or extra-offset, or spacing variables such as markup-system-spacing.

```
\markup { C \small \bold \raise #1.0 9/7+ }
```

C 9/7+

\right-align *arg* (markup)
Align *arg* on its right edge.

```
\markup {
  \column {
    one
    \right-align two
    three
  }
}
```



\right-column *args* (markup list)
Put *args* into a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

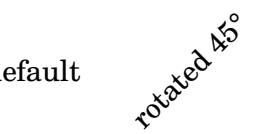


Used properties:

- `baseline-skip`

\rotate *ang* (number) *arg* (markup)
Rotate *arg* by *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45°
  }
}
```



\translate *offset* (pair of numbers) *arg* (markup)
Translate *arg* relative to its surroundings.

offset is a pair of numbers representing the displacement in the X and Y axes. See also `\translate-scaled`.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing

mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as `padding`, `X-offset`, `Y-offset` or `extra-offset`, or spacing variables such as `markup-system-spacing`.

```
\markup {
  *
  \translate #'(2 . 3)
    \line { translated two spaces right, three up }
}
```



`\translate-scaled` *offset* (pair of numbers) *arg* (markup)

Translate *arg* by *offset*, scaling the offset by the font size.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as `padding`, `X-offset`, `Y-offset` or `extra-offset`, or spacing variables such as `markup-system-spacing`.

See also `\translate`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}
```



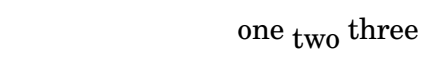
Used properties:

- `font-size` (0)

`\vcenter` *arg* (markup)

Align *arg* to its Y center.

```
\markup {
  one
  \vcenter two
  three
}
```



`\vspace` *amount* (number)

Create an invisible object taking up vertical space of *amount* multiplied by 3.

```
\markup {
    \center-column {
    one
    \vspace #1
    two
    \vspace #3
```

```
    three
  }
}
```

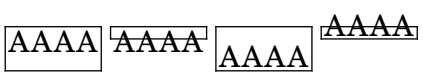
one

two

three

amount can be also a negative value, which can be best visualized as if the current drawing point gets moved up.

```
\markup {
  \vspace #1
  \box \column { AAAA \vspace #0.4 }
  \box \column { AAAA \vspace #-0.4 }
  \box \column { \vspace #0.4 AAAA }
  \box \column { \vspace #-0.4 AAAA }
}
```



See also `\abs-vspace`.

`\wordwrap` *args* (markup list)
Print *args* as left-aligned lines.

This function provides simple word-wrap. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua.  Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

The baseline of the output of `\wordwrap` is the baseline of its first line.

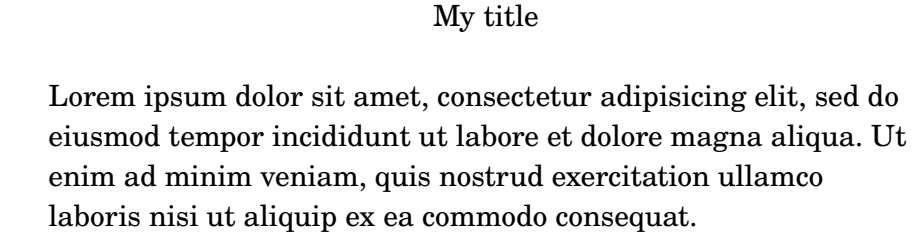
Used properties:

- `text-direction` (`1`)
- `word-space`
- `line-width` (`#f`)
- `baseline-skip`

`\wordwrap-field` *symbol* (symbol)

Word-wrap the data that has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt ut
    labore et dolore magna aliqua.  Ut enim ad minim
    veniam, quis nostrud exercitation ullamco laboris nisi
    ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:myText
    }
  }
}

\markup {
  \null
}
```



`\wordwrap-string` *arg* (string)

Print string *arg* as left-aligned lines.

Paragraphs are indicated by double newlines. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \override #'(line-width . 40)
  \wordwrap-string "Lorem ipsum dolor sit amet,
      consectetur adipisicing elit, sed do eiusmod tempor
      incididunt ut labore et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation
      ullamco laboris nisi ut aliquip ex ea commodo
      consequat.


      Excepteur sint occaecat cupidatat non proident,
      sunt in culpa qui officia deserunt mollit anim id
```

```
      est laborum"
}
```

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et
dolore magna aliqua.
Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo
consequat.
Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

The baseline of the output of `\wordwrap-string` is the baseline of its first line.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

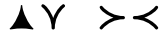
A.1.3 Graphical markup

`\arrow-head` *axis* (integer) *dir* (direction) *filled* (boolean)

Print an arrow head along *axis* in direction *dir*.

Fill the head if *filled* is set to `#t`.

```
\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}
```



`\beam` *width* (number) *slope* (number) *thickness* (number)

Draw a beam with given *width*, *slope*, and *thickness*.

```
\markup {
  \beam #5 #1 #2
}
```



\bracket *arg* (markup)
Draw vertical brackets around *arg*.

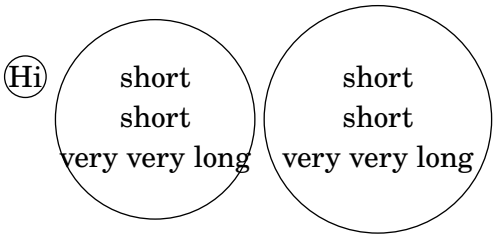
```
\markup {
  \bracket {
    \note {2.} #UP
  }
}
```



\circle *arg* (markup)
Draw a circle around *arg*.

Use properties `thickness`, `circle-padding`, and `font-size` to set the line thickness and padding around the markup. If `bbox` is set to `#t`, make the circle enclose the bounding box of *arg*, otherwise use either the width or the height of *arg* (whatever is larger) as the diameter.

```
\markup {
  \circle {
    Hi
  }
  \circle {
    \center-column { "short" "short" "very very long" }
  }
  \override #'(bbox . #t) \circle {
    \center-column { "short" "short" "very very long" }
  }
}
```



Note that the circle does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

Used properties:

- `bbox` (`#f`)
- `circle-padding` (`0.2`)
- `font-size` (`0`)
- `thickness` (`1`)

\draw-circle *radius* (number) *thickness* (number) *filled* (boolean)
Draw a circle with given *radius* and *thickness*.

Fill the circle if *filled* is set to `#t`.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
```

```
}
```



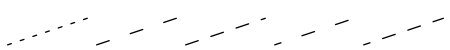
`\draw-dashed-line` *dest* (pair of numbers)

Draw a dashed line along vector *dest*.

Properties `on` and `off` give the length of a dash and the space between two dashes, respectively; `phase` shortens the first dash by the given amount.

If the `full-length` property is set to `#t` (which is the default), the value of property `off` (and `on` under some boundary conditions) gets adjusted so that there is neither whitespace at the end of the line nor the last dash truncated.

```
\markup {
  \override #'((on . 0.3) (off . 0.5))
  \draw-dashed-line #'(6 . 2)

  \draw-dashed-line #'(6 . 2)

  \override #'(full-length . #f)
  \draw-dashed-line #'(6 . 2)

  \override #'(phase . 0.5)
  \draw-dashed-line #'(6 . 2)

  \override #'((full-length . #f) (phase . 0.5))
  \draw-dashed-line #'(6 . 2)
}
```



Used properties:

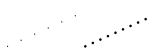
- `full-length` (`#t`)
- `phase` (`0`)
- `off` (`1`)
- `on` (`1`)
- `thickness` (`1`)

`\draw-dotted-line` *dest* (pair of numbers)

Draw a dotted line along vector *dest*.

Property `off` gives the space between two dots; its value gets adjusted so that the first and last dot exactly start and end the line, respectively. `phase` shifts all dots along the vector by the given amount.

```
\markup {
  \draw-dotted-line #'(5.1 . 2.3)
  \override #'((thickness . 2) (off . 0.2))
  \draw-dotted-line #'(5.1 . 2.3)
}
```



Used properties:

- `phase` (`0`)

- `off` (1)
- `thickness` (1)

`\draw-hline`
: Draw a horizontal line.

  The property `span-factor` sets the length of the line as a multiple of the `line-width` property.

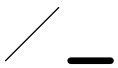
```
\markup {
  \column {
    \draw-hline
    \override #'(span-factor . 1/3)
    \draw-hline
  }
}
```

  Used properties:

  - `span-factor` (1)
  - `line-width`
  - `thickness` (1)

`\draw-line` *dest* (pair of numbers)
: Draw a line along vector *dest*.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



  Used properties:

  - `thickness` (1)

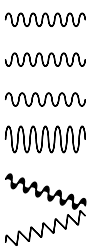
`\draw-squiggle-line` *sq-length* (number) *dest* (pair of numbers) *eq-end?* (boolean)
: Draw a squiggled line along vector *dest*.

  sq-length is the length of the first bow; this value gets always adjusted so that an integer number of squiggles is printed. If *eq-end?* is set to `#t`, the squiggled line ends with a bow in the same direction as the starting one.

  The appearance of the squiggled line may be customized by overriding the `thickness`, `angularity`, `height`, and `orientation` properties.

```
\markup
  \column {
    \draw-squiggle-line #0.5 #'(6 . 0) ##t
    \override #'(orientation . -1)
    \draw-squiggle-line #0.5 #'(6 . 0) ##t
    \draw-squiggle-line #0.5 #'(6 . 0) ##f
    \override #'(height . 1)
    \draw-squiggle-line #0.5 #'(6 . 0) ##t
    \override #'(thickness . 5)
```

```
  \draw-squiggle-line #0.5 #'(6 . -2) ##t
  \override #'(angularity . 2)
  \draw-squiggle-line #0.5 #'(6 . 2) ##t
}
```



Used properties:

- `orientation` (`1`)
- `height` (`0.5`)
- `angularity` (`0`)
- `thickness` (`0.5`)

`\ellipse` *arg* (markup)

Draw an ellipse around *arg*.

Use properties `thickness`, `x-padding`, `y-padding`, and `font-size` to set the line thickness and padding around the markup.

This is the same as function `\oval` but with different padding defaults.

```
\markup {
  \ellipse {
    Hi
  }
}
```



Note that the ellipse does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

Used properties:

- `y-padding` (`0.2`)
- `x-padding` (`0.2`)
- `font-size` (`0`)
- `thickness` (`1`)

`\epsfile` *axis* (number) *size* (number) *file-name* (string)

Inline an image *file-name*, scaled along *axis* to *size*.

See `\image` for details on this command; calling

```
\markup \epsfile axis size file-name
```

is the same as

```
\markup
  \override #'(background-color . #f)
  \image axis size file-name
```

`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)

Draw a box of dimensions *xext* and *yext*, with rounded corners given by *blot*.

For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

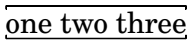
creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

```
\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
  \combine
    \filled-box #'(1 . 8) #'(0 . 7) #0.2
    \with-color #white
      \filled-box #'(3.6 . 5.6) #'(3.5 . 5.5) #0.7
}
```



\hbracket *arg* (markup)
Draw horizontal brackets around *arg*.

```
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
```



\image *axis* (number) *size* (number) *file-name* (string)
Inline an image *file-name*, scaled along *axis* to *size*.

The image format is determined based on the extension of *file-name*, which should be `.png` for a PNG image, or `.eps` for an EPS image (`.PNG` and `.EPS` are allowed as well).

EPS files are only supported in the PostScript backend – for all output formats –, and in the Cairo backend – when creating PostScript or EPS output.

If the image has transparency, it will appear over a colored background with the color specified by the `background-color` property, which defaults to `"white"`.

To disable the colored background, set `background-color` to `#f`. For EPS images, this always works (where EPS images work in the first place). On the other hand, for PNG images, it works in the Cairo backend (which can output all supported formats), as well as in the SVG backend, but *not* in the PostScript backend, which is the default. See Sección “Advanced command-line options for LilyPond” en *Utilización del Programa* for how to activate the Cairo backend.

Use `\withRelativeDir` as a prefix to *file-name* if the file should be found relative to the input file.

Used properties:

- `background-color` (`"white"`)

`\oval` *arg* (markup)

Draw an oval around *arg*.

Use properties `thickness`, `x-padding`, `y-padding`, and `font-size` to set the line thickness and padding around the markup.

This is the same as function `\ellipse` but with different padding defaults.

```
\markup {
  \oval {
    Hi
  }
}
```



Note that the oval does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

Used properties:

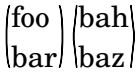
- `y-padding` (`0.75`)
- `x-padding` (`0.75`)
- `font-size` (`0`)
- `thickness` (`1`)

`\parenthesize` *arg* (markup)

Draw parentheses around *arg*.

This is useful for parenthesizing a column containing several lines of text.

```
\markup {
  \parenthesize
  \column {
    foo
    bar
  }
  \override #'(angularity . 2)
  \parenthesize
  \column {
    bah
    baz
  }
}
```



Used properties:

- `width` (`0.25`)
- `line-thickness` (`0.1`)
- `thickness` (`1`)
- `size` (`1`)
- `padding`
- `angularity` (`0`)

\path *thickness* (number) *commands* (list)

Draw a path with line *thickness* according to the directions given in *commands*.

commands is a list of lists where the `car` of each sublist is a drawing command and the `cdr` comprises the associated arguments for each command.

There are seven commands available to use in *commands*: `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto`, and `closepath`. Note that the commands that begin with 'r' are the relative variants of the other three commands. You may also use the standard SVG single-letter equivalents: `moveto` = `M`, `lineto` = `L`, `curveto` = `C`, `closepath` = `Z`. The relative commands are written lowercase: `rmoveto` = `r`, `rlineto` = `l`, `rcurveto` = `c`.

The commands `moveto`, `rmoveto`, `lineto`, and `rlineto` take 2 arguments, namely the X and Y coordinates of the destination point.

The commands `curveto` and `rcurveto` create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The `closepath` command takes zero arguments and closes the current subpath in the active path.

Line-cap styles and line-join styles may be customized by overriding the `line-cap-style` and `line-join-style` properties, respectively. Available line-cap styles are `butt`, `round`, and `square`. Available line-join styles are `miter`, `round`, and `bevel`.

The property `filled` specifies whether or not the path is filled with color.

```
samplePath =
  #'((lineto -1 1)
     (lineto 1 1)
     (lineto 1 -1)
     (curveto -5 -5 -5 5 -1 0)
     (closepath))

\markup \scale #'(2 . 2) {
  \path #0.25 #samplePath

  \override #'(line-join-style . miter)
  \path #0.25 #samplePath

  \override #'(filled . #t)
  \path #0.25 #samplePath
}
```



Used properties:

- `filled` (`#f`)
- `line-join-style` (`round`)
- `line-cap-style` (`round`)

\polygon *points* (list of number pairs)

A polygon delimited by the list of *points*.

Property `extroversion` defines how the shape of the polygon is adapted to its thickness: if it is 0, the polygon is traced as-is. If it is -1, the outer side of the line is just on the given points. If set to 1, the line has its inner side on the points. The `thickness` property controls the thickness of the line; for filled polygons, this means the diameter of the blot.

```
regularPentagon =
  #'((1 . 0) (0.31 . 0.95) (-0.81 . 0.59)
     (-0.81 . -0.59) (0.31 . -0.95))

\markup \scale #'(2 . 2) {
  \polygon #'((-1 . -1) (0 . -3) (2 . 2) (1 . 2))
  \override #'(filled . #f)
    \override #'(thickness . 2)
      \combine
        \with-color #(universal-color 'blue)
          \polygon #regularPentagon
        \with-color #(universal-color 'vermillion)
          \override #'(extroversion . 1)
            \polygon #regularPentagon
}
```



Used properties:

- `thickness` (1)
- `filled` (#t)
- `extroversion` (0)

`\postscript` *str* (string)

Insert *str* directly into the output as a PostScript command string.

This command is meant as a *last resort*. Almost all needs are better fulfilled by other markup commands (see, for example, `\path` and `\draw-line`). If you do use this command, keep the following points in mind:

- `\postscript` does not work in SVG output.
- Only a subset of the PostScript language is supported during the conversion from PostScript to PDF.
- There are no stability guarantees on the details of how LilyPond produces its own output (i.e., the context into which the PostScript code is inserted). They may change substantially across versions.
- LilyPond cannot understand the shape of the drawing, leading to suboptimal spacing. Usually, it is necessary to explicitly set up dimensions with a command like `\with-dimensions`.
- Depending on how you install LilyPond, the version of the PostScript interpreter (Ghostscript) can vary, and some of its features may be disabled.

str is processed with the following constraints.

- The string is embedded into the (intermediate) output file with the PostScript commands

```
gsave currentpoint translate 0.1 setlinewidth
```

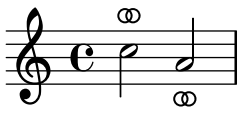
before and

```
grestore
```

after it.

- After these preceding commands (i.e., `currentpoint translate`) the origin of the current transformation is the reference point of `\postscript`. Scale and rotation of the current transformation reflect the global staff line distance and (if applied) other transformation markup commands (e.g., `\scale` and `\rotate`) encapsulating the `\postscript` command.
- The current point is set to the coordinate (0, 0).
- If an unwanted line appears at the beginning of your PostScript code, you are probably missing a call to `newpath`.

```
ringsps = "
  0.15 setlinewidth
  0.9 0.6 moveto
  0.4 0.6 0.5 0 361 arc
  stroke
  1.0 0.6 0.5 0 361 arc
  stroke
  "

rings = \markup {
  \with-dimensions #'(-0.2 . 1.6) #'(0 . 1.2)
  \postscript #ringsps
}

\relative c'' {
  c2^\rings
  a2_\rings
}
```

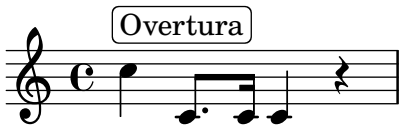


`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*.

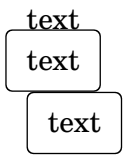
This function looks at properties `thickness`, `box-padding`, and `font-size` to determine the line thickness and padding around the *arg*. The `corner-radius` property defines the radius of the round corners (default value is 1).

```
c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r
```



Note that the box does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

```
\markup {
  \override #'(box-padding . 1.5)
  \column {
    "text"
    \rounded-box "text"
    \left-align \rounded-box "text"
  }
}
```



Used properties:

- `box-padding` (0.5)
- `font-size` (0)
- `corner-radius` (1)
- `thickness` (1)

`\scale` *factor-pair* (pair of numbers) *arg* (markup)

Scale *arg*.

factor-pair is a pair of numbers representing the scaling factor of the X and Y axes. Negative values may be used to produce mirror images.

```
\markup {
  \line {
    \scale #'(2 . 1)
    stretched
    \scale #'(1 . -1)
    mirrored
  }
}
```



`\triangle` *filled* (boolean)

Draw a triangle.

Fill the triangle if *filled* is set to `#t`.

The appearance can be controlled with properties `extroversion`, `font-size`, and `thickness`.

```
\markup {
  \triangle ##t
  \triangle ##f
  \override #'(font-size . 5)
  \override #'(thickness . 5) {
    \override #'(extroversion . 1)
    \triangle ##f
    \override #'(extroversion . -1)
    \triangle ##f
```

```
    }
  }
```



Used properties:

- `thickness` (1)
- `font-size` (0)
- `extroversion` (0)

`\with-url` *url* (string) *arg* (markup)
Add a link to URL *url* around *arg*.

This only works in the PDF backend.[2]

```
\markup {
  \with-url "https://lilypond.org/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

A.1.4 Markup for music and musical symbols

`\accidental` *alteration* (an exact rational number)
Select an accidental glyph for *alteration*, given as a rational number.

Use `\text-accidental` instead if you need glyph representation forms that fit and align well with text.

```
\markup {
  text
  \tiny { \accidental #1/2 \accidental #-1/2 }
  text
  \tiny { \text-accidental #1/2 \text-accidental #-1/2 }
  text
}
```

text ♯𝄲 text ♯𝄲 text

Used properties:

- `alteration-glyph-name-alist`

`\bar-line` *strg* (string)
Print a bar line in markup.

The allowed characters for input string *strg* are ';|.:!S[]{}', having the same meaning as with the `\bar` command. The additional characters '{' and '}' denote left and right braces, respectively.

The output is vertically centered.

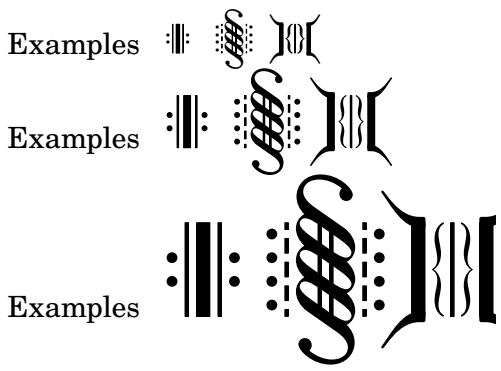
Changes of `font-size` are respected.

[2] Due to technical limitations the link doesn't work here in the Notation Reference.

The default of `height` is 4 staff-space units. Apart from the bracket tips of a bracket bar line and the segno bar line all other bar lines are scaled with `height`. We don't scale bracket tips and segno to meet the behavior of `SystemStartBracket` and the segno bar line.

`\bar-line` is further customizable by overriding `dot-count` and `dash-count` for dotted and dashed bar lines. The values for `hair-thickness`, `kern` and `thick-thickness` are customizable as well; defaults are the same as the values of the corresponding `BarLine` grob properties.

```
\markup {
   \override #'(word-space . 2)
   \column {
      \line {
        Examples
        \fontsize #-5 \translate-scaled #'(0 . 2) {
          \bar-line ":|.|:"
          \bar-line ";!S!;"
          \bar-line "]{|}["
        }
      }
      \line {
        Examples
        \fontsize #0 \translate-scaled #'(0 . 2) {
          \bar-line ":|.|:"
          \bar-line ";!S!;"
          \bar-line "]{|}["
        }
      }
      \line {
        Examples
        \fontsize #5 \translate-scaled #'(0 . 2) {
          \bar-line ":|.|:"
          \bar-line ";!S!;"
          \bar-line "]{|}["
        }
      }
   }
}
```



Used properties:

- `thick-thickness` (`6.0`)
- `kern` (`3.0`)

- hair-thickness (1.9)
- dash-count (5)
- dot-count (4)
- height (4)
- font-size (0)

\coda Draw a coda sign.

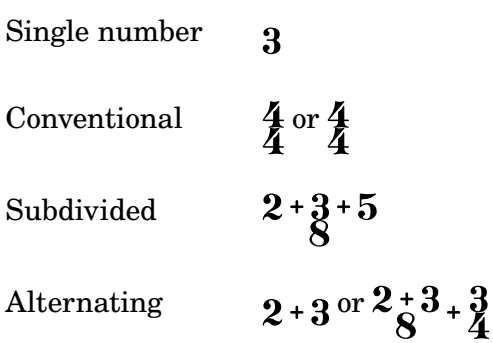
```
\markup {
  \coda
}
```


\compound-meter *time-sig* (number or pair)

Draw a numeric time signature based on *time-sig*.

time-sig can be a single number, a pair of numbers, a simple list, or a list of lists, as the following example demonstrates.

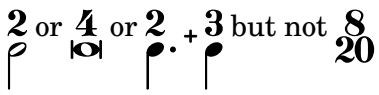
```
\markuplist {
  \override #'(baseline-skip . 4.5)
  \override #'(padding . 4.5)
  \table #'(-1 -1) {
    "Single number"  \compound-meter #3
    "Conventional"   \line {
                       \compound-meter #'(4 . 4) or
                       \compound-meter #'(4 4)
                     }
    "Subdivided"     \compound-meter #'(2 3 5 8)
    "Alternating"    \line {
                       \compound-meter #'((2) (3)) or
                       \compound-meter #'((2 3 8) (3 4))
                     }
  }
}
```



Setting the `denominator-style` property to `note` prints denominators as a note and dots when exact representation is possible. Example:

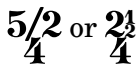
```
\markup {
  \override #'(denominator-style . note)
  \line {
    \compound-meter #'(2 2) or
    \compound-meter #'(4 1/2) or
```

```
    \compound-meter #'((2 8/3) (3 4)) but not
    \compound-meter #'(8 20)
  }
}
```



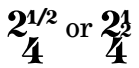
The nested-fraction-mixed property controls whether fractional parts are printed as mixed numbers or as common fractions. Example:

```
\markup {
  \override #'(nested-fraction-mixed . #f)
  \compound-meter #'(5/2 4) or
  \override #'(nested-fraction-mixed . #t)
  \compound-meter #'(5/2 4)
}
```



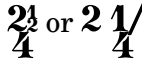
The nested-fraction-orientation property controls how nested fractions are arranged. Supported values are horizontal and vertical. Example:

```
\markup {
  \override #'(nested-fraction-orientation . horizontal)
  \compound-meter #'(5/2 4) or
  \override #'(nested-fraction-orientation . vertical)
  \compound-meter #'(5/2 4)
}
```



The nested-fraction-relative-font-size property controls the size of the numerals in nested fractions. Recommended values are -5.5 and 0. Using large numerals may take precedence over related properties. Example:

```
\markup {
  \override #'(nested-fraction-relative-font-size . -5.5)
  \compound-meter #'(5/2 4) or
  \override #'(nested-fraction-relative-font-size . 0)
  \compound-meter #'(5/2 4)
}
```



Used properties:

- note-staff-position (-2)
- note-head-style (())
- note-flag-style (())
- note-dots-direction (0)
- nested-fraction-relative-font-size (())
- nested-fraction-orientation (default)
- nested-fraction-mixed (#t)

- font-size (0)
- denominator-style (default)

\customTabClef *num-strings* (integer) *staff-space* (number)

Draw a clef in sans-serif style for a tablature with *num-strings* lines spaced by *staff-space*.

This markup command is used to implement \clef moderntab within a TabStaff context.

```
\markup {
  \customTabClef #4 #1
}
```



\doubleflat

Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



\doublesharp

Draw a double sharp symbol.

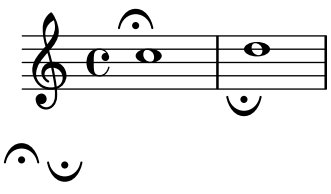
```
\markup {
  \doublesharp
}
```


\fermata Create a fermata glyph.

If property direction is DOWN, use an inverted glyph.

Note that within music, one would normally use the \fermata articulation instead of a markup.

```
{ c''1^\markup \fermata d''1_\markup \fermata }

\markup { \fermata \override #`(direction . ,DOWN) \fermata }
```



Used properties:

- direction (1)

\flat Draw a flat symbol.

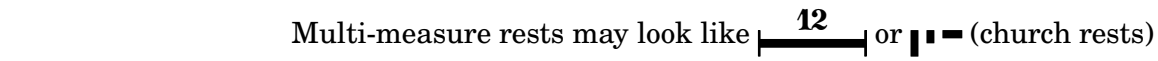
```
\markup {
  \flat
}
```



\multi-measure-rest-by-number *length* (non-negative, exact integer)
Return a multi-measure rest symbol for *length* measures.

If the number of measures is greater than the number given by `expand-limit` a thick horizontal line is printed. For every multi-measure rest lasting more than one measure a number is printed on top. However, if property `multi-measure-rest-number` is set to `#t`, this number gets suppressed.

```
\markup {
  Multi-measure rests may look like
  \multi-measure-rest-by-number #12
  or
  \override #'(multi-measure-rest-number . #f)
    \multi-measure-rest-by-number #7
  (church rests)
}
```



Used properties:

- `multi-measure-rest-number` (`#t`)
- `width` (`8`)
- `expand-limit` (`10`)
- `hair-thickness` (`2.0`)
- `thick-thickness` (`6.6`)
- `word-space`
- `style` (`()`)
- `font-size` (`0`)

\musicglyph *glyph-name* (string)
Print music symbol *glyph-name*.

See Sección “The Emmentaler font” en *Referencia de la Notación* for a complete listing of the possible glyph names.

```
\markup {
  \musicglyph "f"
  \musicglyph "rests.2"
  \musicglyph "clefs.G_change"
}
```



\natural Draw a natural symbol.

```
\markup {
  \natural
}
```



\note *duration* (duration) *dir* (number)
Draw a note with a given duration and a stem.

The note duration is specified with *duration* (setting both the note head type and the number of augmentation dots). The stem direction and length is given by *dir*.

This function is wrapper around `\note-by-number`; its documentation gives more details on the available properties.

```
\markup {
  \note {4..} #UP
  \hspace #2
  \override #'(style . cross)
    \note {4..} #0.75
  \hspace #2
  \note {\breve} #0
}
```



Used properties:

- `style` (`()`)
- `dots-direction` (`0`)
- `flag-style` (`()`)
- `font-size` (`0`)

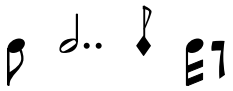
`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Draw a note with a given length, a number of dots, and a stem.

The note length is specified with *log*, the number of dots with *dot-count*, and the stem direction and length with *dir*. Fractional values for *dir* can be used to control the length of the stem. Value 0 suppresses the stem completely; this is useful for note head styles that don't take stems or come with built-in stems (like Kievan).

Ancient note head styles (via the `style` property, véase Sección B.9 [Estilos de cabezas de nota], página 927) get mensural-style flags by default; use the `flag-style` property to override this. Supported flag styles are `default`, `old-straight-flag`, `modern-straight-flag`, `flat-flag`, `stacked`, `mensural`, and `neomensural`. The last flag style is the same as `mensural` and provided for convenience.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
  \hspace #2
  \override #'(style . petrucci)
    \note-by-number #3 #0 #UP
  \hspace #2
  \override #'(flag-style . modern-straight-flag)
    \note-by-number #4 #0 #DOWN
  \override #'(style . kievan)
    \note-by-number #2 #0 #0
}
```



See also function `\note`.

Used properties:

- `style` (`()`)
- `dots-direction` (`0`)
- `flag-style` (`()`)
- `font-size` (`0`)

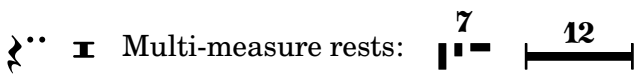
`\rest` *duration* (duration)

Return a rest symbol with length *duration*.

If the `multi-measure-rest` property is set to `#t`, a multi-measure rest symbol may be returned. In this case the duration needs to be entered as `{ 1*`*N*` }` to get a multi-measure rest for *N* bars. Actually, only the scaling factor (i.e., the number after '`*`') determines the length; the basic duration is disregarded.

See also functions `\rest-by-number` and `\multi-measure-rest-by-number` for more information on the used properties.

```
\markup {
  Rests:
  \hspace #2
  \rest { 4.. }
  \hspace #2
  \rest { \breve }
  \hspace #2
  Multi-measure rests:
  \override #'(multi-measure-rest . #t)
  {
    \hspace #2
    \rest { 1*7 }
    \hspace #2
    \rest { 1*12 }
  }
}
```



Used properties:

- `multi-measure-rest-number` (`#t`)
- `width` (`8`)
- `expand-limit` (`10`)
- `hair-thickness` (`2.0`)
- `thick-thickness` (`6.6`)
- `word-space`
- `style` (`()`)
- `font-size` (`0`)
- `style` (`()`)
- `ledgers` (`(-1 0 1)`)
- `font-size` (`0`)

`\rest-by-number` *log* (integer) *dot-count* (integer)

Draw a rest of length *log*, with *dot-count* dots.

For duration logs that appear in the `ledgers` property, rest symbols with ledger lines are selected.

```
\markup {
  \rest-by-number #3 #2
  \hspace #2
  \rest-by-number #0 #1
  \hspace #2
  \rest-by-number #-1 #0
  \hspace #2
  \override #'(ledgers . ())
  \rest-by-number #-1 #0
}
```



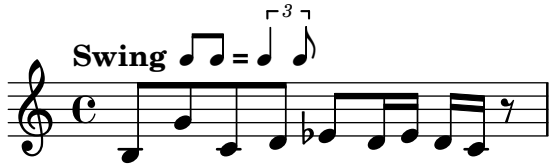
Used properties:

- `style` (())
- `ledgers` ((-1 0 1))
- `font-size` (0)

`\rhythm` *music* (music)

Draw embedded rhythmic pattern as specified by *music*.

```
\relative {
  \tempo \markup {
    Swing
    \hspace #0.4
    \rhythm { 8[ 8] } = \rhythm { \tuplet 3/2 { 4 8 } }
  }
  b8 g' c, d ees d16 ees d c r8
}
```



Within `\rhythm`, there is no time signature and no division in measures (as with `\cadenzaOn`, véase Sección 2.3.4 [Música sin compasear], página 91). Beaming must be added explicitly with the syntax explained in Sección 2.4.3 [Barras manuales], página 114.

```
\markup {
  The rhythmic pattern \rhythm { 16[ 8 16] } is
  a type of syncopation.
}
```

The rhythmic pattern  is a type of syncopation.

`\stemDown` can be used to flip the stems.

```
\markup \rhythm { \stemDown 8 16 8 }
```



\rhythm works by creating a StandaloneRhythmVoice context, which is enclosed in a StandaloneRhythmStaff context, which is enclosed in a StandaloneRhythmScore context. It is possible to apply global tweaks to the output by using a \layout block.

```
\layout {
  \context {
    \StandaloneRhythmVoice
    \xNotesOn
  }
}

\markup \rhythm { 8 16 8 }
```



Used properties:

- font-size (-2)

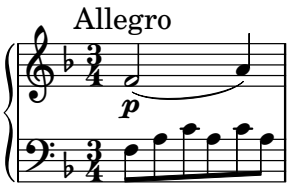
\score *score* (score)

Inline an image of music as specified by *score*.

The reference point (usually the middle staff line) of the lowest staff in the top system is placed on the baseline.

No page breaks and no MIDI output, i.e., both \pageBreak commands and \midi{} blocks get ignored.

```
\markup {
  Text before the score.
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
        \mark \markup { Allegro }
        f2\p( a4)
      }
      \new Staff \relative c {
        \clef bass
        \key f \major
        \time 3/4
        f8( a c a c a
      }
    >>

    \layout {
      indent = 0.0\cm
    }
  }
  Text after the score.
}
```



Text before the score. Text after the score.

Used properties:

- `baseline-skip`

`\segno` Draw a segno symbol.

```
\markup {
  \segno
}
```

𝄋

`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```

𝄳

`\semisharp`

Draw a semisharp symbol.

```
\markup {
  \semisharp
}
```

𝄲

`\sesquiflat`

Draw a 3/2 flat symbol.

```
\markup {
  \sesquiflat
}
```

𝄫

`\sesquisharp`

Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```

𝄪

`\sharp` Draw a sharp symbol.

```
\markup {
  \sharp
}
```

♯

\text-accidental *alteration* (an exact rational number)
: Select an accidental glyph for *alteration* (given as a rational number) that aligns well with text.

```
\markup {
  text
  \tiny { \text-accidental #1/2 \text-accidental #-1/2 }
  text
}
```



Used properties:

- alteration-glyph-name-alist

\text-doubleflat
: Draw a double flat symbol for text.

```
\markup {
  \text-doubleflat
}
```



\text-doublesharp
: Draw a double sharp symbol for text.

```
\markup {
  \text-doublesharp
}
```



\text-flat
: Draw a flat symbol for text.

```
\markup {
  \text-flat
}
```



\text-natural
: Draw a natural symbol for text.

```
\markup {
  \text-natural
}
```



\text-sharp
: Draw a sharp symbol for text.

```
\markup {
  \text-sharp
}
```



\tied-lyric *str* (string)
: Replace '~' tilde symbols with tie characters in *str*.

```
\markup \column {
  \tied-lyric
    "Siam navi~all'onde~algenti Lasciate~in abbandono"
  \tied-lyric
    "Impetuosi venti I nostri~affetti sono"
  \tied-lyric
    "Ogni diletto~e scoglio Tutta la vita~e~un mar."
}
```

Siam navi all'onde algenti Lasciate in abbandono
Impetuosi venti I nostri affetti sono
Ogni diletto e scoglio Tutta la vita e un mar.

Used properties:

- word-space

\varcoda
: Draw a varcoda sign.

```
\markup {
  \varcoda
}
```

A.1.5 Conditional markup

\if *condition?* (procedure) *argument* (markup)
: Test *condition?*, and only insert *argument* if it is true.

The condition is provided as a procedure taking an output definition and a property alist chain. The procedure is applied, and its result determines whether to print the markup. This command is most useful inside oddHeaderMarkup or similar. Here is an example printing page numbers in bold:

```
\paper {
  oddHeaderMarkup =
    \markup \fill-line {
      ""
      \if #print-page-number
           \bold \fromproperty #'page:page-number-string
    }
  evenHeaderMarkup =
    \markup \fill-line {
      \if #print-page-number
           \bold \fromproperty #'page:page-number-string
      ""
    }
}
```

\unless *condition?* (procedure) *argument* (markup)
: Test *condition?*, and only insert *argument* if it is false.

This function is similar to \if; the following example shows how to print the copyright notice on all pages but the last instead of just the first page.

```
\paper {
```

```
  oddFooterMarkup = \markup {
    \unless #on-last-page-of-part \fill-line {
      \fromproperty #'header:copyright
    }
  }
}

\header {
  copyright = "© LilyPond Authors. License: GFDL."
  tagline = "© LilyPond Authors.  Documentation placed
under the GNU Free Documentation License
version 1.3."
}
```

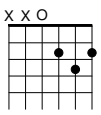
A.1.6 Instrument-specific markup

`\fret-diagram` *definition-string* (string)

Make a (guitar) fret diagram based on *definition-string*.

For example, say

```
\markup \fret-diagram "s:1.25;6-x;5-x;4-o;3-2;2-3;1-2;"
```



for fret spacing 5/4 of staff space, D chord diagram.

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
  - `s:`*number* – Set the fret spacing of the diagram (in staff spaces). Default: 1.
  - `t:`*number* – Set the line thickness (relative to normal line thickness). Default: 0.5.
  - `h:`*number* – Set the height of the diagram in frets. Default: 4.
  - `w:`*number* – Set the width of the diagram in strings. Default: 6.
  - `f:`*number* – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
  - `d:`*number* – Set radius of dot, in terms of fret spacing. Default: 0.25.
  - `p:`*number* – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
  - `c:`*string1-string2-fret* – Include a barre mark from *string1* to *string2* on *fret*.
  - *string-fret* – Place a dot on *string* at *fret*. If *fret* is 'o', *string* is identified as open. If *fret* is 'x', *string* is identified as muted.
  - *string-fret-fingering* – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.
- Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (`0.5`)
- `fret-diagram-details`

- `size` (`1.0`)
- `align-dir` (`-0.4`)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

For example,

```
\markup \fret-diagram-terse "x;x;o;2;3;2;"
```



displays a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by 'x'.
- Open strings are indicated by 'o'.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a '-' followed by the finger indicator, e.g., '3-2' for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with `-(` to start a barre and `-)` to end the barre.

Used properties:

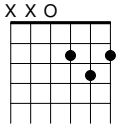
- `thickness` (`0.5`)
- `fret-diagram-details`
- `size` (`1.0`)
- `align-dir` (`-0.4`)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

The following example produces a standard D chord diagram without fingering indications.

```
\markup \scale #'(1.5 . 1.5)
  \fret-diagram-verbose
    #'((mute 6) (mute 5) (open 4)
       (place-fret 3 2) (place-fret 2 3) (place-fret 1 2))
```



Possible elements in *marking-list*:

`(mute` *string-number*`)`

Place a small 'x' at the top of string *string-number*.

`(open` *string-number*`)`

Place a small 'o' at the top of string *string-number*.

(barre *start-string end-string fret-number*)
: Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

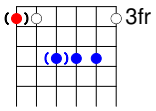
(capo *fret-number*)
: Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

(place-fret *string-number fret-number* [*finger-value*] [*color-modifier*] [*color*] ['parenthesized ['default-paren-color]])
: Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*, an optional color modifier *color-modifier*, an optional color *color*, an optional parenthesis 'parenthesized and an optional parenthesis color 'default-paren-color.

  By default, the fret playing indicator is a solid dot. This can be globally changed by setting the value of the property dot-color or for a single dot by setting the value of *color*. The dot can be parenthesized by adding 'parenthesized. By default the color for the parenthesis is taken from the dot. Adding 'default-paren-color will take the parenthesis color from the global dot-color property; as a fallback black will be used. Setting *color-modifier* to inverted inverts the dot color for a specific fingering.

  The values for *string-number*, *fret-number*, and the optional *finger* should be entered first in that order. The order of the other optional arguments does not matter. If the *finger* part of the place-fret element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

```
\markup \scale #'(1.5 . 1.5)
  \fret-diagram-verbose #'(
    (place-fret 6 3 1 red parenthesized default-paren-color)
    (place-fret 5 3 1 inverted)
    (place-fret 4 5 2 blue parenthesized)
    (place-fret 3 5 3 blue)
    (place-fret 2 5 4 blue)
    (place-fret 1 3 1 inverted)
  )
```



Used properties:

- thickness (0.5)
- fret-diagram-details
- size (1.0)
- align-dir (-0.4)

\harp-pedal *definition-string* (string)
: Make a harp pedal diagram containing the symbols indicated in *definition-string*.

  Possible elements in *definition-string*:

  ^ pedal is up

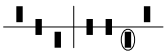
`-` pedal is neutral

`v` pedal is down

`|` vertical divider line

`o` the following pedal should be circled (indicating a change)

```
\markup \harp-pedal "^-v|--ov^"
```



The function also checks whether the string has the typical form of three pedals, then the divider, and then the remaining four pedals, printing a warning otherwise (without preventing the non-standard order).

Use the `size` property to control the overall size, and the `thickness` property for the line thickness of the horizontal line and the divider.

The remaining configuration is done via the `harp-pedal-details` property; it contains the following elements:

`box-offset`
: vertical shift of the box center for up/down pedals

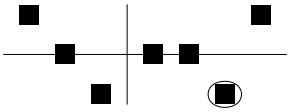
`box-width`
: box width

`box-height`
: box height

`space-before-divider`
: spacing between two boxes before the divider

`space-after-divider`
: spacing between two boxes after the divider

```
\markup {
  \override #'((size . 1.5)
               (harp-pedal-details . ((box-width . 1)
                                      (box-offset . 2))))
  \harp-pedal "^-v|--ov^"
}
```



For global modification of `harp-pedal-details`, i.e., outside of the current `\markup` block, you can also use code similar to

```
\override Voice.TextScript.harp-pedal-details.box-width = 1
```

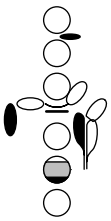
Used properties:

- `thickness` (`0.5`)
- `harp-pedal-details` (`()`)
- `size` (`1.2`)

`\woodwind-diagram` *instrument* (symbol) *user-draw-commands* (list)
: Make a woodwind-instrument diagram for *instrument* using *user-draw-commands*.

user-draw-commands is a list of alists, specifying the left-hand keys, the elements on the central column, and the right-hand keys. For example, this diagram

```
\markup \woodwind-diagram
  #'oboe #'((lh . (d ees))
            (cc . (five3qT1q))
            (rh . (gis)))
```



shows an oboe with the left-hand d key, left-hand ees key, and right-hand gis key depressed, while the five-hole of the central column effectuating a trill between 1/4 and 3/4 is closed.

The following instruments are supported:

- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function call `(print-keys '`*instrument*`)` in your `.ly` file, where *instrument* is the instrument whose keys you want to print.

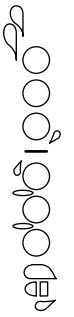
Certain keys allow for special configurations. The entire gamut of configurations possible is as follows:

`1q`	1/4 covered
`1h`	1/2 covered
`3q`	3/4 covered
`R`	ring depressed
`F`	fully covered; the default if no state put

Additionally, these configurations can be used in trills. So, for example, `three3qTR` effectuates a trill between 3/4 full and ring depressed on the three hole. As another example, `threeRT` effectuates a trill between R and open, whereas `threeTR` effectuates a trill between open and shut. To see all of the possibilities for all of the keys of a given instrument, invoke `(print-keys-verbose '`*instrument*`)`.

Lastly, substituting an empty list for the pressed-key alist results in a diagram with all of the keys drawn but none filled, for example

```
\markup \woodwind-diagram #'flute #'()
```



Used properties:

- `woodwind-diagram-details` (`()`)
- `font-size` (`0`)
- `graphical` (`#t`)
- `thickness` (`0.1`)
- `size` (`1`)

A.1.7 Accordion registers

`\discant` *name* (string)

Generate a discant accordion register symbol for *name*.

To make it available,

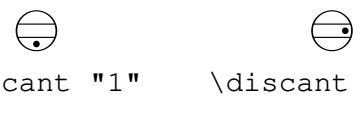
```
#(use-modules (lily accreg))
```

is required near the top of your input file.

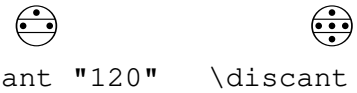
The register names in the default `\discant` register set have been modeled after the numeric Swiss notation (as depicted in `http://de.wikipedia.org/wiki/Register_%28Akkordeon%29`), omitting the slashes and dropping leading zeros.

The string *name* is basically a three-digit number with the lowest digit specifying the number of 16' reeds, the tens the number of 8' reeds, and the hundreds specifying the number of 4' reeds. Without modification, the specified number of reeds in 8' is centered in the symbol. Newer instruments may have registrations where 8' can be used either within or without a tone chamber, 'cassotto'. Notationally, the central dot then indicates use of cassotto. One can suffix the tens' digits '`1`' and '`2`' with '`+`' or '`-`' to indicate clustering the dots at the right or left, respectively, rather than centered.

Some examples are



`\discant "1"` `\discant "1+0"`



`\discant "120"` `\discant "131"`

Used properties:

- `font-size` (`0`)

`\freeBass` *name* (string)

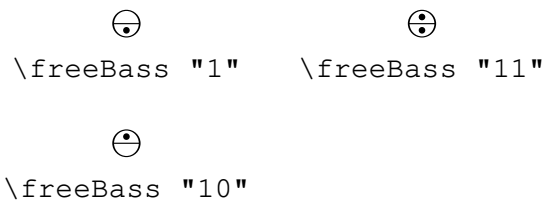
Generate a free bass/converter accordion register symbol for the usual two-reed layout as given by *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

Available registrations are



Used properties:

- `font-size` (0)

`\stdBass` *name* (string)
: Generate a standard bass accordion register symbol for *name*.

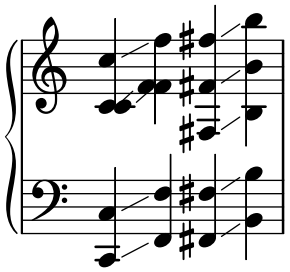
To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

The default bass register definitions have been modeled after the article `http://www.accordions.com/index/art/stradella.shtml` originally appearing in *Accord Magazine*.

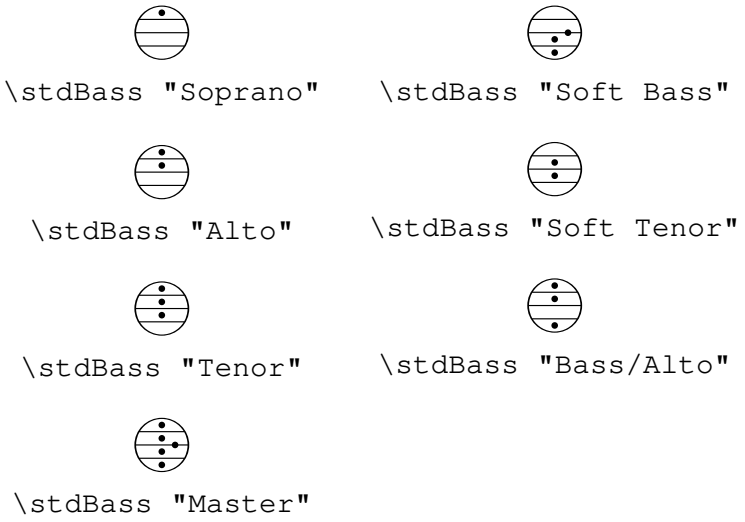
The underlying register model is



This kind of overlapping arrangement is common for Italian instruments though the exact location of the octave breaks differ.

When not composing for a particular target instrument, using the five-reed definitions makes more sense than using a four-reed layout: in that manner, the '`Master`' register is unambiguous. This is rather the rule in literature bothering about bass registrations at all.

Available registrations are



Used properties:

- `font-size` (0)

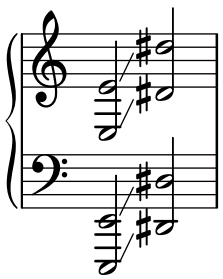
`\stdBassIV` *name* (string)
Generate a standard bass accordion register symbol for *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

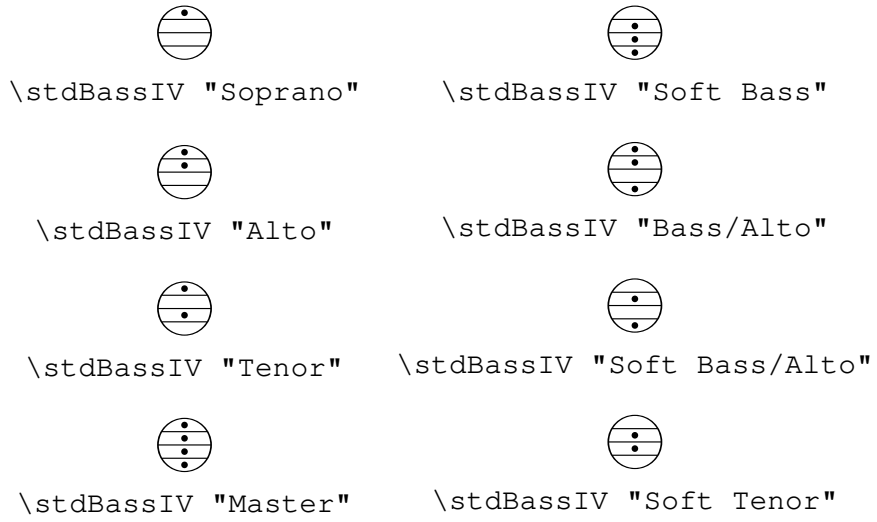
The main use is for four-reed standard bass instruments with reedbank layout



Notable instruments are Morino models with MIII (the others are five-reed instead) and the Atlantic IV. Most of those models have three register switches. Some newer Morinos with MIII might have five or even seven.

The prevalent three-register layout uses the middle three switches '`Tenor`', '`Master`', '`Soft Bass`'. Note that the sound is quite darker than the same registrations of '`c,`'-based instruments.

Available registrations are



Used properties:

- `font-size` (0)

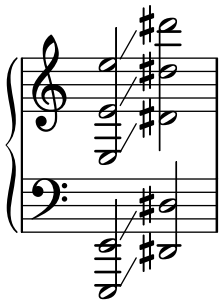
`\stdBassV` *name* (string)
Generate a standard bass accordion register symbol for *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

The main use is for five-reed standard bass instruments with reedbank layout



This tends to be the bass layout for Hohner's Morino series without converter or MIII manual.

With the exception of the rather new 7-register layout, the highest two chord reeds are usually sounded together. Older instruments offer 5 or 3 bass registers. The Tango VM offers an additional '`Solo Bass`' setting that mutes the chord reeds. The symbol on the register buttons of the Tango VM would actually match the physical five-octave layout reflected here, but it is not used in literature.

Composers should likely prefer the five-reed versions of these symbols. The mismatch of a four-reed instrument with five-reed symbols is easier to resolve for the player than the other way round.

Available registrations are



Used properties:

- `font-size` (0)

`\stdBassVI` *name* (string)

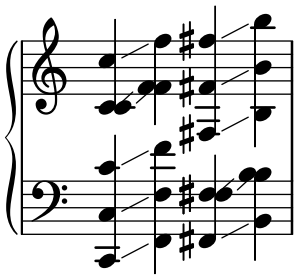
Generate a standard bass accordion register symbol for six-reed basses as given by *name*.

To make it available,

```
#(use-modules (lily accreg))
```

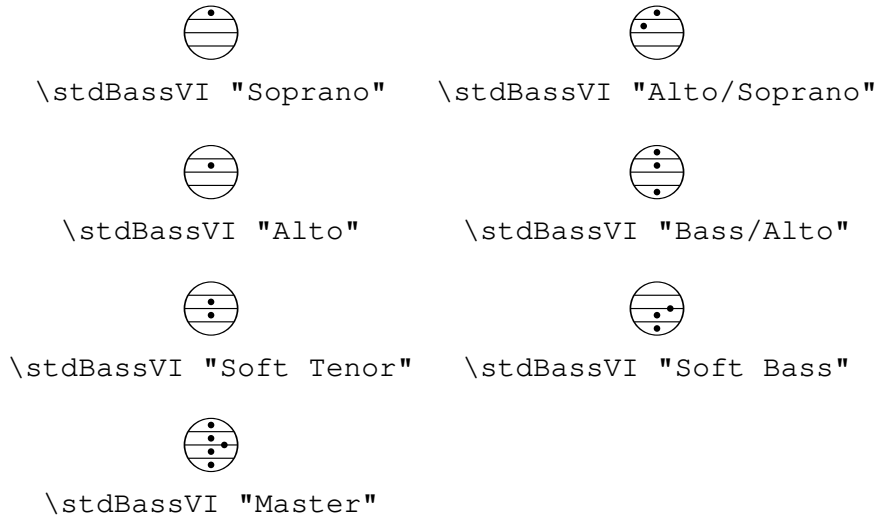
is required near the top of your input file.

This is primarily the register layout for the Hohner "Gola" model. The layout is



The registers are effectively quite similar to that of `\stdBass`. An additional bass reed at alto pitch is omitted for aesthetical reasons from the '`Master`' setting, so the symbols are almost the same except for the '`Alto/Soprano`' register with bass notes at Alto pitch and chords at Soprano pitch.

Available registrations are



Used properties:

- `font-size` (0)

A.1.8 Other markup commands

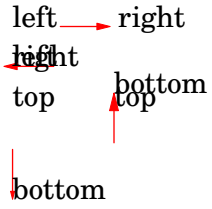
`\annotate-moving` *arg* (markup)

Indicate \vspace and \hspace movement with an arrow.

The arrow changes its size and thickness depending on the printed length; the maximum size of the arrow head can be controlled with the `size` property. If `size` exceeds a third of the length of the final arrow, it falls back to that third.

Note that the arrows do not reflect the actual extents of the objects created by \vspace and \hspace; you might use \box for that.

```
\markup
  \column {
    \line { left \annotate-moving \hspace #4 right }
    \line { left \annotate-moving \hspace #-4 right }
    \line {
      \column {
        top
        \override #'(size . 0.6)
          \annotate-moving \vspace #4/3 bottom
      }
      \column {
        top
        \override #'(size . 2.0)
          \annotate-moving \vspace #-4/3 bottom
      }
    }
  }
```



Used properties:

- `size` (`1`)
- `color` (`"red"`)

`\append-to-tag` *tag* (symbol) *more* (markup) *arg* (markup)

Append *more* to all markup in var *arg* tagged with *tag*.

It works similar to `\appendToTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}

\markup { \append-to-tag #'foo postfoo \tagged }
```



Used properties:

- `tags-with-appends-alist` (`()`)

`\auto-footnote` *mkup* (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```



The footnote will be annotated automatically.

Used properties:

- `padding` (`0.0`)
- `raise` (`0.5`)

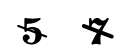
`\backslashed-digit` *num* (integer)

Print number *num* with the Emmentaler font, crossed through with a backslash.

This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
```

```
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```



Used properties:

- `thickness` (`1.6`)
- `font-size` (`0`)

`\char` *num* (integer)
: Produce a single Unicode character with code *num*.

  Characters encoded in hexadecimal format require the prefix `#x`.

```
\markup {
  \char #65 \char ##x00a9
}
```



`\eyeglasses`
: Prints out eyeglasses, indicating strongly to look at the conductor.

```
\markup { \eyeglasses }
```



`\first-visible` *args* (markup list)
: Use the first markup in *args* that yields a non-empty stencil and ignore the rest.

```
\markup {
  \first-visible {
    \fromproperty #'header:composer
    \italic Unknown
  }
}
```



`\footnote` *mkup* (markup) *note* (markup)
: Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \footnote a b
  \override #'(padding . 0.2)
  \footnote c d
}
```

a c

b
d

The footnote will not be annotated automatically.

`\fraction` *arg1* (markup) *arg2* (markup)
Make a fraction of markups *arg1* and *arg2*.

```
\markup {
  π ≈ \fraction 355 113
}
```

$$\pi \approx \frac{355}{113}$$

Used properties:

- `font-size` (0)

`\fromproperty` *symbol* (symbol)
Read *symbol* from the property settings and produce a stencil from the markup contained within.

If *symbol* is not defined or is not a markup, return an empty markup.

Currently, the following properties can be accessed.

- Within a `\paper` block defining titles, headers, or footers, or within a `\header` block: all fields from the `\header` block (that produce markup) are available, with `header:` as a name prefix.
- Within a `\paper` block defining headers or footers: the current page number (symbol `page:page-number-string`).
- Within the `tocItemMarkup` paper variable (or in custom-made Scheme code that uses function `add-toc-item!`) defining a table of contents entry: the entry's text and page number are available as `toc:text` and `toc:page`, respectively. An entry's indentation markup is available as `toc:indent`.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
```

```
  \null
}
```

from *myTitle*

\keep-with-tag *tags* (symbol list or symbol) *arg* (markup)

Keep markup from *arg* that is untagged or tagged with *tags*.

All other parts of *arg* that are using a different tag are replaced with empty stencils. It works similar to \keepWithTag for music, but only with markups.

```
tagged = \markup {
  untagged
  \tag #'foo A
  \tag #'bar B
}

\markup { \keep-with-tag #'bar \tagged }
\markup { \keep-with-tag #'foo \tagged }
```

untagged B

untagged A

Used properties:

- tags-to-keep (())

\left-brace *size* (number)

Print a brace from the music font, of height *size* (in points).

```
\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}
```



\lookup *glyph-name* (string)

Print a brace glyph with name *glyph-name*.

This is a historical command; \left-brace (which looks up the glyph by absolute size and is independent of the font size) is recommended instead.

```
\markup \lookup "brace200"
```



\markalphabet *num* (integer)

Make a markup letter for *num*.

The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

H Z

`\markletter` *num* (integer)

Make a markup letter for *num*.

The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

H AA

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (procedure) *arg* (markup)

Apply the *procedure* markup command to *arg*.

procedure takes the same arguments as `interpret-markup` and returns a stencil.

`\override` *new-prop* (pair) *arg* (markup)

Add the argument *new-prop* to the property list for printing *arg*.

In general, any property may be overridden that is part of `font-interface` (Sección "font-interface" en *Referencia de Funcionamiento Interno*), `text-interface` (Sección "text-interface" en *Referencia de Funcionamiento Interno*), or `instrument-specific-markup-interface` (Sección "instrument-specific-markup-interface" en *Referencia de Funcionamiento Interno*). Additionally, various markup commands listen to other properties, too, as described in a markup function's documentation.

new-prop is either a single alist pair or a non-empty list of alist pairs.

```
\markup {
  \undertie "undertied"
  \override #'(offset . 15)
  \undertie "offset undertied"
  \override #'((offset . 15) (thickness . 3))
  \undertie "offset thick undertied"
}
```

undertied offset undertied offset thick undertied

`\page-link` *page-number* (number) *arg* (markup)

Add a link to a score's page *page-number* around *arg*.

This only works in the PDF backend.

```
\markup {
  \page-link #2 { \italic { This links to page 2... } }
}
```

This links to page 2...

\page-ref *label* (symbol) *gauge* (markup) *default* (markup)
: Print a page number reference.

  label is the label set on the referenced page (using \label or \tocItem), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

  If the current book or book part is set to use roman numerals for page numbers, the reference will be formatted accordingly – in which case the *gauge*'s width may require additional tweaking.

  Used properties:

  - x-align (1)

\pattern *count* (non-negative, exact integer) *axis* (non-negative, exact integer) *space* (number) *pattern* (markup)
: Print a *pattern* markup *count* times.

  Patterns are spaced apart by *space* (defined as for \hspace or \vspace, respectively) and distributed on *axis*.

```
\markup \column {
  "Horizontally repeated:"
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated:"
  \pattern #3 #Y #0.5 \flat
}
```

Horizontally repeated:

♭ ♭ ♭ ♭ ♭ ♭ ♭

Vertically repeated:

♭

♭

♭

\property-recursive *symbol* (symbol)
: Print out a warning when header field markup in *symbol* contains some recursive markup definition.

\push-to-tag *tag* (symbol) *more* (markup) *arg* (markup)
: Prepend *more* to all markup in *arg* tagged with *tag*.

  It works similar to \pushToTag for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \push-to-tag #'foo prefoo \tagged }
```

prefoo A B

Used properties:

- `tags-with-pushes-alist` (())

`\qr-code` *width* (non-negative number) *str* (string)

Insert a QR code for string *str*, usually a URL, with a given *width*.

```
\markup \vcenter {
  \center-column { Engraved with LilyPond }
  \hspace #1.5
  \qr-code #10.0 "https://lilypond.org"
}
```

Engraved with LilyPond 

The `error-correction-level` property can be set to one of the symbols `low`, `medium`, `quarter`, or `high`. The higher the level of error correction is, the more the QR code contains redundancy, potentially helping detectors, e.g., in poor lighting conditions; however, a higher correction level also makes the code denser.

```
\markup \vcenter {
  \center-column { Engraved with LilyPond }
  \hspace #1.5
  \override #'(error-correction-level . high)
    \qr-code #10.0 "https://lilypond.org"
}
```

Engraved with LilyPond 

The `quiet-zone-size` property specifies the width of the "quiet zone", namely the white area around the QR code. It is expressed as a multiple of the width of one little square inside the QR code. Use at least 4 for best results.

Used properties:

- `quiet-zone-size` (4)
- `error-correction-level` (`low`)

`\remove-with-tag` *tags* (symbol list or symbol) *arg* (markup)

Remove markup from *arg* that is tagged with *tags*.

The removed markup is replaced with empty stencils. It works similar to `\removeWithTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \remove-with-tag #'foo \tagged }
\markup { \remove-with-tag #'bar \tagged }
```

B

A

Used properties:

- `tags-to-remove` (`()`)

`\replace-with-tag` *tag* (symbol) *replacement* (markup) *arg* (markup)

Replace tagged markups in *arg*.

Everything tagged with *tag* (including the tagging itself) gets replaced with *replacement*.

It works similar to `\replaceWithTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}

\markup { \replace-with-tag #'foo C \tagged }
```

C B

Used properties:

- `tags-with-replacement-alist` (`()`)

`\right-brace` *size* (number)

A music brace in point size *size*, rotated 180 degrees.

```
\markup {
  \right-brace #45
  \hspace #2
  \right-brace #35
}
```



`\slashed-digit` *num* (integer)

Print number *num* with the Emmentaler font, crossed through with a slash.

This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```

5 7

Used properties:

- `thickness` (`1.6`)

- font-size (0)

\stencil *stil* (stencil)
: Use stencil *stil* as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



\strut
: Create a box of the same height as the space in the current font.

\tag *tags* (symbol list or symbol) *arg* (markup)
: Tag markup *arg* with *tag*.

tag can be one or multiple tags. This allows later on to reference *arg*; for example, to remove it or to add markup before or after the tagged markup. It works similar to \tag for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}

\markup { \keep-with-tag #'bar \tagged }
\markup { \keep-with-tag #'foo \tagged }
```

B

A

Used properties:

- tags-with-replacement-alist (())
- tags-with-appends-alist (())
- tags-with-pushes-alist (())
- tags-to-remove (())
- tags-to-keep (())

\transparent *arg* (markup)
: Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

\verbatim-file *name* (string)
: Read the contents of file *name* and include it verbatim.

```
\markup {
  \verbatim-file "en/included/simple.ly"
}
```

```
% A simple piece in LilyPond, a scale.
\version "2.19.21"
\relative {
  c' d e f g a b c
}
```

Use `\withRelativeDir` as a prefix to *name* if the file should be found relative to the input file.

`\whiteout` *arg* (markup)

Provide a white background for *arg*.

The shape of the white background is determined by the `style` property. The default is `box` which produces a rectangle. `rounded-box` produces a rounded rectangle, and `outline` approximates the outline of the markup.

The color of the background can be controlled with the `color` property, defaulting to `"white"`.

```
\markup {
  \combine
    \filled-box #'(-1 . 62) #'(-3 . 4) #1
    \override #'(line-width . 60)
      \fill-line {
        \override #'(thickness . 1.5)
          \whiteout box
        \override #'((style . rounded-box) (thickness . 3))
          \whiteout rounded-box
        \override #'((style . outline) (thickness . 3))
          \whiteout outline
        \override #'((color . "red") (style . outline))
          \whiteout red-outline
      }
}
```



Used properties:

- `color` (`"white"`)
- `thickness` (`()`)
- `style` (`box`)

`\with-color` *col* (color) *arg* (markup)

Use color *col* to draw *arg*.

Véase Sección 7.1.5 [Colorear objetos], página 292, for valid color specifications.

```
\markup {
  \with-color #red red
  \hspace #2
  \with-color #green green
  \hspace #2
  \with-color "#0000ff" blue
}
```

red green blue

`\with-dimension` *axis* (integer) *val* (pair of numbers) *arg* (markup)
Set the dimension of *arg* along *axis* to *val*.

If *axis* is equal to `X`, set the horizontal dimension. If *axis* is equal to `Y`, set the vertical dimension.

`\with-dimension-from` *axis* (integer) *arg1* (markup) *arg2* (markup)
Print *arg2* but replace the dimension along *axis* with the one from *arg1*.

If *axis* is set to `X`, replace the horizontal dimension. If *axis* is set to `Y`, replace the vertical dimension.

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the horizontal and vertical dimensions of *arg* to *x* and *y*.

`\with-dimensions-from` *arg1* (markup) *arg2* (markup)
Print *arg2* with the horizontal and vertical dimensions of *arg1*.

`\with-link` *label* (symbol) *arg* (markup)
Add a link to the page holding label *label* around *arg*.

This only works in the PDF backend.

```
\markup {
  \with-link #'label {
    \italic { This links to the page
              containing the label... }
  }
}
```

`\with-outline` *outline* (markup) *arg* (markup)
Print *arg* with the outline and dimensions of *outline*.

The outline is used by skylines to resolve collisions (not for whiteout).

`\with-true-dimension` *axis* (integer) *arg* (markup)
Give *arg* its actual dimension (extent) on *axis*.

Sometimes, the extents of a markup's printed ink differs from the default extents. The main case is if glyphs are involved. By default, the extents of a glyph are based on the glyph's *metrics* (i.e., a default vertical and horizontal size for the glyph), which, for various reasons, are often not identical to its *bounding box* (i.e., the smallest rectangle that completely encompasses the glyph's outline) – in most cases, the outline protrudes the box spanned up by the metrics.

```
\markup {
  text
  \fontsize #10
  \override #'((box-padding . 0) (thickness . 0.2))
  \box
    \musicglyph "scripts.trill"
  text
}
```



For purposes other than setting text, this behavior may not be wanted. You can use `\with-true-dimension` in order to give the markup its actual printed extent.

```
\markup {
```

```
  text
  \fontsize #10
  \override #'((box-padding . 0) (thickness . 0.2))
  \box
    \with-true-dimension #X
    \musicglyph "scripts.trill"
  text
}
```



`\with-true-dimensions` *arg* (markup)
: Give *arg* its actual dimensions (extents).

  Calling

  ```
  \markup \with-true-dimensions arg
  ```

  is short for

  ```
  \markup
    \with-true-dimension #X
    \with-true-dimension #Y
    arg
  ```

  i.e., `\with-true-dimensions` has the effect of `\with-true-dimension` on both axes.

A.2 Instrucciones de lista de marcado de texto

Todas las instrucciones siguientes se pueden usar con `\markuplist`.

`\column-lines` *args* (markup list)
: Stack the markups in *args* vertically.

  Like `\column`, but return a list of lines instead of a single markup. The property `baseline-skip` determines the space between each markup in *args*.

  Used properties:

  - `baseline-skip`

`\justified-lines` *args* (markup list)
: Print *args* as lines aligned both at the left and the right.

  Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

  Used properties:

  - `text-direction` (1)
  - `word-space`
  - `line-width` (#f)
  - `baseline-skip`

`\override-lines` *new-prop* (pair) *args* (markup list)
: Add the argument *new-prop* to the property list for printing *args*.

  Like `\override` but for markup lists.

`\score-lines` *score* (score)
: Inline an image of music as specified by *score*.

Like `\score` but return a list of lines instead of a single markup.

Used properties:

- `tags-with-replacement-alist` (())
- `tags-with-appends-alist` (())
- `tags-with-pushes-alist` (())
- `tags-to-remove` (())
- `tags-to-keep` (())

`\string-lines` *str* (string)

Split string *str* into lines.

The character to split at is specified by the property `split-char`, defaulting to `#\newline`. Surrounding whitespace is removed from every resulting string. The returned list of markups is ready to be formatted by other markup or markup list commands like `\column`, `\line`, etc.

```
\markup {
  \column
    \string-lines
     "foo, foo,
     bar, bar,
     buzz, buzz!"
}
```

foo, foo,
bar, bar,
buzz, buzz!

Used properties:

- `split-char` (`#\newline`)

`\table` *column-align* (number list) *lst* (markup list)

Print a table.

column-align specifies how each column is aligned; possible values are -1, 0, and 1. The number of elements in *column-align* determines how many columns will be printed.

The entries to print are given by *lst*, a markup list. If needed, the last row is filled up with `point-stencils`.

Override the `padding` property to increase the horizontal distance between columns. Override `baseline-skip` to increase the vertical distance between rows.

```
% A markup command to print a fixed-width number.
\markup fwnum =
  \markup \override #'(font-features . ("ss01" "-kern"))
    \number \etc

\markuplist {
  \override #'(padding . 2)
  \table #'(0 1 0 -1) {
    \underline { center-aligned right-aligned
                 center-aligned left-aligned }
    one      \fwnum    1 thousandth \fwnum 0.001
    eleven   \fwnum   11 hundredth  \fwnum 0.01
    twenty   \fwnum   20 tenth      \fwnum 0.1
```

```
    thousand \fwnum 1000 one        \fwnum 1.0
  }
}
```

center-aligned	right-aligned	center-aligned	left-aligned
one	1	thousandth	0.001
eleven	11	hundredth	0.01
twenty	20	tenth	0.1
thousand	1000	one	1.0

Used properties:

- `baseline-skip`
- `padding` (0)

`\table-of-contents`
: Print a table of contents.

  This function uses the paper variable `tocTitleMarkup` for the title; it then prints `\tocItem` entries line by line.

  Véase Sección 21.7 [Tabla de contenidos], página 623, for a complete discussion.

  Used properties:

  - `baseline-skip`

`\tag-list` *tags* (symbol list or symbol) *arg* (markup list)
: Tag markup list *arg* with *tag*.

  tag can be one or multiple tags. This allows later on to reference *arg*; for example, to remove it or to add markup before or after the tagged markup list.

  It works like the `\tag` command for markups but with markup lists. You will need it if you have to reference a whole list; for example, to use `\push-to-tag` and `\append-to-tag` without pushing or appending before or after every single item of the list, but before or after the whole list instead.

```
tagged = \markuplist {
  \tag-list #'foo { foo bar }
}

\markup { \push-to-tag #'foo test \tagged }
```

test foo bar

  Used properties:

  - `tags-with-replacement-alist` (())
  - `tags-with-appends-alist` (())
  - `tags-with-pushes-alist` (())
  - `tags-to-remove` (())
  - `tags-to-keep` (())

`\wordwrap-lines` *args* (markup list)

Print *args* as left-aligned lines.

Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where X is the number of staff spaces.

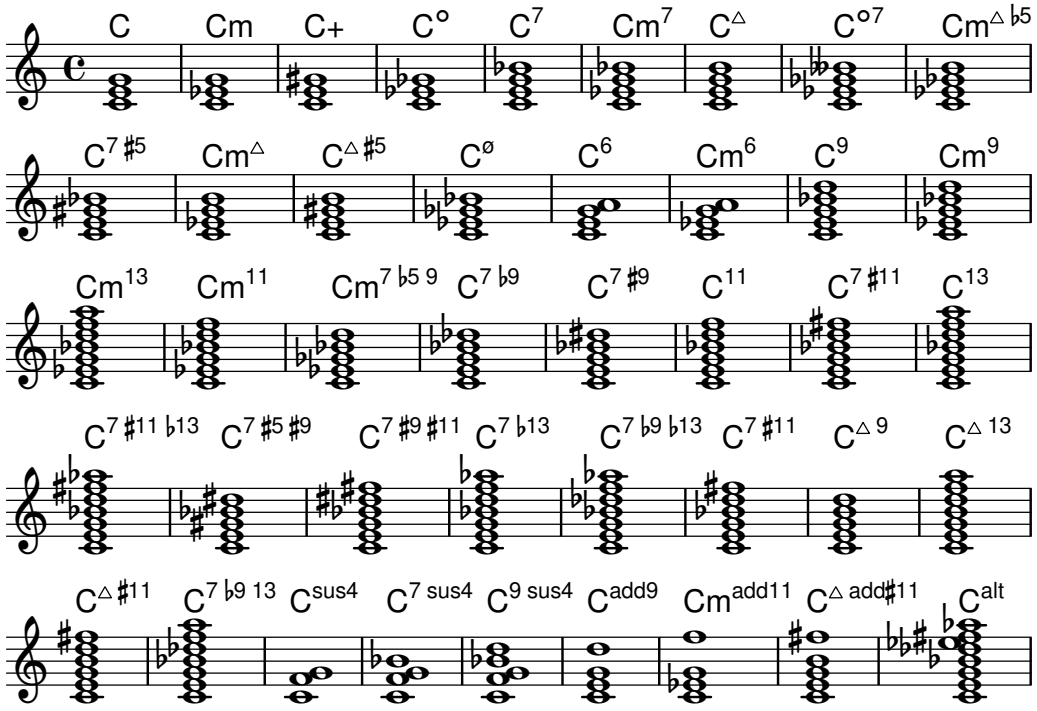
Used properties:

- `text-direction` (`1`)
- `word-space`
- `line-width` (`#f`)
- `baseline-skip`

Apéndice B Tablas del manual sobre notación

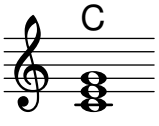
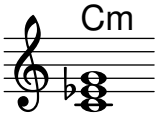
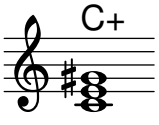
B.1 Cuadro de nombres de acordes

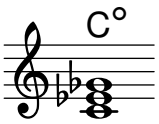
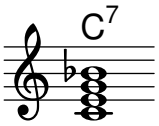
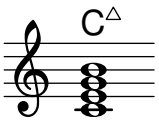
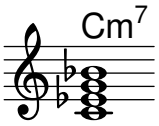
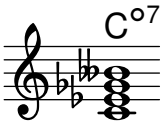
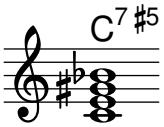
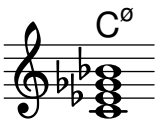
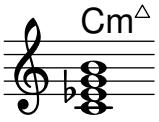
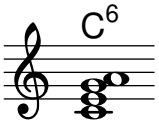
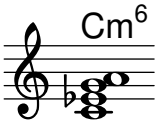
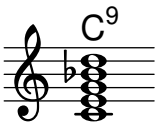
La siguiente tabla muestra el sistema estándar para imprimir los nombres de los acordes, junto a las notas que representa. Se muestran también otros sistemas de nomenclatura (no soportados) en el fragmento de código "Chord names alternative" de la sección Sección "Notación de los acordes" en *Fragmentos de código*, incluida la notación inspirada por Harald Banter (1982) que se utilizaba de forma predeterminada en algunas de las primeras versiones de LilyPond (hasta la 1.7).

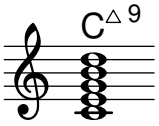
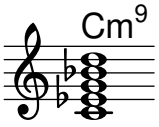
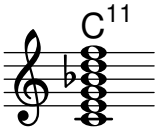
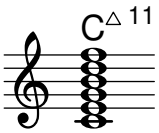
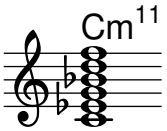
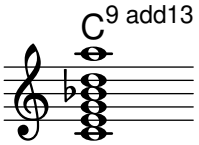
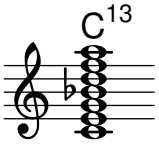
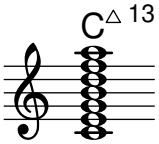
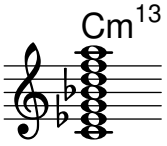
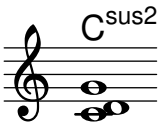
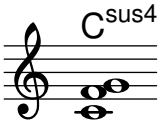


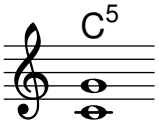
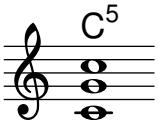
B.2 Modificadores de acorde más usuales

La tabla siguiente presenta los modificadores de acorde que se pueden usar para generar las estructuras de acordes estándar.

Tipo	**Intervalo**	**Modificador**	**Ejemplo**	**Salida**
Mayor	Tercera mayor, quinta justa	(nada)	`c1`	
Menor	Tercera menor, quinta justa	`m` o `m5`	`c1:m`	
Aumentado	Tercera mayor, quinta aumentada	`aug`	`c1:aug`	

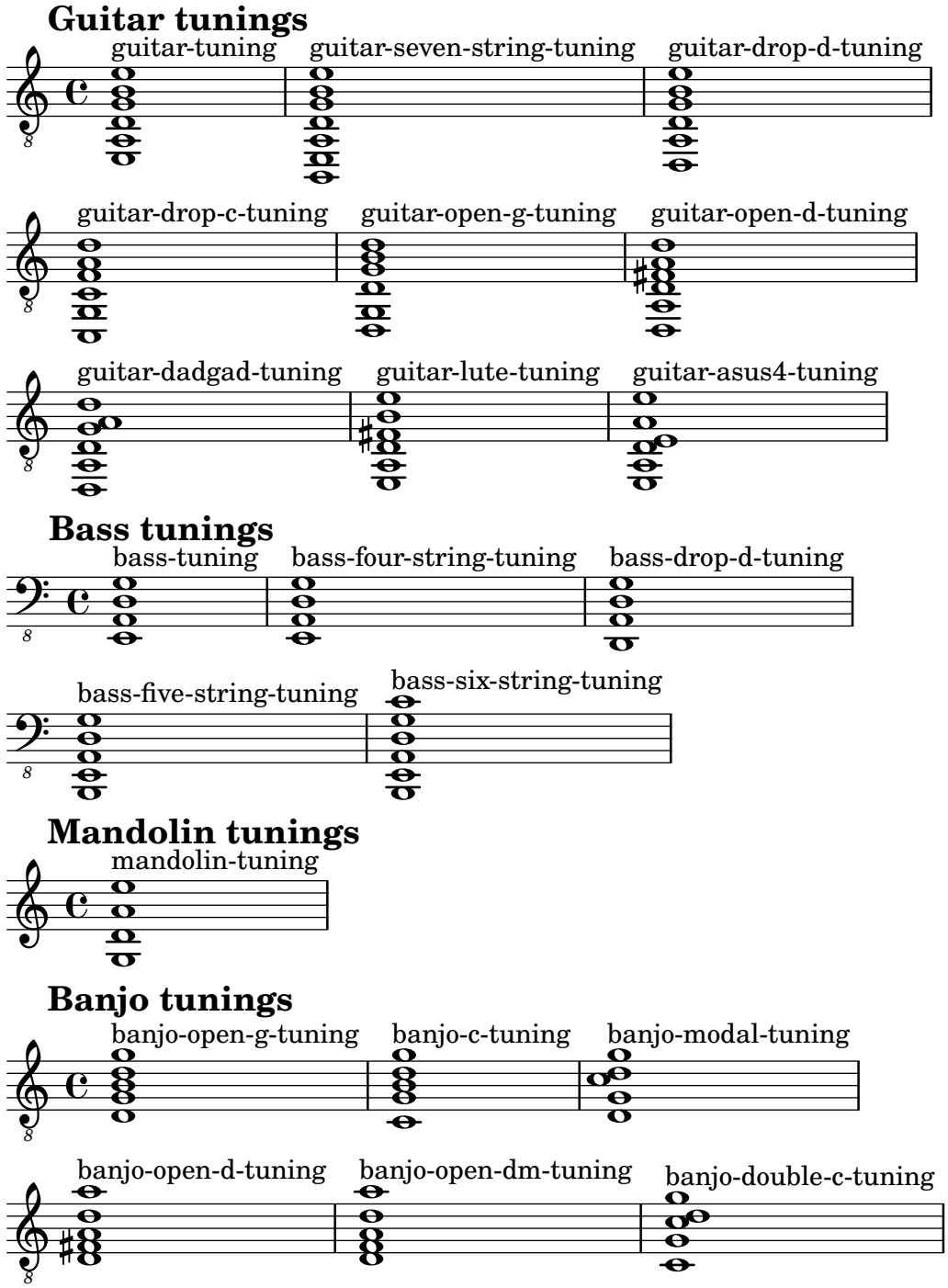
Disminuido	Tercera menor, quinta disminuida	`dim`	`c1:dim`	
Séptima dominante	Tríada mayor, séptima menor	`7`	`c1:7`	
Séptima mayor	Tríada mayor, séptima mayor	`maj7` ó `maj`	`c1:maj7`	
Séptima menor	Tríada menor, séptima menor	`m7`	`c1:m7`	
Séptima disminuida	Tríada disminuida, séptima disminuida	`dim7`	`c1:dim7`	
Aumentado con séptima	Tríada aumentada, séptima menor	`aug7`	`c1:aug7`	
Séptima semi-disminuido	Tríada disminuida, séptima menor	`m7.5-`	`c1:m7.5-`	
Menor con séptima mayor	Tríada menor, séptima mayor	`m7+`	`c1:m7+`	
Mayor con sexta	Tríada mayor, sexta	`6`	`c1:6`	
Menor con sexta	Tríada menor, sexta	`m6`	`c1:m6`	
Novena dominante	Séptima dominante, novena mayor	`9`	`c1:9`	

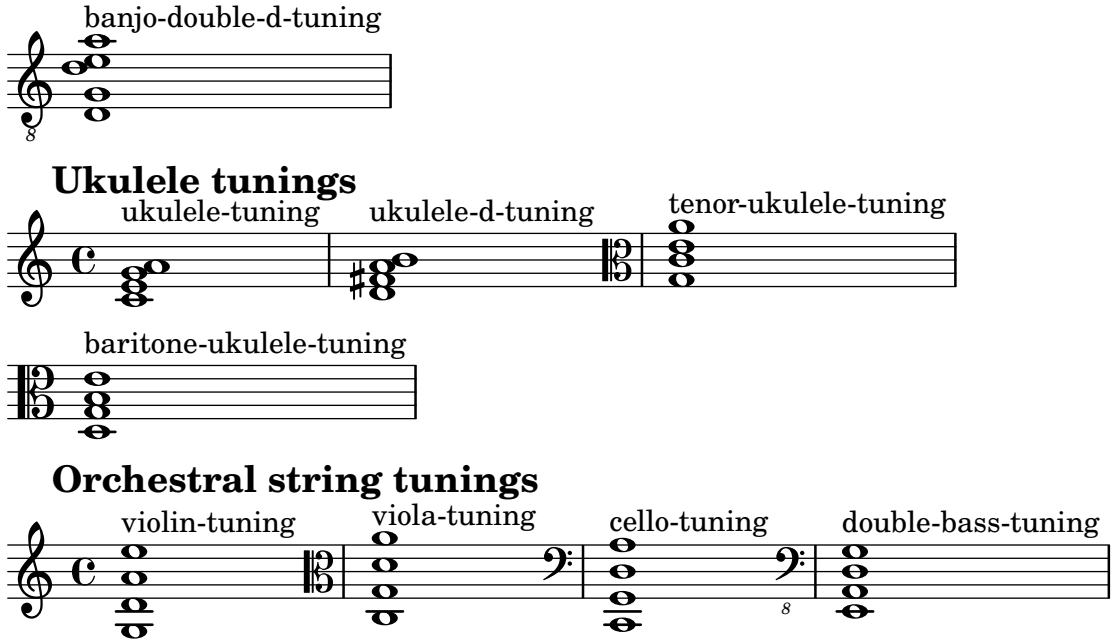
Novena mayor	Séptima mayor, novena mayor	`maj9`	`c1:maj9`	
Menor con novena	Séptima menor, novena mayor	`m9`	`c1:m9`	
Oncena dominante	Novena dominante, oncena justa	`11`	`c1:11`	
Oncena mayor	Novena mayor, oncena justa	`maj11`	`c1:maj11`	
Menor con oncena	Novena menor, oncena justa	`m11`	`c1:m11`	
Trecena dominante	Novena dominante, trecena mayor	`13`	`c1:13`	
Trecena dominante	Oncena dominante, trecena mayor	`13.11`	`c1:13.11`	
Mayor con trecena	Oncena mayor, trecena mayor	`maj13.11`	`c1:maj13.11`	
Menor con trecena	Oncena menor, trecena mayor	`m13.11`	`c1:m13.11`	
Segunda suspendida	Segunda mayor, quinta justa	`sus2`	`c1:sus2`	
Cuarta suspendida	Cuarta justa, quinta justa	`sus4`	`c1:sus4`	

Acorde de potencia o de quinta vacía (de dos notas)	Quinta justa	`1.5`	`c1:5`	
Acorde de potencia o de quinta vacía (de tres notas)	Quinta justa, octava	`1.5.8`	`c1:5.8`	

B.3 Afinaciones predefinidas de instrumentos de cuerda

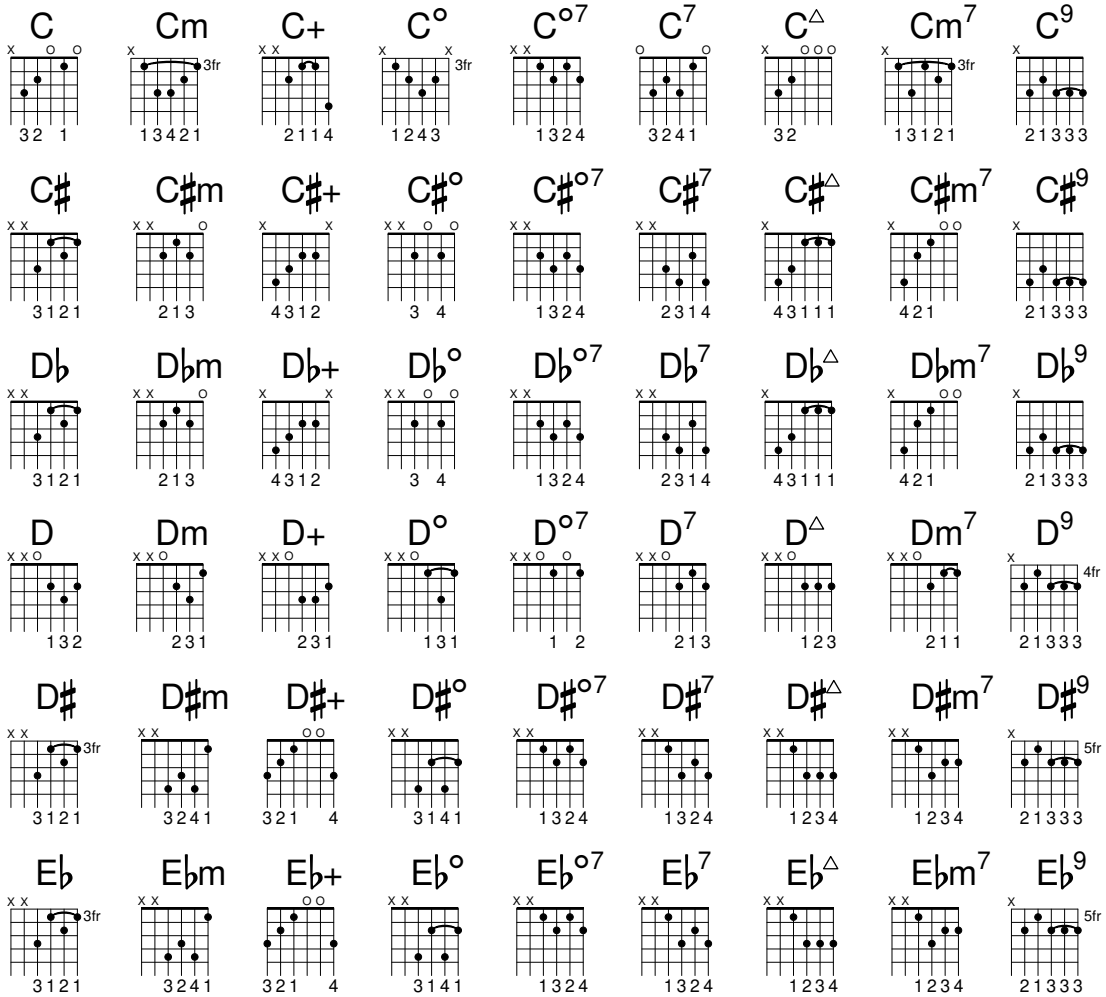
El cuadro de abajo muestra las afinaciones predefinidas de cuerdas.

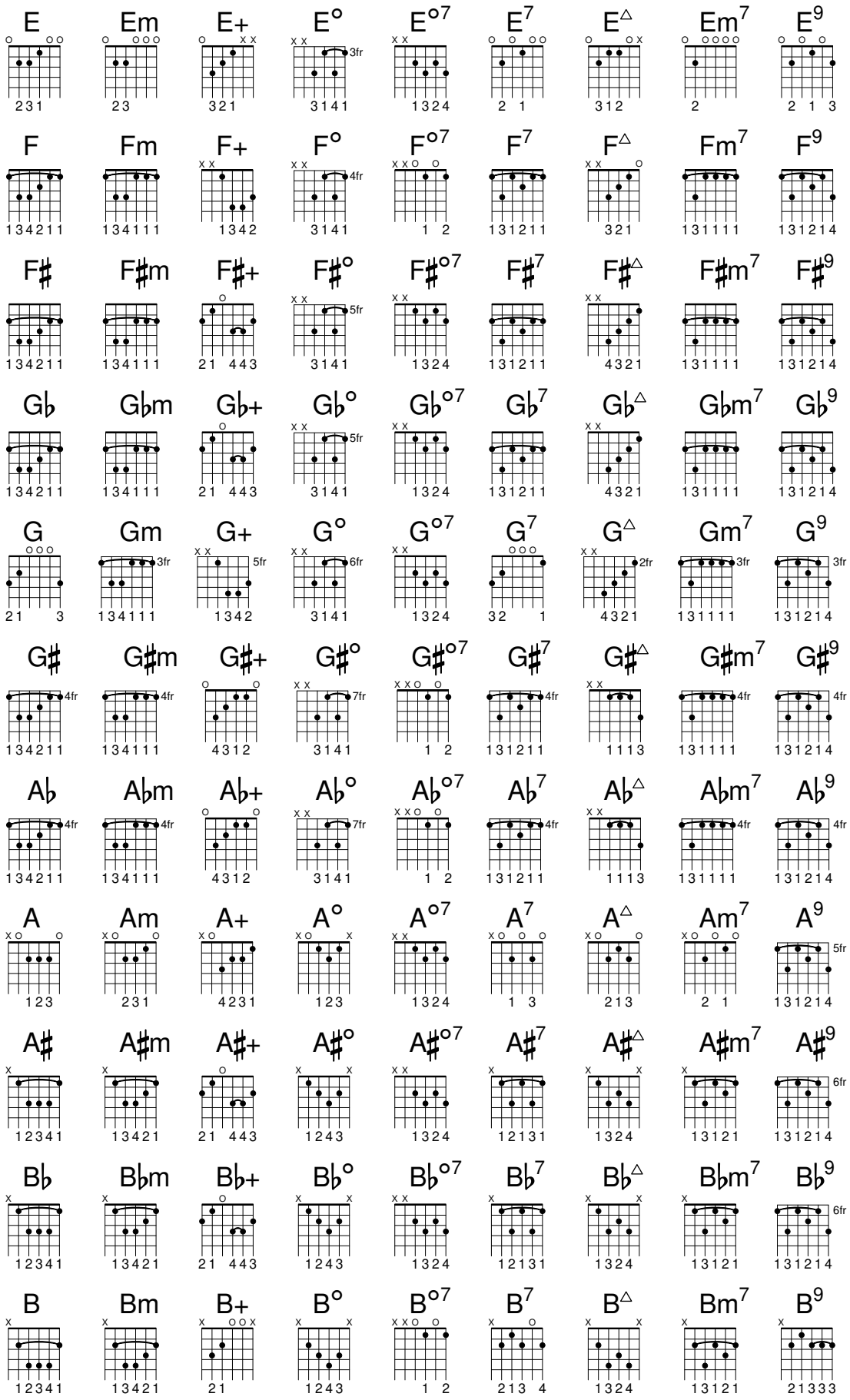




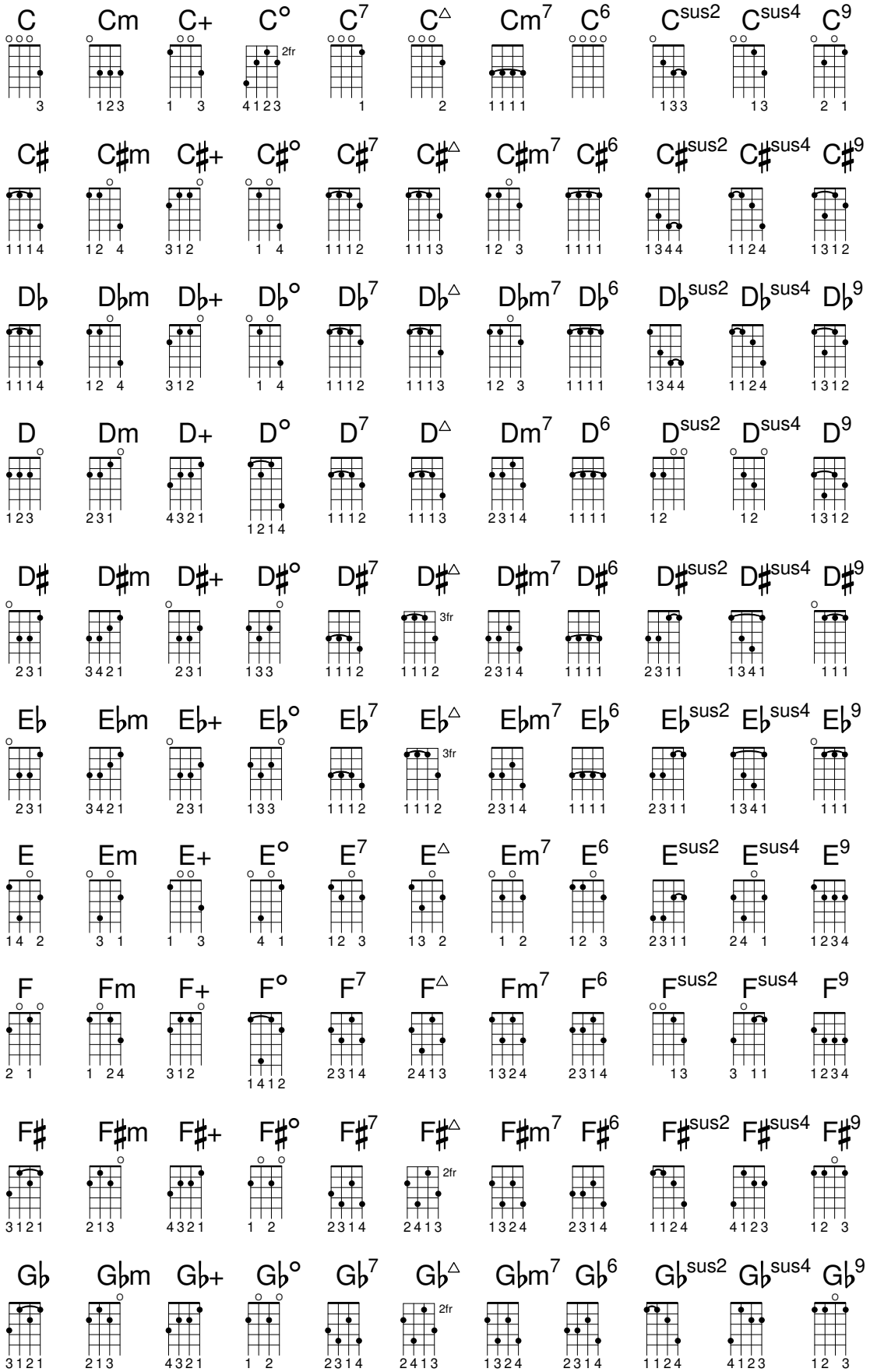
B.4 Diagramas predefinidos de posiciones de trastes

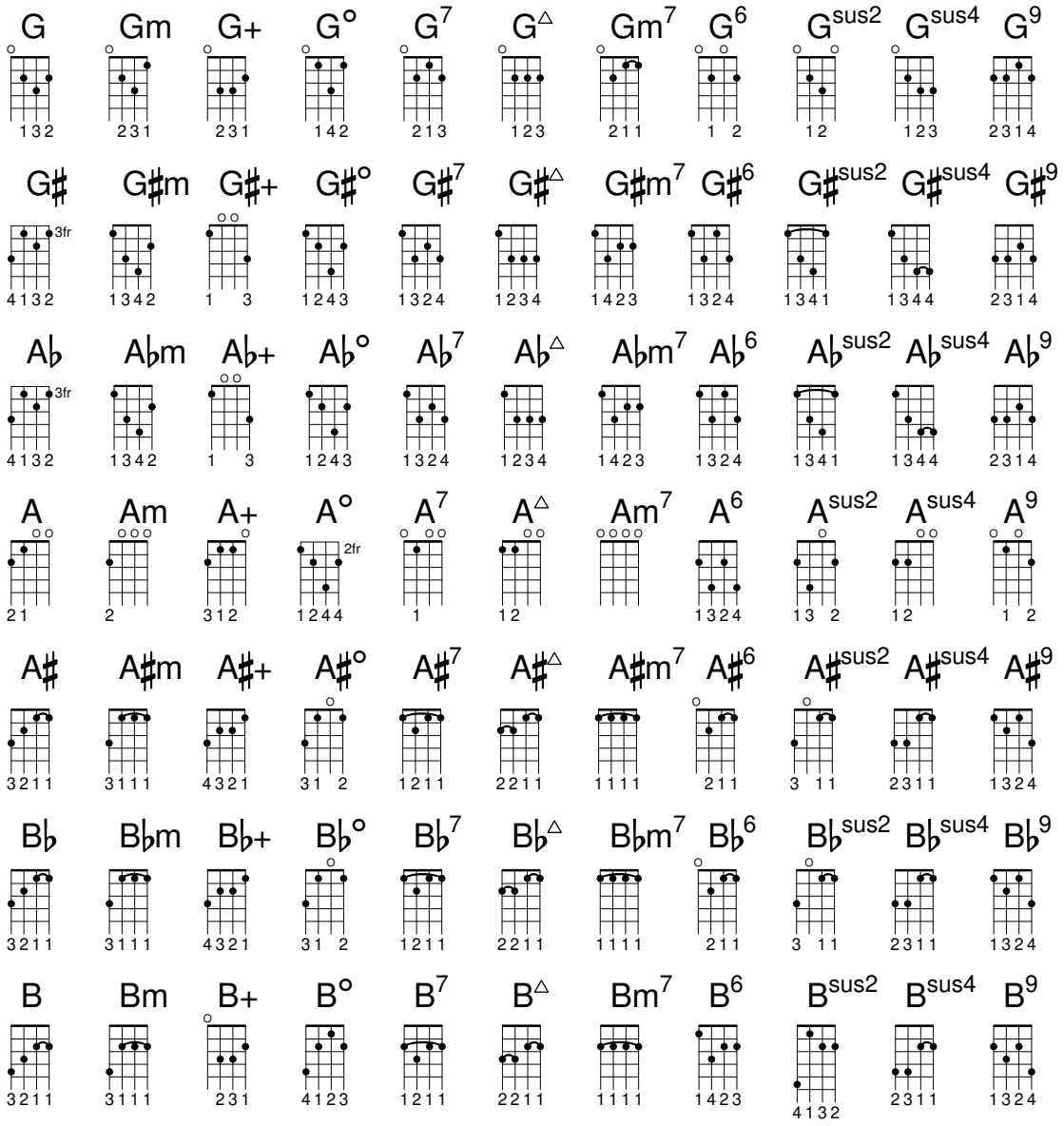
B.4.1 Diagramas para guitarra



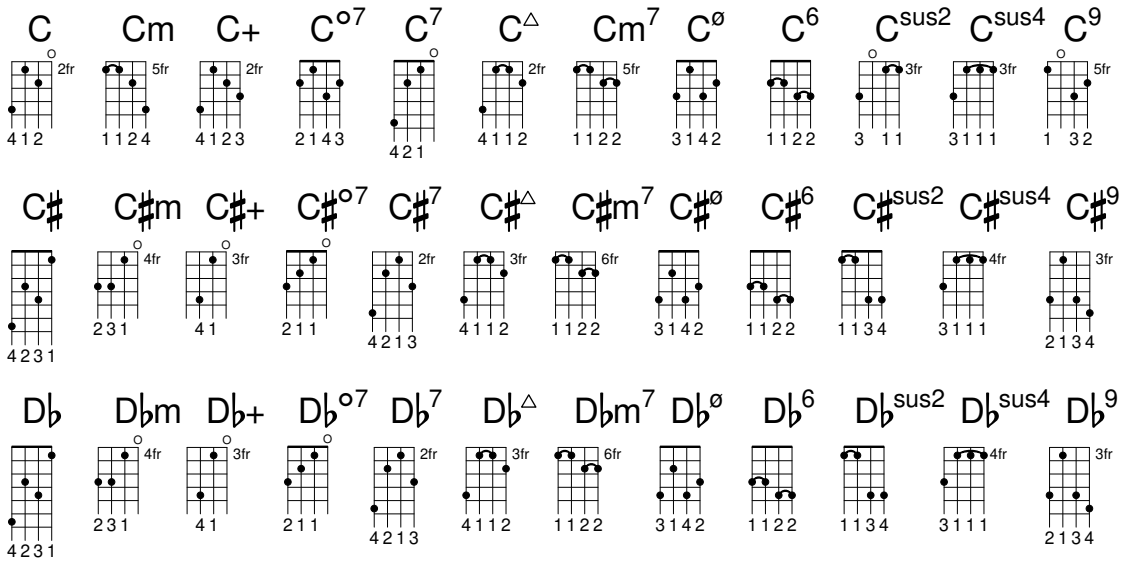

E Em E+ E° E°7 E7 E△ Em7 E9
F Fm F+ F° F°7 F7 F△ Fm7 F9
F♯ F♯m F♯+ F♯° F♯°7 F♯7 F♯△ F♯m7 F♯9
G♭ G♭m G♭+ G♭° G♭°7 G♭7 G♭△ G♭m7 G♭9
G Gm G+ G° G°7 G7 G△ Gm7 G9
G♯ G♯m G♯+ G♯° G♯°7 G♯7 G♯△ G♯m7 G♯9
A♭ A♭m A♭+ A♭° A♭°7 A♭7 A♭△ A♭m7 A♭9
A Am A+ A° A°7 A7 A△ Am7 A9
A♯ A♯m A♯+ A♯° A♯°7 A♯7 A♯△ A♯m7 A♯9
B♭ B♭m B♭+ B♭° B♭°7 B♭7 B♭△ B♭m7 B♭9
B Bm B+ B° B°7 B7 B△ Bm7 B9

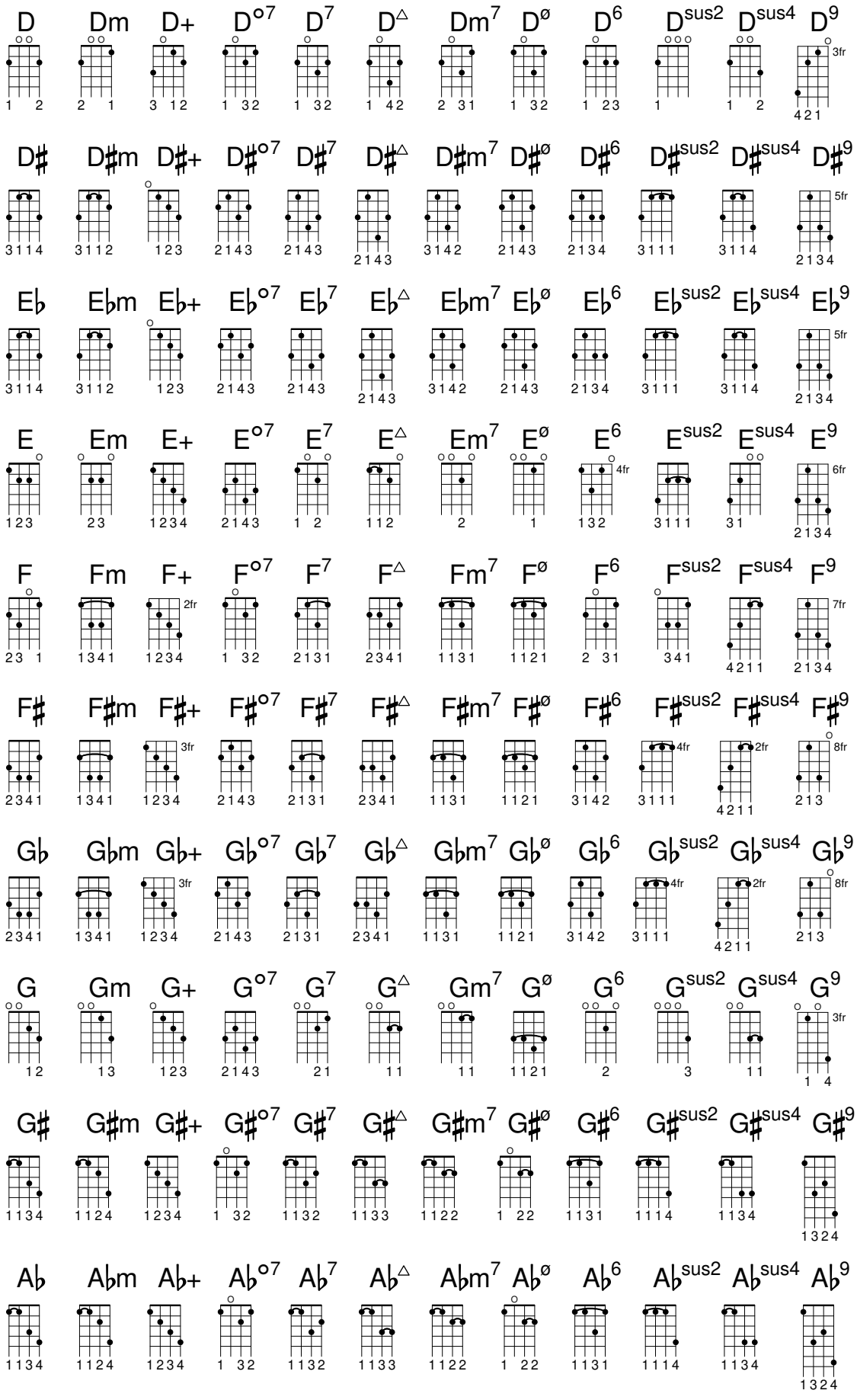
B.4.2 Diagramas para ukelele

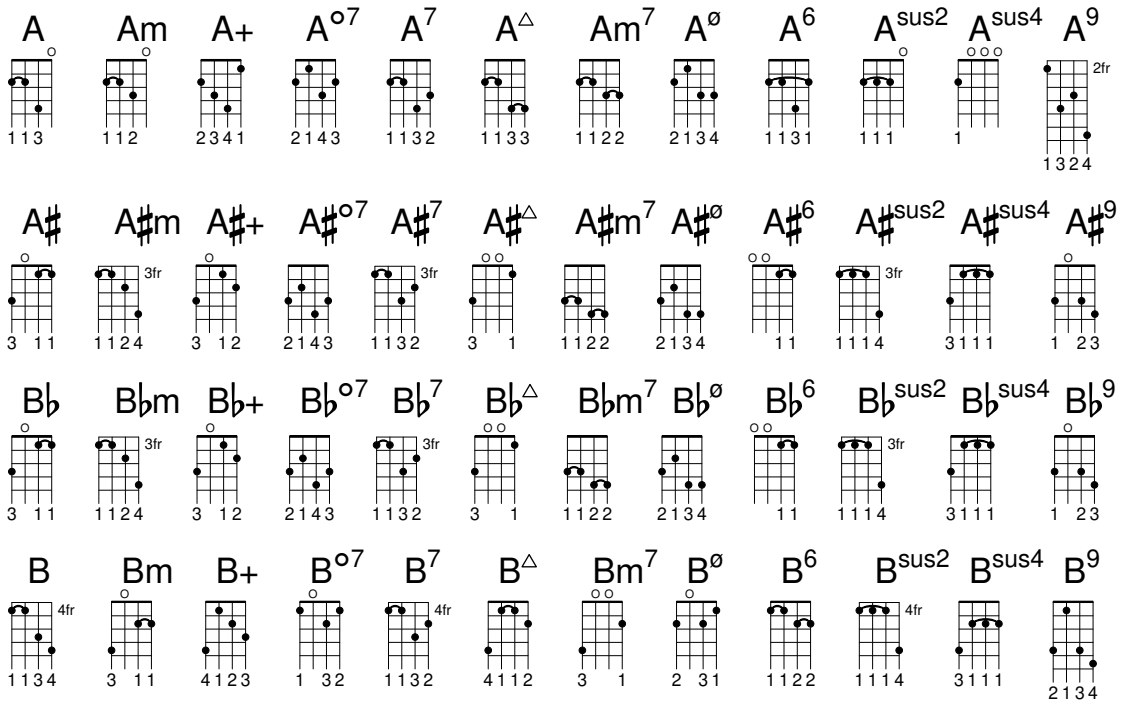




B.4.3 Diagramas para mandolina




D Dm D+ D°7 D7 D△ Dm7 Dø D6 Dsus2 Dsus4 D9
D♯ D♯m D♯+ D♯°7 D♯7 D♯△ D♯m7 D♯ø D♯6 D♯sus2 D♯sus4 D♯9
E♭ E♭m E♭+ E♭°7 E♭7 E♭△ E♭m7 E♭ø E♭6 E♭sus2 E♭sus4 E♭9
E Em E+ E°7 E7 E△ Em7 Eø E6 Esus2 Esus4 E9
F Fm F+ F°7 F7 F△ Fm7 Fø F6 Fsus2 Fsus4 F9
F♯ F♯m F♯+ F♯°7 F♯7 F♯△ F♯m7 F♯ø F♯6 F♯sus2 F♯sus4 F♯9
G♭ G♭m G♭+ G♭°7 G♭7 G♭△ G♭m7 G♭ø G♭6 G♭sus2 G♭sus4 G♭9
G Gm G+ G°7 G7 G△ Gm7 Gø G6 Gsus2 Gsus4 G9
G♯ G♯m G♯+ G♯°7 G♯7 G♯△ G♯m7 G♯ø G♯6 G♯sus2 G♯sus4 G♯9
A♭ A♭m A♭+ A♭°7 A♭7 A♭△ A♭m7 A♭ø A♭6 A♭sus2 A♭sus4 A♭9



B.5 Tamaños de página predefinidos

Los tamaños de página se definen en el archivo `scm/paper.scm`

ISO 216, A series

`"a10"`	26 mm x 37 mm (1.02 in x 1.46 in)
`"a9"`	37 mm x 52 mm (1.46 in x 2.05 in)
`"a8"`	52 mm x 74 mm (2.05 in x 2.91 in)
`"a7"`	74 mm x 105 mm (2.91 in x 4.13 in)
`"a6"`	105 mm x 148 mm (4.13 in x 5.83 in)
`"a5"`	148 mm x 210 mm (5.83 in x 8.27 in)
`"a4"`	210 mm x 297 mm (8.27 in x 11.69 in)
`"a3"`	297 mm x 420 mm (11.69 in x 16.54 in)
`"a2"`	420 mm x 594 mm (16.54 in x 23.39 in)
`"a1"`	594 mm x 841 mm (23.39 in x 33.11 in)
`"a0"`	841 mm x 1189 mm (33.11 in x 46.81 in)

Two extended sizes as defined in DIN 476

`"2a0"`	1189 mm x 1682 mm (46.81 in x 66.22 in)
`"4a0"`	1682 mm x 2378 mm (66.22 in x 93.62 in)

ISO 216, B series

`"b10"`	31 mm x 44 mm (1.22 in x 1.73 in)
`"b9"`	44 mm x 62 mm (1.73 in x 2.44 in)
`"b8"`	62 mm x 88 mm (2.44 in x 3.46 in)
`"b7"`	88 mm x 125 mm (3.46 in x 4.92 in)
`"b6"`	125 mm x 176 mm (4.92 in x 6.93 in)
`"b5"`	176 mm x 250 mm (6.93 in x 9.84 in)
`"b4"`	250 mm x 353 mm (9.84 in x 13.90 in)
`"b3"`	353 mm x 500 mm (13.90 in x 19.69 in)
`"b2"`	500 mm x 707 mm (19.69 in x 27.83 in)
`"b1"`	707 mm x 1000 mm (27.83 in x 39.37 in)
`"b0"`	1000 mm x 1414 mm (39.37 in x 55.67 in)

ISO 269, C series

`"c10"`	28 mm x 40 mm (1.10 in x 1.57 in)
`"c9"`	40 mm x 57 mm (1.57 in x 2.24 in)
`"c8"`	57 mm x 81 mm (2.24 in x 3.19 in)
`"c7"`	81 mm x 114 mm (3.19 in x 4.49 in)
`"c6"`	114 mm x 162 mm (4.49 in x 6.38 in)
`"c5"`	162 mm x 229 mm (6.38 in x 9.02 in)
`"c4"`	229 mm x 324 mm (9.02 in x 12.76 in)
`"c3"`	324 mm x 458 mm (12.76 in x 18.03 in)
`"c2"`	458 mm x 648 mm (18.03 in x 25.51 in)
`"c1"`	648 mm x 917 mm (25.51 in x 36.10 in)
`"c0"`	917 mm x 1297 mm (36.10 in x 51.06 in)

North American paper sizes

`"junior-legal"`	5.0 in x 8.0 in (127 mm x 203 mm)
`"legal"`	8.5 in x 14.0 in (216 mm x 356 mm)
`"ledger"`	17.0 in x 11.0 in (432 mm x 279 mm)
`"17x11"`	17.0 in x 11.0 in (432 mm x 279 mm)
`"letter"`	8.5 in x 11.0 in (216 mm x 279 mm)
`"tabloid"`	11.0 in x 17.0 in (279 mm x 432 mm)
`"11x17"`	11.0 in x 17.0 in (279 mm x 432 mm)

Sizes by IEEE Printer Working Group, for children's writing

`"government-letter"`	8.0 in x 10.5 in (203 mm x 267 mm)
`"government-legal"`	8.5 in x 13.0 in (216 mm x 330 mm)
`"philippine-legal"`	8.5 in x 13.0 in (216 mm x 330 mm)

ANSI sizes

`"ansi a"`	8.5 in x 11.0 in (216 mm x 279 mm)
`"ansi b"`	11.0 in x 17.0 in (279 mm x 432 mm)
`"ansi c"`	17.0 in x 22.0 in (432 mm x 559 mm)
`"ansi d"`	22.0 in x 34.0 in (559 mm x 864 mm)
`"ansi e"`	34.0 in x 44.0 in (864 mm x 1118 mm)
`"engineering f"`	28.0 in x 40.0 in (711 mm x 1016 mm)

North American architectural sizes

`"arch a"`	9.0 in x 12.0 in (229 mm x 305 mm)
`"arch b"`	12.0 in x 18.0 in (305 mm x 457 mm)
`"arch c"`	18.0 in x 24.0 in (457 mm x 610 mm)
`"arch d"`	24.0 in x 36.0 in (610 mm x 914 mm)
`"arch e"`	36.0 in x 48.0 in (914 mm x 1219 mm)
`"arch e1"`	30.0 in x 42.0 in (762 mm x 1067 mm)

Other sizes, including antique sizes still used in the United Kingdom

`"statement"`	5.5 in x 8.5 in (140 mm x 216 mm)
`"half letter"`	5.5 in x 8.5 in (140 mm x 216 mm)
`"quarto"`	8.0 in x 10.0 in (203 mm x 254 mm)
`"octavo"`	6.75 in x 10.5 in (171 mm x 267 mm)
`"executive"`	7.25 in x 10.5 in (184 mm x 267 mm)
`"monarch"`	7.25 in x 10.5 in (184 mm x 267 mm)
`"foolscap"`	8.27 in x 13.0 in (210 mm x 330 mm)
`"folio"`	8.27 in x 13.0 in (210 mm x 330 mm)
`"super-b"`	13.0 in x 19.0 in (330 mm x 483 mm)
`"post"`	15.5 in x 19.5 in (394 mm x 495 mm)

"crown"	15.0 in x 20.0 in (381 mm x 508 mm)
"large post"	16.5 in x 21.0 in (419 mm x 533 mm)
"demy"	17.5 in x 22.5 in (445 mm x 572 mm)
"medium"	18.0 in x 23.0 in (457 mm x 584 mm)
"broadsheet"	18.0 in x 24.0 in (457 mm x 610 mm)
"royal"	20.0 in x 25.0 in (508 mm x 635 mm)
"elephant"	23.0 in x 28.0 in (584 mm x 711 mm)
"double demy"	22.5 in x 35.0 in (572 mm x 889 mm)
"quad demy"	35.0 in x 45.0 in (889 mm x 1143 mm)
"atlas"	26.0 in x 34.0 in (660 mm x 864 mm)
"imperial"	22.0 in x 30.0 in (559 mm x 762 mm)
"antiquarian"	31.0 in x 53.0 in (787 mm x 1346 mm)

PA4-based sizes

"pa10"	26 mm x 35 mm (1.02 in x 1.38 in)
"pa9"	35 mm x 52 mm (1.38 in x 2.05 in)
"pa8"	52 mm x 70 mm (2.05 in x 2.76 in)
"pa7"	70 mm x 105 mm (2.76 in x 4.13 in)
"pa6"	105 mm x 140 mm (4.13 in x 5.51 in)
"pa5"	140 mm x 210 mm (5.51 in x 8.27 in)
"pa4"	210 mm x 280 mm (8.27 in x 11.02 in)
"pa3"	280 mm x 420 mm (11.02 in x 16.54 in)
"pa2"	420 mm x 560 mm (16.54 in x 22.05 in)
"pa1"	560 mm x 840 mm (22.05 in x 33.07 in)
"pa0"	840 mm x 1120 mm (33.07 in x 44.09 in)

Additional format for use in Southeast Asia and Australia

"f4"	210 mm x 330 mm (8.27 in x 12.99 in)

B.6 Instrumentos MIDI

A continuación figura una lista con los nombres que se pueden utilizar para la propiedad `midiInstrument` (instrumento MIDI). Cada una se presenta con su número de entre los 128 números de programa del estándar General MIDI.

1	acoustic grand	2	bright acoustic	3	electric grand
4	honky-tonk	5	electric piano 1	6	electric piano 2
7	harpsichord	8	clav	9	celesta
10	glockenspiel	11	music box	12	vibraphone
13	marimba	14	xylophone	15	tubular bells
16	dulcimer	17	drawbar organ	18	percussive organ
19	rock organ	20	church organ	21	reed organ
22	accordion	23	harmonica	24	concertina
25	acoustic guitar (nylon)	26	acoustic guitar (steel)	27	electric guitar (jazz)
28	electric guitar (clean)	29	electric guitar (muted)	30	overdriven guitar
31	distorted guitar	32	guitar harmonics	33	acoustic bass
34	electric bass (finger)	35	electric bass (pick)	36	fretless bass
37	slap bass 1	38	slap bass 2	39	synth bass 1
40	synth bass 2	41	violin	42	viola
43	cello	44	contrabass	45	tremolo strings
46	pizzicato strings	47	orchestral harp	48	timpani
49	string ensemble 1	50	string ensemble 2	51	synthstrings 1
52	synthstrings 2	53	choir aahs	54	voice oohs
55	synth voice	56	orchestra hit	57	trumpet
58	trombone	59	tuba	60	muted trumpet
61	french horn	62	brass section	63	synthbrass 1
64	synthbrass 2	65	soprano sax	66	alto sax
67	tenor sax	68	baritone sax	69	oboe
70	english horn	71	bassoon	72	clarinet

73	piccolo	74	flute	75	recorder
76	pan flute	77	blown bottle	78	shakuhachi
79	whistle	80	ocarina	81	lead 1 (square)
82	lead 2 (sawtooth)	83	lead 3 (calliope)	84	lead 4 (chiff)
85	lead 5 (charang)	86	lead 6 (voice)	87	lead 7 (fifths)
88	lead 8 (bass+lead)	89	pad 1 (new age)	90	pad 2 (warm)
91	pad 3 (polysynth)	92	pad 4 (choir)	93	pad 5 (bowed)
94	pad 6 (metallic)	95	pad 7 (halo)	96	pad 8 (sweep)
97	fx 1 (rain)	98	fx 2 (soundtrack)	99	fx 3 (crystal)
100	fx 4 (atmosphere)	101	fx 5 (brightness)	102	fx 6 (goblins)
103	fx 7 (echoes)	104	fx 8 (sci-fi)	105	sitar
106	banjo	107	shamisen	108	koto
109	kalimba	110	bagpipe	111	fiddle
112	shanai	113	tinkle bell	114	agogo
115	steel drums	116	woodblock	117	taiko drum
118	melodic tom	119	synth drum	120	reverse cymbal
121	guitar fret noise	122	breath noise	123	seashore
124	bird tweet	125	telephone ring	126	helicopter
127	applause	128	gunshot		

También son posibles los kits de percusión listados a continuación.

1	standard kit	9	room kit	17	power kit
25	electronic kit	26	tr-808 kit	33	jazz kit
41	brush kit	49	orchestra kit	57	sfx kit
128	mt-32 kit/cm-64 kit				

B.7 Lista de colores

Colores normales

Véase Sección 7.1.5 [Colorear objetos], página 292, para ver la sintaxis detallada.

black	white	red	green
blue	cyan	magenta	yellow
grey	darkred	darkgreen	darkblue
darkcyan	darkmagenta	darkyellow	

Nombre de los colores de CSS

Los nombres de colores de CSS se pueden usar tal cual dentro de argumentos de cadena de caracteres.

aliceblue	antiquewhite	aqua
aquamarine	azure	beige
bisque	black	blanchedalmond
blue	blueviolet	brown
burlywood	cadetblue	chartreuse
chocolate	coral	cornflowerblue
cornsilk	crimson	cyan
darkblue	darkcyan	darkgoldenrod
darkgray	darkgreen	darkgrey
darkkhaki	darkmagenta	darkolivegreen
darkorange	darkorchid	darkred
darksalmon	darkseagreen	darkslateblue
darkslategray	darkslategrey	darkturquoise
darkviolet	deeppink	deepskyblue
dimgray	dimgrey	dodgerblue
firebrick	floralwhite	forestgreen
fuchsia	gainsboro	ghostwhite

gold	goldenrod	gray
green	greenyellow	grey
honeydew	hotpink	indianred
indigo	ivory	khaki
lavender	lavenderblush	lawngreen
lemonchiffon	lightblue	lightcoral
lightcyan	lightgoldenrodyellow	lightgray
lightgreen	lightgrey	lightpink
lightsalmon	lightseagreen	lightskyblue
lightslategray	lightslategrey	lightsteelblue
lightyellow	lime	limegreen
linen	magenta	maroon
mediumaquamarine	mediumblue	mediumorchid
mediumpurple	mediumseagreen	mediumslateblue
mediumspringgreen	mediumturquoise	mediumvioletred
midnightblue	mintcream	mistyrose
moccasin	navajowhite	navy
oldlace	olive	olivedrab
orange	orangered	orchid
palegoldenrod	palegreen	paleturquoise
palevioletred	papayawhip	peachpuff
peru	pink	plum
powderblue	purple	rebeccapurple
red	rosybrown	royalblue
saddlebrown	salmon	sandybrown
seagreen	seashell	sienna
silver	skyblue	slateblue
slategray	slategrey	snow
springgreen	steelblue	tan
teal	thistle	tomato
turquoise	violet	wheat
white	whitesmoke	yellow
yellowgreen		

Las definiciones de colores de CSS difieren de los nombres de colores de X por los colres siguientes: `green`, `grey`, `maroon`, `purple`.

Nombres de los colores de X

Los nombres de los colores de X11 (`https://en.wikipedia.org/wiki/X11_color_names`) ofrecen un abanico de opciones más amplio que los nombres de CSS. Admiten distintas variantes:

- Un nombre que se escribe como una palabra única con mayúsculas intercaladas (p.ej. 'LightSlateBlue') se puede escribir también como palabras separadas por espacios, con o sin mayúsculas (p.ej. 'light slate blue').
- La palabra 'grey' siempre se puede escribir como 'gray' (p.ej. 'DarkSlateGray'), sin ninguna diferencia en el resultado.
- Algunos nombres admiten un sufijo numérico (p.ej. 'LightSalmon4').

Use la función de Scheme `x11-color` para acceder a ellos.

Nombres de los colores sin sufijo numérico

La tabla siguiente presenta todos los nombres de los colores que se pueden usar sin un sufijo numérico.

aliceblue
antiquewhite
aquamarine
azure
beige
bisque
black
blanchedalmond
blue
blueviolet
brown
burlywood
cadetblue
chartreuse
chocolate
coral
cornflowerblue
cornsilk
cyan
darkblue
darkcyan
darkgoldenrod
darkgray
darkgreen
darkgrey
darkkhaki
darkmagenta
darkolivegreen
darkorange
darkorchid
darkred
darksalmon
darkseagreen
darkslateblue
darkslategray
darkslategrey
darkturquoise
darkviolet
deeppink
deepskyblue
dimgray
dimgrey
dodgerblue
firebrick
floralwhite
forestgreen
gainsboro
ghostwhite
gold
goldenrod
gray
green
greenyellow
grey
honeydew
hotpink
indianred
ivory
khaki
lavender
lavenderblush
lawngreen
lemonchiffon
lightblue
lightcoral
lightcyan
lightgoldenrod
lightgoldenrodyellow
lightgray
lightgreen
lightgrey
lightpink
lightsalmon
lightseagreen
lightskyblue
lightslateblue
lightslategray
lightslategrey
lightsteelblue
lightyellow
limegreen
linen
magenta
maroon
mediumaquamarine
mediumblue
mediumorchid
mediumpurple
mediumseagreen
mediumslateblue
mediumspringgreen
mediumturquoise
mediumvioletred
midnightblue
mintcream
mistyrose
moccasin
navajowhite
navy
navyblue
oldlace
olivedrab
orange
orangered
orchid
palegoldenrod
palegreen
paleturquoise
palevioletred
papayawhip
peachpuff
peru
pink
plum
powderblue
purple
red
rosybrown
royalblue
saddlebrown
salmon
sandybrown
seagreen
seashell
sienna
skyblue
slateblue
slategray
slategrey
snow
springgreen
steelblue
tan
thistle
tomato
turquoise
violet
violetred
wheat
white
whitesmoke
yellow
yellowgreen

Nombres de colores con sufijo numérico

Se puede usar un subconjunto de los nombres de los colores de X11 con un sufijo numérico dentro del rango de 1 a 4 para dar como resultado tintes más oscuros. Oberve que para estos colores, el color `fulano` es idéntico al color `fulano1`.

antiquewhite1	antiquewhite2	antiquewhite3	antiquewhite4
aquamarine1	aquamarine2	aquamarine3	aquamarine4
azure1	azure2	azure3	azure4
bisque1	bisque2	bisque3	bisque4
blue1	blue2	blue3	blue4
brown1	brown2	brown3	brown4
burlywood1	burlywood2	burlywood3	burlywood4
cadetblue1	cadetblue2	cadetblue3	cadetblue4
chartreuse1	chartreuse2	chartreuse3	chartreuse4
chocolate1	chocolate2	chocolate3	chocolate4
coral1	coral2	coral3	coral4
cornsilk1	cornsilk2	cornsilk3	cornsilk4
cyan1	cyan2	cyan3	cyan4
darkgoldenrod1	darkgoldenrod2	darkgoldenrod3	darkgoldenrod4
darkolivegreen1	darkolivegreen2	darkolivegreen3	darkolivegreen4
darkorange1	darkorange2	darkorange3	darkorange4
darkorchid1	darkorchid2	darkorchid3	darkorchid4
darkseagreen1	darkseagreen2	darkseagreen3	darkseagreen4
darkslategray1	darkslategray2	darkslategray3	darkslategray4
deeppink1	deeppink2	deeppink3	deeppink4
deepskyblue1	deepskyblue2	deepskyblue3	deepskyblue4
dodgerblue1	dodgerblue2	dodgerblue3	dodgerblue4
firebrick1	firebrick2	firebrick3	firebrick4
gold1	gold2	gold3	gold4
goldenrod1	goldenrod2	goldenrod3	goldenrod4
green1	green2	green3	green4
honeydew1	honeydew2	honeydew3	honeydew4
hotpink1	hotpink2	hotpink3	hotpink4
indianred1	indianred2	indianred3	indianred4
ivory1	ivory2	ivory3	ivory4
khaki1	khaki2	khaki3	khaki4
lavenderblush1	lavenderblush2	lavenderblush3	lavenderblush4
lemonchiffon1	lemonchiffon2	lemonchiffon3	lemonchiffon4
lightblue1	lightblue2	lightblue3	lightblue4
lightcyan1	lightcyan2	lightcyan3	lightcyan4
lightgoldenrod1	lightgoldenrod2	lightgoldenrod3	lightgoldenrod4
lightpink1	lightpink2	lightpink3	lightpink4
lightsalmon1	lightsalmon2	lightsalmon3	lightsalmon4
lightskyblue1	lightskyblue2	lightskyblue3	lightskyblue4
lightsteelblue1	lightsteelblue2	lightsteelblue3	lightsteelblue4
lightyellow1	lightyellow2	lightyellow3	lightyellow4
magenta1	magenta2	magenta3	magenta4
maroon1	maroon2	maroon3	maroon4
mediumorchid1	mediumorchid2	mediumorchid3	mediumorchid4
mediumpurple1	mediumpurple2	mediumpurple3	mediumpurple4
mistyrose1	mistyrose2	mistyrose3	mistyrose4

navajowhite1	navajowhite2	navajowhite3	navajowhite4
olivedrab1	olivedrab2	olivedrab3	olivedrab4
orange1	orange2	orange3	orange4
orangered1	orangered2	orangered3	orangered4
orchid1	orchid2	orchid3	orchid4
palegreen1	palegreen2	palegreen3	palegreen4
paleturquoise1	paleturquoise2	paleturquoise3	paleturquoise4
palevioletred1	palevioletred2	palevioletred3	palevioletred4
peachpuff1	peachpuff2	peachpuff3	peachpuff4
pink1	pink2	pink3	pink4
plum1	plum2	plum3	plum4
purple1	purple2	purple3	purple4
red1	red2	red3	red4
rosybrown1	rosybrown2	rosybrown3	rosybrown4
royalblue1	royalblue2	royalblue3	royalblue4
salmon1	salmon2	salmon3	salmon4
seagreen1	seagreen2	seagreen3	seagreen4
seashell1	seashell2	seashell3	seashell4
sienna1	sienna2	sienna3	sienna4
skyblue1	skyblue2	skyblue3	skyblue4
slateblue1	slateblue2	slateblue3	slateblue4
slategray1	slategray2	slategray3	slategray4
snow1	snow2	snow3	snow4
springgreen1	springgreen2	springgreen3	springgreen4
steelblue1	steelblue2	steelblue3	steelblue4
tan1	tan2	tan3	tan4
thistle1	thistle2	thistle3	thistle4
tomato1	tomato2	tomato3	tomato4
turquoise1	turquoise2	turquoise3	turquoise4
violetred1	violetred2	violetred3	violetred4
wheat1	wheat2	wheat3	wheat4
yellow1	yellow2	yellow3	yellow4

Escala de grises

El conjunto de los colores de X11 también contiene una paleta de 101 tonos de gris.

gray0	gray1	gray2	gray3	gray4
gray5	gray6	gray7	gray8	gray9
gray10	gray11	gray12	gray13	gray14
gray15	gray16	gray17	gray18	gray19
gray20	gray21	gray22	gray23	gray24
gray25	gray26	gray27	gray28	gray29
gray30	gray31	gray32	gray33	gray34
gray35	gray36	gray37	gray38	gray39
gray40	gray41	gray42	gray43	gray44
gray45	gray46	gray47	gray48	gray49
gray50	gray51	gray52	gray53	gray54
gray55	gray56	gray57	gray58	gray59
gray60	gray61	gray62	gray63	gray64
gray65	gray66	gray67	gray68	gray69
gray70	gray71	gray72	gray73	gray74

gray75 gray76 gray77 gray78 gray79
gray80 gray81 gray82 gray83 gray84
gray85 gray86 gray87 gray88 gray89
gray90 gray91 gray92 gray93 gray94
gray95 gray96 gray97 gray98 gray99
gray100

La tabla de los nombres de los colores `grey`*N* es similar.

Colores a prueba de ceguera al color

La función de Scheme `universal-color` proporciona un conjunto de ocho colores (`https://jfly.uni-koeln.de/color`) que están pensados para que no presenten ambigüedad para las personas con dicromatismo.

black orange skyblue bluegreen
yellow blue vermillion redpurple

B.8 La tipografía Emmentaler

La fuente tipográfica Emmentaler consiste en dos *subconjuntos* de glifos. "Feta", utilizado para la notación clásica, y "Parmesan", usado para la notación de la música antigua.

Cualquier glifo que esté comprendido dentro de la tipografía Emmentaler se puede acceder directamente usando elementos de marcado de texto junto con el nombre del glifo (según se muestra en las tablas, más adelante). Por ejemplo:

```
g^\markup {\musicglyph "scripts.segno" }
```

o

```
\markup {\musicglyph "five"}
```

Véase Sección 8.2 [Formatear el texto], página 323, para ver más información.

B.8.1 Tablas de glifos modernos

Todos los glifos del subconjunto "Feta".

Glifos de clave

`clefs.C`		`clefs.C_change`	
`clefs.varC`		`clefs.varC_change`	
`clefs.F`		`clefs.F_change`	
`clefs.G`		`clefs.G_change`	
`clefs.GG`		`clefs.GG_change`	
`clefs.tenorG`		`clefs.tenorG_change`	
`clefs.percussion`		`clefs.percussion_change`	
`clefs.varpercussion`		`clefs.varpercussion_change`	
`clefs.tab`		`clefs.tab_change`	

Glifos de indicación de compás

timesig.C44	𝐂	timesig.C22	𝐂

Glifos de cifras

plus	+	comma	,
hyphen	-	period	.
figuredash	‒	endash	–
parenleft	(	parenright	)
slash	/	zero	0
one	1	two	2
three	3	four	4
four.alt	4	five	5
six	6	seven	7
seven.alt	7	eight	8
nine	9	fixedwidth.zero	0
fixedwidth.one	1	fixedwidth.two	2
fixedwidth.three	3	fixedwidth.four	4
fixedwidth.four.alt	4	fixedwidth.five	5
fixedwidth.six	6	fixedwidth.seven	7
fixedwidth.seven.alt	7	fixedwidth.eight	8
fixedwidth.nine	9	fattened.zero	0
fattened.one	1	fattened.two	2
fattened.three	3	fattened.four	4
fattened.four.alt	4	fattened.five	5
fattened.six	6	fattened.seven	7
fattened.seven.alt	7	fattened.eight	8
fattened.nine	9	fattened.fixedwidth.zero	0
fattened.fixedwidth.one	1	fattened.fixedwidth.two	2

fattened.fixedwidth.three 
fattened.fixedwidth.four 
fattened.fixedwidth.four.alt 
fattened.fixedwidth.five 
fattened.fixedwidth.six 
fattened.fixedwidth.seven 
fattened.fixedwidth.seven.alt 
fattened.fixedwidth.eight 
fattened.fixedwidth.nine 
u2007
u2009
u200A

Figured bass symbol glyphs

figbass.twoplus 
figbass.fourplus 
figbass.fiveplus 
figbass.sixstroked 
figbass.sevenstroked 
figbass.ninestroked 

Glifos de alteraciones

accidentals.sharp 
accidentals.sharp.figbass 
accidentals.sharp.arrowup 
accidentals.sharp.arrowdown 
accidentals.sharp.arrowboth 
accidentals.sharp.slashslash.stem 
accidentals.sharp.slashslashslash.stemstem 
accidentals.sharp.slashslashslash.stem
accidentals.sharp.slash.stem 
accidentals.sharp.slashslash.stemstemstem 
accidentals.doublesharp
accidentals.doublesharp.figbass 
accidentals.natural 
accidentals.natural.figbass 
accidentals.natural.arrowup 
accidentals.natural.arrowdown 
accidentals.natural.arrowboth 
accidentals.flat 
accidentals.flat.figbass 
accidentals.flat.arrowup 
accidentals.flat.arrowdown
accidentals.flat.arrowboth 
accidentals.flat.slash 
accidentals.flat.slashslash 
accidentals.mirroredflat.flat 
accidentals.mirroredflat
accidentals.mirroredflat.backslash
accidentals.flatflat
accidentals.flatflat.figbass
accidentals.flatflat.slash

accidentals.sharp.sori		accidentals.flat.koron	
accidentals.rightparen	)	accidentals.leftparen	(

Glifos de las cabezas de nota predeterminadas

noteheads.uM2		noteheads.dM2	
noteheads.sM1		noteheads.s0	
noteheads.s1		noteheads.s2	

Glifos de las cabezas de nota especiales

noteheads.sM1double		noteheads.s0diamond	
noteheads.s1diamond		noteheads.s2diamond	
noteheads.s0triangle		noteheads.s1triangle	
noteheads.s2triangle		noteheads.s0slash	
noteheads.s1slash		noteheads.s2slash	
noteheads.s0cross		noteheads.s1cross	
noteheads.s2cross		noteheads.s2xcircle	
noteheads.d0arrow		noteheads.u0arrow	
noteheads.d2arrow		noteheads.u2arrow	
noteheads.s0harmonic		noteheads.s2harmonic	

Glifos de las cabezas de nota con formas

noteheads.s0do		noteheads.s1do	
noteheads.s2do		noteheads.s0doThin	
noteheads.s1doThin		noteheads.s2doThin	
noteheads.s0re		noteheads.s1re	
noteheads.s2re		noteheads.s0reThin	
noteheads.s1reThin		noteheads.s2reThin	
noteheads.s0mi		noteheads.s1mi	
noteheads.s2mi		noteheads.s0miMirror	

noteheads.s1miMirror		noteheads.s2miMirror	
noteheads.s0miThin		noteheads.s1miThin	
noteheads.s2miThin		noteheads.u0fa	
noteheads.d0fa		noteheads.u1fa	
noteheads.d1fa		noteheads.u2fa	
noteheads.d2fa		noteheads.u0faThin	
noteheads.d0faThin		noteheads.u1faThin	
noteheads.d1faThin		noteheads.u2faThin	
noteheads.d2faThin		noteheads.s0sol	
noteheads.s1sol		noteheads.s2sol	
noteheads.s0la		noteheads.s1la	
noteheads.s2la		noteheads.s0laThin	
noteheads.s1laThin		noteheads.s2laThin	
noteheads.s0ti		noteheads.s1ti	
noteheads.s2ti		noteheads.s0tiThin	
noteheads.s1tiThin		noteheads.s2tiThin	
noteheads.u0doFunk		noteheads.d0doFunk	
noteheads.u1doFunk		noteheads.d1doFunk	
noteheads.u2doFunk		noteheads.d2doFunk	
noteheads.u0reFunk		noteheads.d0reFunk	
noteheads.u1reFunk		noteheads.d1reFunk	
noteheads.u2reFunk		noteheads.d2reFunk	
noteheads.u0miFunk		noteheads.d0miFunk	
noteheads.u1miFunk		noteheads.d1miFunk	
noteheads.s2miFunk		noteheads.u0faFunk	
noteheads.d0faFunk		noteheads.u1faFunk	
noteheads.d1faFunk		noteheads.u2faFunk	

noteheads.d2faFunk		noteheads.s0solFunk	
noteheads.s1solFunk		noteheads.s2solFunk	
noteheads.s0laFunk		noteheads.s1laFunk	
noteheads.s2laFunk		noteheads.u0tiFunk	
noteheads.d0tiFunk		noteheads.u1tiFunk	
noteheads.d1tiFunk		noteheads.u2tiFunk	
noteheads.d2tiFunk		noteheads.s0doWalker	
noteheads.u1doWalker		noteheads.d1doWalker	
noteheads.u2doWalker		noteheads.d2doWalker	
noteheads.s0reWalker		noteheads.u1reWalker	
noteheads.d1reWalker		noteheads.u2reWalker	
noteheads.d2reWalker		noteheads.s0miWalker	
noteheads.s1miWalker		noteheads.s2miWalker	
noteheads.s0faWalker		noteheads.u1faWalker	
noteheads.d1faWalker		noteheads.u2faWalker	
noteheads.d2faWalker		noteheads.s0laWalker	
noteheads.s1laWalker		noteheads.s2laWalker	
noteheads.s0tiWalker		noteheads.u1tiWalker	
noteheads.d1tiWalker		noteheads.u2tiWalker	
noteheads.d2tiWalker			

Glifos de silencios

rests.0		rests.1	
rests.0o		rests.1o	
rests.M3		rests.M2	
rests.M1		rests.M1o	
rests.2		rests.2classical	

rests.2z



rests.3



rests.4



rests.5



rests.6



rests.7



rests.8



rests.9



rests.10



Glifos de corchetes

flags.u3



flags.u4



flags.u5



flags.u6



flags.u7



flags.u8



flags.u9



flags.u10



flags.d3



flags.d4



flags.d5



flags.d6



flags.d7



flags.d8



flags.d9



flags.d10



flags.stackedu3



flags.stackedu4



flags.stackedu5



flags.stackedu6



flags.stackedu7



flags.stackedu8



flags.stackedu9



flags.stackedu10



flags.stackedd3		flags.stackedd4	
flags.stackedd5		flags.stackedd6	
flags.stackedd7		flags.stackedd8	
flags.stackedd9		flags.stackedd10	
flags.ugrace		flags.dgrace	

Glifos de puntillos

dots.dot	

Glifos de matices dinámicos

space		f	
m		n	
p		r	
s		z	

Glifos de inscripciones

scripts.ufermata		scripts.dfermata	
scripts .uhenzeshortfermata		scripts .dhenzeshortfermata	
scripts .uhenzelongfermata		scripts .dhenzelongfermata	
scripts.ushortfermata		scripts.dshortfermata	
scripts .uveryshortfermata		scripts .dveryshortfermata	
scripts.ulongfermata		scripts.dlongfermata	
scripts.uverylongfermata		scripts.dverylongfermata	
scripts.thumb		scripts.sforzato	
scripts.espr		scripts.staccato	
scripts.ustaccatissimo		scripts.dstaccatissimo	
scripts.tenuto		scripts.uportato	

scripts.dportato		scripts.umarcato	
scripts.dmarcato		scripts.open	
scripts.halfopen		scripts.halfopenvertical	
scripts.stopped		scripts.uupbow	
scripts.dupbow		scripts.udownbow	
scripts.ddownbow		scripts.reverseturn	
scripts.turn		scripts.slashturn	
scripts.haydnturn		scripts.trill	
scripts.upedalheel		scripts.dpedalheel	
scripts.upedaltoe		scripts.dpedaltoe	
scripts.pedalheelcircle		scripts.flageolet	
scripts.segno		scripts.varsegno	
scripts.coda		scripts.varcoda	
scripts.rcomma		scripts.lcomma	
scripts.rvarcomma		scripts.lvarcomma	
scripts.raltcomma		scripts.laltcomma	
scripts.arpeggio		scripts.trill_element	
scripts.arpeggio .arrow.M1		scripts.arpeggio.arrow.1	
scripts.prall		scripts.mordent	
scripts.prallprall		scripts.prallmordent	
scripts.upprall		scripts.upmordent	
scripts.prallup		scripts.downprall	
scripts.downmordent		scripts.pralldown	
scripts.lineprall		scripts.bachschleifer	
scripts.caesura.curved		scripts.caesura.straight	
scripts.tickmark		scripts.snappizzicato	

scripts.ictus		scripts.uaccentus	
scripts.daccentus		scripts.usemicirculus	
scripts.dsemicirculus		scripts.circulus	
scripts .usignumcongruentiae		scripts .dsignumcongruentiae	

Glifos de flechas

arrowheads.open.01		arrowheads.open.0M1	
arrowheads.open.11		arrowheads.open.1M1	
arrowheads.close.01		arrowheads.close.0M1	
arrowheads.close.11		arrowheads.close.1M1	

Glifos de puntas de corchete

brackettips.up		brackettips.down	

Glifos de pedal

pedal.*		pedal.M	
pedal..		pedal.P	
pedal.d		pedal.e	
pedal.Ped			

Glifos de acordeón

accordion.discant		accordion.dot	
accordion.freebass		accordion.stdbass	
accordion.bayanbass		accordion.oldEE	
accordion.push		accordion.pull	

Glifos de ligadura

ties.lyric.short		ties.lyric.default	

B.8.2 Tablas de glifos antiguos

All glyphs of the “Parmesan” subset.

Glifos de vaticana

clefs.vaticana.do

clefs.vaticana.do_change

clefs.vaticana.fa 

clefs.vaticana.fa_change 

custodes.vaticana.u0

custodes.vaticana.u1

custodes.vaticana.u2

custodes.vaticana.d0

custodes.vaticana.d1

custodes.vaticana.d2

accidentals.vaticanaM1

accidentals.vaticana0

dots.dotvaticana

noteheads .svaticana.punctum

noteheads.svaticana .punctum.cavum

noteheads.svaticana .linea.punctum

noteheads.svaticana .linea.punctum.cavum

noteheads.svaticana .inclinatum

noteheads.svaticana.lpes

noteheads .svaticana.vlpes

noteheads.svaticana.upes

noteheads .svaticana.vupes

noteheads .svaticana.plica

noteheads .svaticana.vplica

noteheads .svaticana.epiphonus

noteheads.svaticana .vepiphonus

noteheads.svaticana .reverse.plica

noteheads.svaticana .reverse.vplica

noteheads.svaticana .inner.cephalicus

noteheads.svaticana .cephalicus

noteheads .svaticana.quilisma

Glifos de medicea

clefs.medicaea.do 

clefs.medicaea.do_change 

clefs.medicaea.fa 

clefs.medicaea.fa_change 

custodes.medicaea.u0

custodes.medicaea.u1

custodes.medicaea.u2

custodes.medicaea.d0

custodes.medicaea.d1

custodes.medicaea.d2

accidentals.medicaeaM1

noteheads.smedicaea .inclinatum

noteheads .smedicaea.punctum

noteheads .smedicaea.rvirga

noteheads .smedicaea.virga

Glifos de Hufnagel

clefs.hufnagel.do 

clefs.hufnagel.do_change 

clefs.hufnagel.fa 

clefs.hufnagel.fa_change 

clefs.hufnagel.do.fa 

clefs.hufnagel .do.fa_change 

custodes.hufnagel.u0 

custodes.hufnagel.u1 

custodes.hufnagel.u2 

custodes.hufnagel.d0 

custodes.hufnagel.d1 

custodes.hufnagel.d2 

accidentals.hufnagelM1 

noteheads .shufnagel.punctum

noteheads .shufnagel.virga 

noteheads.shufnagel.lpes

Glifos de mensural

rests.M3mensural

rests.M2mensural

rests.M1mensural

rests.0mensural

rests.1mensural

rests.2mensural

rests.3mensural

rests.4mensural

clefs.mensural.c 

clefs.mensural.c_change 

clefs.blackmensural.c 

clefs.blackmensural .c_change 

clefs.mensural.f 

clefs.mensural.f_change 

clefs.mensural.g 

clefs.mensural.g_change 

custodes.mensural.u0 

custodes.mensural.u1

custodes.mensural.u2

custodes.mensural.d0

custodes.mensural.d1

custodes.mensural.d2

accidentals.mensural1

accidentals.mensuralM1

flags.mensuralu03

flags.mensuralu13

flags.mensuralu23

flags.mensurald03

flags.mensurald13

flags.mensurald23

flags.mensuralu04

flags.mensuralu14

flags.mensuralu24

flags.mensurald14

flags.mensuralu05

flags.mensuralu25

flags.mensurald15

flags.mensuralu06

flags.mensuralu26

flags.mensurald16

timesig.mensural44

timesig.mensural32

timesig.mensural94

timesig.mensural68

timesig.mensural48

timesig.mensural24

noteheads.dM3mensural

noteheads.uM2mensural

noteheads.sM2ligmensural

noteheads.urM3mensural

noteheads
.srM3ligmensural

noteheads.drM2mensural

noteheads.srM1mensural

noteheads
.dM3semimensural

noteheads
.uM2semimensural

noteheads
.sM2semiligmensural

flags.mensurald04

flags.mensurald24

flags.mensuralu15

flags.mensurald05

flags.mensurald25

flags.mensuralu16

flags.mensurald06

flags.mensurald26

timesig.mensural22

timesig.mensural64

timesig.mensural34

timesig.mensural98

timesig.mensural68alt

noteheads.uM3mensural

noteheads.sM3ligmensural

noteheads.dM2mensural

noteheads.sM1mensural

noteheads.drM3mensural

noteheads.urM2mensural

noteheads
.srM2ligmensural

noteheads
.uM3semimensural

noteheads
.sM3semiligmensural

noteheads
.dM2semimensural

noteheads
.sM1semimensural

noteheads .urM3semimensural		noteheads .drM3semimensural	
noteheads .srM3semiligmensural		noteheads .urM2semimensural	
noteheads .drM2semimensural		noteheads .srM2semiligmensural	
noteheads .srM1semimensural		noteheads .uM3blackmensural	
noteheads .dM3blackmensural		noteheads .sM3blackligmensural	
noteheads .uM2blackmensural		noteheads .dM2blackmensural	
noteheads .sM2blackligmensural		noteheads .sM1blackmensural	
noteheads.s0mensural		noteheads.s1mensural	
noteheads.s2mensural		noteheads .s0blackmensural	

Glifos de neomensural

rests.M3neomensural		rests.M2neomensural	
rests.M1neomensural		rests.0neomensural	
rests.1neomensural		rests.2neomensural	
rests.3neomensural		rests.4neomensural	
clefs.neomensural.c		clefs.neomensural .c_change	
timesig.neomensural44		timesig.neomensural22	
timesig.neomensural32		timesig.neomensural64	
timesig.neomensural94		timesig.neomensural34	
timesig.neomensural68		timesig.neomensural98	
timesig.neomensural48		timesig.neomensural68alt	
timesig.neomensural24		noteheads.uM3neomensural	
noteheads.dM3neomensural		noteheads.uM2neomensural	
noteheads.dM2neomensural		noteheads.sM1neomensural	
noteheads .urM3neomensural		noteheads .drM3neomensural	
noteheads .urM2neomensural		noteheads .drM2neomensural	
noteheads .srM1neomensural		noteheads.s0neomensural	
noteheads.s1neomensural		noteheads.s2neomensural	

Glifos de Petrucci

clefs.petrucci.c1

clefs.petrucci.c1_change

clefs.petrucci.c2

clefs.petrucci.c2_change

clefs.petrucci.c3

clefs.petrucci.c3_change

clefs.petrucci.c4

clefs.petrucci.c4_change

clefs.petrucci.c5

clefs.petrucci.c5_change

clefs.petrucci.f

clefs.petrucci.f_change

clefs.petrucci.g

clefs.petrucci.g_change

noteheads.s0petrucci

noteheads.s1petrucci

noteheads.s2petrucci

noteheads.s0blackpetrucci

noteheads.s1blackpetrucci

noteheads.s2blackpetrucci

Glifos de Solesmes

noteheads.ssolesmes.incl.parvum

noteheads.ssolesmes.auct.asc

noteheads.ssolesmes.auct.desc

noteheads.ssolesmes.incl.auctum

noteheads.ssolesmes.stropha

noteheads.ssolesmes.stropha.aucta

noteheads.ssolesmes.oriscus

Glifos de la notación del canto kievano

clefs.kievan.do

clefs.kievan.do_change

accidentals.kievan1

accidentals.kievanM1

scripts.barline.kievan

dots.dotkievan

noteheads.sM2kievan

noteheads.sM1kievan

noteheads.s0kievan

noteheads.d2kievan

noteheads.u2kievan

noteheads.s1kievan

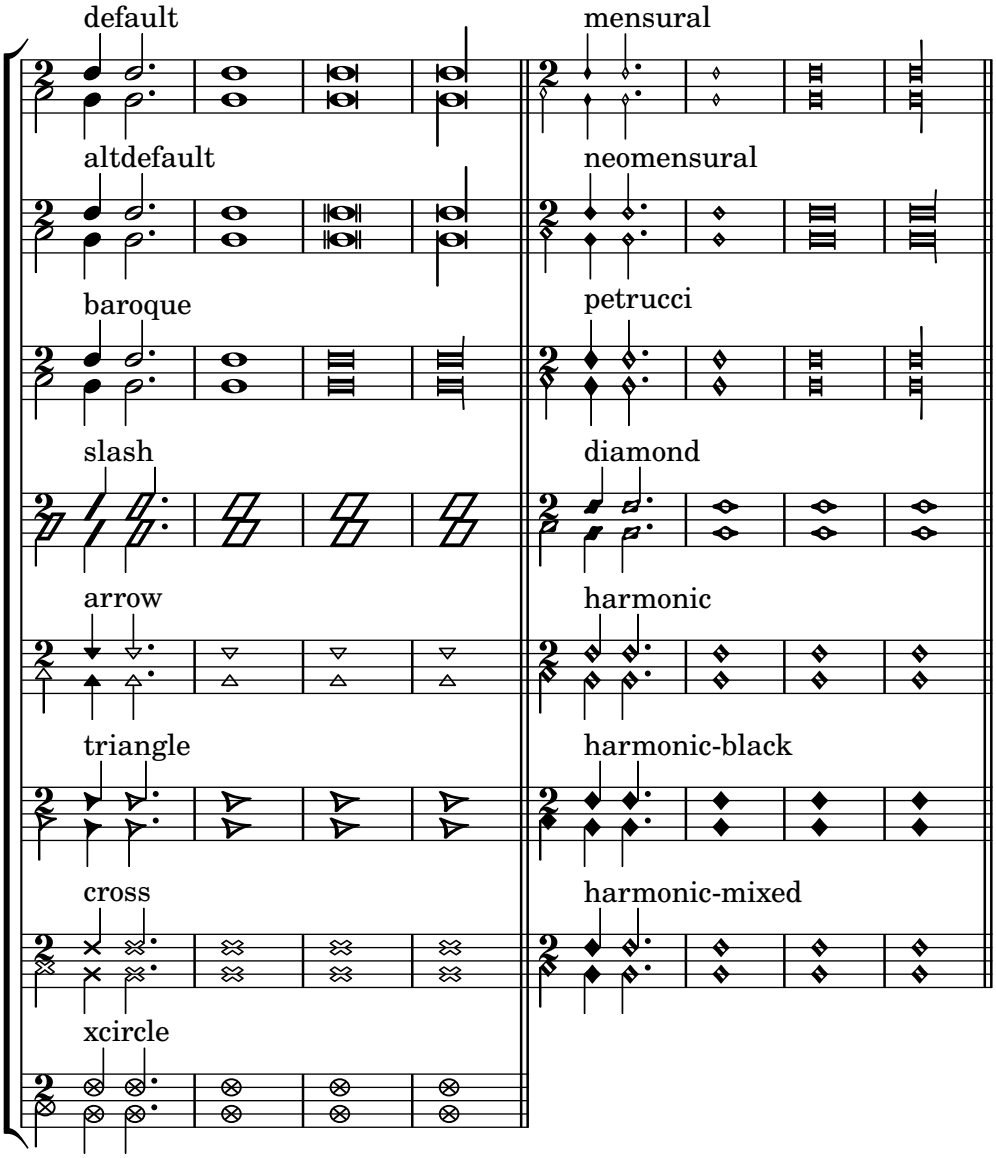
noteheads.sr1kievan

noteheads.d3kievan

noteheads.u3kievan

B.9 Estilos de cabezas de nota

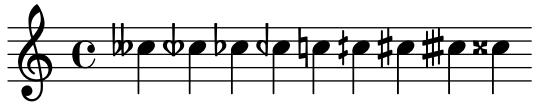
Se pueden usar los siguientes estilos para las cabezas de las notas.



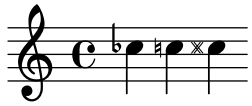
B.10 Conjuntos de glifos de alteraciones accidentales

Están disponibles los siguientes conjuntos de glifos para alteraciones accidentales.

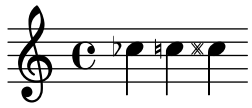
`standard-alteration-glyph-name-alist`



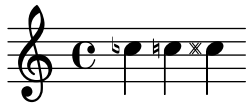
`alteration-hufnagel-glyph-name-alist`



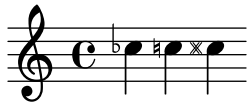
`alteration-medicaea-glyph-name-alist`



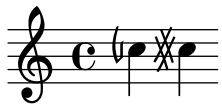
`alteration-vaticana-glyph-name-alist`



`alteration-mensural-glyph-name-alist`



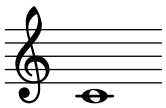
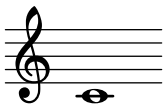
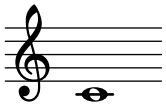
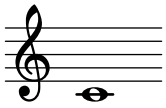
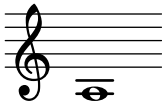
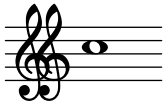
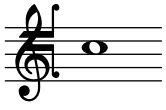
`alteration-kievan-glyph-name-alist`

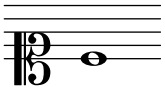


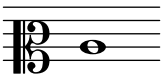
B.11 Estilos de clave

La tabla siguiente presenta todos los estilos de clave posibles (incluidos los casos en que el *Do central* figura como relativo a la clave). Para ver más posibilidades de modificación como las cifras de cambio de octava, véase Sección 1.3.1 [Clave], página 20.

B.11.1 Claves estándar

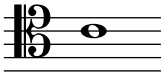
Ejemplo	**Salida**	**Ejemplo**	**Salida**
`\clef G`		`\clef "G2"`	
`\clef treble`		`\clef violin`	
`\clef french`		`\clef GG`	
`\clef tenorG`			

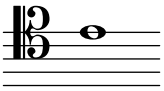
```
\clef soprano
```


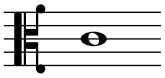
```
\clef mezzosoprano
```


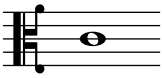
```
\clef C
```

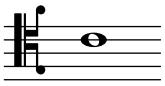

```
\clef alto
```

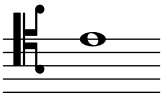

```
\clef tenor
```


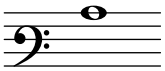
```
\clef baritone
```


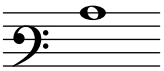
```
\clef varC
```


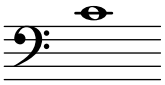
```
\clef altovarC
```


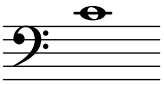
```
\clef tenorvarC
```


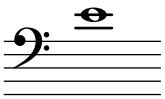
```
\clef baritonevarC
```


```
\clef varbaritone
```


```
\clef baritonevarF
```


```
\clef F
```


```
\clef bass
```


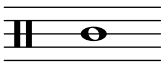
```
\clef subbass
```


B.11.2 Clave de pentagramas de percusión

Ejemplo

```
\clef percussion
```

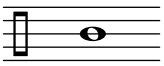
Salida



Ejemplo

```
\clef varpercussion
```

Salida

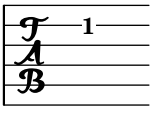


B.11.3 Claves de pautas de tablatura

Ejemplo

```
\new TabStaff {
  \clef tab
}
```

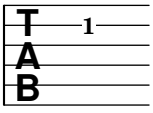
Salida



Ejemplo

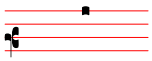
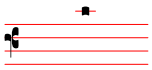
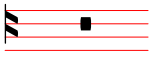
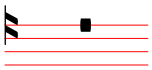
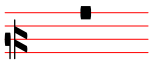
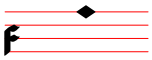
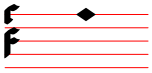
```
\new TabStaff {
  \clef moderntab
}
```

Salida

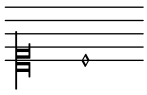
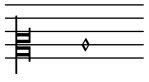
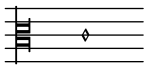
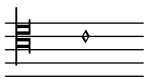
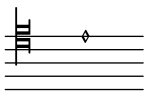


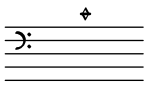
B.11.4 Claves de la música antigua

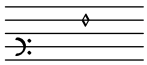
Gregorian

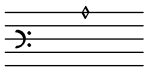
Ejemplo	Salida	Ejemplo	Salida
`\clef "vaticana-do1"`		`\clef "vaticana-do2"`	
`\clef "vaticana-do3"`		`\clef "vaticana-fa1"`	
`\clef "vaticana-fa2"`			
`\clef "medicaea-do1"`		`\clef "medicaea-do2"`	
`\clef "medicaea-do3"`		`\clef "medicaea-fa1"`	
`\clef "medicaea-fa2"`			
`\clef "hufnagel-do1"`		`\clef "hufnagel-do2"`	
`\clef "hufnagel-do3"`		`\clef "hufnagel-fa1"`	
`\clef "hufnagel-fa2"`		`\clef "hufnagel-do-fa"`	

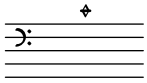
Mensural

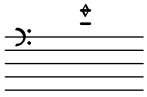
Ejemplo	Salida	Ejemplo	Salida
`\clef "mensural-c1"`		`\clef "mensural-c2"`	
`\clef "mensural-c3"`		`\clef "mensural-c4"`	
`\clef "mensural-c5"`			

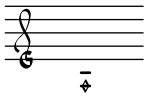
\clef "mensural-f" 

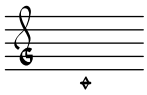
\clef "mensural-f2" 

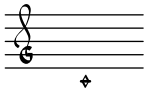
\clef "mensural-f3" 

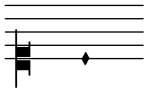
\clef "mensural-f4" 

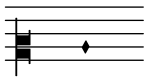
\clef "mensural-f5" 

\clef "mensural-g1" 

\clef "mensural-g2" 

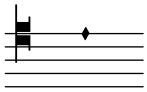
\clef "mensural-g" 

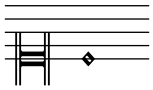
\clef "blackmensural-c1" 

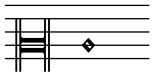
\clef "blackmensural-c2" 

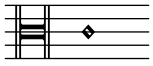
\clef "blackmensural-c3" 

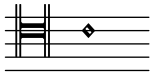
\clef "blackmensural-c4" 

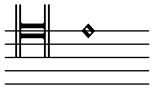
\clef "blackmensural-c5" 

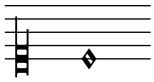
\clef "neomensural-c1" 

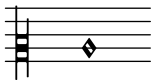
\clef "neomensural-c2" 

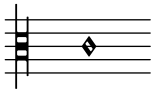
\clef "neomensural-c3" 

\clef "neomensural-c4" 

\clef "neomensural-c5" 

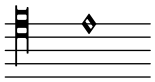
\clef "petrucci-c1" 

\clef "petrucci-c2" 

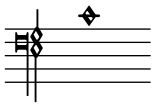
\clef "petrucci-c3" 

\clef "petrucci-c4" 

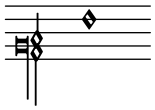
\clef "petrucci-c5"



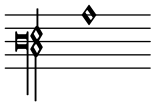
\clef "petrucci-f"



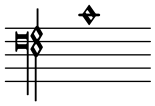
\clef "petrucci-f2"



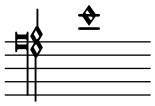
\clef "petrucci-f3"



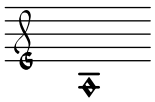
\clef "petrucci-f4"



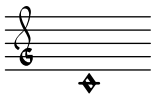
\clef "petrucci-f5"



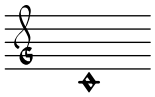
\clef "petrucci-g1"



\clef "petrucci-g2"



\clef "petrucci-g"



Kievan

Ejemplo **Salida**

\clef "kievan-do"



B.12 Lista de caracteres especiales

Se pueden usar las siguientes referencias a carcteres especiales; para ver más detalles, véase Sección 22.4.3 [Alias de ASCII], página 648.

Se usa la sintaxis del HTML y casi todas estas referencias son las mismas que en el HTML. El resto están inspiradas en LaTeX.

Los caracteres están enmarcados en un rectángulo de forma que pueda verse el tamaño que tienen. Se ha añadido un pequeño relleno de separación entre el carácter y el rectángulo para mejorar la legibilidad.

¡	¡	¿	¿	&solidus;	⁄	&flq;	‹		
&frq;	›	&flqq;	«	&frqq;	»	&glq;	‚		
&grq;	‘	&glqq;	„	&grqq;	“	&elq;	‘		
&erq;	’	&elqq;	“	&erqq;	”				
						&nnbsp;			

‍		‌		·	·	•	•		
&copyright;	©	&registered;	®	&trademark;	™	†	†		
‡	‡	№	№	ª	ª	º	º		
¶	¶	§	§	°	°	№	№		
‰	‰	¦	¦	´	´	&acutedbl;	˝		
`	`	˘	˘	ˇ	ˇ	&cedilla;	¸		
&circumflex;	^	&diaeresis;	¨	&macron;	¯	&aa;	å		
&AA;	Å	&ae;	æ	&AE;	Æ	ä	ä		
Ä	Ä	&dh;	ð	&DH;	Ð	&dj;	đ		
&DJ;	Đ	&l;	ł	&L;	Ł	&ng;	ŋ		
&NG;	Ŋ	&o;	ø	&O;	Ø	&oe;	œ		
&OE;	Œ	ö	ö	Ö	Ö	&s;	ſ		
&ss;	ß	&th;	þ	&TH;	Þ	ü	ü		
Ü	Ü	+	+	−	−	×	×		
÷	÷	¹	¹	²	²	³	³		
&sqrt;	√	&increment;	Δ	&infty;	∞	∑	∑		
±	±	&bulletop;	∙	&partial;	∂	&neg;	¬		
&currency;	¤	$	$	€	€	&pounds;	£		
¥	¥	¢	¢						

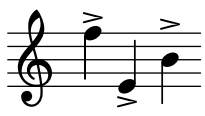
B.13 Lista de articulaciones

En la lógica interna de LilyPond, una 'articulación' es cualquier objeto (que no sea una indicación de matiz dinámico) que puede ser adosado directamente después de un evento rítmico (o con una duración): notas, acordes; incluso silencios y saltos, o el artificio de acorde vacío <> (véase Sección "Estructura de un elemento de nota" en *Manual de Aprendizaje*). Incluso las ligaduras de expresión, las digitaciones y las inscripciones textuales son técnicamente articulaciones, aunque estas no se muestran aquí.

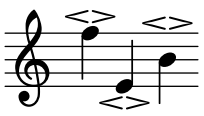
Por tanto, las listas que aparecen a continuación incluyen no solo las marcas de articulación, sino también todos los demás elementos textuales dentro de la tipografía Emmentaler que pueden ser adosados a notas (la forma en que se inserta un acento como '`c'\accent`' o como '`c'->`'). Cada ejemplo muestra la inscripción en sus dos posibles posiciones verticales: respectivamente, *superior* e *inferior*, así como su posición por defecto (*neutra*). Consulte también [Script glyphs], página 919, para ver una lista más completa de glifos, para su utilización con la instrucción de marcado `\musicglyph` tal y como se explica en Sección 8.2.5 [Notación musical dentro de elementos de marcado], página 338.

B.13.1 Indicaciones de articulación

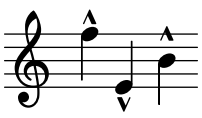
`\accent` o `->`



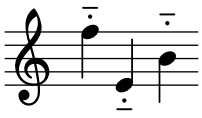
`\espressivo`



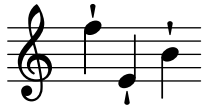
`\marcato` o `-^`



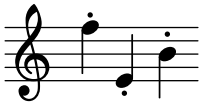
`\portato` o `-_`



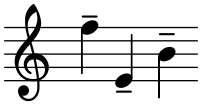
`\staccatissimo` o `-!`



`\staccato` o `-.`

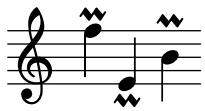


`\tenuto` o `--`

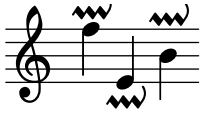


B.13.2 Indicaciones de adornos

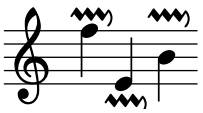
`\prall`



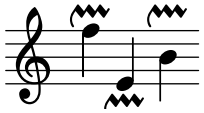
`\prallup`



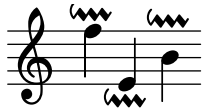
`\pralldown`



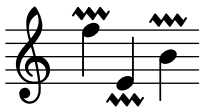
`\upprall`



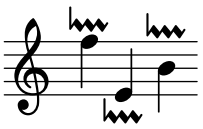
`\downprall`



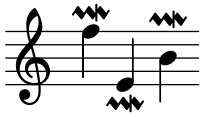
`\prallprall`



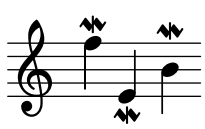
`\lineprall`



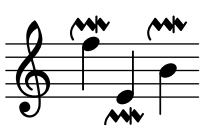
`\prallmordent`



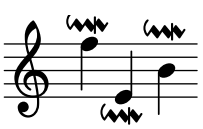
`\mordent`



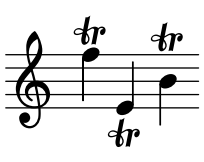
`\upmordent`



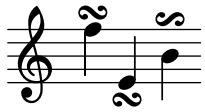
`\downmordent`



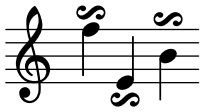
`\trill`



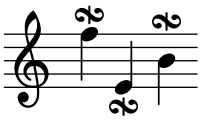
\turn



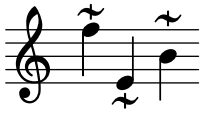
\reverseturn



\slashturn

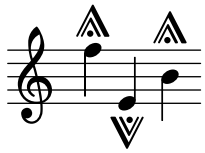


\haydnturn

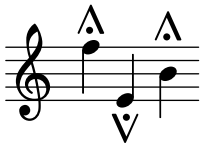


B.13.3 Indicaciones de calderones

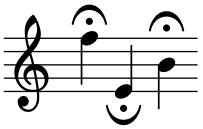
\veryshortfermata



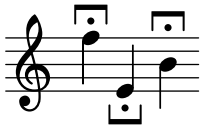
\shortfermata



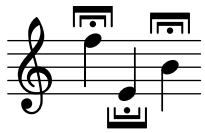
\fermata



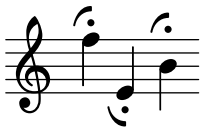
\longfermata



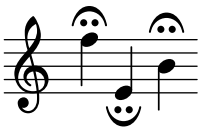
\verylongfermata



\henzeshortfermata

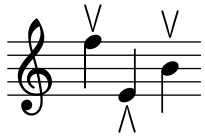


\henzelongfermata

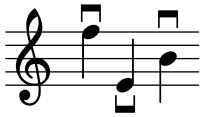


B.13.4 Indicaciones de instrumentos específicos

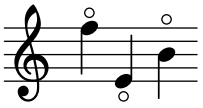
\upbow



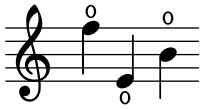
\downbow



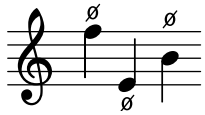
\flageolet



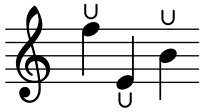
\open



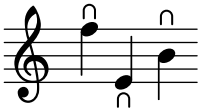
\halfopen



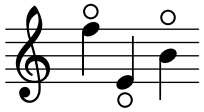
\heel



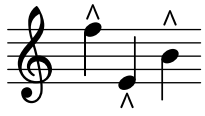
\varheel



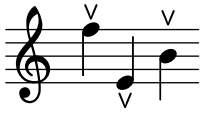
\heelcircle



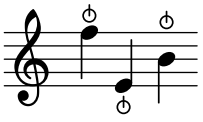
\toe



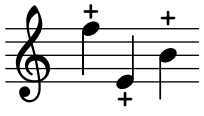
\vartoe



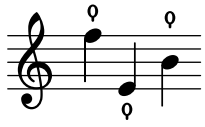
\snappizzicato



\stopped or -+

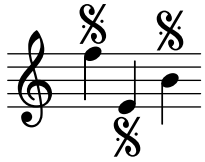


\thumb

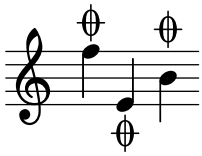


B.13.5 Indicaciones de repetición

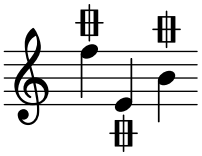
\segno



\coda



\varcoda



B.13.6 Indicaciones antiguas

\accentus



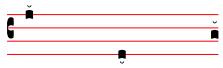
\circulus



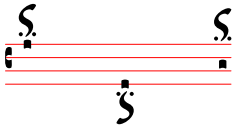
\ictus



\semicirculus

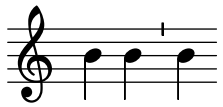


\signumcongruentiae

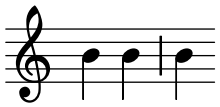


B.14 Lista de signos de respiración

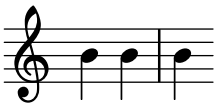
'chantquarterbar



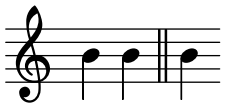
'chanthalfbar



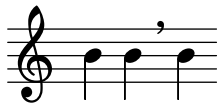
'chantfullbar



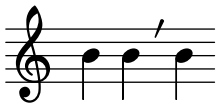
'chantdoublebar



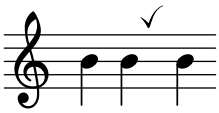
'comma



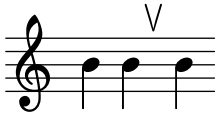
'varcomma



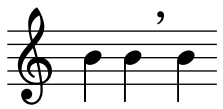
'tickmark



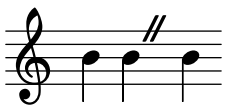
'upbow



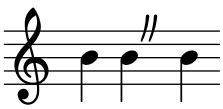
'outsidecomma



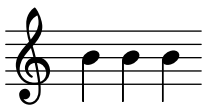
'caesura



'curvedcaesura

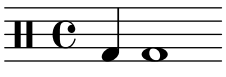


'spacer

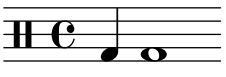


B.15 Notas de percusión

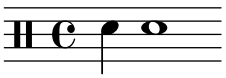
bassdrum
bd



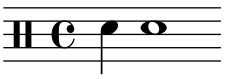
acousticbassdrum
bda



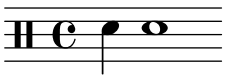
snare
sn



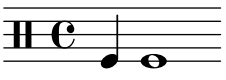
acousticsnare
sna



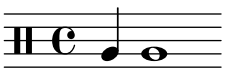
electricsnare
sne



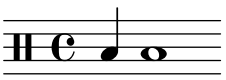
lowfloortom
tomfl



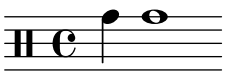
highfloortom
tomfh



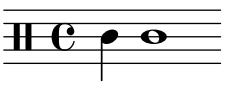
lowtom
toml



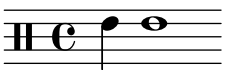
hightom
tomh



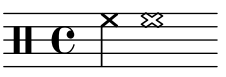
lowmidtom
tomml



himidtom
tommh



hihat
hh



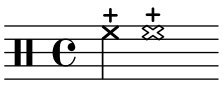
closedhihat
hhc

openhihat
hho

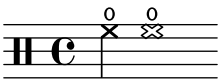
halfopenhihat
hhho

pedalhihat
hhp

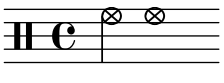
crashcymbal
cymc



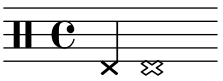
crashcymbala
cymca



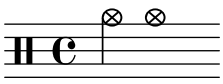
crashcymbalb
cymcb



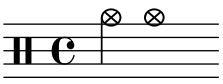
ridecymbal
cymr



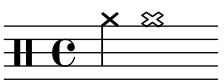
ridecymbala
cymra



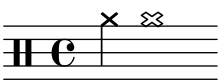
ridecymbalb
cymrb



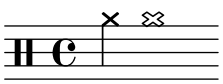
chinesecymbal
cymch



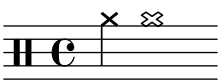
splashcymbal
cyms



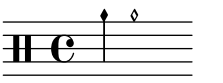
ridebell
rb



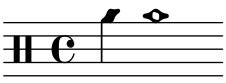
cowbell
cb

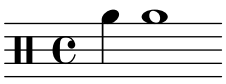


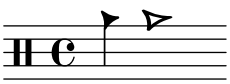
hibongo
boh

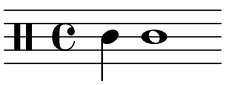


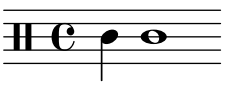
openhibongo
boho



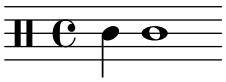




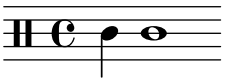




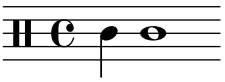
mutehibongo
bohm



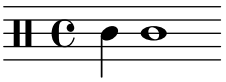
lobongo
bol



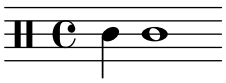
openlobongo
bolo



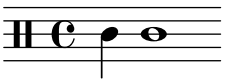
mutelobongo
bolm



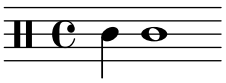
hiconga
cgh



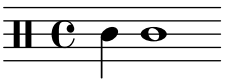
openhiconga
cgho



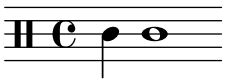
mutehiconga
cghm



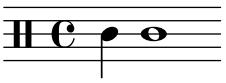
loconga
cgl



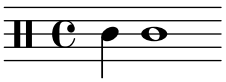
openloconga
cglo



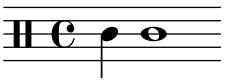
muteloconga
cglm



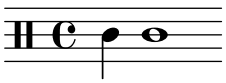
hitimbale
timh



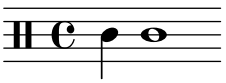
lotimbale
timl



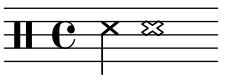
hiagogo
agh



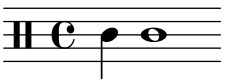
loagogo
agl



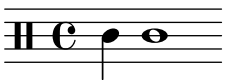
sidestick
ss



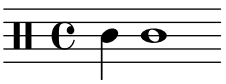
hisidestick
ssh



losidestick
ssl

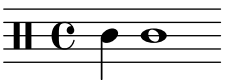


guiro
gui

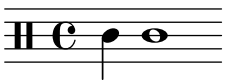


shortguiro
guis

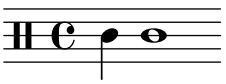
longguiro
guil



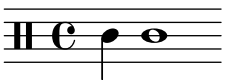
cabasa
cab



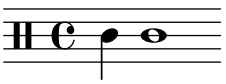
maracas
mar



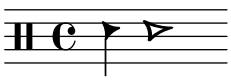
shortwhistle
whs



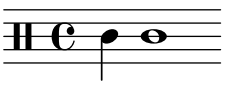
longwhistle
whl



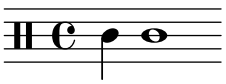
handclap
hc



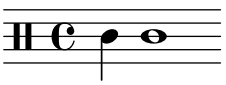
tambourine
tamb

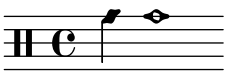


vibraslap
vibs

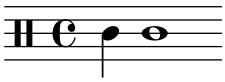


tamtam
tt

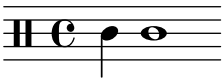




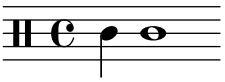
claves
cl



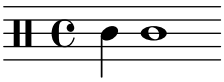
hiwoodblock
wbh



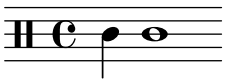
lowoodblock
wbl



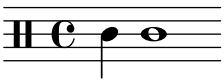
opencuica
cuio



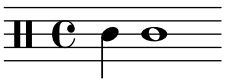
mutecuica
cuim



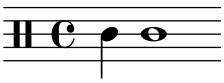
triangle
tri



opentriangle
trio



mutetriangle
trim



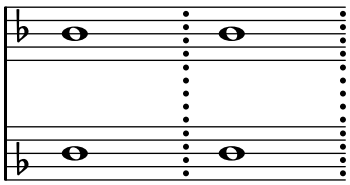
B.16 Lista de líneas divisorias

Cada uno de los sistemas que aparecen aquí muestra el aspecto de un determinado tipo de compás al principio, en medio o al final de una línea.

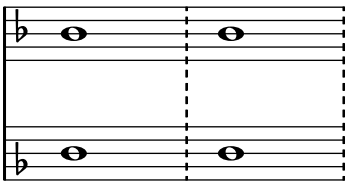
Para ver más información, véase Sección 2.5.1 [Barras de compás], página 120, y Sección 2.5.2 [Automatic bar lines], página 130.

Simple bar lines

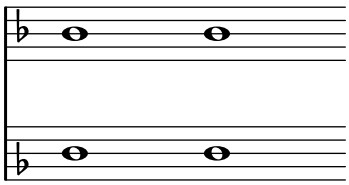
";"



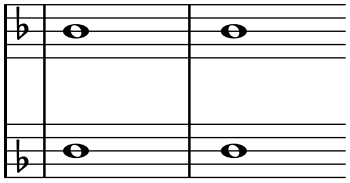
"!"



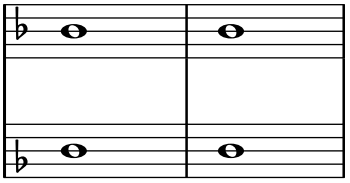
""



"|-s"

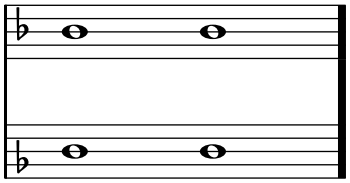


"|"

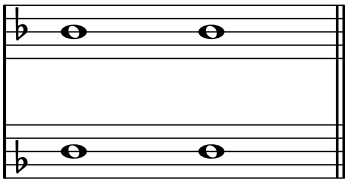


End-of-line bar lines

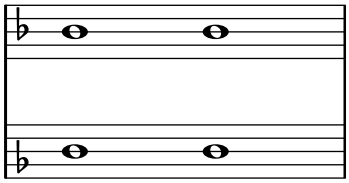
"x-."



"x-||"

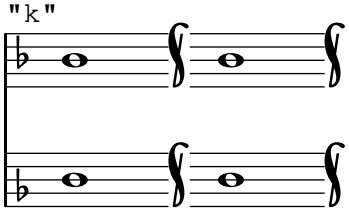


"x-|"

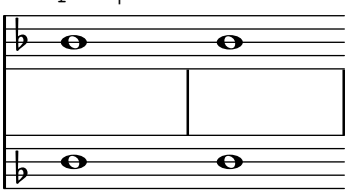


Chant bar lines

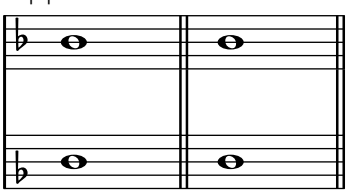
"k"



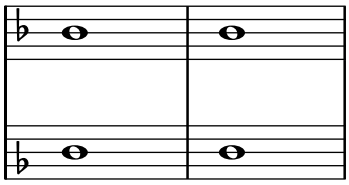
"-span|"



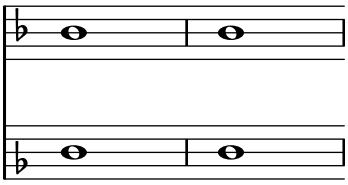
"||"



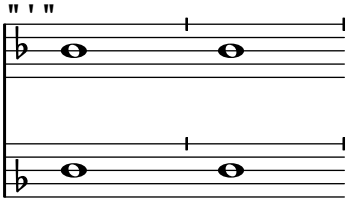
"|"



","

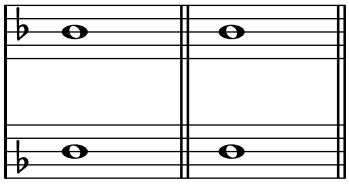


"'"

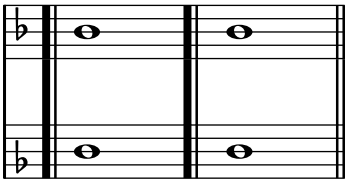


Section bar lines

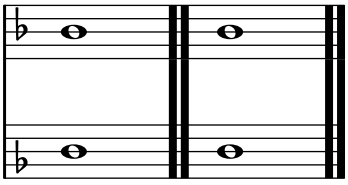
"||"



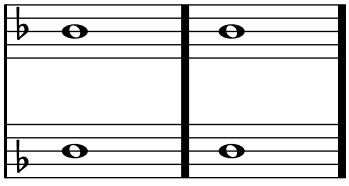
".|-||"



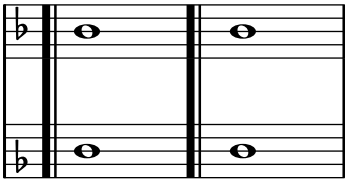
".."



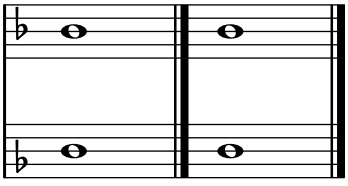
"."



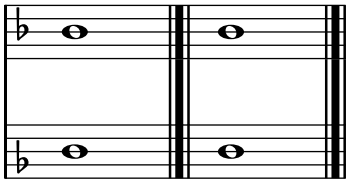
".|-|"



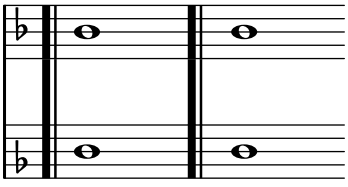
"|."



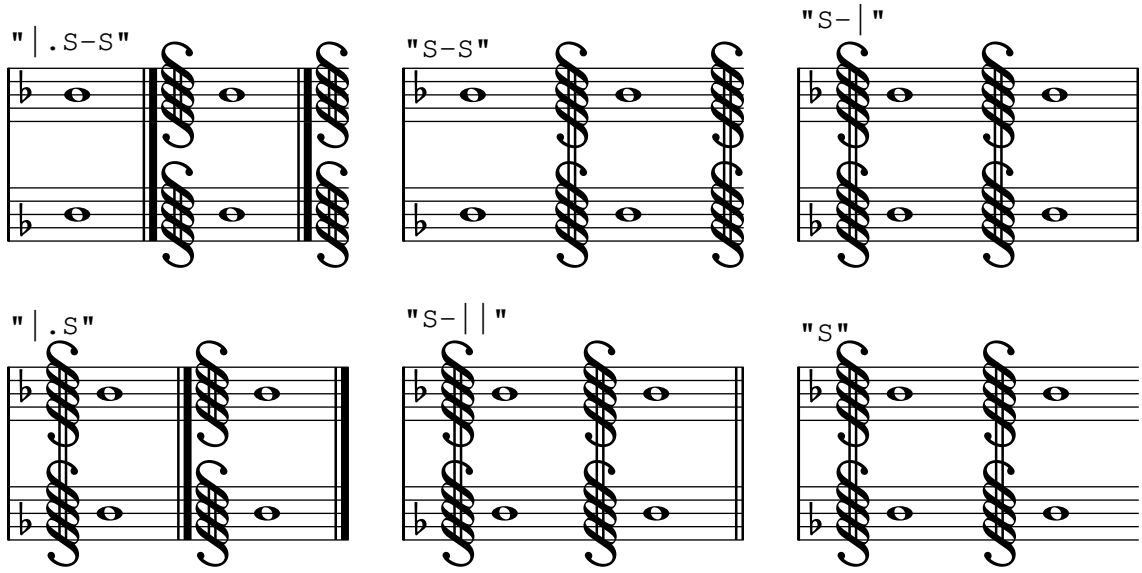
"|.|"



".|"

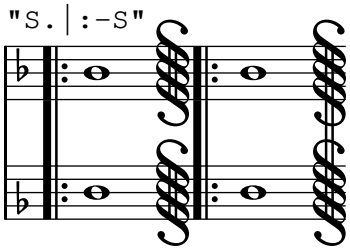


Segno bar lines

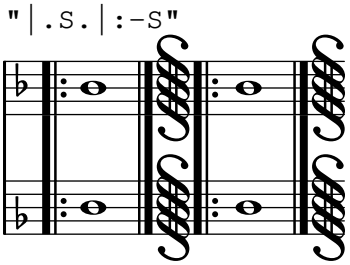


Start-repeat bar lines

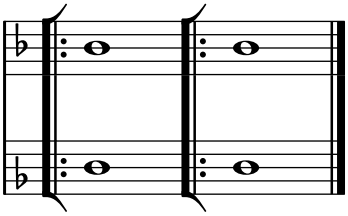
"s.|:-s"



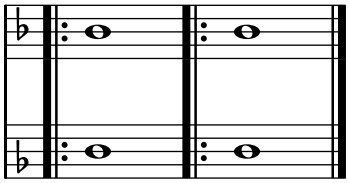
"|.s.|:-s"



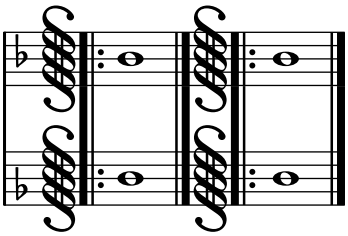
"[|:-|."



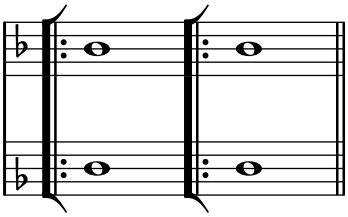
".|:-|."



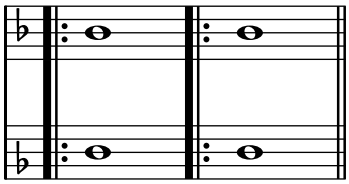
"|.s.|:"



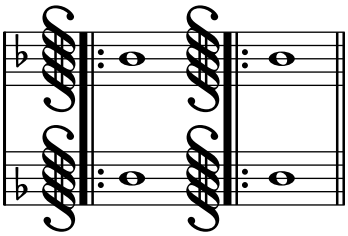
"[|:-||"



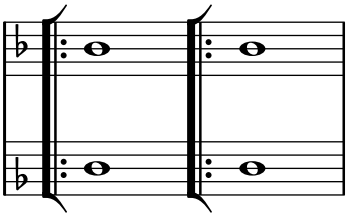
".|:-||"



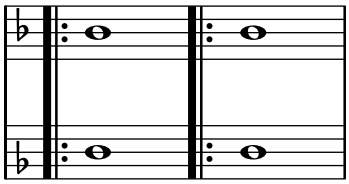
"s.|:-||"



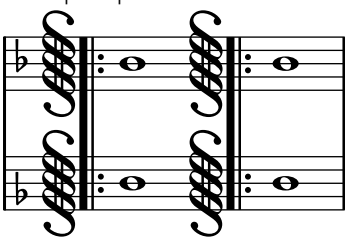
"[|:-|"



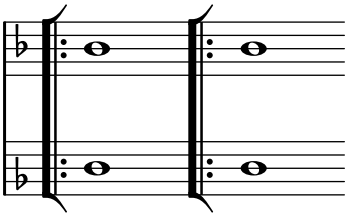
".|:-|"



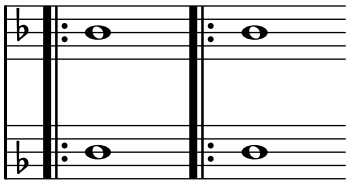
"s.|:-|"



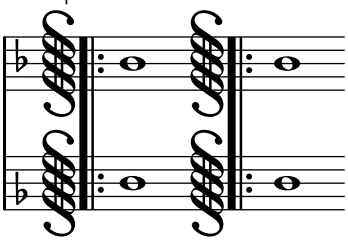
"[|:"



".|:"

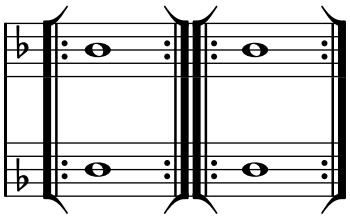


"s.|:"

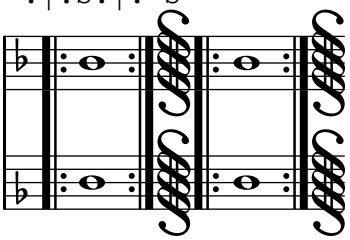


Double-repeat bar lines

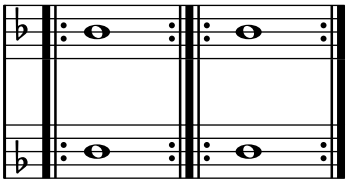
":|][|:"



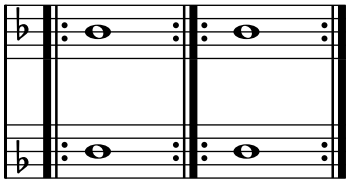
":|.S.|:-S"



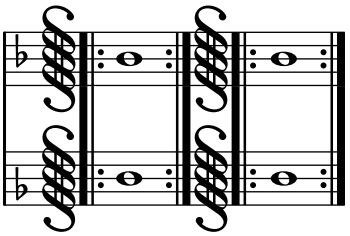
":|.|:"



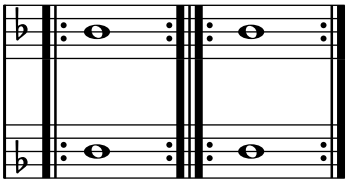
":|.:"



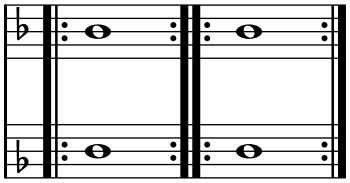
":|.S.|:"



":.|.:"

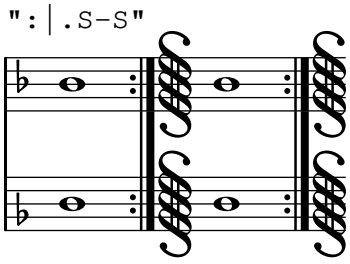


":..:"

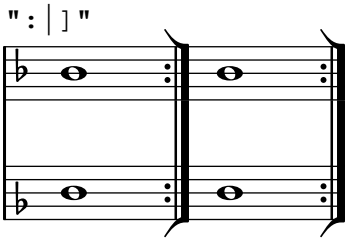


End-repeat bar lines

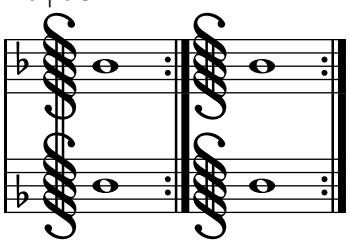
":|.S-S"



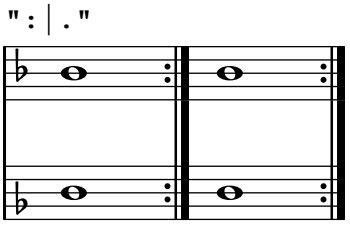
":|]"



":|.S"



":|."



B.17 Valores por omisión para `outside-staff-priority`

La tabla siguiente presenta el valor por omisión de `outside-staff-priority` para todos los objetos gráficos fuera del pentagrama. Los objetos con un valor más bajo se colocan más cerca de la pauta.

Grob	Priority
AccidentalSuggestion	0
BassFigureAlignmentPositioning	25
MultiMeasureRestScript	40
TrillSpanner	50
BarNumber	100
LigatureBracket	200
DynamicLineSpanner	250
TextSpanner	350
OttavaBracket	400
MultiMeasureRestText	450
TextScript	450
CombineTextScript	475
InstrumentSwitch	500
VoltaBracketSpanner	600
MeasureCounter	750
MeasureSpanner	750
HorizontalBracket	800
SostenutoPedalLineSpanner	1000
SustainPedalLineSpanner	1000
UnaCordaPedalLineSpanner	1000
CenteredBarNumberLineSpanner	1200
TextMark	1250
MetronomeMark	1300
JumpScript	1350
CodaMark	1400
SegnoMark	1400
SectionLabel	1450
RehearsalMark	1500

B.18 Valores por omisión para `script-priority`

La tabla siguiente muestra los valores por omisión de `script-priority` para todos los objetos gráficos relacionados con inscripciones (cuyo nombre emplieza en mayúscula) y los nombres de inscripciones (que empiezan en minúscula). Los objetos con valores más bajos se colocan más cerca de la pauta.

Los objetos gráficos `Fingering`, `StringNumber` y `StrokeFinger`, si forman parte de un acorde, añada la posición vertical de la cabeza de nota asociada para obtener el valor final de la propiedad. Los valores de `script-priority` de otras inscripciones que son parte del acorde, permanecen sin cambio.

Para las inscripciones que no son parte de un acorde, en cambio, se añade la posición actual del evento. Esto significa que para inscripciones que tienen el mismo valor de `script-priority`, la primera que está adjuntada a una cabeza de nota se lleva el valor más bajo de la prioridad.

Los objetos gráficos que no están listados aquí (y que están relacionados con las inscripciones) tienen un valor por omisión de 200 en la propiedad `script-priority`.

Grob or Script	Priority
AccidentalPlacement	-100
accentus	-100
circulus	-100
ictus	-100
semicirculus	-100

staccato	-100
tenuto	-50
AccidentalSuggestion	0
Arpeggio	0
ChordBracket	0
ChordSlur	0
flageolet	50
Fingering	100
StrokeFinger	125
StringNumber	150
trill	150
fermata	175
henzelongfermata	175
henzeshortfermata	175
longfermata	175
shortfermata	175
verylongfermata	175
veryshortfermata	175
downbow	180
upbow	180
CombineTextScript	200
TextScript	200

B.19 Objetos gráficos (grobs) y sus símbolos de break-align

La ordenación y espaciado de los elementos de notación tales como las claves, indicaciones de compás, barras de compás, etc., se determina usando ciertos símbolos de Scheme, conocidos como *símbolos de break-align*. De manera más precisa, el orden viene determinado por listas de cómbolos de break-align que están dentro del vector `BreakAlignment.break-align-orders` (véase Sección "break-alignment-interface" en *Referencia de Funcionamiento Interno*); el espaciado se controla por medio de las propiedades de `space-alist` de los objetos gráficos respectivos, cuyos elementos a su vez están indezados a través de símbolos de break-align: por ejemplo, el espaciado desde un objeto `Clef` (clave) hasta una barra de compás que le sigue, está determinado por el valor de `Clef.space-alist.staff-bar`.

Esta tabla relaciona la correspondencia predeterminada entre los símbolos de break-align y los objetos gráficos:

break-align symbol	**used by grob(s)**
ambitus	Ambitus
breathing-sign	BreathingSign
clef	Clef
cue-clef	CueClef
cue-end-clef	CueEndClef
custos	Custos
key-cancellation	KeyCancellation
key-signature	KeySignature
left-edge	LeftEdge
optional-material-end-bracket	OptionalMaterialBracket
optional-material-start-bracket	OptionalMaterialBracket
signum-repetitionis	SignumRepetitionis
staff-bar	SpanBar, DoublePercentRepeat, Divisio and BarLine
staff-ellipsis	StaffEllipsis

time-signature	TimeSignature

Además, existen tres símbolos de break-align abstractos que se pueden usar dentro de las definiciones de space-alist, concretamente first-note, next-note y right-edge. Para ver más detalles, consulte la documentación de la propiedad de objeto space-alist dentro de Sección "break-aligned-interface" en *Referencia de Funcionamiento Interno*.

La tabla siguiente presenta la lista del símbolo de break-align que se usa por cada objeto gráfico:

Grob	**break-align symbol(s)**
Ambitus	ambitus
BarLine	staff-bar
BreathingSign	breathing-sign
Clef	clef
CueClef	cue-clef
CueEndClef	cue-end-clef
Custos	custos
Divisio	staff-bar
DoublePercentRepeat	staff-bar
KeyCancellation	key-cancellation
KeySignature	key-signature
LeftEdge	left-edge
OptionalMaterialBracket	optional-material-start-bracket and optional-material-end-bracket
SignumRepetitionis	signum-repetitionis
SpanBar	staff-bar
StaffEllipsis	staff-ellipsis
TimeSignature	time-signature

B.20 Glosario técnico

Este glosario relaciona los términos técnicos y conceptos que se utilizan internamente en LilyPond. Estos términos pueden aparecer en los manuales, en las listas de distribución de correo o en el código fuente.

alist (lista-A)
: Una lista asociativa o abreviadamente una *lista-A* (alist en inglés) es una pareja de Scheme que asocia un valor con una clave: (*clave . valor*). Por ejemplo, en scm/lily.scm, la lista-A type-p-name-alist asocia ciertos predicadps de tipo (p.ej. ly:music?) con nombres (p.ej. "music") de forma que se pueda informar de los fallos de comprobación de tipo con un mensaje de consola que incluye el nombre del predicado de tipo esperado.

callback
: Una *callback* es una rutina, función o método cuya referencia se pasa como argumento en una llamada a otra rutina, permitiendo así que la runtina llamada invoque a aquélla. La técnica permite que una capa de software de nivel más bajo llame a una función definida en una capa de nivel más alto. Las funciones de callback se usan ampliamente en LilyPond para hacer que el código de Scheme del nivel de usuario controle muchas acciones de bajo nivel.

closure (cerradura)
: En Scheme, se crea una *cerradura* cuando una función, por lo general una expresión lambda (esto es, una función anónima ad-hoc), se pasa como variable. La cerradura contiene el codigo de la función y referencias a las ligaduras léxicas de las variables libres de la función (es decir, las variables que se usan en la expresión pero se definen

fuera de ella). Cuando más tarde se aplica esta función a diferentes argumentos, las ligaduras de variables libres que se capturaron dentro de la cerradura se utilizan para obtener los valores de las variables libres, que a su vez se usarán en el cálculo. Una propiedad útil de las cerraduras es la preservación de los valores internos de las variables de una invocación a otra, permitiendo así que se pueda mantener un estado.

glifo Un *glifo* es una representación gráfica particular de un carácter tipográfico, o una combinación de dos o más caracteres que forman una ligadura. Un conjunto de glifos con un estilo y forma uniformes forman una fuente tipográfica, y un conjunto de fuentes tipográficas que abarcan varios estilos forman un tipo.

Véase también: Sección 8.3 [Fuentes tipográficas], página 341, Sección 22.4 [Caracteres especiales], página 647.

grob (objeto gráfico)
Los objetos de LilyPond que representan elementos de la notación en la salida impresa tales como la cabeza y la plica de las notas, ligaduras de unión y de expresión, digitaciones, claves, etc. se denominan 'objetos de presentación', a menudo conocidos como 'OBjetos GRáficos', o abreviadamente *grobs*. Se representan mediante instancias de la clase `Grob`.

See also: Manual de aprendizaje: Sección "Objetos e interfaces" en *Manual de Aprendizaje*, Sección "Propiedades de los objetos de presentación" en *Manual de Aprendizaje*.
Sección "Convenciones de nombres" en *Manual de Aprendizaje*.
Sección "grob-interface" en *Referencia de Funcionamiento Interno*, Sección "All layout objects" en *Referencia de Funcionamiento Interno*.

inmutable Un objeto *inmutable* es aquel cuyo estado no se puede modificar después de su creación, en contraste con los objetos mutables, que se pueden modificar después de su creación.

En LilyPond, las propiedades inmutables o compartidas definen el estilo y comportamiento predeterminados de los grobs. Se comparten por parte de muchos objetos. En aparente contradicción con su nombre, se pueden cambiar utilizando `\override` y `\revert`.

Véase también más abajo acerca de los objetos mutables.

interfaz Las acciones y propiedades comunes a un conjunto de grobs se agrupan en un objeto denominado `interfaz de grob (grob-inerface)`, o abreviadamente *interfaz*.

Véase también: Sección B.24 [Convenciones de nomenclatura], página 978, Sección 34.2 [Interfaces de la presentación], página 765.
Sección "Objetos e interfaces" en *Manual de Aprendizaje*, Sección "Propiedades de los interfaces" en *Manual de Aprendizaje*.
Sección "Graphical Object Interfaces" en *Referencia de Funcionamiento Interno*.

lexer (analizador léxico)
Un *lexer* o analizador léxico es un programa que convierte una secuencia de caracteres en una serie de elementos o tokens, en un proceso que se llama *análisis léxico*. El analizador léxico de LilyPond convierte el flujo obtenido a partir de un archivo de entrada `.ly` en un flujo descompuesto en tokens más apto para la siguiente fase del procesado: el análisis sintáctico (véase *parser* más adelante). El analizador léxico de LilyPond está construido con el programa `flex` a partir del archivo de lexer `lily/lexer.ll` que contiene las reglas léxicas. Este archivo es parte del código fuente y no se incluye dentro de la instalación binaria de LilyPond.

mutable
: Un objeto *mutable* es aquel cuyo estado se puede modificar después de su creación, en contraste con un objeto inmutable, cuyo estado se fija en el momento de la creación.

  En LilyPond, las propiedades mutables contienen valores específicos de un grob. Por lo general, las listas de otros objetos o los resultados de los cálculos se almacenan en propiedades mutables.

  See also above for immutable objects.

output-def (definición de salida)
: Una instancia de la clase `Output-def` contiene los métodos y estructuras de datos asociados con un bloque de salida. Se crean instancias para los bloques midi, layout y paper.

parser (analizador sintáctico)
: Un *parser* o analizador sintáctico analiza la secuencia de tokens o elementos léxicos producida por un analizador léxico para determinar su estructura gramatical, agrupando los elementos léxicos en conjuntos mayores según las reglas de la gramática. Si la secuencia de elementos léxicos es válida, el producto final es un árbol de tokens cuya raíz es el símbolo inicial de la gramática. Si no se puede conseguir esto, el archivo es inválido y se produce un mensaje de error adecuado. Las agrupaciones sintácticas y las reglas para construir estas agrupaciones a partir de sus elementos constituyentes para la sintaxis de LilyPond están definidas en `lily/parser.yy`. Durante el proceso de compilación, este archivo se procesa por parte de un generador sintáctico, `bison`, para crear el analizador. Es parte del código fuente y no se incluye dentro de la instalación binaria de LilyPond.

variable del analizador sintáctico
: Son variables definidas directamente dentro de Scheme. Su uso directo por parte de los usuarios está fuertemente desaconsejado, porque su semántica de ámbito puede ser confusa.

  Cuando el valor de una de estas variables se modifica dentro de un archivo `.ly`, el cambio es global, y a no ser que se revierta explícitamente, el nuevo valor persiste hasta el final del archivo, afectando a todos los bloques `\score` así como a los archivos externos añadidos con la instrucción `\include`. Esto puede conducir a consecuencias imprevistas y en proyectos de composición tipográfica complejos puede ser difícil de rastrear.

  LilyPond utiliza las siguientes variables del analizador sintáctico, entre otras:

  ```
  afterGraceFraction
  input-language
  musicQuotes
  output-count
  output-suffix
  partCombineListener
  pitchnames
  toplevel-bookparts
  toplevel-scores
  showLastLength
  showFirstLength
  ```

prob (objeto de propiedades)
: Los OBjetos de PRopiedades, o abreviadamente *probs*, son instancias de la clase `Prob`, que es una sencilla clase básica que tiene listas-A de propiedades mutables

e inmutables y los métodos para manipularlas. Las clases `Music` y `Stream_event` derivan de `Prob`. También se crean instancias de la clase `Prob` para almacenar el contenido formateado de los grobs del sistema y los bloques de títulos durante el proceso de disposición de la página.

smob (objeto de Scheme)
: Los *Smobs*, u OBjetos de ScheMe, forman parte del mecanismo utilizado por Guile para exportar objetos de C y de C++ al código de Scheme. En LilyPond, se crean smobs a partir de objetos de C++ por medio de macros. Hay dos tipos de objetos smob: los smobs simples, orientados a objetos inmutables simples como números, y los smobs complejos, usados para objetos con identidades. Si tiene acceso a las fuentes de LilyPond sources, encontrará más información en el archivo `lily/includes/smob.hh`.

spanner (objeto de extensión)
: Los *Spanners*, elementos u objetos de extensión, son una clase de objetos gráficos que no están situados en un lugar fijo de la partitura sino que se extienden de un punto a otro. Entre ellos podemos encontrar las barras de corchea, las ligaduras de unión y las ligaduras de expresión, así como los reguladores y las líneas de pentagrama. Así como los objetos que no son *spanners* slo se pueden dividir en un máximo de dos piezas visibles (por ejemplo, una clave duplicada al final de una línea y el comienzo de la siguiente), los spanners u objetos de extensión se dividen en tantas partes como sea necesario dependiendo de sus puntos de comienzo y final (como un crescendo largo que se extiende a lo largo de tres sistemas, o las líneas del pentagrama, que siempre se extienden a lo largo de la partitura completa).

  Técnicamente, los objetos de extensión se definen como objetos gráficos que tienen la interfaz `spanner-interface`; en la parte C++ de LilyPond, son instancias de la subclase `Spanner` de la clase `Grob`. Los extremos izquierdo y derecho de un objeto de extensión se pueden obtener y modificar usando `ly:spanner-bound` y `ly:spanner-set-bound!`, respectivamente. Los extremos sismpre son items o elementos. El ancestro X de un elemento extenso tiene poco sentido musical, pero se establece normalmente al extremo de la izquierda.

  Véase también: Sección 36.4 [Objetos de extensión (spanners)], página 786.
  Sección “All layout objects” en *Referencia de Funcionamiento Interno*, Sección “spanner-interface” en *Referencia de Funcionamiento Interno*.

stencil (sello)
: Las instancias de la clase `Stencil` contienen la información necesaria para imprimir un objeto tipográfico, un *stencil* o sello. Es un smob simple que contiene una caja de confinamiento, que a su vez define las dimensiones vertical y horizontal del objeto, y una expresión de Scheme que imprime el objeto cuando se evalúa. Los stencils o sellos se pueden combinar para formar sellos más complejos definidos por un árbol de expresiones de Scheme, que a su vez está formado a partir de las expresiones de Scheme de los sellos que lo componen.

  La propiedad `stencil`, que conecta a un grob con su sello, se define dentro de la interfaz `grob-interface`.

  Véase también Sección “grob-interface” en *Referencia de Funcionamiento Interno*.

B.21 Funciones musicales disponibles

En las secciones siguiente, el primer argumento de una función se imprime en un tipo inclinado, seguido del tipo del primer argumento entre paréntesis. A continuación viene el segundo argumento, segudo del tipo del segundo argumento, de nuevo entre paréntesis, y así sucesivamente. Después de la flecha, se imprime el tipo del valor devuelto.

SI un argumento y su tipo están entre corchetes, entonces es opcional y puede omitirse, indicando que LilyPond tendrá que usar en su lugar un valor por omisión. Sin embargo, si el último argumento de una función está etiquetado como opcional, *no se puede* omitir: tiene que pasar el valor `\default` si no quiere especificar un valor explícitamente. Consulte Sección "Scheme function usage" en *Extender* para ver más detalles sobre cómo se manejan los argumentos opcionales.

`\absolute` *music* (music) ⇒ music
: Make *music* absolute.

  This does not actually change the music itself but rather hides it from surrounding `\relative` and `\fixed` commands.

`\acciaccatura` *music* (music) ⇒ music
: Create an acciaccatura from *music*.

`\accidentalStyle` *style* (symbol list) ⇒ music
: Set accidental style to *style*.

  style is a (predefined) symbol list like `piano-cautionary`; véase Sección 1.3.5 [Alteraciones accidentales automáticas], página 32, for the available styles. If it is preceded by a context name, the settings are applied to that context (example: `Staff .piano-cautionary`). Otherwise, the context defaults to `Staff`, except for piano styles, which use `GrandStaff` as a context.

`\addChordShape` *key-symbol* (symbol) *tuning* (pair) *shape-definition* (string or pair) ⇒ void
: Add *shape-definition* as a chord shape.

  It gets added to the `chord-shape-table` hash with the key `(cons` *`key-symbol tuning`*`)`.

`\addInstrumentDefinition` *name* (string) *lst* (list) ⇒ void
: Create instrument *name* with properties *lst*.

  This function is deprecated.

`\addQuote` *name* (string) *music* (music) ⇒ void
: Define *music* as a quotable music expression named *name*.

`\addToTagGroup` *tag-group* (symbol list) *more-tags* (symbol list) ⇒ void
: Add *more-tags* to the tag group identified by *tag-group*.

`\after` *delta* (duration) *ev* (music) *mus* (music) ⇒ music
: Add music *ev* with a delay of *delta* after the onset of *mus*.

  ev is usually a post-event.

`\afterGrace` [*fraction* (non-negative rational, fraction, or moment)] *main* (music) *grace* (music) ⇒ music
: Create *grace* as grace notes after a *main* music expression.

  The musical position of the grace expression is after a given fraction of the main note's duration has passed. If optional argument *fraction* is omitted, the fraction value is taken from `afterGraceFraction`, defaulting to 3/4.

`\allowPageTurn` ⇒ music
: Allow a page turn.

  May be used at top level (i.e., between scores or markups), or inside a score.

`\allowVoltaHook` *bar* (string) ⇒ void
: Allow the volta bracket hook being drawn over bar line *bar*.

\alterBroken *property* (key list or symbol) *arg* (list) *target* (key list or music) ⇒ music
Override *property* for pieces of broken spanner *target* with *arg*.

target may either be music in the form of a starting spanner event, or a symbol list of the form `Context.Grob` or just `Grob`. If *target* is in the form of a spanner event, *property* may also have the form `Grob.property` for specifying a directed tweak.

arg is a list of values, one for each broken piece.

\ambitusAfter *target* (symbol) ⇒ music
Move the ambitus after the break-align symbol *target*.

\appendToTag *tag* (symbol) *more* (music) *music* (music) ⇒ music
Append *more* to *music* tagged with *tag*.

A post-event can be added to the articulations of rhythmic events or chords; other expressions may be added to chords, sequential or simultaneous music.

\appendToTagMarkup *tag* (symbol) *more* (markup) *music* (music) ⇒ music
Append *more* to every markup in *music* tagged with *tag*.

\applyContext *proc* (procedure) ⇒ music
Modify context properties with Scheme procedure *proc*.

\applyMusic *func* (procedure) *music* (music) ⇒ music
Apply procedure *func* to *music*.

\applyOutput *target* (symbol list or symbol) *proc* (procedure) ⇒ music
Apply function `proc` to every layout object matched by *target*.

target takes the form `Context` or `Context.Grob`.

\appoggiatura *music* (music) ⇒ music
Create an appoggiatura from *music*.

\approximatePitch *note* (music) ⇒ music
Indicate that the pitch of *note* approximates a pitch that cannot be known exactly, such as the highest note a singer can sing.

\arpeggioArrowDown ⇒ music
Let \arpeggio produce a wavy line with a down arrow.

\arpeggioArrowUp ⇒ music
Let \arpeggio produce a wavy line with an up arrow.

\arpeggioBracket ⇒ music
Let \arpeggio produce a square bracket.

Note: For a bracket designating a non-arpeggiated chord, it is better to use \nonArpeggiato than to use \arpeggio and alter the appearance.

\arpeggioNormal ⇒ music
Let \arpeggio produce a wavy line without an arrow.

\arpeggioParenthesis ⇒ music
Let \arpeggio produce a vertical slur.

Note: For a vertical slur designating a quasi-non-arpeggiated chord, it is better to use \chordSlur than to use \arpeggio and alter the appearance.

\arpeggioParenthesisDashed ⇒ music
Let \arpeggio produce a dashed vertical slur.

Note: For a vertical slur designating a quasi-non-arpeggiated chord, it is better to use \chordSlur than to use \arpeggio and alter the appearance.

\assertBeamQuant *l* (pair) *r* (pair) ⇒ music
: Testing function: check whether the beam quants *l* and *r* are correct.

\assertBeamSlope *comp* (procedure) ⇒ music
: Testing function: check whether the slope of the beam is the same as *comp*.

\atLeft *m* (music) ⇒ post-event
: Set `side-axis` to `X` and `direction` to `LEFT` for *mus*.

\atRight *m* (music) ⇒ post-event
: Set `side-axis` to `X` and `direction` to `RIGHT` for *mus*.

\augmentum *expr* (music) ⇒ music
: Add augmentum dots (*morae*) to Gregorian chant *expr*.

\autoChange [*pitch* (pitch)] [*clef-1* (context modification)] [*clef-2* (context modification)] *music* (music) ⇒ music
: Make voices for *music* that switch between staves automatically.

  The optional argument *pitch* specifies where to switch staves; the default is `c'`. Optional arguments *clef-1* and *clef-2* specify which clefs to use; this only works for implicitly instantiated staves.

  Example:

```
\autoChange d' \with { \clef alto } { g4 d' g' }
```

\balloonGrobText *grob-name* (symbol) *offset* (pair of numbers) *text* (markup) ⇒ music
: Attach *text* to *grob-name* at offset *offset* (use like `\once`).

\balloonText *offset* (pair of numbers) *text* (markup) ⇒ post-event
: Attach *text* at *offset* (use like `\tweak`).

\bar *type* (string) ⇒ music
: Insert a bar line of type *type*, overriding any automatic bar lines.

\barNumberCheck *n* (integer) ⇒ music
: Print a warning if the current bar number is not *n*.

\beamExceptions *music* (music) ⇒ any type
: Set beam exceptions.

  This function extracts a value suitable for setting `Timing.beamExceptions` from the given pattern with explicit beams in *music*. A bar check '`|`' has to be used between bars of patterns in order to reset the timing.

\bendAfter *delta* (real number) ⇒ post-event
: Create a fall or doit of pitch interval *delta*.

\bendHold *mus* (music) ⇒ post-event
: Set `BendSpanner.style` to `'hold` for *mus*.

\bendStartLevel *idx* (non-negative, exact integer) *mus* (music) ⇒ post-event
: Set `Bendspanner.details.successive-level` to *idx* for *mus*.

\bookOutputName *newfilename* (string) ⇒ void
: Direct output for the current book block to *newfilename*.

  This is equivalent to setting `output-filename` in the current book's `\paper` block.

\bookOutputSuffix *newsuffix* (string) ⇒ void
: Set the output file name suffix for the current book block to *newsuffix*.

  This is equivalent to setting `output-suffix` in the current book's `\paper` block.

`\breakAlignInsert` [*positions* (vector)] *symbol-to-move* (symbol) *sorting-symbol* (symbol) *anchor-symbol* (symbol) ⇒ music

Re-order break-align symbols.

This function is for tasks like putting a clef change after the bar line, which would be accomplished by `\breakAlignInsert clef after staff-bar`.

sorting-symbol can be `before` or `after`. Depending on its value, the break-align symbol *symbol-to-move* gets moved directly before/after the symbol *anchor-symbol* in the `break-align-orders` vector of `Score.BreakAlignment`.

The available break-align symbols and their corresponding grobs are listed in Sección "Grobs and their break-align symbols" en *Referencia de la Notación*.

The optional argument *positions* determines whether the specified change should become visible at the beginning/the end/the middle of a line. Possible values are: `#all-visible` (default), `#begin-of-line-invisible`, `#center-invisible`, `#end-of-line-invisible`, `#begin-of-line-visible`, `#center-visible`, `#end-of-line-visible`, `#all-invisible`.

`\breathe` ⇒ music

Insert a breath mark.

`\caesura` ⇒ music

Insert a caesura.

`\chordRepeats` [*event-types* (list)] *music* (music) ⇒ music

Extend 'q' to also repeat articulation.

This function walks through *music*, putting the notes of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(string-number-event)`.

`\clef` *type* (string) ⇒ music

Set the current clef to *type*.

`\codaMark` [*num* (non-negative, exact integer)] ⇒ music

Create a coda mark.

num may be 1 for the first mark, 2 for the second, etc., or it may be `\default` to use the next number in sequence automatically.

`\compressMMRests` *music* (music) ⇒ music

Convert empty bars to multi-measure rests in *music*.

`\contextPropertyCheck` *property-path* (symbol list or symbol) [*expected-value* (any type)] ⇒ music

Check that the context property identified by *property-path* is set to *expected-value* in that very context: being set in an enclosing context is insufficient. If *expected-value* is `\default`, check that the property is unset in that context.

If *property-path* does not name a context, the check occurs in the current context.

Print a warning if the requested context is not visible looking upward from the current context or if the state of the property in the requested context is unexpected.

Caveat: Like some other context-specific commands, a context given in *property-path* is ignored in certain cases, such as inside `\with`.

`\crossStaff` *notes* (music) ⇒ music

Create cross-staff stems for *notes*.

`\cueClef` *type* (string) ⇒ music

Set the current cue clef to *type*.

`\cueClefUnset` ⇒ music
: Unset the current cue clef.

`\cueDuring` *what* (string) *dir* (direction) *main-music* (music) ⇒ music
: Create a cue.

  This function inserts the contents of quote *what* corresponding to *main-music*, in a `CueVoice` context called `cue` oriented by *dir*.

`\cueDuringWithClef` *what* (string) *dir* (direction) *clef* (string) *main-music* (music) ⇒ music
: Create a cue with clef.

  This function inserts the contents of quote *what* corresponding to *main-music*, in a `CueVoice` context called `cue` oriented by *dir* and using clef *clef*.

`\deadNote` *note* (music) ⇒ music
: Print *note* with a cross-shaped note head.

`\defineBarLine` *bar* (string) *glyph-list* (list) ⇒ void
: Define bar line settings for bar line *bar*.

  The list *glyph-list* must have three entries, defining substitute glyphs for the end of a line, the beginning of a line, and a span bar, respectively. The substitute glyphs may be either strings or Booleans: `#t` calls for the same value as *bar* and `#f` calls for no glyph.

`\displayLilyMusic` [*port* (output port)] *music* (music) ⇒ music
: Write LilyPond's input representation of *music*.

  If *port* is omitted, the output defaults to the console (stdout).

`\displayMusic` [*port* (output port)] *music* (music) ⇒ music
: Write the internal representation of *music*.

  If *port* is omitted, the output defaults to the console (stdout).

`\displayScheme` [*port* (output port)] *expr* (any type) ⇒ any type
: Write the internal Scheme representation of *expr*.

  If *port* is omitted, the output defaults to the console (stdout).

`\dropNote` *num* (integer) *music* (music) ⇒ music
: 'Drop' the *num*-th note in each chord of *music*.

  This function moves the affected notes down (usually by an octave) to be lower than the other notes of the chord. The position in a chord is counted downwards from the top.

  The opposite function is `\raiseNote`.

`\enablePerStaffTiming` ⇒ void
: Enable polymeter with unaligned measures.

  This function moves the timing management from `Score` to `Staff`-like contexts. This is done by removing the `Timing_translator` from `Score`, and adding it to all contexts having the `Staff` alias.

  Use this within an output definition.

`\endSpanners` *music* (music) ⇒ music
: Terminate spanners.

  This function prematurely ends all spanners in *music* by inserting an end-spanner event at the end of the argument, without the need of specific end-spanner commands.

`\eventChords` *music* (music) ⇒ music
: Compatibility function: Handle isolated rhythmic events in *music*.

Use this to wrap `EventChord` around isolated rhythmic events occuring since version 2.15.28, after expanding repeat chords 'q'.

Not needed for new code.

`\featherDurations` *scale* (non-negative rational, fraction, or moment) *music* (music) ⇒ music
Adjust feathered beam durations in *music* by *scale*.

`\finger` *finger* (index or markup) ⇒ post-event
Apply *finger* as a fingering indication.

`\fixed` *pitch* (pitch) *music* (music) ⇒ music
Use the octave of *pitch* as the default octave for *music*.

`\footnote` [*mark* (markup)] *offset* (pair of numbers) *footnote* (markup) *item* (symbol list or music) ⇒ music
Make the markup *footnote* a footnote on *item*.

The footnote is marked with a markup *mark* moved by *offset* with respect to the marked music.

If *mark* is not given or specified as `\default`, it is replaced by an automatically generated sequence number. If *item* is a symbol list of form *`Grob`* or *`Context.Grob`*, then grobs of that type are marked at the current time step in the given context (default `Bottom`).

If *item* is music, the music gets a footnote attached to a grob immediately attached to the event, like `\tweak` does. For attaching a footnote to an *indirectly* caused grob, write `\single\footnote`, use *item* to specify the grob, and follow it with the music to annotate.

Like with `\tweak`, if you use a footnote on a following post-event, the `\footnote` command itself needs to be attached to the preceding note or rest as a post-event with '`-`'.

`\grace` *music* (music) ⇒ music
Insert *music* as grace notes.

`\grobdescriptions` *descriptions* (list) ⇒ any type
Create a context modification from *descriptions*.

The argument is a list in the format of `all-grob-descriptions`.

`\harmonicByFret` *fret* (number) *music* (music) ⇒ music
Convert *music* into mixed harmonics.

The resulting notes resemble harmonics played on a fretted instrument by touching the strings at *fret*.

`\harmonicByRatio` *ratio* (number) *music* (music) ⇒ music
Convert *music* into mixed harmonics.

The resulting notes resemble harmonics played on a fretted instrument by touching the strings at the point given through *ratio*.

`\harmonicNote` *note* (music) ⇒ music
Print *note* with a diamond-shaped note head.

`\harmonicsOn` ⇒ music
Set the default note head style to a diamond-shaped style.

`\hide` *item* (symbol list or music) ⇒ music
Make *item* invisible while still retaining its dimensions.

If *item* is a symbol list of form *GrobName* or *Context.GrobName*, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

This function sets *item*'s `transparent` property to `#t`.

`\incipit` [*offset* (number)] *incipit-music* (music) ⇒ music
Output *incipit-music* as an incipit.

incipit-music is typeset within a `MensuralStaff` context; the result is positioned before the main staff (as part of an `InstrumentName` grob) to indicate the music's original notation. The optional argument *offset* specifies how much the incipit should be moved to the left (default value is 0).

In the special case that *incipit-music* has the form `\new` *xxx* ..., where *xxx* is a context type not accepted by `MensuralStaff`, it is taken directly.

`\inherit-acceptability` *to* (symbol) *from* (symbol) ⇒ void
Make two contexts 'compatible'.

When used in an output definition, modify all context definitions such that context *to* is accepted as a child by all contexts that also accept *from*.

`\initialContextFrom` *music* (music) ⇒ music
Enter the initial context of *music* and ignore the rest of it.

This is useful for prepending music while preserving the influence of the original music on the context.

Example:

```
{
  \initialContextFrom \originalMusic
  \prependedMusic
  \originalMusic
  \appendedMusic
}
```

`\inStaffSegno` ⇒ music
Put the segno variant `varsegno` at this position into the staff.

This is compatible with the repeat command.

`\instrumentSwitch` *name* (string) ⇒ music
Switch instrument to *name*.

name must have been predefined with function `\addInstrumentDefinition`.

This function is deprecated.

`\inversion` *around* (pitch) *to* (pitch) *music* (music) ⇒ music
Invert *music* about *around* and transpose from *around* to *to*.

`\invertChords` *num* (integer) *music* (music) ⇒ music
Invert any chords in *music* into their *num*-th position.

Chord inversions may be directed downwards using negative integers.

`\jump` *text* (markup) ⇒ music
Use *text* to mark a point of departure, e.g., 'Gavotte I D.C.'.

`\keepWithTag` *tags* (symbol list or symbol) *music* (music) ⇒ music
Keep tagged music.

This function only includes elements of *music* that are tagged with one of the tags in *tags*. *tags* may be either a single symbol or a list of symbols.

Each tag may be declared as a member of at most one tag group (defined with `\tagGroup`). If none of a *music* element's tags share a tag group with one of the specified *tags*, the element is retained.

`\key` [*tonic* (pitch)] [*pitch-alist* (list of number pairs)] ⇒ music
Set key to *tonic* and scale *pitch-alist*.

If both arguments are omitted (i.e., replaced by `\default`), just generate a `KeyChangeEvent`, which prints the current key signature again.

`\killCues` *music* (music) ⇒ music
Remove cue notes from *music*.

`\label` *label* (symbol) ⇒ music
Create *label* as a referrable label.

The value stored in *label* is the page number, which can be extracted with the `\page-ref` markup command later on.

`\language` *language* (string) ⇒ void
Set note names for language *language*.

This function updates the Scheme variables `pitchnames` and `input-language`. The former is an alist that holds the pitches of all note names for the selected language; the latter holds the language name itself (as a symbol).

`\languageRestore` ⇒ void
Restore the previously-saved `pitchnames` alist.

`\languageSaveAndChange` *language* (string) ⇒ void
Save current `pitchnames` alist and change note names to *language*.

`\ligature` *music* (music) ⇒ music
Make a ligature from Gregorian Chant *music*.

This is equivalent to enclosing *music* with `\[` and `\]`.

`\magnifyMusic` *mag* (positive number) *music* (music) ⇒ music
Magnify the size of *music* by factor *mag*.

This doesn't change the staff size. Stems, beams, slurs, ties, and horizontal spacing are adjusted automatically.

`\magnifyStaff` *mag* (positive number) ⇒ music
Change the staff size by factor *mag*.

This adjusts notation size and horizontal spacing automatically.

`\makeClusters` *arg* (music) ⇒ music
Display chords in *arg* as clusters.

`\makeDefaultStringTuning` *symbol* (symbol) *pitches* (list) ⇒ void
Define string tuning *symbol* by a list of *pitches*.

symbol also gets registered in `defaultStringTunings` for documentation purposes.

`\mark` [*label* (index or markup)] ⇒ music
Create a rehearsal mark.

If *label* is an integer, create the rehearsal mark for the given sequence number. If *label* is `\default`, create the next sequential rehearsal mark. If *label* is markup, use it for the mark.

`\markupMap` *path* (symbol list or symbol) *markupfun* (markup-function) *music* (music) ⇒ music
Apply *markupfun* to property *path* in *music*.

Argument *path* is either of the form *property* or *MusicExpression.property*. If *MusicExpression* is not given, *markupfun* gets applied to all properties called *property*, otherwise it is restricted to *MusicExpression* events. If *property* is not a markup, it is ignored.

In the following example, both the tempo indication and the bowing instruction are printed in red. If you replace `text` with `TempoChangeEvent.text`, only the tempo indication changes the color.

```
\markupMap
  text
  \markup \with-color #red \etc
  { \tempo "Largo" g'2_"arco" c'' }
```

`\modalInversion` *around* (pitch) *to* (pitch) *scale* (music) *music* (music) ⇒ music
Invert *music* about *around* using *scale* and transpose from *around* to *to*.

`\modalTranspose` *from* (pitch) *to* (pitch) *scale* (music) *music* (music) ⇒ music
Transpose *music* from pitch *from* to pitch *to* using *scale*.

`\musicLength` *music* (music) ⇒ any type
Return the duration of *music* as a `ly:moment`.

Examples:

```
\musicLength 8 ⇒ #(ly:make-moment 1/8)
\musicLength {8. 8 8} ⇒ #(ly:make-moment 7/16)
```

`\musicMap` *proc* (procedure) *mus* (music) ⇒ music
Apply *proc* to *mus* and all of the music it contains.

`\noPageBreak` ⇒ music
Forbid a page break.

May be used at top level (i.e., between scores or markups), or inside a score.

`\noPageTurn` ⇒ music
Forbid a page turn.

May be used at top level (i.e., between scores or markups), or inside a score.

`\octaveCheck` *pitch* (pitch) ⇒ music
Do an octave check.

This prints a warning if the interval between the previous note and *pitch* is not within a fourth.

`\offset` *property* (symbol list or symbol) *offsets* (any type) *item* (key list or music) ⇒ music
Offset the default value of *property* of *item* by *offsets*.

If *item* is a string, the result is an override for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

`\omit` *item* (symbol list or music) ⇒ music
Omit *item* without taking up space.

If *item* is a symbol list of form *GrobName* or *Context.GrobName*, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

This function sets *item*'s `stencil` property to `#f`.

`\once` *music* (music) ⇒ music
Set property `once` to `#t` on all layout instruction events in *music*.

\ottava *octave* (integer) ⇒ music
: Set the octavation to *octave*.

  A positive value *n* indicates *n* octaves higher; a negative value *n* octaves lower, and value 0 means no octavation.

\overrideProperty *grob-property-path* (list of indices or symbols) *value* (any type) ⇒ music
: Set the grob property specified by *grob-property-path* to *value*.

  grob-property-path is a symbol list of the form *Context.GrobName.property* or *GrobName.property*, possibly with subproperties given as well.

  As opposed to \override, which overrides the context-dependent defaults with which a grob is created, this command uses Apply_output_engraver at the grob acknowledge stage. This may be necessary for overriding values set after the initial grob creation.

\overrideTimeSignatureSettings *time-signature* (boolean-or-fraction) *beat-base* (positive exact rational, fraction (as pair), moment, or +inf.0) *structure* (a number list or a list of them) *beam-exceptions* (list) ⇒ music
: Override time signature settings.

  This function sets the timeSignatureSettings entry for *time-signature* to use *beat-base*, *structure*, and *beam-exceptions*.

  beat-base and *beam-exceptions* are used as-is for beatBase and beamExceptions.

  structure is used to derive values for beatStructure and submeasureStructure as described for the \time command.

\pageBreak ⇒ music
: Force a page break.

  May be used at top-level (i.e., between scores or markups), or inside a score.

\pageTurn ⇒ music
: Force a page turn.

  May be used at top-level (i.e., between scores or markups), or inside a score.

\palmMute *note* (music) ⇒ music
: Print *note* with a triangle-shaped note head.

\palmMuteOn ⇒ music
: Set the default note head style to a triangle-shaped style.

\parallelMusic *voice-ids* (list) *music* (music) ⇒ void
: Define parallel music sequences.

  Within *music*, parallel music sequences are separated by '|' characters. The sequences are assigned to the LilyPond music identifiers provided in *voice-ids*.

  For example, this code

```
\parallelMusic A,B,C {
  c c | d d | e e |
  d d | e e | f f |
}
```

  is equivalent to

```
A = { c c | d d }
B = { d d | e e }
C = { e e | f f }
```

  The last bar checks in a sequence are not copied to the result in order to facilitate ending the last entry at non-bar boundaries.

\parenthesize *arg* (symbol list or music) ⇒ music
: Tag *arg* to be parenthesized.

  arg may be either a music event or a grob path.

\partCombine [*chord-range* (pair of numbers)] *part1* (music) *part2* (music) ⇒ music
: Combine two parts into a single staff.

  This takes the music in *part1* and *part2* and returns a music expression containing simultaneous `Voice` contexts (called `one` for the upper and `two` for the lower voice). Where appropriate, *part1* and *part2* are combined into a single voice (called `shared` or `solo`, depending on context).

  Optional argument *chord-range* is a pair `(start . stop)` that defines the range in which the two voices are printed as chords (or unison); the default value is `(0 . 8)`, which means that intervals up to and including a ninth are unified.

\partCombineDown [*chord-range* (pair of numbers)] *part1* (music) *part2* (music) ⇒ music
: Combine two parts into a single staff with all stems downwards.

  See function `\partCombine` for details.

\partCombineForce [*type* (symbol)] ⇒ music
: Override the part-combiner mode with *type*.

  The following table gives the possible values for *type*, together with the corresponding shorthand functions.

apart	\partCombineApart
chords	\partCombineChords
unisono	\partCombineUnisono
solo1	\partCombineSoloI
solo2	\partCombineSoloII
\default	\partCombineAutomatic

\partCombineUp [*chord-range* (pair of numbers)] *part1* (music) *part2* (music) ⇒ music
: Combine two parts into a single staff with all stems upwards.

  See function `\partCombine` for details.

\partial *dur* (duration) ⇒ music
: Adjust the measure position to end the current measure at *dur* past the point of use. As a special case, when used at the start, create an anacrusis before the first measure.

\phrasingSlurDashPattern *dash-fraction* (number) *dash-period* (number) ⇒ music
: Set up a custom dash pattern style for phrasing slurs.

  dash-fraction gives the size of one dash relative to *dash-period*; *dash-period* is the length of one dash plus one space. LilyPond adjusts *dash-period* to produce symmetrical output.

  More complex patterns can be achieved by directly manipulating the `PhrasingSlur .dash-definition` property.

\pitchedTrill *main-note* (music) *secondary-note* (music) ⇒ music
: Print a pitched trill.

  main-note is the main note of the trill; *secondary-note* gets printed as a stemless note head in parentheses.

\pointAndClickOff ⇒ void
: Suppress links to LilyPond source code in music output.

`\pointAndClickOn` ⇒ void
: Generate links to LilyPond source code in music output.

  This enables the creation of code in a PDF or SVG output file to reference the originating LilyPond source code (i.e., file name, line number, and column). This is helpful when developing a score; however, the output file becomes much larger.

`\pointAndClickTypes` *types* (symbol list or symbol) ⇒ void
: Generate point-and-click info for music of type *types* only.

  types is a single music expression (such as `#'note-event`) or a list of music expressions.

`\polymetric` [*time-signature-command* (music)] ⇒ music
: Apply *time-signature-command* to the local context rather than to `Timing`.

  Example:

```
\context Staff \polymetric \time 6/8
```

  The local measure must align with the reference measure defined in `Timing`. So that explicit changes to `Timing.measureLength` (e.g., for ad-hoc irregular measures) remain visible, this command does not set the `measureLength` property in the local context.

  A nominally incompatible local measure can be fitted to the controlling `Timing` measure by applying `\scaleDurations` to this command. That adjusts the value of the internal property `meterScalingFactor` in the local context.

  To unset the local properties and resume using the values from the `Timing` context, use `\polymetric \default`.

`\popContextProperty` *path* (list of indices or symbols) ⇒ music
: Pop value of context property *path* from stack and set it.

  This is the opposite to function `\pushContextProperty`.

`\preBend` *mus* (music) ⇒ post-event
: Set `BendSpanner.style` to `'pre-bend` for *mus*.

`\preBendHold` *mus* (music) ⇒ post-event
: Set `BendSpanner.style` to `'pre-bend-hold` for *mus*.

`\propertyOverride` *grob-property-path* (list of indices or symbols) *value* (any type) ⇒ music
: Set the grob property specified by *grob-property-path* to *value*.

  grob-property-path is a symbol list of the form *`Context.GrobName.property`* or *`GrobName.property`*, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in `\override` command.

`\propertyRevert` *grob-property-path* (list of indices or symbols) ⇒ music
: Revert the grob property specified by *grob-property-path* to its previous value.

  grob-property-path is a symbol list of the form *`Context.GrobName.property`* or *`GrobName.property`*, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in `\revert` command.

`\propertySet` *property-path* (symbol list or symbol) *value* (any type) ⇒ music
: Set the context property specified by *property-path* to *value*.

  This music function is mostly intended for use from Scheme as a substitute for the built-in `\set` command.

\propertyTweak *prop* (key list or symbol) *value* (any type) *item* (key list or music) ⇒ music
Add a tweak to *item*, usually music.

This function sets the value of property *prop* to *value*; it generally behaves like \tweak but will turn into an \override when *item* is a symbol list. In that case, *item* specifies the grob path to override. This is mainly useful when using \propertyTweak as as a component for building other functions like \omit. It is not the default behavior for \tweak since many input strings in \lyricmode can serve equally as music or as symbols, which causes surprising behavior when tweaking lyrics using the less specific semantics of \propertyTweak.

prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

\propertyUnset *property-path* (symbol list or symbol) ⇒ music
Unset the context property specified by *property-path*.

This music function is mostly intended for use from Scheme as a substitute for the built-in \unset command.

\pushContextProperty *path* (list of indices or symbols) ⇒ music
Push the current value of context property *path* to a stack.

The property can later be restored to the saved value with function \popContextProperty.

\pushToTag *tag* (symbol) *more* (music) *music* (music) ⇒ music
Add *more* to the front of *music* tagged with *tag*.

A post-event can be added to the articulations of rhythmic events or chords; other expressions may be added to chords, sequential or simultaneous music.

\pushToTagMarkup *tag* (symbol) *more* (markup) *music* (music) ⇒ music
Prepend *more* to every markup in *music* tagged with *tag*.

\quoteDuring *what* (string) *main-music* (music) ⇒ music
Indicate a section of music to be quoted.

what indicates the name of the quoted voice, as specified in an \addQuote command. *main-music* is used to indicate the length of music to be quoted; it usually contains spacers or multi-measure rests.

\raiseNote *num* (integer) *music* (music) ⇒ music
'Raise' the *num*-th note in each chord of *music*.

This function moves the affected notes up (usually by an octave) to be higher than the other notes of the chord. The position in a chord is counted upwards from the bottom.

The opposite function is \dropNote.

\reduceChords *music* (music) ⇒ music
Reduce chords contained in *music* to single notes.

This is intended mainly for reusing music in a RhythmicStaff context. It does not reduce simultaneous music.

\relative [*pitch* (pitch)] *music* (music) ⇒ music
Make *music* relative to *pitch*.

If *pitch* is omitted, the first note in *music* is given in absolute pitch.

\removeFromTagGroup *tag-group* (symbol list) *tags* (symbol list) ⇒ void
Remove *tags* from the tag group identified by *tag-group*.

\removeWithTag *tags* (symbol list or symbol) *music* (music) ⇒ music
Remove elements of *music* that are tagged with one of the tags in *tags*.

tags may be either a single symbol or a list of symbols.

\replaceWithTag *tag* (symbol) *replacement* (music) *music* (music) ⇒ music
Replace tagged elements in *music*.

Everything tagged with *tag* (including the tagging itself) gets replaced with *replacement*.

\replaceWithTagMarkup *tag* (symbol) *replacement* (markup) *music* (music) ⇒ music
Replace tagged markup in *music*.

All markup tagged with *tag* (including the tagging itself) gets replaced with *replacement*.

\resetRelativeOctave *pitch* (pitch) ⇒ music
Set the octave inside a \relative section to *pitch*.

\resetTagGroup *tag-group* (symbol list) ⇒ void
Reset the tag group equal to *tag-group*.

\resetTagGroups ⇒ void
Reset all tag groups previously created with \tagGroup.

\responsum *music* (music) ⇒ music
Prepend character U+211F (RESPONSE) to the lyrics represented by *music*.

\retrograde *music* (music) ⇒ music
Return *music* in reverse order.

\revertTimeSignatureSettings *time-signature* (pair) ⇒ music
Revert timeSignatureSettings for time signatures equal to *time-signature*.

\rightHandFinger *finger* (index or markup) ⇒ post-event
Apply *finger* as a right-hand fingering indication.

\scaleDurations *fraction* (non-negative rational, fraction, or moment) *music* (music) ⇒ music
Multiply the duration of events in *music* by *fraction*.

\sectionLabel *text* (markup) ⇒ music
Mark the beginning of a named passage with *text*, e.g., “Coda”.

This is well suited for use at a section division created with \section, but it does not imply \section and may be used alone.

\segnoMark [*num* (non-negative, exact integer)] ⇒ music
Create a segno mark (or bar line).

num may be 1 for the first segno, 2 for the second, etc., or it may be \default to use the next number in sequence automatically.

If the segnoStyle context property is 'bar-line, a segno bar line is created instead of a segno mark.

\shape *offsets* (list) *item* (key list or music) ⇒ music
Offset control points of *item* by *offsets*.

offsets is a list of number pairs (*x* . *y*) or a list of such lists. Each pair represents an offset to a control point. The ‘y’ value of each pair is scaled by staff space.

If *item* is a string, the result is \once\override for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied.

`\shiftDurations` *dur* (integer) *dots* (integer) *arg* (music) ⇒ music
Change duration of *arg*.

This function walks over all durations and dot counts in *arg*, adding *dur* to the durations and *dots* to the dot counts.

`\single` *overrides* (music) *music* (music) ⇒ music
Convert *overrides* to tweaks and apply them to *music*.

This does not convert `\revert`, `\set` or `\unset`.

`\skip` *arg* (duration-or-music) ⇒ music
Skip over *arg*, which may be music or a duration.

`\slashedGrace` *music* (music) ⇒ music
Create slashed graces from *music*.

This produces slashes through stems, but no slur.

`\slurDashPattern` *dash-fraction* (number) *dash-period* (number) ⇒ music
Set up a custom dash pattern style for slurs.

dash-fraction gives the size of one dash relative to *dash-period*; *dash-period* is the length of one dash plus one space. LilyPond adjusts *dash-period* to produce symmetrical output.

More complex patterns can be achieved by directly manipulating the `Slur .dash-definition` property.

`\staffHighlight` *color* (color) ⇒ music
Start a highlight with color *color*.

`\storePredefinedDiagram` *fretboard-table* (hash table) *chord* (music) *tuning* (pair) *diagram-definition* (string or pair) ⇒ void
Add a predefined fret diagram to *fretboard-table*.

It is defined by *diagram-definition* for the chord pitches *chord* and the string tuning *tuning*.

`\stringTuning` *chord* (music) ⇒ any type
Convert *chord* to a string tuning.

chord must be in absolute pitches and should have the highest string number (generally the lowest pitch) first.

`\styledNoteHeads` *style* (symbol) *heads* (symbol list or symbol) *music* (music) ⇒ music
Set *heads* in *music* to *style*.

`\tabChordRepeats` [*event-types* (list)] *music* (music) ⇒ music
Extend 'q' to also repeat string and fingering information.

This function walks through *music* putting the notes, fingerings and string numbers of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(articulation-event)`.

`\tabChordRepetition` ⇒ void
Include the string and fingering information in a chord repetition.

This function is deprecated; use `\tabChordRepeats` instead.

`\tag` *tags* (symbol list or symbol) *music* (music) ⇒ music
Tag *music* with *tags*.

This function adds the single symbol or symbol list *tags* to the `tags` property of *music* and returns the result.

\tagGroup *tags* (symbol list) ⇒ void
: Define a tag group comprising the symbols in the symbol list *tags*.

  Tag groups must not overlap.

\tagGroupRef *tags* (symbol list) ⇒ any type
: Define a tag group comprising the symbols in the symbol list *tags*.

  It returns the created tag group for further reference.

\temporary *music* (music) ⇒ music
: Make \override reversible with \revert.

  This function makes any \override in *music* replace an existing grob property value only temporarily, restoring the old value when a corresponding \revert is executed. This is achieved by clearing the pop-first property normally set on \overrides.

  An \override/\revert sequence created by using \temporary and \undo on the same music containing overrides will cancel out perfectly or cause a warning.

  Non-property-related music is ignored, warnings are generated for any property-changing music that isn't an \override.

\textEndMark *text* (markup) ⇒ music
: Create a right-aligned text mark using *text*.

\textMark *text* (markup) ⇒ music
: Create a (left-aligned) text mark using *text*.

\tieDashPattern *dash-fraction* (number) *dash-period* (number) ⇒ music
: Set up a custom dash pattern style for ties.

  dash-fraction gives the size of one dash relative to *dash-period*; *dash-period* is the length of one dash plus one space. LilyPond adjusts *dash-period* to produce symmetrical output.

  More complex patterns can be achieved by directly manipulating the Tie .dash-definition property.

\time [*structure* (a number list or a list of them)] *time-sig* (time signature) ⇒ music
: Set the time signature to *time-sig*.

  When *structure* is a plain list of numbers, it is used as-is for beatStructure, and submeasureStructure is left at the default.

  When *structure* is a list of lists, beatStructure is derived by discarding the grouping, and submeasureStructure is derived by summing each group. For example, a *structure* of '((1 2) (4 8)) yields a beatStructure of '(1 2 4 8) and a submeasureStructure of '(3 12).

  time-sig may be a fraction, e.g., 3/4.

  time-sig may also describe a complex time signature as a Scheme expression. Fractions are represented as pairs, (*numerator* . *denominator*), where the denominator is always a number. The numerator is one number or a list of two or more numbers. A list represents concatenation.

  For example, a time signature of (3+1)/8 + 2/4 can be created with \time #'(((3 1) . 8) (2 . 4))

\timeAbbrev *time-sig* (pair) ⇒ music
: Set the time signature to *time-sig*.

  This is like \time *time-sig*, except that it allows abbreviating fractions as lists. For example, a time signature of (3+1)/8 + 2/4 can be created with \timeAbbrev #'((3

`1 8) (2 4))`, and a time signature of (3+2)/8 with either `\timeAbbrev #'((3 2 8))` or the shorter version `\timeAbbrev 3,2,8`.

This is for backward compatibility. Using `\time` instead is recommended.

`\times` *fraction* (fraction, as pair) *music* (music) ⇒ music
Scale *music* in time by *fraction*.

`\tocItem` [*label* (symbol list or symbol)] *text* (markup) ⇒ music
Add *text* as an entry to the table of contents.

This uses the `tocItemMarkup` paper variable markup for formatting and assigns it to *label* if one is provided. If a hierarchy of labels is given, make the current item a child of the corresponding objects.

`\transpose` *from* (pitch) *to* (pitch) *music* (music) ⇒ music
Transpose *music* from pitch *from* to pitch *to*.

`\transposedCueDuring` *what* (string) *dir* (direction) *pitch* (pitch) *main-music* (music) ⇒ music
Create a transposed cue.

This function inserts notes from the part *what* into a `CueVoice` context called `cue`, using the transposition defined by *pitch*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`\transposition` *pitch* (pitch) ⇒ music
Set instrument transposition to *pitch*.

`\tuplet` *ratio* (fraction, as pair) [*tuplet-span* (duration)] *music* (music) ⇒ music
Scale the given *music* to tuplets.

ratio is a fraction that specifies how many notes are played in place of the nominal value: it will be 3/2 for triplets, namely three notes being played in place of two.

If the optional duration *tuplet-span* is specified, it is used instead of `tupletSpannerDuration` for grouping the tuplets. For example,

```
\tuplet 3/2 4 { c8 c c c c c }
```

results in two groups of three tuplets, each group lasting for a quarter note.

`\tupletSpan` [*tuplet-span* (duration)] ⇒ music
Set `tupletSpannerDuration` to the duration *tuplet-span*.

This context property is the length into which `\tuplet` without an explicit tuplet span argument of its own will group its tuplets. To revert to the default of not subdividing the contents of a `\tuplet` command without an explicit tuplet span argument, use

```
\tupletSpan \default
```

`\tweak` *prop* (key list or symbol) *value* (any type) *music* (music) ⇒ music
Add a tweak to *music*.

Layout objects created by *music* get their property *prop* set to *value*. If *prop* has the form *`Grob.property`*, like with

```
\tweak Accidental.color #red cis'
```

an indirectly created grob (`Accidental` is caused by `NoteHead`) can be tweaked; otherwise only directly created grobs are affected.

prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

If *music* is an `event-chord`, every contained `rhythmic-event` is tweaked instead.

\undo *music* (music) ⇒ music
: Convert \override and \set in *music* to \revert and \unset, respectively.

  Any reverts and unsets already in *music* cause a warning. Non-property-related music is ignored.

\unfolded *music* (music) ⇒ music
: Mask *music* until the innermost enclosing repeat is unfolded.

\unfoldRepeats [*types* (symbol list or symbol)] *music* (music) ⇒ music
: Unfold \repeat.

  This forces \repeat volta, \repeat tremolo or \repeat percent commands in *music* to be interpreted as \repeat unfold, if specified in the optional symbol-list *types*. The default for *types* is an empty list, which forces any of those commands in *music* to be interpreted as \repeat unfold. Possible entries are volta, tremolo or percent. Multiple entries are possible.

\versus *music* (music) ⇒ music
: Prepend character U+2123 (VERSICLE) to the lyrics represented by *music*.

\voices *ids* (list of indices or symbols) *music* (music) ⇒ music
: Specify voice order in simultaneous music.

  This takes the key list *ids* of numbers (indicating the use of '\voiceOne'...) or symbols (indicating voice names, typically converted from strings by argument list processing) and assign the following \\-separated music in *music* to contexts according to that list. Named rather than numbered contexts can be used for continuing one voice (for the sake of spanners and lyrics), usually requiring a \voiceOne-style override at the beginning of the passage and a \oneVoice override at its end.

  The default

  ```
  << ... \\ ... \\ ... >>
  ```

  construct would correspond to

  ```
  \voices 1,2,3 << ... \\ ... \\ ... >>
  ```

\void *arg* (any type) ⇒ void
: Accept a Scheme argument *arg* and return a void expression.

  Use this if you want to have a Scheme expression evaluated because of its side effects but its return value being ignored.

\volta *volta-numbers* (number list) *music* (music) ⇒ music
: Mark *music* as being limited to the volte given in *volta-numbers*.

  This gets used when the innermost enclosing repeat is unfolded. Volta number begins at 1 and increases by 1 with each repetition.

\vshape *offsets* (list) *item* (key list or music) ⇒ music
: Like \shape, but additionally show control points for ease of tweaking.

\withMusicProperty *sym* (symbol) *val* (any type) *music* (music) ⇒ music
: Set music property *sym* to *val* in *music*.

\withRelativeDir *file-name* (string) ⇒ any type
: Prepend directory of current input file to string *file-name*.

  Use this for markup commands that include files, and where such files should be found relative to the input file. Example:

  ```
  \markup { \image #X #3 \withRelativeDir "test.png" }
  ```

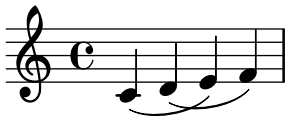
\xNote *note* (music) ⇒ music
: Print *note* with a cross-shaped note head.

\= *id* (index or symbol) *event* (post-event) ⇒ post-event
: Assign an ID to a spanner or an item.

  This sets the `spanner-id` or `id` property of *event* to the given *id*, which is a non-negative integer or a symbol.

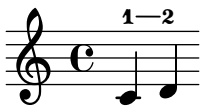
  For spanners this can be used to tell LilyPond how to connect overlapping or parallel slurs or phrasing slurs within a single `Voice` context.

```
\fixed c' { c\=1( d\=2( e\=1) f\=2) }
```



  For items this can be used, for example, to tell LilyPond how to connect a `FingerGlideSpanner` with non-matching fingers.

```
\fixed c' { c\glide \= #'foo -1 d\= #'foo -2 }
```



\% *count* (number) *music* (music) ⇒ music
: This is the same as `\repeat percent` *count music*.

* *count* (number) *music* (music) ⇒ music
: This is the same as `\repeat unfold` *count music*.

B.22 Identificadores de modificación de contextos

Se definen las siguientes instrucciones para su uso como modificaciones de contexto dentro de un bloque `\layout` o `\with`.

`\EnableGregorianDivisiones`
: Configure division commands such as `\section` to create `Divisio` grobs rather than `BarLine` grobs. This does not affect measure bar lines or the properties of the grobs themselves.

  - Sets translator property `caesuraTypeTransform` to `caesura-to-divisio`.
  - Sets translator property `doubleRepeatBarType` to `'()`.
  - Sets translator property `endRepeatBarType` to `'()`.
  - Sets translator property `fineBarType` to `""`.
  - Sets translator property `sectionBarType` to `""`.
  - Sets translator property `startRepeatBarType` to `'()`.
  - Sets translator property `underlyingRepeatBarType` to `""`.
  - Sets translator property `doubleRepeatSegnoBarType` to `"S-||"`.
  - Sets translator property `endRepeatSegnoBarType` to `"S-||"`.
  - Sets translator property `fineSegnoBarType` to `"S-||"`.
  - Sets translator property `fineStartRepeatSegnoBarType` to `"S-||"`.
  - Sets translator property `segnoBarType` to `"S-||"`.
  - Sets translator property `startRepeatSegnoBarType` to `"S-||"`.

`\RemoveAllEmptyStaves`
: Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`, including those in the first system.
  - Sets grob property `remove-empty` in Sección "VerticalAxisGroup" en *Referencia de Funcionamiento Interno* to `#t`.
  - Sets grob property `remove-first` in Sección "VerticalAxisGroup" en *Referencia de Funcionamiento Interno* to `#t`.

`\RemoveEmptyStaves`
: Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`.
  - Sets grob property `remove-empty` in Sección "VerticalAxisGroup" en *Referencia de Funcionamiento Interno* to `#t`.

B.23 Variables del papel

Está predefinidos por LilyPond las siguinetes variables *de papel* o del bloque `paper`. Los valores por omisión que están escalados al tamaño de la hoja, se dan para el tamaño predeterminado, que es A4. Las variables mencionadas *xxx*`-default` son también variables del papel.

Ciertos valores como `mm` o `pt` son constantes y por tanto son esencialmente de solo lectura; si se modifican, LilyPond podría quedar muy confundido.

Casi todas las variables del papel pueden establecerse tanto en el bloque `\paper` como en el bloque `\layout`, véase Sección 27.1 [El bloque `\layout`], página 686.

`annotate-spacing` (boolean)
: If this value is set to `#t`, dimensions of vertical layout variables that may be altered for page formatting are graphically displayed. Default: `#f`.

`auto-first-page-number` (boolean)
: The page-breaking algorithm is affected by the first page number being odd or even. If set to `#t`, the algorithm decides whether to start with an odd or even number. This results in the first page number remaining as is or being increased by one. Default: `#f`.

`binding-offset` (number)
: This amount gets added to `inner-margin` so that nothing is hidden by the binding. Has only an effect if `two-sided` is set to `#t`. Default: `binding-offset-default` (5 mm, scaled to paper size).

`blank-after-score-page-penalty` (number)
: The penalty for having a blank page after the end of one score and before the next. By default, this is smaller than `blank-page-penalty` so that LilyPond prefers blank pages after scores to blank pages within a score. Default: 2.

`blank-last-page-penalty` (number)
: The penalty for ending the score on an odd-numbered page. Default: 0.

`blank-page-penalty` (number)
: The penalty for having a blank page in the middle of a score. Note that this is not used by `ly:optimal-breaking`, which never considers blank pages in the middle of a score. Default: 5.

`blot-diameter` (non-negative number)
: This value globally defines the smallest 'round' feature LilyPond uses while constructing almost all non-glyph elements like beams or stems. Essentially, it sets up how

round the corners and line ends are. It only makes sense to change the value if you use a different music glyph font with crisper corners, say. Default: 0.4 pt.

`book-title` (procedure)
: Internal. This is what LilyPond actually uses for handling `bookTitleMarkup`.

`book-title-properties` (association list (list of pairs))
: Internal.

`bookpart-level-page-numbering` (boolean)
: If set to `#t`, restart page numbering for each `\bookpart` block. Otherwise all pages in the document are enumerated continuously. Default: `#f`.

`bookTitleMarkup` (markup)
: The titling markup within a `\book`, using standard fields from the `\header` block. The default value is defined in file `titling-init.ly`.

`bottom-margin` (non-negative number)
: The margin between the bottom of the printable area and the bottom of the page. Default: `bottom-margin-default` (10 mm, scaled to paper size).

`bp` (positive number)
: The big point unit, also called *desktop publishing point* (DTP point). It is 1/72 of an inch, approx. 0.353 mm (0.0138 in).

`check-consistency` (boolean)
: If set to `#t`, print a warning if the left margin, line width, and right margin do not exactly add up to `paper-width`, and replace each of these (except `paper-width`) with its default value (scaled to the paper size if necessary). If set to `#f`, ignore any inconsistencies and allow systems to run off the edge of the page. Default: `#t`.

`clip-regions` (association list (list of pairs))
: A list of rhythm location pairs to output fragments of a score.

```
clip-regions
= #(list (cons (make-rhythmic-location 2 0 1)
               (make-rhythmic-location 4 0 1))
         (cons (make-rhythmic-location 5 1 2)
               (make-rhythmic-location 7 3 4)))
```

  The above example defines two regions: the first is from the beginning of bar 2 to the beginning of bar 4, the second starts after a half note duration in bar 5 and ends after the third quarter in bar 7.

  Has only an effect if LilyPond's command-line option `-dclip-systems` is active. Default: unset.

`cm` (positive number)
: The centimeter unit, approx. 0.39 in.

`debug-beam-scoring` (boolean)
: If set to `#t`, print demerits together with their cause, followed by the number of configurations that have been scored before concluding. Default: `#f`.

  Example:

  > `L 18.95 C 655.12 c19/625` → demerits for stem lengths (‘`L`’) and collisions (‘`C`’), scored 19 out of 625 initially considered configurations.

  Possible demerit causes: collision (‘`C`’), inappropriate stem length (‘`L`’), beam direction different from damping direction (‘`Sd`’), difference between beam slope and musical slope (‘`Sm`’), deviation from ideal slope (‘`Si`’), horizontal inter-quants (‘`H`’), forbidden quants (‘`Fl`’/‘`Fs`’).

Demerits are configurable, see Sección “beam-interface” en *Referencia de Funcionamiento Interno* for a list of tunable parameters.

`debug-slur-scoring` (boolean)
: If set to `#t`, print demerits together with their cause, followed by the sum of all demerits and the index of the slur configuration finally chosen. Default: `#f`.

  Example:

  > `slope=2.00, R` `edge=10.51, variance=0.03 TOTAL=12.54 idx=4` → demerits for slope, distance of the right edge to the attachment point, variance of distance between note heads and slur. Total demerits: 12.54, index of the chosen configuration: 4.

  Possible demerit causes: distance of the left/right slur edge to the attachment points (‘`L edge`’/‘`R edge`’), inappropriate slope (‘`slope`’), distance variations between note heads and slur (‘`variance`’), distances for heads that are between the slur and an imaginary line between the attachment points (‘`encompass`’), too small distance between slur and tie extrema (‘`extra`’).

  Demerits are configurable, see Sección “slur-interface” en *Referencia de Funcionamiento Interno* for a list of tunable parameters.

`debug-tie-scoring` (boolean)
: If set to `#t`, print the basic configuration of ties, followed by demerits and their corresponding causes and the total sum of demerits. Default: unset.

  Example:

  > `0 (0.23) u: vdist=1.08 lhdist=1.79 tie/stem dir=8.00 TOTAL=10.87` → offset from the center of the staff according tie specification: 0 staff-spaces, vertical distance of the tie’s center in y-direction to the bottom (or top) of the tie: 0.23, direction: up. Demerits for vertical and horizontal distance to note head, same direction of stem and tie. Total demerits: 10.87.

  Possible demerit causes: wrong tie direction (‘`wrong dir`’), vertical distance to note heads (‘`vdist`’), horizontal distance to left or right note head (‘`lhdist`’/‘`rhdist`’), same direction of stem and tie (‘`tie/stem dir`’), position and direction of tie not matching, e.g., tie is in the upper half of the staff but has direction `DOWN` (‘`tie/pos dir`’), tie is too short (‘`minlength`’), tip of tie collides with staff line (‘`tipline`’), collision with dot (‘`dot collision`’), center of tie is too close to a staff line (‘`line center`’), y-position (edge or center) of currently considered tie is less than the y-position of the previous tie (‘`monotone edge`’/ ‘`monotone cent`’), edge or center of tie is too close to the one considered previously (‘`tietie center`’/‘`tietie edge`’), unsymmetrical horizontal positioning with respect to the note heads (‘`length symm`’), unsymmetrical vertical positioning with respect to the note heads (‘`pos symmetry`’).

  Demerits are configurable, see Sección “tie-interface” en *Referencia de Funcionamiento Interno* for a list of tunable parameters.

`dimension-variables` (list)
: Internal.

`evenFooterMarkup` (markup)
: The footer markup used for even-numbered pages. If not set, `oddFooterMarkup` is used instead. The default value is defined in file `titling-init.ly`.

`evenHeaderMarkup` (markup)
: The header markup used for even-numbered pages. If not set, `oddHeaderMarkup` is used instead. The default value is defined in file `titling-init.ly`.

`first-page-number` (integer)
: The value of the page number on the first page. Default: 1.

`footnote-footer-padding` (number)
: The padding between the footer and the bottom-most footnote. Default: 0.5 mm.

`footnote-number-raise` (number)
: This controls how high the annotation numbers of both footnotes and in-notes are raised relative to the footnote or in-note text. Default: 0.5 mm.

`footnote-numbering-function` (procedure)
: This variable holds the name of the function that formats both footnote and in-note numbers (without positioning it). LilyPond provides two predefined functions (in file `output-lib.scm`): `numbered-footnotes` ('1', '2', etc.) and `symbol-footnotes` ('*', '†', etc.). Default: `numbered-footnotes`.

  The function takes an integer as an argument (starting with value 0) and returns the appropriate markup. You can create your own function, for example,

```
footnote-numbering-function =
#(lambda (x)
  #{ \markup \concat { "[" #(number->string (1+ x)) "]" } #})
```

`footnote-padding` (number)
: The padding between two footnotes. Default: 0.5 mm.

`footnote-separator-markup` (markup)
: Markup to separate the music from the footnotes, usually a horizontal line. The default value is defined in file `paper-defaults-init.ly`.

`horizontal-shift` (number)
: If set, the main content block of all pages (but not headers and footers) is shifted horizontally. Positive values shift to the right (on both even and odd pages). Default: 0 mm.

`in` (positive number)
: The inch unit, equal to 2.54 cm.

`in-note-padding` (number)
: The padding between two in-notes. Default: 0.5 mm.

`in-note-system-padding` (number)
: The padding between an in-note and the associated music system. Default: 0.5 mm.

`incipit-width` (positive number)
: The width of an incipit as created by the `\incipit` command. Must be smaller than or equal to `indent`. Default: `(indent / 2)`.

`indent` (non-negative number)
: The indentation of the first system in a score. The space within the `line-width` available for the first system is reduced by this amount. Default: `indent-default` (15 mm, scaled to paper size).

`inner-margin` (non-negative number)
: The margin all pages have at the inner side if they are part of a book. Has only an effect if `two-sided` is set to `#t`. If unset, `inner-margin-default` (15 mm, scaled to paper size) is used in computations. Default: unset.

`is-last-bookpart` (boolean)
: To be documented.

`label-alist-table` (list)
: Internal. LilyPond uses this to construct a table of contents.

`label-page-table` (association list (list of pairs))
: Internal. LilyPond uses this to implement the `\with-link` functionality and to construct a table of contents.

`landscape` (boolean)
: Internal. Use `set-default-paper-size` or `set-paper-size` to set the paper orientation.

`last-bottom-spacing` (association list (list of pairs))
: The distance from the last system or top-level markup on a page to the bottom of the printable area (i.e., the top of the bottom margin). The default value is defined in file `paper-defaults-init.ly`.

`left-margin` (non-negative number)
: The margin between the left edge of the page and the start of the staff lines in unindented systems. If `left-margin` is not set, and both `line-width` and `right-margin` are set, then `left-margin` is set to `(paper-width - line-width - right-margin)`. If only `line-width` is set, then both margins are set to `((paper-width - line-width) / 2)`, and the systems are consequently centered on the page. If unset, `left-margin-default` (15 mm, scaled to paper size) is used in computations. Default: unset.

  If `two-sided` is set to `#t`, this value is ignored. Also see `check-consistency`.

`line-thickness` (positive number)
: This value globally defines the default line thickness LilyPond uses while constructing many non-glyph elements like lines or boxes. The actual line thickness is usually specified as a factor of this base value, either as a default value, or overridden by the user. The default depends on the staff space and is computed as `(0.328571 + 0.0342857 * staff-space)` (see function `calc-line-thickness` in file `paper.scm`), which is synchronized with the Emmentaler font.

`line-width` (positive number)
: The horizontal extent of the staff lines in unindented, non-ragged systems, equal to `(paper-width - left-margin - right-margin)` if not set. If `line-width` is set, and both `left-margin` and `right-margin` are not set, then the margins are updated to center the systems on the page automatically. Also see `check-consistency`. Default: unset.

  As a feature of LilyPond, if the command-line option `-dcrop` is set, the dimensions of the cropped output file are always rounded up to integer (PostScript) big points; this might cause a little bit of whitespace at the right margin. To avoid that, set `line-width` to an integer big point value like `450\bp`.

`make-footer` (procedure)
: Internal. This is what LilyPond actually uses for for handling `evenFooterMarkup` and `oddFooterMarkup`.

`make-header` (procedure)
: Internal. This is what LilyPond actually uses for for handling `evenHeaderMarkup` and `oddHeaderMarkup`.

`markup-markup-spacing` (association list (list of pairs))
: The distance between two (title or top-level) markups. The default value is defined in file `paper-defaults-init.ly`.

`markup-system-spacing` (association list (list of pairs))
: The distance between a (title or top-level) markup and the system that follows it. The default value is defined in file `paper-defaults-init.ly`.

`max-systems-per-page` (non-negative, exact integer)
: The maximum number of systems that are placed on a page. This is currently supported only by the `ly:optimal-breaking` algorithm. Default: unset.

`min-systems-per-page` (non-negative, exact integer)
: The minimum number of systems that are placed on a page. This may cause pages to be overfilled if it is made too large. This is currently supported only by the `ly:optimal-breaking` algorithm. Default: unset.

`mm` (positive number)
: The millimeter unit, approx. 0.039 in.

`number-footnote-table` (list)
: Internal. LilyPond uses this to manage footnotes and footnote numbers.

`oddFooterMarkup` (markup)
: The footer markup used for odd-numbered pages. The default value is defined in file `titling-init.ly`.

`oddHeaderMarkup` (markup)
: The header markup used for odd-numbered pages. The default value is defined in file `titling-init.ly`.

`orphan-penalty` (number)
: To be documented. See `lily/page-breaking.cc`. Default is 100000.

`outer-margin` (non-negative number)
: The margin all pages have at the outer side if they are part of a book. Has only an effect if `two-sided` is set to `#t`. If unset, `outer-margin-default` (15 mm, scaled to paper size) is used in computations. Default: unset.

`output-filename` (string)
: If set, use this value as the output file name for LilyPond output. See also `output-suffix`. The default is the input file name (without the file extension).

`output-scale` (positive number)
: Internal. This value globally defines the output scale LilyPond uses while creating output. Use `set-global-staff-size` or `layout-set-staff-size` to change the scaling. Default: 1.7573, which corresponds to a 20 pt staff size.

`output-suffix` (string)
: If set, use this value as the output file name suffix for LilyPond output, appended to the output name with a dash in between. See also `output-filename`. The default is no output suffix if there is only a single book block (either implicit or explicit), and a running integer starting with value 1 otherwise.

`page-breaking` (procedure)
: The page-breaking algorithm to use. Choices are `ly:minimal-breaking`, `ly:page-turn-breaking`, `ly:one-page-breaking`, `ly:one-line-breaking`, `ly:one-line-auto-height-breaking`, and `ly:optimal-breaking`. Default: `ly:optimal-breaking`.

`page-breaking-system-system-spacing` (association list (list of pairs))
: This tricks the page breaker into thinking that `system-system-spacing` is set to something different than it really is. For example, if

page-breaking-system-system-spacing.padding is set to something substantially larger than system-system-spacing.padding, then the page breaker puts fewer systems on each page. Default: unset.

page-count (non-negative, exact integer)
: The number of pages to be used for a score. Default: unset.

page-number-type (symbol)
: The type of numerals used for page numbers. Choices include arabic, roman-ij-lower, roman-ij-upper, roman-lower, and roman-upper. Default: arabic.

page-post-process (procedure)
: If this function is defined (within the \paper block), LilyPond uses it to post-process pages, for example, to extract the table of contents and writing the information to an auxiliary file. The call happens after page breaking has been performed.

  The syntax is

  ```
  #(define (page-post-process layout pages) ... )
  ```

page-spacing-weight (number)
: When using the ly:optimal-breaking algorithm for page breaking, LilyPond has to make trade-offs between horizontal and vertical stretching so that the overall spacing is more acceptable. This parameter controls the relative importance of (vertical) page spacing and (horizontal) line spacing. High values makes page spacing more important. Default: 10.

paper-height (positive number)
: The height of the page. Note that the automatic scaling of some vertical dimensions is not affected if you set this value directly. Default: 297 mm (A4 paper height).

paper-width (positive number)
: The width of the page. While setting paper-width directly has no effect on the automatic scaling of some horizontal dimensions, it does influence the line-width variable. If both paper-width and line-width are set, then left-margin and right-margin also get updated. Also see check-consistency. Default: 210 mm (A4 paper width).

papersizename (string)
: Internal. Use set-default-paper-size or set-paper-size to set the paper size.

print-all-headers (boolean)
: If set to #t, print all headers for each \score in the output. If set to #f, only the piece and opus header variables are printed. Default: #f.

print-first-page-number (boolean)
: If set to #t, a page number is printed on the first page. Default: #f.

print-page-number (boolean)
: If set to #f, page numbers are not printed. Default: #t.

property-defaults (association list (list of pairs))
: This variable holds some settings that are used for top-level markups and as fallback values if nothing else is specified, for example, the standard fonts or the baseline skip. The default value is defined in file paper-defaults-init.ly.

pt (positive number)
: The point unit, equal to approx. 0.351 mm (0.0139 in).

`ragged-bottom` (boolean)
: If set to `#t`, systems are set at at their natural spacing, neither compressed nor stretched vertically to fit the page. Default: `#f`.

`ragged-last` (boolean)
: If set to `#t`, the last system in the score does not fill the line width. Instead, the last system ends at its natural horizontal length. Default: `#f`.

`ragged-last-bottom` (boolean)
: If set to `#f`, then the last page, and the last page in each section created with a `\bookpart` block, is vertically justified in the same way as the earlier pages. Default: `#t`.

`ragged-right` (boolean)
: If set to `#t`, systems don't fill the line width. Instead, systems end at their natural horizontal length. Default: `#t` for scores with only one system, and `#f` for scores with two or more systems.

`reset-footnotes-on-new-page` (boolean)
: If set to `#t`, footnote and in-note numbers are reset on each page break. For footnotes and in-notes numbered consecutively across page breaks, set to `#f`. Default: `#t`.

`right-margin` (non-negative number)
: The margin between the right edge of the page and the end of the staff lines in non-ragged systems. If `right-margin` is not set, and both `line-width` and `left-margin` are set, then `right-margin` is set to `(paper-width - line-width - left-margin)`. If only `line-width` is set, then both margins are set to `((paper-width - line-width) / 2)`, and the systems are consequently centered on the page. If unset, `right-margin-default` (15 mm, scaled to paper size) is used in computations. Default: unset.

  If `two-sided` is set to `#t`, this value is ignored. Also see `check-consistency`.

`score-markup-spacing` (association list (list of pairs))
: The distance between the last system of a score and the (title or top-level) markup that follows it. The default value is defined in file `paper-defaults-init.ly`.

`score-system-spacing` (association list (list of pairs))
: The distance between the last system of a score and the first system of the score that follows it, if no (title or top-level) markup exists between them. The default value is defined in file `paper-defaults-init.ly`.

`score-title` (procedure)
: Internal. This is what LilyPond actually uses for handling `scoreTitleMarkup`.

`score-title-properties` (association list (list of pairs))
: Internal.

`scoreTitleMarkup` (markup)
: The titling markup of a score within a `\book`, using standard fields from the `\header` block. The default value is defined in file `titling-init.ly`.

`short-indent` (non-negative number)
: The level of indentation for all systems in a score besides the first system. The space within the `line-width` available for systems other than the first one is reduced by this amount. Default: `short-indent-default` (0 mm).

`staff-height` (positive number)
: Internal. Use `set-global-staff-size` or `layout-set-staff-size` to set the staff height. Default: 20 pt.

staff-space (positive number)
: Internal. One staff height contains four staff spaces. Use set-global-staff-size or layout-set-staff-size to set this value. Default: 5 pt.

system-count (non-negative, exact integer)
: The number of systems to be used for a score. Default: unset.

system-separator-markup (markup)
: A markup object that is inserted between systems, often used for orchestral scores. The \slashSeparator markup, defined in file titling-init.ly, is provided as a sensible standard value. Default: unset.

system-system-spacing (association list (list of pairs))
: The distance between two systems in the same score. The default value is defined in file paper-defaults-init.ly.

systems-per-page (non-negative, exact integer)
: The number of systems that should be placed on a page. This is currently supported only by the ly:optimal-breaking algorithm. Default: unset.

tagline (markup)
: This markup gets placed at the bottom of the last page (via oddFooterMarkup). The default value is defined in file titling-init.ly.

text-font-size (positive number)
: This value globally defines the standard text size for markups. The default depends on the staff height and is computed as (staff-height / 20 * 11).

tocFormatMarkup (procedure)
: How the top-level entries of the table of contents are formatted (if there are several hierarchical levels). This variable holds a function like make-bold-markup. The default value is defined in file toc-init.ly.

tocIndentMarkup (markup)
: How the outline's hierarchy in the table of contents is made apparent. This markup is printed zero, one, or several times depending on the level of each entry. The default value is defined in file toc-init.ly.

tocItemMarkup (markup)
: How an item in the table of contents is formatted. The default value of this markup is defined in file toc-init.ly.

tocTitleMarkup (markup)
: How the title of the table of contents is formatted. The default value of this markup is defined in file toc-init.ly.

top-margin (non-negative number)
: The margin between the top of the page and the top of the printable area. Default: top-margin-default (10 mm, scaled to paper size).

top-markup-spacing (association list (list of pairs))
: The distance from the top of the printable area (i.e., the bottom of the top margin) to the first (title or top-level) markup on a page, when there is no system between the two. The default value is defined in file paper-defaults-init.ly.

top-system-spacing (association list (list of pairs))
: The distance from the top of the printable area (i.e., the bottom of the top margin) to the first system on a page, when there is no (title or top-level) markup between the two. The default value is defined in file paper-defaults-init.ly.

`two-sided` (boolean)
: If set to `#t`, use `inner-margin`, `outer-margin` and `binding-offset` to determine margins depending on whether the page number is odd or even. `left-margin` and `right-margin` are then ignored. Default: `#f`.

B.24 Convenciones de nomenclatura

Se usan unas convenciones de nomenclatura para que sea más sencillo retener en la cabeza la naturaleza de un bojeto.

Objeto o tipo de propiedad	**Convención de nomenclatura**	**Ejemplos**
Palabras clave	`aaaa`	`\score`, `\addlyrics`
Funciones musicales y abreviaciones	`aaaa` o `aaaaAaaaAaaa`	`\time`, `\stemUp`
Funciones de marcado	`aaaa` o `aaaa-aaaa-aaaa`	`\bold`, `\with-dimensions`
Variables del papel	`aaaa` o `aaaa-aaaa-aaaa`	`indent`, `page-breaking`
Expresiones musicales	`Aaaa` o `AaaaAaaaAaaa`	`NoteEvent`, `SequentialMusic`
Clases musicales	`aaaa` o `aaaa-aaaa-aaaa`	`note-event`, `rhythmic-event`
Propiedades musicales	`aaaa` o `aaaa-aaaa-aaaa`	`force-accidental`, `iterator-ctor`
Contextos	`Aaaa` o `AaaaAaaaAaaa`	`Staff`, `GrandStaff`
Propiedades de contexto	`aaaa` o `aaaaAaaaAaaa`	`stanza`, `instrumentName`
Grabadores	`Aaaa_engraver` o `Aaaa_aaaa_aaaa_engraver`	`Clef_engraver`, `Note_heads_engraver`
Objetos de presentación	`Aaaa` o `AaaaAaaaAaaa`	`Slur`, `NoteHead`
Propiedades de los objetos de presentación	`aaa` o `aaa-aaa-aaa`	`direction`, `beam-thickness`
Interfaces	`aaa-aaa-interface`	`grob-interface`, `break-aligned-interface`
Funciones de Scheme definidas dentro de Guile	`aaaa` o `aaaa-aaaa-aaaa`	`map`, `for-each`
Funciones de Scheme definidas dentro de LilyPond	`ly:aaaa` o `ly:aaaa-aaaa-aaaa` or `aaaa` o `aaaa-aaaa-aaaa`[1]	`ly:round-filled-box`, `music-map`

B.25 Predicados de tipo predefinidos

Predicates return `#t` (true) if their argument is of the named type and `#f` (false) if it isn't.

B.25.1 R5RS primary predicates

Primary predicates can be applied to any expression. They can be used on their own as predicates for LilyPond functions. The predicates here are part of the Scheme standard R5RS.

Type predicate	**Description**
`boolean?`	boolean
`char?`	character
`complex?`	complex number
`eof-object?`	end-of-file object

[1] Por razones históricas, algunas pero no todas las funciones definidas por LilyPond comienzan con el prefijo `ly:` .

`input-port?`	input port
`integer?`	integer
`list?`	list (use `cheap-list?` for faster processing)
`null?`	null
`number?`	number
`output-port?`	output port
`pair?`	pair
`port?`	port
`procedure?`	procedure
`rational?`	rational number
`real?`	real number
`string?`	string
`symbol?`	symbol
`vector?`	vector

B.25.2 R5RS secondary predicates

Secondary predicates are only applicable to specific expressions (for example, to numbers). They will throw a type error when applied to expressions they are not intended for. The predicates here are part of the Scheme standard R5RS.

Type predicate	**Description**
`char-alphabetic?`	alphabetic character
`char-lower-case?`	lower-case character
`char-numeric?`	numeric character
`char-upper-case?`	upper-case character
`char-whitespace?`	whitespace character
`even?`	even number
`exact?`	exact number
`inexact?`	inexact number
`negative?`	negative number
`odd?`	odd number
`positive?`	positive number
`zero?`	zero

B.25.3 Guile predicates

These predicates are defined by Guile but are not part of a Scheme standard.

Type predicate	**Description**
`hash-table?`	hash table

B.25.4 LilyPond scheme predicates

These predicates are only available within LilyPond and defined in Scheme.

Type predicate	**Description**
`alist?`	association list (list of pairs)
`boolean-or-symbol?`	boolean or symbol
`cheap-list?`	list (use this instead of `list?` for faster processing)
`color?`	color
`exact-rational?`	an exact rational number
`fraction?`	fraction, as pair
`grob-list?`	list of grobs
`grouped-number-list?`	list of non-empty lists of numbers
`index?`	non-negative, exact integer

index-or-markup?	index or markup
key?	index or symbol
key-list?	list of indices or symbols
key-list-or-music?	key list or music
key-list-or-symbol?	key list or symbol
ly:skyline-pair?	pair of skylines
markup?	markup
markup-command-list?	markup command list
markup-list?	markup list
moment-pair?	pair of moment objects
musical-length?	non-negative exact rational, fraction (as pair), moment, or +inf.0
musical-length-as-moment?	non-negative moment with no grace part
musical-length-as-number?	non-negative exact rational or +inf.0
non-negative-number?	non-negative number
number-list?	number list
number-or-grob?	number or grob
number-or-number-pair?	number or pair of numbers
number-or-pair?	number or pair
number-or-string?	number or string
number-pair?	pair of numbers
number-pair-list?	list of number pairs
optionally-grouped-beat-structure?	a number list or a list of them
positive-exact-rational?	a positive, exact rational number
positive-fraction?	positive, finite fraction, as pair
positive-musical-length?	positive exact rational, fraction (as pair), moment, or +inf.0
positive-musical-length-as-moment?	positive moment with no grace part
positive-musical-length-as-number?	positive exact rational or +inf.0
positive-number?	positive number
rational-or-procedure?	an exact rational or procedure
rhythmic-location?	rhythmic location
sane-simple-time-signature?	simple time signature
sane-time-signature?	time signature
scale?	non-negative rational, fraction, or moment
scheme?	any type
string-or-music?	string or music
string-or-pair?	string or pair
string-or-symbol?	string or symbol
symbol-key-alist?	alist, with symbols as keys
symbol-list?	symbol list
symbol-list-or-music?	symbol list or music
symbol-list-or-symbol?	symbol list or symbol
time-signature?	time signature
void?	void

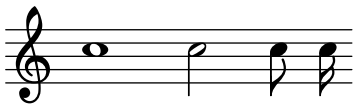
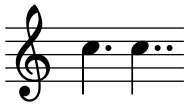
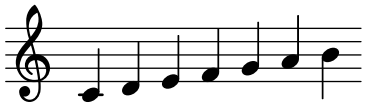
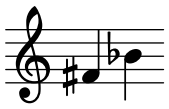
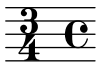
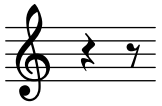
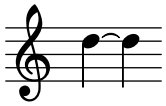
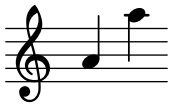
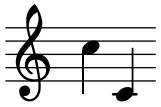
B.25.5 LilyPond exported predicates

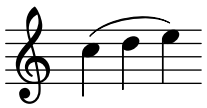
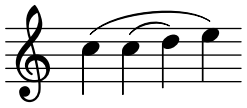
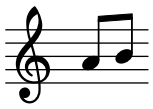
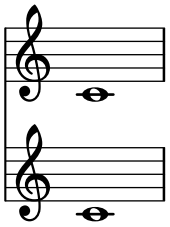
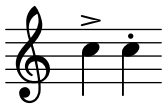
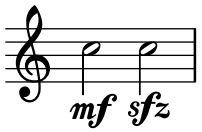
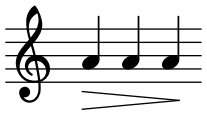
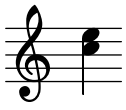
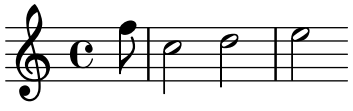
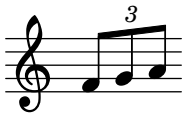
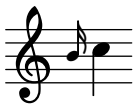
These predicates are only available within LilyPond and usually defined in C++.

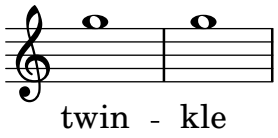
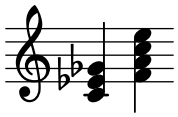
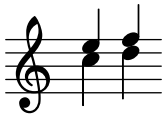
Type predicate	**Description**
ly:book?	book
ly:context?	context

`ly:context-def?`	context definition
`ly:context-mod?`	context modification
`ly:dimension?`	dimension, in staff space
`ly:dir?`	direction
`ly:dispatcher?`	dispatcher
`ly:duration?`	duration
`ly:event?`	post-event
`ly:font-metric?`	font metric
`ly:grob?`	graphical (layout) object
`ly:grob-array?`	array of grobs
`ly:grob-properties?`	grob properties
`ly:input-location?`	input location
`ly:item?`	item
`ly:iterator?`	iterator
`ly:lily-lexer?`	lily-lexer
`ly:lily-parser?`	lily-parser
`ly:listener?`	listener
`ly:moment?`	moment
`ly:music?`	music
`ly:music-function?`	music function
`ly:music-list?`	list of music objects
`ly:music-output?`	music output
`ly:note-scale?`	note scale
`ly:otf-font?`	OpenType font
`ly:output-def?`	output definition
`ly:page-marker?`	page marker
`ly:pango-font?`	Pango font
`ly:paper-book?`	paper book
`ly:paper-system?`	paper-system Prob
`ly:pitch?`	pitch
`ly:prob?`	property object
`ly:score?`	score
`ly:skyline?`	skyline
`ly:source-file?`	source file
`ly:spanner?`	spanner
`ly:spring?`	spring
`ly:stencil?`	stencil
`ly:stream-event?`	stream event
`ly:transform?`	coordinate transform
`ly:translator?`	translator
`ly:translator-group?`	translator group
`ly:unpure-pure-container?`	unpure/pure container

Apéndice C Hoja de referencia rápida

Sintaxis	**Descripción**	**Ejemplo**
`1 2 8 16`	duraciones	
`c4. c4..`	puntillos	
`c d e f g a b`	escala	
`fis bes`	alteración	
`\clef treble \clef bass`	claves	
`\time 3/4 \time 4/4`	compás	
`r4 r8`	silencio	
`d ~ d`	ligadura de unión	
`\key es \major`	armadura	
nota`'`	subir una octava	
nota`,`	bajar una octava	

`c( d e)`	ligadura de expresión	
`c\( c( d) e\)`	ligadura de fraseo	
`a8[ b]`	barra	
`<< \new Staff ... >>`	más pentagramas	
`c-> c-.`	articulaciones	
`c2\mf c\sfz`	matices dinámicos	
`a\< a a\!`	crescendo	
`a\> a a\!`	decrescendo	
`< >`	acorde	
`\partial 8`	anacrusa o compás incompleto	
`\tuplet 3/2 {f g a}`	tresillos	
`\grace`	mordentes	

`\lyricmode { twinkle }`	escribir la letra	twinkle
`\new Lyrics`	imprimir la letra	twinkle
`twin -- kle`	separador de sílabas	
`\chordmode { c:dim f:maj7 }`	acordes	
`\new ChordNames`	imprimir los nombres de los acordes	C° F△
`<<{e f} \\ {c d}>>`	polifonía	
`s4 s8 s16`	silencios de separación	

Apéndice D GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
https://fsf.org/

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTEX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one

stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `https://www.gnu.org/licenses/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Apéndice E Índice de instrucciones y conceptos

Además de todas las instrucciones y palabras clave de LilyPond, este índice es una lista de términos musicales y las palabras que tienen relación con cada uno de ellos, con enlaces a aquellas secciones del manual que describen o se ocupan de dicho término.

!

! 8
\! 160

"

"|" 141

#

#f (false) 978
#t (true) 978

%

% 591, 596
\% 211, 968
%{ ... %} 591, 596

’

' 3

(

(..... 171
\(..... 175

)

) 171
\) 175

*

* 191, 968

,

, ... 3

—

-! 934
-+ 935
-- 361, 934
-. 934
-> 934
-^ 934
- 156, 537, 783
-_ 934

.

. 54

/

/+ 514
/ 514

:

: 214, 511

<

<...> 218
<< ... \\ ... >> 226
<< ... >> 222
<< 222
<> 162, 219, 417
< 218
\< 160

=

= 13
\= 171, 968

>

>> 222
> 218
\> 160

?

? 8

[

[..... 114
\[..... 542

\

\\ 226

]

] 114
\] 542

^

^ 514, 783

_

_ 357, 783
__ 361

|

| 141

~

~ 63, 357

1

15ma 27
15mb 27

8

8va 27
8vb 27

A

a due, partes 235
abajo, arco 430
abajo, dirección (_) 783
abierta, articulación 156, 499, 935
abierto, high hat 156, 937
\abs-fontsize 327, 815
\abs-hspace 827
\abs-vspace 827
absoluta, especificación, de la octava 3
absoluta, introducción, de la octava 3
absoluta 3
\absolute 950
absolutos, matices 160
acampanado, regulador 166
'accent', articulación 156
\accent 156, 934
'accentus', articulación gregoriana 554
'accentus', articulación, gregoriana 936
\accentus 554, 936
\accepts 758, 759, 762
acciaccatura, de varias notas 152
acciaccatura 148
\acciaccatura 148, 950
accidental, alteración, glifos alternativos 40
accidental 8
\accidental 856
\accidentalStyle 32, 950
AccidentalSuggestion 159
acento, articulación 156, 934
acorde alterado 512
acorde arpegiado 183
acorde de jazz 510, 518
acorde extendido 512
acorde quebrado 183
acorde, alteración de notas 513
acorde, alteraciones dentro de 39
acorde, bajo de 514
acorde, corchete de pentagrama cruzado 423
acorde, corchete de 185
acorde, de pentagrama cruzado 422
acorde, de potencia 486
acorde, de quinta vacía 486
acorde, diagrama automático de 480
acorde, diagramas de trastes con nombres de 470
acorde, digitación 285
acorde, división entre pentagramas con \autoChange 421
acorde, eliminar repetidos 517
acorde, en bloque 185
acorde, formas de, en instrumentos de trastes 473
acorde, glissando en tablatura 452
acorde, introducción relativa de la octava en 7
acorde, inversión 515
acorde, ligadura 185
acorde, mayor 7, disposición 522
acorde, modificación de una nota 773
acorde, modo de 510
acorde, nombre alternativo 518
acorde, nombre de 515
acorde, nombres de, excepciones 521
acorde, nombres de 510
acorde, realización 515
acorde, rejilla 524
acorde, repetición de volta, por debajo 522
acorde, rotación 515
acorde, separador, modificación 523
acorde, supresión en 514
acorde, suprimir repetido 477
acorde, vacío 150, 162, 219, 417
acorde, variantes de 511
acorde, y altura relativa 219
acorde, y ligadura 64
acorde 218, 510, 515
acordeón, cambios en el 426
acordeón, símbolo de registro 426
acordeón, signo de cambio en el 426
acordeón, signo de discanto 426
acordeón 426
acordes, diagramas de 459
acordes, repetición de 220, 438
acortar, casillas de primera y segunda vez 203
add-grace-property 150
add-stem-support 286
add-toc-item! 625
\addChordShape 473, 950
adding white background, to text 889
\addInstrumentDefinition 950
additionalPitchPrefix 519
\addlyrics 351, 353, 354
\addQuote 268, 950
\addToTagGroup 636, 950
adición en acordes 513
adicional, evitar becuadro 9
adicional, interna, línea 252
adicional, línea 252
adicional, modificar línea 252
adicional, voz, en música polifóncia 233
adicional, voz, para gestionar saltos 695
aditivo, indicación de compás 80
adorno, letra y notas de 385
adorno, nota de, seguimiento 148
adorno, nota, trucaje 150
adorno, nota, y espaciado estricto 151
adorno, sincronización de notas de 152
\aeolian 25
afinación del laúd 489
afinación, del banjo 488
afinación, no occidental 573
\after 161, 212, 950
\afterGrace 148, 950

afterGraceFraction 948
agogó 937
agrupación de grupes especiales 56
agrupar, pulsos 111
Aiken, cabezas de nota con forma 48
Aiken, cabezas de nota, variante fina 50
\aikenHeads 48
\aikenHeadsMinor 48
\aikenThinHeads 48
\aikenThinHeadsMinor 48
aislada, altura 54
aislada, duración 54
ajustado a palabras, texto 335
ajustar comienzo y final, ligadura de expresión ... 173
ajuste de línea, texto con 335
al aire, indicación de cuerda 430
al coda 198
al fine 198
al niente, regulador 166
alfa, transparencia 292
\alias 758
\align-on-other 828
alignAboveContext 259, 364, 763
alignBelowContext 259, 364, 373, 763
alineación a un objeto 804
alineación a una cadenza (cadencia) 153
alineación vertical de matices, romper 167
alineación vertical de matices 167
alineación vertical de textos, romper 167
alineación vertical de textos 167
alineación, compás, número 138
alineación, horizontal, letra 369
alineación, horizontal, melisma 369
alineación, texto 336
alineación, vertical, de texto 331
alinear, elementos de marcado 330
alinear, texto de marcado 330
alinear, texto 330
alist 946
alla breve 80, 81
alla coda 198
\allowBreak 120, 692
\allowPageTurn 699, 950
\allowVoltaHook 950
alteración con paréntesis 8
alteración de cuartos de tono 10
alteración de precaución 8
alteración en nota ligada 9
alteración mensural 549
alteración no restablecer, estilo de 38
alteración recordatoria 8
alteración, accidental 32
alteración, automática 32
alteración, bajo cifrado, glifos 532
alteración, bajo cifrado, posición 532
alteración, cadencias 91
alteración, conjunto de glifos 927
alteración, coral con precaución 36
alteración, en acordes 39
alteración, en varias voces 35
alteración, estilo coral 36
alteración, estilo de piano 35
alteración, estilo de voz de precaución moderno ... 35
alteración, estilo de voz de 34
alteración, estilo de, olvidar 38
alteración, estilo de 32
alteración, estilo dodecafónico sin repetir 37
alteración, estilo dodecafónico-primero 38
alteración, estilo moderno de precaución 35
alteración, estilo moderno de 35
alteración, estilo moderno 34
alteración, estilo predeterminado de 32
alteración, estilo, dodecafónico 37
alteración, glifos alternativos 40
alteración, música sin metro 91
alteración, moderna 35
alteración, ocultar, en notas ligadas al principio de un sistema 9
alteración, piano de precaución, estilo de 36
alteración, trino con 188
alteración, y notas simultáneas 39
alteraciones de la musica ficta 549
alteraciones estilo moderno 34
alteraciones gregorianas 553
alteraciones kievanas 564
alteraciones, estilo de, neo-moderno 36
alteraciones, estilo predeterminado 34
alteraciones, estilo teaching (enseñanza) 38
alteraciones, estilo, moderno-precaución 34
alteraciones, estilo, neo-moderno de precaución 37
alteraciones, estilo, neo-moderno de voz 37
alteraciones,estilo, neo-modern-voice-cautionary ... 37
\alterBroken 786, 951
alternativa, tabla de trastes 478
\alternative 192, 194
alternativo, estilo de la breve 55
alternativo, final, con ligaduras de unión 64
alternativo, nombre de acorde 518
alternativos, finales 192
alterno, indicación de compás 80
alto varC, clave 928
alto, clave 20, 928
altura aproximada 10
altura, aislada 54
altura, extrema 10
altura, nombres de, otros idiomas 10
altura, predeterminada 54
altura, sin especificar 10
altura, transporte 14
altura, transposición 'inteligente' 15
altura 3
amalgama, indicación de compás de 80
'Amazing Grace', ejemplo de gaita 502
ambitus, colocación 44
ambitus 41
ambitusAfter 44, 951
AmbitusLine 43
análisis musicológico 306
análisis, corchete de, con etiqueta 307
anacrusa, en una repetición 192
anacrusa 90
analizador léxico para LilyPond 947
analizador sintáctico (parser) 948
ángulo, regulador en 801
anidada, repetición 206
anidados, corchetes de pentagrama 250
anidamiento, de pentagramas 250
annotate moving by spacing, in text 879
\annotate-moving 879
annotate-spacing 734, 969

anthem ... 391
antigua, música, clave ... 930
antiguas, claves ... 20
apilados, corchetes ... 116
\append-to-tag ... 638, 880
\appendToTag ... 637, 951
\appendToTagMarkup ... 639, 951
\applyContext ... 748, 951
\applyMusic ... 951
\applyOutput ... 951
\applySwing ... 667
\applySwingWithOffset ... 667
appoggiatura ... 148
\appoggiatura ... 148, 951
\approximatePitch ... 10, 951
aproximada, altura ... 10
árabe, armadura de tonalidad ... 575
árabe, ejemplo de música ... 578
árabe, improvisación ... 578
árabe, música ... 573
árabe, maqam ... 573
árabe, notas, nombres ... 574
árabe, plantilla de música ... 578
árabe, semi-bemol ... 575
árabes, armaduras ... 578
\arabicStringNumbers ... 434
archivos, empotrar en PDF ... 652
'arco abajo', indicación de arco ... 935
arco abajo ... 156, 935
'arco arriba', indicación de arco ... 935
arco arriba ... 156, 935
arco, abajo, indicación ... 430
arco, arriba, indicación ... 430
arco, indicación ... 430
armónico, en instrumento de trastes ... 485
armónico, en tablaturas ... 448
armónico, flageolet ... 156
armónico, natural ... 430
armónicos artificiales ... 431
armónicos de flageolet ... 935
armónicos en tablatura ... 441
armónicos, cabezas de nota ... 44
armadura de la tonalidad ... 25
armadura gregoriana ... 553
armadura mensural ... 549
armadura, cancelación, antes de la barra ... 724
armadura, evitar becuadros ... 26
armadura, no tradicional ... 26
armadura, visibilidad después de un cambio explícito ... 798
arpa sacra, cabezas de nota con forma ... 48
arpa, diagrama de pedal ... 428
arpa, pedal ... 428
arpa ... 427
arpeggio-direction ... 183
\arpeggio ... 183
\arpeggioArrowDown ... 183, 951
\arpeggioArrowUp ... 183, 951
\arpeggioBracket ... 184, 951
\arpeggioNormal ... 183, 951
\arpeggioParenthesis ... 184, 951
\arpeggioParenthesisDashed ... 184, 951
arpegiado, acorde, símbolos especiales ... 184
arpegio con paréntesis de pentagrama cruzado ... 187
arpegio, entre voces ... 186
arpegio, y ligadura ... 66
arpegio ... 183
arranger ... 601
arrastre ... 290
arriba, arco ... 430
arriba, dirección (^) ... 783
\arrow-head ... 337, 845
articulación accent ... 156
articulación marcato ... 156
articulación portato ... 156
articulación staccatissimo ... 156
articulación staccato ... 156
articulación tenuto ... 156
articulación, abierta ... 156, 499, 935
articulación, acento ... 156, 934
articulación, espressivo ... 156, 934
articulación, gregoriana, accentus ... 936
articulación, gregoriana, circulus ... 936
articulación, gregoriana, ictus ... 936
articulación, gregoriana, semicirculus ... 936
articulación, gregoriano, accentus ... 554
articulación, gregoriano, circulus ... 554
articulación, gregoriano, episem finis ... 554
articulación, gregoriano, episem initium ... 554
articulación, gregoriano, ictus ... 554
articulación, gregoriano, semicirculus ... 554
articulación, marcato ... 156, 934
articulación, nota tapada ... 156
articulación, portato ... 156, 934
articulación, semiabierta ... 156, 935
articulación, staccatissimo ... 156, 934
articulación, staccato ... 156, 934
articulación, tapada ... 499, 935
articulación, tenuto ... 156, 934
articulación, valores predeterminados, modificación ... 158
articulación ... 156
articulaciones gregorianas ... 554
articulaciones ... 934
articulate.ly, script ... 666
\articulate ... 666
articulation, bachschleifer ... 156
articulation-event ... 271
artificiales, armónicos ... 431
\ascendens ... 556, 561
aspas, cabezas de nota ... 44
\assertBeamQuant ... 952
\assertBeamSlope ... 952
associatedVoice ... 351, 353, 386
\atLeft ... 157, 952
\atRight ... 157, 952
\auctum ... 556, 561
aug ... 511
\augmentum ... 562, 952
auto-first-page-number ... 683, 969
\auto-footnote ... 880
autobarrado, propiedades para indicaciones de compás ... 80
autoBeaming ... 104, 742
\autoBeamOff ... 101, 422
\autoBeamOn ... 101
\autoBreaksOff ... 694
\autoBreaksOn ... 694
\autoChange y el modo relativo ... 420
\autoChange ... 419, 952

\autoLineBreaksOff 694
\autoLineBreaksOn 694
automática, alteración 32
automática, barra de compás 130
automática, combinación de partes 235
automática, sustitución, de cadenas 649
automático, cambio de pentagrama 419
automático, diagrama de acordes 480
automático, diagrama de traste 480
\autoPageBreaksOff 696
\autoPageBreaksOn 696
ayuda, globos de 303

B

Bézier, curva de, puntos de control 808
'bachschleifer', articulación 156
bachschleifer 157
\bachschleifer 156
backslashed digit 880
\backslashed-digit 880
bajo acústico 937
bajo cifrado, alineación 535
bajo cifrado, líneas de extensión 532
bajo cifrado 528
bajo de los acordes 514
bajo, clave 20, 928
bajo 937
balance estéreo, en MIDI 665
balance, en MIDI 665
Balloon_engraver 303
\balloonGrobText 303, 952
\balloonLengthOff 303
\balloonLengthOn 303
\balloonText 303, 952
banjo, afinación del 488
banjo, tablatura de 433, 487
banjo-c-tuning 488
banjo-double-c-tuning 488
banjo-double-d-tuning 488
banjo-modal-tuning 488
banjo-open-d-tuning 488
banjo-open-dm-tuning 488
banjo-open-g-tuning 488
bar line, in markup 856
\bar-line 856
\bar 120, 129, 952
barítono varC, clave 928
barítono varF, clave 928
barítono, clave 20, 928
BarNumber 133
\barNumberCheck 141, 952
barNumberVisibility 133
barré, línea 484
barra corta, en canto gregoriano 554
barra cruzada 125
barra de compás, automática 130
barra de compás, cesura, en himnos 411
barra de compás, cesura 131
barra de compás, corta 121
barra de compás, de raya 121
barra de compás, doble 131, 147
barra de compás, en canto gregoriano 554
barra de compás, en canto kievano 564
barra de compás, en himnos 411
barra de compás, final 131, 147
barra de compás, frase 131
barra de compás, música sin metro 91
barra de compás, para repetición al comienzo de la pieza 192
barra de compás, predeterminada 131
barra de compás, punteada 131
barra de compás, repetición subyacente 131
barra de compás, repetición 131
barra de compás, sección 131, 147
barra de compás, secundaria 131
barra de compás, segno 122, 131, 201
barra de compás, sencilla 131
barra de compás 131
barra de frase, en canto gregoriano 554
barra de raya, en canto gregoriano 554
barra de sección, en canto gregoriano 554
barra final, en canto gregoriano 554
barra horizontal 447
barra inclinada, repetición 211
barra, cadencias 91
barra, cruzada 119
barra, de compás 120
barra, de pentagrama cruzado 417
barra, de trémolo 214
barra, doblada, cambiar 103
barra, doble 120
barra, en tablatura 447
barra, estricta 110
barra, final 120
barra, finales con varias voces 112
barra, finales en la partitura 112
barra, grupo especial, cortar línea 61
barra, música sin metro 91
barra, manual, abreviatura de dirección 114
barra, manual, notas de adorno 114
barra, manual 101, 114
barra, \partCombine con \autoBeamOff 103
barra, progresiva 118
barra, reglas personalizadas 101
barra, salto de línea 102
barra, secundaria, orientación de 110
barra, subdivisión 109
barra, y letra 104
barra, y melismas 102
barradas, cabezas de nota 52
barrado automático, propiedades para las indicaciones de compás 81
barrado, propiedades predefinidas para el compás . 80
barrado, propiedades predeterminadas de la indicación de compás 81
barras de compás manuales 120
barras, en música polimétrica 94
Bartók, pizzicato 432
bartype 129
base-shortest-duration 718
\bassFigureExtendersOff 531
\bassFigureExtendersOn 531
\bassFigureStaffAlignmentDown 534
\bassFigureStaffAlignmentNeutral 534
\bassFigureStaffAlignmentUp 534
basso continuo 528
batería 490, 492
\beam 845

\beamExceptions ... 104, 952
beatBase ... 104, 109
beatStructure ... 104, 109
becuadro, adicional, evitar ... 9
becuadro, evitar en armadura ... 26
becuadro ... 8
bemol, doble ... 8
bemol ... 8
\bendAfter ... 178, 952
\bendHold ... 443, 952
\bendStartLevel ... 443, 952
binding-offset ... 681, 969
bisbiglando ... 427
bison, programa ... 948
bitmap, fuentes ... 342
\blackTriangleMarkup ... 519
blank-after-score-page-penalty ... 683, 969
blank-last-page-penalty ... 683, 969
blank-page-penalty ... 683, 969
bloque, acorde ... 185
blot-diameter ... 969
BNF ... 948
\bold ... 327, 815
bongó abierto ... 937
bongó agudo ... 937
bongó grave ... 937
bongó tapado ... 937
bongó ... 937
\book, y \layout ... 593
\book, y variables ... 592
book-title-properties ... 970
book-title ... 970
\book ... 591, 594
\bookOutputName ... 952
\bookOutputSuffix ... 952
\bookpart, y \layout ... 593
\bookpart, y variables ... 592
bookpart-level-page-numbering ... 684, 970
\bookpart ... 592, 595, 696
bookTitleMarkup ... 606, 970
bottom-margin ... 676, 970
Bottom ... 746
bounding box, of glyph ... 890
\box ... 336, 815
\bp ... 784, 970
brace, in markup ... 883
\bracket ... 169, 336, 846
break-align, símbolo ... 723
break-align-orders ... 724
break-align-symbol ... 945
break-align-symbols ... 804
break-visibility ... 796
\break ... 692
breakable ... 102
\breakAlignInsert ... 953
breakbefore ... 604
\breakDynamicSpan ... 167
\breathe ... 177, 953
breve, nota, estilo alternativo ... 55
breve, silencio de ... 68
\breve ... 53, 68

C

cántico ... 403, 407
caída de tono (fall) ... 178
cabasa ... 937
cabezas de nota antiguas ... 547, 563
cadena, sustitución automática ... 649
cadencia, alteraciones ... 91
cadencia, barras de compás ... 91
cadencia, barras ... 91
cadencia, números de compás ... 91
cadencia ... 91
cadenza (cadencia), alinear a ... 153
cadenza (cadencia) ... 153
cadenza, saltos de línea ... 93
cadenza, saltos de página ... 93
\cadenzaOff ... 91
\cadenzaOn ... 91
caesura, barra de compás, en canto kievano ... 564
caesura ... 76
\caesura ... 76, 78, 131, 411, 554, 564, 953
caja acústica ... 937
caja china ... 937
caja circundante ... 785
caja electrónica ... 937
caja ... 937
calderón corto ... 156
calderón de Henze ... 156
calderón largo ... 156
calderón, corto ... 935
calderón, Henze ... 935
calderón, largo ... 935
calderón, sobre la línea divisoria ... 317
calderón, sobre un silencio multicompás ... 73
calderón ... 156, 935
callback ... 946
cambiar, nombre de instrumento ... 267
cambiar, propiedades ... 768, 774
cambio automático de pentagrama ... 419
cambio de dedo ... 284
cambio de la fuente tipográfica ... 327
cambio, de pentagrama, manual ... 417
cambio, de tempo, sin indicación metronómica ... 89
cambio, digitación ... 287
campana ... 937
canción, a varias voces ... 391
cantante, nombre del ... 383
capa ... 795
capo ... 464
\caps ... 816
carácter, especial, en modo de marcado ... 323
caracteres especiales ... 647
caracteres no ASCII ... 647
casilla, de repetición, con texto ... 208
casillas de primera y segunda vez ... 207
\cavum ... 556, 561
cejilla, indicación de ... 459
cejilla ... 464
cencerro ... 937
centímetros (cm) ... 784
\center-align ... 330, 828
\center-column ... 334, 828
centering column of text ... 828
centrado, matices, en música de piano ... 416
centrar texto en la página ... 334

cerradura ... 946
cesura, barra de compás ... 131
cesura ... 76
CFF2, fuente, formato ... 341
\change ... 417
changing direction of text column ... 829
\char ... 881
check mark ... 177
check-consistency ... 680, 970
ChoirStaff ... 743
'choral', estilo de alteraciones accidentales ... 36
'choral-cautionary', estilo de alteraciones ... 36
choral-cautionary ... 36
choral ... 36
chordChanges ... 477, 517
ChordGrid ... 524, 744
ChordGridScore ... 743
\chordmode ... 6, 17, 470, 587
chordNameExceptions ... 520, 521
chordNameLowercaseMinor ... 519
ChordNames, con líneas de compás ... 522
ChordNames ... 470, 745
chordNameSeparator ... 520, 523
chordNoteNamer ... 520
chordPrefixSpacer ... 521
\chordRepeats ... 438, 953
chordRootNamer ... 519
\chords ... 516, 587
\chordSlur ... 185
chorus, nivel, en MIDI ... 666
Christian Harmony, cabezas de nota con forma ... 48
cifrado americano ... 510
cifrado, bajo, alteración, posición ... 532
cifrado, bajo, distancia vertical de los elementos ... 534
cifrado, bajo, glifos de alteración ... 532
cifrado, bajo ... 528
\circle ... 336, 846
circling text ... 846
'circulus', articulación gregoriana ... 554
'circulus', articulación, gregoriana ... 936
\circulus ... 554, 936
citar otras voces ... 268, 272
clave de música antigua ... 545
clave de percusión ... 490
clave de Petrucci ... 545
clave de tablatura ... 457
clave kievana ... 545
clave mensural negra ... 545
clave mensural ... 545
clave moderntab ... 457
clave transpositora ... 21
clave, alto varC ... 928
clave, alto ... 928
clave, antigua, música ... 930
clave, antigua ... 20
clave, bajo ... 928
clave, barítono varC ... 928
clave, barítono varF ... 928
clave, canto kievano ... 930
clave, de alto ... 20
clave, de bajo ... 20
clave, de barítono ... 20
clave, de contrabajo ... 20
clave, de Do ... 20
clave, de Fa ... 20
clave, de guía ... 21
clave, de mezzosoprano ... 20
clave, de Sol ... 20
clave, de soprano ... 20
clave, de tenor ... 20
clave, de violín ... 20
clave, Do ... 928
clave, estilo ... 928
clave, estilos ... 545
clave, F ... 928
clave, francesa ... 20, 928
clave, G ... 928
clave, G2 ... 928
clave, GG ... 928
clave, mensural negra ... 930
clave, mensural ... 930
clave, mezzosoprano ... 928
clave, percusión ... 929
clave, Petrucci ... 930
clave, soprano ... 928
clave, sub-bajo ... 928
clave, tablatura ... 929
clave, tenor Sol ... 928
clave, tenor varC ... 928
clave, tenor ... 928
clave, treble ... 928
clave, trucaje de propiedades ... 23
clave, varbaritone ... 928
clave, varC ... 928
clave, violín ... 928
clave, visibilidad de la transposición ... 800
clave, visibilidad después de un cambio explícito ... 798
clave ... 20, 553, 563
claves ... 937
clef ... 8
\clef ... 20, 953
clip-regions ... 650, 970
cluster ... 223
\cm ... 784, 970
'coda de variación', signo ... 936
'coda de variante', signo ... 156
'coda', símbolo, variante ... 156
'coda', signo manual ... 209
'coda', signo, con repeticiones ... 197
'coda', signo ... 156, 936
'coda', variante del signo ... 203
coda ... 156, 198
\coda ... 156, 858, 936
\codaMark ... 131, 201, 209, 953
codificación, de texto ... 647
colisión, del número de compás ... 140
colisión, ignorar ... 223, 234
colisión, voces de pentagrama cruzado ... 418
colisión ... 229
colocación, de los silencios multicompás ... 74
color en acordes ... 294
color, adecuados para el dicromatismo ... 912
color, códigos de CSS ... 292
color, ceguera ... 293
color, de X11 ... 292
color, fuentes ... 341
color, RGB ... 293
color, voz ... 229
color ... 292
color ... 292

coloreada, nota de acorde ... 294
coloreada, nota ... 292
coloreado de un objeto ... 795
coloreado, objeto ... 292
colorear notas ... 292
colorear objetos ... 292
colores semitransparentes ... 292
colores, lista de ... 907
coloring text ... 889
\column-lines ... 891
\column ... 334, 829
columna, de papel ... 722
columna, texto en ... 322
columna ... 722
columnas, texto ... 334
coma, intervalos de ... 580
combinad silencios ... 235
combinar, partes, cambiar el texto ... 240
\combine ... 337, 829
comentario de bloque ... 591, 596
comentario de una línea ... 591, 596
comentario de varias líneas ... 591, 596
comentario ... 591, 596
comienzo de repetición ... 207
comienzo del sistema ... 246
comillas, en la letra ... 350, 357
common-shortest-duration ... 718
compás alterno estricto ... 131
compás de anacrusa ... 90
compás parcial ... 90
compás polimétrico ... 93
compás, agrupación ... 111
compás, barras, invisibles ... 120
compás, completo, silencios de ... 72
compás, comprobación de ... 141
compás, comprobación ... 141
compás, compuesto, indicación de ... 80
compás, contador ... 144
compás, definir línea de ... 125
compás, doble ... 93
compás, estilo de la indicación de ... 80
compás, estilo de ... 80
compás, indicación de, ajustes predeterminados ... 80
compás, indicación de, en medio de un compás ... 90
compás, indicación de, estilos de ... 546
compás, indicación de, imprimir solo el numerador ... 84
compás, indicación de, mensural ... 546
compás, indicación, y casillas de repetición de primera y segunda vez ... 85
compás, línea de, dentro de ChordNames ... 522
compás, línea, lista ... 939
compás, número de, formato ... 136
compás, número de ... 155
compás, número, alineación ... 138
compás, número, comprobación ... 141
compás, número, eliminar ... 139
compás, número, imprimir a intervalos regulares ... 134
compás, número, imprimir en el primer compás ... 133
compás, números de ... 132
compás, parcial, en himnos ... 412
compás, posición ... 91
compás, propiedades, restaurar los valores predeterminados ... 82
compás, repetición ... 211
compás, repeticiones y números de ... 206
compás, sub-agrupar ... 111
Completion_heads_engraver ... 97, 570
Completion_rest_engraver ... 97, 570
composer ... 601
\compound-meter ... 858
compresión de música ... 63
\compressEmptyMeasures ... 277
\compressMMRests ... 72, 74, 277, 953
comprobación de compás, con repeticiones ... 192
compuesto, indicación de compás ... 80
\concat ... 829
concatenating text ... 829
condensación, de silencios ... 76
conformar ligaduras ... 808
conga abierta ... 937
conga aguda ... 937
conga grave ... 937
conga tapada ... 937
conga ... 937
\consists ... 752, 758
constante, regulador de tipo ... 166
contador, repetición de porcentaje ... 212
contador, visibilidad de repetición de porcentaje ... 212
contemporáneo, glissando ... 180
contenido opcional ... 298
\context dentro de un bloque \layout ... 753
context-spec-music ... 233
\context ... 746, 753
contexto implícito ... 763
contexto, cambiar propiedades predeterminadas de ... 753
contexto, crear alias ... 758
contexto, definición de nuevos ... 758
contexto, definición en el MIDI ... 760
contexto, definiciones con MIDI ... 660
contexto, nivel Score ... 761
contexto, orden de disposición ... 762
contexto, propiedades de ... 768
contexto, superponer ... 247
contextos, creación y referenciación de ... 746
contextos, mantener vivos ... 749
contextos, tiempo de vida ... 749
\contextPropertyCheck ... 953
continua, ligadura ... 172
continuación, de notas ... 537
contrabajo, clave ... 20
control, alturas de ... 13
control, trucar un punto de ... 775
controlling general text alignment ... 831
controlpitch ... 13
copyright, símbolo de ... 648
copyright ... 601
coral, clave de tenor ... 21
corchete de fraseo con etiqueta ... 307
corchete de fraseo ... 306
corchete horizontal ... 306
corchete mensural ... 548
corchete vertical ... 246
corchete, al comienzo de un pentagrama único ... 248
corchete, antiguo recto ... 116
corchete, apilado ... 116
corchete, cuadrado, al inicio de un grupo ... 248
corchete, de pentagrama cruzado ... 423
corchete, moderno recto ... 116
corchete, non-arpeggiato ... 185

corchete, recto 115
corchete, vertical, señalar voces divididas 393
corchete 298, 306
corchetes, anidados 250
corchetes, tipo barra 116
\corcheteStyleDefault 116
\corcheteStyleStacked 116
coro, sistema de 246
corta, barra de compás 121
corto, calderón 156, 935
countPercentRepeats 212
\cr 160
creating a table 892
creating empty text object 884
creating horizontal space, in text 827, 834
creating text fraction 882
creating vertical space, in text 827, 842, 888
\cresc 162
crescendo-event 271
crescendo 160
crescendoSpanner 168
crescendoText 168
\crescHairpin 163
\crescTextCresc 163
cross 44
\crossStaff 422, 953
cruce de pentagramas 421
cruzada, barra 119
cruzado, acordes de pentagrama 422
cruzado, línea de pentagrama 421
cruzado, notas de pentagrama 422
cruzado, pentagrama, barras de 417
cruzado, pentagrama, colisión de voces 418
cruzado, pentagrama, corchete de acorde 423
cruzado, pentagrama, notas de 417
cruzado, pentagrama 421
cruzado, plicas de pentagrama 422
CSS, códigos de color 292
cuadrado, corchete, al inicio de un grupo 248
cuadrado, ligadura de neuma 556
cuartos de tono, alteraciones 10
cuartos de tono, tablatura 457
cuartos de tono 8
\cueClef 272, 953
\cueClefUnset 272, 954
\cueDuring, and tags 632
\cueDuring 273, 954
\cueDuringWithClef 273, 954
cuerda al aire, indicación de 430
cuerda, curvatura, en tablatura 443
cuerda, número de la 434
cuerda, número 430
cuerda, números de, orientación 285
cuerdas orquestales 429
cuerdas, escribir música para 429
CueVoice 272
CueVoice 745
cuica 937
currentBarNumber 132, 155
curvatura, cuerda, en tablatura 443
custodes (custodias) 543
\customTabClef 860
custos 543

D

dórico, modo 25
D.C., manual 210
D.C. 199
D.S., forzado al comienzo 201
D.S., manual 210
D.S. 199
da capo, manual 210
da capo 199
dal segno, forzado al comienzo 201
dal segno, manual 210
dal segno 199
\dashBang 158
\dashDash 158
\dashDot 158
\dashHat 158
\dashLarger 158
\dashPlus 158
\dashUnderscore 158
\deadNote 45, 954
\deadNotesOff 45
\deadNotesOn 45
debug-beam-scoring 685, 970
debug-slur-scoring 685, 971
debug-tie-scoring 685, 971
decorar texto 336
\decr 160
\decresc 162
decrescendo 160
decrescendoSpanner 168
decrescendoText 168
dedication 601
'default', estilo de alteraciones accidentales 34
default-staff-staff-spacing 701
default 34
\default 142, 613
\defaultchild 744, 763
\defaultTimeSignature 80
\defineBarLine 125, 954
definición, traste predefinido 475
definir línea de compás 125
\deminutum 556, 561
\denies 758, 760, 762
depuración, espaciado vertical 734
derecha, mano, digitación, posicionamiento 483
derecha, mano, digitaciones para instrumentos de traste 482
derecho, pedal 424
\descendens 556, 561
deslizamiento en la digitación 287
deslizamiento en notación de tablatura 451
desplazamiento en marcado, vertical 331
desplazamiento, de silencios, automático 229
desplazamiento, de voces 229
desplazamiento, silencio automático de 229
desplazamiento 70, 775
desplazar nota 229
desplegada, repetición 191
Devnull 744, 745
diagrama de acordes para instrumentos de trastes 459
diagrama de acordes 469
diagrama de posiciones, de mandolina 469
diagrama de posiciones, de ukelele 469
diagrama de posiciones 469

diagrama de traste, transporte de ... 471
diagrama de trastes personalizado ... 466
diagrama de trastes personalizados, añadir ... 472
diagrama de trastes ... 459
diagrama, de digitación ... 499
dibujar, objeto gráfico ... 336
dicromatismo, colores adecuados ... 912
digitación de pulgar ... 156, 285
digitación, añadir, a diagramas de trastes ... 481
digitación, de la mano derecha para instrumentos de trastes ... 482
digitación, dentro del pentagrama ... 286
digitación, deslizamiento ... 287
digitación, diagrama de ... 499
digitación, frente a números de cuerda ... 434
digitación, línea entre ... 287
digitación, orientación ... 285
digitación, posicionamiento, mano derecha ... 483
digitación, pulgar ... 935
digitación, símbolos para instrumentos de viento ... 499
digitación, y plicas ... 286
digitación, y silencios multicompás ... 76
digitación ... 284
\dim ... 162
dim ... 511
dimensión, undades ... 784
dimensión ... 785
dimension, of bounding box ... 890
dimension-variables ... 971
\dimHairpin ... 163
diminuendo ... 160
\dimTextDecr ... 163
\dimTextDecresc ... 163
\dimTextDim ... 163
dinámica del MIDI ... 657
dinámica personalizada en MIDI ... 657
dinámica, indicaciones centradas en música de tecla ... 416
dinámica, posición vertical ... 163
dinámica, varias indicaciones sobre una nota ... 161
dinámica ... 160
dinámicas, indicaciones editoriales ... 169
dinámicas, indicaciones entre paréntesis ... 169
dinámicas, indicaciones, nuevas ... 169
\dir-column ... 829
dirección automática de la plica sobre la línea media ... 300
dirección predeterminada de la plica sobre la línea media ... 300
dirección, hacia abajo (_) ... 783
dirección, hacia arriba (^) ... 783
dirección, predeterminada (-) ... 783
dirección, símbolos ... 111
\discant ... 874
discanto, signo de, del acordeón ... 426
discontinua, ligadura ... 65, 172
discontinuas, ligaduras de fraseo ... 175
discontinuidad, definir patrón de, ligadura de expresión ... 174
diseño «mensurstriche» ... 249
diseño de página ... 728
\displayLilyMusic ... 669, 954
\displayMusic ... 954
\displayScheme ... 954
disposición, acorde mayor 7 ... 522
disposición, archivo de ... 688
disposición, vertical, depuración ... 734
distancia, absoluta ... 784
distancia, entre pentagramas en música de piano ... 422
distancia, entre pentagramas ... 701
distancia, escalada ... 784
dividida, letra ... 379
dividida, voz ... 393
dividir en música etiquetada ... 637
divisi, pentagramas ... 262
división, de notas ... 97
división, de silencios ... 97
divisible, glissando ... 181
divisio ... 554
\divisioMaior ... 554
\divisioMaxima ... 554
\divisioMinima ... 554
divisiones ... 554
Do, clave ... 20, 928
doblada, barra, cambiar ... 103
doble barra de compás ... 131, 147
doble bemol ... 8
doble compás ... 93
doble ligadura de expresión para acordes legato ... 174
doble puntillo, nota con ... 54
doble sostenido ... 8
doble, barra de compás ... 131
doble, barra ... 120
'dodecaphonic', estilo de alteraciones ... 37
'dodecaphonic-first', estilo de alteración ... 38
dodecaphonic-first ... 38
'dodecaphonic-no-repeat', estilo de alteración ... 37
dodecaphonic-no-repeat ... 37
dodecaphonic ... 37
doit (elevación de tono) ... 178
\dorian ... 25
\dotsDown ... 54
\dotsNeutral ... 54
\dotsUp ... 54
\doubleflat ... 860
\doublesharp ... 860
doubleSlurs ... 174
\downbow ... 156, 430, 935
\downmordent ... 156, 934
\downprall ... 156, 934
\draw-circle ... 337, 846
\draw-dashed-line ... 847
\draw-dotted-line ... 847
\draw-hline ... 848
\draw-line ... 337, 848
\draw-squiggle-line ... 848
drawing beam, within text ... 845
drawing box, with rounded corners, around text ... 854
drawing box, with rounded corners ... 849
drawing circle, within text ... 846
drawing dashed line, within text ... 847
drawing dotted line, within text ... 847
drawing ellipse, around text ... 849
drawing line, across a page ... 848
drawing line, within text ... 848
drawing oval, around text ... 851
drawing path ... 852
drawing polygon ... 852
drawing solid box, within text ... 849
drawing squiggled line, within text ... 848

drawing triangle, within text 855
\dropNote 515, 954
\drummode 244, 490, 587
drumPitchNames 495
drumPitchTable 496
\drums 490, 587
DrumStaff 244, 744
drumStyleTable 495
DrumVoice 745
duración aislada 54
duración predeterminada 54
duración, de las notas 53
duración, escalado de 62
duración, línea 537
\dwn 575
dynamic-event 271
\dynamic 169, 816
\dynamicDown 163
DynamicLineSpanner 163, 167
\dynamicNeutral 163
Dynamics 745
\dynamicUp 163

E

\easyHeadsOff 46
\easyHeadsOn 46
ecualización predeterminada, instrumento MIDI, sustitución de la 659
editoriales, indicaciones dinámicas 169
efectos en MIDI 665
elemento, musical 722
elemento, no musical 722
elevación de tono (doit) 178
eliminación de acordes repetidos 517
eliminar, número de compás 139
eliminar, notas guía 276
\ellipse 849
Emmentaler, tipografía 912
Emoji, fuentes 341
empotrar archivos, en PDF 652
\EnableGregorianDivisiones 968
\enablePerStaffTiming 954
encabezamiento, página 605
encabezamiento 598
enclosing text, in box with rounded corners 854
enclosing text, in box 815
encuadernación 680
\endcr 160
\enddecr 160
\endSpanners 794, 954
\englishChords 518
Engraver_group 759
enmarcar texto 336
ensayo, letra de, estilo 142
ensayo, letra de, formato 142
ensayo, letra de, manual 142
ensayo, letras de 142
ensayo, marca de, debajo del pentagrama 89
ensemble, música de 395
eolio, modo 25
eonio, modo 25
'episem finis', articulación gregoriana 554
'episem initium', articulación gregoriana 554
\episemFinis 554
\episemInitium 554
EPS image 850
EPS, salida 651
\epsfile 338, 849
escala, duración 62
escalado, del pentagrama 256
escribir música en paralelo 241
espaciado horizontal 718
espaciado vertical 701
espaciado, de la letra 368
espaciado, horizontal, entre columnas 725
espaciado, horizontal, entre elementos no musicales 723
espaciado, mostrar la disposición 734
espaciado, nueva seccción de 719
espaciado 718
espacio, dentro de los sistemas 701
espacio, en la letra 350, 357
espacio, entre pentagramas 701
espacios en blanco 596
especial, símbolos de arpegio 184
especiales, cabezas de figura 44
'espressivo', articulación 156, 162, 934
espressivo 156
\espressivo 156, 162, 934
estándar, tamaño de la tipografía (de notación) .. 284
estéreo 665
estilo de la ligadura 172
estilo, de cabeza de nota 44
estilo, de silencio 69
estilo, letra de ensayo 142
estilo, matiz de, de texto 168
estilo, número de compás 136
estilo, silencios multicompás 279
estilo, voz 229
estricta, barra 110
estricto, espaciado, notas de adorno y 151
\etc 326, 643
evenFooterMarkup 971
evenHeaderMarkup 971
\eventChords 954
evento, contenedor 219
exacto, posicionamiento, de las pautas 708
excepción, nombres de acorde 521
\expandEmptyMeasures 277
expansión de música 63
explícita, repeticción 191
explícita, trinos de nota 188
explicitClefVisibility 798
explicitKeySignatureVisibility 798
expresión musical, compartir 644
expresión musical, copiar 644
expresión, de marcado 323
expresión, en MIDI 665
extensión, elemento de 949
extensión, matiz, ocultar línea de 168
extensión, objetos de, modificar 786
extensora, línea 361
extent, of actual inking 890
extent, of bounding box 890
extra-offset 701
extra-spacing-height 725
extra-spacing-width 725
extrema, altura 10

\eyeglasses 881
Ez_numbers_engraver 46

F

\f 160
fácil, notación, cabezas de nota de 46
fácil, notación, con números 46
F, clave 928
Fa, clave 20
fall (caída de tono) 178
falsa, percusión, nota 497
false (#f) 978
fantasma, nota 298
fantasma, percusión , nota 497
\featherDurations 118, 955
feature, OpenType font 821
fermata o calderón 156
fermata, sobre la línea divisoria 78
fermata 935
\fermata 78, 156, 860, 935
Ferneyhough, regulador de 166
Feta, tipografía 912
\ff 160
\fff 160
\ffff 160
\fffff 160
figura, cabeza de la 44
\figured-bass 816
FiguredBass 745
figuredBassAlterationDirection 532
figuredBassPlusDirection 532
figuredBassPlusStrokedAlist 532
\figuremode 529, 587
\figures 529, 587
\fill-line 334, 830
\fill-with-pattern 626, 831
\filled-box 337, 849
final de repetición 207
final, barra de compás, en canto kievano 564
final, barra de compás 131, 147
final, barra 120
finales alternativos y letra 374
finalis 554
\finalis 554
fine 198
\fine 131, 147, 196, 198, 564
\finger 284, 285, 817, 955
fingeringOrientations 285
first-page-number 683, 972
\first-visible 881
\fixed 4, 955
'flageolet', armónicos de 935
'flageolet', armónicos 156
flageolet 156
\flageolet 156, 935
\flat 860
flex, programa 947
flexa, en ligaduras mensurales 550
\flexa 562
followVoice 421
font feature, OpenType 821
font-interface 282
font-size 282, 284
\fontCaps 817
FontConfig 341
fontSize 282
\fontsize 327, 817
footnote-footer-padding 972
footnote-number-raise 972
footnote-numbering-function 972
footnote-padding 972
footnote-separator-markup 685, 972
\footnote 612, 881, 955
Forbid_line_break_engraver 61
'forget', estilo de alteraciones 38
forget 38
formas, notas con 48
formato, de la letra 350
formato, de los elementos de extensión de texto 315
formato, letra de ensayo 142
forte, pedal 424
forzada, dirección vertical, de grobs 783
four-string-banjo 488
\fp 160
\fraction 882
fragmentado, objeto de extensión, modificar 786
fragmento de música 650
fragmento, citar 268
fragmento 272
francesa, clave 20, 928
francesa, partituras a la 260
francesa, pentagrama a la 256, 260
frase, barra de compás, en canto kievano 564
frase, barra de compás 131
fraseo, barra de compás, en himnos 411
fraseo, corchetes de 306
fraseo, en la letra 358
fraseo, ligadura de, definir patrones de discontinuidad 176
fraseo, ligadura de, mitad continua y mitad discontinua 176
fraseo, ligadura de, simultáneas 175
fraseo, ligadura de, varias 175
fraseo, ligadura de 171, 175
fraseo, ligaduras de, discontinuas 175
fraseo, ligaduras de, punteadas 175
fraseo, marca de 175
\freeBass 874
\frenchChords 518
fret-diagram, marcado con 460
fret-diagram-interface 466
\fret-diagram-terse, marcado con 462
\fret-diagram-terse 462, 870
\fret-diagram-verbose, marcado con 464
\fret-diagram-verbose 464, 870
\fret-diagram 460, 869
FretBoards 469, 744
frigio, modo 25
\fromproperty 882
fuente tipográfica (de notación), tamaño estándar de 284
fuente tipográfica (de notación), tamaño 280
fuente tipográfica (marcado) 327
fuente tipográfica, cambiar para todo el documento 345
fuente tipográfica, cambiar 327
fuente tipográfica, familias de 329
fuente tipográfica 947

fuente, archivos externos ... 342
fuente, búsqueda ... 342
fuente, establecer la familia de ... 345
fuente, formato, bitmap ... 342
fuente, formato, CFF2 ... 341
fuente, formato, Font Variations (VF) ... 341
fuente, formato, OTC ... 341
fuente, funcionalidad, OpenType ... 344
fuentes, color ... 341
fuentes, for Emojis ... 341
funcionalidad, OpenType, fuente ... 344
fundamental de un acorde ... 511
Funk, cabezas de nota con forma ... 48
\funkHeads ... 48
\funkHeadsMinor ... 48

G

gaita escocesa de Highlands ... 501
gaita, ejemplo de ... 502
gaita ... 501
\general-align ... 332, 831
\germanChords ... 518
\glide ... 287
glifo, alteración, alternativos ... 40
glifo, conjunto, alteración ... 927
glifo, para alteraciones del bajo cifrado ... 532
glifo ... 947
glissando contemporáneo ... 180
glissando, a través de repetición ... 182
glissando, divisible ... 181
glissando, en acordes de tablatura ... 452
glissando, marca de tiempo ... 180
glissando, y repeticiones ... 206
glissando ... 178
\glissando ... 178
\glissandoMap ... 179
global, sustitución automática de cadenas ... 649
Global ... 742
globo de ayuda ... 303
globo ... 303
glyph, bounding box ... 890
golpeado, ritmo, mostrar ... 100
gráfica, notación ... 337
gráfico, dibujar objeto ... 336
gráfico, insertar objeto ... 336
gráfico, insertar ... 336, 338
grabador, incluir en contextos ... 758
\grace ... 148, 955
gramática, de LilyPond ... 948
grandes, puntos (bp) ... 784
GrandStaff ... 743
gregoriano, alteración ... 553
gregoriano, armadura ... 553
gregoriano, articulación, accentus ... 554, 936
gregoriano, articulación, circulus ... 554, 936
gregoriano, articulación, episem finis ... 554
gregoriano, articulación, episem initium ... 554
gregoriano, articulación, ictus ... 554, 936
gregoriano, articulación, semicirculus ... 554, 936
gregoriano, articulación ... 554
gregoriano, canto, pauta de transcripción de ... 245
gregoriano, ligadura de neumas ... 556
gregoriano, transcripción moderna ... 408
GregorianTranscriptionLyrics ... 745
GregorianTranscriptionStaff ... 245, 744
GregorianTranscriptionVoice ... 745
Grid_line_span_engraver ... 304
Grid_point_engraver ... 304
gridInterval ... 304
grob, posición vertical ... 783
grob, propiedad de ... 770
grob, sobreescritura de ... 795
grob, sobreescritura ... 66
grob, visibilidad ... 794
grob, y circundado en blanco (whiteout) ... 795
grob, y perfilado en bordes blancos ... 66
grob ... 765, 947
\grobdescriptions ... 955
grow-direction ... 118
grupeto circular barrado, ornamento ... 156, 934
grupeto circular de Haydn, ornamento ... 934
grupeto circular invertido, ornamento ... 156, 934
grupeto circular, ornamento ... 156, 934
grupeto de Haydn, ornamento ... 156
grupeto de retardación, ornamento ... 159
grupeto ... 156
grupo de pentagramas, corchete cuadrado al inicio ... 248
grupo de pentagramas ... 246
grupo de valoración especial, barra, cortar línea ... 61
grupo de valoración especial, colocación del corchete ... 56
grupo de valoración especial, corchete, visibilidad ... 59
grupo de valoración especial, modificación del número ... 58
grupo de valoración especial, número, no predeterminado ... 58
grupo especial, corchete, del lado de la cabeza ... 60
grupo especial, corchete, visibilidad ... 57
grupo especial, formato de ... 58
grupo especial, introducir muchos ... 58
grupo especial, ligadura ... 57
grupo especial, número, visibililidad ... 57
grupo especial ... 56
grupos de etiquetas ... 634
guía, en partitura vocal ... 399
guía, formateo de las notas ... 272
guía, musicales ... 397
guía, nota, eliminar ... 276
guía, nota, formato de ... 272
guía, nota ... 272
guía, notas, clave con ... 21
guía, voces ... 268
guión ... 156
guion ... 361
guiro ... 937
guitarra, cabezas de nota ... 44
guitarra, mostrar ritmos rasgueados ... 99
guitarra, ritmo golpeado, mostrar ... 100
guitarra, tablas de acordes ... 99
guitarra, tablatura de ... 433

H

hairpin, aligning ends to NoteColumn directions .. 165
Hal Leonard 46
half-open high hat 495
\halfopen 156, 495, 935
\halign 331, 832
hammer on (deslizamiento de entrada) 453
Harmonia Sacra, cabezas de nota con forma 48
\harmonic 45, 431, 441
\harmonicByFret 441, 955
\harmonicByRatio 441, 955
\harmonicNote 955
\harmonicsOff 430
\harmonicsOn 430, 955
\harp-pedal 428, 871
\haydnturn 156, 934
\hbracket 336, 850
\hcenter-in 833
\header 594
\heel 156, 426, 935
\heelcircle 156, 426, 935
Henze, calderón de 156
Henze, calderón 935
\henzelongfermata 156, 935
\henzeshortfermata 156, 935
\hide 795, 955
\hideKeySignature 501
\hideNotes 291
\hideSplitTiedTabNotes 440
\hideStaffSwitch 421
high hat, abierto 156, 937
high hat, de pedal 937
high hat, half open 495
high hat, pedal splash 495
high hat, pedal 156, 495
high hat, semi abierto 156
high hat, semiabierto 937
high hat 937
himno, y compás parcial 412
himno 403, 407
hoja guía de acordes (lead sheet) 517
horizontal, alineación de texto 330
horizontal, alineación, letra 369
horizontal, alineación, melisma 369
horizontal, barra 447
horizontal, corchete 306
horizontal, desplazamiento de notas 234
horizontal, espaciado 718
horizontal, ornamento 157
horizontal, relleno en marcado 331
horizontal-shift 681, 972
Horizontal_bracket_engraver 306
HorizontalBracketText 307
horizontally centering text 828
\hspace 331, 834
Hufnagel 540, 541
\huge 283, 330, 817
hyperlink, as QR code 886

I

'ictus', articulación gregoriana 554
'ictus', articulación, gregoriana 936
\ictus 554, 936
idioma, alturas en otros 10
idioma, nombres de nota en otros 10
\if 610, 868
iglesia, silencios de 279
\iij 557
\IIJ 557
\ij 557
\IJ 557
image 850
\image 850
imagen, insertar 338
implícito, contexto 763
importing stencil, into text 888
impresión de los nombres de acorde 515
impresión, de caracteres especiales 323
impresión, orden de 795
improvisación, árabe 578
improvisación 52
\improvisationOff 52, 99
\improvisationOn 52, 99
in-note-direction 621
in-note-padding 621, 972
in-note-system-padding 621, 972
\in 784, 972
incipit, añadir 566
incipit-width 972
\incipit 566, 956
inclinación, modificación, extensor de octava 30
\inclinatum 556, 561
\include 596, 627
inclusión de ajustes 641
inclusión de archivos 627
indent 266, 681, 727, 972
independiente, texto 321
indicación de compás aditiva 80
indicación de compás alterna 80
indicación de compás de amalgama 80
indicación de compás, ajustes predeterminados 81
indicación de compás, estilo de la 80
indicación de compás, visibilidad 79
Indicación de compás 79
indicación de pedal 156
indicación de pedalero de órgano 156
indicación de pulgar 156
indicaciones dinámicas nuevas 169
indicar 'No Chord' (sin acorde) en ChordNames ... 516
índice general, funciones personalizadas 625
índice general 623
\inherit-acceptability 759, 956
\initialContextFrom 956
inner-margin 680, 972
input-language 948, 957
inscripción, ordenación vertical 158
inscripción, sobre un silencio multicompás 73
inscripciones textuales 934
insertar, gráfico 338
insertar, objeto gráfico 336
inserting music, into text 865
inserting PostScript directly, into text 853
inserting URL link, into text 856

\inStaffSegno 131, 209, 956
instrument 601
instrumento MIDI, sustituir la ecualización predeterminada 659
instrumento, nombre abreviado 265
instrumento, nombre del 265
instrumento, nombre, centrar 266
instrumento, nombre, complejo 265
instrumento, nombre 661
instrumento, transpositor 14
instrumentos transpositores 15
\instrumentSwitch 956
interfaz de la presentación 765
interfaz, de grob 947
interfaz, de objeto gráfico 947
interfaz 947
introducir, la letra 350
introductorio, material 722, 723
inversión modal 19
inversión 17
\inversion 17, 956
\invertChords 515, 956
invisible, nota 291
invisible, plica 300
invisible, silencio 70
\ionian 25
irregular measure 90
is-last-bookpart 972
\italianChords 518
\italic 327, 818

J

jazz, acorde de 518
jazz, acordes de 510
\jump 210, 956
justificar, texto 335
\justified-lines 326, 891
\justify-field 835
\justify-line 835
\justify-string 836
\justify 335, 834
justifying lines of text 891
justifying text 834

K

\keep-with-tag 638, 883
keepAliveInterfaces 261
\keepWithTag 630, 956
key signature 8
\key 25, 48, 957
Kievan_ligature_engraver 564
kievana, clave 545
kievano, alteraciones de 564
kievano, canto, clave 930
kievano, ligaduras de 564
\kievanOff 563
\kievanOn 563
KievanStaff 562, 744
KievanVoice 562, 745
\killCues 276, 957
Kirchenpause 279
koron 582

L

línea de cambio de pentagrama 421
línea de extensión 361
línea de rejilla 304
línea de seguimiento de pentagrama 421
línea divisoria entre pentagramas 125
línea divisoria predeterminada, cambiar 129
línea divisoria, detener regulador 164
línea divisoria 131
línea vertical entre pentagramas 304
línea, adicional 252
línea, barré 484
línea, comentario de 591
línea, de compás, entre pautas 249
línea, del pentagrama 252
línea, divisoria, evitada por la letra lyrics 370
línea, divisoria, invisible 120
línea, divisoria, símbolos sobre las 317
línea, divisoria, texto sobre las 317
línea, divisorias 120
línea, entre digitaciones 287
línea, indicando duración 537
línea, salto, barra 102
línea, salto, en cadencias 93
línea, salto, en música sin medida 93
línea, salto, gestión con una voz adicional 695
línea, salto 120
líneas divisorias manuales 120
laúd, afinación del 489
laúd, tablaturas de 488
label-alist-table 973
label-page-table 973
\label 623, 957
laissez vibrer 65
\laissezVibrer 65
landscape 973
\language 957
\languageRestore 957
\languageSaveAndChange 957
\large 283, 330, 818
\larger 327, 330, 818
largo, calderón 156, 935
last-bottom-spacing 678, 713, 973
\layout, and paper variables 686
\layout, y \book 593
\layout, y \bookpart 593
layout-set-staff-size 688
\layout 594, 686, 742, 753
lead sheet (hoja guía de acordes) 517
\left-align 330, 836
left-aligning text 836
\left-brace 883
\left-column 837
left-margin 679, 973
letra de ensayo, estilo 142
letra de ensayo, formato 142
letra dividida 379
letra, alineación a una melodía 351
letra, alineación con melodía esporádica 750
letra, alineación horizontal 369
letra, asignada a una voz 224
letra, aumentar la separación 368
letra, compartida entre voces 380
letra, desplazamiento de 70

letra, evitar líneas divisorias ... 370
letra, formateado ... 350
letra, introducir ... 350
letra, mantener dentro del margen ... 314, 370
letra, posicionamiento ... 364
letra, puntuación en la ... 350
letra, repetición con finales alternativos ... 374
letra, repetición con primera y segunda vez ... 374
letra, repetición ... 370
letra, saltar notas ... 375
letra, separada, en una línea única ... 366
letra, sobre notas de adorno ... 385
letra, usar variables ... 362
letra, y barras ... 104
letra, y marcado ... 351
letra, y melodías ... 353
letra, y notas ligadas ... 376
letra ... 350
letras de ensayo ... 142
\lheel ... 425
\lheeltoe ... 425
lidio, modo ... 25
ligadura continua ... 172
ligadura de expresión y repeticiones ... 206
ligadura de expresión, ajsutar comienzo y final ... 173
ligadura de expresión, debajo de las notas ... 171
ligadura de expresión, definición del patrón de discontinuidad ... 174
ligadura de expresión, definir patrones de discontinuidad ... 172
ligadura de expresión, doble, para acordes legato ... 174
ligadura de expresión, encima de las notas ... 171
ligadura de expresión, estilo ... 172
ligadura de expresión, marcado de texto por dentro ... 174
ligadura de expresión, mitad continua y mitad discontinua ... 172
ligadura de expresión, modificar ... 808
ligadura de expresión, posicionamiento manual ... 171
ligadura de expresión, simultáneas ... 171
ligadura de expresión, varias ... 171
ligadura de expresión ... 171
ligadura de fraseo discontinua ... 175
ligadura de fraseo mitad continua y mitad discontinua ... 176
ligadura de fraseo punteada ... 175
ligadura de fraseo, definir patrones de discontinuidad ... 176
ligadura de fraseo, simultáneas ... 175
ligadura de fraseo ... 171, 175
ligadura de grupo especial ... 57
ligadura de puntos ... 172
ligadura de unión, a nada ... 65
ligadura de unión, corchetes de primera y segunda vez ... 64
ligadura de unión, desde nada ... 64
ligadura de unión, en repeticiones ... 64
ligadura de unión, finales alternativos ... 64
ligadura de unión, modificar ... 808
ligadura de unión, repetición y ... 64
ligadura de unión ... 63
ligadura discontinua ... 172
ligadura laissez vibrer ... 65
ligadura mensural blanca ... 550
ligadura mensural ... 550
ligadura vertical ... 185
ligadura, alteraciones y ... 9
ligadura, apariencia ... 65
ligadura, colocación ... 65
ligadura, con arpegio ... 66
ligadura, de puntos ... 65
ligadura, discontinua ... 65
ligadura, edición manual ... 67
ligadura, en la letra ... 357
ligadura, neumas cuadrados ... 556
ligadura, y acordes ... 64
ligadura ... 542
ligaduras de fraseo, varias ... 175
ligaduras kievanas ... 564
ligature, in text ... 829
\ligature ... 957
Ligature_bracket_engraver ... 550, 564
LilyPond, analizador léxico ... 947
line-break-system-details ... 708
line-thickness ... 973
line-width ... 335, 679, 727, 973
\line ... 325, 326, 837
\linea ... 556, 561
\lineprall ... 156, 934
lista de asociación ... 946
lista de colores ... 907
lista de diagramas de viento ... 506
lista de llaves en diagramas de viento ... 508
lista de marcados ... 324
lista-A ... 946
llave vertical ... 246
llave, al comienzo de un pentagrama único ... 248
llaves, anidado ... 250
\locrian ... 25
locrio, modo ... 25
longa, silencio de ... 68
\longa ... 53, 68
\longfermata ... 156, 935
longitud mínima, regulador ... 164
longitud, de las notas ... 53
longitud, de un silencio multicompás ... 74
\lookup ... 883
\lower ... 331, 837
lowering text ... 837
\ltoe ... 425
\ltoeheel ... 425
ly:minimal-breaking ... 697
ly:one-line-auto-height-breaking ... 698
ly:one-line-breaking ... 698
ly:one-page-breaking ... 698
ly:optimal-breaking ... 697
ly:page-turn-breaking ... 698
\lydian ... 25
\lyricmode ... 350, 351, 588
lyrics, posicionamiento ... 259
\lyrics ... 588
Lyrics ... 745
\lyricsto ... 351, 353, 354

M

m (en un acorde) ... 511
máxima altura ... 10
mínima altura ... 10
música antigua, claves ... 545
música de cuatro compases por línea ... 694
música intercalada ... 241
música para principiantes ... 46
música religiosa ... 403
música sin medida, saltos de línea ... 93
música sin medida, saltos de página ... 93
música sin medida ... 155
música sin metro, alteraciones ... 91
música sin metro, barras de compás ... 91
música sin metro, barras ... 91
música sin metro, números de compás ... 91
música sin metro ... 91
música, dentro de marcado ... 338
música, fragmento ... 650
magnification->font-size ... 282, 688
\magnify ... 327, 818
magnifying text ... 818
\magnifyMusic ... 280, 957
\magnifyStaff ... 688, 957
magstep ... 282, 688, 785
maj ... 511
\major ... 25
majorSevenSymbol ... 519, 522
makam, ejemplo ... 581
makam, turco, ejemplo ... 581
makam ... 573, 580
makamlar ... 573, 580
make-dynamic-script ... 169
make-footer ... 973
make-header ... 973
make-pango-font-tree ... 345
make-relative ... 645
\makeClusters ... 223, 957
\makeDefaultStringTuning ... 957
mantener música etiquetada ... 630
mantenimiento, pedal, estilo de ... 424
manual, barra ... 101, 114
manual, cambio de pentagrama ... 417
manual, edición de ligadura ... 67
manual, marca de ensayo ... 142
manual, marca de repetición ... 207
manuales, barras de compás ... 120
manuales, líneas divisorias ... 120
maqam ... 573, 580
maracas ... 937
marca de ensayo, estilo ... 142
marca de ensayo, formato ... 142
marca de ensayo, manual ... 142
marca de fraseo ... 175
marca de pausa ... 76
marca de pulso ... 177
marca, de ensayo, debajo del pentagrama ... 89
marca, en todos los pentagramas ... 143
marcado, caracteres especiales en modo de ... 323
marcado, centrar en la página ... 334
marcado, condicional ... 610
marcado, de texto ... 323
marcado, de varias líneas ... 334
marcado, de varias páginas ... 326
marcado, desplazamiento horizontal ... 331
marcado, desplazamiento vertical ... 332
marcado, dos columnas ... 322
marcado, elementos, alinear ... 330
marcado, en letra ... 351
marcado, expresión de ... 323
marcado, lista ... 324
marcado, notación musical dentro de ... 338
marcado, objeto ... 311
marcado, partitura dentro de ... 340
marcado, relleno de textos de ... 337
marcado, relleno horizontal ... 331
marcado, silencio multicompás ... 75
marcado, sintaxis ... 323
marcado, sobre silencios multicompás ... 73
marcado, tamaño del texto ... 327
marcado, texto, ajustado a palabras ... 335
marcado, texto, alineación ... 336
marcado, texto, alinear ... 330
marcado, texto, ancho de línea ... 335
marcado, texto, de varias páginas ... 326
marcado, texto, decorar ... 336
marcado, texto, dentro de las ligaduras ... 174
marcado, texto, enmarcar ... 336
marcado, vertical, desplazamiento ... 331
marcado ... 323
marcadores ... 622, 623
'marcato', articulación ... 156, 934
\marcato ... 156, 934
margen de encuadernación ... 680
margen, texto fuera del ... 314
\mark ... 142, 317, 957
Mark_engraver ... 143
\markalphabet ... 883
\markLengthOff ... 88, 319
\markLengthOn ... 88, 319
\markletter ... 884
markup, image ... 850
markup, rhythm ... 864
markup-markup-spacing ... 678, 973
markup-system-spacing ... 678, 974
\markup ... 311, 317, 321, 322, 323, 588
\markuplist ... 322, 326
\markupMap ... 957
material, introductorio ... 722, 723
matices absolutos ... 160
matices, posición vertical ... 163
matices, varios sobre una nota ... 161
matices ... 160
matiz, alineación vertical, romper ... 167
matiz, alineación vertical ... 167
matiz, estilo de texto ... 168
matiz, extensión de texto, personalizar ... 316
matiz, línea de extensión, ocultar ... 168
max-systems-per-page ... 681, 974
maxima, silencio de ... 68
\maxima ... 53, 68
mayor 7, acorde, disposición ... 522
mayor, modo ... 25
mayor, séptima, símbolo de ... 521
measure position ... 155
measure, irregular ... 90
Measure_grouping_engraver ... 111
measureBarType ... 129
measureLength ... 104, 155

measurePosition ... 155
Medicaea, Editio ... 540, 541
medida, música sin ... 155
medida, unidades ... 784
medida ... 79
medio, intervalo ... 573
melisma, alineción horizontal ... 369
melisma, barras en ... 102
melisma ... 358, 361
\melisma ... 358
\melismaEnd ... 358
melismata ... 358
melodía alternativa, cambio a una ... 386
melodía, mostrar el ritmo ... 98
menor, modo ... 25
mensuración, símbolo de ... 546
mensural negra, clave ... 545, 930
mensural, clave ... 545, 930
mensural, corchete ... 548
mensural, ligadura, flexa ... 550
mensural, ligadura ... 550
mensural, música, transcripción de ... 249
mensural, notación, signum congruentiae ... 936
mensural ... 540, 541
Mensural_ligature_engraver ... 550
MensuralStaff ... 245, 544, 744
MensuralVoice ... 544, 745
mensurstriche, diseño ... 249
Mensurstriche ... 566
\mergeDifferentlyDottedOff ... 229
\mergeDifferentlyDottedOn ... 229
\mergeDifferentlyHeadedOff ... 229
\mergeDifferentlyHeadedOn ... 229
merging text ... 829, 838
meter ... 601
metrónomo, indicación de ... 86
metrónomo, indicación, debajo del pentagrama ... 89
metro polimétrico ... 93
metro, estilo de ... 80
metronómica, marca de, personalizada ... 89
mezclar notas ... 229
mezzosoprano, clave ... 20, 928
\mf ... 160
microtono ... 12
microtonos, tablatura ... 457
\midi ... 594, 742
MIDI con repeticiones ... 662
MIDI, balance estéreo ... 665
MIDI, balance ... 665
MIDI, bloque ... 656
MIDI, canales ... 663
MIDI, chorus, nivel ... 666
MIDI, definiciones de contexto ... 660
MIDI, dinámicas personalizadas ... 657
MIDI, ecualización ... 658
MIDI, efectos ... 665
MIDI, expresión ... 665
MIDI, instrumento ... 661
MIDI, matices dinámicos ... 657
MIDI, metadatos de ... 611
MIDI, notación contemplada ... 655
MIDI, notación no contemplada ... 656
MIDI, pistas ... 663
MIDI, posición de panorama ... 665
MIDI, reverberación ... 666
MIDI, sustituir la ecualización de instrumento predeterminada ... 659
MIDI, transposición ... 30
MIDI, un canal por cada voz ... 664
MIDI, volumen ... 658
MIDI ... 30, 655
midiBalance ... 665
midiChannelMapping ... 663
midiChorusLevel ... 666
midiDrumPitches ... 496
midiExpression ... 665
midiInstrument ... 661
midiPanPosition ... 665
midiReverbLevel ... 666
milímetros (mm) ... 784
min-systems-per-page ... 681, 974
minimum-Y-extent ... 701
minimumFret ... 438, 481
\minor ... 25
minorChordModifier ... 520
mirroring markup ... 855
mixolidio, modo ... 25
\mixolydian ... 25
\mm ... 784, 974
modal, inversion ... 19
modal, transformación ... 18
modal, transposición ... 18
\modalInversion ... 19, 958
\modalTranspose ... 18, 958
'modern', estilo de alteraciones ... 34, 35
'modern-cautionary', estilo de alteraciones ... 34, 35
modern-cautionary ... 35
'modern-voice-cautionary', estilo de alteración ... 35
modern-voice-cautionary ... 35
modern-voice ... 35
modern ... 34
moderna, transcripción del canto gregoriano ... 408
moderntab, clave ... 457
modificación del separador de acordes ... 523
modificador de acorde ... 511
modificar contextos ... 768
modificar propiedades ... 768
modificar símbolo de marca de respiración ... 177
modificar, número de puntillos ... 55
modo eclesiástico ... 25
modo ... 25
\mordent ... 156, 934
mordente ascendente, ornamento ... 156, 934
mordente descendente, ornamento ... 156, 934
mordente, ascendente ... 156
mordente, circular, cruzado ... 156
mordente, circular, de Haydn ... 156
mordente, circular, inverso ... 156
mordente, circular ... 156
mordente, descendente ... 156
mordente, ornamento ... 156, 934
mordente ... 148, 156
mordentes ... 501
mostrar, disposición vertical ... 734
movimiento, varios ... 591
\mp ... 160
muerta, percusión, nota ... 497
multi-measure rest, within text, by duration ... 863
multi-measure rest, within text, by number of measures ... 861

\multi-measure-rest-by-number 861
multicompás, posicionamiento de los silencios 74
multicompás, silencio, adjuntar texto 73
multicompás, silencio, aplicar un calderón a 73
multicompás, silencio, comtraer varios 277
multicompás, silencio, con marcados 73
multicompás, silencio, contraer varios 277
multicompás, silencio, estilo 279
multicompás, silencio, expandir varios 277
multicompás, silencio, longitud 74
multicompás, silencio, numerar 279
multicompás, silencios, inscripción en 73
multicompás, silencios, y digitación 76
multicompás, silencios 72
MultiMeasureRestScript 73
MultiMeasureRestText 73
musica ficta 549
musical, elemento 722
musical, guía 397
musical, teatro 394
\musicglyph 861
\musicLength 958
\musicMap 958
musicológico, análisis 306
musicQuotes 948

N

\n 160
número de compás y repeticiones 206
número de compás, alternativo, en repetición 204
número de compás, cadencias 91
número de compás, centrado 139
número de compás, colisión 140
número de compás, comprobación de 141
número de compás, dividido 135
número de compás, formato de 136
número de compás, módulo 135
número de compás, música sin metro 91
número de compás 155
número de cuerda, frente a digitaciones 434
número, de compás, espaciado regular 133
número, de compás 132
número, de página, automático 683
número, de página, especificar el primero 683
número, de página, suprimir 683
número, grupo de valoración especial, modificación 58
números de página, romanos 683
números, en notación simplificada 46
N.C. symbol, customizing 517
N.C., símbolo 516
\name 758
natural, armónico 430
natural, nota 8
\natural 861
'neo-modern', estilo de alteraciones 36
'neo-modern-cautionary', estilo de alteraciones 37
neo-modern-cautionary 37
'neo-modern-voice', estilo de alteración 37
'neo-modern-voice-cautionary', estilo de alteraciones 37
neo-modern-voice-cautionary 37
neo-modern-voice 37
neo-modern 36
neomensural 541
\new 746
\newSpacingSection 719
niente, al, regulador 166
nivel superior, texto del 321
'No Chord' (sin acorde), símbolo 516
'No Chord' symbol, customizing 517
no especificada, altura 10
no musical, elemento, espaciado horizontal 723
no musical, elemento 722
no musicales, símbolos 337
no vacíos, textos 314
'no-reset', estilo de alteraciones 38
no-reset 38
\noBeam 114
\noBreak 694
noChordSymbol 517
nombre de nota 3
nombre del cantante 383
nombre, de instrumento, añadir a otros contextos 267
nombre, de instrumento, cambiar 267
nombres, de personaje 395
non-arpeggiato 185
\nonArpeggiato 185, 393, 423
NonMusicalPaperColumn 708, 722
nonstaff-nonstaff-spacing 701
nonstaff-relatedstaff-spacing 701
nonstaff-unrelatedstaff-spacing 701
\noPageBreak 696, 958
\noPageTurn 699, 958
\normal-size-sub 819
\normal-size-super 329, 819
\normal-text 819
\normal-weight 819
\normalsize 281, 283, 330, 820
nota al pie dentro de expresiones musicales 612
nota al pie, basada en eventos 613
nota al pie, basada en tiempo 615
nota al pie, en texto independiente 618
nota al pie 612
nota coloreada en acordes 294
nota coloreada 292
nota de adorno, cambiar los ajustes de disposición 149
nota de adorno, trucar 149
nota de adorno 501
nota de espaciado 70
nota entre paréntesis 298
nota fantasma 298
nota invisible 291
nota oculta 291
nota tapada, articulación 156
nota transparente 291
nota, abierta 156
nota, altura predeterminada 54
nota, cabeza de, Aiken 48
nota, cabeza de, armónico 44
nota, cabeza de, arpa sacra 48
nota, cabeza de, barradas 52
nota, cabeza de, Christian Harmony 48
nota, cabeza de, en aspas 44
nota, cabeza de, en parlato 44
nota, cabeza de, en rombo 44
nota, cabeza de, estilos 44
nota, cabeza de, estudio 46
nota, cabeza de, forma de rombo 430

nota, cabeza de, Funk 48
nota, cabeza de, guitarra 44
nota, cabeza de, Harmonia Sacra 48
nota, cabeza de, improvisación 52
nota, cabeza de, notación simplificada 46
nota, cabeza de, para practicar 46
nota, cabeza de, Southern Harmony 48
nota, cabeza de, Walker 48
nota, cabeza especial de 44
nota, cabeza, Aiken, variante fina 50
nota, colisiones de 229
nota, con doble puntillo 54
nota, con puntillo, cambiar número 55
nota, con puntillo, mover horizontalmente 233
nota, con puntillo 54
nota, continuación 537
nota, corchetes de agrupación de 306
nota, de adorno, cambiar ajustes de presentación 150
nota, de pentagrama cruzado 417, 422
nota, desplazamiento horizontal 234
nota, división de 97
nota, duración de 53
nota, duración predeterminada 54
nota, espaciar horizontalmente 719
nota, estilo de cabeza 927
nota, guía 268, 272
nota, interior 621
nota, longitud de 53
nota, nombres de, en otros idiomas 10
nota, nombres neerlandeses de 8
nota, nombres predeterminados 8
nota, nombres, árabes 574
nota, nombres, Hel-arabic 574
nota, pequeña 272
nota, tapada 156
nota, transporte 14
notación gráfica 337
notación, dentro de marcado 338
notación, explicación de 303
notación, tamaño de la fuente 280
notas, cabeza de 280
notas, cabezas de, con formas 48
notas, nombre, imprimir 301
note head, shape, merging 50
note, within text, by duration 861
note, within text, by `log` and `dot-count` 862
`\note-by-number` 862
`note-event` 271
`\note` 861
`Note_heads_engraver` 97
`Note_name_engraver` 301
`NoteColumn` 722
`\notemode` 588
`noteNameFunction` 301
`NoteNames` 301, 745
`noteNameSeparator` 301
nuevo pentagrama 244
nuevos contextos 746
`\null` 884
`NullVoice` 380, 745
`number-footnote-table` 974
`\number` 820
`\numericTimeSignature` 80

O

objecto, marcado 311
objeto de presentación 947
objeto gráfico 947
objeto inmutable 947
objeto, coloreado de 795
objeto, colorear 292
objeto, de extensión (spanner) 786
objeto, de propiedades 948
objeto, de Scheme 949
objeto, rotación 801
objeto, sobreescritura de 795
objeto, visibilidad 794
objetos mutables 947
octava, comprobación de 13
octava, corrección de 13
octava, elemento de extensión, modificar inclinación 30
octava, escritura relativa 4
octava, especificación absoluta 3
octava, especificación relativa 4
octava, introducción absoluta 3
octava, marca de cambio de 3
octava, para una voz 29
octava, transposición, opcional 21
octava, transposición 21
octavación, texto 29
octavación 27
`octavación` 29
`\octaveCheck` 13, 958
oculta, nota 291
ocultar, alteración, en notas ligadas al principio de un sistema 9
ocultar, línea de extensión de matiz 168
ocultar, pauta de Vaticana 260
ocultar, pentagrama de música antigua 260
ocultar, pentagrama rítmico 260
ocultar, pentagrama 260
`oddFooterMarkup` 974
`oddHeaderMarkup` 974
`\offset` 775, 958
`\omit` 794, 958
`\on-the-fly` 884
`\once` 771, 777, 786, 958
`OneStaff` 247, 743
`\oneVoice` 224
opcional, material 298
`\open` 156, 430, 495, 499, 935
OpenType, colecciones (`.otc`) 341
OpenType, font feature 821
OpenType, fuente (`.otf`) 341
OpenType, funcionalidad de fuente 344
ópera 394
operación de inversión modal 19
operación de inversión 17
operación de retrogradación 17
operación de transposición 18
operación modal 18
optional, pasaje 298
`opus` 601
oratorio 391
ordenación vertical de las inscripciones 158
órgano, indicación de pedal 935
órgano 425

orientación de las digitaciones 285
orientación, de las digitaciones 285
orientación, de los números de cuerda 285
orientación, modificación de, diagrama de trastes . 466
\oriscus 556, 561
ornamentación 156
ornamento, grupeto circular barrado 156
ornamento, grupeto circular de Haydn 156, 934
ornamento, grupeto circular invertido 156, 934
ornamento, grupeto circular 156, 934
ornamento, grupeto de retardación 159
ornamento, horizontalmente 157
ornamento, mordente, ascendente 156, 934
ornamento, mordente, descendente 156, 934
ornamento, mordente 156, 934
ornamento, semitrino, ascendente 156, 934
ornamento, semitrino, descendente 156, 934
ornamento, semitrino, largo 156, 934
ornamento, semitrino 156, 934
ornamento, signum congruentiae 156
ornamento, trino 156, 934
ornamento 148, 156
orphan-penalty 974
orquesta, notación para 395
orquestal, música 395
orquestales, cuerdas 429
ossia, posicionamiento 259
ossia 256, 264
otomana, música clásica 573
otomana, música 580
ottava 27
\ottava 27, 959
ottavation-numbers 28
ottavation-ordinals 28
ottavation-simple-ordinals 28
ottavationMarkups 28
outer-margin 681, 974
output-count 948
output-def 948
output-filename 593, 974
output-scale 974
output-suffix 593, 948, 974
outside-staff-horizontal-padding 716
outside-staff-padding 716
outside-staff-priority 715, 943
\oval 851
\overlay 838
\override-lines 891
\override 770, 772, 884
\overrideProperty 772, 959
\overrideTimeSignatureSettings 82, 959
overriding property, within text markup 884
\overtie 822

P

\p 160
página, disposición de la 728
página, encabezamiento 605
página, número, independiente para la introducción 684
página, número, por parte de libro 684
página, orientación de la 675
página, pie 605
página, referenciar números de 622
página, salto de 727
página, salto, dentro de un marcado 326
página, salto, en cadencias 93
página, salto, en música sin medida 93
página, saltos manuales 696
\pad-around 337, 838
\pad-markup 337, 838
\pad-to-box 337, 839
\pad-x-left 839
\pad-x-right 840
\pad-x 337, 839
padding (relleno) 766
padding text horizontally 839, 840
padding text 838
page-breaking-system-system-spacing 682, 974
page-breaking 682, 974
page-count 682, 975
\page-link 884
page-number-type 683, 975
page-post-process 975
\page-ref 623, 885
page-spacing-weight 682, 975
\pageBreak 696, 959
\pageTurn 699, 959
pageTurnMinimumRepeatLength 699
pageTurnMinimumRestLength 699
palmas 937
\palmMute 959
\palmMuteOn 959
pandereta 937
Pango 341
panorama, en MIDI 665
papel, tamaño, apaisado 675
papel, tamaño, orientación 675
papel, variable, predefinida 674
paper variables, in \layout block 686
paper-height 676, 975
paper-width 679, 975
\paper 594, 674
papersizename 975
paréntesis en ángulo 218
paréntesis, alteración con 8
paréntesis 298
paralelo, música en 241
\parallelMusic 241, 959
parcial, compás, en himnos 412
parcial, compás 90
\parenthesize 298, 851, 960
parlato, cabezas de nota 44
parlato 401
Parmesan, tipografía 912
parser (analizador sintáctico) 948
\partCombine y letra 238, 380
\partCombine 235, 380, 960
\partCombineApart 237
\partCombineAutomatic 237
\partCombineChords 237
\partCombineDown 960
\partCombineForce 960
partCombineListener 948
\partCombineSoloI 237
\partCombineSoloII 237
\partCombineUnisono 237
\partCombineUp 960
parte, de atril 265

parte, de sección ... 265
parte, individuales ... 265
partes, combinación automática ... 235
partes, combinador de ... 235
partes, combinador, cambiar textos ... 240
\partial ... 90, 155, 192, 960
particellas ... 265
partitura general ... 395
partitura, dentro de un marcado ... 340
partitura, general ... 394
partitura, vocal, añadir guías ... 399
path, drawing ... 852
\path ... 852
patrón de discontinuidad, ligadura de expresión ... 172
\pattern ... 885
pausa, marcas de ... 177
pauta, de coro ... 246
pauta, de percusión ... 244
pauta, escalado de ... 256
pauta, posicionamiento exacto ... 708
pauta ... 244
PDF, empotrar archivos ... 652
PDF, metadatos de ... 611
pedal de órgano, indicaciones ... 425
pedal de órgano ... 156
pedal de mantenimiento, estilo de ... 424
pedal high hat ... 495
pedal splash high hat ... 495
pedal, de piano ... 424
pedal, del arpa, diagrama ... 428
pedal, del arpa ... 428
pedal, derecho ... 424
pedal, estilos de indicación ... 424
pedal, forte ... 424
pedal, high hat ... 156, 937
pedal, indicación de punta ... 425
pedal, indicación de talón ... 425
pedal, indicación mixta ... 424
pedal, indicación por corchete ... 424
pedal, indicación textual ... 424
pedal, indicación, órgano ... 156, 935
pedal, indicación, punta ... 156, 935
pedal, indicación, talón ... 156, 935
pedal, sostenuto ... 424
pedalSustainStyle ... 424
pentagrama único, polifonía de ... 224
pentagrama a la francesa ... 256
pentagrama cruzado, arpegios con paréntesis ... 187
pentagrama cruzado, plica de ... 422
pentagrama, único, con cochete o llave ... 248
pentagrama, único ... 244
pentagrama, cambio automático de ... 419
pentagrama, cambio de ... 421
pentagrama, cambio ... 421
pentagrama, cambios manuales de ... 417
pentagrama, cruzado, trémolo de ... 215
pentagrama, de instrumentos de teclado ... 416
pentagrama, de percusión ... 244
pentagrama, de piano ... 246, 416
pentagrama, detener e iniciar líneas del ... 252
pentagrama, distancia entre los ... 701
pentagrama, fijar el tamaño de ... 688
pentagrama, indicación metronómica debajo ... 89
pentagrama, iniciar ... 244
pentagrama, instanciar ... 244
pentagrama, línea de cambio de ... 421
pentagrama, mofidicar las líneas del ... 252
pentagrama, nuevo ... 244
pentagrama, ocultación de ... 260
pentagrama, resaltado ... 294
pentagrama, símbolo del ... 252
pentagrama, vacío ... 260
pentagramas, anidados ... 250
pentagramas, divisi ... 262
pentagramas, varios ... 246
pequeña, nota ... 272
percusión, clave ... 490, 929
percusión, nota falsa ... 497
percusión, nota fantasma ... 497
percusión, nota muerta ... 497
percusión, nota silenciada ... 497
percusión, nota tapada ... 497
percusión, pauta de ... 244
percusión, personalizada ... 495
percusión ... 490, 492, 937
perfilado en bordes blancos ... 66
Performer_group ... 760
persa, alteraciones accidentales (sori, koron) ... 582
persa, música clásica ... 582
persa, makam ... 573
personaje, nombre ... 395
personalización de los nombres de acorde ... 518
personalizada, marca de ensayo ... 142
personalizada, marca metronómica ... 89
personalizadas, afinaciones de cuerdas ... 455
personalizado, añadir diagramas de traste ... 472
personalizado, diagrama de traste ... 474
personalizado, diagrama de trastes ... 459, 467
personalizados, diagramas de trastes ... 466
\pes ... 562
Petrucci, clave ... 545, 930
Petrucci ... 540, 541
PetrucciStaff ... 550, 744
PetrucciVoice ... 550, 745
\phrasingSlurDashed ... 175
\phrasingSlurDashPattern ... 176, 960
\phrasingSlurDotted ... 175
\phrasingSlurDown ... 175
\phrasingSlurHalfDashed ... 176
\phrasingSlurHalfSolid ... 176
\phrasingSlurNeutral ... 175
\phrasingSlurSolid ... 175
\phrasingSlurUp ... 175
\phrygian ... 25
piano staff ... 247
'piano', estilo de alteraciones de ... 35
piano, música de, matices centrados ... 416
piano, pedales de ... 424
piano, pentagramas de ... 246, 416
piano, sistema de ... 246
'piano-cautionary', estilo de alteraciones ... 36
piano-cautionary ... 36
piano ... 35
PianoStaff ... 416, 419, 743
pie de página ... 598
pie, indicación de ... 156
pie, marca de ... 935
pie, página ... 605
piece ... 601
Pitch_squash_engraver ... 99

\pitchedTrill 188, 960
pitchnames 948, 957
pizzicato, Bartók 432
pizzicato, snap 432
place-fret 871
placing horizontal brackets, around text 850
placing parentheses, around text 851
placing vertical brackets, around text 846
plantilla, música árabe 578
platillo, chino 937
platillo, ride 937
platillo, splash 937
platillo 937
platillos, varios 937
plica abajo 300
plica arriba 300
plica de pentagrama cruzado 422
plica invisible 300
plica neutral 300
plica, de pentagrama cruzado 422
plica, dirección automática sobre la línea media .. 300
plica, dirección predeterminada sobre la línea media 300
plica, dirección 300
plica, en tablatura 447
plica, tachada 150
plica 300
PNG image 850
poet 601
\pointAndClickOff 960
\pointAndClickOn 961
\pointAndClickTypes 961
polifónica, música 229
polifonía, en tablaturas 447
polifonía, en un solo pentagrama 224
polifonía, letra compartida 380
polifonía, voces adicionales 233
polimétrico, compás 93
polimétrico, metro, barras en 94
\polygon 852
\polymetric 93, 961
\popContextProperty 769, 961
porcentaje, contador de repetición 212
porcentaje, repetición, aislada 213
porcentaje, repetición, visibilidad del contador ... 212
porcentaje, repetición 211
'portato', articulación 156, 934
\portato 156, 934
posición y cejilla, indicación para instrumentos de trastes 484
posición, alteración de bajo cifrado 532
posición, en el compás 91
posición, vertical, de grobs 783
posicionamiento exacto de las pautas 708
posicionamiento, de la letra 364
posicionamiento, de letra 259
posicionamiento, de ossia 259
posicionamiento, digitación, mano derecha 483
position, in measure 155
post-eventos 934
PostScript encapsulado, salida de 651
PostScript 338
\postscript 338, 853
potencia, acorde de 486
\pp 160
\ppp 160
\pppp 160
\ppppp 160
\prall 156, 934
\pralldown 156, 934
\prallmordent 156, 934
\prallprall 156, 934
'pralltriller', ascendente, ornamento 156
'pralltriller', descendente, ornamento 156
'pralltriller', largo, ornamento 156
'pralltriller', ornamento 156
\prallup 156, 934
\preBend 443, 961
\preBendHold 443, 961
precaución, alteración de 8
predefinedDiagramTable 478
\predefinedFretboardsOff 480
\predefinedFretboardsOn 480
predefinido, variables de papel 674
predeterminada, altura 54
predeterminada, barra de compás 131
predeterminada, dirección vertical (-) 783
predeterminada, duración de la nota 54
predeterminada, línea divisoria, cambiar 129
predeterminadas, cambio de propiedades de contexto 753
predeterminado, estilo de las alteraciones 32
predeterminado, nombres de nota 8
presentación, interfaz de la 765
prima volta 192
primera y segunda vez, acortar casilla 203
primera y segunda vez, casillas en pautas adicionales 204
primera y segunda vez, corchetes y ligaduras en ... 64
primera y segunda vez, repetición 192
primera y segunda vez, y letra 374
principiantes, música para 46
print-all-headers 684, 975
print-first-page-number 683, 975
print-page-number 683, 975
printAccidentalNames 301
printNotesLanguage 301
printOctaveNames 301
prob 948
progresiva, barras 118
property-defaults 975
\property-recursive 885
\propertyOverride 961
\propertyRevert 961
\propertySet 961
\propertyTweak 962
\propertyUnset 962
propiedad de grob 770
propiedad, compartida 947
propiedad, inmutable 947
propiedad, modificar 768
propiedades mutables 947
propiedades 774
\pt 784, 975
pulgadas (in) 784
pulgar, digitación 156, 285, 935
pull off (deslizamiento de salida) 453
pulso, agrupación 111
pulso, repetición 211
pulsos por minuto 86

'punta', indicación de pedal ... 425, 935
punta, marca de pedal ... 156
punteada, bara de compás ... 131
punteada, ligadura de fraseo ... 175
puntillo, cambiar número de ... 55
puntillo, mover horizontalmente ... 233
puntillo, nota con ... 54
puntos (pt) ... 784
puntos de control en curvas de Bézier ... 808
puntos, grandes (bp) ... 784
puntos, ligadura de ... 65, 172
puntuación, en la letra ... 350
puntuación ... 350
\push-to-tag ... 638, 885
\pushContextProperty ... 769, 962
\pushToTag ... 637, 962
\pushToTagMarkup ... 639, 962
\put-adjacent ... 840
putting space around text ... 838

Q

q (repetición de acordes) ... 220, 438
\qr-code ... 886
QR code ... 886
quebrado, acorde ... 183
\quilisma ... 556, 561
quinta vacía, acorde de ... 486
quitar música etiquetada ... 630
quotedCueEventTypes ... 271
quotedEventTypes ... 271
\quoteDuring, and tags ... 632
\quoteDuring ... 268, 272, 962

R

r (silencio) ... 68
rítmica, pauta ... 245
R (silencio de compás completo) ... 72
racimo (cluster) ... 223
ragged-bottom ... 676, 976
ragged-last-bottom ... 676, 976
ragged-last ... 680, 728, 976
ragged-right ... 680, 727, 976
\raise ... 331, 840
\raiseNote ... 515, 962
raising text ... 840
rango de alturas ... 41
rasgueado, mostrar ritmos ... 99
Ratisbona, Editio ... 541
raya, barra de compás ... 121
recordatoria, alteración ... 8
recto, corchete, antiguo ... 116
recto, corchete, moderno ... 116
recto, corchete ... 115
redonda, silencios de, para un compás completo ... 72
\reduceChords ... 99, 962
Referencia de funcionamiento interno ... 741
referenciar contextos ... 746
referencing page label, in text ... 890
referencing page number, in text ... 884, 885
registro, símbolo, acordeón ... 426
regulador de Ferneyhough ... 166
regulador de tipo constante ... 166
regulador en ángulo ... 801
regulador, acampanado ... 166
regulador, al niente ... 166
regulador, desplazar extremos ... 165
regulador, longitud mínima ... 164
regulador ... 160
reguladores sobre líneas divisorias ... 164
rejilla, líneas de, modificación ... 305
rejilla, líneas de ... 304
relativa, especificación de octava ... 4
relativa, introducción de la octava, y acordes ... 7
relativa, octava, escritura y transposición ... 6
\relative ... 4, 6, 17, 420, 645, 962
relativo, acordes en altura ... 219
relativo, modo, \autoChange y ... 420
relativo ... 4
relleno, alrededor de texto ... 337
relleno, en marcado, vertical ... 332
relleno ... 766
remove-empty ... 261
remove-first ... 261
remove-grace-property ... 150
remove-layer ... 262
\remove-with-tag ... 638, 886
\remove ... 752
\RemoveAllEmptyStaves ... 261, 969
\RemoveEmptyStaves ... 261, 969
\removeFromTagGroup ... 636, 962
\removeWithTag ... 630, 963
Renacimiento, música del ... 249
\repeat percent ... 211
\repeat segno ... 131, 197
\repeat tremolo ... 214
\repeat unfold ... 191
\repeat volta ... 131, 191, 192
repeat, al fine ... 198
repeat, D.S. ... 199
repeat, dal segno ... 199
\repeat ... 191, 192
repeatCommands ... 207
repeatCountVisibility ... 212
\repeatTie ... 64, 376
repetición con q ... 438
repetición manual ... 207
repetición, al alzar ... 192
repetición, alla coda ... 198
repetición, ambigua ... 206
repetición, anidada ... 206
repetición, barra de compás al comienzo de la pieza ... 192
repetición, barra de compás ... 131
repetición, barra de ... 120
repetición, barra, subyacente ... 131
repetición, cambiar finales alternativos ... 207
repetición, cambiar número de ... 207
repetición, comienzo de ... 207
repetición, con anacrusa ... 192
repetición, con comprobación de compás ... 192
repetición, con finales alternativos ... 192
repetición, con ligadura de unión ... 64
repetición, con q ... 220
repetición, con segno ... 201
repetición, contador de tiempo de la ... 206
repetición, contador, porcentaje ... 212
repetición, corta ... 211

repetición, D.C., manual ... 210
repetición, D.C. ... 199
repetición, D.S., forzado al comienzo ... 201
repetición, D.S., manual ... 210
repetición, da capo, manual ... 210
repetición, da capo ... 199
repetición, dal segno, forzado al comienzo ... 201
repetición, dal segno, manual ... 210
repetición, de barra inclinada ... 211
repetición, de compás ... 211
repetición, de la letra con finales alternativos ... 374
repetición, de la letra con primera y segunda vez ... 374
repetición, de porcentaje ... 211
repetición, de pulso ... 211
repetición, de trémolo ... 214
repetición, desplegada ... 191
repetición, doble, estilo para volta ... 132
repetición, e poi la coda ... 198
repetición, explícita ... 191
repetición, final de ... 207
repetición, letras y números de compás ... 204
repetición, números de compás alternativos ... 204
repetición, números de compás de primera y segunda vez ... 204
repetición, porcentaje, aislada ... 213
repetición, porcentaje, visibilidad del contador ... 212
repetición, primera y segunda vez, e indicación de compás ... 85
repetición, sencilla ... 191
repetición, y glissando ... 206
repetición, y glissandos ... 182
repetición, y letra ... 370
repetición, y ligadura de expresión ... 206
repetición, y números de compás ... 206
repetición ... 123
repeticiones en MIDI ... 662
repetido, acorde, eliminación ... 517
repetido, acorde, suprimir ... 477
\replace-with-tag ... 638, 887
\replace ... 822
\replaceWithTag ... 638, 963
\replaceWithTagMarkup ... 639, 963
resaltado, pentagrama ... 294
reservados, impresión de caracteres ... 323
reset-footnotes-on-new-page ... 684, 976
\resetRelativeOctave ... 7, 963
\resetTagGroup ... 635, 963
\resetTagGroups ... 635, 963
respiración, cambiar símbolo de marca de ... 177
respiración, símbolo, lista ... 936
respiraciones ... 177
\responsum ... 557, 963
rest, within text, by duration ... 863
rest, within text, by log and dot-count ... 863
\rest-by-number ... 863
rest-event ... 271
\rest ... 68, 863
restaurar, las propiedades predeterminadas del tipo de compás ... 82
restNumberThreshold ... 279
restrainOpenStrings ... 438
retardación, grupeto de, ornamento ... 159
retrogradación, transformación de ... 17
\retrograde ... 17, 963
reverberación, en MIDI ... 666
\reverseturn ... 156, 934
reversión de una sobreescritura ... 771
\revert ... 771
\revertTimeSignatureSettings ... 82, 963
\rfz ... 160
rgb-color ... 293
RGB, color ... 293
\rheel ... 425
\rheeltoe ... 425
rhythm, in text ... 864
\rhythm ... 743, 744, 745, 864
RhythmicStaff ... 245, 744
\right-align ... 330, 841
right-aligning text ... 841
\right-brace ... 887
\right-column ... 841
right-margin ... 680, 976
\rightHandFinger ... 482, 963
ritmo, de la melodía, mostrar ... 98
\romanStringNumbers ... 430, 434
rombo, cabezas de nota ... 44
rombo, cabezas en forma de ... 430
romper la alineación vertical de matices ... 167
rotar un objeto ... 801
\rotate ... 841
rotating text ... 841
\rounded-box ... 336, 854
\rtoe ... 425
\rtoeheel ... 425

S

s (invisible rest) ... 70
séptima, acorde de ... 511
sílaba, duración automática de ... 353
símbolo de séptima mayor ... 521
símbolo, break-align ... 723
símbolo, cambiar, marca de respiración ... 177
símbolo, coda variante ... 156
símbolo, coda ... 156
símbolo, de dirección ... 111
símbolo, segno ... 156
símbolos no musicales ... 337
\sacredHarpHeads ... 48
\sacredHarpHeadsMinor ... 48
salida de gráficos de vector escalables ... 651
salida de PostScript encapsulado ... 651
salida, definiciones de ... 742
salmo ... 403, 407
saltar el tipografiado ... 650
saltar notas en la letra ... 375
salto de línea, a intervalos ... 694
salto de línea, manual ... 692
salto de línea ... 692
salto de página, gestión con una voz adicional ... 695
salto de página ... 727
salto, en música sin medida ... 93
\sans ... 822
SATB ... 391
\scale ... 855
\scaleDurations ... 63, 93, 963
scaling markup ... 855
scaling text ... 842
Score, contexto, sustitución de ... 761
\score-lines ... 891

score-markup-spacing 678, 976
score-system-spacing 678, 976
score-title-properties 976
score-title 976
Score 742
\score 590, 594, 865
Score_engraver 761
Score_performer 761
scoreTitleMarkup 606, 976
script-priority 158, 944
sección de espaciado nueva 719
sección, barra de compás, en canto kievano 564
sección, barra de compás 131, 147
sección, etiqueta 316
seconda volta 192
\section 131, 147, 564
\sectionLabel 198, 316, 963
secundaria, barra de compás 131
'segno' signo, con repeticiones 197
'segno', barra de compás 122, 131, 201
segno, forzado al comienzo 201
segno, forzado al inicio 209
'segno', signo manual 209
'segno', signo 156, 936
segno 156, 199
\segno 156, 866, 936
\segnoMark 131, 201, 209, 963
seguimiento de voz 421
selección del tamaño de fuente (de notación) 280
self-alignment-X 701
sello, eliminar 794
sello 949
Semai, forma 578
semi abierta, articulación 156
semi abierto, high hat 156
semi-bemol, apariencia 575
semi-bemol, símbolo, persa (koron) 582
semi-bemol 12
semi-bemoles 8
semi-sostenido, símbolo, persa (sori) 582
semi-sostenido 12
semi-sostenidos 8
'semiabierta', articulación 935
semiabierto, high hat 937
'semicirculus', articulación gregoriana 554
'semicirculus', articulación, gregoriana 936
\semicirculus 554, 936
\semiflat 866
\semiGermanChords 518
\semisharp 866
semitrino ascendente, ornamento 156, 934
semitrino descendente, ornamento 156, 934
semitrino doble 156
semitrino largo, ornamento 934
semitrino, ornamento 934
semitrino 156
sencilla, barra de compás 131
separación de sistemas, marca de 251
separado, texto 321
\serif 823
sesqui-bemol 12
sesqui-sostenido 12
\sesquiflat 866
\sesquisharp 866
set-global-staff-size 688
\set 104, 768, 772
setting extent of text object 890
setting horizontal text alignment 832
setting subscript, in standard font size 819
setting superscript, in standard font size 819
\sf 160
\sff 160
\sfz 160
shape note, merging 50
\shape 808, 963
\sharp 866
\shiftDurations 964
\shiftOff 229
\shiftOn 229
\shiftOnn 229
\shiftOnnn 229
short-indent 266, 681, 976
\shortfermata 156, 935
showFirstLength 650, 948
\showKeySignature 501
showLastLength 650, 948
\showStaffSwitch 421
sidestick 937
signo, barra de compás 122, 201
signo, coda de variación 156, 936
signo, coda, con repeticiones 197
signo, coda, manual 209
signo, coda, variante 203
signo, coda 156, 936
signo, con repeticiones 201
signo, segno, con repeticiones 197
signo, segno, manual 209
signo, segno 156, 936
signo, snappizzicato 156, 935
'signum congruentiae', notación mensural 936
'signum congruentiae', ornamento 156
\signumcongruentiae 156, 936
silbato 937
silenciada, percusión, nota 497
silencio antiguo 548
silencio de separación 70
silencio invisible 70
silencio multicompás, colocación 74
silencio, colisiones 76
silencio, combinar 235
silencio, condensación de, ordinario 76
silencio, de compás completo 72
silencio, de redonda para un compás completo 72
silencio, división de 97
silencio, eclesiástico 279
silencio, especificar la posición vertical 68
silencio, estilo de 69
silencio, multicompás, marcado 75
silencio, multicompás 72
silencio 68
silencios de compás completo 68
silencios multicompás 68
silencios, indicar duraciones 68
simple text string, with tie characters 868
\simple 823
simplificada, notación 46
simultáneas, ligaduras de expresión 171
simultáneas, ligaduras de fraseo 175
simultáneas, notas y alteraciones 39
\single 616, 773, 774, 777, 964

sintaxis del marcado ... 323
sistema de coro ... 246
sistema de piano ... 246
sistema, delimitadores de inicio anidados ... 250
sistema ... 246
sistemas, separador, marca de ... 251
\skip ... 70, 375, 964
skipBars ... 278
skipTypesetting ... 650
slashChordSeparator ... 520
slashed digit ... 887
\slashed-digit ... 887
\slashedGrace ... 148, 964
\slashSeparator ... 684
\slashturn ... 156, 934
slur-event ... 271
\slurDashed ... 172
\slurDashPattern ... 172, 964
\slurDotted ... 172
\slurDown ... 171
\slurHalfDashed ... 172
\slurHalfSolid ... 172
\slurNeutral ... 171
\slurSolid ... 172
\slurUp ... 173
\small ... 283, 330, 823
\smallCaps ... 823
\smaller ... 327, 330, 823
smob ... 949
snap, pizzicato ... 432
'snappizzicato', articulación ... 156
'snappizzicato', signo de ... 935
\snappizzicato ... 156, 935
sobreescritura de grob ... 66
sobreescritura de un objeto ... 795
sobreescritura para un solo momento ... 771
sobreescritura, reversión ... 771
Sol, clave ... 20
Solesmes ... 541
solo, partes ... 235
sonido ... 655
soprano, clave ... 20, 928
sori ... 582
sost. ... 424
sostenido, doble ... 8
sostenido ... 8
sostenuto, pedal ... 424
\sostenutoOff ... 424
\sostenutoOn ... 424
\sourcefileline ... 596
\sourcefilename ... 596
Southern Harmony, cabezas de nota con forma ... 48
\southernHarmonyHeads ... 48
\southernHarmonyHeadsMinor ... 48
\sp ... 160
space-alist ... 723
Span_stem_engraver ... 422
spanner (objeto de extensión) ... 786
\spp ... 160
Sprechgesang ... 401
'staccatissimo', articulación ... 156, 934
\staccatissimo ... 156, 934
'staccato', articulación ... 156, 934
\staccato ... 156, 934
stacking text, in a column ... 829
staff brace, in markup ... 883
staff, piano ... 247
staff-affinity ... 701
staff-height ... 976
staff-padding ... 286
\staff-space ... 785, 977
staff-staff-spacing ... 701
Staff ... 743
Staff_collecting_engraver ... 143, 321
Staff_symbol_engraver ... 260
staffgroup-staff-spacing ... 701
StaffGroup ... 743
StaffGrouper ... 701
\staffHighlight ... 294, 964
StandaloneRhythmScore ... 743
StandaloneRhythmStaff ... 744
StandaloneRhythmVoice ... 745
start-repeat ... 207
startAcciaccaturaMusic ... 150
startAppoggiaturaMusic ... 150
startGraceMusic ... 150
\startGroup ... 306
\startOptionalMaterial ... 298
\startStaff ... 252, 256, 743
\startTextSpan ... 315, 793, 794
\startTrillSpan ... 187
\stdBass ... 875
\stdBassIV ... 876
\stdBassV ... 877
\stdBassVI ... 878
stem-spacing-correction ... 718
\stemDown ... 300
stemLeftBeamCount ... 115
\stemNeutral ... 300
stemRightBeamCount ... 115
\stemUp ... 300
stencil ... 949
\stencil ... 888
stopAcciaccaturaMusic ... 150
stopAppoggiaturaMusic ... 150
stopGraceMusic ... 150
\stopGroup ... 306
\stopOptionalMaterial ... 298
\stopped ... 156, 495, 499, 935
\stopStaff ... 252, 256, 260, 743
\stopStaffHighlight ... 294
\stopTextSpan ... 315, 793, 794
\stopTrillSpan ... 187
\storePredefinedDiagram ... 473, 478, 964
strictBeatBeaming ... 110
\string-lines ... 892
stringNumberOrientations ... 285
\stringTuning ... 455, 964
stringTunings ... 455, 469
strokeFingerOrientations ... 285, 483
\stropha ... 556, 561
\strut ... 888
style, doble repetición para volta ... 132
\styledNoteHeads ... 964
sub-bajo, clave ... 928
\sub ... 329, 824
subíndice ... 329
subdivideBeams ... 109
subdivisión, de barra ... 109
\submeasureBarsOn ... 131

submeasureStructure 131
subscript text 824
subsubtitle 601
subtitle 601
subyacente, barra de repetición 131
suggestAccidentals 159, 549
\super 329, 824
superíndice 329
superscript text 824
supresión, acorde repetido 477
sus 514
\sustainOff 424
\sustainOn 424
sustitución, de cadenas, automática 649
sustitución, función, y escritura de octava relativa . 645
SVG, salida 651
swing.ly, script 667
swing 864
system-count 682, 977
system-separator-markup 684, 977
system-system-spacing 678, 977
systems-per-page 682, 977

T

título 598
\tabChordRepeats 438, 964
\tabChordRepetition 964
\tabFullNotation 437
tabla de contenidos 623
tablatura de cuartos de tono 457
tablatura de microtonos 457
tablatura predeterminada 436
tablatura y armónicos 441
tablatura y deslizamientos 451
tablatura, afinaciones personalizadas 455
tablatura, afinaciones predefinidas de 455
tablatura, armónicos y 448
tablatura, barras y 447
tablatura, clave de 457
tablatura, clave 929
tablatura, curvatura de cuerda 443
tablatura, de bajo 455
tablatura, de banjo 433, 455, 487
tablatura, de contrabajo 455
tablatura, de guitarra 455
tablatura, de mandolina 455
tablatura, de ukelele 455
tablatura, de violín 455
tablatura, de viola 455
tablatura, de violoncello 455
tablatura, glisando en acorde de 452
tablatura, hammer on (deslizamiento de entrada) . 453
tablatura, laúd 488
tablatura, personalizada 455
tablatura, plicas y 447
tablatura, principios básicos 436
tablatura, pull off (deslizamiento de salida) 453
tablatura, y polifonía 447
tablatura 245, 433
\table-of-contents 626, 893
\table 892
TabStaff 245, 436, 744
TabVoice 436, 745
tachada, plica 150
\tag, and quoting music 632
\tag-list 640, 893
tag 630
\tag 630, 888, 964
\tagGroup 634, 965
\tagGroupRef 634, 965
tagline 601, 977
'talón', indicación de pedal 425, 935
talón, marca de pedal 156
tam tam 937
tamaño de la página 674
tamaño del papel 674
\taor 501
'tapada', articulación 499, 935
tapada, nota, en instrumento de trastes 485
tapada, percusión, nota 497
taqasim 578
'teaching', estilo de alteraciones 38
teaching 38
tecla, pentagramas para instrumentos de 416
teclado, matices centrados 416
teclado, pentagramas para instrumentos de 416
\teeny 283, 330, 824
tempo, cambio, sin indicación metronómica 89
tempo, with rhythm 864
tempo 86
\tempo 86
tempoCountPrecision 86
\temporary 777, 786, 965
tenor Sol, clave 928
tenor varC, clave 928
tenor, clave, coral 21
tenor, clave 20, 928
'tenuto', articulación 156, 934
\tenuto 156, 934
tesitura, línea de separación 43
tesitura, por voz 42
tesitura, varias voces 42
tesitura 41
text column, left-aligned 837
text column, right-aligned 841
text, with a tie above 822, 824
text, with a tie below 824, 826
\text-accidental 867
\text-doubleflat 867
\text-doublesharp 867
\text-flat 867
text-font-size 327, 977
\text-natural 867
\text-sharp 867
Text_mark_engraver 321
\textEndMark 317, 965
\textLengthOff 74, 75, 314
\textLengthOn 74, 75, 167, 314
\textMark 317, 965
texto ajustado a palabras 335
texto de marcado ajustado a palabras 335
texto del nivel superior 321
texto destacado 314
texto independiente 321
texto justificado 335
texto separado 321
texto, al final de la partitura 317
texto, al inicio de una línea 317

texto, alineación de ... 330
texto, alineación horizontal ... 330
texto, alineación vertical de inscripciones de ... 167
texto, alineación vertical ... 331
texto, alineación ... 336
texto, ancho de línea ... 335
texto, centrar en la página ... 334
texto, codificación ... 647
texto, de octava ... 29
texto, de varias líneas ... 334
texto, decorar ... 336
texto, elemento de extensión de ... 315
texto, elementos no vacíos de ... 314
texto, en casilla de repetición ... 208
texto, en columnas ... 322, 334
texto, en otros idiomas ... 311
texto, en todos los pentagramas ... 321
texto, enmarcar ... 336
texto, entre las notas ... 317
texto, entrecomillado en modo de marcado ... 323
texto, extendido sobre más de una página ... 326
texto, extensión de, matiz, personalizar ... 316
texto, formateado de los elementos de extensión ... 315
texto, fuera del margen ... 314
texto, indicaciones de ... 317
texto, inscripciones de ... 314
texto, mantener dentro del margen ... 314
texto, marca, alineación ... 320
texto, marca, apilamiento vertical ... 319
texto, marca, debajo del pentagrama ... 319
texto, marcado de, dentro de las ligaduras ... 174
texto, marcado de ... 323
texto, matiz de estilo de ... 168
texto, objeto ... 311
texto, relleno de ... 337
texto, romper alineación vertical ... 167
texto, sobre la barra de compás ... 316, 317
texto, sobre un silencio multicompás ... 73
texto, tamaño de ... 327
\textSpannerDown ... 315
\textSpannerNeutral ... 315
\textSpannerUp ... 315
\thumb ... 156, 285
tie, above text ... 822, 824
tie, below text ... 824, 826
\tie ... 824
TieColumn ... 67
\tied-lyric ... 868
\tieDashed ... 65
\tieDashPattern ... 65, 965
\tieDotted ... 65
\tieDown ... 65
\tieHalfDashed ... 65
\tieHalfSolid ... 65
tiempo, administración ... 155
tiempo, control del (dentro de la partitura) ... 155
tiempo, marca de, para glissando ... 180
tiempo, repeticiones y contador de ... 206
\tieNeutral ... 65
\tieSolid ... 65
\tieUp ... 65
tieWaitForNote ... 66
timbala abierta ... 937
timbala aguda ... 937
timbala grave ... 937
timbala tapada ... 937
timbala ... 937
\time ... 79, 104, 965
\timeAbbrev ... 965
\times ... 966
timeSignature ... 93
timing ... 155
Timing ... 155, 743
\tiny ... 283, 330, 825
tiple, clave ... 928
tipografía Parmesan ... 912
tipografía, Emmentaler ... 912
tipografía, Feta ... 912
tipografía, fijar el tamaño de ... 688
tipografía ... 947
tipografiado, saltar ... 650
tipografiar texto ... 323
title ... 601
tocFormatMarkup ... 626, 977
tocIndentMarkup ... 626, 977
\tocItem ... 626, 966
tocItemMarkup ... 626, 977
\tocItemWithDotsMarkup ... 625
tocTitleMarkup ... 626, 977
\toe ... 156, 426, 935
toeHeelStyle ... 425
tom tom agudo ... 937
tom tom de suelo ... 937
tom tom grave ... 937
tom tom medio ... 937
tom tom ... 937
top-margin ... 676, 977
top-markup-spacing ... 678, 977
top-system-spacing ... 678, 977
toplevel-bookparts ... 948
toplevel-scores ... 948
trémolo, indicación de ... 214
trémolo, pentagrama cruzado ... 215
trémolo ... 214
tríada ... 511
transcripción de música mensural ... 249
transcripción de mensural a moderno ... 570
transcripción, moderna del canto gregoriano ... 408
transformación de retrogradación ... 17
transformación, modal ... 18
\translate-scaled ... 332, 842
\translate ... 332, 841
translating text ... 841, 842
transparencia, semi ... 292
\transparent ... 888
transparente, nota ... 291
transparentes, hacer los objetos ... 795
transportada, visibilidad de una clave ... 800
transporte de diagramas de trastes ... 471
transporte modal ... 18
transporte ... 14, 18
\transpose ... 6, 14, 17, 966
\transposedCueDuring ... 275, 966
transposición de MIDI ... 30
transposición instrumental ... 30
transposición modal ... 18
transposición, alturas, 'inteligente' ... 15
transposición, de alturas ... 14
transposición, de notas ... 14
transposición, de octava, opcional ... 21

transposición, y escritura de octava relativa 6
transposición 14
\transposition 30, 268, 966
transpositor, instrumento 14, 15, 30
transpositoras, claves 21
traste, definir predeterminado 475
traste, diagrama personalizado de 474
traste, diagramas de, personalizados 459
traste 438
trastes personalizados, añadir diagramas de 472
trastes, añadir digitaciones a diagramas de 481
trastes, armónicos en instrumento de 485
trastes, diagrama automático de 480
trastes, diagrama de, cambiar la orientación 466
trastes, diagrama de, con nombres de acorde 470
trastes, diagrama de, personalizado 467
trastes, diagrama de, transporte 471
trastes, diagrama, zurdo 462
trastes, diagramas de 459
trastes, diagramas personalizados de 466
trastes, instrumentos con, afinaciones predefinidas . 455
trastes, instrumentos de, digitaciones de la mano derecha 482
trastes, instrumentos de, formas de acordes 473
trastes, instrumentos de, indicación de la posición y cejilla 484
trastes, notas tapadas en instrumento de 485
trastes, tabla alternativa 478
tre corde 424
\treCorde 424
tresillo, formato de 58
tresillo 56
triángulo 937
\triangle 337, 855
\trill 156, 187, 934
trino con nota y alteración 188
trino, ornamento 934
trino 156, 187
trinos de nota explícita 188
\tripletFeel 667
trucaje, de las notas de adorno 149
trucaje, nota de adorno 150
trucar un punto de control 775
trucar 773
true (#t) 978
tuplet-slur 57
\tuplet 56, 93, 966
\tupletDown 56
\tupletNeutral 56
TupletNumber 58
\tupletSpan 58, 966
tupletSpannerDuration 58
\tupletUp 56
turco, makam 573
\turn 156, 934
Turquía, música clásica 573
Turquía, música de 580
Turquía, nombres de notas en 580
\tweak, relación con \override 772
\tweak 772, 773, 966
two-sided 680, 978
\type 758, 759
\typewriter 825

U

U.C. 424
ukelele 460
ultima volta 192
una corda 424
\unaCorda 424
\underline 327, 825
underlining text 825
\undertie 826
\undo 777, 967
\unfolded 194, 967
\unfoldRepeats 194, 662, 967
\unHideNotes 291
Unicode 647
unidades, de medida 784
universal-color 293, 912
\unless 610, 868
\unset 768
upbeat 90
\upbow 156, 430, 935
\upmordent 156, 934
\upprall 156, 934
\upright 826
URL link, as QR code 886
UTF-8 647

V

vacía, acorde de quinta 486
vacío, acorde 150, 162, 417
vacío, pentagrama 260
valores predeterminados, desplazar 775
varC, clave 928
varcoda 156
\varcoda 156, 868, 936
\varheel 156, 426, 935
variable de Scheme 948
variable del analizador sintáctico 948
variable global 948
variable, y \book 592
variable, y \bookpart 592
variable 595
variables, uso de las 628
variante del signo 'coda' 203
varias ligaduras de expresión 171
varias ligaduras de fraseo 175
varias voces 229
varias, líneas, marcado 334
varias, líneas, texto 334
Variation Font (VF), formato 341
varios matices sobre una nota 161
\vartoe 156, 426, 935
Vaticana, Editio 540, 541
VaticanaLyrics 552, 745
VaticanaScore 552, 743
VaticanaStaff 245, 552, 744
VaticanaVoice 552, 744
\vcenter 842
\verbatim-file 888
\version 596
verso, número de 382
\versus 557, 967
vertical, alineación de texto 331
vertical, desplazamiento, en marcado 331

vertical, dirección, forzada, de grobs 783
vertical, dirección, predeterminada, de grobs 783
vertical, distancia, elementos del bajo cifrado 534
vertical, espaciado, depuración 734
vertical, espaciado 701, 728
vertical, inscripciones, alineación 167
vertical, línea, entre pentagramas 304
vertical, ligadura 185
vertical, matices, alineación, romper 167
vertical, matices, alineación 167
vertical, orden de las inscripciones 158
vertical, posición, de las indicaciones dinámicas .. 163
vertical, relleno, en marcado 332
VerticalAxisGroup 701
vertically centering text 842
\verylongfermata 156, 935
\veryshortfermata 156, 935
vibraslap 937
viento, instrumento de 498
viento, instrumento, diagrama, lista de claves 508
viento, instrumento, diagrama, lista 506
viento, instrumento, diagrama, modificar 507
viento, instrumentos, símbolos de digitación 499
violín clave 928
violín, clave 20
\virga 556, 561
\virgula 554
visibilidad de los grupos especiales 57
visibilidad de un objeto 794
visibilidad de una clave transportada 800
visibilidad, corchete de grupo especial 59
vocal, partitura, añadir guías 399
vocal, partitura 394
vocal, transición de 362
voces, alteraciones en varias 35
voces, arpegio entre 186
'voice', estilo de alteraciones 34
Voice 224, 744
voice 34
\voiceFour 224
\voiceFourStyle 229
\voiceNeutralStyle 229
\voiceOne 224
\voiceOneStyle 229
\voices 228, 967
\voiceThree 224
\voiceThreeStyle 229
\voiceTwo 224
\voiceTwoStyle 229
\void 669, 967
volta, doble repetición, estilo 132
volta, prima 192
volta, repetición, barra de compás al comienzo de la pieza 192
volta, repetición, debajo de los acordes 522
volta, seconda 192
volta, ultima 192
\volta-number 827
volta 191, 192
\volta 192, 194, 967
Volta_engraver 204
\vowelTransition 362
voz adicional, en música polifónica 233
voz adicional, para la gestión de los saltos 695
voz, citar 272
voz, con octavación 29
voz, dividida 393
voz, estilo 229
voz, \partCombine con \autoBeamOff 103
voz, seguimiento de 421
voz, tesitura 42
voz, varias 229
voz 224
\vshape 810, 967
\vspace 332, 842

W

Walker, cabezas de nota con forma 48
\walkerHeads 48
\walkerHeadsMinor 48
whiteout (circundado en blanco) 795
\whiteout 889
\whiteTriangleMarkup 519
\with-color 292, 889
\with-dimension-from 890
\with-dimension 890
\with-dimensions-from 890
\with-dimensions 890
\with-link 890
\with-outline 890
\with-string-transformer 827
\with-true-dimension 890
\with-true-dimensions 891
\with-url 856
\with 752, 756
\withMusicProperty 967
\withRelativeDir 967
\woodwind-diagram 872
\wordwrap-field 844
\wordwrap-lines 326, 894
\wordwrap-string 844
\wordwrap 335, 843

X

x11, color de 294
x11-color 292, 294, 908
X-offset 701
\xNote 45, 967
\xNotesOff 45
\xNotesOn 45

Z

zurdo, diagrama de trastes 462