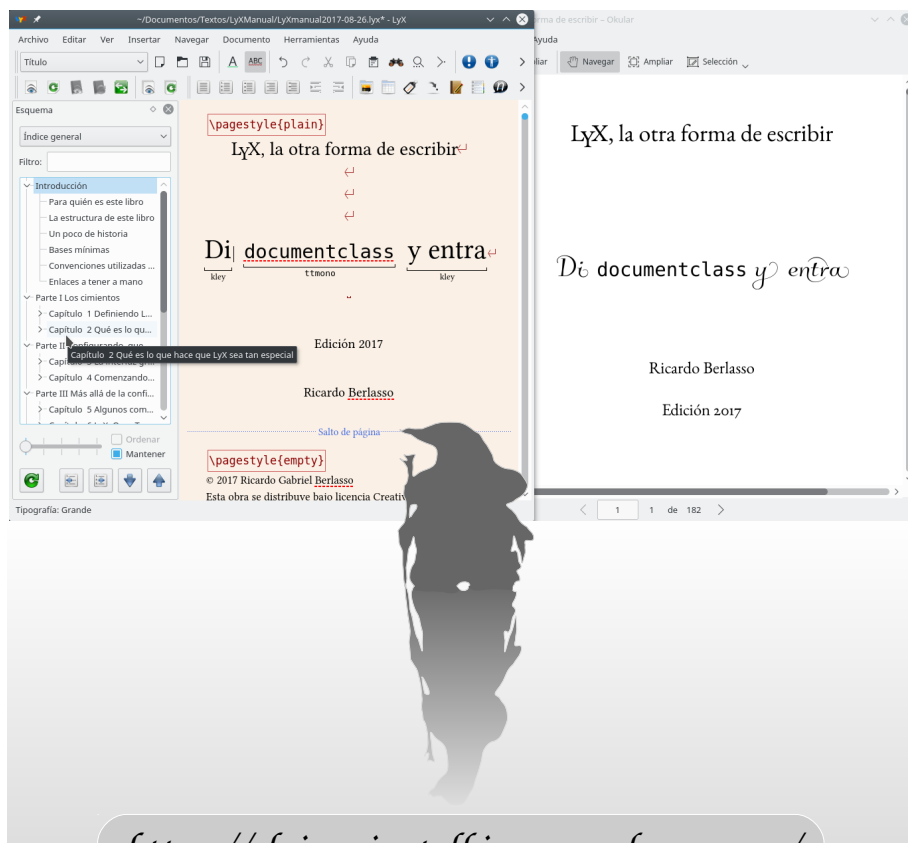


# LyX, la otra forma de escribir

Edición 2017

Una de las mejores herramientas para trabajar  
con el editor de documentos más potente

RICARDO G. BERLASSO



<https://elpinguinotolkiano.wordpress.com/>

L<sup>A</sup>T<sub>E</sub>X y derivados como X<sub>Y</sub>L<sub>T</sub>E<sub>X</sub> tienen la enorme ventaja de su potencia y el gran problema, al menos en su forma «pura», de su difícil «curva de aprendizaje».

El proyecto L<sup>A</sup>X resuelve la dificultad de utilizar L<sup>A</sup>T<sub>E</sub>X sin comprometer el poder del sistema, regalándonos un programa robusto y simple de utilizar que satisfará tanto a usuarios noveles como a veteranos.

En este libro se explora el poder de L<sup>A</sup>X, L<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sub>T</sub>E<sub>X</sub> y OpenType, mostrando hasta dónde se puede llegar.

Quizás más conocido por «Domando al escritor», el libro en el que se habla extensamente sobre LibreOffice Writer, el autor tiene muchos años de experiencia divulgando sobre procesadores de texto libres y tipografía, tanto desde su blog personal como desde varios foros y listas de correo.

LyX, la otra forma de escribir

*Di documentclass y entra*

Ricardo Berlasso

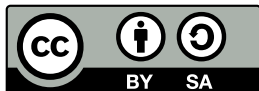
Edición 2017

© 2017 Ricardo Gabriel Berlasso

Esta obra se distribuye bajo licencia Creative Commons

Atribución-CompartirIgual 4.0 Internacional (CC BY-SA 4.0)

(<http://creativecommons.org/licenses/by-sa/4.0/>)



**ATRIBUCIÓN:** Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.



**COMPARTIR IGUAL:** Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted podrá distribuir su contribución siempre que utilice la misma licencia que la obra original.

**NO HAY RESTRICCIONES ADICIONALES:** Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

Cualquier marca mencionada en este libro pertenece a su respectivo propietario.

La última versión de este libro se encuentra siempre en la siguiente dirección:

<https://elpingunotolkiano.wordpress.com/latexxetexlyx/>

# Índice general

Introducción	1
Para quién es este libro	1
La estructura de este libro	4
Un poco de historia	4
Bases mínimas	6
Convenciones utilizadas en este libro	6
Enlaces a tener a mano	7
 I Los cimientos	 9
1 Definiendo $\text{\LaTeX}$	11
1.1 Qué es $\text{\LaTeX}$	11
1.2 $\text{\LaTeX}$ como lenguaje de etiquetas (o casi...)	13
1.3 Tipos de documentos	16
1.4 La «construcción» del documento	17
1.5 Y <i>OpenType</i> llegó a $\text{\LaTeX}$	17
 2 Qué es lo que hace que $\text{\LaTeX}$ sea tan especial	 19
2.1 El concepto «WYSIWYM»	21
2.2 Tipografías y «formatos de salida»	23
2.3 Características principales de $\text{\LaTeX}$	23
2.4 Cómo es trabajar con $\text{\LaTeX}$	25

2.5	LyX para los irreducibles de $\text{\LaTeX}$ . . . . .	26
2.6	Cuándo sí, cuándo no . . . . .	27
II	Configurando, que es un gerundio . . . . .	29
3	La interfaz gráfica de LyX . . . . .	31
3.1	Descripción general de la interfaz . . . . .	31
3.2	Los atajos de teclado . . . . .	34
3.3	Muchos documentos, en muchas formas . . . . .	35
4	Construir documentos . . . . .	37
4.1	Trabajando en LyX . . . . .	37
4.1.1	Formato general del documento . . . . .	37
4.1.2	Texto . . . . .	41
4.1.3	Listas . . . . .	42
4.1.4	Matemáticas . . . . .	43
4.1.5	Cuadros: ¡que no se llaman tablas! . . . . .	45
4.1.6	Marcos . . . . .	48
4.1.7	Figuras, cuadros y otros «flotantes» . . . . .	48
4.1.8	Referencias cruzadas . . . . .	50
4.1.9	Índices . . . . .	52
4.1.10	Generar PDF «dinámicos»: el paquete <code>Hyperref</code> . . . . .	52
4.1.11	Navegar y organizar el documento . . . . .	55
4.2	Apéndices . . . . .	58
4.3	Bibliografía . . . . .	58
4.4	Idioma . . . . .	59
4.4.1	Nuevo en 2.3: «comillas “anidadas” dinámicas» . . . . .	59
4.4.2	Diccionarios . . . . .	60
4.4.3	Babel vs. Poliglosia . . . . .	61
4.4.4	Pasar opciones a Babel en $\text{\LaTeX}$ /LyX . . . . .	62
4.5	Importar y exportar documentos . . . . .	64
4.6	Copiar información desde otros programas . . . . .	64
4.7	Control de versiones . . . . .	65
4.8	Trabajar con «ramas» . . . . .	66
4.9	Documentos maestros . . . . .	69

III	Más allá de la configuración básica	71
5	Algunos comandos	73
5.1	El preámbulo $\LaTeX$	73
5.2	La caja $\TeX$	74
5.3	Una cuestión de «tamaño»	75
5.4	Una cuestión de «forma» (tipográfica)	76
5.5	¡Números!	77
5.6	¡Reconfigurar!	78
6	$\LaTeX$ , OpenType y la tipografía	79
6.1	OpenType para la fuente predefinida	80
6.2	Definir una familia de fuentes tipográficas	82
6.3	OpenType para todo el documento	83
6.4	Distintas fuentes para distintas «formas»	84
6.5	Distintas opciones OpenType para cada «forma»	84
6.6	Aplicar un set estilístico a un carácter particular	86
6.7	OpenType para las fórmulas matemáticas 1: el paquete <code>unicode-math</code>	87
6.8	OpenType para las fórmulas matemáticas, 2: el paquete <code>mathspec</code>	89
6.9	Interlineado	91
6.10	Separación silábica	93
6.11	Microtipografía	94
6.11.1	Qué significa «microtipografía» y para qué sirve	94
6.11.2	Cómo aplicar esta «microtipografía»	96
6.12	Sub- y superíndices	97
7	Notas al pie de página y notas al final	101
7.1	Número de las notas sin superíndice	101
7.2	Notas al pie en columnas con texto principal sin columnas	102
7.3	Notas al pie sin quiebres de párrafo	102
7.4	Reiniciar la numeración de las notas en cada página	103
7.5	«Numerar» las notas con símbolos	103
7.6	Cambiar la fuente de las notas al pie	104
7.7	Notas al final	105

7.8	Una <i>yapa</i> : notas al margen . . . . .	106
8	Módulos, formato local y macros matemáticas . . . . .	107
8.1	Módulos . . . . .	107
8.2	Crear nuevas instrucciones . . . . .	108
8.3	Crear estilos de carácter en $\text{\LaTeX}$ . . . . .	110
8.4	Macros en modo matemático para $\text{\LaTeX}$ . . . . .	113
9	Listas numeradas . . . . .	117
9.1	Modificar listas particulares . . . . .	117
9.2	Modificar todas las listas del documento . . . . .	118
10	Sobre los índices . . . . .	121
10.1	Títulos cortos . . . . .	121
10.2	Índices parciales para cada capítulo . . . . .	121
10.3	Corregir espaciado en listas de figuras o cuadros . . . . .	122
10.4	Evitar las líneas de puntos en los índices . . . . .	123
10.5	Unas (pocas) palabras sobre los índices alfabéticos . . . . .	123
11	Títulos . . . . .	125
11.1	Renombrar los entornos «Parte» y «Capítulo» . . . . .	125
11.2	Numerar capítulos con números romanos . . . . .	126
11.3	Hacer que los capítulos comiencen en cualquier página . . . . .	126
11.4	Evitar los quiebres de página al comenzar un capítulo . . . . .	127
11.5	Cambiar el formato de los títulos en $\text{\LaTeX}$ . . . . .	127
11.5.1	Los paquetes a utilizar . . . . .	128
11.5.1.1	<code>xcolor</code> . . . . .	128
11.5.1.2	<code>titlesec</code> . . . . .	128
11.5.2	Un ejemplo completo . . . . .	129
11.6	Reiniciar la numeración luego de una «Parte» . . . . .	131
12	Cabeceras y pie de página, estilos de página . . . . .	133
12.1	Preparativos . . . . .	133
12.2	El paquete <code>fancyhdr</code> . . . . .	134
12.3	Un ejemplo . . . . .	136
12.4	Crear y utilizar estilos de página . . . . .	137



12.5	Primera página del capítulo . . . . .	141
12.6	Páginas en blanco . . . . .	141
12.7	Cambiar el estilo de página del entorno «Parte» . . . . .	143
13	Letras capitales y afines . . . . .	147
13.1	Algo tradicional . . . . .	147
13.2	¡Atención al número de líneas! . . . . .	149
13.3	Una variante: letra capital «alzada» . . . . .	150
13.4	Un desafío . . . . .	151
13.5	Una alternativa: primera línea de un párrafo en versalitas . . . . .	151
14	Figuras y otros flotantes . . . . .	153
14.1	Ajustar tamaño de las imágenes insertadas . . . . .	153
14.1.1	Ajustar el tamaño de la imagen manualmente . . . . .	153
14.1.2	Ajustar el tamaño automáticamente . . . . .	155
14.1.3	Tamaño de la imagen en la ventana de edición . . . . .	157
14.1.4	Por no repetir: grupos de imágenes . . . . .	157
14.2	Figuras «envueltas» . . . . .	157
14.3	Centrar el contenido de figuras y cuadros . . . . .	160
14.4	Configurar las leyendas de figuras y cuadros . . . . .	160
14.5	Renombrar flotantes . . . . .	161
14.6	Cambiar el tipo de numeración de figuras y cuadros . . . . .	162
15	Extras . . . . .	163
15.1	Describir teclas y menús: el paquete <code>menukeys</code> . . . . .	163
15.1.1	Qué es <code>menukeys</code> . . . . .	163
15.1.2	Usar <code>menukeys</code> en <code>L<sup>A</sup>T<sub>E</sub>X</code> . . . . .	165
15.2	Leyendas laterales para figuras/cuadros . . . . .	167
15.3	Cajas coloreadas . . . . .	169
15.4	Cosas de las que no he hablado . . . . .	170
15.4.1	Soporte <code>BIB<sub>Λ</sub>T<sub>E</sub>X</code> . . . . .	170
15.4.2	Listado de código con <code>listings</code> . . . . .	170
15.4.3	Formularios PDF . . . . .	171
15.4.4	<code>Sweave</code> y <code>knitr</code> . . . . .	171
15.4.5	Sistemas de álgebra computacional . . . . .	171
15.4.6	Y mucho, mucho más . . . . .	171

IV	Apéndices	173
A	Sobre fuentes tipográficas	175
A.1	Las fuentes utilizadas en este libro . . . . .	175
A.2	Más sobre fuentes tipográficas . . . . .	176
B	Formato local	177
C	Preámbulo $\LaTeX$	181
D	Otros formatos... y un saludo	185
D.1	Otros formatos utilizados en este libro . . . . .	185
D.2	Y sí, también formato directo . . . . .	186
D.3	Saludo final . . . . .	187

# Introducción

Hace ya muchos años existió una revista digital en español que hablaba de software libre llamada «Begins». Para esta revista escribí varios artículos, algunos sobre Writer, otros sobre LyX: de los primeros nació el libro «domando al escritor»<sup>1</sup>, de los segundos el curso básico de LyX en mi blog<sup>2</sup>, el cual dio lugar a más y más artículos que son ahora presentados, en forma actualizada, ordenada y por sobre todas las cosas *ampliada*, en este libro.

Muchos de los artículos de mi blog han quedado desactualizados, otros no eran del todo claros (y es que para muchos de ellos me tocó aprender mientras los escribía), varios se superponían y hasta se repetían en los temas tratados. En este libro intento resolver todos esos problemas, creando un contenido coherente, estructurado y fácil de seguir, ofreciendo además mucho material nuevo, no presente en el blog.

## Para quién es este libro

Decir que LyX es una «interfaz gráfica para L<sup>A</sup>T<sub>E</sub>X» no solo es impreciso: también resulta tramping.

LyX es un procesador de textos construido *sobre* L<sup>A</sup>T<sub>E</sub>X (y derivados, que en este libro hablaremos mucho sobre X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X) que ofrece a sus usuarios una variedad de herramientas gráficas y de texto que satisfarán incluso a los más exigentes.

---

1. Disponible en <https://elpinguinotolkiano.wordpress.com/domando-al-escriptor-2/>

2. El pinguino tolkiano: <https://elpinguinotolkiano.wordpress.com/>

*En principio* (que la vida siempre es complicada) resultaría posible el utilizar LyX sin conocer qué hay debajo, pero claramente esto sería una enorme limitación ya que, como veremos a lo largo del libro, este programa nos ofrece mucho más que una serie de menús y botones.

LyX puede ayudar tanto a los usuarios noveles que nunca han utilizado  $\text{\LaTeX}$  como a los expertos que no tienen problemas a la hora de «programar» sus documentos<sup>3</sup>: los usuarios noveles encontrarán que su ingreso en el mundo  $\text{\LaTeX}$  será más tranquilo que con otras alternativas mientras que los avanzados verán que resulta increíblemente sencillo el implementar sus viejos trucos en LyX.

Y aquí llegamos a un punto fundamental en este discurso: tarde o temprano algo de  $\text{\LaTeX}$  tendremos que aprender, nos guste o no. De hecho, muchas herramientas gráficas de LyX como la creación de estilos de carácter o de párrafo exigen un conocimiento más o menos importante de  $\text{\LaTeX}$ . Entonces, ¿vale la pena el esfuerzo?

Pues como todo en la vida, *depende*.

$\text{\LaTeX}$  es un sistema increíblemente poderoso, pero como dice el viejo cliché «un gran poder viene con una gran responsabilidad». Y en este caso la responsabilidad está en aprender a utilizar el sistema.

Si alguien nos proporciona el *documentclass* de base (la *plantilla* del documento) y solo nos toca completar el contenido  $\text{\LaTeX}$  resulta *muy* simple de utilizar: los comandos de base son sencillos y fáciles de comprender, la estructura del documento resulta lógica y clara, cuando nos acostumbramos a escribir expresiones matemáticas estas fluyen naturalmente, la numeración y las referencias cruzadas son simples de realizar y mantener, etcétera. Y todo esto se vuelve idílico en un programa como LyX donde no tenemos que preocuparnos de paréntesis olvidados y el código no molesta en la lectura de lo que estamos escribiendo.

Si nadie nos proporciona un *documentclass*, pero estamos conformes con uno de los predefinidos y no necesitamos cambiar nada  $\text{\LaTeX}$  sigue resultando bastante simple, sobre todo utilizando una interfaz gráfica que nos ayuda en muchos de los detalles «pesados» (definir la página y sus márgenes, las

---

3.  $\text{\TeX}$ , la base sobre la cual se construye  $\text{\LaTeX}$ , es un lenguaje de programación *Turing completo*: básicamente cualquier procedimiento que pueda ser escrito como un algoritmo puede ser resuelto con  $\text{\TeX}$ . Es decir, nadie dice que sea fácil, pero es *posible*.

fuentes tipográficas, idiomas, etcétera). Un usuario novel podría tener algunos inconvenientes a la hora de crear páginas especiales (dedicatoria, copyright), pero seguramente podrá resolverlo en pocos minutos preguntando a su buscador de internet preferido.

Si tenemos que ir más allá de la configuración básica... Pues todo dependerá de qué tan lejos tenemos que ir. Ciertamente LyX nos ayudará en nuestro camino, pero también es claro que no puede hacerlo todo por nosotros:  $\text{\LaTeX}$  es *demasiado* grande.

Para darnos una idea el preámbulo  $\text{\LaTeX}$  de este documento tiene unas de 120 líneas de código, el «formato local» (definición de estilos) tiene más de 90 líneas y en muchos lugares ha sido necesario recurrir a código en el documento mismo (las famosas «cajas  $\text{\TeX}$ » de las que hablaré repetidamente en este libro). También es necesario prestar mucha atención a los paquetes utilizados ya que algunos de ellos podrían entrar en conflicto y no funcionar apropiadamente: por ejemplo `titlesec`, un paquete para dar formato a los títulos, no se lleva bien con `tocloft`, un paquete para dar formato a los índices<sup>4</sup>.

En definitiva,  $\text{\LaTeX}$  es tan potente que se vuelve un universo en sí mismo: un universo por momentos contradictorio en el cual no siempre es sencillo el conocer sus reglas.

$\text{\LaTeX}$  nos permite realizar todo lo que queramos en términos de formato. No solo la facilidad de escribir expresiones matemáticas lo vuelve imprescindible en la escritura de documentos técnicos, su versatilidad nos permite buscar no solo la belleza del contenido sino también la de su presentación: opciones tipográficas avanzadas, simplicidad a la hora de manejar figuras y cuadros<sup>5</sup> (sí, eso que tantos dolores de cabeza causa en Writer o en Word resulta increíblemente sencillo en  $\text{\LaTeX}$ ), todo está a nuestro alcance.

Por lo tanto la pregunta de si vale la pena el aprender  $\text{\LaTeX}$  no es correcta, deberíamos preguntarnos si nos interesa alcanzar los resultados que  $\text{\LaTeX}$  puede ofrecernos y si estamos dispuestos a realizar el esfuerzo que ese aprendizaje exige.

---

4. Esto seguramente está relacionado con el hecho de que el paquete `titlesec` viene siempre junto a otro paquete llamado `titletoc` que sirve para configurar los índices. El problema está en que `titletoc` no es ni remotamente tan potente como `tocloft`.

5. Algunos los llaman «tablas», pero no por eso dejan de ser «cuadros».

Y eso es algo que solo tu, lector, puedes responder.

Te invito por lo tanto a acompañarme en esta aventura tipográfica, así quizás se aclaren tus ideas sobre estos temas. No temas experimentar: siendo LyX software libre solo tienes que instalarlo y ver si te atrapa tanto como lo ha hecho con migo.

## La estructura de este libro

El libro está separado en tres partes más algunos apéndices. Iremos desde lo más simple a lo más complejo comenzando con una introducción superficial a qué es  $\text{\LaTeX}$ , siguiendo con una descripción de la interfaz de LyX y de cómo se trabaja en este programa y terminando por temas avanzados de construcción del documento.

Como ya comentamos, algunas de las herramientas gráficas de LyX requieren conocer algo de código  $\text{\LaTeX}$  antes de poder utilizarlas por lo que su explicación quedará relegada a la parte «avanzada». Un ejemplo de esto es la creación de estilos de carácter: solo luego de saber cómo aplicar las distintas opciones tipográficas «a mano», de ser capaces de crear nuevas instrucciones y comandos  $\text{\LaTeX}$ , de perderle el miedo al «preámbulo», solo entonces podremos ir al «formato local» para crear nuestros estilos los cuales serán posteriormente accesibles con un simple clic derecho.

Esto hará que durante el desarrollo del texto algunos temas parezcan ser más difíciles de lo que realmente son, pero no hay alternativa: antes de poder correr tenemos que aprender a caminar sin tropezarnos.

## Un poco de historia

Siguiendo con la broma de la cubierta de este libro podríamos escribir  
*Las puertas de LyX, señor de los textos. Di, documentclass, y entra.*  
*Matthias Ettrich me ha creado, la comunidad me ha hecho crecer.*

Pero para llegar aquí muchas cosas han sido necesarias. Demos un exageradamente rápido y absurdamente simplificado repaso de las principales.

1977

Cuando Donal Knuth recibió la prueba de impresión de la segunda edición de su libro, *The Art of Computer Programming*,

la juzgó horrible. En esos tiempos él ya había comenzado a interesarse en los sistemas tipográficos digitales por lo que la pobre presentación de su libro lo empujó a crear su propio sistema tipográfico, T<sub>E</sub>X. Pasarían muchos años antes de que el sistema estuviera listo para uso general (más de los que él previó en un principio), pero todo comenzó en aquel momento.

- 1985      L<sup>A</sup>T<sub>E</sub>X fue creado por Leslie Lamport como una forma de simplificar el uso de T<sub>E</sub>X a través de una serie de *macros* que realizaran tareas específicas como crear notas al pie, títulos, etcétera, además de dar herramientas para compilar los documentos.
- 1995      Matthias Ettrich comenzó a desarrollar una interfaz gráfica para L<sup>A</sup>T<sub>E</sub>X llamada Lyrix. Poco después el proyecto cambió nombre para evitar conflictos con un ahora desaparecido procesador de textos comercial. El nombre finalmente elegido, L<sub>Y</sub>X, proviene de la extensión de los archivos utilizados por Lyrix. Al año siguiente, Matthis Ettrich crearía el proyecto por el cual es más conocido: KDE.
- 2004      Se publica X<sub>Y</sub>T<sub>E</sub>X, un sistema basado en L<sup>A</sup>T<sub>E</sub>X con soporte *unicode* y que puede utilizar tecnologías tipográficas modernas como OpenType, AAT y Graphite.
- 2011      A partir de la versión 2.0.0 L<sub>Y</sub>X ofrece soporte directo para X<sub>Y</sub>T<sub>E</sub>X, LuaT<sub>E</sub>X y XHTML.

Y el resto es historia.

Más información:

<https://en.wikipedia.org/wiki/TeX>

<https://en.wikipedia.org/wiki/LaTeX>

<https://en.wikipedia.org/wiki/LyX>

<https://en.wikipedia.org/wiki/XeTeX>

[https://en.wikipedia.org/wiki/Donald\\_Knuth](https://en.wikipedia.org/wiki/Donald_Knuth)

[https://en.wikipedia.org/wiki/Leslie\\_Lamport](https://en.wikipedia.org/wiki/Leslie_Lamport)

[https://en.wikipedia.org/wiki/Matthias\\_Ettrich](https://en.wikipedia.org/wiki/Matthias_Ettrich)

<http://www.lyx.org/misc/archaeology/>

[http://tolkiengateway.net/wiki/Doors\\_of\\_Durin](http://tolkiengateway.net/wiki/Doors_of_Durin)

## Bases mínimas

Daré por supuesto que el lector tiene *al menos* las mismas herramientas que he utilizado para escribir este libro. Estas son:

- L<sup>A</sup>T<sub>E</sub>X 2.2.3 o posterior mínimo, 2.3 o posterior recomendado
- Una distribución T<sub>E</sub>XLive 2015 o posterior
- (Opcional) Pandoc 1.17 o posterior

También daré por supuesto que el lector comprende, o al menos tiene a quién preguntarle, cómo se instalan los paquetes de esa distribución T<sub>E</sub>X. Esto implica claramente que doy por supuesto que el lector sabe utilizar su computadora.

Y lo más importante, daré por cierta la voluntad del lector de experimentar y de profundizar por su cuenta los temas aquí tratados. En este libro no encontrarás detalladas guías sobre cómo hacer esto o aquello, o ejercicios para acostumbrarte a utilizar el programa: para eso ya existe la excelente documentación incluida en L<sup>A</sup>T<sub>E</sub>X. Por esto y si bien muchos temas serán tratados en forma más o menos exhaustiva muchos otros serán solo *presentados*, dejando como tarea para el lector curioso el familiarizarse con la documentación y el tratar de ver más allá de las opciones predefinidas. ¡La mejor forma de *aprender es hacer*!

## Convenciones utilizadas en este libro

Las rutas a los distintos menús serán indicadas mediante un formato especial<sup>6</sup>. Por ejemplo, la ruta para llegar a la edición del «preámbulo T<sub>E</sub>X» (menú «Documento», luego a «Configuración» y por último «Preámbulo L<sup>A</sup>T<sub>E</sub>X») será indicada así:

Documento » Configuración » Preámbulo L<sup>A</sup>T<sub>E</sub>X

Los atajos de teclado también tendrán un formato especial<sup>7</sup>. Por ejemplo la combinación de teclas «control» y «E» se presentará como

Ctrl E

6. Cortesía del paquete menukeys.

7. También cortesía del paquete menukeys. Volveremos sobre esto en el apartado 15.1.2.



mientras que la secuencia «alt» con «P», luego «2» se mostrará en forma similar a lo siguiente:

Alt P, luego 2.

Soy consciente de que el sistema operativo de las «mac» requiere adaptar estos atajos, pero no tengo acceso a ese sistema por lo que no puedo proporcionar las equivalencias.

Para indicar código o nombres de paquetes se utilizará una fuente de ancho fijo:

```
\usepackage{emptypage}  
titlesec
```

Para indicar direcciones, ya sean estas en internet o en el sistema de archivos (mayormente para indicar la localización de la documentación de un paquete particular) se utilizará una fuente palo seco (sans serif):

<https://www.lyx.org/>  
`/usr/share/texmf/doc/latex/fancyhdr/`

Sobre esto último: escribo desde un sistema Linux por lo que las direcciones que daré serán las de mi sistema. Si el lector utiliza otro sistema operativo o algo diferente de T<sub>E</sub>XLive deberá adaptar las direcciones, pero no debería resultar difícil el hacerlo.

Para quienes tengan la versión PDF de este documento, los enlaces a internet estarán en color azul mientras que los enlaces internos al documento (índice, referencias cruzadas, etcétera) se mostrarán en color cian. Todos los enlaces son «activos» en el PDF: un clic en ellos llevará al lector al lugar enlazado, ya sea abriendo el navegador de internet en la página indicada o desplazándose dentro del documento.

## Enlaces a tener a mano

La página del proyecto L<sub>Y</sub>X se encuentra (en inglés) en

<https://www.lyx.org/>

Su wiki (también en inglés) en

<https://wiki.lyx.org/>

Para obtener soporte sobre L<sub>Y</sub>X se tiene la lista de correo en español. Para suscribirse es suficiente enviar un correo electrónico a

[lyx-es-subscribe@lists.lyx.org](mailto:lyx-es-subscribe@lists.lyx.org)

Luego de confirmar la suscripción será suficiente escribir a

[lyx-es@lists.lyx.org](mailto:lyx-es@lists.lyx.org)

Es importante notar que la lista de correo está configurada para «responder al remitente» por lo que al mantener una conversación allí no olviden agregar la lista como destinatario. Ni les cuento las veces que lo he olvidado.

El movimiento en la lista es más bien escaso, pero cuando alguien tiene un problema siempre encuentra respuesta por lo que aconsejo a todos los usuarios hispanos de LyX que se suscriban. Eso sí, recuerden que es una lista pública por lo que es conveniente utilizar una dirección de correo dedicada a comunicarse con listas y no la dirección privada.

El sistema para reportar errores del programa se encuentra en

<https://www.lyx.org/trac/wiki/BugTrackerHome>

A medida que avancemos en este libro necesitaremos cada vez más información sobre L<sup>A</sup>T<sub>E</sub>X y el lugar apropiado para obtener la documentación de cualquier paquete (además de nuestra instalación local) es el repositorio CTAN (*Comprehensive T<sub>E</sub>X Archive Network*):

<http://ctan.org/>

Para aquellos que quieran profundizar en el sistema que hace que LyX funcione existe un «curso no convencional de L<sup>A</sup>T<sub>E</sub>X» en español en el siguiente sitio:

<https://ondiz.github.io/cursoLatex/>

o bien, en «formato blog», aquí:

<https://ondahostil.wordpress.com/curso-no-convencional-de-latex/>

Y por supuesto, el lugar donde todas las respuestas sobre L<sup>A</sup>T<sub>E</sub>X pueden encontrarse (en inglés) es *T<sub>E</sub>X-L<sup>A</sup>T<sub>E</sub>X Stack Exchange*:

<https://tex.stackexchange.com/>

Si nadie allí sabe resolverlo, es imposible. Y como casi nada es imposible en L<sup>A</sup>T<sub>E</sub>X, allí estará la respuesta.

Parte I

Los cimientos

*Donde se definen los conceptos generales sobre los que se basará todo lo que sigue*

*On the shoulders of giants*

## Capítulo 1

# Definiendo $\text{\LaTeX}$

Como ya dijimos en la introducción, en principio no es necesario conocer  $\text{\LaTeX}$  (o  $\text{\XeTeX}$ ) para utilizar  $\text{\LyX}$ : este último nos ofrece una agradable interfaz gráfica que simplifica el uso del primero. Pero también es cierto que conocer *algo*, que saber qué es y cómo funciona  $\text{\LaTeX}$  nos permitirá controlar la apariencia de nuestros documentos en un nivel de detalle inimaginable por procesadores de texto tradicionales.

Y es por esto que el libro comienza con una introducción superficial de  $\text{\LaTeX}$ , que tener en claro qué es lo que sucede «debajo del cofre» nos permitirá llegar más lejos.

En la tercera parte de este libro, a partir del capítulo 5, volveremos sobre estos temas, que para sumergirnos en la parte «avanzada» no nos quedará alternativa a escribir código. Pero por ahora nos bastará el recorrer en forma relajada y superficial algunos conceptos básicos.

Comencemos.

### 1.1 Qué es $\text{\LaTeX}$

La idea fundamental en  $\text{\LaTeX}$  y derivados como  $\text{\XeTeX}$  (que no siempre se logra, todo hay que decirlo) es separar el *contenido* del *formato*.

Para indicar el título de un capítulo decimos solamente «capítulo» (en lenguaje  $\text{\LaTeX}$ , se entiende) y no nos preocupamos de la tipografía, del tamaño, de la numeración... De esos detalles se encarga  $\text{\LaTeX}$  en forma auto-

mática. ¡Nosotros solo nos preocupamos por el título en sí mismo!

Y es que el formato se encuentra definido separadamente en un archivo especial llamado *documentclass* el cual viene «llamado» en nuestro documento, junto a otras opciones de configuración, en lo que se conoce como «preámbulo  $\LaTeX$ ».

En este sentido  $\LaTeX$  es como un lenguaje de programación interpretado en donde se dan instrucciones que el sistema luego usará para «construir» el documento. Estas instrucciones deben ser escritas en un archivo de texto normal, por lo que en principio no se precisa de un programa especial para escribir un documento  $\LaTeX$ , si bien es altamente recomendable el usar uno. Y para esto tenemos al protagonista de este libro:  $\text{LyX}$ .

Esta doble estructura, *archivo de contenido* + *archivo con definiciones de formato* es a la vez la mayor fuerza y el principal problema de  $\LaTeX$  y derivados.

La mayor fuerza porque es posible cambiar de un *documentclass* a otro sin mayores inconvenientes, porque tener los estilos separados del contenido da mayor «robustez» a nuestro documento, porque si alguien nos provee el *documentclass* nuestro único trabajo será escribir ya que la estructura del archivo con el contenido es estándar y fácil de comprender...

El mayor problema porque alguien nos tiene que dar ese famoso *documentclass* ya que es sumamente difícil el definir uno por nuestra cuenta, porque utilizar otras fuentes tipográficas que las predefinidas puede tener sus bemoles, porque tratar de desviarse aunque más no sea ligeramente de lo que ofrece el *documentclass* elegido puede resultar en mucho trabajo extra.

Afortunadamente cualquier instalación estándar de  $\LaTeX$  como  $\text{T}\text{E}\text{X}$  Live (para Linux<sup>1</sup>) o  $\text{M}\text{i}\text{K}\text{T}\text{E}\text{X}$  (para el «sistema de la ventana»<sup>2</sup>) traen empaquetadas una infinidad de *documentclass* que cubren prácticamente todos los gustos y con un poco de trabajo siempre podemos personalizar todo.

El problema está en que  $\LaTeX$  está más cerca de un lenguaje de programación que de un sistema de redacción de textos tradicional: el «usuario común» (es decir, el no académico que no necesita escribir complejas fórmulas matemáticas) suele intimidarse con su estructura ya que si bien las

---

1. O GNU/Linux, si el lector prefiere. O GNU/BSD/MIT/Apache/MPD/etcétera/Linux, para ser más precisos. Yo uso openSUSE.

2. Que no quiero infringir cuestiones de copyright.

instrucciones L<sup>A</sup>T<sub>E</sub>X son simples e intuitivas siguen siendo *instrucciones*...

Pero a no preocuparse que como veremos este problema no es (al menos no completamente) real.

## 1.2 L<sup>A</sup>T<sub>E</sub>X como lenguaje de etiquetas (o casi...)

El lenguaje de programación por etiquetas más famoso es quizás el *html* por lo que si el lector conoce algo al respecto podrá hacerse rápidamente una idea sobre qué estamos hablando: cada elemento de un documento L<sup>A</sup>T<sub>E</sub>X viene «envuelto» en etiquetas que identifican las propiedades que queremos asignarle.

Por ejemplo, si queremos escribir una famosa expresión matemática como puede ser<sup>3</sup>  $E^2 = (mc^2)^2 + (pc)^2$ , para hacerlo «en línea» (es decir, incluido en un párrafo) en L<sup>A</sup>T<sub>E</sub>X escribimos

```
$E^2=\left(mc^2\right)^2+
\left(pc\right)^2$
```

mientras que si queremos que la expresión resalte en su propio párrafo, con un número que cambiará automáticamente al incluir otras ecuaciones

$$E^2 = (mc^2)^2 + (pc)^2 \quad (1.1)$$

entonces escribimos

```
\begin{equation}
E^2=\left(mc^2\right)^2+
\left(pc\right)^2
\end{equation}
```

Por otra parte, si el número no nos interesa podemos utilizar

```
\[
E^2=\left(mc^2\right)^2+
\left(pc\right)^2
\]
```

---

3. No, no está equivocada: *la otra* está equivocada (o mejor dicho, incompleta)

También en L<sup>A</sup>T<sub>E</sub>X se utilizan instrucciones más semejantes a «funciones» (de allí lo de «casi»). Por ejemplo, para indicar el título de una sección se escribe

```
\section{Esto es un título de sección}
```

Muchas instrucciones pueden anidarse, por lo que si queremos insertar una expresión matemática en el título de una sección simplemente escribimos

```
\section{La energía relativista:
$E^2=\left(mc^2\right)^2+
\left(pc\right)^2$}
```

También podemos «darle un nombre» a la ecuación que hemos escrito antes para después citarla con una «referencia cruzada». Para esto se usa la instrucción

```
\label{nombre}
```

Con todo esto un típico documento L<sup>A</sup>T<sub>E</sub>X «puro y duro» sería parecido a lo siguiente (más algunas otras cosas que definen el tipo de documento y de las cuales hablaré más adelante):

```
\begin{document}
\section{La energía relativista:
$E^2=\left(mc^2\right)^2+
\left(pc\right)^2$}
```

La famosa relación entre energía, masa y el valor de la velocidad de la luz elevado al cuadrado,  $E=mc^2$  es en realidad un caso particular de una expresión más importante, la cual presentamos, numerada y en toda su gloria, aquí:

```
\begin{equation}
E^2=\left(mc^2\right)^2+
```



```
\left(pc\right)^{2}
\label{eq:Einstein}
\end{equation}
La ecuación \ref{eq:Einstein} trajo no
pocos dolores de cabeza a los padres de
la mecánica cuántica.
\end{document}
```

Insisto en que con L<sup>A</sup>T<sub>E</sub>X no será necesario el escribir todo esto, solo estamos mostrando cómo funcionan las cosas «debajo del cofre».

Como puede comprobar el lector, en el citado código se ha hecho una «referencia cruzada» (como esta: ver ecuación [1.1](#)) a la ecuación numerada usando la instrucción

```
\ref{nombre}
```

Es importante notar que tanto `\label` como `\ref` sirven para mucho más que «llamar» ecuaciones. Estas instrucciones son «inteligentes» y si colocamos `\label` junto al título de una sección (o junto a una figura o un cuadro), `\ref` comprenderá dónde se encuentra la referencia a la cual está llamando y la mostrará en el modo correcto.

De este simple (y claramente incompleto) ejemplo resulta claro que durante la escritura de un documento L<sup>A</sup>T<sub>E</sub>X nos ocupamos solamente del contenido, no de su forma: incluso la numeración de las secciones y de las ecuaciones (y de las figuras y de los cuadros) se realiza en forma automática.

Otra cosa que en L<sup>A</sup>T<sub>E</sub>X se realiza en forma automática y que es una verdadera bendición en documentos muy complejos es la ubicación de figuras y cuadros: no solo cambiarán de página según sea necesario, sino que además el texto en torno a ellas se moverá automáticamente en forma tal de no dejar «huecos» en la página.

Sí, leyeron bien, sin mayor esfuerzo por parte del escritor<sup>4</sup> desaparecen en L<sup>A</sup>T<sub>E</sub>X las eternas luchas con los espacios en blanco que quedan cuando una figura pasa a la página siguiente ya que el sistema, usando criterios que pueden personalizarse fácilmente, se encarga de que todo quede perfecto.

---

4. Más adelante veremos los límites de todo esto, que en la vida nada es perfecto, pero no hay que preocuparse que son fáciles de superar.

### 1.3 Tipos de documentos

Lo comentado en la entrada anterior, me dirá el lector, de pensar más en el contenido de un documento que en su forma está muy bien, pero tarde o temprano tendré que preocuparme también sobre cómo se verá lo que estoy escribiendo. ¿Cómo decido qué tipo de letra usar, sus tamaños y todas esas cosas que siempre son necesarias?

Como ya hemos comentado,  $\text{\LaTeX}$  toma la definición de cada estilo en particular de un *documentclass*, o «clase de documento», el cual no es más que un archivo especial en el que se definen todas las cuestiones de formato. Por ejemplo, qué tamaño y característica de letra debe utilizarse cuando la instrucción `\section` se presenta está determinado por la *documentclass*.

Para elegir un determinado tipo de documento como por ejemplo «article» (artículo) y el idioma en el que queremos escribir, por ejemplo español, debemos escribir al principio (antes de `\begin{document}`):

```
\documentclass[spanish]{article}
```

Para crear un *documentclass* de la nada se necesitan conocimientos avanzados de  $\text{\LaTeX}$ , pero afortunadamente por lo general no es necesario el hacerlo: toda distribución  $\text{\LaTeX}$  moderna viene con un enorme número de *documentclass* predefinidos y en el repositorio de CTAN podemos encontrar muchos más. Esto, sumado a la posibilidad de realizar «ajustes finos» en las características del documento cambiando sus valores por defecto nos permitirá darle a nuestros documentos el formato que queramos casi sin problemas.

Por ejemplo, si en lugar de utilizar un tamaño de papel A4 que es el formato por defecto del *documentclass* «article» nosotros queremos usar A5, la instrucción de arriba cambia a:

```
\documentclass[a5paper,spanish]{article}
```

En esta parte del documento anterior al `\begin{document}`, parte que viene llamada *preámbulo*  $\text{\LaTeX}$ , pueden definirse muchas cosas. A partir del capítulo 5 haremos uso intensivo del mismo, por lo que ahora dejaré descansar al lector de estos códigos. Pero solo por un momento.

## 1.4 La «construcción» del documento

Dado que la idea de este libro es introducir una herramienta gráfica para manejar L<sup>A</sup>T<sub>E</sub>X no entraré en detalles técnicos sobre cómo se construye el documento ya que L<sup>A</sup>X lo hará por nosotros: solo delinearé las ideas básicas.

Nuestro documento es hasta ahora sólo una sucesión de instrucciones L<sup>A</sup>T<sub>E</sub>X que deben ser «compiladas» (generalmente, más de una vez) para crear el resultado final. Esta compilación se realiza llamando el programa «`latex`» u otros asociados (`pdflatex`, `xelatex`, etcétera) con una serie de parámetros entre los que se encuentra, claramente, el nombre del archivo por nosotros escrito. L<sup>A</sup>T<sub>E</sub>X genera entonces un archivo `.dvi` o `.pdf` que puede ser visto e impreso con Okular u otros programas semejantes y en donde todas las instrucciones del *documentclass* resultan implementadas. También es posible, utilizando el paquete `hyperref`, obtener un documento «clicable» en el que los enlaces, ya sean estos internos al documento o externos a otros documentos o sitios web, serán activos (hablaré de esto más adelante).

## 1.5 Y OpenType llegó a L<sup>A</sup>T<sub>E</sub>X

Algo de lo que no hemos hablado hasta ahora es de cómo seleccionar fuentes tipográficas. Esta omisión se debe a dos razones, siendo la primera que hacer esta selección desde L<sup>A</sup>X es infinitamente más simple que hacerlo manualmente.

La segunda razón es un tanto más «pesada»: L<sup>A</sup>T<sub>E</sub>X utiliza su propio sistema de fuentes tipográficas, el cual no solo es completamente incompatible con todo otro sistema sino que además es realmente complicado.

Las fuentes en L<sup>A</sup>T<sub>E</sub>X no se llaman por familia o tipo de fuente, sino por nombre de archivo<sup>5</sup>. Y para complicar aún más las cosas L<sup>A</sup>T<sub>E</sub>X en su forma «tradicional» no soporta *unicode*: esto último significa que para dar soporte a lenguajes diferentes del inglés se deben dar algunos rodeos. Incluso insertar una vocal acentuada en L<sup>A</sup>T<sub>E</sub>X «puro y duro» puede tener sus vueltas.

---

5. Antes de que los puristas L<sup>A</sup>T<sub>E</sub>X me protesten, sí, ya sé que existen *paquetes* que permiten seleccionar grupos de fuentes rápidamente. Pero esto en realidad refuerza lo que estoy diciendo: manejar fuentes *metafont* «a mano» no es tarea sencilla.

Existen variantes de L<sup>A</sup>T<sub>E</sub>X que permiten solucionar estos dos problemas, sobre todo el de la falta de soporte *unicode*. En este libro me centraré en el que es quizás el más maduro de todos los proyectos: X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X.

X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, inicialmente desarrollado por la gente de Sil internacional justamente para dar soporte a lenguajes no occidentales, es una variante de L<sup>A</sup>T<sub>E</sub>X que no solo soporta *unicode* sino que además nos permite utilizar fuentes del sistema. De hecho X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X es uno de los pocos sistemas libres en dar soporte completo a fuentes OpenType<sup>6</sup>, permitiendo utilizar todas sus características avanzadas como ser sustitución automática de caracteres (por ejemplo, ligaduras tipográficas), uso de verdaderas versalitas, verdaderos sub- y superíndices... todo lo que las fuentes provean.

Esto tiene enormes ventajas, entre ellas el no tener que hacer malabares para insertar un acento o un carácter especial.

X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X llama las fuentes en forma distinta de como lo hace L<sup>A</sup>T<sub>E</sub>X, pero como no hemos entrado en detalles de cómo lo hace este último tampoco hablaremos sobre cómo lo hace el primero.

Al menos no por el momento: volveremos sobre estos temas brevemente en el capítulo 4 y en gran profundidad (carga tus tanques de oxígeno) en el capítulo 6.

Pero bueno, ya está bien de «bases teóricas» y de esta historia de ver qué hay «debajo del cofre»: este libro es sobre L<sup>A</sup>T<sub>E</sub>X por lo que en el próximo capítulo comenzaremos a hablar de sus posibilidades.

Sírvete un poco más de café, estimado lector, que ya estamos en camino.

---

6. LibreOffice 5.3 ha logrado sorprenderme implementando soporte casi completo para esta tecnología. ¡Quién lo hubiera dicho! Ahora tengo que actualizar el *otro* manual.

## Capítulo 2

# Qué es lo que hace que LyX sea tan especial

Existe una larga guerra entre quienes creen que la mejor forma de crear documentos es con editores de texto como se hace con  $\text{\LaTeX}$  puro y quienes creen que nada supera una buena interfaz gráfica con sus menús desplegables y sus botones por todas partes como... bueno, el lector seguramente conoce *al menos* un producto de ese tipo.

Los defensores de la edición basada en texto dicen, y suelen tener razón, que cuando se conocen los códigos correctos es mucho más simple escribir que andar haciendo clic por aquí y allá y que en los editores gráficos se pasa uno más tiempo buscando la función que necesita que usándola.

Los defensores de los programas de edición gráficos dicen, y suelen tener razón, que recordar una infinidad de comandos por más simples que estos sean implica un esfuerzo la mayor parte de las veces injustificado y que la «curva de aprendizaje» de un programa gráfico es siempre más suave que en un programa basado en texto.

A esta guerra se agrega un tercer grupo formado por aquellos que dicen que los otros dos grupos son unos idiotas y que nada es mejor que un buen atajo de teclado, a lo cual los dos grupos anteriores responden, y suelen tener razón, que hacer todo con atajos de teclado requiere unas habilidades tanto en los dedos como en la memoria propias de un pianista virtuoso.

Pues bien, LyX ha resuelto este antiguo conflicto en la forma más elegante posible.

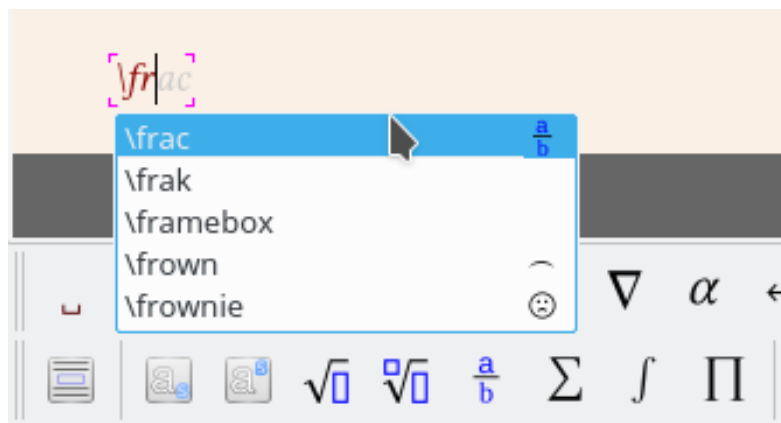


FIGURA 2.1: *El editor de ecuaciones en acción*

Por una parte esas acciones pesadas en  $\text{\LaTeX}$  que deben realizarse solo una vez por cada documento (la selección del tipo de documento, fuentes tipográficas, formato de página, etcétera) se manejan a través de menús.

Por otra parte aquellas acciones que solo se deben realizar una vez en documentos particulares (cambiar la numeración de las páginas del índice, insertar páginas especiales) se manejan a través de código.

Finalmente, y aquí reside la gran belleza del programa, *todo lo demás puede manejarse ya sea a través de código, menús/botones o atajos de teclado.*

Efectivamente:  $\text{\LaTeX}$  ofrece lo mejor de *todos* los mundos, dejando contentos a los usuarios más variados.

Consideremos por ejemplo la figura 2.1.

Aquí podemos ver el editor de ecuaciones de  $\text{\LaTeX}$ , pero el concepto es válido también para otras funciones. Como puede verse se tiene una barra de herramientas que permite insertar una fracción presionando un botón como en los editores gráficos, pero si en lugar de buscar botones o menús comenzamos a escribir la instrucción `\frac`  $\text{\LaTeX}$  no solo comprende lo que estamos escribiendo sino que además muestra un menú con las posibles instrucciones que pueden usarse con lo que se ha escrito hasta el momento: `\frac` para fracciones `\frak` para utilizar el tipo de letra *Fraktur* sobre un carácter o `\framebox` para insertar un rectángulo en torno al texto escrito.

Pero concentrémonos por un momento en `\frac`: si luego de terminar de escribir la instrucción presionamos la barra espaciadora la instrucción será reemplazada por una verdadera fracción con dos «campos de entrada» listos para recibir numerador y denominador. Ahora bien, si en lugar de escribir `\frac` (o de utilizar el botón correspondiente) presionamos Alt M, seguido de F obtendremos el mismo resultado.

Y así con todo: tanto `\sqrt` como el botón con la raíz cuadrada como la combinación Alt M, luego R nos dará la raíz cuadrada que estábamos buscando, con el campo libre listo para aceptar el argumento.

Como dijimos esta versatilidad no está limitada a las ecuaciones. Por ejemplo, en el texto normal podemos utilizar la caja de arriba a la izquierda para seleccionar el tipo de «entorno» Sección (o Capítulo, o párrafos especiales... los entornos disponibles dependerán del *documentclass* elegido) o utilizar el atajo Alt P, seguido de 2.

Así es, la interfaz de L<sup>A</sup>T<sub>E</sub>X nos permite trabajar en las tres modalidades: texto, menús/botones y atajos de teclado. Lo que el usuario prefiera.

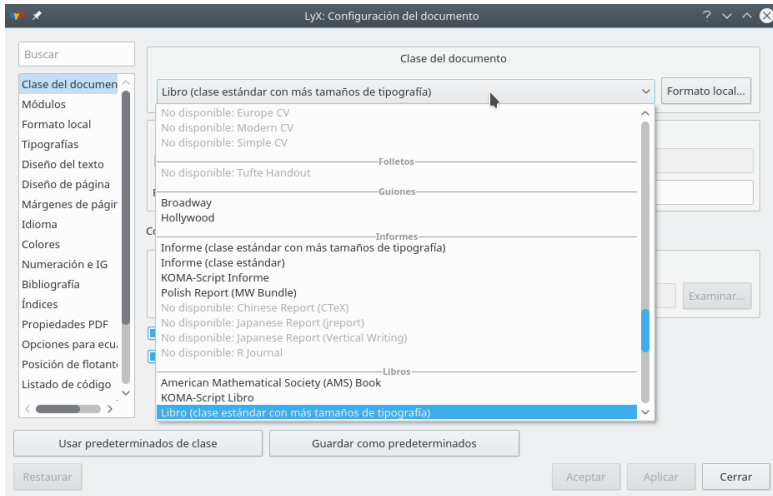
## 2.1 El concepto «WYSIWYM»

Lo que muestra L<sup>A</sup>T<sub>E</sub>X durante la edición del documento *no* es exactamente igual a lo que sale una vez que este es «compilado», pero nos da una idea exacta de lo que el documento *quiere decir*. Incluso L<sup>A</sup>T<sub>E</sub>X no muestra los quiebres de página ya que para trabajar con el *contenido* son completamente inútiles (pero sí, obviamente, los genera al compilar).

Otra cosa que *no* se ve en la ventana de edición son los códigos L<sup>A</sup>T<sub>E</sub>X: todos los `\documentclass`, `\begin{document}`, `\end{document}`... están «ocultos» y son manejados por L<sup>A</sup>T<sub>E</sub>X sin que el escritor tenga que intervenir directamente. Incluso los títulos de los capítulos y las secciones se ven como títulos verdaderos (si bien en un formato diferente del resultado final) y no con la instrucción `\section`.

Otras cosas a las que los nuevos usuarios de L<sup>A</sup>T<sub>E</sub>X tienen que acostumbrarse, pero que tienen su lógica:

- No hay reglas en la página.
- Si presionamos la barra espaciadora dos veces seguidas, la segunda

FIGURA 2.2: *Eligiendo el documentclass*

será ignorada. Lo mismo sucede con la tecla  $\leftarrow$ : los espacios son manejados automáticamente por los estilos, el autor no necesita preocuparse.

- La tecla  $\leftrightarrow$  solo sirve para moverse entre partes del documento (partes de una expresión matemática, celdas de una tabla...).
- Todo lo que sea espaciado, sangrado de párrafos, formato del texto, contenido de las cabeceras y los pie de página, posición de figuras y cuadros... todas estas cosas se manejan en LyX de forma automática y de acuerdo al contexto, basándose en el *documentclass* elegido en `Documento >> Configuración >> Clase de documento`.

Como se ve en la figura 2.2 tenemos muchas clases de documentos a nuestra disposición y siempre podremos conseguir más a través de nuestra distribución  $\text{\LaTeX}$ . También tenemos, en otros menús, muchas formas de «personalizar» nuestra elección cambiando por ejemplo la forma en la que se separan los párrafos (sangrado o espacio vertical), el tamaño de la página y sus márgenes, la ubicación de los «flotantes», la numeración, etcétera. Las opciones disponibles dependerán del *documentclass* elegido, pero son



siempre tan numerosas que satisfarán incluso al más empedernido de los detallistas.

Además, la forma en la que cada parte del documento se construye no está gobernada solamente desde allí ya que más allá de estas opciones generales es siempre posible realizar ajustes finos en partes determinadas del documento: en el menú **Editar** » ... podremos cambiar las características de un párrafo en particular, o de una palabra aislada, o de una página en especial, mientras que desde el menú **Insertar** » ... podremos usar saltos de página, espacios especiales entre palabras y párrafos, lo que sea necesario.

¡Ha! **Ctrl** **B** activa y desactiva la negrita mientras que **Ctrl** **E** activa y desactiva la itálica: al menos eso es (casi) igual que siempre.

## 2.2 Tipografías y «formatos de salida»

Entre las muchas opciones que el menú **Documento** » **Configuración** ofrece (Diseño de página, Márgenes, Idioma...) tenemos dos entradas que son de interés para este libro: *Tipografías* y *Salidas*.


Estas dos opciones nos permitirán configurar LyX para que utilice X<sub>Ǝ</sub>TeX (o LuaTeX, pero no estamos hablando de esa opción aquí) en la construcción de nuestros documentos.

Si seleccionamos **Tipografías** » **Usar tipografías no TeX (via XeTeX/LuaTeX)** podremos seleccionar las fuentes que tengamos instaladas en el sistema en lugar de las fuentes L<sup>A</sup>TeX, pero para que LyX no nos proteste al compilar el documento tenemos que también decirle que efectivamente utilice X<sub>Ǝ</sub>TeX para esa compilación: bajo **Salidas** » **Formato de salida predeterminado** es necesario seleccionar «PDF (XeTeX)».

## 2.3 Características principales de LyX


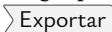

LyX está basado en las librerías Qt por lo puede funcionar en muchas plataformas diferentes.

LyX puede cargar varios documentos simultáneamente, abriéndolos en «pestañas» separadas al estilo firefox o bien en una modalidad de «ventana dividida» (volveremos sobre esto más adelante).

El programa recuerda en qué parte del documento estábamos trabajando al cerrarlo, por lo que al abrirlo nuevamente nos llevará allí. Cuenta también con un panel que nos muestra la estructura del documento, el cual puede ser activado desde  Panel del esquema. Este panel funciona como un «índice» del documento que nos permite no solo llegar rápidamente al punto que nos interesa sino también cambiar rápidamente el orden de los distintos apartados o modificar el «nivel» del título seleccionado («degradar» una sección a subsección, por ejemplo). La casilla de verificación «Mantener» es interesante para evitar que el «árbol» de niveles se cierre cuando hacemos doble clic en un nivel superior. Volveremos sobre todo esto en la sección [4.1.11](#).




Se cuenta además con un sistema de registro de versiones, el cual permite registrar, aceptar o rechazar cambios facilitando así el trabajo cooperativo en un único documento (lo comentaremos brevemente en la sección [4.7](#)). Puede usar *hunspell* para la corrección ortográfica en tiempo real y posee un tesoro que puede utilizar los diccionarios de sinónimos de LibreOffice (sección [4.4.2](#)). Permite trabajar fácilmente con documentos escritos en varios idiomas. Nos ofrece una forma increíblemente simple y robusta de trabajar con documentos maestros (sección [4.9](#))...

Algo importante de remarcar es que LyX trabaja con *unicode* sin problemas, por lo que incluso en el caso que decidamos no utilizar X<sub>Y</sub>TeX no tendremos que preocuparnos a la hora de insertar símbolos o acentos: LyX se encargará de traducir todo para que L<sup>A</sup>T<sub>E</sub>X lo comprenda sin dificultad.

Debe tenerse en cuenta que las imágenes no son incorporadas en los archivos LyX, solo son vinculadas y los archivos originales quedan en su lugar. Por lo tanto si tenemos que trabajar en un mismo documento desde varios sistemas (todo estudiante de doctorado sabe que se trabaja tanto o incluso más en casa que en la oficina) nos tocará un poco de trabajo para mantener todo sincronizado. Para simplificar la tarea de mover nuestro proyecto a otro sistema es posible exportar un «paquete» comprimido como archivo `tar.gz` que contenga el documento LyX y todas las imágenes desde   Exportar  Archivo LyX (`tar.gz`): simplemente descomprimos el paquete en una carpeta vacía del sistema de destino y ya estamos listos para continuar trabajando.

## 2.4 Cómo es trabajar con LyX

Volveremos sobre todo esto en gran detalle a partir del próximo capítulo, pero veamos ahora rápidamente del «método de trabajo» necesario con LyX.














Comenté antes que la ubicación de las figuras y cuadros se realiza automáticamente. Para esto se debe primero insertar un «flotante» () apropiado y dentro de él colocar nuestra figura o cuadro junto con su título (en LyX se llama «leyenda»). Estos flotantes, como así también las notas al pie de página y otros elementos especiales se muestran en la ventana de edición de LyX como una «caja» roja con una etiqueta que explica de qué se trata. Estas cajas tienen la particularidad de poder «cerrarse» dejando solo la etiqueta cuando no necesitamos ver su contenido: esto puede hacerse en cada caja particular simplemente haciendo clic sobre su etiqueta o globalmente con  . Muchas de estas cajas ofrecen menús contextuales para configurar su comportamiento.

A este punto es importante notar que L<sup>A</sup>T<sub>E</sub>X puede trabajar mayormente solo con imágenes postscript encapsuladas (EPS), X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X o pdf<sub>L</sub>A<sub>T</sub>E<sub>X</sub> solo con imágenes PDF... pero a no preocuparse que LyX se sirve de distintas librerías para aceptar virtualmente cualquier tipo de imagen, convirtiéndola en el formato apropiado cuando sea necesario. En general, si se trata de diagramas será mejor utilizar imágenes EPS, PDF o SVG ya que al ser formatos vectoriales darán mejores resultados, pero cualquier tipo de imagen de mapa de bits (png, jpg... lo que sea) funcionará a la perfección.

Un concepto importante en LyX es el de *entorno*. Los entornos sirven para distinguir las diferentes partes del documento, como por ejemplo el título de un capítulo del texto simple que le sigue y se asignan con el menú que se encuentra a la izquierda de la primera barra de herramientas (o con el atajo de teclado correspondiente...), justo debajo del menú Archivo.



Los entornos que tengamos a nuestra disposición dependerán del *documentclass* elegido. Por ejemplo, al escribir un artículo o una carta no tendremos a nuestra disposición Capítulo, como sí lo tendremos al escribir un libro.



Ya hemos introducido los diversos atajos de teclado que nos ofrece LyX. Estos atajos son de dos tipos: atajos «directos» como las combinaciones

  o    para introducir una ecuación o bien «secuencias» de teclas como la ya comentada   seguido de  para introducir una fracción, o  , luego  seguido de una letra (a, b...) para introducir un carácter griego ( $\alpha$ ,  $\beta$ ...). Como hemos ya insinuado la asignación de los principales «entornos» puede realizarse desde el teclado a través de las secuencias   seguido de un número del 0 al 6 (o de «b», o de «e», ¡prueben qué se obtiene!). Es importante notar que alguno de estos atajos podrían no hacer nada dependiendo del *documentclass* elegido.




El último paso importante en el trabajo con LyX es ciertamente la «compilación» del archivo. Seguramente habrás visto la barra de herramientas que se encuentra justo debajo del menú «Entorno»: de izquierda a derecha tenemos «ver salida en el formato elegido (dvi o pdf)», «actualizar vista», «ver documento maestro»... los dos primeros botones serán seguramente los más utilizados, al menos para aquellos usuarios que se inician en LyX.

## 2.5 LyX para los irreducibles de $\text{\LaTeX}$

Si bien LyX utiliza su propio formato de archivo basado en XML para almacenar el trabajo a no preocuparse, fanáticos  $\text{\LaTeX}$ , que la conversión al formato  $\text{\LaTeX}$  es siempre perfecta y por una buena razón: LyX *siempre* convierte a formato  $\text{\LaTeX}$  cada vez que compila el documento por lo que es algo que *necesariamente* debe funcionar. Por lo tanto, si alguien por alguna razón necesita obtener un archivo en formato  $\text{\LaTeX}$  legible por seres humanos, pues es suficiente ir a  .

Como ya hemos comentado en LyX es posible introducir directamente código no solo en el modo matemático. Fuera del modo matemático contamos de hecho con el botón  $\text{\TeX}$  (atajo de teclado:  ) que nos abre una «caja» especial donde podemos escribir nuestro fantástico código. Por ejemplo, si usando alguno de los *documentclass* para libros (todas las variantes de *book* que nos da  $\text{\LaTeX}$ ) insertamos con nuestra caja roja el comando



`\frontmatter`




justo delante del índice (  ) y del mismo modo insertamos

`\mainmatter`

justo después lograremos que las páginas en las que se encuentra el índice sean numeradas con números romanos, mientras que las páginas siguientes tendrán números arábigos que comenzarán desde 1 en el primer capítulo.

Ya volveremos sobre estas (y otras) cosas más adelante en el libro.

L<sup>A</sup>T<sub>E</sub>X permite además ver el código L<sup>A</sup>T<sub>E</sub>X en tiempo real, ya sea del párrafo que estamos editando o de todo el documento, simplemente habilitando la opción   Panel de código fuente.

Por último, aunque no menos importante, L<sup>A</sup>T<sub>E</sub>X ofrece un modo realmente simple de editar el «preámbulo L<sup>A</sup>T<sub>E</sub>X», es decir (casi) todo el código L<sup>A</sup>T<sub>E</sub>X que se puede escribir entre las instrucciones `\documentclass` y `\begin{document}` y que afecta el comportamiento general del documento:  Documento  Configuración  Preámbulo L<sup>A</sup>T<sub>E</sub>X.

Las posibilidades son casi ilimitadas y en las próximas páginas daremos varios ejemplos.

## 2.6 Cuando sí, cuándo no

No existe la herramienta perfecta capaz de hacerlo todo y ciertamente no se utiliza un destornillador para clavar un clavo. L<sup>A</sup>T<sub>E</sub>X es poderoso para una enorme variedad de trabajos, pero podría resultar molesto en otros.

Si queremos escribir un libro, especialmente uno de carácter técnico con fórmulas matemáticas, gráficos, cuadros, etcétera, resulta perfecto (estoy tentado a decir *imprescindible*), pero si queremos realizar un póster o una presentación o diseñar una revista o cualquier otro documento que necesite un control manual de todos los elementos será mejor buscar otras opciones. Es decir, es *posible* realizar un póster o una presentación en L<sup>A</sup>T<sub>E</sub>X (L<sup>A</sup>T<sub>E</sub>X cuenta con más de un *documentclass* para presentaciones), pero no resulta difícil el encontrar argumentos para decir que no es la herramienta adecuada para el trabajo.

A pesar de ser uno de los más viejos sistemas electrónicos de edición de texto L<sup>A</sup>T<sub>E</sub>X y sus derivados siguen siendo por lejos los más poderosos.

Sus limitaciones son pocas mientras que sus beneficios son enormes, especialmente gracias a L<sup>A</sup>T<sub>E</sub>X que hace fácil las cosas molestas de L<sup>A</sup>T<sub>E</sub>X como la

selección del *documentclass*, la creación de figuras y cuadros y el ajuste de los «detalles» (márgenes, formato de párrafos) entre otras cosas.

Si bien algunos programas, especialmente Writer con su filosofía basada en estilos y su amplio soporte OpenType, se están acercando a las características de  $\text{\LaTeX}$  todavía tienen un largo camino por recorrer, especialmente cuando hablamos de documentos técnicos complejos.

El público más amplio de este sistema se encuentra ciertamente entre la gente de ciencia (físicos, matemáticos, astrónomos...) e ingeniería, pero no se limita solo a este especializado grupo de personas:  $\text{\LaTeX}$  o  $\text{\XeTeX}$  +  $\text{\LyX}$  está abierto a todo aquel que tenga algo para decir a través de un texto y que quiera obtener documentos bellos con (relativamente) poco esfuerzo.

Y aquí llegamos a un punto «filosófico»: *¿existe algún motivo que nos obligue a preocuparnos por la belleza de nuestros documentos?*

De hecho, si vamos a una librería y tomamos de un estante un libro cualquiera nos encontraremos casi seguramente con una presentación realmente pobre. Y no me refiero solamente a una mala encuadernación: incluso a esta altura del siglo y de la tecnología muchas de las editoriales populares siguen sin preocuparse tan siquiera de utilizar ligaduras tipográficas, ni hablemos entonces de alternativas contextuales o microtipografía. ¡Incluso he tenido en mis manos libros que presentaban un horrible interletraje!<sup>1</sup>

Cada persona tendrá su propia reacción ante este hecho tristemente indiscutible. La mía es *¿y qué?*

¿Existe algún motivo que me obligue a preocuparme por lo que los demás *no hacen*?

Ciertamente no. A mi me gusta que un texto esté bien presentado. Me encantan las ligaduras tipográficas. Adoro la microtipografía. ¿Por qué tendría yo que privarme de utilizar las potentes herramientas que  $\text{\LyX}$ ,  $\text{\XeTeX}$  y OpenType me ofrecen?

¿Por qué tendrías que privarte tú, estimado lector?

Si no quieres renunciar a vestir de belleza el contenido de tus documentos acompáñame, que este viaje recién comienza.

---

1. <https://es.wikipedia.org/wiki/Interletraje>  
<https://www.xkcd.com/1015/>

## Parte II

# Configurando, que es un gerundio

*Donde se comienza a trabajar sobre  $\text{LyX}$ , nivel básico a medio*

*Don't Panic!*



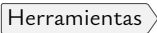


## Capítulo 3

# La interfaz gráfica de LyX

LyX presenta una típica interfaz de menús, barras de herramientas y paneles. Pero hay mucho más de lo que aparece a simple vista.

Veamos cómo está conformada.

### 3.1 Descripción general de la interfaz

Como dijimos al comienzo la interfaz gráfica de LyX es bastante tradicional en apariencia: una serie de menús arriba seguidos por un par de barras de herramientas. Con un clic derecho sobre las barras de herramientas podemos modificar el tamaño de los iconos y en el menú  **Herramientas**   **Preferencias** tenemos varias opciones para modificar la apariencia general del programa.

Pero por supuesto esto es solo el comienzo, tal y como se puede apreciar en la figura 3.1.

Un espacio de la interfaz de LyX que suele ser pasado por alto es la delgada barra que se encuentra justo debajo del área de edición. Aquí LyX presenta información importante de «lo que está pasando»:

- la función del botón de la barra de herramientas sobre la que se encuentra el puntero del ratón
- el formato del texto sobre el que se encuentra el cursor
- las opciones que tenemos para completar un atajo de teclado

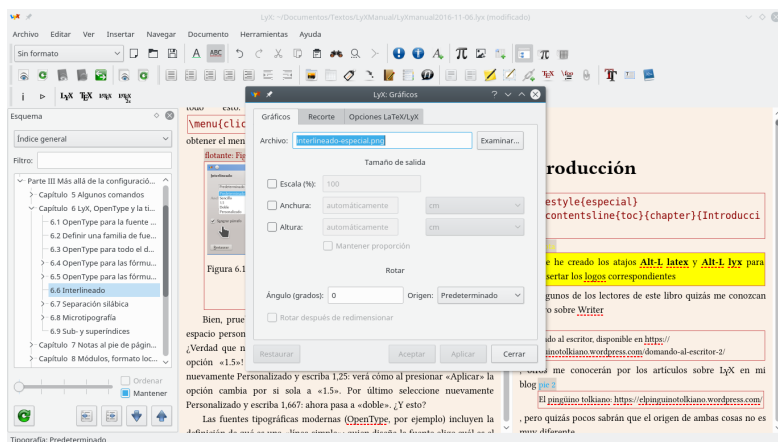


FIGURA 3.1: *LyX* y su interfaz «multidocumento» mostrando el panel de esquema y un cuadro de diálogo con la configuración de una imagen insertada

- el comando que se ha apenas utilizado desde algún menú
- notificaciones del compilador (advertencias, errores, etcétera)

Algunas de estas cosas pueden verse en la figura 3.2.

La combinación **Ctrl** **F** nos dará la herramienta de «buscar y reemplazar» básica, mientras que **Ctrl** **↑** **F** nos dará un panel con opciones de búsqueda avanzadas en el que podremos utilizar expresiones regulares, buscar expresiones matemáticas, etcétera.

**NOTA:** Quienes conozcan mi libro sobre *Writer* seguramente recordarán el extenso capítulo sobre la búsqueda avanzada con expresiones regulares por lo que podrían preguntarse ¿y dónde está el capítulo sobre expresiones regulares en *LyX*?

Pues bien... en ningún lado. Y es que no logro hacerlas funcionar a mi gusto. Algunas expresiones van mientras que otras no: por ejemplo, si bien es posible utilizar expresiones regulares en la caja de «reemplazar con» los resultados son... extraños, sobre todo cuando tratamos de hacer referencias. En fin, que pueden usarse, pero luego de que se me acalabrara el ce-

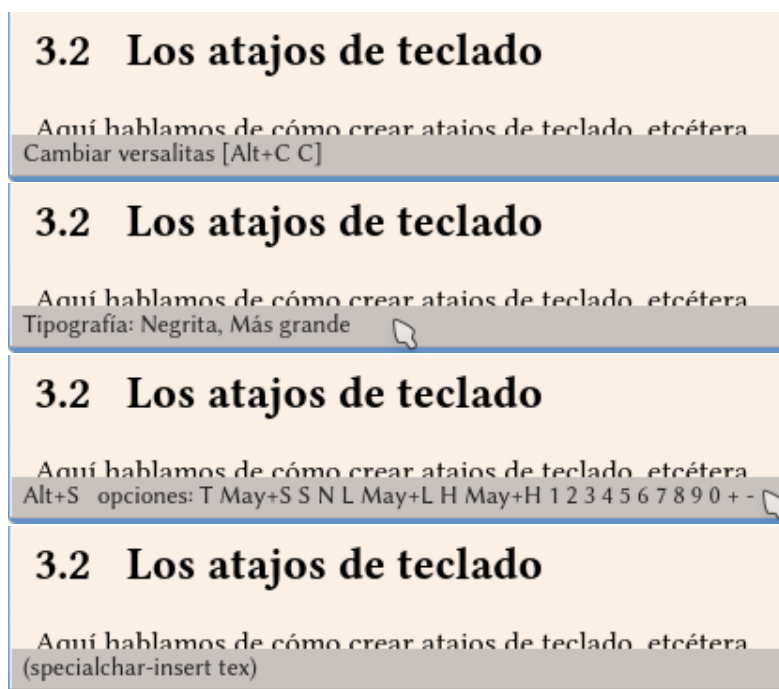


FIGURA 3.2: La primera captura nos presenta la «barra de estado» mostrando la función de un botón de la barra de herramientas sobre el que se encuentra el puntero del ratón (nótese que si la acción tiene asociada un atajo de teclado, este se mostrará). La segunda captura nos presenta la información del formato sobre el cual se encuentra el cursor. La tercera captura muestra cómo al activar la primera parte de algunos atajos de teclado LyX nos presenta todas las opciones disponibles. En la última captura vemos información relevante de una acción realizada a través de los menús y que no tiene (aún... volveremos a esto en la siguiente sección) un atajo de teclado asignado.

ño al tenerlo fruncido por muchas horas he decidido abandonar el tema, al menos por el momento: en la presente edición del libro no hablaré de las expresiones regulares.

Créanme, es mejor así.

Si el lector quiere probar por sí mismo, en la sección 6.13.3 del manual que se encuentra en [Ayuda](#) > [Guía del usuario](#) hay más información sobre la «búsqueda avanzada». ¡Buena suerte con eso!

Desde el menú [Ver](#) > [...](#) tenemos acceso a varias opciones.

Sobre el «panel de esquema» hablaremos en la sección [4.1.11](#).

El «panel de código fuente» nos muestra en tiempo real el código  $\text{\LaTeX}$  generado por  $\text{\LyX}$ .

El «panel de mensajes» nos muestra los mensajes producidos por  $\text{\LyX}$  mientras trabaja. Solo lo necesitaremos si encontramos muchos errores a la hora de tratar de compilar el documento.

Bajo «barras de herramientas» podemos activar o desactivar otras barras. Debe tenerse en cuenta que aquellas barras que dicen «auto» entre paréntesis se activarán solo cuando sean necesarias, como las barras relacionadas con el editor de ecuaciones.

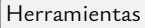
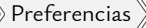


## 3.2 Los atajos de teclado

Como ya hemos contado tenemos distintos tipos de atajos de teclado en  $\text{\LyX}$


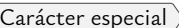

- Los «tradicionales» como [Ctrl](#) [B](#) para pasar a negrita.
- Los «secuenciales» donde se utiliza un «atajo de activación» que luego viene seguido por otras teclas o combinaciones de teclas. Por ejemplo [Alt](#) [P](#) seguido de [2](#) activa el entorno «sección».

Como vimos en la figura [3.2](#) cuando comenzamos a escribir un «atajo secuencial» la barra de estado nos ofrece las opciones disponibles para completarlo.

Claramente no todas las funciones de  $\text{\LyX}$  tienen un atajo de teclado asignado. Para crear un nuevo atajo para una opción que utilizamos mucho

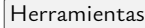
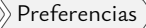
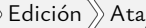

debemos dirigirnos a    : elegimos de la lista la función, hacemos clic en «modificar», presionamos la combinación elegida y luego aceptamos: LyX nos avisará amablemente si el atajo seleccionado ya está en uso por lo que solo tendremos que cambiar la selección.






Ahora bien, no todas las funciones de LyX están en esta lista, ¿es posible crear un atajo de teclado para esas funciones ausentes? ¡Ciertamente! Aquí tenemos que prestar atención nuevamente a la «barra de estado» de la que hablamos al final de la sección anterior: como también vimos en la figura 3.2, cuando una acción es realizada desde el menú allí se mostrará el «código» de la misma.

Supongamos que queremos crear un atajo de teclado secuencial para insertar el «logo» de LyX. Este se encuentra en    LyX logo y al insertarlo desde allí veremos en la barra de estado el siguiente texto:

```
specialchar-insert lyx
```

Esto es lo que necesitamos para crear el nuevo atajo. Eso sí: ¡atención, que hay que respetar las mayúsculas!

Volvamos a    , pero hagamos clic sobre «nuevo»: en «función» escribimos el código que hemos leído en la barra de estado y en la casilla correspondiente introducimos nuestro atajo, tal y como se ve en la figura 3.3

Este nuevo atajo es un tributo a los atajos «secuenciales»:   y luego la secuencia ,  y .

### 3.3 Muchos documentos, en muchas formas

LyX nos permite trabajar simultáneamente con varios documentos separados en pestañas en la misma ventana o con varias instancias del mismo documento en forma de «pantalla dividida».

Cada vez que abrimos un nuevo documento este se agrega en una nueva pestaña (como en los navegadores de internet). Para dividir la vista y tener simultáneamente dos instancias del mismo documento una al lado de la otra

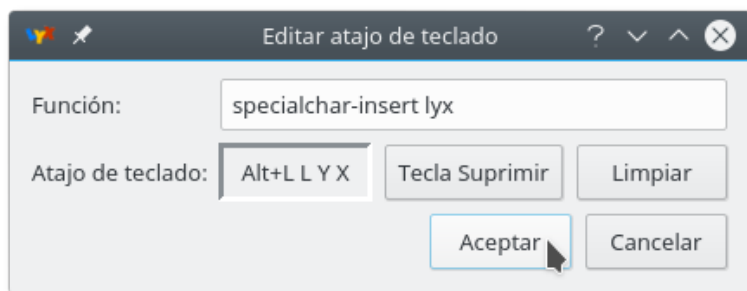


FIGURA 3.3: *Crear un nuevo atajo de teclado*

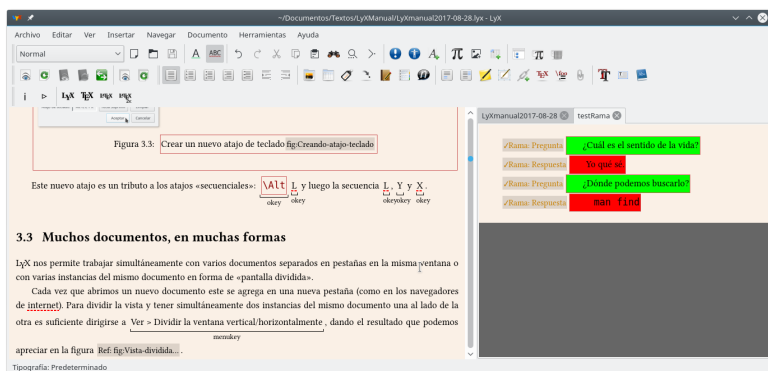


FIGURA 3.4: *Vista dividida + pestañas: para todos los gustos*

es suficiente dirigirse a **Ver >> Dividir la ventana vertical/horizontalmente**, dando el resultado que podemos apreciar en la figura 3.4.

Es importante notar que si luego de dividir la vista, digamos, verticalmente abrimos otro documento este aparecerá como pestaña *en la vista activa*.

Dejaré al lector el jugar con estas opciones. Eso sí: no olvide de probar el clic derecho sobre una pestaña.



## Capítulo 4

# Construir documentos

En este capítulo veremos rápidamente (que para algo está la documentación del programa) cómo crear un documento simple desde cero, elegir su tipo, navegar y modificar su contenido. Presentaremos las herramientas básicas que nos ofrece LyX y cómo trabajar con ellas. Para todo esto supondremos que el lector ha ya leído el capítulo anterior.

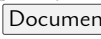
No entraremos aún en los «detalles L<sup>A</sup>T<sub>E</sub>X» (que para eso tenemos los capítulos más avanzados), pero lo que aquí se presenta resulta esencial para comprender el manejo del programa.

### 4.1 Trabajando en LyX

Crear un documento nuevo LyX no es diferente que en otros programas: podremos utilizar  Nuevo,  o el correspondiente botón de la barra de herramientas. Pero claro, eso es solo el comienzo.

Por defecto un documento nuevo de LyX utiliza el *documentclass* «artículo», el cual es seguramente el más simple de los «tipos de documento» ofrecidos por L<sup>A</sup>T<sub>E</sub>X.

#### 4.1.1 Formato general del documento

Todo (bueno, casi, pero ya volveremos sobre esto) lo que necesitamos se encuentra en el menú  Configuración, tal y como se puede ver en la figura 4.1.

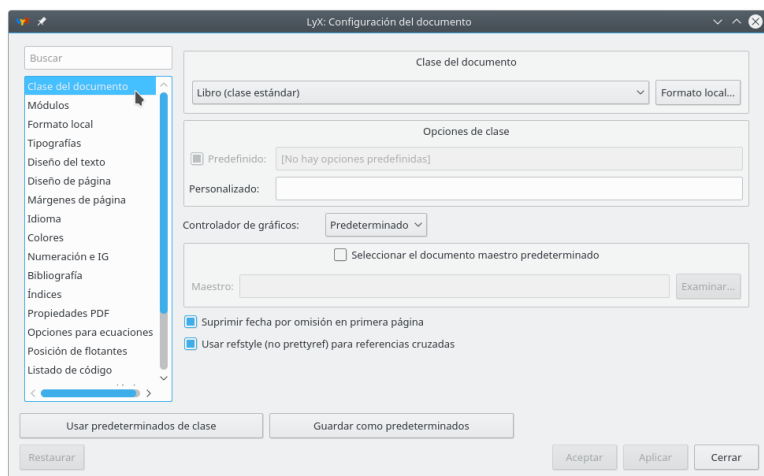


FIGURA 4.1: El menú **Documento** » **Configuración**

En la primera voz podemos seleccionar el *documentclass* que nos interesa. Desafortunadamente la única forma de saber cuál nos interesa es creando un documento de prueba para «ver cómo se ve», por lo que no entraré en detalles sobre cómo elegir el tipo correcto.

Eso sí, he aquí un consejo general: un par de *documentclass* están «repetidas» ofreciendo una versión «estándar» y otra «con más tamaños de tipografías». Por ejemplo, «Libro (clase estándar)» y «Libro (clase estándar con más tamaños de tipografía)». Olvídense de la segunda.

Si bien volveremos a hablar sobre las cuestiones de «tamaño» en la sección 5.3, veamos en qué se diferencian estas clases.

Si nos dirigimos a la voz *tipografías*, bajo *tamaño de base* nos encontraremos con una serie de posibilidades. Además de la opción *Predeterminado* las clases «normales» ofrecen tres tamaños de fuente para el texto: 10 puntos (generalmente, la opción predefinida), 11 y 12, mientras que las clases «con más tamaños» ofrecen, pues, más opciones.

Para este documento he utilizado «Libro (clase estándar)» seleccionando el tamaño de 11 puntos, pero las otras opciones son perfectamente válidas y algunas ofrecen un sinnúmero de posibilidades, todas ellas explicadas en la documentación que acompaña cada paquete.



En esta primera voz también tenemos la opción «suprimir fecha por omisión en la primera página» que hace... justamente eso: por defecto en las clases de documento de libro  $\LaTeX$  agrega la fecha en la que se compila el documento, lo cual no siempre es lo que se quiere. De hecho, casi nunca es lo que se quiere. Pues bien, marcando esa opción tal y como se ve en la captura de pantalla la fecha no se presenta.

Algo muy importante que también se encuentra allí es el apartado «opciones de clase». Esta caja nos permite pasar parámetros al *documentclass* que afecten el comportamiento del documento. Por ejemplo, en las clases de documento de «libros», cuando seleccionamos la opción «a dos caras» tendremos que por defecto los capítulos comenzarán en páginas impares (derechas): si queremos que los capítulos puedan comenzar en cualquier página podemos agregar en «personalizado» la instrucción *openany*. Las opciones disponibles dependerán del *documentclass* elegido por lo que invito al lector a consultar la documentación correspondiente.

Sobre «módulos» y «formato local» hablaremos en el capítulo 8.

La voz «tipografías» es, como resulta fácil imaginar, el lugar donde se seleccionan las fuentes tipográficas. El elemento más importante allí es la caja «Usar tipografías no TeX (via XeTeX/LuaTeX)»: esta opción nos permitirá seleccionar fuentes OpenType instaladas en el sistema en lugar de las fuentes propias de  $\LaTeX$ . Volveremos sobre estos temas en el capítulo 6.

En este menú la versión 2.3 nos ofrece una novedad bastante «técnica»: hacer que los guiones largos – y — (que en inglés se conocen como *n-dash* y *m-dash*, respectivamente) sean convertidos en, o más bien «mantenidos como», *ligaduras*.

La cuestión va más o menos así. Históricamente, en  $\LaTeX$  los guiones largos se introducen como «ligaduras  $\TeX$ »: básicamente dos guiones cortos se convierten en un guión medio (el *n-dash*) mientras que tres guiones cortos se convierten en uno largo (el *m-dash*) de la misma forma que una *f* seguida de un *i* se convierte en la ligadura «fi» que tenemos en la palabra «final». Este tipo de ligaduras  $\TeX$  no son verdaderas ligaduras ofrecidas por la fuente sino un «truco» que  $\LaTeX$  nos propone para escribir menos. Existe sin embargo otra forma de utilizar estos guiones: las instrucciones `\textendash{}` y `\textemdash{}`. A los efectos prácticos no hay mucha diferencia (al menos no en el PDF resultante) y en las últimas versiones  $\LaTeX$

automáticamente transformaba el número correcto de guiones cortos en el correspondiente guión largo usando las instrucciones apenas mencionadas, por lo que el usuario no necesita preocuparse. El problema surge con documentos viejos donde se tienen estos guiones como ligaduras: ¿qué hacer, transformar las ligaduras en guiones literales o usar solo ligaduras? Pues a partir de la versión 2.3 tenemos la posibilidad de elegir nuestro veneno.

Para los usuarios nuevos que escriben documentos nuevos en LyX aconsejo dejar esta opción en su valor por defecto (es decir, *no* dejar los guiones largos como ligaduras), pero para gustos los colores: quienes tengan que exportar el documento LyX a L<sup>A</sup>T<sub>E</sub>X puro podrían estar interesados en esta opción ya que haría más simple la lectura del código, pero insisto, para trabajar dentro de LyX mejor dejar las cosas como son.

Resuelto esto, sigamos con el temas de las fuentes tipográficas utilizadas.

En un documento L<sup>A</sup>T<sub>E</sub>X tenemos tres «familias» básicas de fuentes: la «Romana» que es la fuente «base», «palo seco» y «ancho fijo». Como ya vimos más arriba la caja «Tamaño» nos permite definir el tamaño de la fuente de base mientras que el tamaño de las otras dos familias puede determinarse en forma proporcional al tamaño de base: por ejemplo, en este libro he utilizado una escala del 80% para la fuente de ancho fijo<sup>1</sup>.

Las voces «Diseño de texto», «Diseño de página» y «Márgenes de página» son simples... salvo por el interlineado. Pero ese tema tiene sus vueltas por lo que lo discutiremos más adelante, también en el capítulo 6.

Solo una cuestión a remarcar aquí, importante para quien venga de trabajar con Writer: en un documento L<sup>A</sup>T<sub>E</sub>X/LyX los márgenes indican la distancia del borde de la página al área de texto por lo que las cabeceras y pie de página caen *dentro* del margen. Es por esto que en la voz «Diseño de página» tenemos entradas específicas para configurar la posición y la altura de las cabeceras o pie de página.

Sobre la parte del idioma hablaremos más abajo.

La voz «Colores» nos permite seleccionar, bien, los colores del documento (texto, notas, fondo de página).


«Numeración e IG» se refiere a hasta dónde queremos numerar la estructura del documento. Por defecto en las clases de libros tenemos nume-

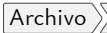
---

1. Y es que DejaVu Sans Mono tiene un tamaño «generoso».

ración automática para Parte, Capítulo, Sección, SubSección y SubSubSección, pero los párrafos no son numerados. Además, el índice general muestra por defecto solo los «niveles» que tienen numeración. Pues bien, deslizando las dos barras superiores podemos regular qué se numera y qué no, o qué se muestra en el índice o qué no aparece.

Y así siguiendo...

Es importante notar que en LyX el concepto de «plantilla» existe, pero es más *débil* que en otros programas: como podemos ver en el menú  tenemos la opción de *cargar* una plantilla, pero no vemos cómo esta pueda ser *creada*... y es que para LyX las plantillas son documentos normales, por lo que si queremos reutilizar las opciones elegidas para un documento particular tenemos dos opciones:

- El botón «Guardar como predeterminado» que tenemos en el menú de configuración del documento hará que la configuración que hemos elegido se aplique a *todos* los documentos nuevos;
- Salvar un documento vacío por cada grupo de opciones (uno para los libros, otro para los artículos, etcétera) y luego llamar esos documentos desde  Nuevo desde plantilla para que se cree una copia sin nombre del mismo desde la cual comenzar a trabajar.

LyX ya viene con varias plantillas predefinidas para crear documentos especiales (publicaciones específicas como *Springer* y *Scientific American*, por ejemplo). Una de estas plantillas, PDF-cropped.lyx nos permite crear un documento PDF a partir de una fórmula matemática, por ejemplo, que luego podamos utilizar en otras aplicaciones.

En próximos capítulos hablaremos de algunas de las otras voces del menú de configuración (discutiremos las «Propiedades PDF» y las «Ramaz» al final de este capítulo, la parte de «Salidas» en el capítulo 6 y el «Preámbulo LaTeX»... en muchos lugares).

#### 4.1.2 Texto

Ya hemos elegido fuentes, página, márgenes y demás, ¡a escribir!


Como comentamos en la introducción un concepto fundamental a la hora de utilizar LyX es el de «entorno»: con el menú (o con atajos de teclado) que se encuentra a la izquierda de la barra de herramientas principal,

justo debajo del menú «archivo», podemos asignar a cada párrafo la «función» que queremos darle: título de capítulo, de sección, texto normal, lista numerada, viñeta, etcétera.

Todo está perfectamente clasificado y comprensible por lo que dejaré al lector la tarea de jugar un poco con esas opciones para acostumbrarse a sus funciones. Lo único importante a comentar es que los elementos «sin numerar» (los que tienen el asterisco) no serán incluidos en el índice: volveremos sobre esto en el capítulo 10.

De esta forma el documento mostrará una estructura bien definida y coherente.





En LyX tenemos algunos estilos de texto predefinidos: **Ctrl E** activa y desactiva la *cursiva* (el comando `\emph{}`), **Ctrl B** activa y desactiva la negrita (el comando `\textbf{}`) mientras que la secuencia **Alt C**, luego **C** hace lo propio con las VERSALITAS (`\noun{}`). Estas tres opciones también pueden manejarse desde la barra de herramientas.

NOTA: Además de volver a aplicar el «estilo» al texto para desactivarlo, tenemos la combinación **Alt C**, luego  para volver al formato predeterminado.

Para esas raras situaciones en las que el formato directo pueda ser necesario tenemos los menús **Editar** » **Estilo del texto** » **Personalizado** y **Editar** » **Configuración del párrafo**, ambos accesibles desde el clic derecho.


### 4.1.3 Listas



Directo desde la barra de herramientas o desde el menú desplegable «entorno» podemos elegir entre los entornos «enumeración» (**Alt P**, luego **E**), «enumeración\*» (**Alt P**, luego **B**), «lista» (**Alt P**, luego **L**) y «descripción» (**Alt P**, luego **D**).

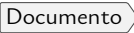
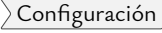
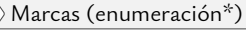
Las dos primeras son las clásicas «listas numeradas» (la primera) y «viñetas» (la del asterisco) a las que otros procesadores de texto nos tienen acostumbrados. Estas pueden anidarse y mostrar varios «niveles», pasando de uno a otro ya sea desde la barra de herramienta o con los atajos **Alt**   para subir un nivel y **Alt**   para bajarlo. Algo así:

- Primer nivel

- Segundo nivel
  - Otro segundo nivel
    - ◊ Tercer nivel
  - Un nuevo segundo
- Un nuevo primero

Para salir de la lista es suficiente seleccionar el entorno «normal» o presionar  dos veces.

Tanto las listas como las descripciones separan una «palabra a definir» del texto que la define, cambiando solamente en los detalles del formato del párrafo y cómo la descripción viene separada de la «palabra clave». Dejo al lector el jugar con estas opciones. Solo una cosa: si queremos más de una palabra en la «etiqueta» tendremos que unir las con espacios protegidos ( ) para que LyX las considere como una única palabra.

LyX nos permite configurar en forma realmente simple las viñetas de nuestro documento. Para esto simplemente nos dirigimos a    donde podremos seleccionar el símbolo para los cuatro posibles niveles, ya sea de las seis «tipografías» propuestas o insertando nuestros propios símbolos. También podemos cambiar el tamaño de los símbolos seleccionando una de las opciones que va desde «diminuta» hasta «gigante»<sup>2</sup>.

Configurar las listas numeradas no es tan sencillo por lo que lo dejaremos para la última parte de este libro, en el capítulo 9.

#### 4.1.4 Matemáticas

Existen dos «estilos» de expresiones matemáticas: «en línea» y «presentada», esta última pudiendo ser «numerada» o no.

La diferencia principal entre estos dos estilos es que las expresiones «en línea» presentan signos (integral, sumatoria) más pequeños para integrarse mejor como parte de un párrafo. Comparemos cómo se ve esta expresión en

---

2. Volveremos al significado de esos nombres en el capítulo 6 donde hablamos sobre tipografía.

línea  $\int_a^b f(x)dx$  con esta otra «presentada»

$$\int_a^b f(x)dx$$

La forma «en línea» es accesible desde la barra de herramientas principal mientras que ambas formas pueden insertarse desde el menú **Insertar** > **Ecuación**, si bien la forma más simple de insertar expresiones matemáticas es seguramente con los atajos de teclado **Ctrl** **M** para las expresiones en línea y **Ctrl** **⇧** **M** para las «presentadas». Estos atajos de teclado pueden utilizarse para transformar un tipo de expresión en otra: presionando **Ctrl** **⇧** **M** dentro de una expresión «en línea» esta se transformará en una «presentada».





Las ecuaciones numeradas pueden crearse desde **Insertar** > **Ecuación** > **Numerada** o bien podemos transformar una expresión «presentada» con **Alt** **M**, luego **N** (esta combinación también «quita» la numeración):


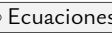
$$\int_a^b f(x)dx \tag{4.1}$$

Más adelante en este capítulo veremos cómo realizar referencias cruzadas a expresiones matemáticas numeradas.

Por defecto las expresiones «presentadas» vienen centradas en la página, pero a partir de la versión 2.3 LyX ofrece soporte para la instrucción `mathindent`, lo cual facilita el lograr que las expresiones estén sangradas en lugar de centradas. Para esto es suficiente dirigirse a **Documento** > **Configuración** > **Opciones para ecuaciones** y modificar la opción correspondiente. Allí también resulta posible seleccionar de qué lado de la ecuación se presentará el número.

En la figura 2.1 en la página 20 vimos al editor de ecuaciones en pleno trabajo, mostrando cómo pueden utilizarse menús, botones o código puro. Claramente muchísimos atajos de teclado se encuentran también disponibles. Muchos de ellos comienzan con la combinación **Alt** **M**: **Alt** **M**, luego **G**, luego una letra nos dará un carácter griego, **Alt** **M**, luego **F** una

fracción, , luego  los paréntesis escalables, , luego  pasa los límites de una integral de la derecha a que se vean arriba y abajo (la instrucción `\limits`), etcétera. Al pasar el puntero del ratón sobre los botones de las barras de herramientas podremos ver los atajos correspondientes para así ir aprendiéndolos cómodamente.

Las posibilidades que ofrece el editor de ecuaciones de L<sup>A</sup>T<sub>E</sub>X son casi infinitas y realmente no tiene sentido que aquí me dedique a duplicar lo que ya está magníficamente explicado en la documentación del programa, por lo que luego de esta breve introducción solo me queda invitar al lector a consultar el manual que se encuentra en  : a no preocuparse de sus más de ochenta páginas, que la lectura es amable y rápida.

#### 4.1.5 Cuadros: ¡que no se llaman tablas!

Antes de entrar en el tema específico de este apartado me permitiré una pequeña discusión filológica: cuadro vs. tabla.

Aquí nos encontramos ante otra intrusión del idioma inglés que cambia el significado de palabras establecidas en nuestro idioma. Claramente y dado que los idiomas «vivos» son dinámicos tarde o temprano el cambio terminará por hacerse oficial<sup>3</sup>, pero eso aún no ha sucedido por lo que me permitiré insistir en que la palabra correcta para referirnos a *datos tabulados* es *cuadro* y no *tabla*.

Según la Real Academia Española una de las definiciones de *cuadro* es, justamente

9. m. Conjunto de nombres, cifras u otros datos presentados gráficamente, de manera que se advierta la relación existente entre ellos.

La relación entre las palabras *cuadro* y *tabla* es por lo tanto la misma que en inglés existe entre *table* y *tabular*. En castellano se puede hablar de *la tabla periódica de los elementos*, pero si tenemos datos tabulados con un título (lo que L<sup>A</sup>T<sub>E</sub>X llama «leyenda») numerado, pues que eso es un *cuadro*.

Y es justamente por eso, para complicar las líneas de razonamiento todo lo posible, que en este apartado que tiene el nombre que tiene no hablaré

3. Como el verbo «nominar»: <http://blog.lengua-e.com/2017/errores-correctos-nominar/>

sobre cómo crear un *cuadro* completo con todo y leyenda (volveremos sobre eso más abajo al hablar de *flotantes* en el apartado 4.1.7) sino sobre cómo crear los datos tabulados que irán dentro de esos cuadros.

Y es que soy una persona complicada...

Pues bien, terminada esta divagación sigamos con lo nuestro: ya sea desde la barra de herramientas estándar o desde el menú **Insertar** » **cuadro/tabla** nos encontraremos con las clásicas opciones para elegir el número de filas y columnas que nos sirven.

Con el cursor dentro del cuadro/tabla tendremos a disposición en la parte baja de la pantalla de edición la barra de herramientas de cuadros, la cual nos permite insertar o quitar filas o columnas, unir celdas (multicolumnas o multifilas, los dos últimos botones a la derecha), seleccionar bordes, girar una celda o todo el cuadro de 90°, etcétera.

Los cuadros/tablas pueden *anidarse*: colocando el cursor en una celda podemos insertar otro cuadro. Un uso de este anidamiento es simular celdas de varias líneas ya que es imposible insertar quiebres de párrafo o de línea en una celda: simplemente insertamos allí un cuadro sin bordes de una columna y el número necesario de filas.

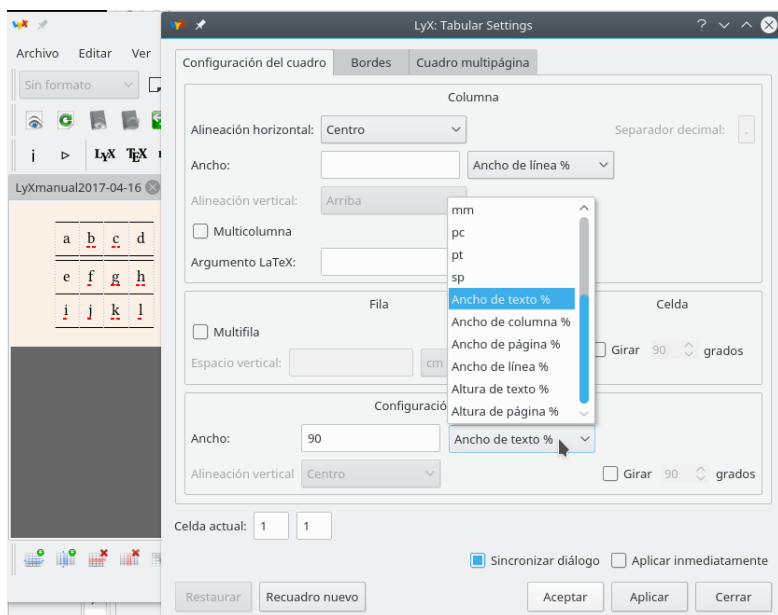
Con un **clic derecho** » **Configuración** tenemos muchas más opciones como girar el contenido de una celda en un ángulo arbitrario o configurar un cuadro «multipágina» cuyo contenido pueda dividirse en varias páginas en forma automática.

En la pestaña «bordes» de la configuración del cuadro tenemos a disposición dos «estilos» para nuestro cuadro: predefinido y elegante. El primero presenta bordes tanto horizontales como verticales mientras que el segundo solo horizontales.

Ciertamente podemos crear el formato que queramos para nuestro cuadro, pero por el momento no existe un modo de salvar formatos personalizados para utilizarlos más tarde por lo que si no nos gustan los estilos disponibles y no queremos rehacer el formato del cuadro cada vez que necesitamos uno nuevo la única opción que nos queda es el mantener un documento separado con algunos ejemplos de cuadros desde los cuales copiar y pegar a nuestros documentos.

Algo importante de recordar es que una de las cosas que podemos modificar de la configuración del cuadro es el ancho del mismo: por defecto ten-




FIGURA 4.2: *Configurar el ancho de un cuadro*

drá el ancho mínimo necesario para acomodar el contenido, pero podemos fácilmente hacer que ocupe un porcentaje determinado del área del texto, por ejemplo, como puede verse en la figura 4.2. De hecho, fijado el ancho del cuadro/tabla podremos luego fijar los anchos relativos de cada columna.


Ahora bien, en ninguna de las opciones presentadas se habla de *color*, ¿cómo hacemos para dar color a nuestras tablas?

Para cambiar el color del texto no se necesita más que un clic derecho Estilo de texto Personalizado o bien utilizar un estilo de carácter definido por nosotros (volveremos sobre eso en el capítulo 8). ¿Pero cómo hacemos para darle color al fondo de una celda? Pues para eso necesitamos un paquete  $\text{\LaTeX}$  especial (`colortbl`) y un poco de trabajo: todo está explicado en la sección 2.11 del manual que encontramos en Ayuda Objetos insertados, pero antes de intentarlo, lector, te recomiendo leer la tercera parte de este libro, que en este tema no podrás escapar del código  $\text{\LaTeX}$  puro y duro.

### 4.1.6 Marcos

En el menú  Marco tenemos varias opciones (sin borde, borde sencillo, oval...), la pregunta, claro está, es qué hacen esas opciones.

Las opciones presentadas en ese menú lo que hacen es insertar algo que en L<sup>A</sup>T<sub>E</sub>X se llama `minipage` (minipágina): básicamente nos da una pequeña página flotando sobre la página real. Esta pequeña página puede tener algún borde o incluso un color de fondo. Es importante notar que estos marcos no se dividen por lo que si el contenido no cabe en la página real, pues tendremos algunos problemas... El uso de estos marcos es por lo tanto el de insertar notas, por ejemplo, o pequeñas «aclaraciones» que complementen el contenido principal.

Ahora bien, si lo que hace  Marco es insertar una minipágina, ¿no debería llamarse «insertar minipágina»? La respuesta es, claramente, que no, que tenemos más opciones.

Con un clic derecho en la etiqueta de la caja que indica el marco obtenemos un menú que no solo nos permite cambiar entre las opciones disponibles, sino que también nos da acceso a la configuración de ese marco.

En este menú de configuración podemos cambiar varias cosas, como el ancho del marco, etcétera. Pero por sobre todas las cosas podemos elegir el tipo de «marco interior», tal como vemos en la figura 4.3.

Tenemos tres opciones allí:

**MINIPÁGINA** Lo que ya hemos discutido más arriba.

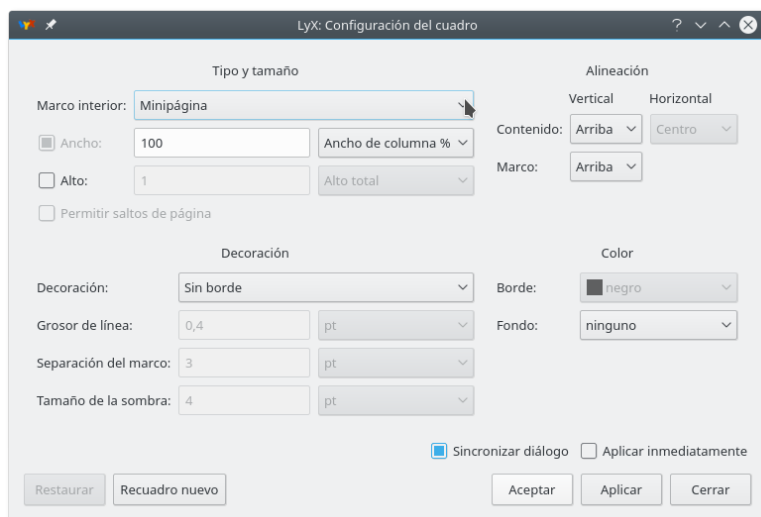
**MARCO DE LÍNEA** Para textos sumamente breves que deben ocupar solamente una línea

**MARCO DE PÁRRAFO** Para enmarcar un párrafo particular

Las opciones de este menú son claras, por lo que otra vez dejaré al lector el jugar un poco con ellas.

### 4.1.7 Figuras, cuadros y otros «flotantes»

En programas como Writer si queremos una imagen o un cuadro (que ya dije que no son tablas) con un título/leyenda generalmente insertamos

FIGURA 4.3: *Configurando un marco*

primero la imagen y luego la leyenda. Pues bien, en LyX esto se hace exactamente al contrario.

Ya sea con el correspondiente botón de la barra de herramientas o desde el menú **Insertar > Flotante > ...** podemos insertar un «flotante» en el cual escribir la leyenda e insertar luego la imagen o cuadro.


Estos flotantes hacen exactamente lo que sus nombres indican: flotar sobre la página. A dependencia del espacio disponible en la página estos flotantes se colocarán ya sea en la parte alta o baja de la página o pasarán a la página siguiente (si no hay espacio) permitiendo al texto fluir cómodamente alrededor de ellos sin mayor trabajo por parte del usuario.

El comportamiento de los flotantes puede ser configurado ya sea para cada flotante en particular (clic derecho sobre la etiqueta del mismo) como en forma general para el documento desde **Documento > Configuración > Posición de flotantes**. Generalmente dejar que L<sup>A</sup>T<sub>E</sub>X haga este trabajo por nosotros («usar ubicación predeterminada») es una buena idea ya que lo hace *realmente* bien.

Ahora bien, LyX nos permite elegir fácilmente si queremos que la ima-

gen o el cuadro esté por encima o por debajo de su título simplemente moviendo el cursor hacia arriba o hacia abajo de este último, tal y como muestra en la parte superior de la figura 4.4.


Otra cosa que muestra la figura 4.4 es la posibilidad de trabajar con subflotantes: figuras (o cuadros) dentro de figuras, numeradas como «figura 3A», etcétera. Simplemente insertamos flotantes dentro del flotante principal hasta satisfacer nuestras necesidades. Estos «subflotantes» pueden ser puestos en cualquier posición: uno arriba y otro abajo, uno al lado del otro... dejo al lector interesado la responsabilidad de explorar esta posibilidad.

Es importante notar que si queremos un flotante *sin* leyenda simplemente tenemos que borrar la caja correspondiente. Si más tarde queremos reintroducirla, con el cursor en el flotante nos dirigimos a  Insertar » Leyenda.

Hablaremos más sobre figuras y otros flotantes en el capítulo 14.

#### 4.1.8 Referencias cruzadas

LyX no cuenta con un mecanismo de etiquetado automático por lo que para realizar una referencia cruzada, ya sea esta al título de un capítulo, a una ecuación o a una figura numerada tenemos que realizar dos simples pasos.

Primero tenemos que crear la etiqueta, ya sea con el botón «insertar etiqueta» o desde  Insertar » Etiqueta junto al elemento al cual queremos hacer referencia y luego «llamar» a la misma con la opción «insertar referencia cruzada», otra vez desde el menú o del correspondiente botón.

Como ya comentamos en la introducción general las etiquetas pueden ser colocadas en distintos tipos de contenido: figuras, cuadros, ecuaciones numeradas, títulos de capítulo, parte o sección, notas al pie de página, lo que sea necesario. LyX comprenderá sin problemas qué tipo de contenido estamos etiquetando por lo que al insertar finalmente la referencia cruzada la numeración será correcta.

Es importante al insertar la etiqueta darle un nombre que luego podamos reconocer sin problemas, por lo que resulta conveniente aceptar el esquema que nos ofrece LyX por defecto anteponiendo unas pocas letras para saber si estamos llamando a un capítulo, figura, ecuación, etcétera.

Ahora bien, supongamos que hemos cambiado el título de un capítulo previamente etiquetado, o que promovimos una subsección a sección, ¿podemos cambiar el nombre de la etiqueta? ¡Por supuesto! ¿Y sin temor a

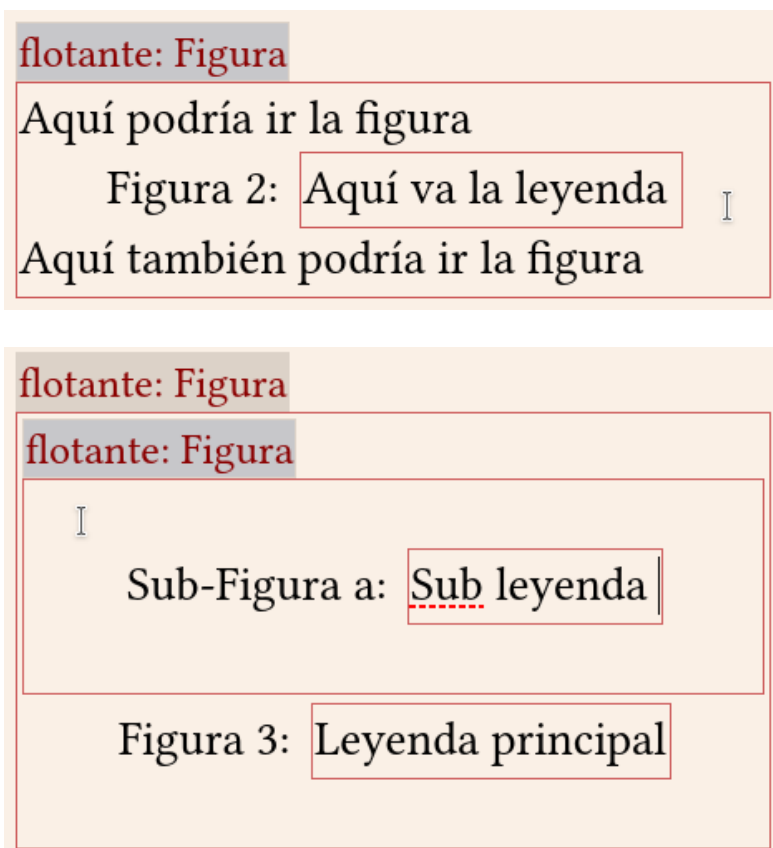


FIGURA 4.4: *Trabajando con flotantes. En la parte superior de la figura vemos cómo se presenta el flotante en la ventana de edición de LyX, indicando con texto dónde podríamos colocar la imagen. En la parte inferior de la figura vemos que resulta muy simple trabajar con subfiguras: simplemente insertamos los flotantes que necesitamos dentro del flotante principal y LyX automáticamente sabrá qué hacer. Como puede verse en esta figura, no me agradan las subfiguras...*

*arruinar las referencias cruzadas a la misma?* ¡Faltaba más! Con un simple clic sobre la etiqueta podremos cambiar más tarde y en total tranquilidad su nombre por otro más apropiado que las referencias cruzadas a la misma se actualizarán automáticamente.


Hablando de referencias cruzadas, el menú que se presenta al *insertarlas* es más que claro, por lo que a continuación solo indicaremos sus características más interesantes:

- En la parte superior nos permite elegir entre los documentos que tengamos abiertos, lo cual simplifica el trabajo de realizar referencias cruzadas entre documentos
- Sobre la lista de etiquetas disponibles tenemos un «filtro» que nos facilita encontrar la que estamos buscando
- En la parte inferior tenemos varias opciones de referencias cruzadas: número o texto del objeto al que se hace referencia, la página en la que se encuentra, ambas cosas...

En la figura 4.5 pueden verse ambos diálogos, el de crear la etiqueta y el de llamarla.

#### 4.1.9 Índices

Configurar un índice es quizás una de las tareas más pesadas en  $\text{\LaTeX}$ . Existen varias formas de hacer esto y por si fuera esto poco incluso varias métodos de crear uno.

Afortunadamente en la mayor parte de los casos el índice por defecto será perfecto, por lo que solo tenemos que ir a  para crear nuestro índice, ya sea este el general, el de figuras, cuadros, etcétera.

Siendo la configuración de los índices un tema avanzado de  $\text{\LaTeX}$  volveremos sobre esto en la parte final de este libro, en el capítulo 10.

##### 4.1.10 Generar PDF «dinámicos»: el paquete **Hyperref**

El paquete `hyperref` permite configurar la salida a PDF de documentos  $\text{\LaTeX}$  de modo tal que se tengan «enlaces vivos», ya sea dentro del documento, a sitios web o a otros archivos.

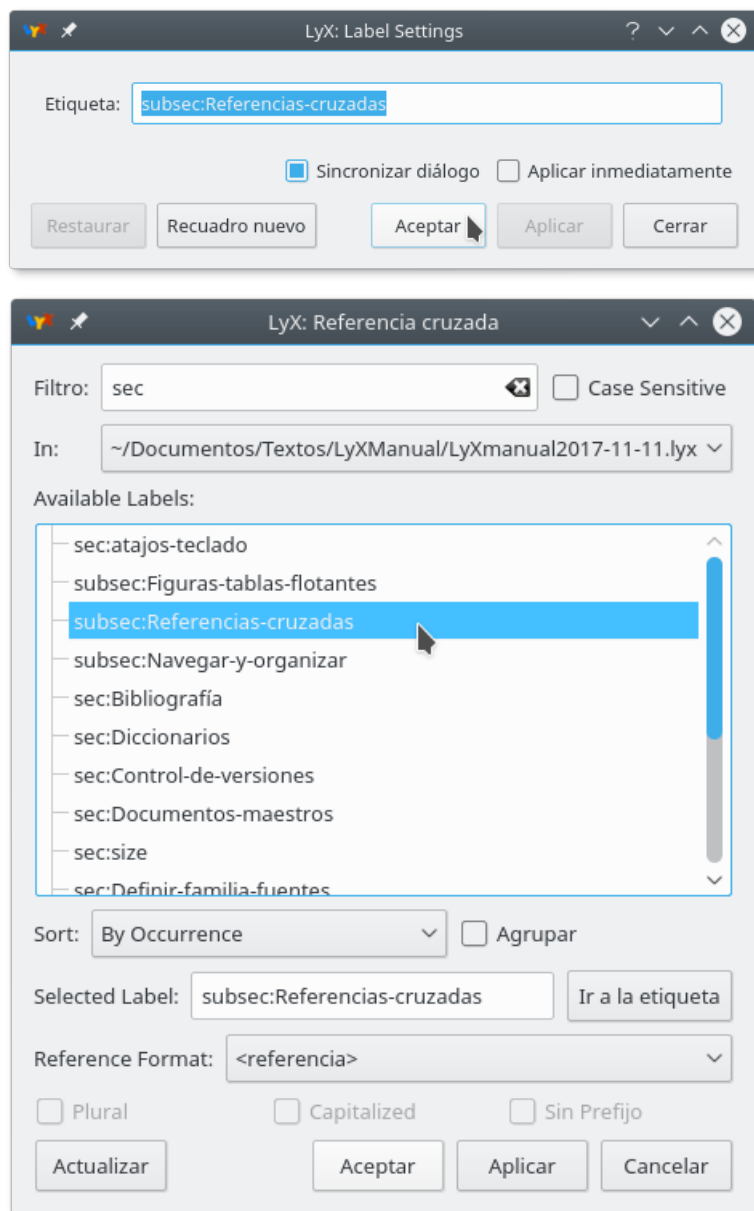


FIGURA 4.5: Crear una etiqueta (arriba) e insertar una referencia (debajo)

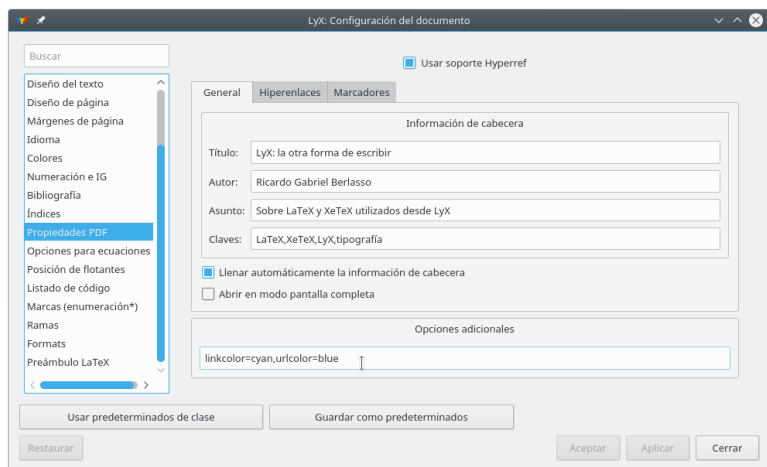


FIGURA 4.6: *Configurar el comando Hyperref*

LyX permite configurar varias opciones del paquete `Hyperref` simplemente seleccionando algunas casillas de control, mientras que las otras opciones (como el elegir los colores particulares de los enlaces) se pueden configurar escribiendo algunos parámetros en una línea de texto.

Si vamos a `Documento` `Configuración` `Propiedades PDF` nos encontramos con lo que nos muestra la figura 4.6.

Como puede verse la pestaña `General` nos permite no solo activar el paquete `Hyperref` para ese documento sino también establecer las propiedades generales del PDF que se generará, como el nombre, autor, si el PDF debe abrirse a pantalla completa, etcétera.

En la pestaña `Hiperenlaces` vemos algunas opciones del paquete `Hyperref`, como colorear los enlaces... pero no tenemos una opción gráfica para seleccionar los colores.

En la caja llamada `Opciones adicionales` es donde deben escribirse, separadas por comas, las opciones de `Hyperref` que no pueden configurarse gráficamente.

Como ejemplo, luego de marcar «enlaces coloreados» en la pestaña `Hiperenlaces` podremos determinar los colores de cada tipo de enlace utilizando las siguientes instrucciones:



**citecolor** fija el color de los enlaces a la bibliografía

**filecolor** fija el color de los enlaces a archivos

**linkcolor** fija el color de los enlaces internos del documento (referencias cruzadas, índice)

**urlcolor** fija el color de los enlaces a direcciones web

Los colores pueden indicarse ya sea por nombre (en inglés: *red*, *green*, *cyan*, *magenta*, *black*) o por valores RGB con tres números entre cero y uno. Yo suelo marcar la opción «sin marcos alrededor» ya que detesto que los enlaces estén recuadrados, pero para quienes les guste es posible determinar también el color de los marcos con instrucciones del tipo `linkbordercolor` y similares.

La documentación del paquete `Hyperref` que suele instalarse en `/usr/share/texmf/doc/latex/hyperref/manual.pdf`

Escribiendo algo como

```
citecolor=cyan,filecolor=green,linkcolor=red,  
urlcolor=blue
```


tendremos un PDF no solo «vinculado» sino también «colorido».

Algo importante para marcar en la pestaña Hiperenlaces es la opción «Quebrar enlaces en líneas», ya que de lo contrario un texto que contenga un enlace podría seguir y atravesar impunemente el margen derecho.

En la pestaña Marcadores no debemos olvidar de tener marcado no solo «generar marcadores (IG)» sino también la opción «marcadores numerados»: de esta forma el índice del PDF estará numerado apropiadamente.

Como último comentario, recuerde el lector que desde hace ya varias versiones en L<sup>A</sup>T<sub>E</sub>X existe una herramienta gráfica para insertar enlaces externos, por lo que no es necesario utilizar una caja T<sub>E</sub>X para eso (ver figura 4.7).

#### 4.1.11 Navegar y organizar el documento

Desde el menú  tenemos acceso a todo el contenido del documento: podemos ir desde allí rápidamente a cualquier capítulo, figura, cuadro, nota al pie, ecuación...

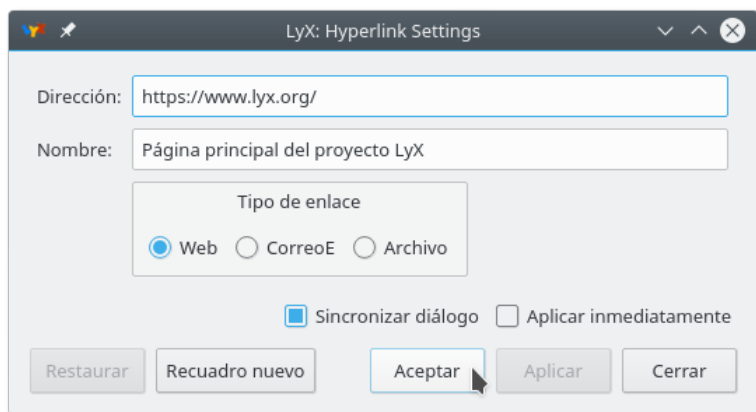


FIGURA 4.7: *Crear un enlace en el documento*

No quiero que el lector se pase por alto el menú **Navegar** **Otras listas** **Etiquetas y referencias**. Allí veremos no solo la lista de todas las etiquetas que hemos utilizado para «marcar» algo en el documento: también tendremos la lista de todas las *referencias* a esas etiquetas los que nos permite ver rápidamente desde dónde las estamos llamando.

Pero existe otra forma de navegar un documento que además nos permite *organizarlo*: **Ver** **Panel de esquema**, el cual puede verse en la figura 4.8.

Como nos muestra la captura de pantalla en la parte superior podemos elegir qué tipo de información queremos navegar o modificar (el índice general, la lista de figuras, etcétera). Debajo de esta opción tenemos un campo que nos permite filtrar la lista que sigue, la cual está organizada jerárquicamente en una vista «en árbol».

Es importante notar que haciendo clic sobre un elemento de la lista el punto de edición se moverá allí, *pero también colapsará esa rama de la jerarquía*. Para evitar que esto suceda tenemos el botón «Mantener» que hace que las ramas puedan abrirse o cerrarse solo haciendo clic en las flechas a la izquierda de la entrada.

La barra deslizable a continuación sirve para elegir en forma global hasta qué nivel las ramas estarán abiertas.

Finalmente, los botones de abajo a la derecha (que solo estarán dispo-

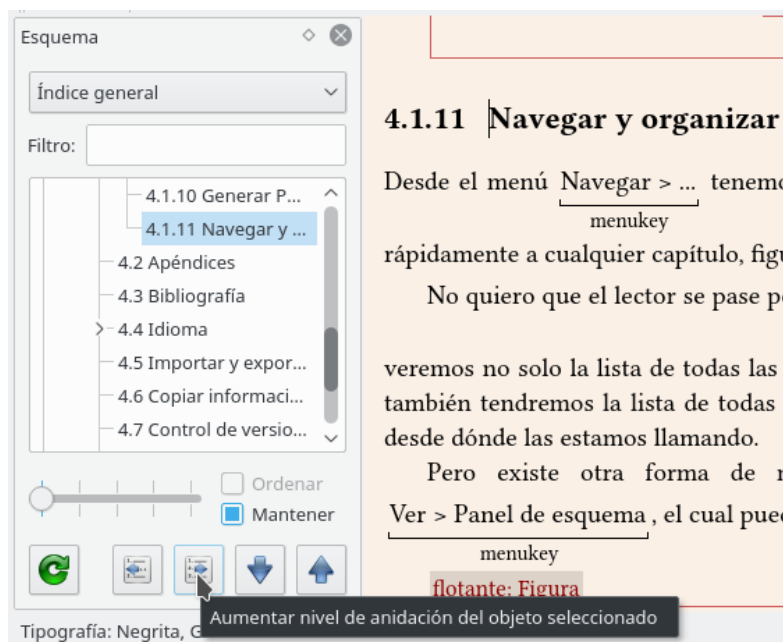
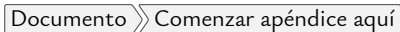


FIGURA 4.8: El panel de esquema durante la edición de este libro. La casilla «Mantener» resulta muy útil cuando es necesario ir y venir por el documento mientras que los cuatro botones de abajo a la derecha nos permiten modificar rápidamente tanto el orden como la jerarquía de los capítulos, secciones y demás elementos de nuestro libro.

nibles para el índice general) nos permiten cambiar el nivel de un título o mover el mismo sobre o debajo de otro del mismo nivel, permitiéndonos cambiar el orden de las entradas del documento en forma rápida y simple.

## 4.2 Apéndices

Con  iniciaremos una nueva «parte» del documento, indicada en la ventana de edición de  $\text{LyX}$  con una caja roja que dice «apéndice» en la línea superior, que contendrá los apéndices.

Para insertar un apéndice o un subapéndice simplemente utilizamos los entornos Capítulo y sección, respectivamente.

Las referencias cruzadas a los apéndices y a todo lo que esté en ellos funcionará correctamente sin más trabajo por parte del escritor.

Es importante notar que por defecto (¿o por definición?) los apéndices van al final del documento. Es *posible* crear apéndices «por capítulo» o «por parte», pero aún no he encontrado un motivo lo suficientemente bueno como para preocuparme de hacerlo.

## 4.3 Bibliografía


En un documento  $\text{LyX}$  tenemos dos formas de trabajar con referencias bibliográficas: interna al documento o utilizando una base de datos bibliográfica externa como  $\text{BIB}\text{\LaTeX}\text{E}\text{X}$ .

A partir de la versión 2.3  $\text{LyX}$  ofrece soporte para  $\text{BIB}\text{\LaTeX}\text{E}\text{X}$ , pero entrar en tales temas realmente escapa del propósito general de este libro por lo que solo daré una breve introducción al método «interno».

Todo comienza con el entorno «bibliografía». Tendremos luego de insertarlo una nueva «sección» llamada «Referencias» y una primera entrada vacía con una etiqueta en gris llamada «key-1[]». Con un clic sobre esa etiqueta podremos cambiar la clave por algo más descriptivo o darle una etiqueta que será utilizada para mostrar la cita en el documento: si la clave se deja en blanco, la cita será numerada.

Es importante notar que en ese diálogo no se aceptan espacios o caracteres no ASCII.



Luego de la etiqueta escribimos el texto de la referencia dándole formato «a mano» (itálica para el título, negrita para el número de página).

Finalmente nos dirigimos a  Cita para seleccionar la(s) cita(s) que necesitamos, seleccionar el formato, agregar texto a la cita (por ejemplo, «[5, capítulo 6]»), etcétera.

Dos cosas importantes a notar sobre las referencias así insertadas:

- Siendo un entorno siempre podemos «salir» del mismo para agregar otro entorno como un capítulo o sección *después*. Esto significa que podríamos crear un grupo de referencia nuevo por cada capítulo o sección del documento.
- La lista de referencias quedará en el exacto orden en la cual la hemos escrito, por lo que es necesario tener cuidado de insertar las entradas en el orden correcto. Recuerde el lector que siempre podemos mover párrafos hacia arriba o abajo con un simple clic derecho.









## 4.4 Idioma

En  Configuración  tenemos unas pocas opciones, pero todas ellas de suma importancia. En esta sección hablaremos de cada una.

### 4.4.1 Nuevo en 2.3: «comillas “anidadas” dinámicas»

En este menú siempre fue posible elegir qué tipo de comillas utilizaba nuestro documento, por ejemplo si las «latinas» o las «inglesas», pero nada más. En realidad el tema de las comillas es más complejo y muchos idiomas, como el castellano, admiten el uso de comillas anidadas como la que vemos en el título de esta subsección, siguiendo reglas propias del idioma<sup>4</sup>. Pues bien, a partir de la versión 2.3 tenemos la posibilidad de configurar también este tipo de situaciones más complejas, donde L<sup>A</sup>T<sub>E</sub>X automáticamente utilizará las comillas correctas en cada «nivel». Con un clic derecho sobre las comillas es posible cambiar su tipo. Pero qué quiere que le diga, yo ya estoy muy acostumbrado a utilizar el teclado español que me ofrece mi Linux, por

4. <https://es.wikipedia.org/wiki/Comillas>

lo que me resulta más simple utilizar   para «,   para »,   para “ y   para ”.

#### 4.4.2 Diccionarios

Para los usuarios Linux probablemente los diccionarios necesarios para corregir la ortografía estén ya instalados (los habrá arrastrado la instalación de LibreOffice), pero podrían faltar los diccionarios de sinónimos. O quizás queramos utilizar otros diccionarios más actualizados que los que nos ofrece nuestra distribución. Incluso podría darse que alguien tenga que utilizar L<sup>y</sup>X en otros sistemas operativos.

En esta sección comento cómo obtener diccionarios actualizados y cómo utilizarlos en L<sup>y</sup>X.

Primero, descargamos las extensiones de los diccionarios, por ejemplo desde la página de LibreOffice<sup>5</sup>, o bien, si queremos alguna de las variantes locales del español, desde el repositorio del proyecto «Recursos lingüísticos abiertos del español» (RLA-es) en GitHub<sup>6</sup>. Ahora cambiamos los nombres de estas extensiones de <diccionario>.oxt a <diccionario>.zip y los descomprimos, lo que nos dará varios archivos y posiblemente algunas carpetas. De todo esto lo que nos interesa son los archivos con las siguientes extensiones:

- .dic y .aff: estos archivos contienen los diccionarios de corrección ortográfica
- .dat y .idx: estos archivos contienen los diccionarios de sinónimos

Copiando estos archivos en una carpeta cualquiera, abrimos L<sup>y</sup>X y nos dirigimos a                      

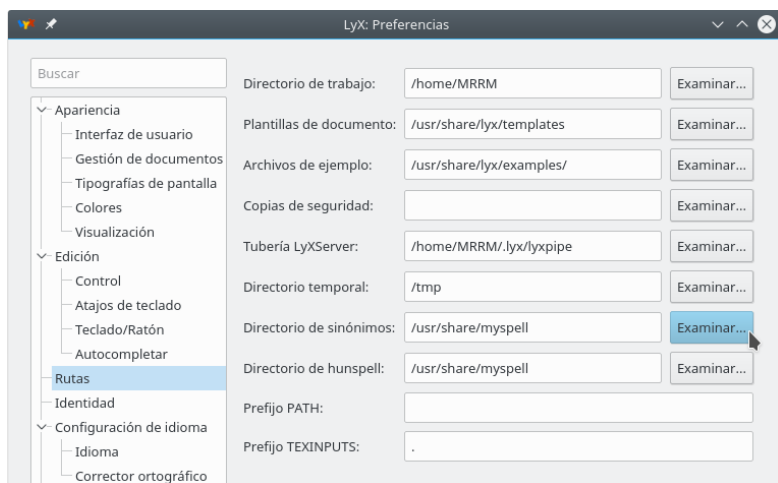


FIGURA 4.9: *Modificar las rutas a los diccionarios*

En el menú **Preferencias** » **Configuración de idioma** podemos elegir que la corrección ortográfica marque los errores encontrados en tiempo real, mientras que el atajo de teclado **⇧ F7** nos abrirá el diccionario de sinónimos en la palabra sobre la cual se encuentre el cursor. Es importante notar que si tenemos los archivos .aff en nuestra carpeta de diccionarios el diccionario de sinónimos reconocerá palabras «modificadas» (plurales, verbos conjugados, etcétera), si bien solo dará sugerencias «sin modificar» (ofrecerá singulares, verbos en infinitivo, etcétera).

#### 4.4.3 Babel vs. Poliglosia

La definición del idioma del documento afecta a mucho más que a la corrección ortográfica: desde cuestiones inmediatas como llamar a los capítulos «Capítulo» y no «Chapter» hasta temas más sutiles como decidir si luego de un título se mantiene la sangría de primera línea son todos temas altamente relacionados al lenguaje utilizado.

Históricamente  $\text{\LaTeX}$  siempre ha descansado en un programa externo llamado *Babel* para tomar esas decisiones. El problema estaba en que, también históricamente, *Babel* solía tener problemas con lenguajes que utilizan

otros sistemas de escritura.

Esta situación se vio empeorada cuando el soporte *unicode* llegó al sistema, lo que llevó a que surgiera un proyecto alternativo llamado *Poliglosia*. De hecho, en documentos que utilizan las fuentes del sistema L<sup>A</sup>T<sub>E</sub>X utiliza por defecto *Poliglosia* en lugar de *Babel*.

Ahora bien, no solo en las últimas versiones *Babel* ha resuelto casi todos sus problemas: para el castellano esos problemas nunca existieron. Si a esto sumamos el hecho de que el paquete *Spanish* para Babel ofrece muchas más opciones que el de Poliglosia, pues ya vemos el porqué en todo lo que sigue de este libro hablaré solo sobre *Babel*.

#### 4.4.4 Pasar opciones a Babel en L<sup>A</sup>T<sub>E</sub>X/L<sup>A</sup>X

Como ya insinuamos más arriba, quien decide qué nombre tienen las figuras, cuadros, entornos, si se tiene que mantener o no la sangría de primera línea incluso en párrafos que siguen un título y demás cosas es Babel. L<sup>A</sup>X selecciona las opciones por defecto de babel al elegir un idioma para el documento, pero es posible obtener más: de hecho, el paquete «spanish» ofrece innumerables opciones que nos permitirán cambiar fácilmente algunas de estas configuraciones.

NOTA: Las opciones por defecto de babel para el español *son correctas desde el punto de vista ortotipográfico*<sup>7</sup>. Esto también se aplica a otras opciones como mantener el sangrado de primera línea también en párrafos que siguen a un título: el quitarlo es característico de documentos escritos en inglés británico, mientras que en inglés americano se deja... Pero bueno, para gustos los colores y algunas de las opciones ofrecidas por el «spanish» de babel son realmente interesantes.

Para pasar parámetros a Babel desde L<sup>A</sup>X simplemente nos dirigimos al menú Documento Configuración Idioma y, bajo Paquete de idiomas, seleccionamos Personalizado, como se muestra en la figura 4.10.

En el casillero a derecha escribimos la instrucción para babel, que debe ser del tipo:

7. <http://www.tex-tipografia.com/spanish2.html>



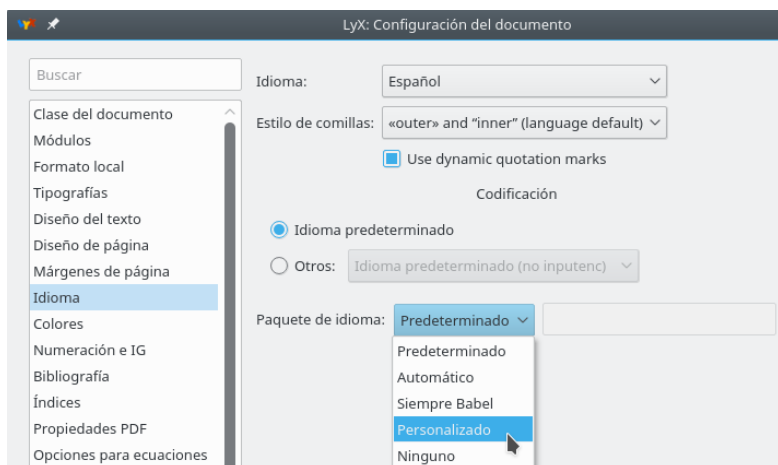


FIGURA 4.10: *Pasar opciones a Babel en LyX*

```
\usepackage[spanish,opción1,opción2]{babel}
```

donde *opción1*, *opción2*... son las opciones que queremos (podemos usar solo una o más de dos). ¿Que cuáles son esas opciones? Puedes encontrar una lista de las más importantes en este enlace:

<http://www.tex-tipografia.com/spanishopt.html>

Por ejemplo, para hacer que no se utilice la sangría de primera línea en párrafos que siguen a títulos y que los números romanos estén escritos en versalitas (para así evitar los horribles números de página en números romanos en minúscula de las páginas del índice), la instrucción sería

```
\usepackage[spanish,es-noindentfirst,
es-ucroman]{babel}
```

También resulta interesante la opción *es-nosectiondot*, que evita los puntos que siguen a la numeración de secciones.

Otra forma de pasar estas opciones es, bajo «paquete de idiomas», seleccionar *Ninguno* y escribir la instrucción directamente en el Preámbulo  $\LaTeX$ .


Dejo al lector el explorar las opciones disponibles.

## 4.5 Importar y exportar documentos

Uno de los grandes mitos de la informática moderna es la «interoperatividad»: la idea de que se puede trabajar sin problemas en distintos sistemas y distintos programas sobre un único documento.

No entraré en discusiones del porqué esta interoperatividad es prácticamente imposible fuera del mundo web (me alejaría demasiado del contenido de este libro), pero la triste realidad es que si trabajamos sobre un documento de texto con un programa y luego queremos pasar a otro *siempre* tendremos problemas.

Pero eso no quiere decir que sea imposible.







Desde la versión 2.2 LyX ofrece integración con el programa Pandoc, el cual nos permite convertir entre distintos tipos de documentos. Con Pandoc y otros filtros ya presentes podemos importar y exportar hacia y desde LyX documentos odt, doc, docx, xhtml... simplemente desde el menú  » Importar / Exportar.

Eso sí, no hay que esperar milagros: si los documentos que estamos mirando son complejos seguramente necesitarán de mucha atención.

Evidentemente importar y exportar documentos L<sup>A</sup>T<sub>E</sub>X funciona a la perfección. De hecho si copiamos al portapapeles un fragmento de un documento L<sup>A</sup>T<sub>E</sub>X tenemos la opción de hacer un «pegado especial» que convertirá todo a formato LyX sin esfuerzo.




## 4.6 Copiar información desde otros programas







Copiar y pegar información entre distintos programas siempre ha sido, es y será problemático y con LyX esto no es una excepción. Pero «difícil» no es sinónimo de «imposible», por lo que he aquí algunos pequeños trucos para simplificar la ingrata tarea.

Pegando con   en LyX solamente obtenemos «texto simple, líneas juntas», lo que significa que además de perder todo el formato cualquier quiebre de párrafo o línea será ignorado. Con    obtendremos «texto simple normal», con quiebres de párrafo. Bajo el menú  » Pegado especial tenemos un par de opciones más, pero más allá de «pegar de L<sup>A</sup>T<sub>E</sub>X» (siempre que copiemos desde un documento L<sup>A</sup>T<sub>E</sub>X, se entiende)

no hay que poner muchas esperanzas en ellas. De hecho copiar imágenes u objetos resulta casi imposible y siempre tendremos que recrearlos.

Algo que *sí* puede copiarse (mayormente) sin problemas son los cuadros, tanto desde documentos HTML mostrados en nuestro navegador de internet (fácil), planillas de cálculo como LibreOffice Calc o Calligra Sheets (igual de fácil) o incluso desde tablas en procesadores de texto como Writer (con algunas vueltas).

Imaginemos que en un documento HTML o en una planilla de cálculo tenemos un cuadro de, digamos, 6 filas por 4 columnas. Luego de copiarlo vamos a LyX y creamos un cuadro vacío de las mismas dimensiones. Finalmente, colocando el cursor en la primera celda de arriba a la izquierda hacemos   . Listo: el formato se perderá, pero el contenido irá en las celdas correctas.

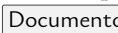
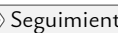




Copiar un cuadro desde Writer tiene más problemas. Como en el caso anterior creamos un cuadro en LyX de las dimensiones correctas. Ahora nos dirigimos a Writer, seleccionamos el cuadro y hacemos   , seleccionando la opción de utilizar tabuladores como separador. Copiamos el texto resultante, vamos a la primera celda del documento LyX y hacemos   .

Paradójicamente copiar desde LyX *hacia* Writer es más simple: Los entornos son automáticamente mapeados a estilos de párrafo apropiados (las secciones se convierten en «Título 2», por ejemplo) mientras que la negrita y la redonda (que algunos llaman *cursiva* o *itálica*, gente extraña) vienen respetadas sin problemas. ¡Incluso los cuadros/tablas se copian correctamente!

Eso sí, los vínculos a las imágenes se copian mal. Nada que no se pueda arreglar con un clic derecho.

## 4.7 Control de versiones

LyX tiene un sistema de control de versiones que permite registrar las diferentes etapas de la redacción de un documento y que puede resultar particularmente útil cuando varias personas trabajan sobre el mismo archivo.

Todo comienza en   , o bien con   . Los cambios se irán indicando con texto subrayado para indicar lo nuevo, tachado para indicar lo borrado, etcétera.

Al activar el control de versiones tendremos una nueva barra de herramientas que nos permitirá aceptar o rechazar cambios, navegar entre ellos y demás.

El sistema es muy simple y está perfectamente explicado en la sección 6.16 de la guía del usuario que se encuentra en [Ayuda](#) » [Guía de usuario](#).

Además, debo admitirlo, no es algo que utilice mucho. Podríamos decir que como escritor soy un «lobo solitario»... y desorganizado.

## 4.8 Trabajar con «ramas»

Supongamos que queremos crear un documento con «variantes» que puedan ser seleccionadas rápidamente tanto para ocultarlas como para mostrarlas en la versión final. El ejemplo más simple es un examen con varias preguntas y sus correspondientes respuestas: queremos que los alumnos solo vean las preguntas, pero también darle a los docentes que deben corregir el examen el documento completo con todo y respuestas.

O quizás tengamos que escribir un manual para un producto que tiene tres modelos similares pero no idénticos, por lo que seleccionando que se muestre solo lo correspondiente a un determinado modelo cuando creamos un PDF obtendremos tres manuales distintos a partir del mismo documento.

LyX nos ofrece una herramienta muy potente para lograr este objetivo: las ramas.

En un documento cualquiera podemos insertar una nueva rama desde el menú [Insertar](#) » [Rama](#) » [Insertar rama nueva](#). Veremos una exclamación (*¡no hay ramas definidas en el documento!*) y la opción para crear la primera, que es la que tenemos que utilizar. Alguna rama tiene que ser la primera.

Si llamamos a nuestra primera rama «Preguntas», luego de aceptar nos encontraremos con algo como lo que muestra la figura 4.11.

¿Nota el lector la «equis» a la izquierda de la etiqueta? Ya volveremos sobre eso.

En la caja roja podemos ahora insertar cualquier tipo de contenido: texto normal, títulos, figuras, cuadros, ecuaciones...

Si volvemos al menú [Insertar](#) » [Rama](#) » [Insertar rama nueva](#) veremos que además de la posibilidad de crear otra rama ha desaparecido la exclamación



FIGURA 4.11: *Nuestra primera rama. Lo del color verde lo discutimos luego.*

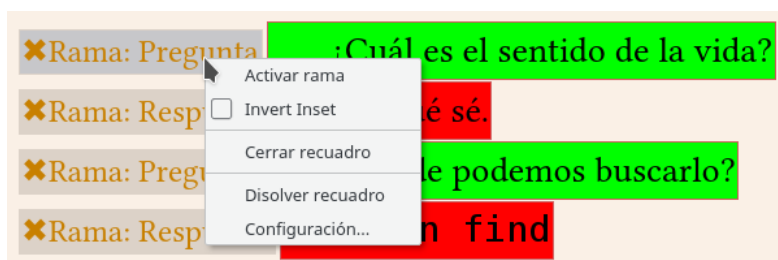


FIGURA 4.12: *Activar una rama*

del principio, ofreciéndonos en su lugar la rama creada anteriormente.

Y así es como funciona la cosa: podemos tener varios bloques separados del documento, todos ellos agrupados bajo la misma rama en modo tal que al ocultar esa rama todos esos bloques se ocultan simultáneamente. Además podemos tener varias ramas que puedan ser ocultadas o mostradas independientemente.

Ahora bien, por defecto las ramas están deshabilitadas por lo que si compilamos el documento no se mostrarán<sup>8</sup>. Esto se muestra con la «equis» a la izquierda de la etiqueta de la cual hablamos antes. Para activar o desactivar una rama es suficiente hacer clic derecho sobre una etiqueta cualquiera de las correspondientes a la rama de interés, como se ve en la figura 4.12.

Ahora la rama Preguntas mostrará un «gancho» a la izquierda indicando que se ha activado y que aparecerá en el documento compilado.

En el menú apenas mostrado tenemos cuatro opciones más:

INVERTIR RECUADRO esta es una novedad en LyX 2.3: la posibilidad de

8. De hecho si lo único que tenemos en el documento son las ramas y todas ellas están deshabilitadas, *el documento no compilará*.

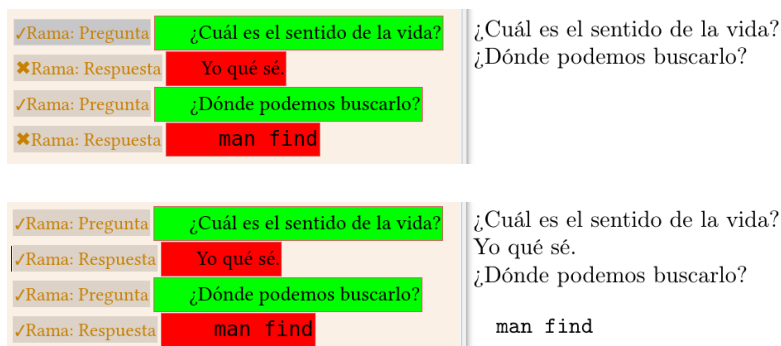


FIGURA 4.13: *Dos ramas en acción con solo una activa (arriba) o ambas activas (abajo). En ambos casos tenemos la ventana de edición de LyX a la izquierda y un fragmento del PDF generado a la derecha.*

«invertir» un cuadro perteneciente a una rama para que se muestre solo cuando la rama está *oculta*

**CERRAR RECUADRO** simplemente oculta el contenido de ese bloque, dejando solo la etiqueta visible

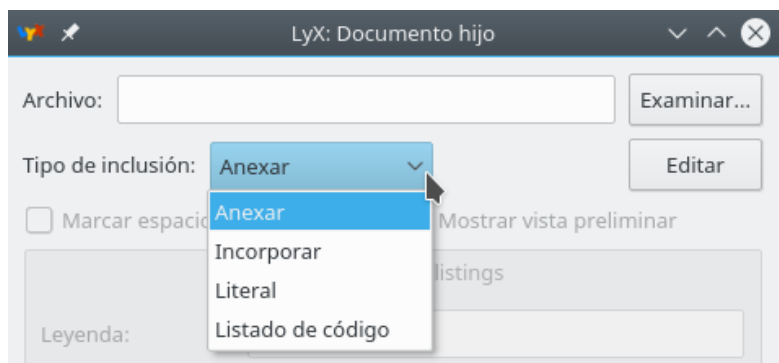
**DISOLVER RECUADRO** sirve para pasar el contenido de ese bloque al documento de base, quitándolo de la rama

**CONFIGURACIÓN** nos lleva a un simple menú que sirve para asignar el bloque a otra rama.

Otra forma de controlar las ramas, no solo para activarlas o desactivarlas sino también para unirlas, cambiar nombre, asignar un color (solo se mostrará durante la edición)... es dirigirse al menú **Documento** » **Configuración** » **Ramas**.

Y así, con un par de clics, podremos pasar de la situación que se ven en la parte superior de la figura 4.13 (a la izquierda el documento LyX y a la derecha el correspondiente PDF) a lo que se ve debajo de la misma.

En definitiva, una magnífica herramienta para poder irse por las ramas en total tranquilidad.

FIGURA 4.14: *Insertar un «documento hijo»*

## 4.9 Documentos maestros

En LyX resulta increíblemente sencillo el dividir un documento en «subdocumentos» para combinar todo en un «documento maestro». Trabajar en cada documento, realizar referencias cruzadas entre ellos... todo es un paseo. Veamos las opciones que tenemos a disposición.

Primero que nada aclarar que, a diferencia de otras aplicaciones como Writer, no existe una diferencia real entre un «documento maestro» y un «documento hijo»: podríamos incluso tener varios niveles de documentos hijos, con un primer documento que funciona como «maestro», un segundo que es el «primer hijo» y en este un «documento nieto». Lo importante a notar es que la *documentclass* que prevalece será aquella del documento en el nivel «más alto», por lo que si en un subdocumentos utilizamos alguna característica especial (algún módulo, por ejemplo) tenemos que asegurarnos que la misma esté también habilitada en el documento principal.

Bueno, manos a la obra.

En el documento principal, con el cursor en el punto donde queremos insertar el documento «hijo», nos dirigimos a **Insertar** » **Archivo** » **Hijo** para obtener el menú que se ve, parcialmente, en la figura 4.14.

Este menú nos ofrece varias formas de incluir un documento hijo. En la primera, Anexar, el documento será representado en la ventana de edición por un botón centrado en su propio renglón: clic sobre ese botón y se vuelve

a acceder a las propiedades del documento Hijo, donde se tiene el botón Editar que lo abrirá para su edición. Al compilar el documento maestro el archivo así insertado irá a partir de su propia página y luego del mismo se tendrá otro quiebre de página.

Si utilizamos Incorporado el documento irá en la misma página donde fue insertado, iniciando en un renglón nuevo.

Las dos últimas opciones, Literal y Listado de código, introducen el *código* del documento hijo. Las dejo a la curiosidad del lector.

Algo realmente interesante si tenemos todos los documentos abiertos simultáneamente, cada uno en su pestaña, es que la numeración de secciones y capítulos en la ventana de edición de LyX seguirá el orden real del documento completo, dándonos una idea más clara de la estructura del documento terminado.

Hacer una referencia cruzada entre los distintos documentos y subdocumentos es increíblemente simple si tenemos todos los documentos abiertos simultáneamente: *todos* los marcadores estarán disponibles desde *todos* los documentos.

Y eso es todo. No, en serio, no hay más: trabajar con documentos maestros e hijos (o nietos) en LyX es igual de fácil que trabajar en un único archivo que lo tenga todo.

Una pasada.

Hay más información en la sección 3.2 del manual que se encuentra en

Ayuda » Características adicionales. También aquí:

<https://wiki.lyx.org/FAQ/Multidoc>



## Parte III

# Más allá de la configuración básica

*Donde se libera todo el poder de L<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>T<sub>E</sub>X y OpenType desde LyX*

*It's full of stars!*

## Capítulo 5

# Algunos comandos

L<sup>A</sup>T<sub>E</sub>X no puede dar soporte para cada aspecto de L<sup>A</sup>T<sub>E</sub>X en su interfaz gráfica, por lo que si queremos pasar ciertas fronteras del formato no nos quedará otra opción que arremangarnos y comenzar a dar instrucciones en L<sup>A</sup>T<sub>E</sub>X puro y duro, ya sea en el preámbulo o en el cuerpo del documento.

Esto suena más difícil de lo que en realidad es por lo que no hay que entrar en pánico solo porque tendremos que utilizar un poco de código extraño... pero tampoco hay que relajarse demasiado ya que hacer cosas sin comprender cómo estas funcionan es una invitación al desastre.

Por esto invito al lector a recorrer las restantes páginas de este libro con calma, que el camino que nos queda por delante es interesante y las herramientas que nos dejará están llenas de posibilidades.

En este capítulo veremos cómo insertar código en nuestros documentos además de dar un rápido repaso a algunas de esas instrucciones que necesitaremos varias veces a lo largo del texto.

Comencemos.

### 5.1 El preámbulo L<sup>A</sup>T<sub>E</sub>X

Ya hemos hablado sobre la importancia del preámbulo L<sup>A</sup>T<sub>E</sub>X en la introducción de este libro: es el lugar que nos permite definir cosas que podrán ser utilizadas en todo el documento.

Accedemos a él desde  Documento > Configuración > preámbulo L<sup>A</sup>T<sub>E</sub>X y

allí podremos escribir todo el código que veremos en los próximos capítulos.

Un tipo de instrucción que se repetirá muchas veces es la que nos permite activar un paquete de nuestra distribución L<sup>A</sup>T<sub>E</sub>X. Por ejemplo, para poder insertar el «logo» de X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X he debido activar el paquete `hologo`, para lo cual he escrito en el preámbulo

```
\usepackage{hologo}
```

Como puede verse se utiliza una instrucción con un claro nombre (`\usepackage`) para decir que queremos utilizar un paquete particular (`hologo`).

Algunos de estos paquetes ofrecen la posibilidad de ajustar distintos «parámetros» para configurar su comportamiento. Esto puede hacerse directamente al llamar el paquete o mediante instrucciones especiales que estarán disponibles solo si el paquete ha sido activado mientras que otros paquetes ofrecen, o incluso exigen, *ambas* posibilidades. En algunos casos tendremos que «parchear» un paquete modificándolo con el comando `\renewcommand` e incluso nos veremos empujados a utilizar instrucciones lógicas para determinar cuándo (o si) un cambio debe efectuarse.

En fin, muchas opciones que iremos viendo en las siguientes páginas.

Desafortunadamente no todas las instrucciones son tan claras: por ejemplo `\makeatletter` puede fácilmente confundir a quien «no esté en tema».

Simplificando un poco las cosas, el carácter `@` (*at*) es especial en L<sup>A</sup>T<sub>E</sub>X... pero solo a veces. Cuando se redefine un comando que utiliza `@` en su nombre `\makeatletter` (*make @ a letter*) transforma `@` en un carácter «normal» mientras que `\makeatother` (*make @ other*) lo «vuelve a lo que L<sup>A</sup>T<sub>E</sub>X dice que es».

En fin, nada que deba preocuparnos mucho ya que no iremos *tan* profundo en estas cosas, pero como pueden ver estamos ante un mundo complejo en el que es necesario moverse con pie firme.

## 5.2 La caja T<sub>E</sub>X

Ya sea con el correspondiente botón en la barra de herramientas o simplemente presionando la combinación de teclas Ctrl L nos encontraremos frente a una caja roja a la espera de código.

Para seguir con el ejemplo de `hologo`, para hacer que aparezca el logo de  $\text{\LaTeX}$  podríamos escribir dentro de una de estas cajas el comando

```
\hologo{XeTeX}
```

lo cual en la ventana de edición de  $\text{\LaTeX}$  se vería más o menos así:

```
\hologo{XeTeX}
```

NOTA: En este libro he utilizado un estilo de carácter en lugar de escribir cada vez esta instrucción. Volveremos sobre esto en el capítulo 8 y en el apéndice B.

Las cajas pueden aceptar varias líneas de código y todas las instrucciones que necesitemos, por lo que no estamos limitados a instrucciones «pequeñas».

### 5.3 Una cuestión de «tamaño»

Existen instrucciones  $\text{\LaTeX}$  para definir el tamaño del texto. Estas instrucciones pueden verse, ordenadas desde el valor más pequeño al mayor, en el cuadro 5.1 donde se presentan los valores del tamaño para las diferentes clases de documentos y para algunos valores del tamaño de texto de base<sup>1</sup>.

Como puede verse estos tamaños son fijos, proporcionales al tamaño de base definido para el documento, por lo que si bien tenemos una amplia *variedad* que cubrirá casi todas las situaciones imaginables no nos da una *libertad* absoluta. Para obtener tamaños de fuente arbitrarios existe el comando `\fontsize{tamaño-fuente}{altura-línea}`, el cual es muy sencillo de utilizar:

```
{\fontsize{5cm}{5.5cm} ¡Un texto muy grande!}
```

nos dará una fuente de 5 cm con un interlineado de 5,5 cm.

Para conocer más al respecto,

[https://en.wikibooks.org/wiki/LaTeX/Fonts#Sizing\\_text](https://en.wikibooks.org/wiki/LaTeX/Fonts#Sizing_text)

1. Un detalle técnico:  $\text{\LaTeX}$  usa una definición de «punto» (el «punto  $\text{\TeX}$ ») diferente a la estándar (o «punto PostScript»). La diferencia es mínima, menor al 0,4%, pero existe: [https://en.wikipedia.org/wiki/Point\\_\(typography\)#American\\_points](https://en.wikipedia.org/wiki/Point_(typography)#American_points)

Tamaño	Clases estándar, <i>beamer</i>			Clases $\mathcal{A}\mathcal{M}\mathcal{S}$ , <i>memoir</i>		
	[10pt]	[11pt]	[12pt]	[10pt]	[11pt]	[12pt]
<code>\tiny</code>	5	6	6	6	7	8
<code>\scriptsize</code>	7	8	8	7	8	9
<code>\footnotesize</code>	8	9	10	8	9	10
<code>\small</code>	9	10	10.95	9	10	10.95
<code>\normalsize</code>	10	10.95	12	10	10.95	12
<code>\large</code>	12	12	14.4	10.95	12	14.4
<code>\Large</code>	14.4	14.4	17.28	12	14.4	17.28
<code>\LARGE</code>	17.28	17.28	20.74	14.4	17.28	20.74
<code>\huge</code>	20.74	20.74	24.88	17.28	20.74	24.88
<code>\Huge</code>	24.88	24.88	24.88	20.74	24.88	24.88

CUADRO 5.1: Valores usados por las distintas instrucciones en función del tamaño de base del texto. En esta tabla se ha ignorado la clase para generar transparencias (slides). Fuente: [https://en.wikibooks.org/wiki/LaTeX/Fonts#Sizing\\_text](https://en.wikibooks.org/wiki/LaTeX/Fonts#Sizing_text)

## 5.4 Una cuestión de «forma» (tipográfica)

Hablaremos largo y tendido sobre tipografía en el próximo capítulo, pero en ocasiones no es necesario entrar en tanto detalle, muchas veces alcanza (y sobra) con indicar las *variantes* de las fuentes predefinidas que se desean. Esto nos será de mucha utilidad en el capítulo 11.

Como ya hemos visto en un documento  $\text{\LaTeX}$  se tienen tres «familias» de fuentes tipográficas: con gracias, también llamada «romana» (*roman*), sin gracias, también llamada palo seco (*sans serif*) y monoespaciada, también llamada «teletipo» (*teletype*).

Cada una de estas familias de fuentes tienen un «peso» que viene indicado mediante una «serie» (negrita, seminegrita y ligera) o una «forma» (normal, itálica, inclinada, versalitas, etcétera).

Cada una de estas familias y cada una de estas formas puede ser llamada con una instrucción  $\text{\LaTeX}$  determinada. Así tenemos tres tipos de instrucciones,

`\XYfamily`  
`\XYseries`  
`\XYshape`

donde XY son dos caracteres que representan la característica buscada.

Las familias son representadas por

- `rm` para romana: `\rmfamily`
- `sf` para sans serif: `\sffamily`
- `tt` para monoespaciada: `\ttfamily`

Las «series» son representadas por

- `bf` para negrita: `\bfseries`
- `md` para «mediana»: `\mdseries`
- `lf` para «ligera»: `\lfseries`

Por último, las «formas» son representadas (no listaremos todas) por

- `it` para itálica: `\itshape`
- `sl` para «inclinada» (*slanted*): `\slshape`
- `sc` para versalitas (Small Caps): `\scshape`
- etcétera

Para una lista completa de estos comandos:

[https://en.wikibooks.org/wiki/LaTeX/Fonts#Font\\_styles](https://en.wikibooks.org/wiki/LaTeX/Fonts#Font_styles)

## 5.5 ¡Números!

Sin duda se nos presentará la situación de tener que elegir entre numerar con números «normales», romanos o con letras.

Cada tipo de numeración está asociada a una instrucción:



- `arabic` da los números arábigos de toda la vida: 1, 2, 3, 4...

- Roman da números romanos en mayúsculas: I, II, III, IV...
- roman da números romanos en minúsculas: i, ii, iii, iv... (¡si los vieran los romanos!)
- alph da letras en minúsculas: a, b, c, d...
- Alph da letras en mayúscula: A, B, C, D...

Ya veremos en los ejemplos más adelante cómo se utilizan estas instrucciones.

## 5.6 ¡Reconfigurar!

En los próximos capítulos nos encontraremos con varios paquetes  $\text{\LaTeX}$  que necesitaremos para llevar adelante todos los «trucos» que serán presentados. Muchos de esos paquetes vienen instalados en forma predefinida en las distribuciones  $\text{\LaTeX}$ , pero no todos.

Ahora bien, no es suficiente instalar un paquete para poder utilizarlo desde  $\text{\LyX}$ : tenemos que avisar a este último que el paquete ha sido instalado. Para esto, luego de modificar cualquier cosa en nuestra instalación  $\text{\LaTeX}$  debemos siempre dirigirnos a  , dar unos segundos para que  $\text{\LyX}$  mire bien cómo están las cosas y reiniciar el programa cuando nos pida amablemente de hacerlo.



## Capítulo 6

# L<sup>A</sup>T<sub>E</sub>X, OpenType y la tipografía

Antes de comenzar, una nota importante: en este capítulo y en los que sigan haremos uso extensivo de las «cajas T<sub>E</sub>X» y del preámbulo L<sup>A</sup>T<sub>E</sub>X ya que no nos quedará alternativa que utilizar código directo. Pero a no preocuparse, que no será difícil. De hecho, más adelante en este libro (capítulo 8) veremos cómo simplificar todo aún más con la creación de estilos.

Pues bien, al tema de este capítulo: ya tenemos definida una fuente OpenType para nuestro documento, pero queremos ir más allá de las opciones por defecto de la tipografía elegida y activar algunas «especialidades».

Esto puede realizarse en diferentes niveles:

1. Pocos y pequeños fragmentos de texto
2. Muchos y quizás no tan pequeños fragmentos de texto
3. Todo el documento
4. Las fórmulas matemáticas

En este capítulo hablaré de estas posibilidades, pero también de algunos temas más generales: interlineado, separación silábica y microtipografía.

Para utilizar OpenType debemos conocer cómo llamar a cada una de sus tablas de sustitución. En lo que sigue de este capítulo y del resto del libro tomaré como referencia la página de Dario Taraborelli:

<http://nitens.org/taraborelli/TeXOpenType>

y la página de la wikipedia

[https://en.wikipedia.org/wiki/List\\_of\\_typographic\\_features](https://en.wikipedia.org/wiki/List_of_typographic_features)

A modo de ejemplo, en el cuadro 6.1 encontramos algunas de las etiquetas principales. Es importante notar que no todas las fuentes ofrecen todas estas opciones. Además algunas de las más comunes estarán activas por defecto (como las ligaduras tipográficas) mientras que otras tienen que activarse manualmente.

## 6.1 OpenType para la fuente predefinida

Supongamos que queremos activar una opción especial de la fuente predefinida en un punto particular de nuestro documento. Por ejemplo, y para impresionar a nuestros lectores, podríamos querer escribir un breve texto utilizando *ligaduras tipográficas históricas* o quizás escribir fracciones del tipo  $12\frac{3}{37}$  sin utilizar el editor de ecuaciones.

Como ya hemos dicho, más adelante veremos cómo definir estilos de carácter para así simplificar el uso de estas instrucciones, pero todavía tenemos que aprender un par de cosas por lo que por ahora seremos rudos y escribiremos código, ¡sí!

Tenemos que combinar la instrucción `\addfontfeature` con el «nombre OpenType» de la característica que necesitamos para lo cual podemos utilizar utilizar dos «cajas T<sub>E</sub>X» rodeando el texto a modificar. Por ejemplo, las ligaduras históricas pueden llamarse con

```
\addfontfeature{RawFeature={+hlig}} históricas }
```

(el signo «+» activa la opción «hlig») o bien con

```
\addfontfeature{Ligatures=Historic} históricas }
```

que obtendremos el mismo resultado: *históricas*.

La llave que abre la primera «caja T<sub>E</sub>X» y la solitaria en la segunda son importantes ya que nos aseguran que el código será aplicado solamente al texto que nos interesa y no a todo lo que sigue.

Ciertamente es posible activar varias sustituciones simultáneamente: solo hay que separarlas con comas. Por ejemplo

```
\addfontfeature{RawFeature={+hlig,+onum}}
```

activará simultáneamente las ligaduras históricas y los numerales «estilo antiguo».

Característica	Etiqueta	Nombre en Fontspec
Números		
En «mayúsculas»	lnum	Numbers={Lining}
Estilo antiguo	onum	Numbers={Oldstyle}
Proporcionales	pnum	Numbers={Proportional}
Tabulares	tnum	Numbers={Monospaced}
Cero «tachado»	zero	Numbers={SlashedZero}
Ligaduras		
Estándar	liga	Ligatures=Common
Contextuales	clig	Ligatures=Contextual
Discrecionales	dlig	Ligatures=Rare
Históricas	hlig	Ligatures=Historical
Formas, estilos, posición y variantes		
Versalitas	smcp	Letters=SmallCaps
Superíndice	sup	VerticalPosition=Superior
Subíndice	sub	VerticalPosition=Inferior
Inferiores científicos	sinf	VerticalPosition=ScientificInferior
Ordinales	ordn	VerticalPosition=Ordinal
Sets estilísticos	ss01...ss20	StylisticSet=1 ... 20
Alternativas estilísticas	salt	Style=Alternate
«Swash»	swsh	Style=Swash
Fracciones		
Numerador	numr	VerticalPosition=Numerator
Denominador	dnom	VerticalPosition=Denominator
Fracciones	frac	Fractions=0n
Fracciones alternativas	afrc	Fractions=Alternate

CUADRO 6.1: *Algunas etiquetas OpenType y sus usos*

Es importante notar que estas instrucciones también pueden utilizarse para *deshabilitar características habilitadas por defecto*. Todo está en la página de Dario Taraborelli comentada más arriba, pero por ejemplo podemos deshabilitar las «ligaduras contextuales», ya sea con

```
\addfontfeature{Ligatures=NoContextual}
```

o con

```
\addfontfeature{RawFeature={-clig}}
```

(el signo «-» *desactiva* la opción «clig»).

## 6.2 Definir una familia de fuentes tipográficas

Lo que contamos en la sección anterior está (relativamente) bien solo si lo tenemos que hacer un par de veces: escribir todo ese código una y otra vez es ciertamente molesto.

Por esto resulta conveniente definir una «familia de fuentes» que tenga la característica deseada. Además, no estaremos limitados a la tipografía predefinida del documento.

Supongamos que queremos definir una nueva fuente a utilizar en nuestro documento, por ejemplo EB Garamond con las ligaduras históricas activadas. En el menú Documento Configuración Preámbulo LaTeX escribimos:

```
\newfontfamily{\storiclig}[RawFeature={+hlig}]
{EB Garamond}
```

Para utilizar esta definición hacemos algo parecido a lo comentado en la sección anterior (claramente, escribiendo menos): dos «cajas TeX» rodeando el texto a modificar:

```
{\storiclig}{históricas}}
```

Nótese la doble llave de cierre.

Y bien, esto es más sencillo que lo anterior... pero aún no resulta ideal. A no preocuparse, insisto: volveremos a esto en el capítulo 8 donde la cosa se volverá casi idílica.


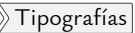

## 6.3 OpenType para todo el documento

Supongamos que queremos un documento que utilice la fuente tipográfica EB Garamond, pero que no nos gustan los numerales «estilo antiguo» que esta fuente activa por defecto y queremos utilizar los números «normales», los que son todos de la misma altura. Pues bien, antes de hacer esto tenemos que considerar algunas cosas.

En un documento L<sup>A</sup>T<sub>E</sub>X/X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X se definen tres tipos de fuentes principales: la, valga la redundancia, «principal», (llamada «Romana» en L<sup>A</sup>T<sub>E</sub>X) una fuente palo seco (sans serif) y una fuente monoespaciada.

Difícilmente necesitemos características OpenType en una fuente monoespaciada por lo que concentrémonos en las otras dos. Si bien L<sup>A</sup>T<sub>E</sub>X nos oculta estas dificultades, la fuente principal se define en el preámbulo del documento con la instrucción `\setmainfont` mientras que la fuente palo seco se define con la instrucción `\setsansfont`.

La razón para dar estas funciones ahora es que como L<sup>A</sup>T<sub>E</sub>X las oculta no nos da la oportunidad de modificar sus parámetros. Pues bien, que tendremos que usarlas explícitamente.

En L<sup>A</sup>T<sub>E</sub>X, bajo  Configuración  Tipografías marcamos «Usar tipografías no Tex (vía XeTeX/LuaTeX)», *pero no seleccionamos las fuentes allí, dejamos todo en Predeterminado*: vamos a indicar las fuentes manualmente en el preámbulo. En ese mismo menú, bajo  Salidas, seleccionamos PDF(XeTeX) para «Formato de salida». Finalmente en el «Preámbulo L<sup>A</sup>T<sub>E</sub>X» escribimos:

```
\setmainfont{EB Garamond}[
    Mapping=tex-text,
    RawFeature={+lnum}
]
```

Si queremos otra característica en lugar de los números «en mayúsculas» simplemente cambiamos `+lnum` por lo que corresponda dentro de las llaves que siguen a «RawFeature». Si queremos varias características simultáneamente, las escribimos todas dentro de esas mismas llaves y separadas por comas.

Y por supuesto, si queremos otra fuente, pues escribimos su nombre en lugar de EB Garamond.

Ahora bien, ¿qué cuernos es ese `Mapping=tex-text`? Buena pregunta. Básicamente esa línea habilita algunas «ligaduras T<sub>E</sub>X» como aquellas de las que hablamos en el capítulo 4: hacer que tres guiones seguidos se conviertan automáticamente en una raya o cosas por el estilo. Ahora bien, como ya vimos L<sub>A</sub>T<sub>E</sub>X hace varias de esas conversiones por sí mismo antes de enviar el documento a compilar por lo que la línea en cuestión podría no ser realmente necesaria. Pero ya saben, por motivos históricos y para aquellos que utilizan opciones de configuración extrañas.

La instrucción `\setsansfont` se usa de la misma forma.

## 6.4 Distintas fuentes para distintas «formas»

Vayamos directamente a un ejemplo: la fuente *Gentium Plus* es muy completa, pero ofrece solo la forma «normal» y la cursiva, no nos da negrita o negrita cursiva. La fuente *Gentium Basic* nos da las cuatro formas básicas, pero no es tan completa. Ambas fuentes tienen el mismo diseño general, ¿cómo hacemos entonces para que X<sub>E</sub>L<sub>A</sub>T<sub>E</sub>X utilice automáticamente *Gentium Plus* para el texto normal y la cursiva mientras que para la negrita y negrita cursiva utilice *Gentium Basic*?

```
\setmainfont[BoldFont = Gentium Basic Bold,
              ItalicFont = Gentium Plus Italic,
              BoldItalicFont = Gentium Basic
                          Bold Italic,
              Mapping = tex-text]{Gentium Plus}
```

## 6.5 Distintas opciones OpenType para cada «forma»

Supongamos que queremos activar algunas características OpenType para la fuente normal, otras características para la cursiva, otras para la negrita... Para esto, a las instrucciones introducidas en la sección anterior debemos agregar un poco de código.

Para dejar las cosas en claro desarrollemos un ejemplo completamente arbitrario y absurdo donde para la fuente *Vollkorn* activamos el primer set estilístico solo para el texto normal, el set estilístico 17 solo para la cursiva, el 14 solo para la negrita y la opción «zero» (el cero tachado) solo para negrita cursiva:

```
\setmainfont{Vollkorn}[
  UprightFeatures={
    RawFeature=+ss01,
  },
  ItalicFeatures={
    RawFeature=+ss17,
  },
  BoldFeatures={
    RawFeature=+ss14,
  },
  BoldItalicFeatures={
    RawFeature=+zero,
  },
]
```

¡Atención a las comas!

Si queremos agregar más opciones para cada tipo, simplemente se utilizan más líneas y más comas:

```
ItalicFeatures={
  RawFeature=+hlig,
  RawFeature=+swsh,
  Numbers=OldStyle,
},
```

Incluso podríamos decidir que determinada forma tenga un *color* diferente... pero para eso dejaré al lector el explorar el capítulo 6 de la documentación de fontspec la cual se instala en

`/usr/share/texmf/doc/latex/fontspec/fontspec.pdf`

## 6.6 Aplicar un set estilístico a un carácter particular

Y en caso lo comentado en las secciones anteriores no resulte suficiente para apreciar el poder de  $\text{\LaTeX}$  ahora presentaremos cómo aplicar un set estilístico a *un solo carácter*.

Pongámonos en situación. La fuente tipográfica Vollkorn de la cual hablamos en la sección anterior ofrece una variación de la Q mayúscula con una «cola larga»:

Qué Más Ninguno (Normal)

Qué Más Ninguno (+ss01)

El «problema», como puede verse, es que ese cambio viene con otros: la «a», la «g», la «M» y la «N». La pregunta surge entonces espontáneamente (al menos *para mi* surge espontáneamente): ¿es posible aplicar el primer set estilístico *solo para la letra Q*?

¡Por supuesto!

La inspiración viene del siguiente artículo donde aplican un set estilístico a todas las mayúsculas:

<http://tex.stackexchange.com/questions/33855/can-one-set-a-specific-font-variant-in-xelatex-for-uppercase-letters-only>

En el preámbulo  $\text{\LaTeX}$  tenemos que escribir:

```
\newfontfamily{\upfont}[RawFeature={+ss01}]{Vollkorn}
\XeTeXinterchartokenstate=1
\newXeTeXintercharclass \uppercaseclass
\makeatletter
\@tempcnta=`\Q
\XeTeXcharclass \@tempcnta \uppercaseclass
\makeatother
\XeTeXinterchartoks 0 \uppercaseclass =
    {\begingroup\upfont}
\XeTeXinterchartoks \uppercaseclass 0 =
    {\endgroup}
\XeTeXinterchartoks 255 \uppercaseclass =
    {\begingroup\upfont}
```



```
\XeTeXinterchartoks \uppercaseclass 255 =
{\endgroup}
```

¡Todo listo!

## 6.7 OpenType para las fórmulas matemáticas 1: el paquete **unicode-math**

Para completar la migración a fuentes OpenType, por supuesto falta hacer que las fuentes utilizadas en las fórmulas matemáticas también sean cambiadas. Esto es simple de lograr en LyX gracias al paquete `unicode-math` y a varios proyectos tipográficos que ofrecen fuentes especialmente diseñadas para este propósito como

- el proyecto STIX:
  - <http://stixfonts.org/>
- Libertinus Math:
  - <https://github.com/khaledhosny/libertinus>
- Las fuentes TeX Gyre Pagella y Termes:
  - <http://www.gust.org.pl/projects/e-foundry/tex-gyre>
- Las fuentes Latin Modern:
  - <http://www.gust.org.pl/projects/e-foundry/latin-modern>
- Asana Math:
  - <http://www.ctan.org/tex-archive/fonts/Asana-Math/>
- DejaVu Math TeX Gyre:
  - <https://dejavu-fonts.github.io/>
- ... y no muchas más, pero con estas tendremos bastante.

Para utilizar esto, a partir de la versión 2.1 de  $\text{L}\text{\AA}\text{X}$  es necesario realizar dos pequeños pasos, uno gráfico y el otro «de código». Primero tenemos que dirigirnos a `Documento`  $\gg$  `Configuración`  $\gg$  `Tipografías`  $\gg$  `Ecuaciones` donde en el menú desplegable es necesario seleccionar la críptica opción «Tipografías no- $\text{T}\text{E}\text{X}$  predeterminadas». Ahora vamos a `Documento`  $\gg$  `Configuración`  $\gg$  `Preámbulo  $\text{L}\text{a}\text{T}\text{E}\text{X}$`  y escribimos lo siguiente para utilizar las fuentes STIX Two:

```
\setmathfont{STIX Two Math}
```

Si en cambio queremos utilizar las fuentes Libertinus, pues

```
\setmathfont{Libertinus Math}
```

O con DejaVu (a alguien podría gustarle...)

```
\setmathfont{DejaVu Math TeX Gyre}
```

Si alguna de estas fuentes ofrecen «sets estilísticos» u otras especialidades OpenType podemos activarlas al momento de llamar la fuente. Por ejemplo para utilizar los glifos alternativos de algunas letras y obtener integrales más «verticales» con STIX Two Math (set estilísticos 2 y 8, respectivamente, consulten la documentación de la fuente para más información) podemos escribir

```
\setmathfont{STIX Two Math}[RawFeature={+ss02;+ss08}]
```

Otra cosa que podemos hacer es elegir distintas fuentes para diferentes «rangos». Por ejemplo podríamos insistir de utilizar Latin Modern Math para los caracteres «caligráficos» aún cuando la fuente matemática principal es otra. Simplemente debemos escribir

```
\setmathfont[range={\mathcal,
\mathbfcal}]{Latin Modern Math}
```

Eso sí, haciendo esto *podrían* presentarse problemas de espaciado entre los caracteres de la fórmula. Para resolver esto podemos agregar otra línea estableciendo un «rango vacío» para la fuente matemática de base:

```
\setmathfont[range={}]{STIX Two Math}
```

Listo, incluso nuestras ecuaciones utilizarán fuentes OpenType.

Eso sí, ¡recuerden de instalar la fuente antes!

En Linux y con TeXLive, la documentación de `unicode-math` se instala en

```
/usr/share/texmf/doc/latex/unicode-math/unicode-math.pdf
```

Existe otra forma de seleccionar la fuente tipográfica utilizada en las fórmulas, la cual será presentada en la siguiente sección.

## 6.8 OpenType para las fórmulas matemáticas, 2: el paquete **mathspec**

El paquete `Unicode-math` presentado en la sección anterior tiene una pequeña limitación: el número de fuentes «matemáticas» disponibles no es ciertamente abundante.




Existe sin embargo una forma de utilizar *cualquier* fuente del sistema para escribir nuestras fórmulas: el paquete `mathspec`.

Con este paquete los símbolos (integrales, paréntesis, un  $\pm$ ) provienen de las fuentes estándar L<sup>A</sup>T<sub>E</sub>X, pero tanto las variables como el nombre de las funciones y las letras griegas provienen de otra tipografía elegida por nosotros.

Utilizar `mathspec` en LyX tiene sus vueltas, pero conociendo el secreto no es un verdadero problema. Veamos como aprovecharnos de esta gran herramienta.

Primero que nada, es importante notar que `mathspec` llama al paquete `fontspec` que es el que nos permite configurar las fuentes en X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X, por lo tanto, si activamos la opción que ofrece LyX de «utilizar fuentes no TeX» (opción que se dedica a configurar el paquete `fontspec`) junto con `mathspec` pues que tendremos una larga serie de errores al tratar de compilar el documento ya que estaríamos llamando a `fontspec` dos veces. ¿Solución? *Olvidarnos de las opciones que LyX nos ofrece para configurar las fuentes tipográficas y hacer todo «a mano» en el preámbulo.*

En [Documento]»[Configuración]»[Tipografías] dejamos SIN marcar la opción «Usar tipografías no TeX (via XeTeX/LuaTeX)», asegurándonos que

*todas* las opciones estén en Predeterminado/Automático. Luego, en   Salidas  Formato de salida elegimos de utilizar PDF (XeTeX) (con LuaTeX también debería funcionar).

Ahora, el Preámbulo  $\LaTeX$ . Para este ejemplo utilizaremos la fuente Cochineal<sup>1</sup>, pero puede ser cualquier otra:

```
\usepackage{mathspec}
\setmathfont{Digits, Latin}[] {Cochineal}
\setmathfont{Greek} [Uppercase=Plain,
                    Lowercase=Regular] {Cochineal}

\setmainfont[Mapping=tex-text] {Cochineal}
\setsansfont[Mapping=tex-text]
    {Libertine Sans}
\setmonofont{DejaVu Sans Mono}
```

La primera línea dice de utilizar el paquete `mathspec`. La segunda y tercera líneas configuran la fuente a utilizar en las expresiones matemáticas: en la segunda línea simplemente decimos de utilizar Cochineal para los números y el texto (variables, nombres de funciones, constantes, etcétera) mientras que en la tercera no solo decimos de dónde tomar los caracteres griegos, también pasamos algunas «opciones» como hacer que las letras griegas en minúscula no se muestren en bastardilla (la opción `Lowercase=Regular...` no es que me guste mucho, pero ya saben, para dar un ejemplo).

Las últimas tres líneas solo definen la fuente principal, la fuente «sans» y la monoespaciada, tal y como alguna vez comentamos que podía hacerse.

Ciertamente el lector es libre de elegir sus tipografías preferidas para cada opción, que el ejemplo aquí presentado es un tanto ecléctico.<sup>2</sup>

En Linux y con TeXLive, la documentación de `mathspec` se instala en `/usr/share/texmf/doc/xelatex/mathspec/mathspec.pdf`

Fuente de inspiración para esta sección:

<http://tex.stackexchange.com/questions/29100/using-old-style-numerals-in-xelatex-in-math-mode>

1. <http://www.ctan.org/pkg/cochineal>

2. A esta altura algunos lectores podrían comenzar a sospechar que elijo muchos de mis ejemplos en modo tal que no se puedan simplemente «copiar y pegar», que necesiten ser editados y personalizados antes de que resulten útiles en aplicaciones reales. Y eso.

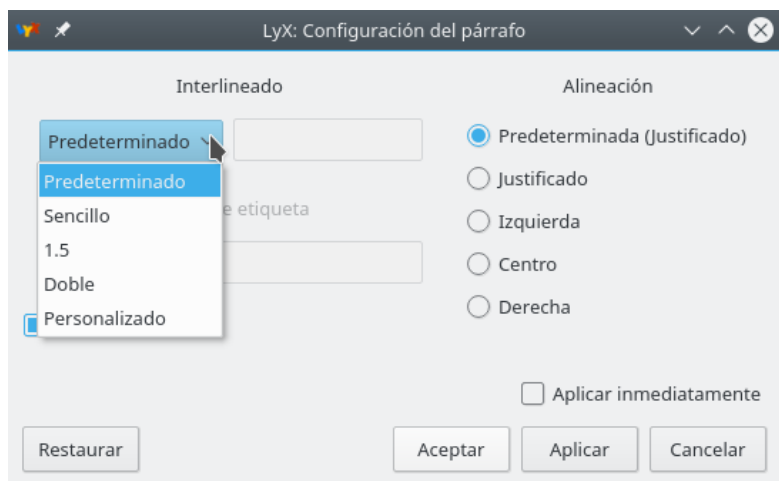


FIGURA 6.1: *Seleccionar el interlineado en un párrafo particular*

## 6.9 Interlineado

He aquí un dilema: tanto para el documento como un todo como para un párrafo particular, LyX ofrece varias opciones de interlineado. Las tres principales («Sencillo», «Uno y medio» en la configuración del documento o «1.5» en la del párrafo —esta doble denominación resulta importante— y «Doble») parecen lo suficientemente claras y comprensibles, pero solo hasta que nos damos cuenta de que carecemos de la menor idea del significado de «Sencillo».

La cuestión no es menor y ha dado y dará muchos dolores de cabeza a quienes utilizan L<sup>A</sup>T<sub>E</sub>X.

Hagamos un par de pruebas para comprender cómo funciona todo esto. Si sobre un párrafo hacemos clic derecho » Configuración del párrafo obtendremos el menú presentado en la figura 6.1.

Bien, prueba, lector, lo siguiente: crear un texto en «1.5» y otro con espacio personalizado de 1,5 (¡atención a la coma!). ¿Puedes ver lo mismo? ¿Verdad que no? ¡Personalizado de 1,5 nos da líneas más separadas que la opción «1.5»! Ahora vuelve a la configuración del párrafo, selecciona nuevamente Personalizado y escribe 1,25: verás cómo al presionar «Aplicar» la

opción cambia por sí sola a «1.5». Por último selecciona nuevamente Personalizado y escribe 1,667: ahora pasa a «doble». ¿Y esto?

Las fuentes tipográficas modernas incluyen la definición de qué es una «línea sencilla»: quien diseña la fuente elige cuál es el espacio mínimo entre líneas para que estas no se «choquen» unas a otras. Este tamaño depende del diseño de los caracteres (su altura, la forma en la que se apilan los acentos —sí, hay idiomas en los que una misma letra puede tener varios acentos simultáneamente—), las preferencias personales del autor... Procesadores de texto como Writer utilizan este parámetro interno de la fuente como base del «interlineado».

Ahora bien, lo que sucede con  $\text{\LaTeX}$  es que este sistema fue creado décadas antes de unicode, OpenType y la mar en coche. Y es por esto que  $\text{\LaTeX}$  calcula la distancia entre líneas *a su modo*.

La cosa va más o menos así:  $\text{\LaTeX}$  define la altura «básica» de la línea como el tamaño de la fuente. Es decir, una fuente de 10 puntos tendrá una altura de línea de 10 puntos y por esto «uno y medio» (o «1.5», según sea el menú que miramos) será de 15 mientras que la doble será de 20... PERO, ¡el interlineado sencillo no será de 10, sino de 12!

Efectivamente, dado que tener una distancia entre líneas igual al tamaño de la fuente dará un texto demasiado compacto e ilegible donde las letras más altas de una línea chocarán con las más bajas de la anterior,  $\text{\LaTeX}$  agrega automáticamente un 20% de esa altura de base, ¡pero solo al interlineado sencillo, el «uno y medio» y el doble se calculan *sin* ese aumento! Y por si fuera poco ¡el interlineado personalizado es un factor sobre el interlineado sencillo, el que ha sido aumentado!

Ahora te pediré, lector, que hagas un esfuerzo por recordar cuando en el colegio trabajabas con proporciones y porcentajes. Si sabemos que una cantidad (la distancia entre dos líneas con «interlineado sencillo») es el 120% de una incógnita y queremos calcular el 150% de esa incógnita, ¿cuál es el factor que nos queda? Pues

$$\frac{1}{1,2} \cdot 1,5 = 1,25$$

Y si lo que queremos es tener el doble de la distancia de base, pues

$$\frac{1}{1,2} \cdot 2 \approx 1,667$$

Pues ya digo yo que  $\text{\LaTeX}$  puede ser extraño...

Como puede verse la confusión nace del hecho de que las opciones «uno

y medio» (o «1.5») y «doble» están definidas sobre un valor diferente al valor utilizado para definir «personalizado».

Y antes de que alguien lo pregunte, pues sí, esto también afecta a X<sub>Y</sub>TeX y a LuaTeX.

## 6.10 Separación silábica

Cuando en un documento hacemos que el texto esté «justificado» (hacer que ambos márgenes, izquierdo y derecho, formen una línea) nos enfrentamos a un problema: si el texto está en columnas o la fuente tipográfica usada es grande en modo tal de tener pocas palabras por línea, el software que usemos se verá obligado a variar la distancia entre las palabras para ajustarse a los márgenes, lo que dará espacios en blanco de tamaño impredecible que harán incómoda la lectura. Para evitar esto se recurre a la separación silábica de palabras a final de línea en modo tal que se reduzcan esos espacios... pagando el precio de tener muchas palabras «cortadas».

En general L<sup>A</sup>T<sub>E</sub>X hace un gran trabajo equilibrando los espacios con la separación silábica, pero solo cuando el texto es «lo suficientemente ancho». ¿Es posible cambiar manualmente el comportamiento de L<sup>A</sup>T<sub>E</sub>X? Sí, aunque lo que hay que hacer es más cercano a la «magia negra» que a un método de trabajo.

L<sup>A</sup>T<sub>E</sub>X usa un algoritmo complejo para decidir cómo distribuir el texto en las líneas del párrafo. Simplificando un poco el proceso, podríamos decir que calcula un parámetro que mide «qué tan mal» se ve el texto y trata de minimizarlo en dos pasadas. En la primera pasada calcula qué tan mal se ve el texto sin la separación silábica y si el resultado es aceptable, pues todo listo. Si el resultado de la primera pasada no es aceptable (y suele no serlo), pasa de nuevo permitiendo esta vez la separación silábica al final de la línea.

Lo que dice si un resultado es «aceptable» o no es un parámetro llamado «tolerancia»: si lo mal-que-se-ve está dentro de lo tolerado, pues pasa.

Y dado que el algoritmo hace dos pasadas tenemos dos parámetros de «tolerancia», uno para la primera pasada y otro para la segunda.

¿Y como controlamos estas tolerancias? En el preámbulo L<sup>A</sup>T<sub>E</sub>X se escribe

```
\pretolerance=<valor>
```

`\tolerance=<valor>`

Donde `<valor>` es un número entero entre 0 y 10000.

Y es en estos valores donde viene la «magia negra»: el valor correcto para `pretolerance` y `tolerance` dependerá del tamaño de la página, de la fuente tipográfica, de los márgenes, del número de columnas... en fin, que encontrarlos es cosa de prueba y error. Y es que cuanto más «angosto» el texto es, menos estables estos parámetros resultan, lo cual significa que cuando más los necesitemos más difícil será el encontrarlos.

Y sí, la vida es cruel.

Más información información sobre estos temas en:

<http://wiki.lyx.org/FAQ/Spacing>

<http://tex.stackexchange.com/questions/13742/what-does-double-spacing-mean>

<https://en.wikipedia.org/wiki/Leading>

## 6.11 Microtipografía

Antes que nada, una advertencia: el tema de esta sección es puramente sobre *estética*. De lo que vamos a hablar es de cómo hacer que el texto que creamos en  $\text{LyX}$ , ya sea con  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  o con  $\text{X}_{\text{Y}}\text{T}_{\text{E}}\text{X}$ , se vea «más bello». No tiene utilidad por sí mismo y de hecho la mayor parte de los lectores no notará lo que haremos... ¡lo cual significará que lo hemos hecho bien! Pero vayamos por partes.

### 6.11.1 Qué significa «microtipografía» y para qué sirve

En la introducción del manual de `microtype` nos encontramos con esta inspirada definición (la traducción —libre—, a continuación)

Micro-typography is the art of enhancing the appearance and readability of a document while exhibiting a minimum degree of visual obtrusion. It is concerned with what happens between or at the margins of characters, words or lines. Whereas the macro-typographical aspects of a document (i.e., its layout) are clearly visible even to the untrained eye, micro-typographical



refinements should ideally not even be recognisable. That is, you may think that a document looks beautiful, but you might not be able to tell exactly why: good micro-typographic practice tries to reduce all potential irritations that might disturb a reader.

Microtipografía es el arte de mejorar la apariencia y legibilidad de un documento entrometiéndose lo menos posible con su presentación. Está relacionado con qué sucede entre o a los bordes de los caracteres, palabras o líneas. Mientras que los aspectos macrotipográficos de un documento (su diseño) son claramente visibles incluso al ojo no entrenado, las mejoras microtipográficas idealmente no deberían ser reconocibles. Esto es, usted debería pensar que un documento se ve hermoso, pero no debería ser capaz de decir exactamente porqué: las buenas prácticas microtipográficas tratan de reducir todas las potenciales irritaciones que podrían molestar al lector.

Perfecto, la idea es modificar sutilmente el texto para que «se vea mejor», pero que quien lo vea no sea capaz, en un primer análisis, de decir qué se modificó. ¿Y esto cómo se logra?

Bien, primero que nada debemos hacer una declaración sumamente fuerte: es terriblemente simple engañar a nuestro cerebro. Siento desilusionarte, lector, pero esa maravillosa máquina biológica que con cariño llamamos «materia gris» puede fallar más que una escopeta de feria: es más que suficiente revisar el artículo sobre «ilusiones ópticas» en la wikipedia:

[https://en.wikipedia.org/wiki/Optical\\_illusion](https://en.wikipedia.org/wiki/Optical_illusion)

Y es justamente un tipo de «ilusión óptica» lo que enfrentamos al ver un texto justificado «normal»: cada tanto tenemos la sensación de que el margen derecho es «ondulado», con «entradas». Justamente donde se tienen los signos de puntuación y los guiones parece que la línea retrocediera, aún cuando esto no sea así (podríamos comprobarlo con una regla).

Lo que sucede aquí es que los puntos, comas y guiones tienen «menos peso» que los caracteres que los anteceden por lo que nuestro cerebro interpreta la imagen como si hubiera «huecos» allí.

Miremos ahora *este* texto, el del presente libro. El margen derecho se ve más «recto» que el de un texto normal, ¿verdad? ¡Falso! Como puede

comprobarse con una regla las comas, los guiones y los puntos «atravesan» el margen derecho ligeramente. En esta situación el borde de las líneas parece más «lleno» y nuestro cerebro interpreta que el margen derecho es más «uniforme»... aún cuando no lo sea. El cambio es realmente mínimo, pero mejora notablemente la apariencia del texto.

s espacio  
entre di-  
... en fin,  
«queda

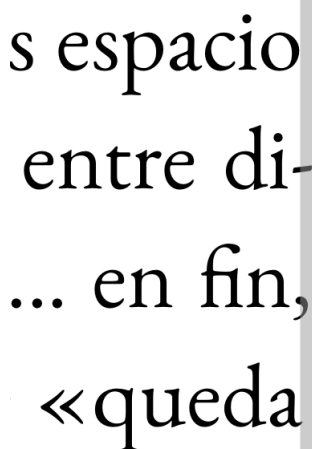


FIGURA 6.2: *Ejemplo de protrusión*

Esta técnica de *microtipografía* se llama «*protrusión*» y podemos ver cómo funciona en la figura 6.2: la línea gris muestra claramente cómo el guión y la coma «se pasan» del margen.

Otra técnica de *microtipografía* es deformar los caracteres para que ocupen más o menos espacio horizontal y así evitar el que los espacios entre palabras sean diferentes entre diferentes líneas. También se puede modificar la distancia entre caracteres... en fin, que las técnicas son varias. Pero a diferencia de la protrusión que siempre «queda bien», estas otras técnicas son importantes

mayormente cuando se tienen columnas angostas de texto, es decir, cuando la «macrotipografía» comienza a mostrar sus limitaciones.

¿Interesado? Si la respuesta es sí, si crees, lector, que las páginas de este libro se «ven mejor» que otros textos creados con, por ejemplo, Writer, pues entonces nos estamos entendiendo.

### 6.11.2 Cómo aplicar esta «microtipografía»

Lector, no intentes siquiera pensar en tratar de aplicar alguna de estas técnicas en procesadores de texto normales como Writer o Word: simplemente no es posible. Se podría pensar en una serie de «trucos sucios», pero el esfuerzo sería demasiado alto y los resultados no necesariamente buenos.

Según la wikipedia *Scribus* permite trabajar limitadamente con algunas de estas técnicas, pero nunca he utilizado este programa por lo que no puedo

comentar al respecto.

Lo único que nos queda en el mundo FLOSS es L<sup>A</sup>T<sub>E</sub>X y sus derivados con el paquete `microtype`, cuya documentación se instala, en Linux, en `/usr/share/texmf/doc/latex/microtype/microtype.pdf`

El manual tiene más de 240 páginas, pero a no preocuparse: en un uso normal las opciones por defecto son simplemente perfectas por lo que no tendremos que preocuparnos de cómo funciona.

Para habilitar las opciones por defecto de microtipografía, en L<sup>A</sup>T<sub>E</sub>X 2.3 y posteriores nos dirigimos al menú de configuración de fuentes y marcamos la correspondiente casilla, mientras que en versiones anteriores tenemos que dirigirnos al preámbulo L<sup>A</sup>T<sub>E</sub>X y escribir<sup>3</sup>

```
\usepackage{microtype}
```

Listo, no hay más que hacer. Esto funciona ya sea con L<sup>A</sup>T<sub>E</sub>X normal, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X o LuaL<sup>A</sup>T<sub>E</sub>X, aunque en estos dos últimos solo en forma limitada: por ejemplo, al momento de escribir este libro X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X solo ofrece soporte para protrusión.

Para el lector curioso de estos temas, he aquí algunos enlaces

<http://hstuart.dk/2007/11/27/microtypography/>

<http://wiki.lyx.org/Tips/Typography>

<https://en.wikipedia.org/wiki/Microtypography>

## 6.12 Sub- y superíndices

Me refiero a sub- y superíndices en el texto, que en modo matemático es fácil: el guión bajo para para subíndices, el acento circunflejo para los superíndices y a otra cosa. Ahora bien, tener sub- y superíndices en modo texto también es necesario, ¿cómo los logramos?

Antes de seguir tenemos que distinguir entre sub- y superíndices *reales* y *simulados*. Los sub- y superíndices reales son caracteres especiales diseñados

---

3. Claramente también podemos hacer esto en 2.3, simplemente hay que evitar el tener *simultáneamente* la opción marcada y la instrucción en el preámbulo, que nunca hay que llamar la misma opción dos veces.

explícitamente para actuar como tales mientras que los simulados simplemente provienen de hacer la fuente más pequeña y subirla (o bajarla) un poco, lo cual nunca se ve igual. Por ejemplo,

$2a^{22}$

donde el primer «2» después de la  $a$  es un superíndice real mientras que el segundo ha sido simulado (el 2 delante de la  $a$  sirve de referencia).

Como puede verse los índices reales se ven mejor que los simulados, con menos contraste y menos detalles lo que facilita su legibilidad. Y es que al escalar la fuente para simular el índice también se reducen el grosor de los trazos, haciendo que los índices se vean más «claros» que la letra de base.

Ya hemos hablado sobre cómo aplicar especialidades OpenType sobre una fuente por lo que activar los índices verdaderos con las etiquetas +sup, +sinf o +subs no debería ser un problema para el lector... salvo por el hecho de que no todas las fuentes tipográficas ofrecen estas opciones y de que las pocas que sí no nos dan todos los caracteres que podríamos necesitar.

En fin, que tendremos que utilizar índices simulados, otra no nos queda. Para aliviar este problema resulta conveniente elegir una fuente tipográfica que ofrezca «tamaños ópticos» como por ejemplo Latin Modern, Antykwa Półtawskiego o, en menor medida, EB Garamond, ya que los índices serán generados a partir de la fuente correcta para ese tamaño y se verán bien.

L<sup>A</sup>T<sub>E</sub>X nos permite trabajar con índices simulados gracias a las instrucciones `\textsuperscript{}` y `\textsubscript{}`. Para ambas L<sub>X</sub> ofrece soporte completo desde la interfaz gráfica.

Para insertar estos índices «en modo texto» en L<sub>X</sub> tenemos los menús

Insertar » Formato » Superíndice/Subíndice.

Por defecto L<sub>X</sub> no ofrece atajos de teclado para estas opciones, pero ya hemos visto cómo solucionar este problema (sección 3.2): lo único que necesitamos son los «códigos» correspondientes que para un superíndice es:

```
script-insert superscript
```

mientras que para el subíndice es:

```
script-insert subscript
```

NOTA: Si nuestra fuente es particularmente completa con todos los sub- y superíndices reales existe un paquete (cuando no...) que al encontrarse con las instrucciones `\textsuperscript` y `\textsubscript` las reemplaza automáticamente por la opción OpenType correspondiente. Este paquete se llama `realscripts`, pero no hablaré de él aquí: no solo su documentación es muy simple (seis páginas), sus resultados pueden ser un tanto «peligrosos». Y es que si se nos escapa un carácter para el cual la fuente no tiene un índice (muchas fuentes tienen solo los números y casi ninguna ofrece acentos o la ñ) tendremos resultados desagradables.

Ahora bien, ¿cómo podemos hacer para modificar cuánto se reduce la fuente del índice simulado y cuánto es este subido o bajado? Pues bien, para lograrlo tendremos que recurrir a código L<sup>A</sup>T<sub>E</sub>X puro y duro a través de las famosas «cajas T<sub>E</sub>X».

La instrucción a utilizar es `\raisebox` y funciona del siguiente modo

```
\raisebox{<altura>}{<texto>}
```

donde `<altura>` indica el desplazamiento vertical y `<texto>` aquello que queremos convertir en índice.

El valor de `<altura>` puede ser positivo (desplaza hacia arriba) o negativo (desplaza hacia abajo) y puede ser expresado en cualquier unidad de medida. Por ejemplo

```
\raisebox{6pt}{superíndice}  
\raisebox{-6pt}{subíndice}
```

nos dará (la primera) un superíndice a 6 puntos por encima de la línea de base y un subíndice (la segunda) a 6 puntos por debajo de la línea de base.

Para el tipógrafo detallista, también podemos utilizar como unidad el «ex». Muy básicamente, un ex es un valor igual al de la altura de una letra minúscula sin trazos ascendentes o descendentes: por ejemplo, la altura de una «x». Podríamos por lo tanto utilizar

```
\raisebox{1ex}{superíndice}
```

para hacer que el superíndice suba a la altura del «techo» de las letras bajas.

Ahora bien, la instrucción `\raisebox` no cambia por si misma el tamaño del texto por lo que tendremos que utilizar una instrucción extra. Por ejemplo

```
\raisebox{1ex}{\scriptsize{superíndice}}
```

nos dará un superíndice de tamaño «scriptsize» (introducimos los tamaños de fuente para  $\text{\LaTeX}$  en la sección 5.3) a una altura de 1ex sobre la línea de base. En principio, para utilizar este código se necesitan dos «cajas  $\text{\TeX}$ », una antes y la otra después del texto a poner como índice, siguiendo el modelo siguiente

base `{\raisebox{1ex}{\scriptsize{` superíndice `}}`

obtendremos el siguiente resultado

base<sup>superíndice</sup>

Sé lo que estás pensando, lector, escribir todo esto todas las veces que necesitamos un sub- o superíndice... ¿no existe una forma de abreviar? Ciertamente, pero hablaremos de ello en el capítulo 8.

## Capítulo 7

# Notas al pie de página y notas al final

Aquí hablaremos de cómo modificar la presentación y el funcionamiento de las notas al pie de página en un documento L<sup>A</sup>T<sub>E</sub>X y de cómo utilizar notas al final.

Para todo esto necesitamos el paquete `texlive-yafont`.

Varios de los comandos que indicaré en este capítulo aceptan parámetros de configuración, los cuales no voy a comentar. En la mayor parte de los casos hacer lo comentado aquí será más que suficiente por lo que queda a la curiosidad del lector el consultar la documentación de estos paquetes para comprender cómo ir más allá.

### 7.1 Número de las notas sin superíndice

Por defecto, en L<sup>A</sup>T<sub>E</sub>X y derivados no solo el número que identifica la nota al pie en el texto está como superíndice, sino también el que numera la nota en sí. Una forma de cambiar esto último es agregar lo siguiente en el preámbulo L<sup>A</sup>T<sub>E</sub>X:

```
\makeatletter
\renewcommand\@makefnmark[1]{%
  \noindent\makebox[0pt][r]{
    \@thefnmark.\space#1}
```

```
\makeatother
```

Aquí estamos redefiniendo «la marca» de las notas al pie, haciendo que no tengan sangría (`noindent`), que el número esté a la derecha (la `r`) de una «caja» (`makebox`) alineada a `o pt` del margen (y por lo tanto el número entrará en el margen izquierdo) mientras que al número en sí (`@thefnmark`) no se le aplica formato alguno, por lo que aparecerá «normal» y no como superíndice. ¡Mucho código aquí!

## 7.2 Notas al pie en columnas con texto principal sin columnas

Supongamos que tenemos muchas notas al pie, pero todas ellas sumamente breves (por ejemplo, para indicar la página en una de las referencias). En esta situación podría resultar interesante el tener las notas al pie de página en dos columnas aún cuando el cuerpo del texto se presente sin columnas.

Para esto escribimos en el preámbulo  $\LaTeX$  lo siguiente

```
\usepackage{dblfnote}
```

y todo funcionará como se desea. La única cosa a tener presente es que si la página está medio vacía las notas al pie se acomodarán a sus anchas haciendo que queden en la primera columna.

## 7.3 Notas al pie sin quiebres de párrafo

Si agrupar las notas al pie en dos columnas no es suficiente y queremos obtener notas al pie en la misma línea sin quiebres de párrafo, el paquete `footmisc` es nuestro amigo. En el preámbulo  $\LaTeX$  es suficiente escribir:

```
\usepackage[para]{footmisc}
```

y todo estará listo.



## 7.4 Reiniciar la numeración de las notas en cada página

Es suficiente agregar, en el preámbulo  $\LaTeX$ , lo siguiente

```
\usepackage{perpage}
\MakePerPage{footnote}
```

La primera línea declara el uso de la instrucción `perpage`, una instrucción que reinicia la numeración de distintos contadores en cada página, mientras que la segunda indica cuál contador queremos reiniciar (el de las notas al pie).

## 7.5 «Numerar» las notas con símbolos

Supongamos que queremos la siguiente sustitución:

- 1  $\rightarrow$  \* (comando  $\LaTeX$ : `\ast`)
- 2  $\rightarrow$  † (comando  $\LaTeX$ : `\dagger`)
- 3  $\rightarrow$  ‡ (comando  $\LaTeX$ : `\ddagger`)
- 4  $\rightarrow$  § (comando  $\LaTeX$ : `\S`)
- 5  $\rightarrow$  \*\*

etcétera. Es importante tener en cuenta que «numerar con símbolos» solo tiene sentido si reiniciamos la numeración de las notas en cada página y si se tienen pocas notas en una misma página, de lo contrario el documento se volvería caótico.

Esto tendría que poder hacerse con el paquete `footmisc`, pero la verdad es que no logro hacerlo funcionar por lo que aquí presento una alternativa que, si bien resulta un poco más complicada, puede utilizarse sin mayores problemas.

Dirigiéndonos al preámbulo  $\LaTeX$  escribimos lo siguiente:

```
\def\mysymbols#1{\@mysymbols{\@nameuse{c@#1}}}{
\def\@mysymbols#1{\ifcase#1\or $\ast$\or
$\dagger$\or $\ddagger$\or $\S$\or
$\ast\ast$\or $\dagger\dagger$\or
$\ddagger\ddagger$\or $\S\S$}
```

```

\else\@ctrerr\fi}
\def\thefootnote{\mysymbols{footnote}}
\usepackage{perpage}
\MakePerPage{footnote}

```

La primera línea define una nueva instrucción llamada `\mysymbols`, la segunda configura esa instrucción estableciendo las sustituciones que comentamos más arriba mientras que más abajo se redefine la instrucción `\thefootnote` para hacer que a la numeración venga aplicada la sustitución anterior. Las últimas dos líneas *deben estar al final* y aplican el paquete `perpage` a las notas al pie.

Es importante notar que si tratamos de utilizar en una página más notas del número de símbolos que hemos definido  $\text{\LaTeX}$  dará error, por lo que mejor pecar de exagerado: si queremos definir más símbolos, simplemente agregamos otro «`\or  $\$$ \símbolo $\$$` » en la lista de la segunda línea de código.

Fuente de inspiración para esta sección:

[https://groups.google.com/d/msg/comp.text.tex/QZrTF7tzo5k/TvT\\_RsCyBfsj](https://groups.google.com/d/msg/comp.text.tex/QZrTF7tzo5k/TvT_RsCyBfsj)

## 7.6 Cambiar la fuente de las notas al pie

Por defecto en un documento  $\text{\LaTeX}$  las notas al pie de página utilizan la misma fuente que el cuerpo de texto, pero en el tamaño, prepárense para la sorpresa, `footnotesize`.

Para modificar tanto la fuente como el tamaño de las notas al pie podemos utilizar el paquete `footmisc` (para esto sí funciona).

Escribiendo en el preámbulo (como siempre, estamos utilizando  $\text{\XeTeX}$ )

```

\newfontfamily{\kley}[]{\Kleymisky}
\usepackage{footmisc}
\renewcommand*{\footnotelayout}{\small\kley}

```

lograremos utilizar la fuente `Kleymisky`<sup>1</sup> en la nota al pie (no es que lo recomiende, pero ya saben, para dar un ejemplo claro) cambiando el tamaño

1. <http://www.glukfonts.pl/font.php?font=Kleymisky>

por small.

NOTA: Esto funciona perfectamente con lo comentado en las tres secciones precedentes, *pero no con lo que presentamos en la primera sección de este capítulo*. Ya saben, conflicto de paquetes...

NOTA 2: (que es un corolario de la nota anterior) con esto no cambiamos la fuente de las etiquetas de las notas al pie.

## 7.7 Notas al final

No solo de notas al pie vive el escritor, en ciertos documentos es necesario el utilizar notas al final.

LyX ofrece soporte desde la interfaz gráfica del paquete `endnotes`. Para utilizarlo nos dirigimos a Documento Configuración Módulos para habilitar el módulo «Notas finales».

Para insertar una nota final en el documento simplemente nos dirigimos a Insertar Recuadro personalizado Notas finales y escribimos el texto de la nota dentro de la caja que se nos presenta.

Finalmente, para *mostrar* las notas en el documento tenemos que insertar una caja T<sub>E</sub>X con el siguiente código


```
\renewcommand{\notesname}{Notas al final}
\theendnotes
```

La primera línea le da nombre a la «sección» de las notas al final mientras que la segunda inserta la lista de las notas.

Algo interesante de este sistema es que podemos insertar estas instrucciones tantas veces como queramos, por lo que podríamos tener las notas al final acumuladas por capítulos, por ejemplo, y no todas al final. Eso sí, si queremos separar las notas al final de esta forma resultará conveniente reiniciar la numeración de las mismas cada vez, agregando la instrucción

```
\setcounter{endnote}{0}
```

## 7.8 Una *yapa*: notas al margen

LyX ofrece soporte para insertar notas al margen: simplemente nos dirigimos a  Nota al margen y ya la tenemos lista para completarla. ¡Hol@!

No hay mucho que pueda hacerse para modificar su formato desde la interfaz gráfica del programa por lo que tendremos que recurrir a las «cajas T<sub>E</sub>X» si queremos modificarlas.

Por ejemplo las notas al margen se colocan en el margen «externo» del documento, pero esto puede cambiarse rápidamente insertando una caja T<sub>E</sub>X con el código



```
\reversemarginpar{}
```

mientras que para volver «a la normalidad» solo tenemos que escribir, en otra caja T<sub>E</sub>X, lo siguiente:

```
\normalmarginpar{}
```

LyX utiliza la implementación por defecto de L<sup>A</sup>T<sub>E</sub>X de las notas al margen, la cual tiene algunas limitaciones. Para utilizar el paquete `marginnote` que es mucho más completo podemos escribir en el preámbulo L<sup>A</sup>T<sub>E</sub>X

```
\usepackage{marginnote}
\let\marginpar\marginnote
```

A este punto me veo obligado a confesar algo: no soy particularmente «fanático» de las notas al margen<sup>2</sup> por lo que no seguiré hablando de ellas y dejaré que el lector interesado consulte la documentación de LyX sobre el tema. Todo lo que se puede saber sobre las notas al margen en LyX se encuentra magníficamente explicado en  Objetos insertados  sección 4.3.

---

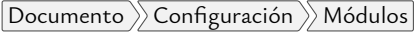
2. Hasta podría decirse que las detesto.

## Capítulo 8

# Módulos, formato local y macros matemáticas

Otra de las notables características de  $\text{\LaTeX}$  es que puede *extenderse*. En este capítulo veremos cómo agregar funciones, el modo correcto de definir formatos no presentes en la *documentclass* utilizada y cómo preparar algunos «trucos» para escribir fórmulas matemáticas con mayor facilidad.

### 8.1 Módulos

En el capítulo 7 mencionamos, al pasar y como quien no quiere la cosa, los «módulos»: bajo  (figura 8.1) tenemos una larga lista de nuevas funciones que solo esperan a ser activadas.

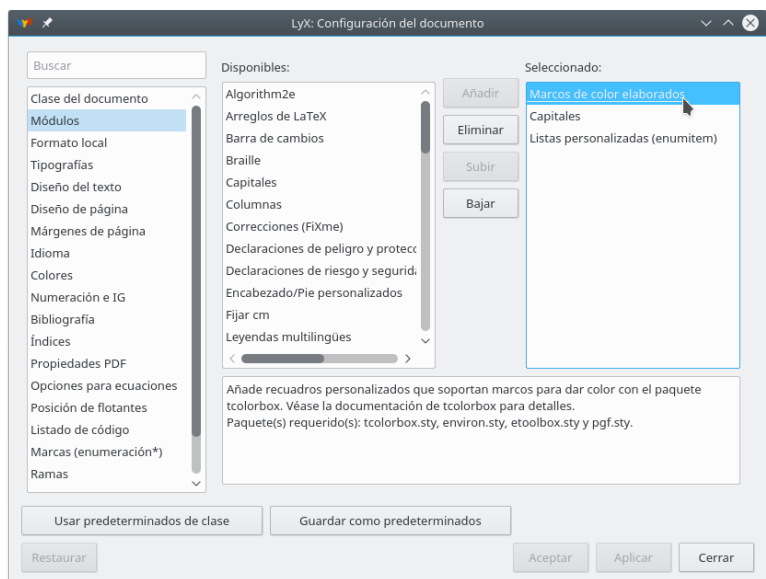
Estos módulos simplifican el uso de determinados paquetes  $\text{\LaTeX}$ . Hablar de todos ellos es simplemente imposible, pero en próximos capítulos veremos algunos por lo que por el momento dejo al lector el jugar con las opciones. Ahora bien, ¿es posible agregar *otros* módulos no ofrecidos en la instalación por defecto?

¡Ciertamente! Veamos cómo.

En esta página de la wiki de  $\text{\LaTeX}$

<http://wiki.lyx.org/Layouts/Modules>

es posible encontrar varios módulos listos para ser utilizados. Tenemos módulos que agregan nueva funcionalidad (comentarios, índices de nombres, formato), integración con el programa Sage, modificaciones del for-

FIGURA 8.1: *Activando módulos*

mato de distintas *documentclass*, etcétera.

La forma de instalar estos módulos es, luego de descargar los archivos \*.module, copiarlos en la carpeta «layouts» del perfil de usuario de LyX que en Linux se encuentra en

`~/.lyx/layouts/`

¡Recuerden siempre el menú reconfigurar!

## 8.2 Crear nuevas instrucciones

Ya hemos creado instrucciones al definir nuevas familias de fuentes en la sección 6.2. Pero podríamos necesitar más: podría ser necesario tener a disposición una «abreviatura» para insertar un texto que va a repetirse muchas veces o incluso crear una «función» que dada una entrada rápida nos de en el documento final algo perfectamente formateado.

La forma más simple de lograr esto en  $\text{\LaTeX}$ /LyX es definir una nueva instrucción con `\newcommand{ } [ ] { }`. Este comando se utiliza en el preámbulo

bulo  $\text{\LaTeX}$  de la siguiente forma:

```
\newcommand{\micomando}[<número de
    parámetros>]{<texto e instrucciones LaTeX>}
```

Supongamos que queremos crear una nueva instrucción para insertar un superíndice en modo texto utilizando el comando `raisebox`. Aquí `\micomando` es el nombre que le daremos a la instrucción, por ejemplo `\subir`, `<número de parámetros>` indica cuántos parámetros necesitará nuestro comando, en este caso 1, mientras que en `<texto e instrucciones LaTeX>` debemos escribir lo que nos sirva para crear el formato buscado, llamando a cada parámetro con `#<número>` (`#1` para el primer parámetro, `#2` para el segundo, etcétera).

El código a insertar en el preámbulo  $\text{\LaTeX}$  será entonces el siguiente

```
\newcommand{\subir}[1]{\{\raisebox{1ex}{\footnotesize{#1}}\}}
```

Para utilizar esta instrucción dentro de  $\text{LyX}$  abrimos una primera «caja  $\text{\TeX}$ » con `[Ctrl][L]`, escribimos `\subir{`, salimos de la caja para escribir el texto del superíndice y finalmente creamos otra caja  $\text{\TeX}$  donde escribimos `}`. De esta forma, teniendo lo siguiente en la ventana de edición de  $\text{LyX}$

base `\subir{` superíndice `}`  
 obtendremos  
 base<sup>superíndice</sup>

Podemos utilizar más parámetros, o incluso ninguno. En este último caso la instrucción tendría la forma

```
\newcommand{\micomando}{<texto e
    instrucciones LaTeX>}
```

¡Todo listo! Bueno, no todo, que esto sigue siendo poco práctico. ¡Pero ya hemos llegado! Finalmente veremos cómo hacer esto y mucho más al mejor estilo  $\text{LyX}$ .

### 8.3 Crear estilos de carácter en LyX

En LyX podemos aplicar rápidamente algunos estilos de carácter predefinidos como «énfasis» (con `Ctrl E`, por defecto da cursiva) o negrita (`Ctrl B`), pero no mucho más. Activando el módulo «marcación lógica» tendremos algunos más, los cuales aparecerán listados en el [clic derecho](#) `Estilo de texto` o en `Editar` `Estilo de texto`. Ahora bien, ¿si queremos crear otro? Podríamos necesitar un modo más simple de activar una característica OpenType determinada en lugar de tanta «caja T<sub>E</sub>X» o escribir menos a la hora de utilizar el paquete `hologo` del que hablamos antes.

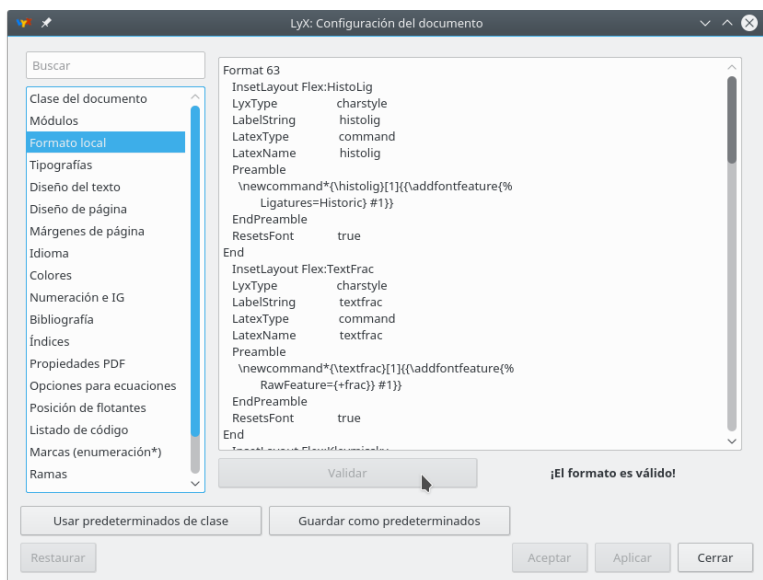
NOTA: ¡Recuerda, lector, que para este segundo ejemplo hay que escribir `\usepackage{hologo}` en el preámbulo!

Nos dirigimos a `Documento` `Configuración` `Formato local` donde tenemos que escribir algo como lo que se muestra a continuación:

```
Format 60
InsetLayout Flex:HistoLig
LyxType          charstyle
LabelString      histolig
LatexType        command
LatexName        histolig
Preamble
  \newcommand*{\histolig}[1]{\%
    \addfontfeature{Ligatures=Historic} #1}}
EndPreamble
ResetsFont       true
End

InsetLayout Flex:LogoTipo
LyxType          charstyle
LabelString      HoLogo
LatexType        command
LatexName        loguito
Preamble
  \newcommand*{\loguito}[1]{\hologo{#1}}}
```



FIGURA 8.2: *Crear un formato local*

```

EndPreamble
ResetsFont           true
End

```

## NOTAS:

- Antes de que alguien pregunte, el `Format 60` del inicio indica la versión del sistema utilizado por `LyX 2.2.x` para crear los estilos (en `2.3.x` es `64`). Esta primera línea es obligatoria, *pero solo necesitamos escribirla una vez* si definimos varios «formatos locales».
- Luego de los dos puntos en la segunda línea **NO** debe quedar espacio. Esta línea define el nombre del nuevo formato.
- `LyxType` define el tipo de formato que estamos creando: en el ejemplo un estilo de carácter (`charstyle`).
- `LabelString` es la etiqueta que veremos en la ventana de edición de `LyX` marcando el formato aplicado.

- `LatexType` define que estamos aplicando un comando, sea este de base o, como en el ejemplo, definido por nosotros.
- `LatexName` es el nombre (sin la barra inicial) del comando  $\LaTeX$  a ser aplicado.
- Si queremos que *en la ventana de edición* nuestro estilo se presente en forma diferente podemos variar el tipo de fuente haciendo algo como lo siguiente:

Font

Family	Serif
Shape	Italic

EndFont

- La parte entre `Preamble` y `EndPreamble` sirve para definir código que tendría que estar en el preámbulo  $\LaTeX$ . Evidentemente si el código en cuestión ya está en el preámbulo no es necesario definirlo aquí.
- `ResetsFont` sirve para que el formato que aplicamos no se extienda más allá del texto seleccionado.
- Antes de poder «aceptar» tendremos que utilizar el botón «validar»: solo si LyX confirma que el código está bien escrito podremos seguir adelante.
- También es posible crear «entornos» (estilos de párrafo), pero eso está ya magníficamente explicado en los manuales de LyX ([Ayuda](#) [Personalización](#) [sección 5.3.6](#)), por lo que no entraré en ese tema.

Ahora solo nos queda seleccionar el texto «a ser estilizado», hacer un [clic derecho](#) [Estilo de texto](#) [HistoLig](#) y aquello que en la ventana de edición de LyX se ve más o menos así

esto es texto normal esto es estándar histórico esto es texto normal  
histolig

en el resultado se verá así:

esto es texto normal esto es estándar histórico esto es texto normal


Mientras que

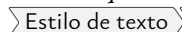

$\text{\texttt{XeTeX}}$   $\text{\texttt{AmS}}$

$\text{\texttt{HoLogo}}$   $\text{\texttt{HoLogo}}$

se verá así:

$\text{\texttt{XeTeX}}$   $\mathcal{AMS}$

Para *quitar* un estilo de texto ya aplicado es suficiente hacer 

En la subsección 15.1.2 daremos otro ejemplo de formato local mientras que en el apéndice B el lector encontrará la definición de todos los estilos utilizados para crear este libro.

## 8.4 Macros en modo matemático para $\text{\texttt{LX}}$

Consideremos dos situaciones:

**SITUACIÓN 1:** podríamos tener una expresión matemática compleja que tengamos que repetir, con pequeñas variantes, varias veces en el mismo documento. Por ejemplo, podríamos estar escribiendo un ensayo sobre transformadas de Fourier lo que nos obligaría a introducir muchas veces una expresión donde solo cambiaría la función a transformar, no el resto de la integral.

**SITUACIÓN 2:**  $\text{\texttt{LX}}$  no puede dar soporte a cada comando  $\text{\texttt{LX}}$  posible por lo que a veces tendremos que recurrir a escribir código directamente, código que, por si fuera poco, podría verse «extraño» en la ventana de edición.

Afrontar cualquiera de estas situaciones significaría un enorme trabajo manual de ajustar detalles y compilar «para ver cómo va quedando», trabajo que se repetiría hasta el cansancio... o no, o al menos no siempre.

Bienvenidos a una de las características más interesantes, pero menos conocidas, de  $\text{\texttt{LX}}$ : *las macros matemáticas*.

La idea es que nosotros creamos una «macro» con las instrucciones  $\text{\texttt{LX}}$  necesarias (la parte fija de la expresión), los «argumentos» que nos sirvan (la parte que cambiaremos cada vez) y le damos un breve nombre de

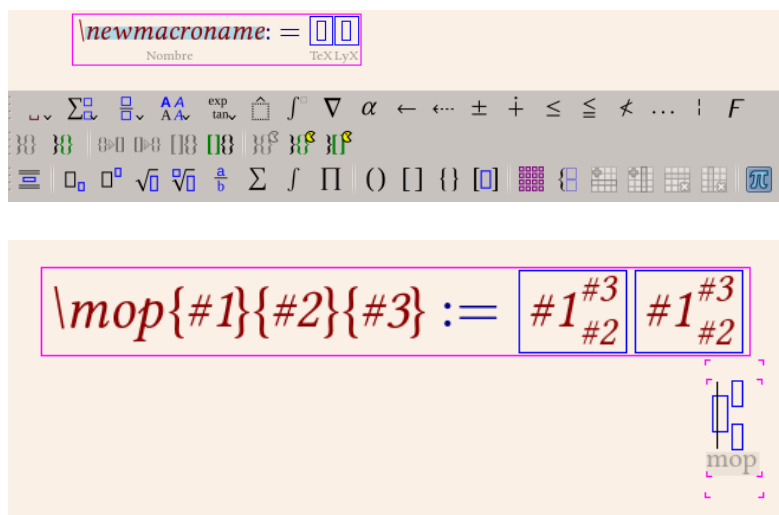


FIGURA 8.3: *Crear una macro matemática: arriba, iniciar el proceso; abajo, cómo se ve todo en la ventana de LyX, con la expresión esperando a que los tres parámetros sean completados*

nuestra autoría. Una vez creada esta macro solo tendremos que llamarla desde cualquier lugar del documento utilizando su nombre, que la única tarea que nos quedará será la de llenar los «huecos» dejados por los argumentos.

Con cada nueva versión de LyX la SITUACIÓN 2 resulta cada vez más rara, ya que los comandos  $\text{\LaTeX}$  que quedan sin implementar son justamente los más «exóticos», por lo que comentemos cómo resolver la SITUACIÓN 1 que en el proceso resultará claro cómo funciona todo.

Para esto utilizaremos el comando `mathop` para el cual la versión 2.3 de LyX agrega soporte gráfico. Este comando nos sirve para convertir cualquier signo o texto en un «operador» que acepte límites.

Primero, al principio del documento nos dirigimos a [Insertar](#) [Ecuación](#) [Macro](#) que aparecerá algo como lo mostrado en la imagen superior de la figura 8.3.

Como vemos, en el documento se abrirá una «caja» que utilizaremos para crear nuestra macro mientras que debajo tendremos, además de las barras de herramientas tradicionales del editor de ecuaciones, una nueva para

crear o quitar argumentos, hacerlos opcionales, etcétera.

NOTA: Esta caja no aparecerá en el documento compilado por lo que no debemos preocuparnos de su presencia. De hecho podríamos colocarla en cualquier lugar, pero es mejor agrupar todas las macros al inicio del documento para así tenerlas «más a mano».

Ahora reemplazamos `newmacroname` por algo que nos sea cómodo. Dado que en este ejemplo queremos simplificar el uso del comando `\mathop` podemos llamarlo `mop`.

IMPORTANTE: ¡Cuidado con el nombre elegido, que no podemos utilizar un comando  $\LaTeX$  válido!

En la primera de las dos pequeñas «subcajas», la marcada como «TeX», tendremos que escribir nuestro código  $\LaTeX$  mientras que en la segunda, marcada como «LyX», aquello que queremos que  $\LaTeX$  muestre durante el uso de la macro. Dado que `\mathop` tiene soporte gráfico (en 2.3, insisto: en versiones anteriores tendremos el comando a la vista), simplemente escribimos lo mismo en ambas cajas.

Y aquí es donde vemos cómo resolver la SITUACIÓN 2: en la segunda caja *no* escribimos lo mismo que en la primera, sino algo que nos permita entender qué obtendremos en la salida.

Recuerde el lector que en el editor de ecuaciones de  $\LaTeX$  la forma más simple de insertar los delimitadores `{ }` para agrupar expresiones es escribiendo `\{` (el de cierre aparecerá automáticamente).

A continuación utilizamos la nueva barra de herramientas para agregar el número de argumentos necesarios, que en este caso son tres y les aplicamos las instrucciones necesarias a cada uno (`\mathop` al primer parámetro, subíndice al segundo y superíndice al tercero), lo que nos dará lo que se muestra en la imagen inferior de la figura 8.3.

Ya hemos llegado. Para utilizar nuestra flamante macro solo tenemos que escribir dentro de una ecuación

`\mop`

para luego presionar la barra espaciadora una vez. Ahora tendremos tres cajas esperando contenido.

Finalmente, poniendo un igual en el primer parámetro, dejando vacío el segundo y escribiendo «def» en el tercero, o bien escribiendo la suma para  $i$  y  $j$  iguales a uno hasta  $N$  en el primer parámetro,  $i > j$  en el segundo y dejando vacío el tercero, en el documento compilado (recuerden que la caja que define la macro no aparecerá) obtendremos

$$f(x) \stackrel{def}{=} \ln(x)$$

$$\sum_{\substack{i,j=1 \\ i>j}}^N A_{ij}$$

Podemos «anidar» nuestras expresiones o, como vemos, dejar algunas de las cajas en blanco: el único límite lo pone nuestra creatividad a la hora de resolver nuestros problemas.

¡Todo listo! Como puedes ver, estimado lector, las macros matemáticas en LyX resultan una herramienta increíblemente flexible y simple de utilizar, ¡no dejes de aprovecharlas!

## Capítulo 9

# Listas numeradas

Supongamos que queremos construir una lista numerada en un documento  $\text{L}\text{X}$ , pero que queremos numerarla con números romanos o incluso letras. Buscando en internet encontraremos algunos trucos para  $\text{L}\text{A}\text{T}\text{E}\text{X}$  para hacer esto modificando el preámbulo con códigos ilegibles, los cuales no son ni simples de usar ni perfectos.

En este capítulo veremos cómo modificar una lista particular y luego cómo hacer que varias listas de nuestro documento utilicen el mismo formato por nosotros elegido.

### 9.1 Modificar listas particulares

Para este trabajo  $\text{L}\text{X}$  nos ofrece una alternativa que, si bien no resulta perfecta, es realmente simple y robusta por lo que nos evitará muchos problemas.

La primera cosa a comprobar es que nuestra distribución  $\text{L}\text{A}\text{T}\text{E}\text{X}$  nos ofrezca el paquete `enumitem`, el cual nos permite justamente administrar el tipo de numeración de las listas. Luego, en nuestro documento nos dirigimos a [Documentos] » [Configuración] » [Módulos], seleccionamos Listas personalizadas (`enumitem`) y presionamos Añadir.

Ahora nos dirigimos al principio de la primera línea de una lista numerada y seleccionamos el menú [Insertar] » [Opciones de enumeración]. En la caja que se ha abierto tenemos que insertar una «caja  $\text{T}\text{E}\text{X}$ ».

Dentro de estas dos cajas anidadas debemos escribir la instrucción para cambiar el tipo de numeración. Por ejemplo, si queremos que la numeración sea con números romanos en mayúsculas y sin separador sería

```
\label=\Roman{enumi}
```

Para números romanos en minúsculas, cambiamos `\Roman` por `\roman`, para numerar con letras, usamos `\Alph` para letras mayúsculas y `\alph` para minúsculas mientras que `\arabic` nos dará números arábigos normales.

Si tenemos listas de varios niveles y queremos que todos ellos se indiquen en cada entrada podemos utilizar la instrucción `label` en cada uno de ellos. Por ejemplo, para tener un primer nivel en números romanos y un segundo en alfabéticos, mostrando ambos en el segundo nivel usamos

```
\label=\Roman{enumi}.\alph{enumii}
```

como «opción de enumeración» de la primera instancia del segundo nivel. Como puede verse para el segundo nivel he escrito `enumii`, duplicando la `i` final. Esto puede extenderse a cualquier nivel, agregando una `i` por cada nivel modificado... ¡pero solo hasta el nivel 3! En el nivel 4 tenemos que utilizar `enumiv`.

Sí, números romanos en minúsculas... se ve que quienes desarrollaron este paquete son del «gran país del norte».

Otra cosa que puede verse en el ejemplo anterior es que he incluido un separador (el punto).

Desafortunadamente la doble caja con la instrucción `label` deberá ser incluida nuevamente cada vez que se reinicia el subnivel de la lista.

Algo realmente interesante de este módulo de `LyX` es que las referencias cruzadas a los ítem de la lista numerada funcionan a la perfección.

Dado que esta operación de la doble caja debe realizarse cada vez, en cada nivel y en cada lista, no es precisamente una comodidad, pero al menos funciona.

## 9.2 Modificar todas las listas del documento

Ahora presentaré un método que modifica *varias* listas del documento simultáneamente.



Supongamos que queremos que cada subnivel de nuestra lista numerada muestre el número del nivel superior además del suyo propio y que se tenga algo del tipo

I Primer nivel

I.A Segundo nivel

I.A.1 Tercer nivel

I.A.2 Otro tercer nivel

I.A.2.a Cuarto nivel

Es decir, el primer nivel con números romanos en mayúsculas y negrita, el segundo con letras mayúsculas, el tercero con números y el cuarto con letras minúsculas en bastarda, cada nivel mostrando la numeración de los niveles anteriores. Insertando una caja para código T<sub>E</sub>X al principio del documento (o al menos antes de la primera lista), debemos escribir algo como lo que sigue

```
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\textbf{\theenumi}}
\renewcommand{\theenumii}{\Alph{enumii}}
\renewcommand{\labelenumii}{\textbf{\theenumi}.
    {\theenumii}}
\renewcommand{\theenumiii}{\arabic{enumiii}}
\renewcommand{\labelenumiii}{\textbf{\theenumi}.
    {\theenumii}. {\theenumiii}}
\renewcommand{\theenumiv}{\alph{enumiv}}
\renewcommand{\labelenumiv}{\textbf{\theenumi}.
    {\theenumii}. {\theenumiii}.
    \textit{\theenumiv}}}
```

NOTA: Por motivos de presentación en este texto he introducido quiebres de línea luego de los separadores en los esquemas de cada numeración, *pero esto introduce un espacio entre los distintos elementos cuando se compila el documento*. Las líneas que definen qué se muestra en la numeración de cada nivel *deben ser escritas sin quiebres*.

Es decir, nunca hay que copiar y pegar código  $\text{\LaTeX}$  en forma irreflexiva.

De los comandos `\textbf`, `\textit`, `\Roman`, `\alph`, etcétera ya hemos hablado (capítulo 5) por lo que creo que el lector no tendrá problemas en moverse entre ellos para crear el formato que busca.

¿Pero qué significan *los otros* comandos  $\text{\LaTeX}$ ?

Pues bien, sin entrar en detalles sobre qué es y qué no es `\renewcommand` (todo un mundo allí), concentrémonos en los comandos `\theenum` y `\labelenum`. Como puede verse del código estos comandos están seguidos de i, ii, iii y iv para indicar el nivel de la lista numerada (otra vez los romanos en minúsculas). `\theenum` define el tipo de numeración y `\labelenum` qué se muestra en cada nivel.

Las referencias cruzadas (insertar una etiqueta en un elemento de la lista y llamarla a continuación) funcionan... más o menos bien: el tipo de numeración elegido es presentado, pero si como en el presente ejemplo elegimos que un subnivel muestre los niveles anteriores, en la referencia cruzada no se mostrarán los separadores.

Claramente si solo queremos que un dado nivel muestre su número y no el de los niveles superiores es suficiente quitar lo que no corresponda. Algo así:

```
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\textbf{\theenumi}}
\renewcommand{\theenumii}{\Alph{enumii}}
\renewcommand{\labelenumii}{\textit{\theenumii}}
\renewcommand{\theenumiii}{\arabic{enumiii}}
\renewcommand{\labelenumiii}{\textit{\theenumiii}}
\renewcommand{\theenumiv}{\alph{enumiv}}
\renewcommand{\labelenumiv}{\theenumiv}
```

debería funcionar. En este caso las referencias cruzadas no darán problemas.

El formato de numeración asignado será válido hasta el final del documento o hasta que ingresemos nuevamente una caja  $\text{\TeX}$  con un nuevo formato.





## Capítulo 10

# Sobre los índices

Configurar índices es quizás uno de los puntos más oscuros de  $\text{\LaTeX}$ . En este capítulo daremos una breve introducción a algunas herramientas para resolver cuestiones específicas relacionadas con los índices.

### 10.1 Títulos cortos

Comencemos con algo simple: ¿qué tal si el título de un capítulo o sección nos ha quedado demasiado largo y queremos que tanto el índice general como la cabecera de la página muestre una versión más corta del mismo?

Con el cursor en el título en cuestión: , o bien  o quizás, para los más memoriosos, , luego . La caja que se nos presenta nos permitirá de escribir una versión más corta del título para ser utilizada en el índice y en las referencias cruzadas, sobrescribiendo el título «real».

### 10.2 Índices parciales para cada capítulo

En el preámbulo  $\text{\LaTeX}$  debemos escribir

```
\usepackage{minitoc}
```

(obviamente, el paquete `texlive-minitoc` debe estar instalado). En el documento, justo antes del índice general insertamos una caja  $\text{\TeX}$  con el co-

mando

```
\dominitoc
```

Finalmente, por cada capítulo, justo debajo del nombre del capítulo y antes de cualquier sección insertamos otra caja T<sub>E</sub>X con el comando

```
\minitoc
```

### 10.3 Corregir espaciado en listas de figuras o cuadros

Cuando se usa una de las *documentclass* de artículo las figuras se numeran en secuencia sin interrupción, mientras que si se usa el formato libro la numeración por defecto de las figuras es <capítulo>.<número>.

En L<sub>A</sub>T<sub>E</sub>X es muy simple el activar una numeración por secciones, que para los artículos será <sección>.<número> mientras que en los libros nos dará <capítulo>.<sección>.<número>: es suficiente dirigirse a Documento Configuración Módulos (ya hablamos de los módulos en el capítulo 8) y activar «Numerar figuras por sección».

El problema con esto surge cuando tenemos un *documentclass* de libro y creamos el índice de figuras: el número extra ocupa más espacio que el previsto en la configuración por defecto del índice por lo que la numeración «choca» con el texto que le sigue.

Para conseguir más espacio entre número y texto hay que dirigirse al preámbulo L<sup>A</sup>T<sub>E</sub>X y escribir

```
\makeatletter
\renewcommand*\l@figure{\@dottedtocline{1}{
  1.5em}{3em}}
\makeatother
```

Para variar la distancia solo necesitamos jugar con el valor de {3em} que se encuentra a la derecha de la línea central del código indicado.

Si queremos hacer lo mismo con el índice de cuadros, agregamos

```
\renewcommand*\l@table{\@dottedtocline{1}{
  1.5em}{3em}}
```

entre `\makeatletter` y `\makeatother` y ya está todos listo.

Para profundizar un poco en este tema pueden consultarse los siguientes enlaces:

<http://tex.stackexchange.com/questions/133747/reduce-space-between-table-number-and-title-in-list-of-tables>  
<http://tex.stackexchange.com/questions/87019/list-of-figures-numbers-overlap-image-caption>

## 10.4 Evitar las líneas de puntos en los índices

Por defecto muchas *documentclass* usan líneas punteadas como «guía» de los índices que unen el texto del título con el número de página correspondiente.

Supongamos que queremos eliminar estas líneas punteadas. En el preámbulo  $\LaTeX$  agregamos

```
\renewcommand{\@dotsep}{10000}
```

Listo.

Este comando actuará sobre *todos* los índices del documento.

Lo que sucede aquí es que el comando `\@dotsep` controla la separación entre los puntos (*dot separation*, en inglés), por lo que si en lugar de 10000 usamos valores como 20 o 30 simplemente obtendremos una separación notablemente mayor que en la configuración por defecto. El valor 10000 es especial en  $\LaTeX$  y se usa para decir «infinito» en cualquier instrucción que necesite un parámetro numérico, por lo que en este caso termina deshabilitando la línea punteada.

## 10.5 Unas (pocas) palabras sobre los índices alfabéticos

En  $\LaTeX$  existen muchas formas de generar índices alfabéticos y  $\LaTeX$  ciertamente permite utilizarlas todas.

En  $\LaTeX$  podemos elegir entre `makeindex`, `texindy` o `xindy`. Si no sabes de lo que estoy hablando, estimado lector, a no preocuparte que es este un

tema complejo que requiere estudio para poder aprovecharlo. Esto significa que el hablar en detalle sobre los índices alfabéticos se escapa del propósito general de este libro por lo que solo diré lo mínimo indispensable. El lector curioso puede profundizar tanto en la guía del usuario de  $\text{LyX}$  (sección 6.6) como en los distintos manuales  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  disponibles en línea.

Para marcar una palabra o texto a ser incluido en el índice alfabético solo tenemos que seleccionarlo y dirigirnos a **Insertar**  $\gg$  **Entrada de índice**. Esto nos dará una «caja» con la marca justo después del texto marcado. Por defecto el contenido de la marca será igual al texto seleccionado, pero podemos cambiarlo sin problemas.

El índice propiamente dicho se introduce en cualquier parte del documento con **Insertar**  $\gg$  **Índices**  $\gg$  **Alfabético**. A diferencia del índice general, haciendo clic sobre el «botón» resultante no sucede nada, pero es posible configurar el «procesador» para generar el índice alfabético en **Documento**  $\gg$  **Configuración**  $\gg$  **Índices**. Allí además de elegir entre `makeindex`, `texindy` o `xindy` podemos pasar opciones a estos procesadores, elegir de crear más de un índice (podríamos tener uno de palabras, otro de nombres, etcétera) y administrar los índices existentes.

## Capítulo 11

# Títulos

En este capítulo veremos cómo modificar la presentación y la numeración de los títulos de nuestro documento, sean estos de «parte», «capítulo», «sección», etcétera. Iremos desde los cambios más simples a los más complejos.

Comencemos.

### 11.1 Renombrar los entornos «Parte» y «Capítulo»

Supongamos que queremos escribir un libro con  $\text{\LaTeX}$  y seleccionamos, por ejemplo, el *documentclass* «Book».

De esta forma tendremos a disposición varios «entornos», entre ellos Capítulo (comando  $\text{\LaTeX}$  `\chapter{}`) para separar las distintas partes de la obra y Parte (comando  $\text{\LaTeX}$  `\part{}`) para agrupar capítulos.

Pero supongamos que en lugar de separar en «Partes» queremos separar en «Libros» y que en lugar de que aparezca «Capítulo 1: Título elegido» salga «Jornada 1: Título elegido». ¿Cómo hacemos para renombrar estos entornos?

La forma más simple es ir al preámbulo  $\text{\LaTeX}$  y agregar un poco de código:

```
\usepackage[spanish]{babel}  
\addto\captionsspanish{  
  \renewcommand{\partname}{Libro}
```

```
\renewcommand{\chaptername}{Jornada}
}
```

Básicamente estamos «engañando» a L<sup>A</sup>T<sub>E</sub>X diciéndole que en lugar de utilizar las traducciones predefinidas para Part y Chapter (Parte y Capítulo, respectivamente) utilice las que nosotros queremos.

De hecho si no queremos nada allí simplemente dejamos las llaves vacías: `\renewcommand{\partname}{}{}` nos mostrará solo el número y el título que escribamos, sin etiquetas.

En la ventana de edición de L<sup>y</sup>X nada cambiará, pero en la salida PDF todo se mostrará como queremos.

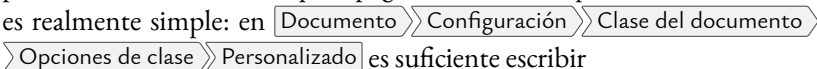
## 11.2 Numerar capítulos con números romanos

Para hacer que los capítulos estén numerados con números romanos debemos modificar el «contador» de los mismos aplicándole un «formato» y para esto utilizaremos en comando `renewcommand`:

```
\renewcommand{\thechapter}{\Roman{chapter}}
```

Para utilizar letras en mayúscula cambiamos Roman por Alph, tal y como vimos en el capítulo 5.

## 11.3 Hacer que los capítulos comiencen en cualquier página

Por defecto en las *documentclass* de libros los capítulos comienzan siempre en una página nueva que es además una página «derecha» (impar). Para quien se perdió el fugaz comentario en el capítulo 4, lograr que los capítulos comiencen en cualquier página, sea esta «izquierda» o «derecha» es realmente simple: en  es suficiente escribir

`openany`



## 11.4 Evitar los quiebres de página al comenzar un capítulo

No es que me guste, pero ya saben, por completitud.

Supongamos que por algún motivo queremos evitar completamente que se produzca un quiebre de página al comienzo de cada capítulo para así lograr que un capítulo comience en la misma página donde ha terminado el anterior. Para lograr esto tendremos que redefinir el entorno capítulo por lo que dirigiéndonos al preámbulo  $\LaTeX$  escribimos

```
\makeatletter
\renewcommand\chapter{%
    \global\@topnum\z@
    \@afterindentfalse
    \secdef\@chapter\@schapter}
\makeatother
```

Este código lo vi aquí:

<http://compgroups.net/comp.text.tex/latex-report-class-page-break-after-every-chapt/1924905>

## 11.5 Cambiar el formato de los títulos en $\LaTeX$

Como ya hemos comentado el *documentclass* elegido determina el formato general del documento y esto ciertamente incluye cómo se muestran los títulos de capítulos, secciones, etcétera.

En la mayor parte de los *documentclass* se utiliza para los títulos la misma fuente del cuerpo de texto. Esto generalmente da buenos resultados estéticos, ¿pero qué tal si deseamos cambiar este formato? ¿O cambiar el color del texto? ¿O agregar «efectos especiales»? Por ejemplo, podríamos querer una «caja» rodeando el título o utilizar un color diferente.

Existen varias formas de lograr cambios en el formato de los títulos y algunos *documentclass* como *Koma Script* ofrecen herramientas para lograr algunas de estas cosas, pero la opción que presentaré a continuación es quizás la más general posible (si bien no es completamente general ya que no funciona con la clase de documento *Memoir*).

### 11.5.1 Los paquetes a utilizar

Generalmente se instalan por defecto, pero no está de más confirmar que los paquetes `xcolor` y `titlesec` estén instalados en nuestra distribución  $\text{\LaTeX}$ .

La documentación de `xcolor` suele instalarse, en Linux y con `texlive`, en

```
/usr/share/texmf/doc/latex/xcolor/  
mientras que la documentación de titlesec se instala en  
/usr/share/texmf/doc/latex/titlesec/
```

Daré aquí indicaciones generales sobre estos paquetes que resultarán más que suficientes en la mayor parte de los casos. Para más información y otros usos extremos, siempre siempre es posible leer el maldito manual...

#### 11.5.1.1 `xcolor`

El paquete `xcolor` sirve para definir, sí, lo han adivinado, colores. La forma más simple de utilizarlo es declararlo en el preámbulo  $\text{\LaTeX}$  con, por ejemplo

```
\usepackage{xcolor}  
\definecolor{AzulClaro}{rgb}{.3,.5,.7}
```

La primera línea declara que queremos utilizar paquete en nuestro documento mientras que la segunda define un color: los parámetros entre llaves son, respectivamente, el nombre del color, el método de definición y los «valores» de cada color básico en forma de tres números entre cero y uno (cuánto rojo, cuánto verde y cuánto azul), separados por comas.

#### 11.5.1.2 `titlesec`

Este paquete ofrece varias instrucciones para realizar los cambios en los títulos. Se puede utilizar en dos formas, una «simplificada» y otra «completa».

La forma simplificada solo puede utilizarse en secciones, subsecciones y niveles más bajos mientras que para los entornos Parte y Capítulo (o para disponer de toda la pirotecnia que ofrece el paquete) debe utilizarse la forma completa.

### 11.5.2 Un ejemplo completo

En el preámbulo  $\LaTeX$ :

```
\usepackage{titlesec}
\usepackage{xcolor}
\definecolor{AzulClaro}{rgb}{.3,.5,.7}
\definecolor{Veeerde}{rgb}{.3,.7,.3}
\newfontfamily{\coral}{TeX Gyre Chorus}
\titleformat{\chapter}[frame]
  {\LARGE\coral\color{AzulClaro}}{
  \filright\chaptertitlename\
  \thechapter}{20pt}{\Huge}
\titleformat*{\section}{
  \Large\bfseries\sffamily
  \color{Veeerde}}
```

Las dos primeras líneas declaran el uso de los paquetes `titlesec` y `xcolor`.

Las dos siguientes definen dos colores, `AzulClaro` y `Veeerde` (no, no fue un error de escritura).

La quinta línea define una nueva familia tipográfica para utilizar en el documento. Ya hemos hablado de eso en la sección 6.2.

En el ejemplo puede verse tanto la forma simple como la completa de la instrucción `\titleformat`, parte del paquete `titlesec`.

La forma con el asterisco es la «versión resumida» de la instrucción y ha sido utilizada para definir rápidamente el formato de la sección: aplicada al entorno `\section`, usa una fuente en tamaño `\Large`, en negrita, con la fuente palo seco definida para el documento (`\sffamily`) y le asigna el color *Veeerde* que hemos definido.

La forma completa de la instrucción usada para modificar el entorno `\chapter` (capítulo). Comienza indicando que se debe construir un marco alrededor del título (`[frame]`: si no queremos el marco debemos escribir `[display]`). Luego tenemos el tamaño `\LARGE` para el texto de la etiqueta (la palabra Capítulo seguida del número), la familia de fuente que llamamos *coral*, el color *AzulClaro*. Luego vienen una serie de instrucciones de formato que comienzan con «alinear a la derecha», siguen dando la estructura



Texto general con <http://es.lipsum.com/>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque in massa vel risus facilisis finibus. Duis auctor mi quam, sit amet vestibulum tortor iaculis at. Ut malesuada nisi vel molestie commodo. Sed non libero dolor. Aenean ac cursus est. Vestibulum et ultricies neque. Suspendisse ac tincidunt nisi, ac dignissim ligula.

## 1.1 Título de sección

Nulla sagittis, purus sed suscipit bibendum, lacus purus placerat quam, id egestas libero dolor in ligula. Donec molestie tincidunt neque id fringilla. Suspendisse egestas tempor feugiat. Aliquam placerat mi vel metus dapibus imperdiet. Aliquam finibus sapien felis, non

FIGURA 11.1: *Un ejemplo de títulos modificados con la instrucción `titlesec`*

de la información presentada (incluyendo un espacio de separación entre etiqueta y texto del título) y terminan indicando que el texto del título debe tener el tamaño `\Huge` que es decididamente mayor que la etiqueta. Podemos ver el resultado de estas instrucciones en la captura de pantalla de la figura 11.1.

Claramente esto también puede hacerse para el entorno «parte», agregando por ejemplo:

```
\titleformat{\part}[display]
{\huge\cormorant}{\filright\partname\
\thepart}{20pt}{\Huge}
```

Aquí no hemos utilizado el «marco» y se ha elegido un tamaño mayor para la etiqueta de la parte (además de utilizar una familia de fuentes definida en otra parte).

Y bien, los ingredientes básicos ya están por lo que solo queda compilar el documento. Las instrucciones aquí mostradas son aún más potentes, pero no entraré en más detalles: para eso ya existe la documentación de cada paquete.

## 11.6 Reiniciar la numeración luego de una «Parte»

Por defecto  $\LaTeX$  no reinicia la numeración de los capítulos luego de un entorno Parte, dando una numeración «continua»:

- Parte I
  - Capítulo 1
  - Capítulo 2
- Parte II
  - Capítulo 3

Si bien esta opción por defecto me parece lógica y de hecho la uso (en mi libro sobre Writer propongo trucos para conseguirla allí), es perfectamente posible y por lo tanto completamente válido que alguien necesite la opción contraria, es decir

- Parte I
  - Capítulo I.1
  - Capítulo I.2
- Parte II
  - Capítulo II.1

La forma más simple de conseguir esto es mediante el paquete `chngcntr`. En  $\TeX$  Live este paquete se instala como

```
texlive-chngcntr
texlive-chngcntr-doc
```

La documentación, al menos en Linux, se instala en

`/usr/share/texmf/doc/latex/chngcntr/chngcntr.pdf`

Son solo dos páginas, por lo que recomiendo leerla

Para utilizar este paquete, en el preámbulo  $\LaTeX$  escribimos

```
\usepackage{chngcntr}
\counterwithin{chapter}{part}
```

La primera línea activa el paquete mientras que la segunda dice que `chapter` debe ser contado dentro de `part`, es decir, debe reiniciar la numeración cada vez que `part` cambia.

Listo, ¿no? No: al agregar la «parte» a la numeración de los capítulos (y de las figuras, y de las tablas) tendremos problemas con el índice, ya que el número se quedará sin espacio y se superpondrá con el texto del título. Ya habíamos resuelto esto para las listas de figuras y cuadros (en el capítulo [10](#)), por lo que siguiendo aquel ejemplo luego de las dos líneas anteriores también tendremos que agregar en el preámbulo lo siguiente:

```
\makeatletter
\renewcommand\l@chapter{
  \@dottedtocline{2}{1.5em}{3em}}
\makeatother
```

Para agregar espacio en las líneas del índice correspondientes a las secciones duplicamos la segunda instrucción mostrada más arriba, pero cambiando `l@chapter` por `l@section`:

```
\makeatletter
\renewcommand\l@chapter{
  \@dottedtocline{2}{1.5em}{3em}}
\renewcommand\l@section{
  \@dottedtocline{2}{1.5em}{3em}}
\makeatother
```

Y así siguiendo con las subsecciones (`subsection`), etcétera. Ahora sí, ¡listo! Aunque no es algo que verán en este libro.

## Capítulo 12

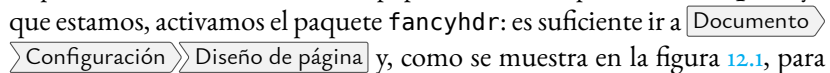
# Cabeceras y pie de página, estilos de página

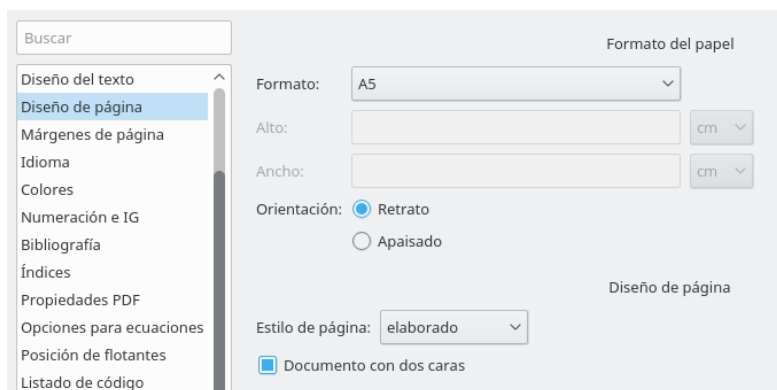
En documentos creados con  $\text{\LaTeX}$  y derivados la forma de modificar los valores por defecto de cabeceras y pie de página es utilizar el paquete `fancyhdr`.  $\text{LyX}$  ofrece soporte parcial para este paquete en su interfaz gráfica, pero de tan parcial resulta poco práctico por lo que solo nos queda arregármolos e ir al preámbulo  $\text{\LaTeX}$ .

Las cabeceras y pie de página del documento están asociadas a un número de estilos de página, tema del que hablaremos sobre el final de este capítulo. Comenzaremos por lo tanto hablando sobre cómo cambiar el contenido de las cabeceras y pie de las páginas del «cuerpo del documento» para luego llegar a la creación y el uso de estilos especiales.

### 12.1 Preparativos

La documentación de `fancyhdr` se instala, en Linux y con  $\text{\TeX Live}$ , en `/usr/share/texmf/doc/latex/fancyhdr/`

Por defecto los documentos  $\text{\LaTeX}$  están pensados como si todos tuviéramos impresoras simples que no pueden dar vuelta la página, por lo que si queremos usar las dos caras del papel tendremos que avisarle a  $\text{LyX}$ . Y ya que estamos, activamos el paquete `fancyhdr`: es suficiente ir a  y, como se muestra en la figura 12.1, para «Estilo de página» seleccionar elaborado y marcar la opción «Documento

FIGURA 12.1: *Activar fancyhdr*

con dos caras».

Ahora sí, en este mismo menú nos dirigimos al Preámbulo  $\LaTeX$  y comenzamos a escribir... pero antes, una pequeña introducción a *qué* tenemos que escribir.

## 12.2 El paquete **fancyhdr**

El paquete fancyhdr permite cambiar el contenido tanto de las cabeceras (head) como de los pie de página (foot). Ambos sectores de la página están divididos en tres partes: izquierda (left o l/L), central (center o c/C) y derecha (right o r/R)

Existen dos modos de utilizar este paquete. La forma más directa, útil para documentos «de un solo lado», es con las instrucciones

```
\head{contenido de la cabecera a la izquierda}
\chead{contenido de la cabecera al centro}
\rhead{contenido de la cabecera a la derecha}
\lfoot{contenido del pie de página a la izquier-
da}
\cfoot{contenido del pie de página al centro}
\rfoot{contenido del pie de página a la derecha}
```



Más abajo veremos algunos ejemplos del tipo de contenido que podemos utilizar.

La forma más potente de este paquete es con la instrucción `\fancyhead` para las cabeceras y `\fancyfoot` para los pie de página, teniendo en cuenta la siguiente convención:

R	representa «derecha» (por right)
L	representa «izquierda» (por left)
C	representa «centro» (por center)
O	representa las páginas impares (por odd)
E	representa las páginas pares (por even)

La instrucción

```
\fancyhead[R0,LE]{CONTENIDO}
```

colocará por lo tanto CONTENIDO a la derecha en las páginas impares y a la izquierda en las pares. Si escribimos

```
\fancyfoot{}
```

tendremos el pie de página vacío.

Ahora bien, ¿qué es lo que podemos poner en CONTENIDO?

Para empezar, el texto puro funciona perfectamente. De hecho, si activamos  $\text{X}_{\text{T}}^{\text{E}}\text{X}$  y definimos una nueva familia de fuentes (como comentamos en la sección 6.2) podemos colocar texto tan «extraño» como queramos en las cabeceras/pie de página (más abajo, un ejemplo).

También tenemos instrucciones para insertar el número de página y el capítulo. Por ejemplo:

```
\thepage
```

inserta el número de página

```
\leftmark
```

```
\rightmark
```

se usan para información del capítulo/sección (ver ejemplo), etcétera.

NOTA: si por algún motivo falta información para una de estas «marcas» (por ejemplo, en un capítulo determinado no tenemos secciones), pues tendremos un espacio vacío en las páginas correspondientes.

En general, el nombre del capítulo aparecerá en mayúsculas. Para evitar esto podemos utilizar la instrucción `\noupเปอร์case{}` (la veremos en acción en el ejemplo). Se debe tener cuidado con esta instrucción ya que si hemos numerado con números romanos, pues estos también podrían aparecer en minúsculas... Sí,  $\text{\LaTeX}$  puede ser extraño.

Usando estos comandos, una línea de separación se mostrará. Para modificar el espesor de la misma usamos

```
\renewcommand{\headrulewidth}{0.4pt}
```

Escribiendo `opt` (cero puntos), la línea desaparece.

## 12.3 Un ejemplo

Comencemos a poner las cosas todas juntas:

```
\newfontfamily{\coral}{TeX Gyre Chorus}
\renewcommand{\headrulewidth}{0.4pt}
\fancyhead[R0,LE]{\thepage}
\fancyhead[L0]{\noupเปอร์case{\leftmark}}
\fancyhead[RE]{\coral{El pingüino tolkiano}}
\fancyfoot{}
```

Aquí tendremos el número de página a la derecha en las páginas impares y a la izquierda en las pares, el nombre del capítulo a la izquierda en las impares y el texto «El pingüino tolkiano», formateado con la fuente TeX Gyre Chorus, a la derecha de las pares.

Ahora bien, el comando `\leftmark` escribe «Capítulo #. Nombre del capítulo». Si queremos que solo aparezca el nombre del capítulo sin el número, también en el preámbulo tenemos que escribir

```
\renewcommand{\chaptermark}[1]{  
\markboth{#1}{}}
```

mientras que si queremos que la sección aparezca sin numerar

```
\renewcommand{\sectionmark}[1]{  
\markright{#1}}
```

Si queremos que el nombre del capítulo también aparezca en la nueva familia de fuente, solo es necesario rodear `\leftmark` con la correspondiente instrucción. Siguiendo con nuestro ejemplo, sería

```
\fancyhead[L0]{\nouppercase{\coral{\leftmark}}}
```

Como siempre, el manual de `fancyhdr` nos dará (¡muchísima!) más información, como por ejemplo la forma correcta de incluir una imagen en la cabecera o el pie de la página.

## 12.4 Crear y utilizar estilos de página

A partir de ahora hablaremos sobre cómo definir y utilizar diferentes estilos de página, resolviendo de paso ~~algunos problemas~~ algunas «características no siempre deseadas» que se nos presentan con las opciones por defecto en la mayor parte de las clases de documento de libros.

Todo esto será con código `LATEX` puro y duro. Qué se le va a hacer.

Supongamos que necesitamos crear un documento con cabeceras de página elaboradas y con esta estructura como la siguiente

1. Índice con páginas numeradas en números romanos (¡en mayúsculas!)
2. Un «capítulo» no numerado que contenga una introducción a la obra
3. Primera parte
4. Primer capítulo numerado

## 5. ...

Sobre la numeración de las páginas en los índices ya hemos hablado (capítulo 10), como así también introdujimos las herramientas para trabajar con las cabeceras y pies de página unas líneas más arriba. Para tener los entornos Parte y Capítulo necesitamos una de las clases de documento *Book*. Un capítulo no numerado se crea seleccionando el entorno Capítulo\* (con el asterisco), por lo que todo parece fácil.

El problema con todo esto, porque si no se presentaran problemas no existiría esta sección, tiene varias partes. Para empezar la introducción creada con el capítulo no numerado no es incluida en el índice y para seguir la información disponible para la cabecera/pie de página de la introducción mostrará «Índice general» en lugar de «Introducción», lo cual es razonable si pensamos en el primer problema: la cabecera de la página no actualiza su información porque al ser el capítulo «no numerado» (y por lo tanto «no indexado») no hay nueva información disponible.

Para resolver esto será necesario crear un estilo de página con la información que nos sirva y decirle manualmente a  $\text{\LaTeX}$  que debe agregar una entrada más al índice.

Para comprender cómo funciona esto, demos un ejemplo que podamos analizar en detalle. En el preámbulo  $\text{\LaTeX}$  debemos escribir lo siguiente:

```
% el estilo de página normal, con sus cabeceras
\pagestyle{fancy}
\fancypagestyle{normal}{%
  \fancyhead{} % limpia las cabeceras para dar
                % lugar a lo que sigue
  \fancyhead[R0,LE]{\thepage}
  \fancyhead[L0]{\nouppercase{\leftmark}}
  \fancyhead[RE]{\nouppercase{Nuestro gran li-
bro}}
  \fancyfoot{}
}
% el estilo de página especial
\fancypagestyle{especial}{%
  \fancyhead{} % limpia las cabeceras para dar
```

```

% lugar a lo que sigue
\fancyhead[R0,LE]{\thepage}
\fancyhead[L0]{\nouppercase{Introducción}}
\fancyhead[RE]{\nouppercase{Nuestro gran li-
bro}}
\fancyfoot{}
}
% el estilo de página del índice
\fancypagestyle{index}{%
  \fancyhead{} % limpia las cabeceras para dar
                % lugar a lo que sigue
  \fancyhead[R0,LE]{\thepage}
  \fancyhead[L0]{\nouppercase{Índice}}
  \fancyhead[RE]{\nouppercase{Índice}}
  \fancyfoot{}
}

```

Con esto hemos redefinido el estilo de página «normal» y hemos creado dos estilos nuevos llamados «especial» e «index» que utilizaremos en el capítulo no numerado y en el índice, respectivamente, dándole a todos la información que nos sirve.

Ahora, al principio del documento donde queremos el índice de contenido insertamos una «caja T<sub>E</sub>X» y en ella escribimos

```

\frontmatter
\setcounter{page}{1}
\pagestyle{index}

```

(la segunda línea suele no ser necesaria). Con esto le decimos a L<sup>A</sup>T<sub>E</sub>X que lo que sigue a continuación es el espacio dedicado al índice con el estilo de página que definimos para el índice y que tiene que numerar las páginas con números romanos.



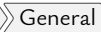
Si queremos insistir en que estos números romanos sean en mayúsculas podríamos agregar

```

\pagenumbering{Roman}

```

Solo para estar seguros.

Debajo de la caja hacemos    y a continuación del mismo insertamos otra caja T<sub>E</sub>X para en ella escribir

```
\mainmatter
\setcounter{page}{1}
```

(esta segunda línea tampoco suele ser necesaria). Este código indica que ahora «comienza lo bueno», que ya estamos en el argumento principal del libro.

Pues bien, ahora insertamos el entorno «capítulo no numerado» utilizando el menú para elegir entornos y escribimos el título que nos sirva, por ejemplo, «Introducción».

*¡Recuerden utilizar el mismo título que definieron para el estilo de página «especial»!*

En el renglón *inmediatamente debajo de este título* insertamos una nueva caja T<sub>E</sub>X con el siguiente contenido

```
\pagestyle{especial}
\addcontentsline{toc}{chapter}{Introducción}
```

La primera línea asigna el estilo de página «especial» a aquello que sigue, mientras que la segunda agrega una línea en el índice de contenido apuntando a esa página, mostrándose en el nivel asignado (*chapter*) y con el texto elegido (Introducción, aquí también es importante recordarse de mantener la coherencia).

Terminada la escritura de la introducción, pues es momento de seguir adelante. Luego de aplicar (si nos sirve) el entorno «Parte» y a continuación el entorno «Capítulo» (*sin* el asterisco, para que sea numerado), agregamos otra caja T<sub>E</sub>X con el siguiente código

```
\pagestyle{normal}
```

así todo vuelve «a la normalidad».

El resto del documento se escribe normalmente, ya no necesitamos más «cajas T<sub>E</sub>X». Al menos no para estos problemas.

Algunos enlaces que me han servido para introducirme al tema:

Crear y utilizar estilos de página en  $\text{\LaTeX}$

<http://tex.stackexchange.com/questions/37574/different-headers-for-different-chapters>

Agregar al índice un capítulo no numerado:

<http://wiki.lyx.org/FAQ/TOC#addcontentsline>

## 12.5 Primera página del capítulo

Por defecto en las clases de documento de libros la primera página de cada capítulo utiliza el estilo «plain». Si queremos aplicar un estilo diferente tenemos que agregar, en el preámbulo  $\text{\LaTeX}$ , algo como lo siguiente

```
\fancypagestyle{primera}{%
  \fancyhead{}
  \fancyhead[R]{\nouppercase{Nuestro gran li-
bro}}
  \fancyfoot[C]{Página \thepage}
}
\makeatletter
\let\ps@plain\ps@primera
\makeatother
```

Las primeras líneas definen un nuevo estilo de página llamado «primera» mientras que las tres últimas (en realidad, solo la central) cambia el estilo de página de «plain» a nuestro estilo «primera». Con este código la primera página de cada capítulo mostrará, a la derecha de la cabecera, el texto «Nuestro gran libro» mientras que en el centro del pie muestra la palabra «Página» seguida del número de página.

## 12.6 Páginas en blanco

Si creamos un documento «a dos lados» con alguna de las clases de documento *book* terminaremos seguramente con varias páginas en blanco insertadas en forma automática: los capítulos inician en páginas «derechas»

(impares), por lo que si el capítulo anterior terminó también en página impar en medio tendremos una página en blanco.

El problema es que por defecto  $\LaTeX$  sigue con el mismo esquema de cabeceras/pie de página de todo el capítulo, por lo que la página en blanco no estará *realmente* en blanco: tendrá las cabeceras/pie de página que hemos configurado.

Para hacer que las páginas en blanco estén realmente en blanco tenemos que instalar el paquete `emptypage`. En  $\TeX$  Live este paquete se llama

`texlive-emptypage`

Una vez instalado, nos dirigimos como de costumbre al Preámbulo  $\LaTeX$  (a esta altura, lector, ya sabrás llegar allí) y agregamos la siguiente línea:

```
\usepackage{emptypage}
```

Ahora bien, ¿qué tal si no queremos que la página esté completamente en blanco? Podríamos necesitar que diga «página en blanco» (es necesario para algunos documentos legales que las páginas en blanco estén indicadas como tales).

Si miramos la documentación del paquete `emptypage` notaremos que lo único que hace es redefinir las páginas en blanco para que utilicen el estilo «empty» por lo que solo tenemos que tomar el ejemplo y escribir, en el preámbulo  $\LaTeX$ , lo siguiente:

```
\fancypagestyle{raro}{%
  \fancyhead{} %
  \fancyhead[LE]{\thepage}
  \fancyhead[RE]{\nouppercase{Página vacía}}
  \fancyfoot[CE]{Página vacía}
}
\def\emptypage@emptypage{%
  \hbox{}%
  \thispagestyle{raro}%
  \newpage%
  \if@twocolumn%
```



```
\hbox{}}%
\newpage%
\fi%
}
\def\cleardoublepage{%
\clearpage%
\if@twoside%
\ifodd\c@page%
\else%
\emptypage@emptypage%
\fi%
\fi%
}
```

Queda como tarea para el lector la interpretación del código y su posible modificación para adaptarlo a otros formatos deseados.

## 12.7 Cambiar el estilo de página del entorno «Parte»

En un apartado anterior vimos cómo crear y aplicar un nuevo estilo de página en un documento  $\text{\LaTeX}$  para así lograr cambiar el contenido de la cabecera o pie de página de una parte del documento. Ahora enfrentaremos un problema similar, pero cuya solución es más compleja: cambiar el estilo de página asociado al entorno Parte.

El entorno Parte, utilizado para agrupar capítulos en las clases de documento «libro», tiene asociado un estilo de página particular el cual podría tener, dependiendo de la clase de documento elegida, un pie de página con el número de página. Supongamos que queremos eliminar ese número de página, ¿podemos utilizar lo visto más arriba? Desafortunadamente, no.

El problema es que el entorno Parte está también asociado con quiebres de página, por lo que cualquier instrucción del tipo `\thispagestyle` que utilicemos antes o después del entorno será aplicada a la página anterior o siguiente y no a la que nos interesa. Tampoco podemos colocar la instrucción `\thispagestyle` dentro del entorno Parte ya que esta instrucción no puede estar anidada. ¿Entonces? Pues que hay que «parchear» el entorno Parte.

El siguiente código lo encontré en este tema del foro *L<sup>A</sup>T<sub>E</sub>X Community* <http://www.latex-community.org/forum/viewtopic.php?f=47&t=5809> y básicamente lo que hace es cambiar la definición del entorno `\part` (Parte) asignándole el estilo de página «empty», que no tiene cabeceras ni pie de página.

La complejidad del código está en que «se cubre» de posibles situaciones, teniendo en cuenta la posibilidad de que el documento sea a una cara o dos, en una o múltiples columnas: de allí los comandos «if».

Para utilizarlo en L<sup>y</sup>X, nos dirigimos al Preámbulo L<sup>A</sup>T<sub>E</sub>X y escribimos lo siguiente:

```
% esto hace que las páginas \part
% queden sin pie de página
\makeatletter
\renewcommand\part{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \thispagestyle{empty}%
  \if@twocolumn
    \onecolumn
    \@tempswattrue
  \else
    \@tempswafalse
  \fi
  \null\vfil
  \secdef\@part\@spart}
\makeatother
```

Listo. Y sí, L<sup>A</sup>T<sub>E</sub>X puede ser *mu*y extraño.

NOTA: Existe una forma más simple, pero por algún motivo no funciona siempre (por ejemplo, no funciona en este libro). Esta forma consiste en «parchear» el entorno «part» usando el comando `etoolbox` de la siguiente forma

```
\usepackage{etoolbox}
\patchcmd{\part}{\thispagestyle{plain}}{
  \thispagestyle{empty}}
  {}{\errmessage{No puedo par-
chear \string\part}}
```

Pero lo dicho: no siempre funciona por lo que muy probablemente al compilar un documento obtengamos el mensaje de error que hemos definido en la tercera línea. Es lo que hay.

Ahora bien, ¿qué hacer con el estilo de página que *sigue* al del entorno «parte». Después de todo en este libro tenemos allí, en esas páginas que suelen estar vacías, diferentes frases.


```
\makeatletter
\def\@endpart{\vfil\newpage
  \if@twoside
  \if@openright
  \null
  \thispagestyle{empty}%
\vspace*{3cm}%
\begin{quote}%
\partnote
\end{quote}%
  \fi
  \fi
  \if@tempswa
  \twocolumn
  \fi}
\newcommand\partnote{}
\makeatother
```

Aquí tenemos varias cosas. Por una parte se redefine el «endpart», el final de la «parte», por otra se define un entorno de «cita» con todo y espacio inicial y se asigna un estilo de página a todo esto.

El lector curioso podrá descubrir cómo creé el estilo de carácter para la «frase» en el apéndice [B](#). A esta altura no debería resultar difícil el interpretar todos esos códigos.


## Capítulo 13

# Letras capitales y afines

 EGURAMENTE el lector habrá encontrado documentos donde sucede lo que ve en este párrafo particular: la primera letra de la primera palabra en la primera línea se extiende por varios renglones, mostrando quizás (como aquí) una fuente especial y adornada, con el resto de esa primera palabra escrito en versalitas.

Este estilo es lo que se conoce como «letra capitular», «letra capital» o simplemente «inicial». Para hacer esto en  $\text{\LaTeX}$ /LyX es necesario utilizar el paquete `lettrine`<sup>1</sup>.

### 13.1 Algo tradicional

 N LyX tenemos soporte para este paquete desde la interfaz gráfica y en la sección 6.3 del manual que encontramos en el menú `Ayuda > Objetos insertados` se explica claramente una de las formas de utilizarlo. Porque como veremos a continuación existe *otra* forma, la cual resulta mucho más simple. Y es que lo que se cuenta en el manual nos ofrece toda la flexibilidad que queramos a la hora de elegir la forma de la letra capital cada vez que la necesitemos, ¿pero qué tal si queremos utilizar letras capitales varias veces (por ejemplo, en el primer párrafo de cada capítulo) y que siempre se vean igual? Hacerlo de la forma indicada en el manual de LyX podría volverse un tanto incómodo por lo que resultaría

---

1. Sin chistes fáciles, por favor.

conveniente encontrar un modo de «automatizar» el proceso.

Para empezar y como siempre, debemos estar seguros de que el paquete que nos interesa esté instalado en nuestro  $\text{\LaTeX}$  y de no ser así instalarlo y luego reconfigurar  $\text{\LyX}$  para que lo reconozca. La documentación del paquete *Lettrine* en Linux se instala en

```
/usr/share/texmf/doc/latex/lettrine/lettrine.pdf
```

```
/usr/share/texmf/doc/latex/lettrine/demo.pdf
```

Ahora nos dirigimos a Documento » Configuración » Módulos (ya hablamos de los módulos en el capítulo 8) y habilitamos el módulo *Capitales*.

¡Clic en Aplicar, no Aceptar, que aún no hemos terminado con este menú! Y es que, como suele suceder en estos casos, debemos ir al Preámbulo  $\text{\LaTeX}$  para agregar algo de código.

Este documento está configurado para utilizar  $\text{\XeTeX}$  con la fuente EB Garamond como fuente de base y define un «alias» de la fuente Pali<sup>2</sup> utilizando las «alternativas estilísticas» (+salt) para que sirva de letra capital.

Esto es lo que he escrito en el preámbulo:

```
\newfontfamily{\Iniciales}[
  RawFeature={+salt}]{Pali}
\renewcommand{\LettrineFontHook}{\Iniciales}
\setcounter{DefaultLines}{4}
```

La primera instrucción define una nueva «familia de fuente» (ya hablamos sobre esto en la sección 6.2), definición que es utilizada en la segunda instrucción para decir al paquete *Lettrine* cuál fuente queremos para las letras capitales. La tercera instrucción dice cuántas líneas debe ocupar la letra capital.

Es posible controlar la posición vertical del carácter agregando una cuarta línea

```
\renewcommand{\DefaultLraise}{0.35}
```

donde el valor «0.35» puede modificarse a aquello que sirva. Generalmente no es necesario utilizar esta instrucción.

2. <http://www.softerviews.org/Fonts.html#Pali>

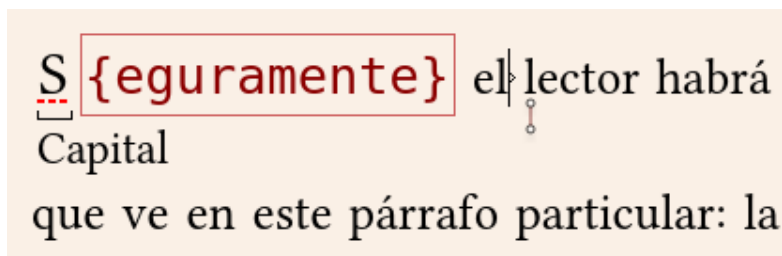







FIGURA 13.1: *Letra capital en la interfaz gráfica de LyX*

Hecho esto, cuando queremos insertar una nueva letra capital seleccionamos el primer carácter del párrafo en cuestión y hacemos   , mientras que para tener el resto de la primera palabra en versalitas debemos encerrarla entre llaves y meter todo (resto de palabra y llaves) en una «caja T<sub>E</sub>X» (se selecciona el texto y se presiona  ). En la interfaz gráfica de LyX esto se verá como lo que muestra la figura 13.1.



La documentación del paquete `Lettrine` es breve por lo que el lector no tendrá problemas en encontrar otras opciones para ajustar todo a sus preferencias en forma rápida y simple.

## 13.2 ¡Atención al número de líneas!

PRECISAMENTE, debemos estar atentos.

Y es que algo muy importante a tener en cuenta es que si el párrafo sobre el cual activamos la letra capitular es breve (por ejemplo, una o dos líneas) y la inicial es grande (por ejemplo, cuatro) *la letra inicial se superpondrá al párrafo siguiente*.

Algo que, ciertamente, no ha sucedido aquí. ¿El truco? Pues unir el párrafo con la inicial al siguiente simulando el quiebre de párrafo con un quiebre de línea.

En L<sup>A</sup>T<sub>E</sub>X los quiebres de línea se introducen con dos barras invertidas consecutivas «`\\`» (sin las comillas), pero en LyX es mucho más simple: presionamos  . En la ventana de edición se verá una flecha quebrada roja.

Al ser todo parte de un mismo párrafo ya no se produce la molesta su-

perposición, *pero el segundo «párrafo» no muestra sangría*. Algo que aquí sí logramos. ¿Truco? Aquí se presenta un pequeño problema, porque los comandos normales de  $\text{\LaTeX}$  para agregar espacios horizontales no funcionan al principio de un párrafo, ¿entonces? Insertar Formato Fantasma Fantasma horizontal y en la caja que se nos presenta escribimos, por ejemplo, dos «m» las cuales se mostrarán en la ventana de edición tachadas y entre paréntesis angulares mientras que en el resultado nos darán lo que se muestra al inicio de esta subsección.

### 13.3 Una variante: letra capital «alzada»



QUÍ el «truco» está en una pequeña variante del método presentado anteriormente.

En el preámbulo  $\text{\LaTeX}$  (por supuesto, luego de activar el módulo Capitales y todo lo que ya conté antes) tenemos que escribir lo siguiente:


```
\newfontfamily{\Iniciales}[
  Scale=2,RawFeature={+salt}]{Pali}
\renewcommand{\LettrineFontHook}{\Iniciales}
\setcounter{DefaultLines}{1}
```

Como puede verse estamos definiendo todo casi de la misma forma que en la sección anterior, pero hemos agregado una instrucción para hacer que la fuente definida en la nueva familia `\Iniciales` se muestre al doble de su tamaño (claramente podemos siempre jugar con el parámetro de la instrucción `Scale`) mientras que el número de líneas lo he cambiado a 1.

Ya con esto es suficiente para lograr que la mayúscula inicial no «baje» sino que se muestre, más grande, en la misma línea del texto.



## 13.4 Un desafío

USTAMENTE en este punto dejo al lector con un desafío: combinar las secciones previas para lograr una «inicial» que tome varias líneas (digamos, dos), pero que también suba lo suficiente como para lograr lo que se ve en este párrafo.

Vamos, que con lo visto debería ser sencillo.

Y no, no verán la solución en el apéndice con el código del preámbulo `ÆTEX` de este documento ya que esta letra inicial la he creado «en el lugar» utilizando el entorno «capital» con opciones introducidas a mano *solo para ese párrafo*.

¡No es posible hacer trampas!

## 13.5 Una alternativa: primera línea de un párrafo en versalitas

UNA ALTERNATIVA A LAS LETRAS CAPITULARES ES QUE LA PRIMERA línea del primer párrafo se encuentre toda ella en versalitas, algo que puede verse principalmente en revistas (además de en este párrafo).

El único modo de obtener este efecto en forma (semi) automática con `ÆTEX` que he encontrado es con el paquete `magaz`.

La documentación de este paquete se instala, en Linux y con `TEXLive`, en

```
/usr/share/texmf/doc/latex/magaz/magaz.pdf
```

Este paquete es un tanto «caprichoso» ya que no solo se lleva mal con el paquete `lettrine`, tampoco acepta ecuaciones, formatos particulares o tan siquiera eliminar la sangría de primera línea que introduce. Pero en su forma más simple es muy fácil de utilizar.

En el preámbulo `ÆTEX` escribimos

```
\usepackage{magaz}
```

y luego, en el párrafo que queremos modificar, abrimos dos «cajas  $\text{\TeX}$ », una al comienzo del párrafo con la instrucción

```
\FirstLine{
```

y la otra más adelante (no necesariamente al final del párrafo, solo debe tomarse suficiente texto como para cubrir la primera línea) con un

```
}
```

para cerrar la instrucción.

El paquete cambiará automáticamente el formato de la primera línea sin más trabajo por parte del escritor, cuidando incluso los casos en los que la última palabra de la primera línea se quiebre, como sucede aquí.

Es importante notar que la instrucción `FirstLine` debe estar al comienzo del párrafo o de lo contrario nos dará error.

Este paquete ofrece más opciones. Por ejemplo es posible seleccionar otro formato para la fuente utilizada en la primera línea con la instrucción

```
\renewcommand\FirstLineFont{\<instrucción de formato>}
```

pero como la breve documentación del paquete es lo suficientemente clara no entraré en esos detalles.

## Capítulo 14

# Figuras y otros flotantes

Ya hemos introducido el concepto de «flotantes» en la subsección 4.1.7: un objeto dentro del cual colocamos figuras o cuadros y que será manejado por  $\text{\LaTeX}$  para colocarlo en la posición correcta a la hora de compilar el documento.

En este capítulo veremos algunas de las opciones que tenemos disponibles para modificar el formato tanto de estos flotantes como de su contenido.

### 14.1 Ajustar tamaño de las imágenes insertadas

Cuando insertamos una imagen en  $\text{\LaTeX}$  esta quedará con su tamaño real, por lo que si es grande al compilar el documento nos daremos cuenta de que ha «perforado el margen». De hecho, dependiendo de la resolución de la imagen y de qué tan grande sea más que perforarlo podría *demolerlo*.

Esto puede resolverse en dos formas, una manual y otra semiautomática. Veamos ambas.

#### 14.1.1 Ajustar el tamaño de la imagen manualmente

Ya sea mientras la insertamos o luego al hacer clic sobre la imagen se nos abrirá el diálogo mostrado en la figura 14.1.

Como puede verse, al marcar *Redimensionar gráficos (%)* podemos reducir la imagen simplemente escribiendo un porcentaje apropiado. Por ejem-

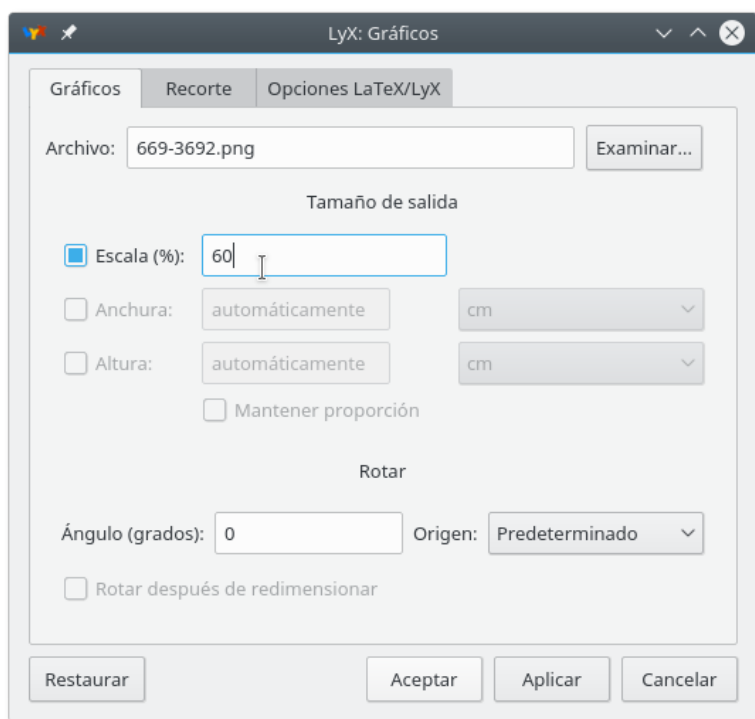



FIGURA 14.1: *Modificar el ancho mientras se inserta la figura*


plo, 60% significa que al compilar el documento la imagen se mostrará al 60% de su tamaño real.

La pregunta que surge instantáneamente es cómo sabemos cuál es el porcentaje correcto. Bueno, en principio si conocemos el tamaño exacto de la imagen y cuán grande es el área de texto debería ser sencillo calcularlo, pero... en fin, que lo único que nos queda es compilar el documento una y otra vez con distintos valores del porcentaje y ver cómo va quedando: al cuarto o quinto intento seguramente lo lograremos.

¿Que no te parece práctico, estimado lector? Ningún problema: sigue leyendo.

#### 14.1.2 Ajustar el tamaño automáticamente

Es posible pedirle a L<sup>A</sup>T<sub>E</sub>X que ajuste por sí solo el tamaño de la imagen como función del ancho del área de texto de la página. Para esto, en el mismo cuadro que usamos arriba (¡sin activar la opción anterior!), cambiemos de pestaña para dirigirnos a  Opciones de LaTeX/LyX (figura 14.2)

Si queremos que la imagen ocupe exactamente el 80% del área de texto en el campo  Opciones LaTeX debemos escribir

```
width=0.8\textwidth
```

La instrucción es simple de comprender: la primera mitad dice «tomar 0.8 (el 80%) del ancho» mientras que la segunda mitad dice «del área de texto». Las dos mitades juntas se leen «el 80% del área de texto».

Si queremos que la imagen ocupe el 100% del ancho del área de texto simplemente podemos evitar la primera parte del comando anterior, dejando solamente:

```
\textwidth
```

Es importante notar aquí que esto forzará a *cualquier* imagen a ocupar *exactamente* el porcentaje del área de texto por nosotros elegido: las imágenes más grandes serán automáticamente reducidas, lo cual está muy bien, *pero las más pequeñas serán aumentadas*, lo cual está muy mal, por lo que se recomienda prestar mucha atención y utilizar esta opción solo cuando sabemos

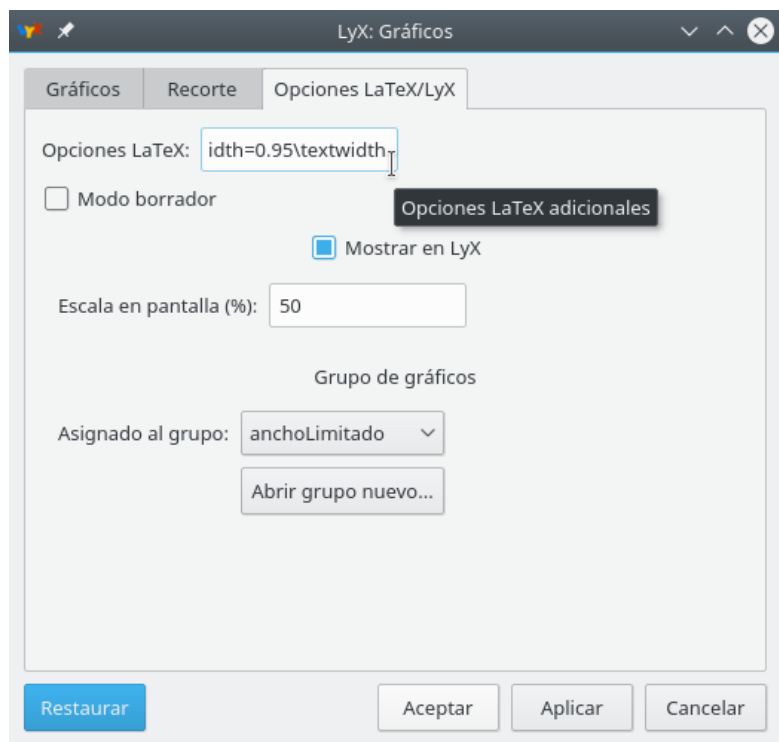


FIGURA 14.2: *Dejar que  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  haga el trabajo sucio*

que la imagen que estamos insertando es demasiado grande para la página de destino.

Si nos interesa utilizar el ancho de la página en lugar del ancho del área de texto podemos reemplazar la instrucción anterior por

```
\paperwidth
```

Pero atención con los márgenes...

### 14.1.3 Tamaño de la imagen en la ventana de edición

Tal y como puede verse en la figura 14.2 también podemos elegir cómo se muestra (o si no se debe mostrar) la imagen en la ventana de LyX durante la edición del documento: mientras trabajamos no es necesario ver la imagen en su máxima resolución por lo que podría resultar cómodo reducirla bastante o incluso decir de no mostrarla.

### 14.1.4 Por no repetir: grupos de imágenes

Ahora bien, ¿tenemos que hacer lo contado en los dos apartados anteriores *cada vez*?

¡No!

Como puede verse *también* en la figura 14.2, tenemos la opción «Grupo de gráficos». Esta opción nos permite dar un nombre a una configuración particular para luego llamarla en la próxima figura: con «Abrir grupo nuevo» asignamos un nombre a la particular configuración que queremos reutilizar mientras que, al insertar la próxima figura, podremos utilizar el menú «Asignado al grupo» para llamar esa configuración.

Más fácil imposible.

## 14.2 Figuras «envueltas»

En esta sección hablaremos de figuras rodeadas de texto, como lo que se muestra aquí mismo.



FIGURA 14.3: *Un bonito pingüino que adopté en internet (al anónimo autor de este y otros pingüinos, gracias)*

Me apresuro en advertir que esto es uno de los pocos puntos que  $\text{\LaTeX}$  no logra manejar del todo bien: el paquete del cual hablaremos aquí, `wrapfig`, puede resultar un tanto «inestable» y dar resultados impredecibles. Por ejemplo, con las opciones por defecto si al compilar el documento la figura cae cerca del margen inferior podría «perforar» el pie de página e incluso dejar un hueco al comienzo de la página siguiente (hagan unas pruebas: puede ser divertido). Y es que el paquete no verificará si existe lugar

en la página para poner la figura allí.

Ya veremos cómo arreglar eso.

$\text{\LaTeX}$  ofrece soporte completo para este paquete, por lo que para utilizarlo solo tenemos que ir a `Insertar >> Flotante >> Figura envuelta` (también podemos insertar cuadros envueltos desde allí). Es importante notar que si colocamos la figura envuelta entre dos párrafos esta será rodeada por el segundo mientras que si la colocamos entre dos palabras dentro de un párrafo será rodeada por las líneas de texto a partir de aquella en la que se encuentran esas palabras.

Esto implica la necesidad de mucho cuidado y control manual de la posición de las figuras envueltas, por lo que surge la pregunta de si es posible hacer que estas «floten» como las figuras normales. Y además, ¿cómo elegimos de qué lado de la página aparecerá la figura o la fracción del ancho de página que esta ocupará?

Haciendo clic derecho sobre la etiqueta del «envoltorio» accedemos al menú de configuración (figura 14.4). Las opciones de «ubicación» son Exterior (a la derecha en páginas impares y a la izquierda en las pares de los documentos «a dos caras»), Interior (lo contrario), siempre Izquierda y siempre Derecha. La opción Permitir flotación es la que corrige los problemas que comentamos anteriormente.



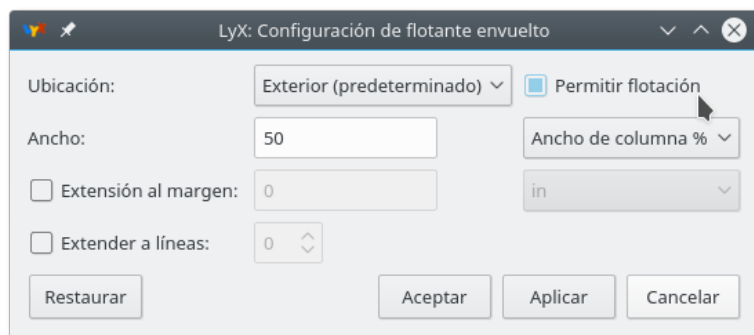


FIGURA 14.4: *La configuración de la «figura envuelta»*

¿Listo? Bueno, no del todo:

- Las figuras envueltas no pueden utilizarse en combinación con listas numeradas o viñetas
- Si la figura envuelta está en el último párrafo y el texto no es suficiente para «envolverla», dependiendo de si se ha activado la flotación o no podemos tener una figura que se desplaza luego del texto y que no respeta el lado elegido (con flotación) o una figura «envuelta a mitad» (sin flotación)
- Si se tienen dos figuras envueltas cercanas, el resultado puede ser malo (hagan la prueba...)
- Si bien la numeración se «integra» con la numeración de las figuras normales podría darse que no aparezcan «en orden», dándonos, por ejemplo, la figura normal de número 5 en el principio de la página y la envuelta de número 4 más abajo
- ...

En fin, que es un paquete cuyos resultados pueden traer algunos dolores de cabeza, requiriendo mucho control manual del documento antes de poder considerarlo «listo».

¡Están advertidos!

### 14.3 Centrar el contenido de figuras y cuadros

Si tenemos una imagen que no toma todo el ancho de la página, ya sea porque es pequeña o porque hemos cambiado su escala para que sea un porcentaje del ancho del área de texto, se presentará al compilar el documento a la izquierda de la página.

Esto es un problema especialmente si, como veremos en la próxima sección, hemos decidido centrar la leyenda.

¿Cómo hacemos entonces para centrar *automáticamente* (que a mano siempre se puede) también la imagen o cuadro en el interior de nuestros flotantes?

La forma más simple es redefinir la macro que da forma a los flotantes, escribiendo en el preámbulo  $\LaTeX$  lo siguiente:

```
\makeatletter  
\g@addto@macro\@floatboxreset\centering  
\makeatother
```

Encontré esta solución aquí:

<https://tex.stackexchange.com/questions/23078/how-can-i-automatically-center-an-image>

### 14.4 Configurar las leyendas de figuras y cuadros

La forma más sencilla de configurar la leyenda de figuras y cuadros en un documento  $\LaTeX$  es utilizar el paquete `caption`.

En Linux y con  $\TeX$ Live (los paquetes `texlive-caption` y `texlive-caption-doc`) la documentación se instala en

```
/usr/share/texmf/doc/latex/caption/caption-eng.pdf
```

La documentación es muy sencilla por lo que en este breve apartado solo daré un ejemplo «cotidiano» para ver cómo se utilizan los diferentes parámetros de este paquete.

La idea de este ejemplo es configurar las leyendas de figuras y cuadros para que sean:

- Justificadas pero con la última línea centrada

- La etiqueta se muestre en *VERSALITAS*
- El texto de la leyenda en *cursiva*

Para lograr esto nos dirigimos al preámbulo  $\LaTeX$  y escribimos

```
\usepackage{caption}  
\captionsetup{justification=centerlast,%  
label font=sc,text font=it}
```

La primera línea activa el paquete `caption` mientras que la segunda lo configura con las tres opciones comentadas más arriba.

Otras opciones ofrecidas por `caption`: cambiar márgenes, el tamaño o el tipo de la fuente, interlineado, color del texto, modificar o incluso eliminar el nombre de la etiqueta (para que se muestre solo el número), insertar un quiebre entre la etiqueta y el texto de la leyenda, la distancia entre la imagen/cuadro y la leyenda (la opción `skip`), etcétera.

Es interesante notar que la instrucción `\captionsetup` puede utilizarse no solo en el preámbulo: colocándola en un flotante particular nos permitirá configurar *esa* leyenda.

Existe un último «truco» para configurar las leyendas de figuras y cuadros: hacer que estas aparezcan *a los lados* en lugar de arriba o abajo. Pero esto requerirá el trabajar con módulos y formato local en gran estilo, por lo que lo dejaremos para la sección 15.2.

## 14.5 Renombrar flotantes

$\LaTeX$  ofrece por defecto dos clases de flotantes: `Figure` y `Table`.  $\LaTeX$  agrega un tercero: `Algorithm`.

Ahora bien, por defecto la traducción de `Figure` es *Figura*, la de `Table` es *Cuadro* y la de `Algorithm` es *Algoritmo*. ¿Pero qué tal si queremos cambiar esos nombres? Podríamos estar interesados en utilizar «Diagrama» en lugar de «Figura» o «Guión» en lugar de «Algoritmo».

Pero atención, que cambiar el nombre a la «etiqueta» no alcanza: los flotantes pueden tener índices propios que deben ser cambiados también para evitar la confusión de nuestros lectores.

Insertar el siguiente código en el preámbulo  $\LaTeX$

```

\usepackage[spanish]{babel}
\addto\captionsspanish{
\def\figurename{Diagrama}
\def\listfigurename{Índice de
diagramas}
}

```

cambiará Figura por Diagrama. Si queremos cambiar el nombre de los cuadros usamos `\tablename{nuevo-nombre}` para la etiqueta y `\listtablename{Índice de nuevo-nombre(s)}` para el índice, etcétera.

En esta página

<https://bioinformatiquillo.wordpress.com/2010/02/22/lyx-latex-cambiar-nombres-fijados/>

tenemos una lista de nombres que podemos cambiar.

## 14.6 Cambiar el tipo de numeración de figuras y cuadros

Esta es fácil, de hecho ya vimos algo parecido en el capítulo 11. Para numerar las figuras con números romanos (¡en mayúscula!), en el preámbulo  $\LaTeX$  escribimos

```
\renewcommand{\thefigure}{\Roman{figure}}
```


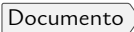
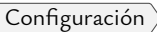
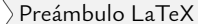
Para los cuadros simplemente cambiamos ambas instancias de `figure` por `table`. Y para cambiar números romanos por letras, bueno, ya saben: capítulo 5.

## Capítulo 15

### Extras

En este capítulo hablaremos de algunos detalles quizás demasiado particulares.

#### 15.1 Describir teclas y menús: el paquete **menukeys**

A lo largo de este libro he mostrado muchas veces combinaciones de teclas y menús, algo como decir  o bien   .

No existe un modo «oficial» de obtener estos efectos, pero uno «extra» que resulta realmente simple es el paquete **menukeys**. Aquí describiré brevemente cómo se utiliza.

El manual de **menukeys** generalmente se instala en  
`/usr/share/texmf/doc/latex/menukeys/menukeys.pdf`

Es muy completo, pero supone un mínimo conocimiento por parte del lector por lo que si bien todas las opciones están claramente escritas, no siempre se dice tan claramente cómo utilizarlas... Pero vayamos por partes.

##### 15.1.1 Qué es **menukeys**

Este paquete ofrece tres «macros»:

- Una macro para mostrar atajos de teclado en forma de «teclas»
- Una macro para mostrar menús en forma destacada

- Una macro para mostrar direcciones de archivos en forma destacada

La forma más simple de lograr que este paquete funcione es activar en  $\text{\LaTeX}$  el uso de fuentes no  $\text{\TeX}$ . En realidad podríamos prescindir de esto e ir en modo « $\text{\LaTeX}$  puro y duro» cambiando la codificación del documento... pero no entraré en esos berenjenales.

Luego de comprobar que el paquete está instalado en nuestro sistema y de pedirle a  $\text{\LaTeX}$  que note que el mismo está disponible (¡siempre hay que reconfigurar!), vamos al preámbulo  $\text{\LaTeX}$  y escribimos

```
\usepackage{menukeys}
```

lo cual activará `menukeys`... con sus valores por defecto, los cuales incluyen utilizar las teclas del teclado Mac en lugar de las de un teclado tipo PC. Es decir, teclas como «Enter», «Tab», «bloqueo mayúsculas» y otras se verán bien extrañas. Para utilizar las teclas tipo PC tenemos que pasar un parámetro al paquete para decirle que el sistema operativo que nos interesa es (¡ouch!) win, modificando la instrucción en el preámbulo que mostramos más arriba por:

```
\usepackage[os=win]{menukeys}
```

Esta es la instrucción que he tenido que utilizar para este documento, pero `menukeys` ofrece varias formas de «teclas» y presentaciones de menú entre las cuales podemos elegir. Por ejemplo, si queremos utilizar las teclas «sombreadas» y que los menús se vean más «redondeados» tenemos que agregar dos líneas en el preámbulo  $\text{\LaTeX}$  para dejarlo así:

```
\usepackage[os=win]{menukeys}
\renewmenumacro{\keys}{shadowedroundedkeys}
\renewmenumacro{\menu}{roundedmenus}
```

Los estilos disponibles se encuentran a partir de la página 6 del manual del paquete.

Es importante notar que no se producirá la separación silábica de palabras «marcadas» por este paquete. Incluso si tenemos una frase como uno de los niveles de un menú esta quedará completa, sin quebrarse, por lo que

podría darse que un menú particularmente largo «perfore» el margen derecho. Al utilizar este paquete es necesario siempre comprobar la salida de la compilación para no encontrarnos con desagradables sorpresas.

### 15.1.2 Usar **menukeys** en LyX

Existen dos formas de utilizar este paquete en LyX: recurriendo a una «caja T<sub>E</sub>X» o con «formato local» (sección 8.3).

Antes de pasar a la segunda (y más cómoda) de las dos opciones convendrá sin embargo conocer la primera, ya que así sabremos qué código nos hace falta escribir.

Pues bien, dentro de la caja T<sub>E</sub>X escribimos, para un directorio:

```
\directory{/usr/share/texmf/doc/latex/
menukeys/menukeys.pdf}
```

mientras que para un menú tenemos dos opciones, utilizar el separador por defecto:


```
\menu{Documento > Configuración > Preámbulo La-
TeX}
```

o indicar explícitamente *cada vez* cuál separador queremos

```
\menu[,]{Documento , Configuración , Preámbu-
lo LaTeX}
```

Para una tecla:

```
\keys{\tab}
```

nos dará . En la página 13 del manual de menukeys encontramos la lista completa de teclas disponibles. Es importante notar que si se escribe una secuencia de letras sin la barra inicial, por ejemplo

```
\keys{Meta}
```

esta secuencia será tomada como una única tecla `[Meta]`. Para más teclas, más instrucciones `\keys` o, si queremos un separador, una sola con el signo «+». Por ejemplo,

```
\keys{\ctrl}\keys{N}
```

nos dará `[Ctrl]` `[N]`, mientras que

```
\keys{\ctrl+N}
```

nos dará `[Ctrl]`+`[N]`.

Pues bien, ahora que ya sabemos cómo funciona la cosa simplemente nos dirigimos a `[Documento]` `[Configuración]` `[Formato local]` y escribimos, para obtener un «atajo» para los menús, lo siguiente:

```
Format 60
InsetLayout Flex:MenuKeys
LyxType          charstyle
LabelString      menukey
LatexType        command
LatexName        menu
Preamble
  \usepackage[os=win]{menukeys}
EndPreamble
ResetsFont true
End
```

(Recuerden, en `LyX` 2.3 es `Format 64`). Ahora en nuestro documento solo tenemos que escribir la ruta del menú separando cada paso con un «>», seleccionar todo y con un `[clic derecho]` `[Estilo de texto]` seleccionar «Menu-Key». De esta forma aquello que en la ventana de edición saldrá así

```
texto Documento > Configuración > Preámbulo LaTeX más texto
                        _____
                        menukey
```

en la salida se verá

```
texto [Documento] [Configuración] [Preámbulo LaTeX] más texto
```



Dejo como ejercicio para el lector imaginar cómo crear un formato local para los atajos de teclado<sup>1</sup>.

Y hasta aquí llegaremos. El paquete ofrece muchísimas opciones más incluyendo la posibilidad de definir estilos propios, pero no entraremos en estos detalles: con lo dicho el resto de la documentación de `menukeys` debería resultar más accesible... por no decir que las opciones por defecto son de mi agrado y no me interesa cambiarlas.

## 15.2 Leyendas laterales para figuras/cuadros

Ya comentamos en la sección 8.1 cómo agregar módulos a LyX, ahora veremos cómo utilizar uno en particular que resulta bastante interesante: `sidecap`.

Este módulo (en realidad, dos módulos) agrega más flotantes que lo que hacen es colocar las leyendas a un costado en lugar de arriba o abajo.

El módulo `sidecap` utiliza el paquete  $\LaTeX$  homónimo para realizar esta tarea, pero nos simplifica el trabajo integrándolo a la interfaz gráfica de LyX.

Desde esta página:

<http://theorie.physik.uni-konstanz.de/jrioux/lyx/>

podemos descargar los siguientes archivos (¡debemos elegir la versión para LyX 2!):

- `sidecap.module`: es la versión principal. Genera flotantes cuya numeración se integra perfectamente a la de las figuras/cuadros normales y cuyo formato respeta lo que hemos configurado para estas últimas.
- `sidecap-starred.module`: como el anterior coloca la leyenda al costado de la imagen, pero elimina la numeración.

Instalado el módulo y reconfigurado LyX nos dirigimos ahora al menú Documento » Configuración » Módulos para activar el módulo `Sidecap`. Luego solo tendremos que ir a Insertar » Flotante » ... para insertar las versiones marcadas como «side caption» que nos sirvan.

---

1. También puede hacer trampa y mirar los apéndices, que cada uno haga cuentas con su conciencia.

Ahora bien, este módulo utiliza las opciones por defecto del paquete `sidecap`, las cuales consisten en colocar la leyenda en la parte «externa» de la página (a la derecha en páginas impares y a la izquierda en las pares para documentos «a dos lados») con la leyenda alineada a la parte inferior de la figura. Cambiar esto último es muy sencillo. En el preámbulo  $\LaTeX$  escribimos

```
\sidecaptionvpos{figure}{c}
```

para hacer que la leyenda esté verticalmente centrada. Si cambiamos la `c` por una `t` la leyenda estará alineada a la parte superior de la imagen. Cambiando `figure` por `table` podremos configurar la alineación vertical de las leyendas de los cuadros.

Desafortunadamente cambiar el lado en el que se coloca la leyenda no es tan simple. Si la opción por defecto es apropiada este módulo es muy útil, pero si queremos otra cosa (por ejemplo, colocar la leyenda en la parte interna de la página, o siempre a la derecha/izquierda) lo que podemos hacer es abrir el archivo del módulo en un editor de texto, copiar su contenido desde «Format» hasta el final y pegarlo como un nuevo «formato local» (sección 8.3). Modificando la línea

```
\usepackage{sidecap}
```

para que quede como

```
\usepackage[innercaption]{sidecap}
```

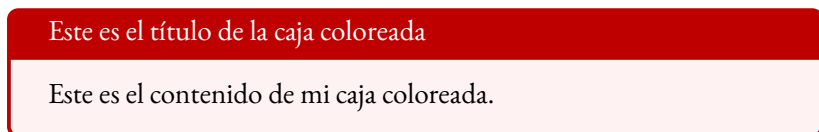
(o bien `leftcaption` o `rightcaption`, lo que queramos). Ahora solo nos queda hacer clic en el botón «convertir al nuevo formato» (el módulo es un tanto viejo), luego en «validar» y finalmente en «aceptar».

¡Listo! No solo hemos visto cómo utilizar un módulo externo, también vimos cómo convertirlo en un «formato local» y cambiarlo según nuestras necesidades. Incluso no debería resultar difícil para el lector el utilizar todo esto para comprender el *cómo crear un módulo nuevo*.

¡Nada mal para una sección *extra*!

### 15.3 Cajas coloreadas

Supongamos que queremos obtener el siguiente efecto en nuestro documento:



Es decir, una «caja» con bordes coloreados, posiblemente con un título y otras clases de pirotecias. Para esto y mucho más existe el paquete `LaTeX tcolorbox` y a partir de la versión 2.1.3 `LyX` ofrece soporte gráfico parcial para el mismo.

Nos dirigimos a `Documento` `» Configuración` `» Módulos` y activamos el módulo Marcos de color elaborados y ya estamos listos para utilizar el módulo: simplemente nos dirigimos a `Insertar` `» Recuadro personalizado` (sin el módulo del cual estamos hablando aquí, esa entrada del menú estará vacía) para elegir una de las opciones disponibles.

Marco color da una caja que toma todo el ancho del área de texto, Marco color dinámico da una caja que se adapta al contenido, etcétera.

Elegida por ejemplo la opción Marco color nos encontraremos en la ventana de edición de `LyX` con una de esas cajas como las de las notas al pie de página: una etiqueta a la izquierda diciendo qué tipo de «caja» es y un espacio a la derecha donde escribir.

Colocando algo de contenido en la caja y compilando el documento nos encontraremos en la salida el texto en una caja gris y sin título, digamos, algo aburrido. Y es que necesitamos pasar al paquete los parámetros necesarios para obtener lo que queremos.

Con el cursor dentro de la caja, ya sea mediante el uso del menú `Insertar` `» Opciones marco color` o a través del  `clic derecho`  `» Opciones marco color` se abrirá una caja dentro de la caja donde pasar los parámetros necesarios. Por ejemplo, para lograr la caja que inicia este apartado he escrito

```
colback=red!5!white,colframe=red!75!black,title=Este
es el título de la caja coloreada
```

¿Que qué significan todos esos códigos? Bueno, la documentación del paquete `tcloborbox`, la cual suele instalarse en Linux en

`/usr/share/texmf/doc/latex/tcolorbox/tcolorbox.pdf`

ocupa más de 150 páginas, por lo que queda completamente fuera del alcance de esta breve sección el comentar todas las opciones disponibles: macros para crear índices, teoremas, cuadros explicativos... para ejemplos mostrando el poder de este paquete es posible consultar el siguiente documento:

`/usr/share/texmf/doc/latex/tcolorbox/tcolorbox-example.pdf`

También es posible consultar el manual de este módulo que acompaña  $\text{L}\text{X}\text{T}\text{E}\text{X}$  y que se encuentra en [Ayuda](#) » [Manuales específicos](#) » [Marcos de color](#).

Estimado lector, ahora que sabes que la herramienta existe tienes material para entretenerte.

## 15.4 Cosas de las que no he hablado

$\text{L}\text{X}\text{T}\text{E}\text{X}$  nos ofrece una lista inmensa de posibilidades para aprovechar la lista incommensurable de habilidades de  $\text{L}\text{A}\text{T}\text{E}\text{X}$  y compañía. En este apartado daré una lista superficial de alguno de los temas que, a pesar de ser interesantes, he decidido no tocar para evitar que el libro crezca desmesuradamente.

### 15.4.1 Soporte $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$

Como ya comentamos en la sección 4.3, desde la versión 2.3  $\text{L}\text{X}\text{T}\text{E}\text{X}$  ofrece soporte para  $\text{BIB}\text{L}\text{A}\text{T}\text{E}\text{X}$ .

Este tema es de interés principalmente para académicos y requiere cierto trabajo para comprender cómo funciona. Ya saben, internet es vuestro amigo.

### 15.4.2 Listado de código con **listings**

$\text{L}\text{X}\text{T}\text{E}\text{X}$  ofrece soporte desde su interfaz gráfica para el paquete `listings` que nos permite dar formato a código en nuestros documentos.

Con [Insertar](#) » [Listado de código](#) obtenemos una «caja» que nos permite escribir el código que queramos y con un clic derecho sobre esa caja podremos seleccionar rápidamente el lenguaje de programación, su «dialecto», la

fuelle, la longitud de las tabulaciones, etcétera. En ese menú también podemos pasar parámetros al paquete para dar color a las instrucciones y todo lo que nos sirva.

En el capítulo 8 del manual que se abre desde **Ayuda** » **Objetos insertados** hay más información al respecto.

### 15.4.3 Formularios PDF

Pues resulta que también podemos utilizar **LyX** para crear documentos «para la oficina». Bajo **Ayuda** » **Manuales específicos** » **Formularios PDF** podemos encontrar una descripción detallada de esta herramienta.

### 15.4.4 Sweave y knitr

En **Ayuda** » **Manuales específicos** » **Sweave / knitr** encontramos dos breves manuales que nos explican cómo utilizar estos interesantes paquetes **LaTeX** que nos permiten trabajar con el programa estadístico R desde **LyX**.

### 15.4.5 Sistemas de álgebra computacional

Si hacemos un clic derecho dentro del editor de ecuaciones de **LyX** tenemos la opción «usar sistema de álgebra computacional (CAS)» y allí varias opciones como utilizar Octave, Maxima, etcétera para evaluar rápidamente la expresión insertada. Por ejemplo, seleccionando Maxima en la expresión  $\int_a^b x^2 dx$  obtendremos  $\int_a^b x^2 dx = \frac{b^3}{3} - \frac{a^3}{3}$ .

### 15.4.6 Y mucho, mucho más

Mejor me detengo aquí, que agregar una lista de las cosas de las que no he hablado *también* implicaría extenderme demasiado.

Crear diagramas de Feynman, escribir partituras musicales con LilyPond, lingüística, presentaciones Beamer, el paquete Xy-pic<sup>2</sup> para realizar todo tipo de gráficos *desde LaTeX*, incluso escritura Braille encuentran soporte directo en la interfaz gráfica de **LyX**.

---

2. <https://ctan.org/pkg/xy-pic>

Y si crees, lector, que lo que hemos presentado en esta tercera parte del libro ha sido un mazazo de código  $\text{\LaTeX}$ , te diré que no hemos enfrentado más que una pequeña fracción de todo lo que es posible hacer: cosas como generar párrafos con formas arbitrarias o incluso utilizar  $\text{\LaTeX}$  para realizar cálculos matemáticos está al alcance del «programador» entusiasmado. ¡No hay que olvidar que  $\text{\TeX}$  es un lenguaje de programación *Turing completo*!

Nadie ha dicho que todas esas cosas sean fáciles y en la gran mayoría de los casos ni siquiera valen la pena el trabajo extra. Por ejemplo, controlar las formas de los párrafos implica muchísimo trabajo manual, pero ciertamente es posible.

Mi intención con este libro era abrir un camino. Está en ti, lector, decidir hasta dónde quieres llegar.

Como dicen por allí, *happy LyXing!*

Parte IV

Apéndices

*Donde se presentan detalles varios sobre el formato de este libro*

*Las personas grandes son bien extrañas*



## Apéndice A

# Sobre fuentes tipográficas

Elegir la fuente tipográfica correcta para un texto es uno de esos temas en los que se tienen más respuestas que gente respondiendo.

Este no es un libro sobre diseño, sino una explicación sobre cómo utilizar LyX para diseñar. No tengo formación en diseño gráfico por lo que no esperen que de un consejo. Pero puedo, eso sí, comentar qué utilizo.

## A.1 Las fuentes utilizadas en este libro

Como ya comenté antes la fuente del cuerpo de texto es EB Garamond:  
<http://www.georgduffner.at/ebgaramond/index.html>

Magnífica recreación de los diseños de Claude Garamond de mediados del siglo XVI esta fuente tiene la enorme ventaja (al menos a mis ojos) de su belleza y el pequeño inconveniente de que no tiene disponible aún la versión en negrita (está en desarrollo). En lugar de negrita he aplicado **VERSALITAS**, creo que con buenos resultados. También he utilizado la «variación 11» con la opción OpenType +cv11. Esta opción cambia la forma del 1 cuando se utilizan numerales «estilo antiguo», haciéndolo pasar de algo que parece una i en versalitas a un verdadero «1» (si bien pequeño).

Para los títulos también he utilizado otra «garamond», Cormorant:  
<https://github.com/CatharsisFonts/Cormorant>

Esta fuente es demasiado detallada y con contraste demasiado alto como para poder ser utilizada en el cuerpo del texto, pero para tamaños grandes

es simplemente perfecta.

Combinar tipografías en forma armoniosa es un difícil arte. Algunos proyectos nos facilitan las cosas con «familias» (Libertinus, Source pro, PT, Alegreya, etcétera), ¿pero qué fuente «sans» tenemos que se lleve bien con una «garamond»? De la gente que produce Cormorant nos llega una interesante propuesta: Eau de Garamond

<https://github.com/CatharsisFonts/EauDeGaramond/>

Esta fuente está en plena fase de desarrollo, pero no he tenido problemas utilizándola en este texto.

No he encontrado una fuente monoespaciada pensada especialmente para acompañar una garamond, por lo que al final me he decantado por utilizar un clásico, DejaVu Sans Mono:

<https://dejavu-fonts.github.io/>

Para las fórmulas matemáticas he seleccionado STIX Two Math, fuente de la que ya hablé en el capítulo 6.

En los ejemplos durante el desarrollo del libro he nombrado otras fuentes tipográficas, muchas veces indicando dónde encontrarlas.

## A.2 Más sobre fuentes tipográficas

Para más información sobre fuentes tipográficas con licencia libre, incluyendo análisis (más o menos) detallados de sus características avanzadas, puedes darte una vuelta por mi blog:

<https://elpinguinotolkiano.wordpress.com/entradas-destacadas/#fuentes>

En particular quiero destacar dos artículos cuyos objetivos resultan demasiado generales como para incluirlos en este libro:

Conceptos tipográficos:

<https://elpinguinotolkiano.wordpress.com/2016/04/28/conceptos-tipograficos/>

Conocer las características OpenType ofrecidas por una fuente

<https://elpinguinotolkiano.wordpress.com/2017/04/20/conocer-las-caracteristicas-opentype-ofrecidas-por-una-fuente>

## Apéndice B

### Formato local

Para desarrollar este texto he creado varios estilos de carácter que simplifiquen el proceso de edición. A continuación el contenido de Documento Configuración Formato local.

Pero antes un par de consideraciones.

Primero que nada es importante notar que algunos de los formatos aquí definidos dependen de instrucciones ya existentes en el preámbulo, del cual hablaremos en el apéndice C.

Y por supuesto, que en un documento «normal» no es necesario definir *tantos* estilos: las características del presente documento (ser él mismo un ejemplo de los temas que en él se tratan) hacen que sea necesaria una lista semejante. De hecho, el estilo «textfrac» creo haberlo utilizado solo en dos ocasiones... ¿o fue una? ¡Maldición!

¡Siempre hay que tener los estilos bajo control! El formato nos ayuda a expresar mejor nuestras ideas, pero solo hasta cierto límite: si la presentación de nuestro documento es demasiado variada terminará por distraer al lector de aquello que queremos transmitir.

Si bien es cierto que ir demasiado lejos en aras del minimalismo puede resultar en una presentación poco clara, tampoco debemos olvidar que «Complejidad» no es sinónimo de «Elegancia».

Ahora sí, a partir de la próxima página el «formato local» que he utilizado para llegar hasta aquí. Recuerda: `Format 60` es para `LATEX 2.2` mientras que en `LATEX 2.3` se debe utilizar `Format 64`.

Format 60

```
InsetLayout Flex:HistoLig
LyxType          charstyle
LabelString      histolig
LatexType        command
LatexName        histolig
Preamble
  \newcommand*{\histolig}[1]{\{
    \addfontfeature{%
      Ligatures=Historic} #1\}
EndPreamble
ResetsFont       true
```

End

```
InsetLayout Flex:TextFrac
LyxType          charstyle
LabelString      textfrac
LatexType        command
LatexName        textfrac
Preamble
  \newcommand*{\textfrac}[1]{\{
    \addfontfeature{%
      RawFeature={+frac}} #1\}
EndPreamble
ResetsFont       true
```

End

```
InsetLayout Flex:Kleymisky
LyxType          charstyle
LabelString      kley
LatexType        command
LatexName        kley
Preamble
  \newfontfamily{\kleym}[]{\Kleymisky}
  \newcommand*{\kley}[1]{\{\kleym #1\}
EndPreamble
ResetsFont       true
```

```
End
  InsetLayout Flex:MenuKeys
  LyxType          charstyle
  LabelString      menukey
  LatexType        command
  LatexName        menu
  ResetsFont      true
End
  InsetLayout Flex:KeysKeys
  LyxType          charstyle
  LabelString      okey
  LatexType        command
  LatexName        keys
  ResetsFont      true
End
  InsetLayout Flex:sinserif
  LyxType          charstyle
  LabelString      sinserif
  LatexType        command
  LatexName        sinserif
  Preamble
    \newcommand*{\sinserif}[1]{\sffamily #1}
  EndPreamble
  Font
    Family          sans
  EndFont
  ResetsFont      true
End
  InsetLayout Flex:ttmono
  LyxType          charstyle
  LabelString      ttmono
  LatexType        command
  LatexName        ttmono
  Preamble
    \newcommand*{\ttmono}[1]{\ttfamily #1}
```

```
EndPreamble
Font
  Family          Typewriter
EndFont
ResetsFont        true
End
  InsetLayout Flex:LogoTipo
  LyxType          charstyle
  LabelString      HoLogo
  LatexType        command
  LatexName        loguito
  Preamble
    \newcommand*{\loguito}[1]{\{\hologo{#1}\}}
  EndPreamble
  ResetsFont      true
End
  InsetLayout Flex:PartNote
  LyxType          charstyle
  LabelString      PartNote
  LatexType        command
  LatexName        notitas
  Preamble
    \newcommand*{\notitas}[1]{\{
      \partnote\itshape #1\}}
  EndPreamble
  ResetsFont      true
End
```

## Apéndice C

### Preámbulo L<sup>A</sup>T<sub>E</sub>X

A continuación lo que he escrito en el preámbulo del documento. En las primeras líneas defino la fuente principal del texto para así aplicarle una propiedad OpenType específica, como comenté en el apéndice [A](#). Esto quiere decir que he dejado en «predeterminado» la opción correspondiente en la configuración del documento, ¡a no olvidarse de eso! También hay que olvidar que he seleccionado «Tipografías no-TeX predeterminadas» allí para así poder configurar la fuente matemática y que, dado que he terminado la edición del libro en una versión de desarrollo de L<sup>A</sup>X 2.3, que en lugar de escribir `\usepackage{microtype}` en el preámbulo he activado el uso de la microtipografía en el menú de las tipografías del documento.

```
\setmainfont{EB Garamond}[  
  Mapping=tex-text,  
  RawFeature={+cv11}]      % cambia la forma del l
```

```
\setmathfont{STIX Two Math}
```

```
\usepackage{hologo} % logo de XeTeX & compañía  
\usepackage[os=win]{menukeys} % teclas y menús  
\newfontfamily{\voll}[]{\Vollkorn}  
                                % para algún ejemplo
```

```
% para el capítulo sobre letras capitulares --
```

```
\newfontfamily{\Iniciales}[RawFeature={+salt}]{Pali}
\renewcommand{\LettrineFontHook}{\Iniciales}
\setcounter{DefaultLines}{4}
\usepackage{magaz}

% Las leyendas -----
\usepackage{caption}
\captionsetup{justification=centerlast,%
    labelfont=sc,textfont=it}
% lo siguiente centra la imagen en el flotante
\makeatletter
\g@addto@macro\@floatboxreset\centering
\makeatother

% la sección de estilos de página -----
\renewcommand{\headrulewidth}{0.4pt}
\usepackage{emptypage} % para que las páginas en
                        % blanco estén en blanco

% quitar el número de capítulo
% y sección de la cabecera -----
\renewcommand{\chaptermark}[1]{
\markboth{#1}{}
\renewcommand{\sectionmark}[1]{
\markright{#1}}
```

```
% el estilo de página normal, con sus cabeceras
\pagestyle{fancy}
\fancypagestyle{normal}{%
    \fancyhead{}
    \fancyhead[R0,LE]{\thepage}
    \fancyhead[L0]{\nouppercase{\leftmark}}
    \fancyhead[RE]{\nouppercase{\rightmark}}
    \fancyfoot{}
}
```



```

% el estilo de página especial
\fancypagestyle{especial}{%
  \fancyhead{}
  \fancyhead[R0,LE]{\thepage}
  \fancyhead[L0]{\nouppercase{Introducción}}
  \fancyhead[RE]{\nouppercase{Introducción}}
  \fancyfoot{}
}

% el estilo del índice
\fancypagestyle{index}{%
  \fancyhead{}
  \fancyhead[R0,LE]{\thepage}
  \fancyhead[L0]{\nouppercase{Índice}}
  \fancyhead[RE]{\nouppercase{Índice}}
  \fancyfoot{}
}

% la sección de las notas al pie de página ---
\makeatletter
\renewcommand\@makefnmark[1]{%
  \noindent\makebox[0pt][r]{%
    \@thefnmark.\space#1}
\makeatother

% la sección de los títulos -----
\usepackage{titlesec}
\newfontfamily{\cormorant}{Cormorant}
\titleformat{\part}[display]
  {\huge\cormorant}{\filright%\thepart\
}{20pt}{\Huge}
\titleformat{\chapter}[frame]
  {\LARGE\cormorant}{\filright
  \chaptertitlename\
  \thechapter}{20pt}{\huge}
\titleformat*{\section}{\Large\bfseries\cormorant}
\titleformat*{\subsection}{\large\bfseries\cormorant}

```

```
% esto hace que las páginas \part queden sin pie
\makeatletter
\renewcommand\part{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \thispagestyle{empty}%
  \if@twocolumn
    \onecolumn
    \@tempwatrue
  \else
    \@tempwafalse
  \fi
  \null\vfil
  \secdef\@part\@spart}
\makeatother
\makeatletter
\def\@endpart{\vfil\newpage
              \if@twoside
              \if@openright
              \null
              \thispagestyle{empty}%
\vspace*{3cm}%
\begin{quote}%
\partnote
\end{quote}%
              \fi
              \fi
              \if@tempwa
                \twocolumn
              \fi}
\newcommand\partnote{}
\makeatother
```

## Apéndice D

# Otros formatos... y un saludo

Una pequeña lista de las *otras* cosas que he configurado para lograr este formato.

## D.1 Otros formatos utilizados en este libro

Para este libro he activado los siguientes módulos:

- Marcos de color elaborados
- Capitales
- Listas personalizadas (enumitem)

El diseño de página usa un formato A5 con estilo elaborado y la opción «documento con dos caras».

Los márgenes son

SUPERIOR 2,5 cm

INFERIOR 1,5 cm

INTERIOR 2,5 cm

EXTERIOR 1,5 cm

He activado el soporte `Hyperref` permitiendo que los enlaces se quiebren en líneas, eliminando los marcos alrededor de los mismos y seleccionando de colorear los enlaces con

```
linkcolor=cyan,urlcolor=blue
```

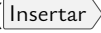
En la pestaña Marcadores he seleccionado los «marcadores numerados» así el índice del PDF también muestra los números de las entradas.

Formato de salida PDF (XeTeX)...

Lo demás ya lo he dicho todo.

## D.2 Y sí, también formato directo

¡Pero solo en una página!

Me refiero, claro está, a la página de la cubierta: definir estilos que solo se utilicen allí carece de sentido por lo que unos pocos clics derechos para cambiar el formato de los párrafos (llevarlos a alineación derecha) o del texto (el tamaño) he utilizado, junto a un par de saltos de página (). Eso sí, el breve texto en Kleymisky utiliza un estilo... si bien a un tamaño modificado manualmente.

Lo que llevó un poco más de trabajo fue colocar la imagen.

La silueta del pingüino que observa las puertas de *Moria* de `LATEX` la creé en LibreOffice Draw sobre el perfil de cierta imagen que utilicé como ejemplo en el capítulo 14, agregando oportunas capturas de pantalla que mostraran el programa y el resultado de su uso.

Dado que Draw no permite exportar solo una «selección» al crear un PDF, la «diapositiva» del documento fue modificada para no tener márgenes y ser de la exacta dimensión necesaria para la imagen.

Es importante al exportar a PDF desde Draw el seleccionar «compresión sin pérdidas» para que la imagen no se arruine... pero bueno, me estoy yendo de tema, que después de todo este libro es sobre `LATEX`.

A partir de ese documento Draw creé entonces un archivo PDF que inserté en la página dedicada a la cubierta de este documento.

Y aquí viene la parte interesante. Crear un estilo de página con sus márgenes solo para la cubierta no tiene sentido, pero si utilizo solo el ancho del área de texto para la imagen esta quedaría muy pequeña por lo que recurrir

a la instrucción `\paperwidth` (también comentada en el capítulo 14) «sueña a solución»... salvo por el pequeño problema de hacer que la imagen comience *antes* del margen izquierdo.

¡Todo se resuelve! He aquí la secuencia de formatos directos utilizada:  
Primero una «caja  $\text{\TeX}$ » con el siguiente código

```
\begingroup
\leftskip=-3em
```

La segunda instrucción hace que el párrafo a continuación adquiriera una sangría negativa de 3 em... y también el siguiente y todos los párrafos que existan después: por eso fue necesario crear un «grupo» con la primera instrucción para así aislar los efectos de `\leftskip` del resto del documento.

Debajo de esa caja  $\text{\TeX}$  inserté la imagen PDF que comenté antes, configurando sus «opciones  $\text{\LaTeX}$ » con

```
width=0.81\paperwidth
```

para que tome el 81 % del ancho de la página completa.

Finalmente, a continuación de la imagen otra caja  $\text{\TeX}$  con

```
\endgroup
```

para cerrar el grupo y terminar el proceso.

Para la «contra cubierta» utilicé también algunos «espacios verticales» desde Insertar Formato Espacio vertical, pero esos son fáciles de comprender.

Y sí, tuve que compilar el documento varias veces hasta lograr ajustar todo y que quedara bien. Qué se le va a hacer.

## D.3 Saludo final

Si has llegado hasta aquí, lector, ¡gracias por tu paciencia!

Espero que hayas disfrutado el libro. Yo ciertamente disfruté escribiéndolo. Tratando de explicar es que he aprendido más de lo que creí que llegaría a saber. El momento en el que decidí escribir este libro se une ahora al

momento en el que cumpliste tu propósito de leerlo hasta el final. El viaje ha sido largo, pero creo ha valido la pena.

¡Hasta la próxima *Aventura Tipográfica*!