

Versión 0.0000000001

Paquete μS LuaT $\text{\text{E}}\text{\text{X}}$

Musicología en castellano en LuaL $\text{\text{A}}\text{\text{T}}\text{\text{E}}\text{\text{X}}$

Pablo Herrera

21 de noviembre de 2021

Índice general

1	Presentación	3
2	Requisitos	3
3	Uso	3
4	Símbolos (comandos)	4
4.1	Dinámicas	4
4.2	Duraciones	4
4.2.1	Figuras	4
4.2.2	Silencios	5
4.2.3	Ritmo	5
4.3	Alteraciones	5
4.4	Claves	6
4.5	Compases	6
4.6	Cuerdas	6
4.7	Cifrados	7
4.7.1	Cifrado americano	7
4.7.2	Cifrados de grados y de funciones	7
5	Diagramas	8
5.1	Diagramas de guitarra	8
5.2	Diagramas de teclado	9
5.3	Diagramas seriales	9
6	Partituras y fragmentos	9
6.1	Música antigua	9
6.1.1	Canto gregoriano	9
6.1.2	Bajo continuo en línea	10
6.2	Fragmentos musicales en línea	11
6.3	L $\text{\text{A}}\text{\text{T}}\text{\text{E}}\text{\text{X}}$ como editor de partituras	11
6.4	Particularidades de la escritura musical	12
6.4.1	Música y letra	12

6.4.2	Cifrado de acordes	13
6.4.3	Bajo continuo	13
6.4.4	Análisis schenkeriano	13

Índice de cuadros

1	Dinámicas.	4
2	Las figuras y los comandos que las producen.	4
2	Las figuras y los comandos que las producen.	5
3	Símbolos de silencios.	5
4	Símbolos de alteraciones más frecuentes.	6
5	Símbolos de claves en línea.	6
6	Escritura de compases.	6
7	Cuerdas de la guitarra.	7
8	Cifrado americano: tipos de acordes más usados.	7
9	Notación en código gabc (Gregorio).	10
10	Grados melódicos en notación schenkeriana.	13

1 Presentación

Dos son las principales motivaciones que impulsaron el presente trabajo: la búsqueda de transparencia para el usuario, y la comodidad de comandos en nuestra lengua madre, el castellano. Si bien las opciones para abordar la temática de la escritura musical y musicológica en \LaTeX , como suele ser en el universo del *software libre*, son múltiples y variadas, nuestra experiencia nos ha hecho inclinarnos por la combinación de dos paquetes que si bien no dan cuenta de *todo* lo que en el particular mundillo de los textos musicológicos acontece sí cubre la inmensa mayoría de las necesidades de escritura en este campo. Tales paquetes son `lilyglyphs` y `lyluatex`. El primero otorga acceso a los glifos de la tipografía *Emmentaler*, fuente usada por el refinado programa de notación musical `lilypond`; el segundo paquete —`lyluatex`— pone a disposición del usuario de \LaTeX un método para incluir notación musical utilizando `lilypond` para generar las imágenes e incrustarlas en los documentos. El uso de este último paquete nos restringe a compilar exclusivamente con el motor \LuaTeX , pero es una restricción que bien vale la pena, ya que en \LuaTeX encontramos toda la patencia de \LaTeX modernizada por la capacidad de utilizar cualquier tipografía instalada en el sistema.

2 Requisitos

Indispensable para el correcto funcionamiento de \mu\LaTeX es que \LaTeX esté instalado con los paquetes `harmony`, `lilyglyphs` y `lyluatex`. El primero otorga la posibilidad de escribir acordes cifrados y ciertos símbolos musicales; el segundo accede por completo a la tipografía *Emmentaler* y combina glifos de dicha fuente tipográfica; el tercer paquete necesario permite hacer un amplio uso de `lilypond` para tipografiar fragmentos musicales o partituras completas. Obvio es a esta altura que también es requisito indispensable tener instalado en el sistema GNU/LilyPond. Y por último, a los fines de sumarle a `lilypond` la capacidad de escribir cifrado funcional —el creado por Hugo Riemann—, utilizaremos `harmonyily`, librería de Karsten Reincke.

3 Uso

Para usar \mu\LaTeX alcanza con incluir en el preámbulo del documento:

```
\usepackage{musluatex}
```

y compilar con:

```
$ lualatex --shell-escape [DOCUMENTO.TEX]
```

Un ejemplo mínimo, del estilo «Hola mundo», a continuación:

```
\documentclass{article}
\usepackage{musluatex}

\begin{document}
\acorde.VII..\bemo17... Hola \muses{do' re' mi' r\fermata} mundo \compas{3}{4}.
\end{document}
```

con el cual se obtiene:



4 Símbolos (comandos)

4.1 Dinámicas

Las dinámicas se escriben con el comando `\dinamica{<dinámica>}`, tal como indica el siguiente cuadro:

Cuadro 1: Dinámicas.

Dinámica	Comando	Comando original (lilyglyphs)
<i>ppp</i>	<code>\dinamica{ppp}</code>	<code>\lilyDynamics{ppp}</code>
<i>pp</i>	<code>\dinamica{pp}</code>	<code>\lilyDynamics{pp}</code>
<i>p</i>	<code>\dinamica{p}</code>	<code>\lilyDynamics{p}</code>
<i>mp</i>	<code>\dinamica{mp}</code>	<code>\lilyDynamics{mp}</code>
<i>mf</i>	<code>\dinamica{mf}</code>	<code>\lilyDynamics{mf}</code>
<i>f</i>	<code>\dinamica{f}</code>	<code>\lilyDynamics{f}</code>
<i>ff</i>	<code>\dinamica{ff}</code>	<code>\lilyDynamics{ff}</code>
<i>fff</i>	<code>\dinamica{fff}</code>	<code>\lilyDynamics{fff}</code>
<i>sf</i>	<code>\dinamica{sf}</code>	<code>\lilyDynamics{sf}</code>
<i>sfz</i>	<code>\dinamica{sfz}</code>	<code>\lilyDynamics{sfz}</code>
<i>fp</i>	<code>\dinamica{fp}</code>	<code>\lilyDynamics{fp}</code>

Por lo general, cualquier combinación de letras relacionadas con las dinámicas estándar funcionan con este comando, como ser `\dinamica{ffffp}` para obtener ***ffffp***.

4.2 Duraciones

4.2.1 Figuras

Cuadro 2: Las figuras y los comandos que las producen.

Figura	Comando	Comando original (lilyglyphs)
♩	<code>\redonda</code>	<code>\wholeNote</code>
♩.	<code>\redondap</code>	<code>\wholeNoteDotted</code>
♪	<code>\blanca</code>	<code>\halfNote</code>
♪	<code>\blancab</code>	<code>\halfNoteDown</code>
♪.	<code>\blancap</code>	<code>\halfNoteDotted</code>
♪.	<code>\blancapb</code>	<code>\halfNoteDottedDown</code>
♪..	<code>\blancapp</code>	<code>\halfNoteDottedDouble</code>
♪..	<code>\blancappb</code>	<code>\halfNoteDottedDoubleDown</code>
♫	<code>\negra</code>	<code>\quarterNote</code>
♫	<code>\negrab</code>	<code>\quarterNoteDown</code>
♫.	<code>\negrap</code>	<code>\quarterNoteDotted</code>
♫.	<code>\negrapb</code>	<code>\quarterNoteDottedDown</code>
♫..	<code>\negrapp</code>	<code>\quarterNoteDottedDouble</code>
♫..	<code>\negrappb</code>	<code>\quarterNoteDottedDoubleDown</code>
♬	<code>\corchea</code>	<code>\eighthNote</code>
♬	<code>\corcheab</code>	<code>\eighthNoteDown</code>
♬.	<code>\corcheap</code>	<code>\eighthNoteDotted</code>
♬.	<code>\corcheapb</code>	<code>\eighthNoteDottedDown</code>

Cuadro 2: Las figuras y los comandos que las producen.

Figura	Comando	Comando original (lilyglyphs)
	<code>\corcheapp</code>	<code>\eighthNoteDotted</code>
	<code>\corcheappb</code>	<code>\eighthNoteDottedDown</code>
	<code>\semicorchea</code>	<code>\sixteenthNote</code>
	<code>\semicorcheab</code>	<code>\sixteenthNoteDown</code>
	<code>\semicorcheap</code>	<code>\sixteenthNoteDotted</code>
	<code>\semicorcheapb</code>	<code>\sixteenthNoteDottedDown</code>
	<code>\semicorcheapp</code>	<code>\sixteenthNoteDottedDouble</code>
	<code>\semicorcheappb</code>	<code>\sixteenthNoteDottedDoubleDown</code>
	<code>\fusa</code>	<code>\thirtysecondNote</code>
	<code>\fusab</code>	<code>\thirtysecondNoteDown</code>
	<code>\fusap</code>	<code>\thirtysecondNoteDotted</code>
	<code>\fusapb</code>	<code>\thirtysecondNoteDottedDown</code>
	<code>\fusapp</code>	<code>\thirtysecondNoteDottedDouble</code>
	<code>\fusappb</code>	<code>\thirtysecondNoteDottedDoubleDown</code>

4.2.2 Silencios

Si se desea obtener un símbolo de silencio con doble o triple puntillo, basta con adicionar `\otroPuntillo` tantas veces como se quiera a los comandos del Cuadro 3. Por ejemplo, si escribimos `\silencioBlancap\otroPuntillo`, obtendremos . y, de agregarse un `'otroPuntillo'` más se obtiene .

Cuadro 3: Símbolos de silencios.

Silencio	Comando	Comando original (lilyglyphs)
	<code>\silencioRedonda</code>	<code>\wholeNoteRest</code>
	<code>\silencioRedondap</code>	<code>\wholeNoteRestDotted</code>
	<code>\silencioBlanca</code>	<code>\halfaNoteRest</code>
	<code>\silencioBlancap</code>	<code>\halfNoteRestDotted</code>
	<code>\silencioNegra</code>	<code>\crotchetRest</code>
	<code>\silencioNegrap</code>	<code>\crotchetRestDotted</code>
	<code>\silencioCorchea</code>	<code>\quaverRest</code>
	<code>\silencioCorcheap</code>	<code>\quaverRestDotted</code>
	<code>\silencioSemicorchea</code>	<code>\semiquaverRest</code>
	<code>\silencioSemicorcheap</code>	<code>\semiquaverRestDotted</code>

4.2.3 Ritmo

Con el comando `\ritmo{}` podemos escribir ritmos usando la sintaxis lilypond. Por ejemplo, si escribimos `\ritmo{f4 f8 f f2}` obtenemos esto:

4.3 Alteraciones

Para escribir bemoles, becuadros y sostenidos en línea existe una manera provista por las tipografías que usan el estándar UTF-8, el cual intenta dar solución a las necesidades de

todos los idiomas del mundo a la vez de poner a disposición del usuario ciertos conjuntos de caracteres y símbolos usados en diferentes quehaceres humanos. Entre ellos, los tres símbolos básicos de alteraciones de notas. El bemol (\flat), por ejemplo se obtiene escribiendo `\fontspec{DejaVu Sans}\symbol{"266D}`. Sin embargo, con $\mu\text{S}\text{LuaT}\text{E}\text{X}$ se simplifica la sintaxis usando `\bemoltxt`. Sostenido (\sharp) y becuadro (\natural) se obtienen con `\sostenidotxt` y `\becuadro` respectivamente.

Sin embargo, la forma que recomendamos usar si se necesitan los doble bemoles, los dobles sostenidos, por ejemplo, y a los fines de mantener coherencia tipográfica, es la de los primeros cinco comandos del Cuadro 4, dejando los tres últimos solamente para los documentos en los que los dos símbolos menos usados estén ausentes.

Cuadro 4: Símbolos de alteraciones más frecuentes.

Alteración	Comando	Comando original (lilyglyphs)
\times	<code>\doblesostenido</code>	<code>\doublessharp</code>
\sharp	<code>\sostenido</code>	<code>\sharp</code>
\natural	<code>\becuadro</code>	<code>\natural</code>
\flat	<code>\bemol</code>	<code>\flat</code>
$\flat\flat$	<code>\doblebemol</code>	<code>\flatflat</code>
\flat	<code>\bemoltxt</code>	<code>\fontspec{DejaVu Sans}\symbol{"266D}</code>
\natural	<code>\bcuadrotxt</code>	<code>\fontspec{DejaVu Sans}\symbol{"266E}</code>
\sharp	<code>\sostenidotxt</code>	<code>\fontspec{DejaVu Sans}\symbol{"266F}</code>

4.4 Claves

Cuadro 5: Símbolos de claves en línea.

Clave	Comando	Comando original (lilyglyphs)
C	<code>\claveSol</code>	<code>\clefG</code>
F	<code>\claveDo</code>	<code>\clefC</code>
B	<code>\claveFa</code>	<code>\clefF</code>

4.5 Compases

Cuadro 6: Escritura de compases.

Compás	Comando	Comando original (lilyglyphs)
$\frac{7}{4}$	<code>\compas{7}{4}</code>	<code>\lilyTimeSignature{7}{4}</code>
$\frac{3}{8} + \frac{3}{8} + 2$	<code>\compas{3 + 3 + 2}{8}</code>	<code>\lilyTimeSignature{3 + 3 + 2}{8}</code>
$\frac{3}{4} \frac{6}{8}$	<code>\compas{3 6}{4 8}</code>	<code>\lilyTimeSignature{3 6}{4 8}</code>
C	<code>\compasC</code>	<code>\lilyTimeC</code>
C	<code>\compasCC</code>	<code>\lilyTimeCHalf</code>

4.6 Cuerdas

Para hacer referencia a las cuerdas de una guitarra se acostumbra utilizar números arábigos encerrados en círculos. El comando para producir tales caracteres no pertenece ni a `lilyglyphs`, ni a `lyluatex`, sino que es un comando de nuestra factura. Su sintaxis es `\cuerda{número}`. Por ejemplo, si escribimos `\cuerda{5}` obtendremos $\textcircled{5}$.

Cuadro 7: Cuerdas de la guitarra.

Cuerda	Comando
①	<code>\cuerda{1}</code>
②	<code>\cuerda{2}</code>
③	<code>\cuerda{3}</code>
④	<code>\cuerda{4}</code>
⑤	<code>\cuerda{5}</code>
⑥	<code>\cuerda{6}</code>

4.7 Cifrados

4.7.1 Cifrado americano

El cifrado americano, el cual consiste en la denominación de una fundamental comprendida entre las letra A y G y un sufijo que la califica, es producido por el comando `\cifAm{fundamental}{sufijo}`. Por ejemplo, si queremos obtener C^{\wedge} , escribimos `\cifAm{c}{maj7}`. Los sostenidos y bemoles se consiguen agregando 'is' y 'es' respectivamente a la fundamental (denominación propia de los Países Bajos); así, si escribimos 'cis', obtendremos C^{\sharp} , y si escribimos 'ces' el resultado será C^{\flat} .

A continuación, una tabla donde presentamos los tipos de acordes más usados:

Cuadro 8: Cifrado americano: tipos de acordes más usados.

Cifrado	Comando	Denominación
D	<code>\cifAm{d}{}</code>	Acorde de Re mayor.
Gm	<code>\cifAm{g}{m}</code>	Acorde de Sol menor.
F ⁷	<code>\cifAm{f}{7}</code>	Acorde de Fa séptima.
D ^{♯m7}	<code>\cifAm{dis}{m7}</code>	Acorde de Re [♯] menor séptima.
B ^{♭+}	<code>\cifAm{bes}{aug}</code>	Acorde de Si [♭] aumentado.
E ⁺	<code>\cifAm{e}{5+}</code>	Acorde de Mi quinta aumentada.
C ^{♯♭5}	<code>\cifAm{cis}{5-}</code>	Acorde de Do [♯] disminuido.
C ^{♯°}	<code>\cifAm{cis}{dim}</code>	Acorde de Do [♯] disminuido séptima.
A ^{sus4}	<code>\cifAm{a}{sus4}</code>	Acorde de La con cuarta suspendida.
C ^{7♭5}	<code>\cifAm{c}{5-7}</code>	Acorde de Do séptima sensible.
E ^{♭7/G}	<code>\cifAm{ees}{7/g}</code>	Acorde de Mi [♭] séptima com bajo en Sol.

4.7.2 Cifrados de grados y de funciones

\LaTeX viene provisto de un paquete llamado `harmony`. Éste fue diseñado para dar soporte al cifrado de funciones armónicas ideado por Hugo Riemann, combinándose a la perfección con `MusixTeX`, paquete que le otorga a \LaTeX la capacidad de la escritura musical en forma nativa pero que su aprendizaje conlleva una dificultad muy alta y los resultados no son tan buenos como para que valga la pena el esfuerzo. Sin embargo, `harmony` sí es útil a la hora de escribir cifrado en línea (no como análisis de una partitura). Por tal motivo, también hacemos uso de ese paquete, rebautizando el original comando `\HH` por el castellanizado `\acorde`. Para cifrar una partitura usaremos la librería para `lilypond` llamada `harmony``lily`, y desarrollaremos ese tema en la subsección 6.4 en la página 12.

El manual de **harmony** presenta de forma completa las posibilidades que otorga el paquete; en **mu^SLaTeX** hacemos un uso limitado a las situaciones más comunes, lo que no impide usar **harmony** del modo que su autor indica en la documentación del paquete. La sintaxis de `\acorde` es la siguiente:

`\acorde.[GRADO O FUNCIÓN].[INVERSIÓN].[NOTA AÑADIDA 1].[NOTA AÑADIDA 2].[NOTA AÑADIDA 3].`

de modo tal que podemos escribir tanto cifrado romano como funcional. Ejemplos:

El comando

`\acorde.\tachado{II}.6.7.\becuadro 5.\sostenido 2.`

genera el resultado



Como puede observarse, hemos incluido en **mu^SLaTeX** el comando `\tachado{}`, que es equivalente a usar el original **LaTeX** `\textst{}`, para dar soporte a la manera de Schoenberg de cifrar grados alterados; también es observable que los comandos `\sostenido` y `\becuadro` funcionan a la perfección, y como el lector sospechará a esta altura, el comando `\bemol` también se puede usar dentro de `\acorde`.

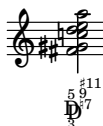
El mismo acorde, expresado al modo de Riemann se escribe del siguiente modo:

`\acorde.\DD.3.\sostenido11.9.\becuadro7.`

obteniéndose el siguiente resultado:




Finalmente, y a modo de anticipo, presentamos el uso de **harmony**lily, en contexto de análisis de partituras:




Hay, además de la indicación de bajo (3 por debajo de la función), una indicación de soprano (5 por encima de la función). Esta particularidad propia de **harmony**lily no está disponible en **harmony**.

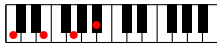

5 Diagramas

5.1 Diagramas de guitarra

Los diagramas de acordes para la guitarra se obtienen con el uso del comando `\diaGtr{ }{ }`. Idéntico uso al comando `\cifAm{ }{ }`, presentado en el apartado 4.7.1 en la página 7, `\diaGtr` necesita dos argumentos: fundamental del acorde y sufijo. En el sufijo puede incluirse la inversión del acorde. Por ejemplo, `\diaGtr{g}{m7}` da como resultado  ,

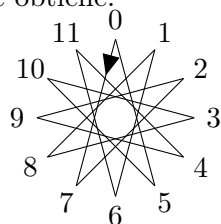
y `\diaGtr{g}{m7/b}`,  .

5.2 Diagramas de teclado

Con el comando `\teclado` se produce un teclado de dos octavas. Por ejemplo, si escribimos `\teclado{0,4,7,10}` obtendremos ; la opción `single` reduce el diagrama a una octava, y la opción `ext` adiciona a esa octava una tecla más. Así, `\teclado[single,ext]{0,4,7,10}` produce .

5.3 Diagramas seriales

El paquete `ddphonism` provee varias herramientas para generar diagramas relacionados con la producción de música serial. Quizás el más usado de los comandos de este paquete sea `\ddiagram`. Su uso básico es simple: por ejemplo, si escribimos `\ddiagram{0,7,2,9,4,11,6,1,8,3,10,5}` se obtiene:



Para más información, consultar la documentación del [paquete ddphonism](#).

6 Partituras y fragmentos

En esta sección nos centramos en la producción de fragmentos musicales y partituras completas. Los fragmentos musicales pueden ir tanto en línea (como parte del texto) como así también enmarcados en un entorno de figura. Es en estos tipos de situaciones en las que se hace uso del paquete `lilyuatex`. Éste nos da pleno acceso a las bondades del *software* de notación musical LilyPond. Por ser LilyPond el verdadero responsable de la escritura de los fragmentos musicales y partituras, inescusable se hace el estudio del manejo de este programa.

El comando `\ritmo{}`, presentado brevemente en el apartado 4.2.3, página 5, es ya un ejemplo de un uso solapado del comando `\lily{}` de `lilyuatex`, restringido a la escritura de signos rítmicos en la línea del texto. Sin embargo, el comando `\lily{}` tiene una potencialidad que supera en mucho a la escritura de ritmos en línea, y que a continuación iremos descubriendo.

6.1 Música antigua

LilyPond es quizás el programa de notación musical que más soporte da a la escritura de la música antigua: notación mensural, neogregoriano, bajo continuo, etc. Para mayor información, dirigirse al [manual de referencia de la notación](#) de LilyPond.

6.1.1 Canto gregoriano

A pesar de que LilyPond ofrece una solución para la escritura de la música gregoriana, ésta es claramente superada por la *performance* del proyecto Gegorio, por lo que recomendamos al usuario hacer uso de éste último a la hora de producir partituras de canto llano.

El paquete `muSLaTeX` llama, entre otros, al paquete `gregoriotex`, estilo especializado en la escritura de música gregoriana en `LuaLaTeX`. Así, pues, al llamar a `muSLaTeX`, el usuario tiene a su disposición *todo* lo ofrecido por tal paquete. Para satisfacer uno de los

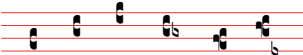
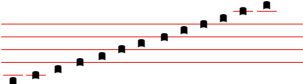



objetivos del presente trabajo, hemos incluido, como traducción del comando `\gabcsnippet` al comando `\gregoriano`. Un ejemplo:

```
\gregoriano{
  (c3)<c>P</c>a(e)ter(f) nos(g)ter,(g',)
  qui(f) es(h) in(g) cæ(f)lis:(e.;)
  sanc(f)ti(e)fi(f)cé(g)tur(fe) no(ef)men(g) tu(fg)um;(f.:)
}
```

genera los primeros versos del *Pater Noster*:



Cuadro 9: Notación en código **gabc** (Gregorio).

Tipo	Notación gabc (Gregorio)
Claves	 c2 c3 c4 cb3 f2 fb3
Punctum	 a b c d e f g h i j k l m
Virga	 ge if
Pes	 eg fi
Barras	 , ' : :: ;1 ;2 ;3 ;4 ;5 ;6

Para más información, referirse a la [página web](#) del proyecto Gregorio y al [manual del paquete](#). Para los interesados en la escritura de canto gregoriano, sin dudas esta es la opción más acabada a la fecha para tal fin. Quizás, a primera vista, el código generador de la escritura gregoriana nos parezca un tanto hermético, esotérico, pero al ver la referencia de la notación de la página oficial de Gregorio, los posibles miedos se disipan rápidamente.

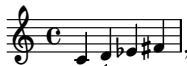
6.1.2 Bajo continuo en línea

Con el comando `\bajoContinuo{}` se puede escribir bajo cifrado en línea con el texto. Entre llaves se utiliza la sintaxis propia de `\figuremode` implementada en LilyPond. Por ejemplo, obtenemos 6/8 si escribimos `\bajoContinuo{<6+> <6 5/>}`. Para la inclusión de bajo continuo en fragmentos musicales y en partituras completas, el método es el indicado en los manuales de LilyPond, es decir un entorno 'FiguredBass' y en él incluir, dentro de un `figuremode{}` el bajo propiamente dicho.


6.2 Fragmentos musicales en línea

Como acabamos de mencionar, los fragmentos musicales escritos en la línea del texto son producidos por el comando `\lily{}` del paquete `lyluatex`. Éste fue traducido a dos comandos en $\mu\text{S}\text{LuaT}\text{E}\text{X}$, a saber: `\mus{}` y `\muses{}`; este último varía respecto al primero en que las notas se deben expresar con sus nombres en castellano (do, re, mi, etc.), mientras que en `\mus{}` ellas se escriben siguiendo la opción por omisión de `lilypond`, que es el lenguaje holandés, donde la convención es parecida a la alemana o norteamericana (c, d, e, etc.). Si escribimos

```
\muses{ \relative do' { do ( re ) mib fas}}
```

obtendremos , mientras que si escribimos


```
\mus{ \relative c' { c ( d ) ees fis }}
```

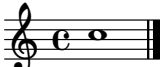
el resultado es exactamente el mismo: . La ventaja de usar `\muses` en vez de `\mus` es la de los nombres de las notas en castellano, y la desventaja es la de tipear más por ser estos nombres más largos que los holandeses; en fragmentos cortos parece algo irrelevante, pero en proyectos grandes tipear más, o menos sí es significativo.

Al igual que en `\lily{}`, tanto en `\mus{}` como en `\muses{}` se pueden pasar opciones entre corchetes antes que la música escrita entre llaves, siguiendo la sintaxis

```
\muses[<opción 1>, <opción 2>,...]{<música>}
```

Una de las opciones más útiles es `'staffsize'`, sirviendo ésta para definir el tamaño de la música. El valor por omisión es de 16, por lo que valores menores a esto producirán fragmentos visualmente más pequeños que los escritos más arriba, y números mayores a 16 generarán fragmentos más grandes. Ejemplos:

```
\muses[staffsize=12]{ \time 3/4 do'2. \bar "|" } 
```

```
\muses[staffsize=22]{ do''1 \bar "|" } 
```

6.3 \LaTeX como editor de partituras

Existen dos vías no excluyentes de incorporar partituras en un documento \LaTeX :

- Usando un entorno `\begin{ly} \end{ly}` y escribiendo en él código `lilypond`.
- Insertando un archivo `lilypond` externo (.ly) a través del comando `lilypondfile{}` o su traducción en $\mu\text{S}\text{LuaT}\text{E}\text{X}$ que es `\archivol{}`.

Ejemplo de escritura en el entorno `ly`:

```
\begin{ly}
\relative d' {
  \tempo "Presto"
  d\p\< a' e cis\f \bar "||"
}
\end{ly}
```

produce:



Si ese mismo código lilypond lo tenemos guardado en un archivo llamado 'ejemplo.ly', basta con escribir:

```
\archivoly{ejemplo}
```

para obtener:



La facilidad que tiene \LaTeX para producir índices, manipular diseños de páginas y establecer referencias cruzadas no lo tiene ni LilyPond ni ningún otro editor de partituras en la actualidad. Tanto LilyPond como Dorico permiten la realización de índices, pero de manera limitada. En cambio en \LaTeX , además de la generación automática de un índice a través de `\tableofcontents`, podemos hacer un índice musical con pequeños fragmentos musicales de las piezas que compongan, por ejemplo, un libro de composiciones, al modo de las antiguas ediciones Ricordi de las sonatas de Beethoven o del *El clave bien temperado* de Bach.

6.4 Particularidades de la escritura musical

6.4.1 Música y letra

Quizás sea LilyPond el editor de partituras que mejor distribuye espacialmente el texto en música vocal. Sin embargo, el método que utiliza en su sintaxis para informarlo es muy poco intuitiva: al anotarse la música en una sección y la letra en otra, el usuario no ve la relación entre nota y sílaba. Por este motivo hemos resuelto incluir esta subsección explicativa de la manera de escribir música vocal, sin pretender ni por cerca agotar el tema, tema que sí está dicho en profundidad en el [manual de LilyPond](#).

Mostraremos, pues, un pequeño ejemplo que dejará notar en qué partes del código van música y letra. Mientras la música va en un contexto *Voice*, la letra lo hace en un contexto *Lyrics*. A los fines prácticos a la hora de realizar breves ejemplos vocales, recomendamos —hay más de uno— el siguiente método:

```
\muses{<música>} \addlyrics{<letra>}}
```

Ejemplo:

```
\muses{
  \relative{ \time 3/4 re''4. la8 mi'4 dos ( re2 ) \bar "|." }
  \addlyrics{ ¡Ho -- la, mi~a -- mor! __ }
}
```



6.4.2 Cifrado de acordes

6.4.3 Bajo continuo

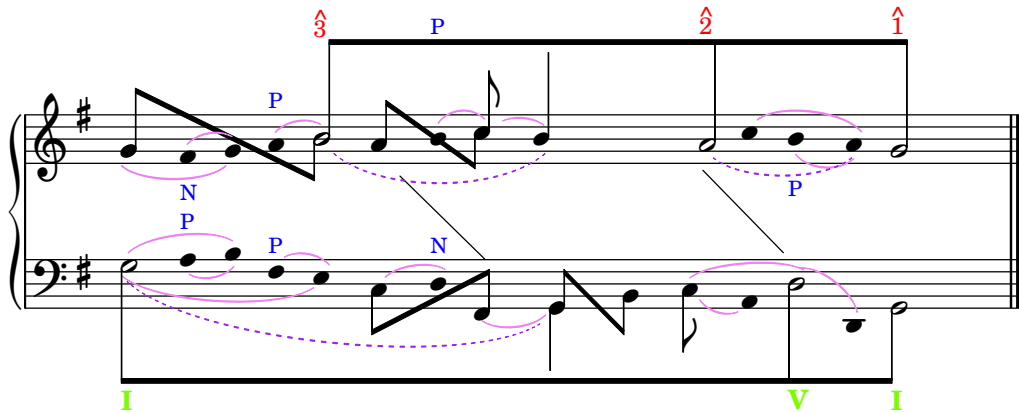
6.4.4 Análisis schenkeriano

Sabido es que en la notación analítica de Heinrich Schenker los grados melódicos se anotan con números arábigos con acentos circunflejos. Con `\musicalnotation` es posible usar el comando `\grado{}` para obtener un carácter de este tipo. Así, por ejemplo, escribiendo `\grado{5}` obtenemos $\hat{5}$.

Cuadro 10: Grados melódicos en notación schenkeriana.

Grado melódico	Comando
$\hat{1}$	<code>\grado{1}</code>
$\hat{2}$	<code>\grado{2}</code>
$\hat{3}$	<code>\grado{3}</code>
$\hat{4}$	<code>\grado{4}</code>
$\hat{5}$	<code>\grado{5}</code>
$\hat{6}$	<code>\grado{6}</code>
$\hat{7}$	<code>\grado{7}</code>
$\hat{8}$	<code>\grado{8}</code>

A continuación ofrecemos un ejemplo de notación schenkeriana:



El método para escribir con LilyPond gráficos con la notación analítica schenkeriana está descrito en el artículo [Make Stunning Schenker Graphs with GNU Lilypond](#).