

# Creación de ficheros $\LaTeX$ con GNU Emacs\*

Joaquín Ataz López (jal@um.es)\*\* .

## Resumen

GNU Emacs es un muy potente editor de textos al que ciertos paquetes de extensión dotan de amplias capacidades para tratar con ficheros de tipo  $\LaTeX$ . Entre ellos destacan AUC  $\TeX$ , Ref $\TeX$  y Bib $\TeX$ .

El presente documento contiene una explicación detallada de las posibilidades que Emacs y sus paquetes de extensión ofrecen en relación con los ficheros de tipo  $\LaTeX$ . La suma de todas estas *habilidades* determina que GNU Emacs, se merezca la denominación de «*entorno integrado*» para la creación y manipulación de ficheros en formato  $\LaTeX$ .

---

\* Copyright (c) 2004 Joaquín Ataz López.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

(Se otorga permiso para copiar, distribuir o modificar este documento en los términos de la Licencia GNU para Documentación Libre, versión 1.2 o cualquier versión posterior publicada por la Free Software Foundation; sin secciones invariantes, sin textos de la cubierta frontal y sin textos de la cubierta posterior. Una copia completa de la licencia (en inglés) se incluye en el apéndice titulado “GNU Free Documentation License”).

\*\*El autor agradece las observaciones recibidas de numerosos lugares. Pero debe destacar especialmente la minuciosa corrección y las atinadas observaciones realizadas por Luis Sanjuán. La *corrección* ha mejorado mucho este documento en sus aspectos formales. Las observaciones han sido causa directa de algunas de las notas a pie, entre ellas las 18, 19 y 20. Asimismo a su amable y desinteresada ayuda se debe la incorporación a este documento de las utilidades ofrecidas por el paquete hyperref.

# Índice

<b>Introducción</b>	<b>4</b>
<b>1. Preparación inicial</b>	<b>9</b>
1.1. Conseguir e instalar Emacs y sus paquetes de extensión	9
1.2. Configurar Emacs para que todo funcione junto	10
<b>2. Utilidades básicas</b>	<b>12</b>
2.1. Ayuda en línea de AUC $\text{\TeX}$	12
2.2. Sangrado y reformateado del texto fuente	13
2.2.1. El modo auto-fill de Emacs	13
2.2.2. Reformateado automático del texto	15
2.2.3. Mandatos para el reformateado	15
2.2.4. Sangrado manual de líneas e inserción de tabuladores	16
2.3. Selección de texto (la región)	17
2.4. Resaltado gráfico de la sintaxis	18
2.5. Auto-completado	19
<b>3. Ayuda general para la escritura del texto fuente</b>	<b>21</b>
3.1. Caracteres con tratamiento especial	21
3.1.1. Marcas de comentario (%)	21
3.1.2. Dobles comillas	23
3.1.3. Símbolo del dólar (\$)	23
3.1.4. Llaves de apertura y cierre de grupos	24
3.2. Facilidades generales para introducir macros de $\text{\LaTeX}$	25
3.2.1. M-TAB: TeX-complete-symbol	25
3.2.2. TeX-insert-macro y TeX-electric-macro	26
<b>4. Macros <math>\text{\LaTeX}</math> para especificar el formato de fuente</b>	<b>28</b>
4.1. Insertar macros para formatos de fuente	28
4.2. Borrar o cambiar una especificación de formato	30
4.3. Resumen sobre la especificación de formatos	31
<b>5. Unidades estructurales de <math>\text{\LaTeX}</math></b>	<b>31</b>
5.1. Insertar unidades estructurales	31
5.2. Control de las tareas asociadas a la inserción de títulos	32
5.3. Asociación automática de etiquetas a los títulos	33
5.4. Utilización del modo menor “outline” de Emacs para ver la estructura y navegar por el documento	34
5.4.1. Activar “outline” de forma compatible con AUC $\text{\TeX}$	35
5.4.2. Mandatos que afectan a la visibilidad del documento	35
5.4.3. Mandatos de movimiento por el texto	37
5.5. Usar $\text{\RefTeX}$ para ver la estructura del documento y navegar por ella	38
<b>6. Entornos</b>	<b>40</b>
6.1. Utilidades generales para los entornos	40
6.1.1. Ocultar y mostrar el contenido de un entorno	41
6.1.2. Movernos entre las líneas de apertura y cierre de un entorno	42
6.1.3. Cerrar un entorno	42
6.2. Inserción y sustitución de entornos	43
6.3. Notas sobre algunos entornos especiales	44
6.3.1. El entorno “document”	44
6.3.2. Entornos de tipo lista	44
6.3.3. Entornos para objetos flotantes	45

6.3.4. Entorno matemático y modo matemático . . . . .	45
<b>7. Etiquetas y referencias cruzadas</b> . . . . .	<b>47</b>
7.1. Etiquetas propiamente dichas . . . . .	47
7.1.1. Estructura de las etiquetas de RefTeX . . . . .	47
7.1.2. Creación de etiquetas . . . . .	48
7.1.3. Hacer referencia a las etiquetas . . . . .	50
7.1.4. Funcionamiento del teclado en la ventana de selección de etiquetas . . . . .	51
7.1.5. Realizar simultáneamente múltiples referencias . . . . .	54
7.2. Citas bibliográficas . . . . .	54
7.2.1. Insertar citas bibliográficas . . . . .	55
7.2.2. Otras características relacionadas con las citas bibliográficas . . . . .	55
7.3. Índices terminológicos . . . . .	56
7.3.1. Insertar entradas de índice . . . . .	56
7.3.2. Generación automática de índices mediante un “ <i>fichero de frases</i> ” . . . . .	57
7.3.3. La ventana de índice de RefTeX . . . . .	61
7.4. Ver las referencias cruzadas . . . . .	63
<b>8. Trabajo con varios documentos</b> . . . . .	<b>64</b>
8.1. Documentos maestros . . . . .	64
8.2. Mandatos para trabajar con varios documentos cuyo documento maestro es común. . . . .	67
<b>9. Trabajo con ficheros BibTeX</b> . . . . .	<b>67</b>
9.1. Inserción de registros . . . . .	68
9.2. Edición de los registros . . . . .	70
9.2.1. Movimiento del cursor . . . . .	70
9.2.2. Añadir, borrar y modificar campos . . . . .	71
9.2.3. Eliminar y recuperar campos y registros . . . . .	71
9.2.4. Limpiar el registro . . . . .	72
9.3. Edición avanzada . . . . .	74
9.3.1. Análisis sintáctico de los registros . . . . .	74
9.3.2. Orden alfabético de los registros . . . . .	74
9.3.3. Generación automática de etiquetas . . . . .	75
<b>10. Uso del compilador L<sup>A</sup>T<sub>E</sub>X y de otras utilidades externas relacionadas</b> . . . . .	<b>76</b>
10.1. Mandatos externos que se pueden ejecutar . . . . .	76
10.2. Cómo ejecutar los mandatos externos . . . . .	79
10.3. Comprobación de errores . . . . .	81
10.4. Control de procesos externos . . . . .	81
<b>Apéndices</b> . . . . .	<b>83</b>
A. Modificar variables de Emacs para la personalización . . . . .	83
B. Resumen de mandatos . . . . .	89
C. GNU Free Documentation License (Licencia GNU para Documentación Libre) . . . . .	92
D. Índice de nombres de órdenes y opciones de usuario . . . . .	100

## Introducción

### **L<sup>A</sup>T<sub>E</sub>X, GNU Emacs, AUC T<sub>E</sub>X y otras extensiones de Emacs**

Crear documentos con L<sup>A</sup>T<sub>E</sub>X implica el uso de un conjunto variado de herramientas diferentes: editor de textos, gestor de bases de datos bibliográficas, analizadores de sintaxis y de ortografía, generador de índices, visores para diferentes tipos de formatos de ficheros, conversores entre dichos formatos y, por supuesto, las distintas versiones del compilador L<sup>A</sup>T<sub>E</sub>X generadoras de salidas en los formatos dvi, pdf y html.

Es para estos casos para lo que se inventaron los *entornos integrados*, que permiten, la integración de numerosas herramientas en una sola, de tal manera que desde esta última podamos usarlas todas.

Entre los entornos integrados existentes en el mundo Unix-Linux que conozco, uno de los más potentes es GNU Emacs complementado con tres paquetes de extensión: AUC T<sub>E</sub>X, RefT<sub>E</sub>X y BibT<sub>E</sub>X:

- **GNU Emacs** posiblemente sea el más completo editor de textos en términos absolutos. Entre los formatos que es capaz de reconocer se encuentra T<sub>E</sub>X (en sus variantes de L<sup>A</sup>T<sub>E</sub>X, Plain-T<sub>E</sub>X, Ams-T<sub>E</sub>X y SliT<sub>E</sub>X). En consecuencia están disponibles las funciones ordinarias de resaltado gráfico de la sintaxis mediante la utilización de colores (en entornos gráficos) y de *indentación inteligente*.
- **AUC T<sub>E</sub>X** es un paquete de ampliación de Emacs que le dota de cerca de 100 funciones específicamente destinadas a los sistemas basados en T<sub>E</sub>X. Estas funciones se pueden agrupar (y así he hecho) en dos categorías: las dirigidas a *facilitar* la escritura del documento fuente, y las dirigidas a *ejecutar* el compilador L<sup>A</sup>T<sub>E</sub>X y sus herramientas relacionadas.
- **RefT<sub>E</sub>X** es otro paquete de ampliación de Emacs que contiene funciones específicamente destinadas a la generación y gestión de todo tipo de etiquetas y referencias cruzadas. Hace menos cosas que AUC T<sub>E</sub>X, pero las hace francamente bien. Además es capaz de coordinarse con el paquete anterior de tal manera que el propio AUC T<sub>E</sub>X delegue en RefT<sub>E</sub>X en materia de creación y gestión de etiquetas.
- **BibT<sub>E</sub>X**, finalmente, dota a Emacs de un modo mayor llamado “bibtex” destinado al manejo de ficheros de bibliografía en el formato del mismo nombre.

En suma: la combinación de todas estas herramientas convierte a Emacs en un sistema en el que hay utilidades dirigidas a facilitar:

1. La lectura y aspecto de los ficheros fuente.
2. Su escritura y gestión.
3. La escritura y gestión de documentos bibliográficos relacionados.
4. La compilación del documento.
5. La depuración de los posibles errores detectados por la compilación.
6. La ejecución de varias herramientas externas adicionales (bibtex, makeindex, dvips, corrector ortográfico, analizador de errores, etc).
7. La visualización de los ficheros de salida de la compilación.

Si a todo ello añadimos un conjunto de facilidades para escribir *proyectos* en los que el documento fuente se encuentre repartido entre varios ficheros, veremos que ciertamente nos encontramos ante un conjunto de herramientas bastante potente.

### **Contenido de este documento y conocimientos que se le presuponen al lector**

Este documento explica las herramientas de Emacs específicamente destinadas al formato  $\text{\LaTeX}$ , pero no explica el funcionamiento de Emacs propiamente dicho, ni el formato  $\text{\LaTeX}$ . Habrá pues que conocerlos de antemano. Aunque no es preciso ser un usuario avanzado de Emacs ni de  $\text{\LaTeX}$ . Basta con que se conozcan ambos sistemas lo suficiente como para generar un documento simple.

En este sentido, para facilitar la lectura a los lectores novatos en alguno de ambos sistemas —o en los dos—, este documento incluye las siguientes características:

- La información excesivamente técnica y toda la relativa a la personalización de Emacs ha sido diferenciada tipográficamente del resto, de modo que el lector que quiera usar estas herramientas *tal y como están* pueda limitarse a leer “*la letra gorda*”.
- En algunas ocasiones —no demasiadas— se usan las notas a pie de página para aclarar conceptos o nociones que un usuario experimentado no necesita que le sean aclaradas.
- Algunas nociones de  $\text{\LaTeX}$ , son *recordadas* al principio de ciertas explicaciones.
- Al final del documento se incluye un apéndice relativo a cómo personalizar en Emacs el valor de las *opciones de usuario*, ya que he pensado que, a fin de cuentas, tampoco es tan difícil abrir Emacs y ponerse a editar, sobre todo en su versión gráfica dotada de menús accesibles con el ratón (o en su variante Xemacs); pero para personalizar Emacs sí se hace preciso tener

ciertos conocimientos sobre su funcionamiento, siendo esta la razón de que el apéndice se limite a esta cuestión.

O sea: el usuario experimentado no tiene que leer las notas a pie de página y puede saltarse algunos párrafos iniciales de ciertas secciones, pero para él está “*la letra chica*”. El novato puede limitarse a “*la letra gorda*” pero, a cambio, es posible que necesite leer las notas a pie de página.

El presente documento, por otra parte, se ha hecho sobre la base de la versión 21.3 de GNU Emacs para sistemas Unix-Linux. En él se contiene la información extraída de:

- La página Info del propio Emacs.
- La página Info de AUC T<sub>E</sub>X.
- La página Info de RefT<sub>E</sub>X.
- La ayuda html de BibT<sub>E</sub>X.
- La información extraída por el procedimiento de comprobar (mediante el mandato “C-h b”) las asociaciones de teclas que Emacs carga y los mandatos asociados. De ahí he extraído información sobre algunas asociaciones de teclado no documentadas.
- La ayuda ofrecida en los *buffers* de configuración de Emacs respecto de la utilidad de las diferentes variables definidas por los paquetes objeto de este documento.
- Mi propia experiencia.

En general he intentado ser minucioso en la explicación de las funciones y de las variables que las controlan; pero, respecto de estas últimas, la verdad es que he actuado con cierta irregularidad. Hay aspectos en los que he entrado en más detalles que en otros. Eso tiene que ver en parte con las tareas que yo acostumbro más a realizar. Pero, en todo caso, en este punto mi consejo para quien quiera profundizar es que acuda a las fuentes de información que acabo de citar<sup>1</sup> —y a cualquier otra que se localice.

Aparte de la *irregularidad* con que me he ocupado de la personalización, hay otros aspectos que no he tratado. En concreto:

- A propósito de RefT<sub>E</sub>X no he entrado en:
  - Cómo crear nuevos entornos numerados o nuevas macros generadoras de etiquetas, referencias o entradas de índice, y que sean reconocibles por RefT<sub>E</sub>X.

---

<sup>1</sup>Emacs, en concreto, al estar basado en Lisp, que es un lenguaje *interpretado*, dispone de una maravillosa utilidad para aprender sus vericuetos: la posibilidad de, en un momento dado, comprobar qué asociaciones de teclas hay cargadas en memoria, y qué mandato ejecuta cada una de ellas. Eso se obtiene simplemente pulsando “C-h b”

- Cómo optimizar su funcionamiento.
  - Cómo hacerlo funcionar con paquetes adicionales.
- No se dice nada de lo que AUC T<sub>E</sub>X llama *personalización y construcción de estilos propios*, cuya comprensión exige conocimientos de Elisp que no estoy seguro de poseer yo mismo pero que, en cualquier caso, en un documento como este no se pueden explicar.
  - En líneas generales no he entrado en la explicación de las opciones que cada paquete incorpora a los menús de Emacs, debido a que, a fin de cuentas, un menú es sólo un *modo alternativo* para la ejecución de mandatos.

Aun con estas omisiones, creo que queda un documento lo suficientemente extenso y detallado como para aprender a usar casi todas las utilidades ofrecidas por Emacs para la creación de ficheros en formato L<sup>A</sup>T<sub>E</sub>X.

En el caso de errores en este documento, o, en general, sugerencias sobre su posible mejora, agradeceré que se me envíe cualquier comentario a “jal@um.es”.

### Convenciones tipográficas y de vocabulario

**Distinción entre texto normal y texto avanzado:** Esta convención tipográfica usada en muchos manuales universitarios es extremadamente cómoda pues permite diferenciar entre los distintos niveles del texto:

*El texto avanzado o de excesivo detalle es el que no tiene por qué interesar a todos los lectores. Se escribe en párrafos como el presente: con sangrado superior al normal, sin indentación en la primera línea, con un tipo de letra ligeramente más pequeña que el resto y con inclinación “oblicua” (letra tipo slanted).*

*En líneas generales la información relativa a la personalización de Emacs o a las opciones de usuario cuya modificación afecta al funcionamiento por defecto, la he considerado avanzada.*

**Pulsaciones de teclas:** Para indicar pulsaciones de teclas, se utiliza la convención normal en la documentación de Emacs: la tecla *Control* se transcribe por “C”, la tecla *Meta* por “M”<sup>2</sup> y un guión uniendo a dichas teclas entre sí o con otra tecla, indica que la pulsación debe ser simultánea<sup>3</sup>.

---

<sup>2</sup>En Emacs se llama tecla *meta* a una especie de *tecla virtual* que en cada tipo de ordenador y sistema operativo puede ser diferente. En teclados ordinarios para PC el efecto correspondiente a dicha tecla puede obtenerse pulsando simultáneamente Alt-Izquierda y otra tecla, o pulsando primero ESC y luego la otra tecla; o sea: por la primera vía la pulsación es *simultánea* y por la segunda *sucesiva*.

<sup>3</sup>A lo dicho en el texto hay que añadir que a las teclas que generan caracteres no imprimibles (o invisibles) se les suele denominar por una abreviatura de su nombre. En particular, SPC se usa para la barra espaciadora, TAB para el tabulador, y RET para referirnos a una tecla que es conocida por

Además las pulsaciones de teclado se transcriben con letra del tipo de máquina de escribir como esta y se encierran entre comillas.

### **Nombres de mandatos (de Emacs) y de variables modificables por el usuario:**

Estos nombres me han ocasionado un verdadero quebradero de cabeza porque, en primer lugar, dado que Emacs distingue entre mayúsculas y minúsculas, deben escribirse tal cual, por lo que conviene un tipo de letra muy legible; de otra parte no admiten espacios en blanco y en ocasiones llegan a ser muy largos, lo que implica que lo más normal era que, al formatear este documento, muchos de ellos se vieran repartidos entre dos líneas. Eso es tarea que normalmente se hace mediante la división silábica, marcando con un guión el punto de separación. Pero, y aquí está el tercer problema, estos nombres utilizan muy a menudo los guiones en su interior, de tal manera que si se autorizaba en ellos la división silábica, se corría el riesgo de que el lector no supiera qué guiones forman parte del nombre y cuáles son fruto de la división silábica<sup>4</sup>.

Al final he decidido con ellos las siguientes medidas.

- Las comillas tipográficas (« ») se reservan para los nombres de mandatos y variables. Eso ayudará a su identificación pues a partir de este momento sabemos que si un nombre está encerrado entre comillas tipográficas, es el nombre de un mandato o el de una opción de usuario de Emacs.
- Se utiliza una letra de tipo sanserif, que resulta muy legible.
- Los guiones que forman parte del nombre se escriben como guiones largos (—).

En consecuencia los guiones *cortos* en nombres encerrados en comillas tipográficas, no forman parte del nombre propiamente dicho, sino que son fruto de la división silábica. En todo lo demás estos nombres, si se quiere usarlos, deben ser tecleados exactamente igual que aparecen aquí, respetando la distinción entre letras mayúsculas y minúsculas.

**Teclado y mandatos en ciertos contextos:** A lo largo del texto aparecen varias tablas de resumen de mandatos. En ellas tanto las transcripciones de teclado como los nombres de mandatos aparecen en letra de tipo máquina de escribir.

---

numerosos nombres: Return, Retorno, Intro, Enter, etc. En definitiva: la tecla que genera los saltos de párrafo.

<sup>4</sup>Por lo menos el lector que no sepa nada de inglés y que por lo tanto no sepa distinguir los guiones de separación silábica de los que forman parte del nombre, que en Emacs se emplean siempre para separar palabras.



Con el mismo tipo de letra se muestran las *líneas de texto* que se sugiere que se introduzcan en el fichero de configuración de Emacs, o que se ponen como ejemplo de posible configuración.

**Cuestiones terminológicas:** A lo largo de este documento normalmente me refiero a  $\LaTeX$ , pero gran parte de lo que se dice es aplicable también a  $\TeX$  y a otros formatos derivados de él (Plain- $\TeX$ , Asm- $\TeX$ , Info- $\TeX$ , etc).

Asimismo, en muchos lugares me refiero a las *unidades estructurales* de  $\LaTeX$ . Con ellas pretendo referirme a los distintos niveles en los que se puede estructurar un documento de  $\LaTeX$ : parte, capítulo, sección, subsección, etc. Normalmente la documentación de  $\LaTeX$  se refiere genéricamente a estas *unidades* con el nombre de “*sección*”; pero siendo las *secciones* un ejemplo concreto de este tipo de unidades, no me parece un buen nombre para referirme genéricamente a ellas.

Aunque técnicamente, tanto los mandatos de Emacs como los de  $\LaTeX$  son en realidad “*macros*”, en este documento, a efectos de obtener una mayor claridad, normalmente he utilizado el término “*mandato*” para referirme a los de Emacs, y “*macro*” para referirme a  $\LaTeX$ .

Por último, debo decir que en cuanto a la terminología específica de Emacs (región en lugar de selección, punto en lugar de cursor, *buffer* en lugar de fichero que se está editando, etc), en líneas generales he sido bastante laxo, pues he pensado que al novato en Emacs le ayudará más a entender lo que un mandato hace si, por ejemplo, digo que afectará sólo *al texto seleccionado* que si digo que afecta *exclusivamente a la región*.

Ruego a los puristas de Emacs que me perdonen por ello, si fuera posible.

## 1. Preparación inicial

### 1.1. Conseguir e instalar Emacs y sus paquetes de extensión

Prácticamente todas las distribuciones de Linux incluyen la última versión de Emacs. Por lo tanto si no la tenemos instalada casi con seguridad que podremos instalarla desde nuestra propia distribución (el procedimiento variará según la distribución que sea).

En todo caso, quien desee descargar las fuentes de la última versión de Emacs puede hacerlo de “<ftp://ftp.gnu.org/gnu/emacs>”. En el momento de redactar este texto la última versión disponible era la 21.3. Una vez descargados los fuentes habrá que configurar, compilar e instalar. Para ello lo mejor es seguir las instrucciones del fichero INSTALL incluido entre las fuentes.

Desde la versión 20.3, Emacs incluye el paquete Ref $\TeX$ , y desde la versión 20.4 incluye también el paquete Bib $\TeX$ , por lo que lo normal es que una vez instalado Emacs ambos paquetes estén disponibles. De no ser así, podríamos descargar dichos paquetes de sus páginas web correspondientes que son, respectivamente:

- <http://zon.astro.uva.nl/~dominik/Tools/reftex/>
- <http://www.ida.ing.tu-bs.de/people/dirk/bibtex/>

Aunque mi consejo en este caso es actualizar nuestra versión de Emacs, descargando la última versión tal y como se acaba de indicar.

AUC  $\TeX$  es incluido en muchas distribuciones de Linux, pero en sí mismo no está integrado en Emacs, sino que debe ser instalado aparte. Si nuestra distribución lo incluye, basta con indicarle que lo instale. En caso contrario podemos descargar la última versión (que en el momento de redactar estas páginas es la 11.14), desde “<ftp://ftp.gnu.org/pub/gnu/auctex/>”. En tal caso el fichero INSTALLATION, incluido entre las fuentes, nos dará las instrucciones precisas sobre cómo instalar AUC  $\TeX$ .

## 1.2. Configurar Emacs para que todo funcione junto

La configuración inicial de Emacs, tras la instalación de AUC  $\TeX$ , es de momento suficiente para nosotros y no hay por qué cambiarla. Mas adelante iremos viendo los diferentes aspectos que son personalizables y decidiendo si queremos o no cambiarlos.

Por ello en este momento el único parámetro cuyo funcionamiento por defecto conviene cambiar es el relativo a la activación del paquete Ref $\TeX$ . Este paquete implementa un *modo menor* de Emacs (“reftex-mode”) que se puede activar o desactivar a voluntad por el procedimiento habitual<sup>5</sup>. Pero como en las próximas páginas asumiremos que se encuentra siempre activado —dada su gran utilidad—, lo mejor es asegurarnos de que se active automáticamente al abrir algún fichero en formato  $\LaTeX$  (o dicho con más precisión: al activarse los modos mayores “LaTeX” o “latex”).

Para ello hay que añadir la siguiente línea<sup>6</sup> al fichero de configuración de Emacs<sup>7</sup>:

---

<sup>5</sup>Es decir, habría que escribir: “M-x reftex-mode”. Esta orden activa el modo reftex, si estaba desactivado, o lo desactiva en caso contrario.

<sup>6</sup>La línea de ejemplo sólo funcionará si AUC  $\TeX$  está activado. En caso contrario, en lugar de LaTeX habría que escribir «latex» (todo con minúsculas)

<sup>7</sup>Ese fichero se encuentra en el directorio *home* de cada usuario, y se denomina *.emacs*. El punto inicial lo convierte en fichero oculto. Además de este fichero, hay otros ficheros que afectan a la configuración de Emacs. Véase al respecto el apéndice sobre la personalización de Emacs (pág. 84).

## Preparación inicial

---

```
(add-hook 'LaTeX-mode-hook 'turn-on-reftex) ; Activar reftex con AucTeX
(setq reftex-plug-into-AUCtEX t) ; Conectar a AUC TeX con RefTeX
```

La primera línea provoca que al cargarse el modo mayor LaTeX se active automáticamente reftex. La segunda línea hace que AUC TeX delegue en RefTeX para todo lo relativo a la generación de etiquetas<sup>8</sup>.

Asimismo, si normalmente sólo vamos a escribir ficheros en formato L<sup>A</sup>TeX, podemos añadir la siguiente línea:

```
(setq TeX-default-mode "LaTeX-mode") ; Modo ordinario para ficheros .tex
(setq TeX-force-default-mode t) ; Activar siempre dicho modo.
```

La primera de estas líneas informa a Emacs de que para los ficheros de extensión “tex” queremos usar el modo LaTeX y no otros modos posibles (Plain-TeX, Asm-TeX, Info-TeX, etc). La segunda línea instruye a Emacs de que la línea anterior es obligatoria y no una simple sugerencia. De este modo conseguiremos que siempre que Emacs abra un fichero de extensión “.tex” automáticamente active el modo L<sup>A</sup>TeX. En otro caso el propio Emacs intentaría determinar por el contenido del fichero qué modo activar y en caso de duda activaría el modo Plain-TeX, lo que luego nos obligaría a activar manualmente el modo L<sup>A</sup>TeX (lo cual, por cierto, tampoco es tanta dificultad).

*En relación con TeX, Emacs por defecto incorpora los modos denominados “tex”, “latex”, “plain-tex”, “infotex” y “slitex”. AUC TeX sobrescribe los tres primeros a los que además cambia el nombre por, respectivamente, “TeX”, “LaTeX” y “plain-TeX”. De lo anterior se deduce que en general las variables en las que los nombres “TeX” y “LaTeX” figuran con las típicas mayúsculas de TeX (la “T” y la “X” en mayúsculas y la “e” en minúscula, proceden de AUC TeX, y aquellas en las que los nombres son totalmente en minúsculas, son nativos de Emacs.*

*Al abrirse un fichero de extensión “.tex”, Emacs necesita determinar cuál de los diferentes modos activar. Para ello se vale de las variables «TeX—default—mode» y «TeX—force—default—mode». En principio Emacs lee el contenido del fichero y atendiendo a él intenta determinar si se trata de un fichero L<sup>A</sup>TeX o SliTeX; si está claro que no es de ninguno de los dos, se activará el modo “plain-TeX” y si se tienen dudas, se activará el modo indicado por la variable «TeX—default—mode». Ahora bien: a esta variable sólo se acude cuando hay dudas sobre si un fichero es o no L<sup>A</sup>TeX; y en ficheros que estén vacíos (porque acaban de ser creados), Emacs no tiene dudas de que no son L<sup>A</sup>TeX, ya que carecen de una sentencia inicial con la orden `\documentclass{}`. En consecuencia para ficheros vacíos se activa siempre*

---

<sup>8</sup>Aún no lo hemos visto, pero los mandatos de AUC TeX en ocasiones nos solicitan automáticamente una etiqueta para ciertas partes del texto; en otros casos nos permiten crear etiquetas propiamente dichas, o referencias a dichas etiquetas... De lo que se trata es de que todas estas tareas las haga RefTeX —que está especializado en ellas— y no AUC TeX.

el modo “plain-TeX” salvo que hayamos dado el valor “t” (verdadero)<sup>9</sup> a la variable «TeX—force—default—mode» que provoca que Emacs no intente determinar el tipo de fichero, sino que directamente active el modo indicado por «TeX—default—mode».

De otro lado, cuando se activa alguno de los modos mayores incorporados por AUC TeX, se ejecutan las tareas almacenadas en la variable «TeX—mode—hook», «LaTeX—mode—hook» o «plain—TeX—mode—hook», dependiendo, respectivamente, de cual de estos tres modos se active. Las tareas aquí almacenadas se ocupan por su parte de dar su valor inicial a otras variables utilizadas internamente por AUC TeX.

## 2. Utilidades básicas

En esta sección veremos un conjunto de utilidades que se dirigen, no tanto a la escritura de nuestros documentos fuente, como a aspectos más básicos: funciones de ayuda, visualización del fichero, formateado, etc.

### 2.1. Ayuda en línea de AUC TeX

La estupenda ayuda en línea de Emacs es complementada por la ayuda ofrecida por AUC TeX que incorpora dos mandatos:

Teclado	Nombre	Acción
C-c TAB	TeX-goto-info-page	Carga en un <i>buffer</i> la página INFO de AUC TeX
C-c ?	describe-mode	Muestra una ventana con todos los mandatos incorporados a AUC TeX

La página INFO de AUC TeX constituye su “*manual oficial*” y en lo que allí se dice está basado un muy alto porcentaje de este documento. No obstante creo que para el usuario poco experimentado en trabajar con AUC TeX es de más ayuda el segundo de los mandatos citados, pues su uso provoca que se abra una ventana en la que se relacionan todos los mandatos y asociaciones de teclas implementados por AUC TeX, de modo que haciendo click sobre el nombre de uno de los mandatos se muestre una breve ayuda, en inglés, sobre su utilidad y uso<sup>10</sup>.

<sup>9</sup>Una variable lógica sólo puede tener dos valores: verdadero o falso. En Emacs para representar ambos valores se utiliza, respectivamente, una “t” (de true = verdad) y la expresión “nil” derivada de *nada*. Estos valores deben introducirse siempre sin entrecorillarse, ya que constituyen *constantes* y no cadenas de texto.

<sup>10</sup>El mandato “C-c ?” equivale a “C-h b” («describe—bindings»), pero en lugar de mostrar todas las asociaciones de teclas cargadas, muestra tan solo las cargadas por el LaTeX-mode implementado por AUC TeX.

De hecho todas las veces que en este documento comento que cierta función o combinación de teclas no está documentada en AUC T<sub>E</sub>X, me refiero a funciones no mencionadas en su página info, pero que sí aparecen relacionadas por el mandato «describe—mode».

*El paquete bibtex carece de página info. Dispone de ayuda en formato html, aunque creo que el fichero no se instala con el paquete (en mi sistema no se instaló). Puede consultarse en la página web de mantenimiento de dicho paquete, mencionada en la sección 1.1, pág. 9.*

*En cuanto a RefT<sub>E</sub>X, dispone de una completa página info que se instala con el paquete. Puede ser cargada en Emacs, aunque no se ha implementado en el paquete ningún mandato que haga eso automáticamente, por lo que el fichero Info deberá cargarse manualmente<sup>11</sup>.*

## 2.2. Sangrado y reformateado del texto fuente

El término “*formatear*” tiene muchos sentidos. En un primer sentido puede decirse que, por ejemplo, L<sup>A</sup>T<sub>E</sub>X es un sistema de *formateo* de textos. Pero también puede emplearse ese término aplicado a un documento fuente de L<sup>A</sup>T<sub>E</sub>X para referirnos a su aspecto visual y facilidad de lectura.

Este es el sentido en el que se usa en esta sección.

### 2.2.1. El modo auto-fill de Emacs

Para formatear el texto fuente, se parte de las siguientes características de L<sup>A</sup>T<sub>E</sub>X:

- Da igual cuántos espacios en blanco consecutivos se acumulen: se tratarán todos ellos como si sólo hubiera uno.
- Los saltos expresos de línea se tratan como espacios en blanco salvo cuando haya una o más líneas completamente en blanco, en cuyo caso se realizará un salto de párrafo (uno sólo, con independencia del número de líneas en blanco).

Pues bien: ambas características —que posiblemente fueron implementadas para evitar errores muy corrientes en el mecanografiado de textos—, pueden ser

---

<sup>11</sup>Es posible acceder a los ficheros Info desde el mismo Emacs pulsando “C-h i” («info»). Ello nos llevará al nivel superior del sistema Info. Para acceder a la Info de RefTeX hay que pulsar “m” y luego escribir “ref`tex`”. También podemos buscar RefTeX con la función de búsqueda normal de Emacs y luego pulsar Intro. En ambos casos podremos acceder al fichero Info de RefT<sub>E</sub>X sin necesidad de salir de Emacs.

usadas para mejorar la apariencia visual y la “*legibilidad*” del documento fuente puesto que es posible utilizar los sangrados (espacios en blanco a la izquierda de la línea) para marcar visualmente la existencia de un entorno, u otras circunstancias.

Para que esto sea posible es imprescindible activar el modo menor “auto-fill” de Emacs. Este modo hace que cada vez que se inserte un espacio en blanco o un salto de línea, se compruebe si la línea supera o no cierta longitud y, en caso afirmativo, se retroceda hasta el último espacio en blanco y en su lugar se inserte un carácter de salto de línea seguido del número de espacios en blanco necesarios para asegurar que la nueva línea así creada tenga exactamente el mismo sangrado que la anterior<sup>12</sup>.

*La longitud de las líneas en el modo auto-fill viene determinada por el valor de la variable «fill—column».*

*En muchas preinstalaciones de Emacs, este se configura para que el modo auto-fill sea cargado por defecto cuando se editan ficheros de texto (y, para Emacs, los ficheros fuente de L<sup>A</sup>T<sub>E</sub>X son un tipo concreto de ficheros de texto). Si no fuera así, podríamos obtener este efecto escribiendo en el fichero de configuración de Emacs*

```
(add-hook 'text-mode-hook 'turn-on-auto-fill)
```

*O si lo preferimos, en lugar de “text” podemos escribir “LaTeX” para conseguir que el modo auto-fill se active automáticamente sólo para los ficheros L<sup>A</sup>T<sub>E</sub>X*

---

<sup>12</sup>El modo “auto-fill” de Emacs va en contra de la filosofía de la mayor parte de los editores de texto, ya que estos suelen partir de la idea de que el salto de línea hay que reservarlo para el caso de cambio de párrafo; lo que es tanto como decir que debe haber un solo salto de línea por párrafo, o lo que es lo mismo: *que el párrafo debe tener una sola línea*. Pero cuando se dice esto se está presuponiendo que en el momento de la edición del texto el editor deberá *introducir algunos cambios en el texto* para que pueda verse todo su contenido; pues sin introducir *dinámicamente* saltos de línea en el *buffer* de edición del texto, cada párrafo ocuparía una sola línea y de él sólo podríamos ver los primeros caracteres (tantos como ancha sea la ventana de edición). Es decir: incluso cuando se dice que no hay que introducir saltos de línea, en realidad estos se introducen *dinámicamente* durante la edición.

Es verdad que introducir saltos dinámicos tiene la ventaja de que ofrece más flexibilidad: un mismo texto se podrá editar sin problemas en ventanas de distinto ancho. Pero junto a tal consideración hay que tener en cuenta otras. En particular, en el caso de ficheros que vayan a ser compilados con T<sub>E</sub>X o con L<sup>A</sup>T<sub>E</sub>X, el compilador en caso de error nos informará de *en qué línea se localizó el error*, lo que significa que cuando editamos algo que se va a compilar, no nos sirven los editores que introducen saltos de línea dinámicos, sino que necesitamos uno que deje claro dónde están realmente las líneas, pues en caso contrario jamás localizaríamos el error. Y al mismo tiempo, nos interesa que las líneas no sean excesivamente largas ya que, en caso contrario, la información sobre el número de línea del error no nos sería de tanta ayuda para localizarlo y resolverlo.

En definitiva; cuando se edita algo que se va a compilar, lo que nos interesa es un editor como Emacs con el modo “auto-fill” activado. De hecho fue esta consideración una de las que más peso tuvo en mi decisión de editar mis ficheros L<sup>A</sup>T<sub>E</sub>X con Emacs.

Pero eso fue al principio. Luego fuí descubriendo más ventajas.

### 2.2.2. Reformateado automático del texto

Estando activado el modo “auto-fill” y cargado el paquete AUC T<sub>E</sub>X, conforme vayamos escribiendo nuestro fichero fuente se irán aplicando los siguientes niveles de sangrado:

- Para cada grupo abierto en una línea y no cerrado hasta una línea posterior, se aumenta el sangrado en dos espacios en blanco adicionales. Ello nos permite detectar visualmente los grupos anidados<sup>13</sup>.
- En el interior de un entorno el sangrado aumenta en dos caracteres con respecto a la que hubiera en su exterior. Esto es así en todo caso salvo en los entornos “document”, “verbatim” y “tabular”<sup>14</sup>.
- En los entornos de tipo lista, se aplica la regla anterior, pero las líneas que empiezan por la macro `\item`, disminuyen su sangrado en dos espacios (para marcar visualmente mejor cada *item* de la lista).

*La cantidad de espacios en blanco que se utiliza como indentación está controlada por las siguientes variables:*

- `«LaTeX-indent-level»`
- `«LaTeX-item-indent»`
- `«LaTeX-brace-indent-level»`

*Respectivamente cada una de estas variables controla el número de espacios en blanco que se añadirá a la indentación por cada entorno anidado, por cada item en una lista y por cada grupo abierto.*

### 2.2.3. Mandatos para el reformateado

Mientras estamos escribiendo el documento, el reformateado automático sólo tiene en cuenta la línea en curso y no el resto del párrafo, y además no se aplica hasta que se pulsa un espacio en blanco o RET en un lugar que sobrepase la anchura

---

<sup>13</sup>Recuérdese que en L<sup>A</sup>T<sub>E</sub>X las llaves abren y cierran grupos, por lo que cuando incluimos entre llaves los argumentos de una macro de L<sup>A</sup>T<sub>E</sub>X, estamos utilizando grupos que provocarán el efecto de indentación que se acaba de describir.

<sup>14</sup>Las razones son diferentes en cada caso. Para el entorno “document” la razón estriba en que, aunque técnicamente sea un entorno, en realidad dentro de él está contenido todo el texto visible del documento, por lo que no tiene sentido sangrarlo. En cuanto al entorno “verbatim”, el sangrado afecta sólo a la línea que lo declara, pero no a su contenido ni a su cierre, puesto que la esencia de un entorno de este tipo es poner en el documento *exactamente lo que hay*, sin conversiones de ningún tipo; no tiene pues sentido que se añadan espacios en blanco no expresamente escritos por el autor. En cuanto al entorno tabular, supongo que no se ve afectado por el sangrado automático debido a su complejidad interna. De hecho ninguno de los mandatos de sangrado que se verán a continuación afecta al entorno tabular. Sin embargo, si escribimos una macro personalizada que actúe como *envoltorio* para el entorno tabular, las apariciones en el texto de tal macro, sí se verían afectadas por el reformateo automático.

máxima autorizada para la línea. Esto lleva a que los resultados sean razonablemente buenos mientras se escribe por primera vez, pero no tan buenos cuando se modifica el texto que ya estaba escrito, añadiendo o suprimiendo fragmentos. Por ello se proporciona un conjunto de mandatos dirigidos al reajuste de la indentación. Todos ellos comparten el prefijo “C-c C-q”:

Teclado	Nombre	Zona reformateada
C-c C-q C-p   M-q	LaTeX-fill-paragraph	Párrafo
C-c C-q C-e	LaTeX-fill-environment	Entorno
C-c C-q C-s	LaTeX-fill-section	Unidad estructural
C-c C-q C-r   M-g	LaTeX-fill-region	Región

**NOTA:** Los mandatos que reformatean el párrafo y la región tienen DOS ASOCIACIONES alternativas de teclado, siendo indiferente que se utilice una u otra. Eso es lo que se ha querido representar en la tabla anterior separando con una barra vertical las dos asociaciones alternativas.

Todos estos mandatos realizan las siguientes acciones (cada uno sobre la zona de texto a la que afecta):

- Suprimir los saltos de línea simples de la zona (es decir: los que no impliquen cambio de párrafo).
- Recalcular el lugar en donde deben colocarse los nuevos saltos de línea teniendo en cuenta que se aplicarán los niveles de sangrado expuestos en el anterior epígrafe para entornos y grupos anidados.

Si cualquiera de estos mandatos es precedido de un prefijo numérico<sup>15</sup> el reformateado se hará insertando los espacios en blanco necesarios para que las distintas líneas del párrafo aparezcan “*justificadas*” tanto por el lado derecho como por el izquierdo, lo que mejora sensiblemente la apariencia del texto fuente sin efecto alguno en la salida formateada que L<sup>A</sup>T<sub>E</sub>X generará.

#### 2.2.4. Sangrado manual de líneas e inserción de tabuladores

En la mayor parte de los editores de textos, la tecla TAB introduce un carácter de tabulación en el lugar en donde sea pulsada. En Emacs, sin embargo la tecla TAB tiene como efecto el *sangrar* la línea en la que se pulse, y en un documento L<sup>A</sup>T<sub>E</sub>X ese sangrado está sujeto a ciertas reglas, de modo que —y esto es una sorpresa

---

<sup>15</sup>En Emacs la secuencia de teclas “C-u” seguida de un número se denomina *prefijo numérico*; su efecto es enviar al mandato que se teclee a continuación un argumento consistente en un número. Así, “C-u 2 C-n” ejecutará el mandato “C-n” con el argumento numérico 2. A este argumento numérico se le llama *prefijo* porque hay que teclearlo antes que el mandato al que se aplica. “C-u” no seguido de ningún número, también es un *prefijo numérico* y normalmente equivale a “C-u 4”



para muchos usuarios novatos— cuando se pulsa la tecla TAB no se obtiene un tabulador sino, según los casos, un desplazamiento de toda la línea a la derecha, o absolutamente ningún efecto.

Para insertar un verdadero tabulador en el lugar donde está el cursor, debemos pulsar “C-q TAB”.

*Pero no es una buena idea hacerlo, ya que ese carácter de tabulación será sustituido automáticamente por espacios en blanco en el momento en que se ordene guardar en disco los cambios, a no ser que establezcamos la variable «TeX—auto—untabify» con el valor de “nil”.*

De otro lado para que la tecla TAB produzca un *sangrado de la línea* es preciso que la línea anterior (sin contar las líneas en blanco) tuviera algún tipo de sangrado (en cuyo caso TAB sangrará la línea actual al mismo nivel) o que se trate de la primera línea de un entorno o de un grupo sin cerrar, en cuyo caso se insertará un sangrado similar al que Emacs realiza automáticamente conforme se escribe el documento.

En consecuencia, si queremos sangrar manualmente un párrafo debemos añadir manualmente los espacios en blanco en la primera línea y luego usar la tecla TAB en el resto de las líneas, de arriba hacia abajo. Aunque debe tenerse en cuenta que si indentamos manualmente un fichero, cualquier indentación automática que luego acometamos suprimirá la indentación manual.

*La tecla TAB ejecuta el mandato «LaTeX—indent—line». La tecla LFD (“C-j”) ejecuta «reindent—then—newline—and—indent». Este mandato produce el resultado que obtendríamos pulsando por orden las teclas TAB, RET, TAB.*

### 2.3. Selección de texto (la región)

A las órdenes ordinarias que permiten *establecer la marca* —en terminología de Emacs— o *seleccionar texto*, en la terminología más extendida, en los ficheros de tipo L<sup>A</sup>T<sub>E</sub>X se añaden las siguientes dos órdenes capaces de reconocer entornos y unidades estructurales de L<sup>A</sup>T<sub>E</sub>X:

Teclado	Nombre	Efecto
C-c *	LaTeX-mark-section	Selecciona la sección actual
C-c .	LaTeX-mark-environment	Selecciona el entorno actual

La utilidad de estos mandatos estriba en que al reconocer las *unidades estructurales básicas* de los documentos L<sup>A</sup>T<sub>E</sub>X, permiten con facilidad aplicar cualquiera de los mandatos que actúan sobre *la región* a una de estas unidades.

No obstante hay que aclarar, respecto del primero, que aunque reconoce las distintas unidades estructurales de los documentos  $\text{\LaTeX}$  (parte, capítulo, sección, etc.), no controla su distinto nivel jerárquico; es decir: partiendo de la posición del cursor busca las sentencias anterior y posterior que declaren alguna unidad estructural, y selecciona el texto entre ellas, sin importarle si la unidad estructural inferior está o no jerárquicamente subordinada a la anterior, con lo que el efecto puede ser el de que *realmente* no llegue a seleccionarse *toda* la unidad estructural de que se trate.

De hecho este problema se produce en gran parte de los mandatos de AUC  $\text{\TeX}$  referidos a unidades estructurales.

## 2.4. Resaltado gráfico de la sintaxis

Ejecutándose Emacs en una terminal con posibilidades gráficas, uno de los procedimientos para facilitar la legibilidad del documento es el del resaltado gráfico de la sintaxis, el cual consiste en diferenciar gráficamente —mediante colores distintos— lo que es texto de lo que son órdenes para el formateo del texto.

*Para que esta posibilidad funcione es preciso tener activada la opción «global—font—lock—mode». Hay además varias opciones de usuario que afectan a cómo funcionará esta función. Entre otras merecen destacarse las siguientes:*

- *«font—lock—maximum—size»:* Determina el tamaño de fichero a partir del cual esta opción se desactiva automáticamente. Esto se debe a que la decoración sintáctica exige mucha memoria: hacerla en un fichero demasiado grande puede exigir un porcentaje muy alto de recursos del ordenador, que se traduciría en una merma importante del rendimiento. Por lo tanto solamente se aplica en los buffers de tamaño inferior al indicado por esta variable. Si queremos que el decorado sintáctico funcione siempre, hay que dar a esta variable el valor “nil”
- *«font—lock—maximum—decoration»:* Determina el nivel de decoración de fuentes. Puede ser mínimo (valor “nil”), máximo (valor “t”), medio (valor numérico) o una lista en cuyo caso cada entrada indicará el nivel numérico para cada modo mayor de Emacs. Por ejemplo:

```
((c-mode . t) (c++-mode . 2) (latex-mode . 1))
```

*Este ejemplo significaría en el el modo C habría que usar la máxima decoración, en modo latex la mínima y en modo C++ algo superior a la mínima.*

- *«font—lock—highlighting—faces»:* Este grupo de opciones determina los colores concretos que se usarán para los distintos apartados de la sintaxis (comentarios, palabras clave, funciones, etc.)

*Personalmente entiendo que esta utilidad ayuda mucho a la legibilidad, pero, sin embargo, no está muy depurada. En ocasiones algún color se enquista y no hay forma de quitárselo de encima. Por ello me gusta mantenerla en el nivel mínimo, con lo que, además, se ahorra memoria.*

## 2.5. Auto-completado

Tras haberse cargado AUC  $\text{T}_{\text{E}}\text{X}$ , Emacs está en condiciones de reconocer la mayor parte de las macros específicas de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , en consecuencia las funciones generales a las que Emacs (y, en general, Unix) llama “*completion*”, y que podríamos traducir por “Auto-completado”, se adaptan a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  y así cuando se trata de insertar unidades estructurales, Emacs “*sabe*” entre cuáles podemos elegir; si se trata de insertar un entorno, Emacs tiene registrado el nombre de muchos de los entornos y nos los ofrece como posibilidad; si se trata de escribir una macro, Emacs “*conoce*” muchas de ellas e incluso sabe que requieren ciertos argumentos.

El auto-completado en Emacs no funciona aquí de manera distinta a la general, tal vez con la única peculiaridad de que cuando hay que seleccionar una macro, entorno o unidad estructural, y se escribe el nombre de una que sea desconocida para Emacs, este la *recordará* como perteneciente a la categoría de que se tratara (macros, entornos, etc), durante toda la sesión de trabajo.

Así en el caso de que, por ejemplo, insertemos un entorno de creación propia (generado mediante la orden `\newenvironment{}`), normalmente la primera vez Emacs no lo conocerá y, en consecuencia, habrá que teclear su nombre completo sin podernos ayudar de la función de auto-completado. Pero a partir de entonces, cada vez que queramos insertar un entorno —durante la misma sesión de trabajo— nuestro entorno personalizado figurará en la lista de entornos disponibles.

El auto-completado funciona, asimismo, con los argumentos de ciertas macros de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ; sobre todo las que exigen la inserción de un *nombre de fichero*. En estos casos Emacs *leerá* en nuestro directorio de trabajo los ficheros que, por su extensión, sean *candidatos* a constituir el argumento de la macro, y nos ofrecerá una lista con ellos; o, si sólo hay uno, directamente nos lo sugerirá cuando pulsemos SPC en el minibuffer.

*Para el auto-completado Emacs conoce numerosas macros y entornos de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Sin embargo en un documento normalmente se utilizan paquetes que amplían las posibilidades de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  incorporando nuevos entornos y macros. Asimismo nosotros mismos podemos definir nuevas macros y entornos, los cuales, en principio, no podrían ser conocidos por Emacs y, en consecuencia, no figurarán —al menos inicialmente— entre las opciones de auto-completado, así como tampoco lo harán las etiquetas ya insertadas en nuestro documento.*

*Podemos conseguir, no obstante, que todas estas macros, entornos y etiquetas lleguen a ser conocidos por Emacs simplemente activando la comprobación automática de ficheros, la cual por defecto está desactivada debido a que puede llegar a ralentizar el funcionamiento normal de Emacs.*

*Para activar esta opción hay que asignar el valor «t» a las siguientes variables:*

**« $\text{T}_{\text{E}}\text{X}$ —parse—self»:** *Esta variable determina que al cargarse en memoria un fichero, se realice una comprobación sintáctica del mismo en el caso de que no se le haya asignado ningún estilo.*

«**TeX—auto—save**»: Esta variable determina que se guarde información sobre el fichero cada vez que sea guardado en disco.

La información sobre los ficheros se guarda en un subdirectorio llamado “auto” y es muy útil, especialmente cuando estamos trabajando con varios documentos.

Podemos activar esta comprobación automática con carácter general, para algún fichero concreto o durante una sesión de trabajo.

Para activarla con carácter general debemos escribir en nuestro fichero `.emacs` las siguientes dos líneas.

```
(setq TeX-parse-self t)
(setq TeX-auto-save t)
```

Para activarla sólo en ciertos ficheros, debemos almacenar estas variables como variables de fichero (véase la [pág 84](#)), y para obligar a Emacs a que genere manualmente dicha información (lo que probablemente sea la mejor opción), podemos ejecutar el mandato «**TeX—normal—mode**» (“`C-c C-n`” o “`C-c #`”). Este mandato borra toda la información sobre el buffer y aplica de nuevo las tareas asociadas a la carga de un modo mayor. Antes de ello guarda el buffer incluyendo información sobre etiquetas, macros, etc.

Ciertamente almacenar esta información ayuda mucho a Emacs, pues le permite conocer las macros y entornos definidos por el usuario, las etiquetas repartidas a lo largo de los documentos que componen el proyecto, etc. El inconveniente está en que el funcionamiento del sistema se ralentiza demasiado cuando esta función está activada. Por ello una solución intermedia puede ser la de limitar la información que se almacenará. Para ello hay que modificar las variables «**TeX—auto—regexp—list**» y «**TeX—auto—parse—length**». La primera de estas variables contiene una lista de las expresiones regulares que coinciden con la información del documento que se almacenará. La segunda contiene el número de caracteres (a contar desde el principio del documento) en donde se recabará información.

Si tenemos en cuenta que casi toda la información significativa se encuentra en el encabezado, podemos usar la función de Emacs «**what-cursor-position**» (“`C-x =`”) para conocer la longitud de nuestro encabezado y dar dicho valor a la variable «**TeX—auto—parse—length**» con lo que posiblemente recabaríamos toda la información útil salvo las etiquetas definidas a lo largo del documento; lo cual tampoco es demasiado inconveniente si usamos `RefTeX` para manejarlas (véase la [sección 7](#), [página 47](#)).

En cuanto a «**TeX—auto—regexp—list**», podemos asignarle cualquiera de los siguientes valores constantes:

«**TeX—auto—empty—regexp—list**»: Provoca que no se almacene ninguna información.

«**TeX—auto—minimal—regexp—list**»: Provoca que se almacene solo la información relativa al estilo del documento.

«**TeX—auto—label—regexp—list**»: Provoca que se almacene sólo información sobre las etiquetas de `LATEX`.

«**LaTeX—auto—regexp—list**»: Provoca que se almacene información sobre las macros normales de `LATEX`.

«**plain—TeX—auto—regex—list**»: Provoca que se almacene información sobre las macros normales de *plain-TeX*.

«**TeX—auto—full—regex—list**»: Provoca que se almacene información sobre todas las macros de *LaTeX* utilizadas en el documento.

### 3. Ayuda general para la escritura del texto fuente

Sin perjuicio de otras funciones más específicas para cierto tipo de macros (fuentes, unidades estructurales, entornos y etiquetas), el modo de edición LaTeX ofrece una ayuda general para la escritura del documento fuente que consiste, de un lado, en el comportamiento especial ante ciertos caracteres, y, de otro, en un conjunto de mandatos dirigidos a facilitar la escritura de las macros de LaTeX conocidas por Emacs.

#### 3.1. Caracteres con tratamiento especial

Hay caracteres que tienen en LaTeX un significado especial o un uso particular. Algunos de ellos reciben un tratamiento especial en Emacs. Concretamente los siguientes (se usa un espacio en blanco para separarlos):

% " ‘ ’ << >> \$ { }

Como podemos ver algunos son *caracteres reservados* de LaTeX y otros son simplemente caracteres con un uso especial, pero no específicamente reservados.

##### 3.1.1. Marcas de comentario ( % )

###### Insertar comentarios en una zona de texto

En los lenguajes de programación los comentarios no sólo sirven para lo que su nombre indica (escribir aclaraciones para el texto), sino que pueden tener usos alternativos. Uno de ellos es el de eliminar temporalmente de la compilación un bloque de texto, pero sin borrarlo. En tal caso necesitaríamos insertar una o varias marcas de comentario al principio de cada una de las líneas del bloque al que queremos eliminar temporalmente. Para ello disponemos de los siguientes mandatos:

Teclado	Nombre	Acción
C-c ;	TeX-comment-region	Marca como comentario toda la zona seleccionada
C-c % C-c ’	TeX-comment-paragraph	Marca como comentario todo el párrafo

Cualquiera de estas órdenes admite un prefijo numérico que indicará cuántos signos de comentario hay que poner al principio de cada línea. Así, por ejemplo,

“C-u 3 C-c %” insertará “%%” al principio de cada una de las líneas del párrafo actual.

*En Emacs la regla es que “C-u” no seguido de ningún número, equivale a “C-u 4”. Sin embargo según el manual oficial de AUC T<sub>E</sub>X, el prefijo “C-u” seguido inmediatamente de alguno de estos mandatos, sin incluir número alguno como argumento, tiene el efecto de eliminar las marcas de comentario. Tras haberlo probado en mi instalación, lo que dice el manual sólo funciona en el caso de las marcas de comentario en la región (“C-u C-c ;”), siempre y cuando la zona seleccionada tuviera marcas de comentario al principio de sus líneas. Pero respecto de los párrafos, cuando se teclaa “C-u C-c %” se ejecuta de manera normal el mandato «TeX—comment—paragraph», con el parámetro “4”, es decir: se insertan cuatro marcas de comentario, si el párrafo no las tenía; y si sí las tenía, las cuatro marcas de comentario se insertan en el párrafo siguiente (lo que significa que posiblemente estemos ante un fallo de AUC T<sub>E</sub>X).*

De los dos mandatos, parece que el primero es mucho más útil, porque combinado con los mandatos que nos permiten seleccionar de golpe un párrafo (“M-h”), una sección (“C-c \*”) y un entorno (“C-c .”) en realidad nos ayuda a marcar como comentario casi cualquier bloque lógico de texto de un documento L<sup>A</sup>T<sub>E</sub>X con sólo dos mandatos: uno para seleccionar y otro para marcar la selección como comentario.

### Eliminar marcas de comentario

Para eliminar marcas de comentario disponemos de los siguientes mandatos, de los que sólo la primera asociación de teclas es mencionada por el *manual oficial* de AUC T<sub>E</sub>X:

Teclado	Nombre	Acción
C-u C-c ;   C-c :	TeX-un-comment-region	Elimina los comentarios de la zona seleccionada
C-c "	TeX-un-comment	Elimina comentarios

Por el solo nombre de «TeX—un—comment—region» sabemos que este mandato eliminará las marcas de comentario de la región. Pero el nombre del segundo mandato, «TeX—un—comment», es muy ambiguo. Según mis pruebas —ya que el mandato no está documentado—, este mandato *elimina la marca de comentario* del párrafo.

Por otra parte si un párrafo o región tienen más de una marca de comentario, cada ejecución de estos mandatos eliminará una, aunque también podemos proporcionar un prefijo numérico.

Para terminar diré que los mandatos que eliminan marcas de comentarios se detienen en cuanto encuentran una línea que carezca de tal tipo de marca, aunque aún queden líneas en la zona seleccionada o en el párrafo.

### 3.1.2. Dobles comillas

En  $\LaTeX$  el signo de las dobles comillas no se utiliza habitualmente sino que en su lugar se emplean, bien las comillas simples de la orientación adecuada (‘ ‘ para abrir comillas y ’ ’ para cerrarlas), bien los caracteres << >> para insertar lo que se suele denominar *comillas tipográficas*.

Emacs nos facilita la introducción de estos caracteres asociándolos a la tecla ("). Cuando la pulsamos se ejecuta el mandato «TeX—insert—quote» que analiza el contexto y decide si hacen falta unas comillas de apertura o de cierre. En el primer caso inserta los caracteres ‘ ‘ y en el segundo inserta ’ ’. Si queremos generar el carácter de las dobles comillas propiamente dicho, debemos pulsar dos veces consecutivas dicha tecla.

Esta utilidad resulta especialmente cómoda en el caso de los teclados españoles; porque como el carácter de comillas simples de apertura es el acento grave, y en los teclados españoles la pulsación de un acento por sí sola no genera ningún carácter sino que después hay que pulsar la barra espaciadora, resulta que para insertar unas comillas de apertura en documentos  $\LaTeX$  normalmente tendríamos que *pulsar cuatro teclas*. Gracias a esta utilidad una sola tecla será suficiente.

*«TeX—insert—quote» comprueba si estamos en posición de abrir o de cerrar comillas. En el primer caso inserta los caracteres asociados a la variable «TeX—open—quote» y en el segundo los asociados a «TeX—close—quote». En consecuencia cambiando el valor de esta variable podemos hacer que la tecla de las dobles comillas genere, por ejemplo, las comillas tipográficas. Para ello bastaría con escribir en el fichero de configuración de Emacs las siguientes sentencias:*

```
(setq TeX-open-quote '"<<") ; Comillas tipográficas de apertura
(setq TeX-close-quote '">>") ; Comillas tipográficas de cierre
```

### 3.1.3. Símbolo del dólar (\$)

No hay que insistir demasiado en el significado que el símbolo del dólar tiene en  $\LaTeX$ : constituye una *manera alternativa* de marcar los entornos matemáticos y en consecuencia:

- Los símbolos de dólar de un documento deben ir por parejas: el primero activará el entorno matemático y el segundo lo desactivará.
- Dentro de un entorno matemático que no haya sido iniciado con el símbolo dólar, sino de otra manera (mediante “(” o mediante “\begin{math}”), no deben insertarse signos de dólar, pues ello provocaría un error.

En consecuencia, cuando se inserta un signo de dólar tienen lugar las siguientes acciones:

- Si tras la comprobación oportuna se concluye que no hay ningún entorno matemático abierto (lo que implica que el número total de dólares en el documento es par), se insertará el nuevo dólar, lo que iniciará un entorno matemático.
- En el caso de que estemos ya en un entorno matemático, Emacs se asegura de que este haya sido iniciado con otro signo de dólar. Cuando encuentra el signo de dólar coincidente, durante unos instantes desplaza el cursor al lugar donde dicho signo se encuentra para que el usuario sepa que con este signo está, efectivamente, cerrando el entorno matemático previamente abierto.
- Si estamos en un entorno matemático que no fue abierto con un signo de dólar, se emite un pitido y un mensaje de error<sup>16</sup> y el dólar solicitado no se llega a insertar. En estas circunstancias solamente pulsando “C-q \$” podríamos insertar un dólar (que generaría un error en la compilación).

*El mandato asociado a la tecla \$ se denomina «TeX—insert—dolar». Si se le facilita un argumento numérico, se insertan tantos signos de dólar como el argumento indique.*

*Por otra parte, si tras haber insertado un doble dólar (“\$\$”) más adelante se inserta un dólar simple, se generará automáticamente un doble dólar que concuerde con el anterior. La verdad es que no veo muy claro eso para qué sirve; aunque también es cierto que nunca he usado L<sup>A</sup>T<sub>E</sub>X para escribir matemáticas, con lo que no estoy muy al tanto del funcionamiento del entorno math.*

### 3.1.4. Llaves de apertura y cierre de grupos

En L<sup>A</sup>T<sub>E</sub>X las llaves tienen un significado especial, pues abren y cierran grupos. Unas llaves descompensadas generan siempre un error en el documento, y a veces estos errores son especialmente difíciles de detectar (aunque Emacs ayuda a ello con sus distintos niveles de sangrado para grupos sin cerrar; véase la sección 2.2.2, pág. 15).

Sin duda lo mejor es siempre que se inserte un signo de apertura de llaves (`{`), insertar inmediatamente el signo de cierre (`}`); de esa manera seguro que no olvidaremos cerrar las llaves que acabamos de abrir.

El mandato para la inserción de llaves es «TeX—insert—braces», (“C-c {”): su efecto es insertar simultáneamente un signo de apertura de llaves y otro de cierre.

---

<sup>16</sup>En depende de la concreta configuración del sistema el cómo se realicen las advertencias de error. A veces son sonoras (pitidos), otras veces son sólo visuales, y en ocasiones se combinan ambas posibilidades.



Además, cuando se ejecuta en modo gráfico, Emacs ofrece la utilidad general de resaltar paréntesis, llaves y corchetes, que se supone que van siempre por pares. Así: colocándose el cursor sobre uno de los dos miembros, se resaltarán el concordante; y si no lo hay, o el que hay no es del mismo tipo, se indicará mediante colores la discordancia.

Otro mandato relacionado con los grupos, pero más general, es “C-c }” («up—list»). No es un mandato implementado por AUC T<sub>E</sub>X, pero resulta útil en caso de grupos muy anidados.

Este mandato lleva el cursor al final del grupo en el que se encuentra. O mejor dicho, en caso de paréntesis anidados «up—list» saca el cursor del nivel actual y lo adelanta hasta el lugar en donde continúa el nivel anterior. El mandato actúa tanto sobre grupos creados con paréntesis, como con llaves o corchetes.

Por ejemplo: si tenemos cinco grupos anidados de la siguiente manera:

```
{ { { { { } } } } }
```

y el cursor se encuentra entre la segunda y la tercera llave de apertura, el mandato “C-c }” colocará el cursor entre la penúltima y la última llave. Antes de ejecutarlo el cursor estaba en el segundo nivel de anidamiento tras su ejecución el cursor pasa a estar en el primer nivel de anidamiento exactamente en el punto en el que termina el nivel de anidamiento en el que antes se encontraba.

Con argumento numérico saltará tantos niveles de anidamiento como se le indiquen.

## 3.2. Facilidades generales para introducir macros de L<sup>A</sup>T<sub>E</sub>X

Además de ciertos mandatos específicos para algunos tipos especiales de macros, Emacs ofrece tres procedimientos genéricamente dirigidos a ayudar con la escritura de macros de L<sup>A</sup>T<sub>E</sub>X. Estos procedimientos son los siguientes:

### 3.2.1. M-TAB: TeX-complete-symbol

El mandato «TeX—complete—symbol», asociado a “M-TAB”<sup>17</sup> produce una especie de *auto-completado* directamente en la ventana de edición de Emacs; es decir: se leerá la parte escrita de una palabra y se buscará la macro de L<sup>A</sup>T<sub>E</sub>X que empiece de la misma manera, si sólo hay una; y si hay más de una completará hasta el punto en el que el nombre de estas macros empieza a diferenciarse; tras ello podemos seguir escribiendo el nombre, para diferenciarlo, o volver a pulsar “M-TAB” para que se muestre una lista con todos los nombres que coinciden con el texto tecleado.

---

<sup>17</sup>**ATENCIÓN:** Cuando Emacs se está ejecutando en un entorno gráfico, este es uno de los casos en donde sólo es posible obtener el efecto de la tecla *meta* pulsando la tecla ESC (si no sabe de qué estoy hablando, consulte la nota 2).

Por lo tanto para ejecutar esta función hay que empezar por escribir una barra invertida (`\`), ya que en  $\LaTeX$  las macros empiezan con dicho carácter, a continuación debemos escribir el principio del nombre buscado y, finalmente, pulsar “M-TAB” para que Emacs busque el resto del nombre de la macro. Si se encuentra una sola macro que empiece por el texto escrito se completará el nombre escrito; si se encuentran varios nombres, se pondrá la parte común, y si no se encuentra ninguna macro de  $\LaTeX$  que coincida con lo escrito, se emitirá un pitido y un mensaje de error. Si tras poner la parte común a las varias macros encontradas volvemos a pulsar “M-TAB” se mostrará una lista con todas las macros de  $\LaTeX$  conocidas por Emacs, que empiezan igual que el texto escrito.

Así, por ejemplo, si escribimos “`\newc`” y luego pulsamos “M-TAB”, se completará el nombre hasta `\newco` pues podríamos haber escrito el principio de la macro `\newcommand` o el de `\newcounter`. Si escribimos “`\newcom` M-TAB” completará hasta `\newcommand`.

### 3.2.2. TeX-insert-macro y TeX-electric-macro

Esta segunda vía para completar el nombre de las macros de  $\LaTeX$  es, en mi opinión, preferible al procedimiento anterior.

- El mandato «TeX—insert—macro» está asociado a las combinaciones de teclas “C-c C-m” y “C-c RET”: tras pulsar cualquiera de ellas se nos pedirá que, en el minibuffer de Emacs, introduzcamos el nombre de la macro deseada, pudiendo usar aquí la función general de auto-completado de Emacs asociada a las teclas SPC y TAB.
- «TeX—electric—macro» nos pregunta por el nombre de una macro de  $\LaTeX$  cuando introducimos el carácter “*eléctrico*” que activa este mandato. Para completar el nombre se pueden usar las funciones de auto-completado.

Ambas funciones son muy parecidas, por ello a partir de ahora hablaré de «TeX—insert—macro», pero lo que diga es aplicable también a «TeX—electric—macro» salvo que se advierta lo contrario.

¿Cuáles son las diferencias reales entre usar el primer procedimiento o cualquiera de estos otros dos? Básicamente las siguientes:

- Si hemos introducido el nombre de la macro mediante una de estas dos funciones y se trataba de una macro *conocida* por Emacs, se nos preguntará por sus opciones y parámetros; pero si para escribir el nombre de la macro hemos utilizado «TeX—complete—symbol», Emacs se limitará a escribir el nombre

completo de la macro, sin proporcionarnos ninguna ayuda adicional sobre sus posibles opciones y parámetros.

- « $\TeX$ —insert—macro» nos sugiere una macro por defecto de manera que para aceptarla basta con pulsar RET sin necesidad de escribir el principio de su nombre; y como la macro por defecto sugerida es siempre la última que se utilizó, esto resulta muy útil cuando queremos insertar varias veces la misma macro, ya que sólo la primera vez habrá que aclarar qué macro queremos introducir: y a partir de entonces podremos aceptar la macro sugerida como macro por defecto.

*Las diferencias entre « $\TeX$ —insert—macro» y « $\TeX$ —electric—macro» se encuentran fundamentalmente en el modo de trabajar internamente que cada una de ellas tiene, pero el funcionamiento externo es casi el mismo.*

*« $\TeX$ —electric—macro» por defecto no está asociada a ninguna combinación de teclas, pero podemos asociarla a la tecla “\”, en tal caso, cada vez que empecemos a escribir el nombre de una macro de  $\LaTeX$ , como éstas siempre empiezan por ese carácter, inevitablemente activaremos el mandato por lo que de modo automático se nos preguntará por el nombre de la macro que deseamos escribir.*

*Para asociar « $\TeX$ —electric—macro» a la tecla “\” no hace falta acudir a los procedimientos estándar de Emacs para la asociación de mandatos con secuencias de teclado, sino que basta con dar a la variable « $\TeX$ —electric—escape» cualquier valor distinto de “nil”. En cuanto se haga eso, la tecla “\” quedará asociada al mandato « $\TeX$ —electric—macro» el cual se activará automáticamente cada vez que intentemos escribir el nombre de una macro de  $\LaTeX$ .*

*Aparte de lo anterior, la única diferencia realmente significativa entre estos dos mandatos se encuentra en que mientras estamos escribiendo en el minibuffer el nombre de la macro que queremos insertar, en « $\TeX$ —insert—macro» podemos usar las funciones ordinarias de auto-completado de Emacs (con el tabulador y con la barra espaciadora), pero en « $\TeX$ —electric—macro» la barra espaciadora provoca siempre el efecto de completar y terminar.*

*Por otra parte hay dos variables que controlan el funcionamiento de estos dos mandatos:*

**« $\TeX$ —default—macro»:** *Contiene el nombre de la macro de  $\LaTeX$  que se sugerirá como macro por defecto la primera vez que se invoque a « $\TeX$ —insert—macro» o a « $\TeX$ —electric—macro». A partir de la primera invocación del mandato, se sugerirá como macro por defecto la última ejecutada insertada mediante esta función.*

**« $\TeX$ —insert—braces»:** *Del valor de esta variable depende el que tras el nombre de la macro a insertar se incluyan también dos llaves vacías o no se incluyan. Si la variable vale “nil” no se incluirán las llaves. En cualquier otro caso se incluirán.*

## 4. Macros $\LaTeX$ para especificar el formato de fuente

### 4.1. Insertar macros para formatos de fuente

Quizás los mandatos de Emacs para facilitar la escritura de documentos  $\LaTeX$  que con más frecuencia se utilicen, sean los destinados a especificar formatos de fuente.

Para insertar formatos de fuente en los documentos podemos usar dos versiones distintas de las macros. Hay una versión que se limita a activar cierto formato, el cual estará vigente hasta que se active otro incompatible con él. Por ejemplo: `\bf`, `\bfseries`, `\sl`, `\slseries`, etc.<sup>18</sup>. La segunda versión aplica el formato sólo al texto que se introduzca como argumento de la macro. Es para esta versión exclusivamente para la que existen en Emacs mandatos dirigidos a generar con rapidez el texto de la macro.

Todos los mandatos de Emacs que a continuación se exponen, funcionan de la misma manera: si se ejecutan en un lugar determinado escriben el texto que  $\LaTeX$  identifica como macro para activar cierto formato. Pero si se ejecuta el mandato cuando cierta porción de texto ha sido seleccionada (sobre una región), entonces se supondrá que la región constituye el texto al que hay que aplicar el formato en cuestión.

En cuanto a las especificaciones de formato que se pueden indicar, para su mejor comprensión las he agrupado en las siguientes categorías:

**Familia de la letra:** En  $\LaTeX$  se admiten tres familias de tipo de letra: roman, sanserif y máquina de escribir. La familia roman es la que se usa por defecto y normalmente no hay que indicarla de forma expresa. Con Emacs podemos especificar estos formatos pulsando las siguientes combinaciones de teclas:

Teclado	Texto insertado	Efecto
C-c C-f C-r	<code>\textrm{}</code>	Inserta texto roman.
C-c C-f C-f	<code>\textsf{}</code>	Inserta texto sanserif.
C-c C-f C-t	<code>\texttt{}</code>	Inserta texto tipo máquina de escribir.

**Grosor del trazo:** Con independencia de la familia a la que pertenezca la fuente, el trazo puede ser normal o grueso; en el segundo caso a la letra se la suele denominar *negrita* (en inglés *bold*). La macro que activa la letra negrita en  $\LaTeX$

<sup>18</sup>Las declaraciones del tipo “`\bf`”, “`\sl`”, etc., se consideran desfasadas. Actualmente la recomendación es que se sustituyan por las macros correspondientes en las que el texto afectado aparece como argumento, o por las órdenes globales “`\bfseries`”, “`\sffamily`”, etc. Al menos eso se señala en el documento “*Some obsolete commands and packages*” localizable en “<http://www.ctan.org/tex-archive/info/l2tabu/english/l2tabuen.pdf>”.

es `\textbf{}`. En Emacs la combinación de teclas “C-c C-f C-b” escribe dicho texto.

**Inclinación:** Un carácter puede tener tres niveles de inclinación: Ninguna inclinación (carácter normal), cierta inclinación (itálica o cursiva) e inclinación acentuada (oblicua o *slanted*). La mayoría de las veces es muy difícil apreciar diferencias entre la cursiva y la oblicua<sup>19</sup>.

A estos tres niveles posibles de inclinación  $\LaTeX$  añade lo que llama *texto enfatizado*. Este tipo de texto destaca de su entorno por el nivel de inclinación, de modo que un texto enfatizado que se encuentre entre texto normal, se verá en cursiva; pero si dentro de un texto en cursiva enfatizamos una porción de texto, éste se verá sin inclinación alguna. El texto enfatizado, por lo tanto, es siempre un texto que destacará de su entorno<sup>20</sup>.

Para escribir con facilidad las macros  $\LaTeX$  que generan todos estos efectos, en Emacs podemos realizar las siguientes acciones:

Teclado	Texto insertado	Efecto
C-c C-f C-i	<code>\textit{}</code>	Inserta texto en <i>cursiva</i> .
C-c C-f C-s	<code>\textsl{}</code>	Inserta texto <i>oblicuo</i> .
C-c C-f C-e	<code>\emph{}</code>	Inserta texto <i>enfatizado</i> .

**Otras especificaciones de formato:** A los formatos que se acaban de exponer, hay que añadir otros formatos posibles:

**Versales:** Se llama versales o versalitas a lo que en inglés se suele denominar *pequeñas mayúsculas*, es decir: unas letras mayúsculas que, sin embargo, no superan en altura a las minúsculas –al contrario de las mayúsculas normales–. Este tipo de letra, al que  $\LaTeX$  considera un *formato*, se activa en Emacs mediante la pulsación de “C-c C-f C-c”.

**Letra normal:** El concepto de letra normal es relativo y depende de la característica de formato de la que estemos hablando. Cuando hablamos del grosor del trazo, la letra normal es la que no tiene el grosor de la negrita, y entonces se la denomina “letra de grosor medio” o “regular”. Por el contrario, con relación a la inclinación de la letra, consideraríamos normal a la que no

---

<sup>19</sup>En realidad, la diferencia entre la *oblicua* y la *cursiva* es de diseño. La forma oblicua es la misma que la normal, pero ha sido “rotada” sobre su eje, se trata por lo tanto de *alteración matemática* de la forma estándar. Por el contrario, la forma itálica es, esencialmente, un diseño distinto de la normal.

<sup>20</sup>Una vez más, hay que precisar lo dicho en el texto. La macro “`\emph`”, no pretende afectar en realidad a la inclinación, sino al contenido: se trata de *destacar* un texto. Cuestión distinta es que, en la práctica, para destacarlo se utilice el recurso de la inclinación, pues ello es secundario y podría cambiar en futuras versiones del compilador. Por el contrario las `\textit` o `\textsl` funcionan, no en el nivel de la estructura lógica del documento, sino en el de la presentación.

tiene inclinación, hablándose entonces de “letra de perfil recto”. Pues bien, aunque habitualmente no es preciso indicar que queremos que la letra sea “normal”, pues para ello basta con que no hayamos especificado ningún grosor o inclinación especial, también hay macros para indicar que queremos un grosor medio y ninguna inclinación. Estas macros se generan desde Emacs pulsando, respectivamente, “C-c C-f RET” y “C-c C-f u”. Junto a ello, la combinación “C-c C-f C-n” genera la macro  $\LaTeX$  `\textnormal{}` que escribe texto *normal*.

## 4.2. Borrar o cambiar una especificación de formato

Si para un determinado texto se ha especificado cierto formato, podemos borrar dicha especificación pulsando “C-c C-f C-d”. También podemos *cambiar* la especificación de formato por otra distinta, pulsando “C-u” y, a continuación, el mandato correspondiente al formato deseado. Así, por ejemplo, la secuencia de teclas “C-u C-c C-f C-b”, cambiará la especificación de formato actual (la que sea) por negrita.

En  $\LaTeX$  es corriente, por otra parte, indicar formatos anidados. Así, para un fragmento de texto podemos indicar que lo queremos en versales, negrita e itálica, lo que se indicaría de la siguiente manera:

```
\textsc{\textbf{\textit{Texto afectado por el formato}}}
```

En tal caso, el mandato para suprimir un formato o para sustituirlo, afectará al formato más anidado en el que se encuentre el cursor. Es decir, si, en el ejemplo anterior, colocamos el cursor en algún lugar de la frase “*Texto afectado por el formato*” y pulsamos “C-c C-f C-d”, el formato que se borrará será la cursiva (`\textit`). Pero si colocamos el cursor sobre la penúltima llave de cierre, se borrará la negrita (`\textbf`).

### 4.3. Resumen sobre la especificación de formatos

Teclado	Texto insertado	Efecto
C-c C-f C-r	<code>\textrm{}</code>	Inserta texto roman.
C-c C-f C-f	<code>\textsf{}</code>	Inserta texto sanserif.
C-c C-f C-t	<code>\texttt{}</code>	Inserta texto tipo máquina de escribir.
C-c C-f C-b	<code>\textbf{}</code>	Inserta texto en <b>negrita</b> (trazo grueso).
C-c C-f RET	<code>\textmd{}</code>	Inserta texto normal (trazo medio).
C-c C-f C-i	<code>\textit{}</code>	Inserta texto en <i>cursiva</i> .
C-c C-f C-s	<code>\textsl{}</code>	Inserta texto <i>inclinado</i> .
C-c C-f C-e	<code>\emph{}</code>	Inserta texto <i>enfaticado</i> .
C-c C-f C-u	<code>\textup{}</code>	Inserta texto de perfil recto.
C-c C-f C-c	<code>\textsc{}</code>	Inserta texto EN VERSALES.
C-c C-f C-n	<code>\textnormal{}</code>	Inserta texto normal.
C-c C-f C-d		Borra la especificación de formato actual.
C-u		Sustituye el formato actual por el que se indique.

Para generar una especificación de formato el mandato que usa Emacs se denomina «TeX—font». La lista de las fuentes usadas por este mandato está almacenada en la variable «TeX—font—list». En ella se almacena una lista en la que cada entrada consta de tres elementos. El primer elemento es la tecla que activa la fuente en cuestión (es decir la tecla que hay que pulsar tras haber pulsado “C-c C-f”. Por ejemplo, para la negrita, en la lista el primer elemento será `^B`, ya que para activar ese formato, tras haber pulsado “C-c C-f” hay que pulsar “C-b”). El segundo elemento de la lista contiene la cadena que se insertará antes del punto, y el tercer elemento contiene la cadena que se insertará detrás del punto. Si la lista tiene un cuarto elemento, ello significa que si este es distinto de “nil”, siempre se debe reemplazar.

## 5. Unidades estructurales de L<sup>A</sup>T<sub>E</sub>X

### 5.1. Insertar unidades estructurales

En L<sup>A</sup>T<sub>E</sub>X hay siete unidades estructurales: cuyos nombres, en español, son: parte, capítulo, sección, subsección, subsubsección, párrafo y subpárrafo. No obstante los “capítulos” no están disponibles en algunos tipos de documento (como por ejemplo en el tipo *article*).

El mandato que genera las unidades estructurales de L<sup>A</sup>T<sub>E</sub>X es “C-C C-s”. Tras pulsarlo se nos preguntará en el minibuffer qué tipo de unidad estructural queremos insertar, tras indicar el tipo de unidad deseada, se nos preguntará por el título y después en ciertas ocasiones se nos preguntará si deseamos colocar una etiqueta identificativa. Sobre este último punto véase la sección 5.3.

Cuando ejecutamos este mandato, Emacs nos sugiere un nivel por defecto. Este nivel es el siguiente:

- Si se trata de la primera unidad estructural del documento, se nos sugerirá que empleemos “chapter” o “section” según nuestro documento sea de tipo “book” o de tipo “article”.
- En cualquier otro caso se nos sugerirá como nivel el actual, es decir: el correspondiente a la unidad estructural inmediatamente por encima del punto de edición. Ello, claro es, salvo en alguno de los casos que a continuación se explican.
- Si hemos llamado al mandato con un prefijo numérico vacío (es decir: si hemos pulsado “C-u C-c C-s” en lugar de “C-c C-s”, se nos sugerirá el nivel inmediatamente inferior al actual.

*El mandato para la inserción de unidades estructurales es «LaTeX—section», el cual admite un argumento numérico que determina la unidad estructural a insertar:*

- *Si el argumento se omite, o su valor es “nil”, se usará como nivel predeterminado el nivel actual.*
- *Si el argumento es un número negativo, dicho número se interpretará como número de niveles que hay que reducir (a partir del actual).*
- *Si el argumento consiste simplemente en “C-u” no seguido de ningún número, el nivel actual aumentará una unidad.*
- *Si el argumento es 0 o un número positivo, se interpretará en términos absolutos, de acuerdo con la siguiente lista:*
  0. *Parte (part)*
  1. *Capítulo (chapter)*
  2. *Sección (section)*
  3. *Subsección (subsection)*
  4. *Subsubsección (subsubsection)*
  5. *Párrafo (paragraph)*
  6. *Subpárrafo (subparagraph)*

*En consecuencia, y a modo de resumen: “C-c C-s” insertará una sección del nivel actual, “C-u C-c C-s” insertará una sección del nivel inmediatamente inferior, y “C-u -1 C-c C-s” insertará una sección del nivel inmediatamente superior al actual.*

## 5.2. Control de las tareas asociadas a la inserción de títulos

*La variable «LaTeX—section—hook» almacena la lista de tareas que se deben realizar cuando se inserta una nueva sección. Antes de que estas tareas se ejecuten se establece el valor de las siguientes variables internas (no modificables directamente por el usuario):*

- **level:** *indica el nivel numérico de la unidad estructural. Es fijado por el argumento del mandato «LaTeX—section».*



- **name**: indica el nombre de la macro necesaria para establecer el nivel indicado por level.
- **title**: el título de la sección. Por defecto es una cadena vacía.
- **toc**: entrada en el índice sistemático. Por defecto tiene el valor “nil”.
- **done-mark**: posición del punto inmediatamente después de ejecutar el mandato. Por defecto es “nil”, lo que significa que el punto se quedará detrás del texto que se inserte.

Con esas variables se ejecutan las tareas predefinidas y las que nosotros podamos añadir. Tareas predefinidas son las siguientes:

- «LaTeX—section—heading»: pregunta al usuario el nombre de la unidad estructural a insertar. Modifica las variables level y name.
- «LaTeX—section—title»: pregunta al usuario por el título de la unidad estructural. Modifica la variable title.
- «LaTeX—section—toc»: esta tarea —que por defecto se encuentra desactivada en las instalaciones normales de Emacs—, pregunta al usuario por la entrada del índice correspondiente a la unidad estructural que se está insertando. Modifica la variable toc. La desactivación de esta tarea se debe a que ya por defecto los macros de L<sup>A</sup>T<sub>E</sub>X añaden entradas en el índice para la mayoría de las unidades estructurales.
- «LaTeX—section—section»: inserta la macro L<sup>A</sup>T<sub>E</sub>X adecuada teniendo en cuenta el valor de las variables name, title y toc. Si toc tiene el valor “nil”, no se inserta ninguna entrada toc. Si toc o title son cadenas vacías, done-mark colocará el punto en el lugar donde se deberían haber insertado esas cadenas.
- «LaTeX—section—label»: inserta una etiqueta después de la macro que inserta la unidad estructural. Está controlado por la variable «LaTeX—section—label».

### 5.3. Asociación automática de etiquetas a los títulos

Acabamos de ver —si leemos la *letra chica*— que una de las tareas asociadas a la inserción de títulos es la de insertar una etiqueta identificativa del título. Esta etiqueta nos servirá, más adelante, para realizar referencias cruzadas a dicha sección.

Por defecto Emacs nos solicitará la inclusión de etiquetas, tan solo en los capítulos, secciones y subsecciones. Además la gestión de esa tarea puede ser hecha por AUC T<sub>E</sub>X o por RefT<sub>E</sub>X.

Mi consejo es que se utilice RefT<sub>E</sub>X, para lo cual hay que insertar en nuestro fichero de configuración la línea a la que me referí en el apartado 1.2 (pág. 10). Para el caso de que se haya hecho así, consúltese la sección dedicada a la generación de etiquetas (sección 7 pág. 47).

En caso de que no se haya conectado el funcionamiento de RefT<sub>E</sub>X con el de AUC T<sub>E</sub>X, el texto que escribamos cuando se nos pregunte por la etiqueta se añadirá como “*label*” de la sección, precedido de un prefijo indicador del nivel estructural de dicha sección.

*La variable «LaTeX—section—label», contiene el prefijo por defecto para las etiquetas añadidas automáticamente a las distintas unidades estructurales.*

*Si consiste en una cadena, se usará tal cual para cualquier tipo de unidad estructural. Si tiene el valor “nil”, no se insertará ninguna etiqueta. Si es una lista, se buscará en la lista el elemento cuyo nombre coincida con la macro correspondiente a la unidad estructural que haya que insertar en cada caso; si se encuentra ese elemento, se insertará como prefijo el elemento de la lista asociado a dicho nombre. Si el elemento asociado es “nil”, o no se encuentra el elemento adecuado, no se insertará ninguna etiqueta.*

Cuando la generación de etiquetas es controlada por RefT<sub>E</sub>X, en primer lugar, se solicita etiqueta al crear cualquier unidad estructural (y no sólo en algunas). El nombre de la etiqueta, por otra parte, es sugerido por el propio RefT<sub>E</sub>X partiendo del “título” que se haya dado a la sección.

#### **5.4. Utilización del modo menor “outline” de Emacs para ver la estructura y navegar por el documento**

Ver en pantalla la estructura del documento es una forma cómoda de comprenderlo con un solo golpe de vista. Con Emacs tenemos dos procedimientos distintos para mostrar en pantalla la estructura (compuesta exclusivamente por los títulos y subtítulos de que se compone el documento), de tal manera que podamos movernos exclusivamente entre los títulos.

El primer procedimiento se basa en el modo menor de Emacs “outline” y constituye una de las utilidades más potentes proporcionadas por AUC T<sub>E</sub>X.

Outline es un modo menor de Emacs pensado para mostrar la estructura de un fichero y desplazarse por ella. En ficheros L<sup>A</sup>T<sub>E</sub>X se considera que la estructura del documento está formada por sus títulos y subtítulos, definidos mediante las macros `\chapter`, `\section`, etc.

En consecuencia, al activar este modo menor podremos:

- Contraer o expandir los títulos.
- Movernos entre los títulos pasando por encima de su contenido.

### 5.4.1. Activar “outline” de forma compatible con AUC T<sub>E</sub>X

Pero hay un pequeño inconveniente: el modo menor “outline” utiliza para sus mandatos el mismo prefijo que AUC T<sub>E</sub>X lo que hace que sea difícil la convivencia entre ambos, ya que los dos inician sus mandatos con la secuencia de teclado “C-c” Esta coincidencia en el prefijo significa que en el conflicto entre un modo mayor y un modo menor, siempre prevalece el modo mayor, es decir: los atajos de teclado del modo menor outline no siempre funcionarán en AUC T<sub>E</sub>X a no ser que cambiemos uno de los dos prefijos.

*En esta tesitura es más sencillo cambiar el prefijo del modo menor outline, ya que este está controlado por la variable de Emacs: «outline—minor—mode—prefix». Podemos darle el valor que queramos, pero conviene asegurarse de que no le demos un valor que entre en conflicto con algún otro grupo de mandatos. Por ello normalmente se le da a esta variable el valor “C-c C-o” que es el que recomienda el manual de AUC T<sub>E</sub>X. En adelante asumiré que ese es el valor que le hemos dado al prefijo en cuestión.*

Una vez que hemos cambiado el prefijo de “outline”, para hacer uso de esta función basta con activar este modo, para lo cual hay que ejecutar el mandato de Emacs «outline—minor—mode»<sup>21</sup>.

Y ya estamos listos para ejecutar las utilidades que el modo outline nos proporciona, las cuales son las siguientes:

### 5.4.2. Mandatos que afectan a la visibilidad del documento

Este grupo de mandatos nos permiten mostrar u ocultar distintas partes del documento.

---

<sup>21</sup>Si estamos ejecutando Emacs en un entorno gráfico, también podemos activar el modo outline colocando el ratón sobre la barra de modos de emacs en la zona donde se informa de los modos menores activos —normalmente estará activo al menos el modo Fill—. Una vez allí, haciendo click con el botón derecho del ratón se mostrará en pantalla (mientras mantengamos presionado el botón del ratón) una lista con los modos menores disponibles. Desplazando el puntero del ratón a uno de los modos (en nuestro caso al modo outline) lo activaremos —si estaba desactivado— o lo desactivaremos en caso contrario.

Teclas	Mandato	Acción
C-c C-o C-t	hide-body	Oculto todo salvo los títulos.
C-c C-o C-a	show-all	Muestra todo el documento.
C-c C-o C-q	hide-sublevels	Muestra sólo los X niveles superiores (que se le deben facilitar como argumento).
C-c C-o C-c	hide-entry	Oculto las líneas entre este título y el próximo (aunque sea un subtítulo).
C-c C-o C-e	show-entry	Muestra el contenido del título actual, incluidos sus subtítulos si estaban ocultos.
C-c C-o C-d	hide-subtree	Oculto todo el contenido del título actual (texto y subtítulos).
C-c C-o C-s	show-subtree	Muestra todo el contenido del título actual (texto y subtítulos).
C-c C-o C-l	hide-leaves	Oculto el contenido del título actual y de sus subtítulos, pero no la cabecera de éstos.
C-c C-o C-k	show-branches	Muestra los subtítulos pero no el contenido de éstos.
C-c C-o C-i	show-children	Muestra los subtítulos de próximo nivel. Con prefijo se puede indicar el nivel al que se mostrarán los subtítulos.
C-c C-o C-o	hide-other	Lo oculto todo salvo el título en donde está el cursor y todos los títulos de los que éste desciende.

Los dos primeros mandatos son muy claros, pues afectan a todo el documento: el primero hace que sólo sean visibles los títulos (y subtítulos) y el segundo hace que todo el documento vuelva a ser visible.

El siguiente mandato “C-c C-o C-q” hace que sólo sean visibles los títulos de cierto nivel. El nivel de visibilidad se le facilita mediante un argumento numérico y se cuentan de la siguiente manera: el nivel 1 muestra sólo las sentencias de L<sup>A</sup>T<sub>E</sub>X `\documentclass` y la apertura y cierre del entorno “document”. A partir de ahí el resto de los niveles se numeran secuencialmente partiendo del nivel más alto posible para el tipo de documento de que se trate. Es decir El nivel 2 mostrará en los documentos de tipo “book” los capítulos y en los documentos de tipo “article” las secciones.

En cuanto al resto de los mandatos todos actúan sobre el título actual, aunque lo hacen de distinta manera:

- Los mandatos “C-c C-o C-c” y “C-c C-o C-e”, se usan cuando el punto de edición está exactamente sobre un título (unidad estructural) del documento, y ocultan o muestran exclusivamente al contenido *normal* de dicho título, pero sin afectar a los posibles *subtítulos*.
- Los mandatos “C-c C-o C-d” y “C-c C-o C-s” también esperan que el punto de edición esté exactamente sobre un título y afectan tanto a las líneas *normales* como a las que constituyen *subtítulos* del título sobre el que esté colocado el cursor.
- Una cosa intermedia entre tener el contenido de un título visible o invisible, es tener visible sólo los subtítulos. Esto se puede conseguir mediante la

pareja de mandatos “C-c C-o C-l” y “C-c C-o C-k”. El primer mandato también funciona con “C-c C-o TAB”.

- El mandato “C-c C-o C-i” muestra exclusivamente a los hijos del título actual, es decir: los subtítulos del nivel exactamente inferior.
- Por último, “C-c C-o C-o” oculta todo el documento salvo la sección en la que se encuentre el punto y aquellas de las que esta sea un subtítulo. Es posiblemente el más cómodo para ocultar las partes del documento en las que no se está trabajando.

### 5.4.3. Mandatos de movimiento por el texto

Este grupo de mandatos nos permite desplazarnos entre los distintos títulos visibles con rapidez. Téngase en cuenta que todos ellos ignoran los títulos que pudieran no estar visibles (por aplicación de los mandatos anteriores).

Teclas	Nombre del mandato	Se desplaza a
C-c C-o C-n	outline-next-visible-heading	Próximo título
C-c C-o C-p	outline-previous-visible-heading	Título anterior
C-c C-o C-f	outline-forward-same-level	Próximo título del mismo nivel
C-c C-o C-b	outline-backward-same-level	Anterior título del mismo nivel
C-c C-o C-u	outline-up-heading	Título del nivel inmediatamente superior

Para más información sobre el modo menor “outline”, consulte la ayuda de GNU Emacs.

*AUC T<sub>E</sub>X modifica lo necesario para que el modo “outline” de Emacs reconozca como líneas de cabecera a las unidades estructurales de L<sup>A</sup>T<sub>E</sub>X. Pero podemos también indicar nuestras propias unidades estructurales personalizadas modificando la variable «TeX—outline—extra». Esta variable consiste en una lista en la que cada elemento tiene dos partes. La primera parte es la expresión regular que permite identificar a la línea como línea de cabecera. El segundo elemento indica el nivel asignado a dicha línea.*

*Consulte la ayuda de Emacs sobre expresiones regulares y tenga en cuenta que como esta lista sólo analiza el principio de las líneas del fichero, en la expresión regular no es preciso indicar que la coincidencia debe estar necesariamente al principio de la línea, sino que automáticamente se añadirá, a la expresión regular que escribamos, el carácter “^” (indicador, en expresiones regulares, de que ésta se refiere al principio de una línea).*

*El siguiente ejemplo, recogido del manual de AUC T<sub>E</sub>X hace que \bibliography e \item sean considerados también líneas de cabecera, con \bibliography al mismo nivel que las secciones, e \item por debajo de los subpárrafos:*

```
(setq TeX-outline-extra
'(("{ \t]-*\\\\\\\\(bib\\)?item\\b" 7)
("\\\\bibliography\\b" 2)))
```

## 5.5. Usar RefT<sub>E</sub>X para ver la estructura del documento y navegar por ella

RefT<sub>E</sub>X proporciona un método rápido, más potente y gráficamente mejor conseguido que el que se acaba de ver: se trata del mandato «ref<sub>E</sub>tex—toc» (“C-c =”)<sup>22</sup>. Al ejecutarlo la pantalla se divide en dos ventanas; en una de ellas permanece el documento objeto de edición y en la otra se muestra su estructura de una manera que está gráficamente más conseguida, pues, a diferencia del modo “outline” que muestra la línea del código fuente que contiene un título tal y como está realmente escrita, RefT<sub>E</sub>X muestra exclusivamente el texto del título, al que se asigna el número que le correspondería en el índice sistemático, e indica su nivel jerárquico mediante el sangrado. Es decir: lo que vemos intenta parecerse a cómo quedará, tras la compilación, el índice sistemático del documento.

Además, en caso de documentos repartidos entre varios ficheros, se muestra el documento entero, con indicación de qué parte se encuentra en cada fichero, todo ello siempre que se hayan seguido las indicaciones que más adelante se especifican sobre cómo trabajar con varios ficheros (véase la sección 8, pág. 64).

Por lo demás, estando activa la ventana de índice, podemos realizar varias acciones previa pulsación de ciertas teclas<sup>23</sup>:

### Teclas de utilidad general

- **[?]** («ref<sub>E</sub>tex—toc—show—help») Muestra un sumario con las opciones disponibles.
- **[0-9]** Argumento prefijo. Se aplica a la próxima orden. Sirve, por ejemplo, para desplazarnos varias líneas de golpe.
- **[q]** («ref<sub>E</sub>tex—toc—quit») Cierra el índice sistemático.
- **[k]** («ref<sub>E</sub>tex—toc—quit—and—kill») *Mata* al índice sistemático, es decir: no sólo cierra la ventana que lo contiene, sino que lo elimina de la memoria (hasta que vuelva a ser llamado).

<sup>22</sup>Para que los mandatos de RefT<sub>E</sub>X estén disponibles, es preciso activar “ref<sub>E</sub>tex-mode” eso se puede hacer automáticamente al cargar un fichero L<sup>A</sup>T<sub>E</sub>X o manualmente. Véase la sección 1.2 (pág. 10) para más información sobre cómo automatizar la activación del modo menor ref<sub>E</sub>tex.

<sup>23</sup>En la próxima lista incluyo, junto a las teclas, el nombre del mandato ejecutado. Esa información no está disponible en la página Info de RefT<sub>E</sub>X y la he obtenido de la función “C-h b”. Cuando no recojo el nombre del mandato es porque ni siquiera con esa función he podido averiguarlo.

- **[x]** («ref<sub>tex</sub>—toc—external») Muestra el índice de un documento externo, en el caso de que nuestro documento utilice el paquete de L<sup>A</sup>T<sub>E</sub>X “xr”.

### Movimiento por el índice

- **[n, Flecha abajo, C-n, C-Flecha abajo]** («ref<sub>tex</sub>—toc—next») Mueve el cursor a la próxima línea.
- **[C-c C-n]** («ref<sub>tex</sub>—toc—next—heading») Mueve el cursor a la próxima cabecera<sup>24</sup>.
- **[p, flecha arriba, C-p, C-Flecha arriba]** («ref<sub>tex</sub>—toc—previous») Mueve el cursor a la anterior línea.
- **[C-c C-p]** («ref<sub>tex</sub>—toc—previous—heading») Mueve el cursor a la anterior cabecera.
- **[AvPag, RePag]** Mueve el cursor una pantalla hacia delante o hacia detrás.
- **[z]** («ref<sub>tex</sub>—toc—jump») Salta a la sección indicada. Previamente se debe haber tecleado el número de sección.

### Teclas para ver el documento principal

- **[SPC]** («ref<sub>tex</sub>—toc—view—line») Muestra en la otra ventana el lugar donde se encuentra el título seleccionado.
- **[f]** («ref<sub>tex</sub>—toc—toggle—follow») Activa / Desactiva el modo de seguimiento. En él la otra ventana de la pantalla va mostrando la localización de los títulos conforme nos vamos moviendo por la ventana de estructura; es como si siempre pulsáramos SPC.
- **[.]** («ref<sub>tex</sub>—toc—show—calling—point») Muestra en otra ventana el punto de inserción, es decir: el lugar desde el que se ejecutó el mandato «ref<sub>tex</sub>—toc».
- **[TAB]** («ref<sub>tex</sub>—toc—goto—line») Mueve el cursor a la ventana del documento en el punto correspondiente al título seleccionado.
- **[RET]** («ref<sub>tex</sub>—toc—goto—line—and—hide») Oculta la ventana de índice y coloca el cursor en el punto correspondiente al título seleccionado.

### Control de visualización en el *buffer* de índice:

- **[F]** («ref<sub>tex</sub>—toc—toggle—file—boundary») Activa / Desactiva la visualización de los indicadores de cambio de fichero cuando se trabaja con varios ficheros.

---

<sup>24</sup>Acabo de transcribir lo que dice la página Info de Ref<sub>T</sub>E<sub>X</sub>. Es cierto que C-n y C-p ejecutan mandatos distintos a “n” y “p”, pero la verdad es que yo no he observado ninguna diferencia entre moverse por las líneas o por las cabeceras.

- **[t]** («reflex—toc—max—level») Cambia el nivel máximo de títulos que se mostrarán. Debe ir precedido de un argumento numérico equivalente al nivel de los títulos visibles. Así “1t” mostrará sólo los capítulos, “2t” mostrará también las secciones, etc.
- **[C-c >]** («reflex—toc—display—index») Muestra el índice terminológico. Precedido del prefijo 2, restringe el índice a la sección en la que se encuentra el cursor.
- **[I]**: («reflex—toc—toggle—labels») Activa / Desactiva la visualización de etiquetas en el índice sistemático. Usado con argumento se preguntará qué etiquetas mostrar.
- **[c]** («reflex—toc—toggle—context») Cuando se están mostrando etiquetas, activa / desactiva la visualización de líneas de ayuda a la contextualización de las etiquetas.
- **[i]** («reflex—toc—toggle—index») Activa / Desactiva la visualización de entradas de índice.

### Actualización del *buffer*

- **[g]** («revert—buffer») Actualiza el *buffer* desde la lista interna de títulos mantenida en memoria por RefTeX, pero no vuelve a analizar el documento.
- **[r, R]** («reflex—toc—rescan») Analiza de nuevo el documento para actualizar y reconstruir el índice. Si la variable «reflex—enable—partial—scans» es distinta de “nil”, y el documento consta de varios ficheros, releerá sólo la parte correspondiente al fichero actual. Para releer todo el documento hay que pulsar “C-u r”.

## 6. Entornos

### 6.1. Utilidades generales para los entornos

Los entornos de L<sup>A</sup>T<sub>E</sub>X se abren con la orden `\begin{Nombre}` y se cierran con la orden `\end{Nombre}`, siendo *Nombre* la denominación del entorno.

En un documento L<sup>A</sup>T<sub>E</sub>X los entornos son fundamentales porque marcan bloques de texto sometidos a ciertas características que pueden ir anidándose. De hecho todo el documento propiamente dicho se considera un *entorno* encerrado entre las órdenes `\begin{document}` y `\end{document}`.

En relación con los entornos, en Emacs podemos:

- *Reformatearlos*, es decir: reescribirlos con los niveles de indentación adecuados.



- *Seleccionarlos* enteros, o sea: marcarlos como región.
- Ocultar o mostrar su contenido.
- Movernos entre las líneas de apertura y cierre.
- Cerrar un entorno que esté abierto.
- Insertar entornos.

De todas estas utilidades, la primera y la segunda ya se han examinado (véanse las secciones 2.2.3 en pág. 15 y 2.3 en pág. 17) y la última, que es la que más explicaciones requiere se examina en los próximos epígrafes. Por lo tanto a continuación explicaré las restantes, aunque antes debo hacer algunas aclaraciones comunes a todos los mandatos relativos a entornos:

- En caso de entornos anidados, cualquiera de estos mandatos afectará exclusivamente al último entorno abierto, contando desde la posición del cursor, el cual debe, obviamente, estar en algún lugar entre las sentencias de apertura y cierre del entorno.
- Por el mismo motivo: si en un documento  $\text{\LaTeX}$  ejecutamos un mandato que actúa sobre el entorno actual y el cursor no está dentro de algún entorno *ordinario*, la orden afectará al entorno “document” (siempre que el cursor estuviera dentro de él), es decir: a todo el texto de nuestro documento, o, con más precisión, a todo el texto del documento que se encuentre en el fichero actual; pero no a aquellas partes del texto que se encuentren en otros ficheros.
- Las órdenes que actúan sobre entornos exigen que en el *buffer* actual puedan localizarse las órdenes de apertura y cierre del entorno. Si alguna de dichas órdenes está en otro fichero (o aún no se ha escrito), el mandato no funcionará. Tampoco funcionarán estos mandatos en documentos excesivamente grandes, cuando hay mucha distancia entre las órdenes de apertura y cierre del entorno, por falta de memoria.
- Lo anterior sólo tiene como excepción el mandato *cerrar entorno* que exige tener acceso a la orden de apertura del entorno que se cerrará pero no requiere —obviamente— que la sentencia de cierre del entorno esté ya escrita.

### 6.1.1. Ocultar y mostrar el contenido de un entorno

Esta utilidad permite una visualización selectiva del documento; es decir: mientras estamos trabajando en un texto podemos hacer que ciertas partes de él no sean visibles para que así sea más fácil movernos por él o echarle un *vistazo* general que nos permita hacernos una idea de su contenido.

Para ello los mandatos a ejecutar son:

Teclado	Mandato	Acción
C-c C-z	LaTeX-hide-environment	Oculto el contenido de un entorno dejando visible sólo la primera línea.
C-c C-x	LaTeX-show-environment	Muestra el contenido de un entorno previamente ocultado.

En algunas implementaciones de AUC T<sub>E</sub>X estos mandatos tienen como prefijo no “C-c”, sino “C-c C-o”. Para asegurarse, lo mejor es pulsar “C-c ?” que nos mostrará en un buffer independiente todas las asociaciones de teclado provocadas por AUC T<sub>E</sub>X.

Estos mandatos están desactivados por defecto, es decir: la primera vez que ejecutemos cualquiera de ellos, Emacs nos solicitará confirmación para activarlo, pudiendo elegir entre activarlo para esa ocasión, para la sesión actual o para siempre.

A mí personalmente no me entusiasma el resultado de estos mandatos. Porque, de un lado, desaparece el contenido del entorno manteniéndose la primera línea (la orden `\begin{NombreEntorno}`), pero sin ninguna indicación de que el resto del entorno *está pero es invisible*, lo que, en mi opinión, es perturbador<sup>25</sup>. Pero, sobre todo, como el texto realmente está ahí pero no se muestra, cuando el cursor llega a la línea inicial de un entorno que ha sido ocultado, aparentemente queda bloqueado produciendo la sensación de que Emacs ha dejado de funcionar correctamente<sup>26</sup>.

### 6.1.2. Movernos entre las líneas de apertura y cierre de un entorno

En cualquier punto de un entorno podemos saltar automáticamente a su línea de apertura (orden `\begin{}`) o a la de cierre (orden `\end{}`). Para ello los mandatos a utilizar son:

Teclado	Mandato	Acción
M-C-a	LaTeX-find-matching-begin	Va a la línea de apertura del entorno actual
M-C-e	LaTeX-find-matching-end	Va a la línea de cierre del entorno actual

### 6.1.3. Cerrar un entorno

Mediante el mandato “C-c ]” («LaTeX—close—environment») podemos hacer que se escriba automáticamente la orden de cierre de un entorno. Cuando este mandato se ejecuta, Emacs busca hacia arriba el último entorno abierto y sin cerrar, comprueba cuál es y genera la orden adecuada para cerrar dicho entorno.

<sup>25</sup>En realidad sí hay una indicación: junto a la línea de apertura del entorno se muestran unos puntos suspensivos. Pero no es una indicación especialmente clara, ya que dichos puntos podrían formar parte del texto.

<sup>26</sup>Aunque parece lo contrario, el cursor no queda bloqueado, sino que como el texto *realmente está*, se produce una disociación entre el *buffer* que se edita (el que contiene el texto) y el que se muestra.

El mandato no se asegura, sin embargo, de que más adelante del cursor no esté ya la línea de cierre del entorno, por lo que si mediante esta orden cerramos un entorno que ya estaba cerrado, se producirá un error en el momento de la compilación.

## 6.2. Inserción y sustitución de entornos

Mediante el mandato “C-c C-e” podemos insertar automática y simultáneamente el par de órdenes de apertura y cierre de un entorno:

```
\begin{NombreEntorno}  
\end{NombreEntorno}
```

Si precedemos la orden de un prefijo numérico mediante “C-u”, en lugar de insertarse un entorno nuevo, se sustituirá el entorno actual por el que indiquemos (lo que exige que el cursor esté dentro de un entorno). Esta última posibilidad es especialmente útil en entornos de lista para cambiar el tipo de lista que sea.

Tras ejecutar el mandato Emacs nos preguntará qué entorno queremos insertar, sugiriéndonos un entorno por defecto. Tras aceptar dicha sugerencia o indicar otro nombre de entorno, se insertarán las órdenes de apertura y cierre adecuadas.

Si el entorno que hemos elegido para insertar se encuentra entre los que Emacs *conoce*, y dicho entorno dispone de argumentos opcionales o requiere la inclusión de parámetros, Emacs nos preguntará por tales opciones individualmente evitando así que, por error, omitamos algún parámetro del entorno.

En general insertar entornos mediante el mandato “C-c C-e” es preferible a hacerlo manualmente, ya que:

- Nos aseguramos, al escribir simultáneamente las órdenes de apertura y de cierre, de que todo entorno abierto en un punto del documento, se cerrará en otro punto.
- Para ciertos entornos se ofrece la ventaja adicional de que Emacs los conoce y sabe qué parámetros requieren, preguntándonos por ellos y evitando así que olvidemos el parámetro preciso (lo que más tarde generaría un error).

*El mandato que inserta o sustituye un entorno es «LaTeX—environment».*

*La variable «LaTeX—default—environment» determina el entorno que se ofrecerá por defecto la primera vez que se inserte un entorno. Téngase en cuenta que aquí —como en otras ocasiones— la noción “primera vez” no se refiere a “primera vez en el documento” sino a “primera vez en la sesión de trabajo actual”, es decir:*

cuando se carga AUC T<sub>E</sub>X, se lee el valor de esta variable, y cuando se inserta un entorno “distinto” mediante el mandato «LaTeX—environment», se cambia el valor de la variable para la sesión actual de trabajo, de tal modo que la próxima vez que ejecutemos dicho mandato, se nos ofrezca, como entorno por defecto, el último que fue seleccionado.

### 6.3. Notas sobre algunos entornos especiales

#### 6.3.1. El entorno “document”

Un entorno especial para Emacs es el entorno “document”. En L<sup>A</sup>T<sub>E</sub>X todo el texto de un documento debe encontrarse entre las sentencias `\begin{document}` y `\end{document}`. Todo lo que haya antes de este entorno forma parte del encabezado o preámbulo, el cual debe iniciarse siempre con la orden `\documentclass`.

Pues bien: para Emacs la orden `\documentclass` es una opción del entorno document, lo que significa que si pulsamos “C-c C-e” y elegimos como tipo de entorno el de “document”, se nos preguntará por el tipo de documento que deseamos escribir y las opciones del mismo, generándose así la orden `\documentclass` adecuada y, tras ella, el par de órdenes `\begin{document}` y `\end{document}`.

*Si el documento en el que nos encontramos está totalmente vacío, o el cursor está colocado en el primer carácter del documento, el entorno por defecto ofrecido no será el indicado por la variable «LaTeX—default—environment», sino siempre el de “document”.*

#### 6.3.2. Entornos de tipo lista

En los entornos de lista (`description`, `enumerate`, `itemize` y `list`), están disponibles dos procedimientos para generar un ítem de la lista: la combinación de teclas “C-c C-j” y la combinación “M-RET<sup>27</sup>”. Cuando generamos los ítems mediante este procedimiento, se comprueba el tipo de entorno que es y en el caso concreto del entorno “description” Emacs nos preguntará por la etiqueta para el ítem a insertar.

*El mandato que inserta un ítem es «LaTeX—insert—item». En teoría este mandato sólo debería funcionar dentro de un entorno de lista; pero lo cierto es que a mí me funciona siempre; es decir: el mandato no comprueba si se está o no dentro de un entorno de tipo lista.*

---

<sup>27</sup>En la documentación de AUC T<sub>E</sub>X se habla sólo de “C-c LFD”. La tecla “LFD”, no obstante, no existe en los teclados estándar actuales y su función es sustituida por “C-j”. En cuanto a la combinación M-RET, no está documentada en el manual de AUC T<sub>E</sub>X.

### 6.3.3. Entornos para objetos flotantes

Los objetos flotantes son un tipo especial de objetos de  $\LaTeX$  que no están asignados a un punto exacto del documento, sino que se permite que el propio  $\LaTeX$  los coloque en el lugar más adecuado.

Para insertar este tipo de objetos se utilizan dos entornos concretos: el entorno “figure” y el entorno “table”. Cuando colocamos, por ejemplo, una tabla dentro de un entorno “table” le estamos indicando a  $\LaTeX$  que esa tabla no forma, en sentido estricto, parte del texto y, por lo tanto, no es imprescindible que se coloque *exactamente* en el lugar donde la escribimos, sino que puede ponerse en otro lugar de la página facilitando así la confección de una “bella página”.

En consecuencia, cuando usamos “C-c C-e” para insertar un entorno y elegimos como entorno figure o table, Emacs nos hará algunas preguntas adicionales. En particular se nos preguntará:

- **float-to:** Se trata de indicarle a  $\LaTeX$  donde queremos que se coloque el objeto flotante. La respuesta puede ser: “h” (aquí), “t” (en la parte superior de una página), “b” (en la parte inferior de una página), o “p” (en una página especial para objetos flotantes). También podemos indicar cualquier combinación de estos valores teniendo en cuenta que  $\LaTeX$  intentará cumplirlos por el orden en el que se indiquen. Pero intentar cumplir esas indicaciones no significa que necesariamente se vayan a cumplir, ya que no siempre es posible colocar un objeto flotante en el lugar donde se haya indicado.
- **label:** Se trata de la etiqueta para este objeto flotante. Al valor que indiquemos aquí se le añadirá, un prefijo indicador de que se trata de una etiqueta de figura o de tabla.
- **caption:** Se trata del título que se pondrá al objeto flotante.
- **center:** Pregunta si el objeto debe o no centrarse.

*Para las dos primeras de estas preguntas una variable controla su valor por defecto. Así «LaTeX—float» controla el valor por defecto de la pregunta “float-to”, «LaTeX—figure—label» controla el valor del prefijo que se añadirá a la etiqueta en los entornos figure, y «LaTeX—table—label» controla el valor del prefijo que se añadirá a la etiqueta en los entornos table.*

### 6.3.4. Entorno matemático y modo matemático

$\TeX$  fue escrito por un matemático y siempre ha destacado especialmente por su excepcional calidad a la hora de formatear texto matemático. Por ello hay un entorno de  $\LaTeX$  (el entorno *math*) que tiende a ser considerado más como un

*modo* que como un *entorno*. Sus peculiaridades son muchas y no es cuestión aquí de explicarlas, tan sólo diré que Emacs asume que  $\LaTeX$  se utiliza con mucha frecuencia para escribir textos matemáticos y en consecuencia intenta simplificar la introducción de las muchas macros matemáticas previstas en  $\LaTeX$ . Para ello hace lo siguiente:

- AUC  $\TeX$  implementa un “modo menor” de Emacs llamado *math*. Este modo se activa pulsando “C-c ~”<sup>28</sup>. Al igual que en el resto de los modos menores, esta combinación de teclas funciona como un interruptor: si el modo matemático está desactivado lo activa y viceversa.
- La activación del modo matemático de Emacs implica que toda tecla precedida de cierto prefijo será sustituida por la cadena de texto que se haya asociado a dicha tecla. Así podemos, por ejemplo, asociar a la tecla “a” la cadena de texto `\alpha{}` de manera que simplemente pulsando el prefijo y la tecla “a” obtengamos escrita la cadena `\alpha{}`.

El prefijo por defecto para activar la función de reemplazo es el carácter de acento grave (‘).

*El mandato «`LaTeX—math—mode`» es el que activa el modo matemático que AUC  $\TeX$  incorpora a Emacs. Se trata de un modo menor que asigna a una tecla determinada por la variable «`LaTeX—math—abbrev—prefix`» (por defecto el acento grave) la función de leer la tecla pulsada a continuación y escribir, en lugar de dicha tecla, el valor asignado a la misma en la lista «`LaTeX—math—list`». Esta lista consta de parejas de elementos, el primero es una tecla y el segundo el nombre de la macro que hay que escribir en lugar de dicha tecla.*

*De otro lado, como «`LaTeX—math—mode`» no comprueba si realmente estamos o no en un entorno matemático, puede ser tentadora la idea de, para las personas que escriben pocas matemáticas, sustituir la lista de «`LaTeX—math—list`» por otra compuesta de mandatos corrientes, de modo que para introducir cualquier mandato baste con pulsar la tecla asociada a él precedida del acento grave. Incluso podríamos cambiar el prefijo del acento grave por el del acento agudo, y asociar a las teclas a, e, i, o, u, las macros “`\’a`”, “`\’e`”, etc, es decir: la forma de escribir caracteres acentuados al estilo de babel (usando sólo siete bits).*

*Aunque no le recomiendo esto a nadie, pues nos acostumbraría a una manera de escribir que sólo tendríamos a nuestra disposición en nuestro propio equipo previamente configurado.*

---

<sup>28</sup>Para obtener el carácter ~ en un teclado para PC español ordinario, podemos mantener pulsada la tecla Alt-Derecha al tiempo que pulsamos, bien la tecla “ñ”, bien la tecla “4” en el teclado normal (no en el numérico).

## 7. Etiquetas y referencias cruzadas

Genéricamente puede decirse que una referencia cruzada se produce cuando en una parte del texto se hace referencia a otra distinta, de tal manera que en el momento de la compilación, habrá que actualizar el contenido concreto de la referencia. Como cuando nos referimos al número de página de cierta sección, o a cierta nota al pie. En el momento de hacer esa referencia no sabemos qué número tendrá dicha página o qué número tendrá dicha nota al pie.

Realizar una referencia cruzada exige que hayamos colocado una *etiqueta* identificativa en la parte del documento a la que queremos referirnos, de modo que en el documento nos referiremos a la etiqueta y luego el compilador sustituirá las referencias a etiquetas por los valores correspondientes. Por razones obvias, cada *etiqueta* debe ser única en todo el documento.

En  $\LaTeX$  las referencias cruzadas son básicamente de tres tipos: etiquetas propiamente dichas, citas bibliográficas e índices terminológicos. Para todas ellas da soporte el paquete  $\text{RefTeX}$ , que además proporciona una utilidad para ver, desde una parte del documento vinculada a otra mediante una referencia cruzada, esa otra parte<sup>29</sup>.

### 7.1. Etiquetas propiamente dichas

#### 7.1.1. Estructura de las etiquetas de $\text{RefTeX}$

Llamo *etiqueta propiamente dicha* a la que es generada con la macro  $\LaTeX$  “ $\backslash\text{label}\{\}$ ”. Su peculiaridad estriba en que son *sensibles al contexto*. Es decir: cuando en algún otro lugar del documento generamos una referencia cruzada, el compilador de  $\LaTeX$  colocará, en el lugar de dicha macro, el valor de un contador asociado a la posición de la etiqueta. Pero qué contador sea depende del contexto de la etiqueta y así, por ejemplo, una etiqueta ubicada en una nota a pie de página, devolverá el valor del contador “*footnote*”, mientras que una ubicada en un entorno *figure* devolverá el contador del mismo nombre.

Esta característica de  $\LaTeX$  es la que aconseja que el nombre de estas etiquetas normales sea precedido de un *prefijo* que discrimine el tipo de etiqueta de que se trata. Tanto  $\text{AUC TeX}$  como  $\text{RefTeX}$  generan etiquetas con prefijos.

---

<sup>29</sup>Para que los mandatos de  $\text{RefTeX}$  estén disponibles, es preciso activar “*reftex-mode*” eso se puede hacer automáticamente al cargar un fichero  $\LaTeX$  o manualmente. Véase la sección 1.2 (pág. 10) para más información sobre cómo automatizar la activación del modo menor *reftex*.

RefTeX, concretamente, distingue los siguientes tipos de etiquetas: ecuaciones (prefijo “eq”), figuras (prefijo “fig”), enumeraciones (prefijo “item”), notas a pie (prefijo “fn”), tablas (prefijo “tab”) y secciones (prefijo “sec”).

*RefTeX reconoce como objetos numerados en donde es posible una etiqueta, la macro “\footnote” y todos los entornos nativos de L<sup>A</sup>T<sub>E</sub>X que pueden ser numerados, además de numerosos entornos definidos en varios paquetes de ampliación (endnotes, fancybox, floatfig, longtable, pincipar, rotating, sidecap, subfigure, supertab y wrapfig).*

*Para ver exactamente los objetos numerados admitidos, y en qué categoría es incluido cada uno de ellos, pulse “C-c ) ?”, y luego, tras haber leído la lista, “C-g” para anular el mandato.*

*De otro lado un conjunto de variables permite definir nuevos entornos o macros numerados que sean reconocidos por RefTeX como tales. Al respecto consúltese la página Info de RefTeX.*

Aparte del prefijo la etiqueta requiere un nombre que, en combinación con el prefijo, constituirá el nombre de la etiqueta.

En este punto, L<sup>A</sup>T<sub>E</sub>X admite cualquier carácter en el nombre de las etiquetas. Pero RefTeX, sin embargo, no permite que en el nombre de las etiquetas haya espacios en blanco ni caracteres no anglosajones.

### 7.1.2. Creación de etiquetas

La creación de etiquetas puede ser automática o expresa. Creación automática se produce cuando se insertan —mediante los mandatos adecuados de Emacs— unidades estructurales, o ciertos entornos como “figure”, “table” o “equation”. En estos casos el mandato de inserción pregunta por una etiqueta; y en el caso de que hayamos activado la conexión entre RefTeX y AUC T<sub>E</sub>X (véase la sección 1.2, pág. 10), AUC T<sub>E</sub>X llamará a RefTeX el cual, tratándose de ecuaciones y otros objetos numerados, generará automáticamente la etiqueta, y tratándose de unidades estructurales, propondrá un nombre de etiqueta al usuario para que lo acepte o modifique.

Para la creación expresa de etiquetas disponemos de la orden «reftex—label», asociada a la combinación de teclas “C-c (”. Este mandato:

- Es sensible al contexto; es decir: detecta si estamos dentro de alguna tabla, nota a pie, figura, etc., y determina así el tipo de etiqueta necesaria. No estando en ningún contexto especial, supone que el tipo de etiqueta a insertar será de “sección”.
- Dependiendo del tipo de etiqueta a insertar:



- Lo generará automáticamente e insertará sin consultar al usuario. Esto se hace para aquellos tipos de etiquetas que se refieren a objetos difíciles de identificar si no es por un número, como, por ejemplo, las ecuaciones, enumeraciones, items y notas al pie.
  - Generará un nombre descriptivo y le preguntará al usuario antes de insertarlo, permitiéndole su modificación. Así se hace fundamentalmente en las etiquetas referidas a secciones, en las que el nombre se genera a partir del título de la sección.
  - Directamente preguntará al usuario por el texto de la etiqueta. Así se hace en el caso de las figuras y tablas.
- Antes de insertar la etiqueta se asegura de que no haya otra igual, ya que las etiquetas deben ser únicas.

Por supuesto que es posible teclear directamente `\label{NombreEtiqueta}`, sin embargo se recomienda no hacerlo así, ya que entonces `RefTeX` no nos advertiría en el caso de que por error quisiéramos crear una etiqueta ya existente; aparte de que generando las etiquetas mediante `RefTeX`, el paquete *sabe* que la etiqueta se ha creado, y no tendrá que analizar posteriormente el documento para buscarla y localizarla.

*Las siguientes variables controlan la creación de etiquetas mediante RefTeX:*

- *«`reftex—insert—label—flags`»:* Contiene los indicadores que controlan la inserción de etiquetas. Consta de dos elementos, el primero determina si se intentará derivar la etiqueta del contexto (valor `t`) o si la etiqueta constará sólo del prefijo y un número (valor `nil`). El segundo elemento determina si se debe o no preguntar al usuario antes de insertar la etiqueta derivada.
- *«`reftex—format—label—function`»:* Si tiene un valor distinto de `nil`, deberá ser una función que proporcione la cadena que se insertará como definición de etiqueta.
- *«`reftex—string—to—label—function`»:* Contiene la función que convierte una cadena arbitraria en una etiqueta.
- *«`reftex—translate—to—ascii—function`»:* Se trata de una función que actúa como filtro para evitar que la etiqueta tenga caracteres con código ASCII superior a 127 (es decir: caracteres no anglosajones).
- *«`reftex—derive—label—parameters`»:* Parámetros para convertir una cadena en una etiqueta. Los parámetros son: número de palabras a usar, número máximo de caracteres, qué hacer con los caracteres ilegales, si hay o no que abreviar las palabras, qué separador se usará, etc.
- *«`reftex—label—illegal—re`»:* Contiene una expresión regular que recoge los caracteres ilegales en una etiqueta.
- *«`reftex—abbrev—parameters`»:* Parámetros para la abreviación de las palabras.

### 7.1.3. Hacer referencia a las etiquetas

Las órdenes que hacen referencia a las etiquetas creadas con `\label` son `\ref` y `\pageref`. La primera devolverá el contador asociado con la etiqueta, el cual varía según el tipo de etiqueta que sea (de sección, nota a pie, tabla, etc.). La segunda devuelve siempre el número de página en donde se encuentra la etiqueta.

El mandato para hacer en un lugar del texto una referencia a una etiqueta definida en otro, es «`reftex—reference`», asociado a “`C-c`”). Este mandato puede ser llamado manualmente, pero si hemos activado la comunicación entre los dos paquetes (véase la sección 1.2, pág. 10), también será llamado cuando insertemos con AUC `TEX` bien la macro `\ref`, bien `\pageref`. Si llamamos manualmente al mandato, la macro de `LATEX` que se insertará será `\ref` (aunque luego podremos cambiarla a mano).

La ventaja que ofrece este mandato es que abrirá una ventana que nos ayudará a localizar la etiqueta *sin necesidad de recordar su nombre*. Aunque sí deberemos recordar el tipo de etiqueta que era y, más o menos, el lugar del documento donde se encontraba.

Lo primero es indicar el tipo de etiqueta que estamos buscando. Esto puede deducirlo `RefTEX` del contexto, o preguntárnoslo, aunque, según mi experiencia, el contexto sólo funciona bien si estamos escribiendo texto en inglés, pues `RefTEX` se basa en ciertas abreviaturas en inglés para determinar a qué queremos hacer referencia, por lo que lo normal es que se nos pregunte en el minibuffer.

Podremos contestar lo siguiente:

- **[RET]**: se buscarán etiquetas de todo tipo.
- **[e]**: se buscarán etiquetas de tipo *equation*.
- **[f]**: se buscarán etiquetas de tipo *figure*.
- **[i]**: se buscarán etiquetas de tipo *enumerate*.
- **[n]**: se buscarán etiquetas de tipo *footnote*.
- **[N]**: se buscarán etiquetas de tipo *endnote*.
- **[s]**: se buscarán etiquetas de tipo *section*.
- **[t]**: se buscarán etiquetas de tipo *table*.

Si tardamos unos segundos en responder se mostrará una ventana con ayuda sobre lo que podemos responder. También podemos abrir directamente esa ventana, sin esperar esos segundos, pulsando “?”. Una vez abierta la ventana de ayuda, cualquier pulsación de tecla que se haga se tomará por respuesta a la pregunta (aunque no sea una respuesta válida) y se cerrará dicha ventana.

Si contestamos con cualquier carácter distinto de los indicados, se cancelará la ejecución del mandato.

Una vez determinado el tipo de etiqueta buscado, la pantalla se dividirá en dos ventanas. En una de ellas seguirá nuestro documento y en la otra se mostrará un *buffer* especial para la búsqueda de etiquetas. En él se muestra la estructura de nuestro documento, es decir: el *índice sistemático* compuesto por capítulos, secciones, subsecciones... En aquellas secciones en las que hay etiquetas del tipo buscado, bajo el nombre de la misma se verá la etiqueta y una línea de texto que pretende ayudarnos a identificar exactamente cada etiqueta. Si esa línea no fuera suficiente, podemos pulsar SPC, lo que hará que en la otra ventana se muestre la parte de nuestro documento en donde la etiqueta está declarada.

La idea es buscar en este *buffer* especial la etiqueta a la que queremos hacer referencia y, una vez localizada, pulsar RET para seleccionarla. Tras ello se cerrará este *buffer* especial y se creará la referencia.

#### 7.1.4. Funcionamiento del teclado en la ventana de selección de etiquetas

En el *buffer* de selección de etiquetas la pulsación de las siguientes teclas implica la realización de las siguientes acciones:

##### Teclas de utilidad general

- [?] («reftex—select—help») Muestra un sumario con las opciones disponibles.
- [0-9] Argumento prefijo. Se aplica a la próxima orden. Sirve, por ejemplo, para desplazarnos varias líneas de golpe.
- [q] («reftex—select—quit») Cierra el *buffer* de selección y anula la ejecución del mandato.
- [s] Cambia el tipo de etiqueta que se está buscando. Tras pulsar esta opción, se nos volverá a preguntar por el tipo de etiqueta.
- [x] Referenciar una etiqueta de un documento externo. Para ello es preciso usar el paquete de L<sup>A</sup>T<sub>E</sub>X “xr”.

##### Movimiento por el índice

- [n, C-n, Flecha abajo] («reftex—select—next») Mueve el cursor a la próxima etiqueta.
- [p, C-p, Flecha arriba] («reftex—select—previous») Mueve el cursor a la anterior etiqueta.
- [C-c C-n, C-flecha abajo] («reftex—select—next—heading») Mueve el cursor a la próxima unidad estructural (aunque en ella no haya etiquetas).

- **[C-c C-p, C-flecha arriba]** («ref<sub>tex</sub>—select—previous—heading») Mueve el cursor a la anterior unidad estructural (aunque en ella no haya etiquetas).
- **[AvPag, RePag]** Mueve el cursor una pantalla hacia delante o hacia detrás.
- **[b]** («ref<sub>tex</sub>—select—jump—to—previous») Salta a la posición en donde se dejó la última vez el *buffer* de selección, que normalmente será aquella en la que se encuentra la última etiqueta referenciada.
- **[z]** («ref<sub>tex</sub>—select—jump») Salta a la sección indicada. Previamente se debe haber tecleado el número de sección.

### Teclas para ayudarnos a localizar el contexto de las etiquetas

- **[SPC]** («ref<sub>tex</sub>—select—callback») Muestra en la otra ventana el lugar donde se encuentra la etiqueta seleccionada.
- **[f]** («ref<sub>tex</sub>—select—toggle—follow») Activa / Desactiva el modo de seguimiento. En él la otra ventana de la pantalla va mostrando la localización de las etiquetas conforme nos vamos moviendo por la ventana de estructura; es como si siempre pulsáramos SPC.
- **[.]** («ref<sub>tex</sub>—select—show—insertion—point») Muestra en otra ventana el punto de inserción, es decir: el lugar desde el que se ejecutó el mandato «ref<sub>tex</sub>—reference».

### Teclas para seleccionar etiquetas

- **[RET A]** («ref<sub>tex</sub>—select—accept») Crea la referencia con la etiqueta seleccionada.
- **[m+.,ua]** Se usan para el caso de que se quiera hacer una referencia múltiple. Véase el próximo epígrafe.
- **[I]** Usa de nuevo la última etiqueta referenciada.
- **[TAB]** («ref<sub>tex</sub>—select—read—label») Permite escribir el nombre de la etiqueta buscada, con ayuda de la función de auto-completado.

### Teclas para determinar la macro referenciadora

- **[v]** («ref<sub>tex</sub>—select—toggle—varioref») Activa / Desactiva la utilización de `\vref{}`, definida en el paquete “varioref”.
- **[V]** («ref<sub>tex</sub>—select—toggle—fancyref») Cambia entre `\ref{}`, `\fref{}` y `\Fref{}`. Estas dos últimas posibilidades requieren la inclusión del paquete “fancyref”.

### Control de visualización en el *buffer* de selección:

- [c] Activa / Desactiva la visualización de una línea para ayuda a la contextualización.
- [F] Activa / Desactiva la visualización de los indicadores de cambio de fichero cuando se trabaja con varios ficheros.
- [t] Activa / Desactiva la visualización de la tabla de contenido propiamente dicha. Precedido de un argumento numérico, determina el nivel de estructuración que se verá (véase la sección 5.5, pág. 38).
- [#] Activa / Desactiva la visualización de un *contador* de etiquetas.
- [%] Activa / Desactiva la visualización de etiquetas temporalmente anuladas por haberse puesto una marca de comentario en la línea que las contenía.

### Actualización del *buffer*

- [g] Actualiza el menú, desde la lista interna de etiquetas mantenida en memoria por RefT<sub>E</sub>X, pero no vuelve a analizar el documento.
- [r, R] Analiza de nuevo el documento para actualizar y reconstruir el menú. Si la variable «*reftex—enable—partial—scans*» es distinta de “nil”, y el documento consta de varios ficheros, releerá sólo la parte correspondiente al fichero actual. Para releer todo el documento hay que pulsar “C-u r”.

La inserción de referencias está controlada por un conjunto de variables entre las que destacan las siguientes:

- «*reftex—label—menu—flags*»: Lista de indicadores que controlan la referenciación de etiquetas.
- «*reftex—multiref—punctuation*»: Cadenas de puntuación para referencias múltiples.
- «*reftex—vref—is—default*»: Indica (cuando vale no “nil”) que la macro a generar, por defecto será `\vref`.
- «*reftex—fref—is—default*»: Indica, cuando vale no “nil”, que la macro a generar, por defecto será `\fref`.
- «*reftex—format—ref—function*»: Indica, cuando vale no “nil”, la función que produce la cadena que se usará como referencia.
- «*reftex—level—indent*»: Número de espacios para ser usados como indentación por cada nivel en la estructura.
- «*reftex—guess—label—type*»: Un valor distinto a “nil” significa que la creación de la referencia intentará determinar el tipo de referencia, para lo cual comprobará la palabra anterior al cursor y la comparará con las palabras almacenadas en «*reftex—label—alist*».

Para más información sobre estas variables, consúltese la página *Info* de RefT<sub>E</sub>X.

### 7.1.5. Realizar simultáneamente múltiples referencias

En la ventana de selección de etiquetas podemos seleccionar varias etiquetas para realizar una referenciación múltiple.

Para ello hay que empezar por marcar las distintas etiquetas a las que queremos referirnos. Para hacer esa marca disponemos de cuatro mandatos:

- **[m]** (`<<refTeX—select—mark>>`): la referencia generada será normal.
- **[+]** (`<<refTeX—select—mark—and>>`): La referencia generada irá precedida de la palabra “and”.
- **[-]** (`<<refTeX—select—mark—to>>`): La referencia generada irá precedida de un doble guión (--).
- **[,]** (`<<refTeX—select—mark—comma>>`): La referencia generada irá precedida de una coma.

De esta manera podemos determinar cómo queremos que se genere la referencia. Lo normal será *marcar* la primera referencia usando la “m”. Si queremos indicar un rango (por ejemplo de páginas), podemos marcar la segunda con “-”; pero también podemos separar las distintas referencias con comas o con la palabra “and”.

OJO: es muy importante el orden en el que se marquen las etiquetas, porque será el orden en el que se generarán las referencias.

Finalmente, tras haber marcado las referencias deseadas (y desmarcado pulsando “u” (`<<refTeX—select—unmark>>`) en caso de haber marcado alguna referencia por error), pulsando RET o “A”, se generarán todas las referencias, con una macro `\ref{}` de  $\LaTeX$  para cada una de ellas.

También podemos generar una sola macro de  $\LaTeX$  que contenga todas las etiquetas. Ello se hace pulsando “a”. En tal caso téngase en cuenta:

- Las etiquetas se colocarán una junto a la otra, sin escribir separador entre ellas. Sería como si todas se hubieran marcado con “m”.
- Esta macro no funcionaría en  $\LaTeX$  estándar. Requeriría la inclusión de algún paquete que permita realizar referencias múltiples.

## 7.2. Citas bibliográficas

Las citas bibliográficas constituyen un tipo de referencia especial realizadas con la macro `\cite{}`, en las que la *etiqueta* se localiza en una base de datos `BibTeX`, o en un elemento `\bibtitem` dentro del entorno `\thebibliography`.

### 7.2.1. Insertar citas bibliográficas

Las citas bibliográficas se insertan con “C-c [” («ref<sub>tex</sub>—citation»). Tras pulsarlo Ref<sub>TEX</sub> nos pedirá que introduzcamos una expresión regular con la que buscar la obra a la que queremos hacer referencia. Con esa expresión regular se hará una búsqueda, bien en una base de datos para Bib<sub>TEX</sub>, a la que se haya hecho referencia en el documento mediante la orden de L<sub>A</sub>T<sub>E</sub>X `\bibliography`, bien en el entorno `\thebibliography` del documento.

Finalmente, se mostrará un *buffer* con una lista de resultados, es decir: registros bibliográficos que coincidían con la expresión regular introducida.

La expresión regular usa una sintaxis extendida en la que `&&` representa una operación de Y lógico. Por ejemplo “Albaladejo&&Bercovitz” extraerá todas las entradas que mencionen a ambos autores.

La lista de resultados se muestra en un *buffer* especial muy parecido al utilizado en la selección de referencias normales (véase la sección 7.1.4). De hecho un porcentaje muy alto de acciones posibles se realiza con las mismas teclas y allí me remito (pág. 51).

Además de dichas teclas, podremos realizar las siguientes acciones:

- **[RET]** Inserta la macro `\cite` con la referencia seleccionada.
- **[m]** Marca la entrada actual para incluir varias referencias en una sola macro `\cite`.
- **[u]** Quita la marca (colocada mediante “m”) de la referencia.
- **[a]** Acepta todas las referencias marcadas y genera la macro `\cite` adecuada.
- **[A]** Acepta todas las referencias marcadas y crea una macro `\cite` para cada una de ellas.
- **[TAB]** Permite escribir, con auto-completado, el nombre de la etiqueta que se busca.

### 7.2.2. Otras características relacionadas con las citas bibliográficas

Ciertos paquetes amplían las capacidades de L<sub>A</sub>T<sub>E</sub>X en materia de citas bibliográficas. Ref<sub>TEX</sub> está preparado para trabajar con esos paquetes. Para los más corrientes (natbib, harvard, chicago) hay incluso una opción de menú que indica el estilo de citas que se empleará.

Alguno de estos paquetes incluyen diferentes macros para distintos estilos de citas, por lo que si son utilizados, «ref<sub>tex</sub>—citation» empezará por preguntar qué macro hay que utilizar.

*Para convertir alguno de esos paquetes en el estilo por defecto, hay que alterar el valor de la variable «`reftex—cite—format`» que por defecto tiene el valor de “`default`”.*

Por otra parte, cuando el cursor se detiene durante unos instantes en el nombre de una referencia bibliográfica citada mediante «`reftex—citation`», en el área de eco se mostrará cierta información sobre el contenido de dicha referencia bibliográfica.

## 7.3. Índices terminológicos

### 7.3.1. Insertar entradas de índice

La creación de un índice terminológico es tarea pesada que requiere una gran atención, pues hay que seguir convenciones muy estrictas para obtener un índice completo y consistente.  $\LaTeX$  implementa dos macros para marcar entradas en el índice: `\index` y `\glossary`. Algunos paquetes redefinen estas macros para otorgarles más versatilidad, al tiempo que añaden otras macros adicionales.

El índice se genera a partir de las entradas de índice existentes a lo largo del documento. Para insertar una nueva entrada de índice  $\text{Ref}\TeX$  proporciona dos mandatos distintos: el primero genera directamente el texto de la etiqueta; el segundo pregunta por ella:

**«`reftex—index—selection—or—word`»:** Este mandato, asociado a “`C-c /`”, como su propio nombre indica, genera automáticamente una entrada de índice (`\index`) usando como etiqueta la palabra sobre la que se encuentra el cursor, o el texto seleccionado en el momento en el que es llamado. Si cuando se llama a este mandato el cursor no está sobre ninguna palabra, automáticamente se llamará a «`reftex—index`», descrito más abajo.

*El mandato, por defecto, genera una entrada `\index`, aunque podemos configurarlo para que genere otra macro simplemente cambiando el valor de la variable «`reftex—index—default—macro`».*

Usando un prefijo numérico (“`C-u C-c /`”), antes de insertarse la etiqueta, se nos permitirá editarla.

**«`reftex—index`»:** Este mandato, asociado a “`C-c <`”, empieza por preguntar el tipo de entrada que queremos generar (de índice o de glosario) y luego nos solicita que insertemos la etiqueta para el índice. Para insertar la etiqueta podremos usar la función de auto-completado (basada en las etiquetas ya insertadas mediante este procedimiento).



Si estamos usando los paquetes `multind` o `index`, que admiten múltiples índices, este mandato nos preguntará también por el índice al que queremos asociar la entrada.

### 7.3.2. Generación automática de índices mediante un “*fichero de frases*”

RefTeX ofrece una utilidad mediante la cual podemos generar un fichero con las palabras o frases que hay que buscar en nuestro documento, asociando a cada una de ellas la entrada de índice que hay que insertar cuando se encuentre dicha palabra. Luego de forma automática se irá recorriendo nuestro documento y ante cada aparición de alguna de las frases almacenadas en tal fichero, se generará la macro  $\LaTeX$  adecuada.

En líneas generales el procedimiento funciona de la siguiente manera:

1. En primer lugar hay que crear un fichero en donde almacenar las etiquetas que usaremos y los índices a los que éstas se asocian. Para ello disponemos del mandato “`C-c |`” («`reftex—index—visit—phrases—buffer`»), que abre o crea (si es la primera vez que se ejecuta para cierto documento) un fichero de extensión `.rip`<sup>30</sup> específicamente preparado para la generación de índices globales; es decir: un *buffer* en el que se han definido los mandatos que a continuación se exponen.
2. En segundo lugar hay que ir indicando qué palabras hay que buscar en nuestro documento, y qué etiquetas se insertarán donde aparezcan tales palabras. Para ello, en nuestro documento fuente podemos ir colocando el cursor sobre las palabras significativas (o sobre las frases significativas previamente seleccionadas), y pulsar “`C-c \`” («`reftex—index—phrase—selection—or—word`»). Este mandato enviará al fichero de frases la palabra sobre la que se encuentre el cursor (o el texto seleccionado). Cada palabra a buscar sólo debe ser incluida en el fichero de frases una vez.
3. Tras ejecutar el mandato anterior, el *buffer* con el fichero de índices será cargado y se nos permitirá editar la palabra que se acaba de incorporar. Cuando terminemos, para guardar los cambios y volver a nuestro documento, hay que pulsar “`C-c C-c`” («`reftex—index—phrases—save—and—return`»).

Más adelante (en la letra chica) se dan los detalles sobre la *edición de las frases*. Aquí sólo diré que si queremos que donde se encuentre la frase seleccionada se genere una etiqueta *con esa frase* hay que dejarla tal cual. Si la

---

<sup>30</sup>iniciales de Reftex Index Phrases, no de *Requiescant In Pace* (= Descanse en paz). Me pregunto si el autor de RefTeX quería “acabar” ya con la aplicación cuando su subconciencia le dijo que la extensión para los ficheros de frases debía ser RIP; o tal vez los anglosajones no usen esa expresión e ignore el juego de palabras.

etiqueta debe ser distinta, hay que escribir el texto de la etiqueta, separándolo de la frase de búsqueda mediante un tabulador.

4. Disponemos de un conjunto de mandatos sobre el fichero índice dirigidos a poder comprobar posibles inconsistencias en el índice final. Estos mandatos incluyen la posibilidad de ordenar las frases (“C-c C-s” («reflex—index—sort—phrases»)), lo que se hará globalmente o por bloques si en el fichero se han ido dejando líneas totalmente en blanco que marquen los bloques; podemos obtener información sobre cómo quedará cierta entrada del índice así como una estimación aproximada del número de apariciones (“C-c C-i” («reflex—index—phrases—info»)), o comprobar si hay entradas dobles o superpuestas (“C-c C-t” («reflex—index—find—next—conflict—phrase»)).
5. Cuando estemos listos y el documento ultimado (o casi), podemos generar el índice: ello debe hacerse pulsando en el *buffer* con el índice de frases, “C-c C-x”, “C-c C-r” o “C-c C-a”, según queramos generar el índice exclusivamente para la frase sobre la que esté el cursor, para todas las frases seleccionadas, o para todo el *buffer* de frases.
6. Finalmente, una vez introducidas las etiquetas necesarias en sus lugares, podremos mostrar y editar el índice en un *buffer* independiente mediante el mandato “C-c >”. Este mandato admite diferentes prefijos numéricos que determinan la región o sector del documento a que debe concretarse la parte de índice editada. Así el prefijo 2 restringe el índice a la sección actual y el prefijo 3 lo restringe a la región.

El penúltimo paso hará que se vaya recorriendo nuestro documento y se vayan localizando todas las apariciones de las correspondientes *frases*, generando para cada una de ellas la entrada de índice especificada en el fichero de frases. Aunque las entradas de índice no se generan de modo totalmente automático, sino que ante cada coincidencia el programa se detendrá y nos preguntará qué hacer. Podremos entonces contestarle alguna de las siguientes cosas:

- [y] Reemplaza esta coincidencia con la cadena propuesta.
- [n] Se salta esta coincidencia.
- [!] Reemplaza la actual y todas las restantes de este fichero.
- [q] Se salta esta coincidencia y empieza con el próximo fichero.
- [Q] Se salta esta coincidencia y empieza con la próxima frase (del fichero de frases).
- [o] Selecciona para esta coincidencia una macro diferente.
- [1-9] Selecciona uno de los múltiples índices posibles.
- [e] Edita el texto de reemplazo.
- [C-r] Inicia una edición recursiva. Para volver a este proceso, cuando terminemos la edición, hay que pulsar “C-M-c”.
- [s] Guarda el *buffer* y vuelve a preguntar por esta coincidencia.

- [S] Guarda todos los *buffers* activos y vuelve a preguntar por esta coincidencia.
- [C-g] Aborta la generación del índice.

En cuanto al último paso, en la ventana de edición del índice podemos usar un grupo de teclas que se mostrarán en pantalla pulsando “?”.

*Los detalles para este proceso son muchos y por ello no me puedo extender demasiado.*

*El fichero de frases debe tener un encabezamiento y una lista de frases. El encabezamiento ha de contener una línea para cada posible macro que se utilizará. Así, por ejemplo, el encabezamiento de un fichero de frases en el que sólo se usará como macro de índice la orden de  $\LaTeX$  `\index`, sería así:*

```
% -*- mode: reftex-index-phrases; TeX-master: "guia-atx.tex" -*-
%                               Key      Macro Format      Repeat
%-----
>>>>>INDEX_MACRO_DEFINITION:      i      \index{%s}          t
%-----
```

*Obsérvese que las tres primeras líneas vienen marcadas como comentarios por lo que el encabezamiento se inicia en la cuarta línea la cual empieza diciendo que se trata de una definición de macro de índice. A continuación se contiene una letra, que se usará en el resto del fichero para identificar a esta macro, lo que es necesario porque este buffer permite usar varias macros distintas, y hace falta algún carácter que identifique de manera exclusiva a cada una de ellas. Luego se recoge la orden  $\LaTeX$  que hay que escribir en el documento, aunque “%s” representa la etiqueta concreta que se aplicará. Finalmente una “t” en el campo “repeat” indica que el texto original debe quedar como estaba, es decir: que la palabra clave debe repetirse o volverse a escribir.*

*Así en el ejemplo siguiente:*

```
Mercurio es el planeta más próximo al sol
Mercurio es el \index{planeta} más próximo al sol
Mercurio es el \index{planeta}planeta más próximo al sol
```

*La primera línea recoge lo que podría ser el texto original de nuestro documento. La segunda línea indica cómo quedaría después de escribir la macro “index” convirtiendo la palabra “planeta” en etiqueta y, por lo tanto, haciéndola desaparecer del texto. La tercera línea indica cómo quedaría, usando la palabra “planeta” como etiqueta, y volviéndola a escribir, es decir: repitiéndola, para evitar que desaparezca del texto.*

*En un mismo fichero de frases podemos incorporar varias macros generadoras de índices. Por ejemplo: para escribir un índice terminológico y un glosario, el encabezamiento podría ser:*

```
>>>>>INDEX_MACRO_DEFINITION: i \index{%s} t
>>>>>INDEX_MACRO_DEFINITION: g \glossary{%s} t
```

En cuanto a las frases recogidas en el fichero, para cada una de ellas hay que indicar:

- Qué macro se generará cuando sea localizada (`\index`, `\glossary`, personalizada, incluida en un paquete...). Esto se indica mediante la letra asignada como clave a cada macro en el encabezamiento del fichero. Si no se indica ninguna clave, se usará la macro correspondiente a la primera línea del encabezamiento.
- Qué texto hay que buscar en el documento. Este es el único campo obligatorio. Se separa del campo anterior y del siguiente por tabuladores.
- Etiqueta a utilizar en la macro que se generará cada vez que se encuentre el texto a buscar. Si queremos que la macro se limite a usar exactamente el mismo texto localizado, no hay que indicar nada. Pero en caso contrario, aquí debemos indicar la o las etiquetas que se usarán para generar la macro.

El formato de cada entrada por lo tanto es el siguiente:

```
[Macro] <TAB> Buscar [<TAB> Etiquetas]
```

donde, Etiqueta puede a su vez asumir cualquiera de las tres siguientes formas:

```
Buscar <Tab> Etiqueta
```

```
Buscar <Tab> Etiqueta1 && Etiqueta2 && ... Etiqueta N
```

```
Buscar <Tab> Etiqueta1 || Etiqueta2 || ... Etiqueta N
```

La diferencia entre estas tres líneas es la siguiente:

- En el primer caso, se usará como etiqueta el texto separado del texto a buscar por un tabulador.
- En el segundo caso, los caracteres “&&” indican que hay que escribir tantas macros como etiquetas haya, usando en cada una de ellas una etiqueta diferente.
- En el tercer caso, los caracteres “||” indican que sólo queremos escribir una macro, pero esa macro variará según los casos, de modo que cada vez que aparezca el texto buscado, elegiremos entre las posibilidades separadas con “||”.

Por ejemplo: en la confección del índice de este documento, me encuentro con que hay un mandato de AUC T<sub>E</sub>X llamado «`TeX-insert-braces`» y también hay una opción de usuario con el mismo nombre. En el índice que pienso generar distingo entre mandatos y opciones, por lo tanto, cada vez que aparezca el nombre de ese mandato y opción, tendré que decidir una entrada de índice entre dos posibles. Para ello tendría que escribir en el fichero de frases lo siguiente:

```
TeX-insert-braces TeX-insert-braces~(mandato)||TeX-insert-braces~(opción)
```

*La posibilidad de insertar varias etiquetas, o una a elegir de entre varias, es útil cuando se piensan generar varios índices: podemos tener todos los textos de búsqueda en un solo fichero de frases, y ahí indicar cada texto de búsqueda en qué índices generará una entrada.*

*De otro lado, hay un grupo de variables que afectan de diferentes maneras al proceso. Podemos ver cuáles son y leer la utilidad y valores de cada una de ellas, ejecutando el mandato “M-x customize-option reftex-index-support”, que abrirá un buffer de personalización con este grupo de variables.*

### 7.3.3. La ventana de índice de RefTeX

Una vez escritas las etiquetas para la generación de índice, RefTeX puede leerlas y en una ventana específica mostrar una aproximación a cómo quedará el índice, permitiendo realizar determinados cambios. Para ello hay que pulsar el mandato “C-c >” («reftex—display—index»)

Las acciones que podemos realizar en la ventana de índice son las siguientes:

#### Teclas de utilidad general

- **[?]** («reftex—index—show—help») Muestra un sumario con las opciones disponibles.
- **[0-9]** Argumento prefijo. Se aplica a la próxima orden. Sirve, por ejemplo, para desplazarnos varias líneas de golpe.

#### Movimiento por el índice

- **[n, C-n, Flecha abajo]** («reftex—index—next») Mueve el cursor a la próxima línea.
- **[p, C-p, Flecha arriba]** («reftex—index—previous») Mueve el cursor a la línea anterior.
- **[AvPag, RePag]** Mueve el cursor una pantalla hacia delante o hacia detrás.
- **[! A...Z]** Salta a la sección del índice correspondiente a la letra pulsada. “!” salta a la sección de símbolos.

#### Acceso al documento principal

- **[SPC]** («reftex—index—view—entry») Muestra en la otra ventana el lugar donde se encuentra la entrada de índice actual.

- **[f]** («reftex—index—toggle—follow») Activa / Desactiva el modo de seguimiento. En él la otra ventana de la pantalla va mostrando la localización de las entradas de índice conforme nos vamos moviendo por él; es como si siempre pulsáramos SPC.
- **[RET]** («reftex—index—goto—entry—and—hide») Va a la entrada seleccionada y oculta la ventana de índice.
- **[TAB]** («reftex—index—goto—entry») Activa la ventana del documento principal que saltará al lugar correspondiente a la entrada de índice seleccionada.

### Edición de las entradas del índice

- **[e]** («reftex—index—edit») Edita la macro  $\text{L}^{\text{T}}\text{E}^{\text{X}}$  generadora de la entrada de índice actual.
- **[\*]** («reftex—index—edit—key») Edita la etiqueta de la entrada de índice actual.
- **[|]** («reftex—index—edit—attribute») Edita la parte de atributo de la entrada de índice actual. Funciona en entradas escritas con el formato  $\backslash\text{index}\{\text{Cadena1}|\text{see}\{\text{Cadena2}\}\}$  .. Con prefijo numérico (“C-u”) esta orden borra la parte de la entrada posterior a “|”.
- **[@]** («reftex—index—edit—visual») Edita la parte visual de la entrada de índice, es decir: la parte posterior a “@” en entradas de índice escritas con el formato  $\backslash\text{index}\{\text{Cadena1}@\text{Cadena2}\}$ . Llamada con “C-u” se eliminará toda esa parte de la entrada.
- **[{]** («reftex—index—toggle—range—beginning») Cambia el principio de la propiedad de rango de páginas (parte “{” de la macro).
- **[}]** («reftex—index—toggle—range—end») Cambia el final del rango de páginas (parte “}” de la macro).
- **[\_]** («reftex—index—level—down») Convierte la entrada actual en una subentrada. Se nos preguntará por el nombre de la entrada de la que va a depender.
- **[^]** («reftex—index—level—up») En una subentrada, elimina la parte más a la izquierda, elevándola de nivel en la jerarquía: si, por ejemplo, la entrada era “aaa!bbb!ccc”, pasará a ser “bbb!ccc”.

### Control de lo que se visualiza

- **[c]** («reftex—index—toggle—context») Activa / Desactiva la visualización de una línea para ayuda a la contextualización.
- **[s]** («reftex—index—switch—index—tag») Cambia a un índice diferente. Tras pulsar esta opción, se nos volverá a preguntar por el tipo de índice que queremos editar.

- `[{}` («») Restringe el índice a una sección del documento. El número concreto de sección se mostrará en la línea de modo (indicador `R<>`) y en la cabecera del *buffer*.
- `[}]` («») Amplía el índice —si estaba restringido— para mostrar todo el documento.
- `[<]` («») Si el índice está restringido, hace que se muestre el índice de la sección previa.
- `[>]` («») Si el índice está restringido, hace que se muestre el índice de la sección anterior.

### Actualización del *buffer*

- `[g]` («`revert—buffer`») Actualiza el índice, desde la lista interna de entradas mantenida en memoria por RefTeX, pero no vuelve a analizar el documento.
- `[r, R]` («`reftex—index—rescan`») Analiza de nuevo el documento para actualizar y reconstruir el índice. Si la variable «`reftex—enable—partial—scans`» es distinta de “nil”, y el documento consta de varios ficheros, releerá sólo la parte correspondiente al fichero actual. Para releer todo el documento hay que pulsar “`C-u r`”.

### Terminación

- `[q]` («`reftex—index—quit`») Cierra el *buffer* con el índice y anula la ejecución del mandato.
- `[k]` («`reftex—index—quit—and—kill`») Mata al índice, es decir: no sólo cierra la ventana que lo contiene, sino que lo elimina de la memoria (hasta que vuelva a ser llamado).

## 7.4. Ver las referencias cruzadas

Hasta ahora hemos visto cómo generar referencias cruzadas de distintos tipos. En todas ellas dos partes del documento aparecen vinculadas. Pues bien RefTeX nos permite, cuando tenemos el cursor sobre la referencia, ver en ventana separada la parte del documento donde se encuentra la etiqueta a la que apunta la referencia.

Esta utilidad es invocada presionando “`C-c &`” («`reftex—view—crossref`») cuando el cursor se encuentra sobre el nombre de la etiqueta de las siguientes sentencias:

- `\ref`, `\pageref`: muestra el lugar donde se define la etiqueta.
- `\label`: la primera vez que se pulse mostrará la primera referencia a la etiqueta. Pulsando sucesivas veces se irán viendo más referencias a la etiqueta.

- `\cite`: muestra, bien la entrada `\bibitem` a que se hace referencia, bien el registro del fichero en formato Bib $\TeX$ .
- `\bibitem`: muestra la localización de una cita a esta entrada bibliográfica. Pulsando sucesivas veces iremos viendo sucesivas citas.
- `\index`: va mostrando otros lugares del documento en donde haya una entrada de índice para la misma etiqueta.

Este mandato funciona también en bases de datos bibliográficas que se estén editando en Emacs con el modo mayor “`bibtex`”<sup>31</sup>. En este caso la primera vez que se ejecute el mandato (en la sesión de trabajo actual) se nos preguntará por el documento dentro del cual queremos localizar referencias.

Por otra parte, sin necesidad de invocar expresamente a ningún mandato, estando cargado Ref $\TeX$ , cada vez que el cursor se detenga sobre una referencia o cita bibliográfica durante más tiempo del indicado en la variable «`refTeX—idle—time`», en el área de eco de Emacs (siempre que no esté ocupada con algún mensaje), podremos ver cierta información sobre la etiqueta referenciada.

## 8. Trabajo con varios documentos

### 8.1. Documentos maestros

Las macros de  $\LaTeX$  `\include{}` e `\input{}` nos permiten manejar varios ficheros de texto y luego compilarlos juntos. Hay además algunos casos en los que eso es especialmente recomendable, como cuando un documento está siendo confeccionado entre varias personas, o cuando el documento resulta extremadamente largo, lo que conlleva inconvenientes de memoria necesaria para su edición, posible pérdida del coloreado de la sintaxis (véase la sección 2.4, pág. 18), y mayor dificultad para desplazarnos dentro del documento y navegar por él (aunque esto último se puede *atenuar* mediante las utilidades examinadas en las secciones 5.4 y 5.5).

Para estos casos en Emacs podemos definir a uno de los documentos como *maestro* e indicar en los otros documentos, dependientes de él, cuál es el documento maestro del que dependen. Esto a su vez nos permitirá, desde cualquiera de los documentos que componen el proyecto:

- Abrir el documento maestro.

---

<sup>31</sup>Aunque debe tenerse en cuenta que cada vez que carguemos en Emacs un fichero tipo `bibtex`, deberemos activar a mano el modo menor “`refTeX`”.



- Compilar la totalidad del proyecto, o ejecutar cualquier otro mandato externo sobre el documento maestro y no sobre el documento que se esté editando.
- Guardar con una sola orden todos los documentos relacionados con el mismo documento maestro que se estén editando simultáneamente.

En un proyecto multidocumental, debe definirse como documento maestro a aquel que contenga dentro de sí, directa o indirectamente a todos los demás, es decir: a aquel cuya compilación implique necesariamente la compilación de la totalidad de los documentos parciales.

Lo que se suele hacer en estos casos es generar un documento que contenga el encabezado, las órdenes `\begin{document}` y `\end{document}`, y una serie de sentencias “input” o “include” referidas a los ficheros que contengan cada una de las diferentes unidades estructurales que componen nuestro documento. Ese es el contenido típico de los documentos maestros.

Por lo demás, una vez decidido cuál será el documento maestro, lo único que hay que hacer es, en los demás ficheros `.tex` del proyecto indicar el nombre del documento maestro, para lo cual hay que asignar un valor a la variable de fichero «TeX—master».

Si se desea una explicación detallada de este tipo de variables, consúltese lo dicho en la página 84. Aquí solo añadiré que para definir un documento como maestro basta con incluir en la parte final de todos los documentos implicados las siguientes líneas:

```
%%% Local Variables:  
%%% TeX-master: valor  
%%% End:
```

cosa que, por otra parte, hace Emacs por defecto cada vez que se abre un documento por primera vez en el modo mayor LaTeX.

El término *Valor* representa (en las líneas que acabo de decir que hay que añadir al final del documento) el nombre del documento maestro. Al respecto téngase en cuenta lo siguiente:

- En el propio documento maestro, o en los documentos que no están fraccionados, podemos no indicar ningún valor, o indicar el valor “t” (sin las comillas), o indicar el valor “nil” sin las comillas.
- En los documentos que dependan del documento maestro, hay que escribir el nombre de éste entre comillas verdaderas (no los caracteres que T<sub>E</sub>X utiliza

para representarlas). No es preciso poner la extensión `.tex` ni el directorio si el documento maestro y el dependiente están en el mismo directorio.

- El nombre del documento (o el del directorio) no puede contener ni espacios en blanco ni caracteres no anglosajones.

Esta última observación es importante: el compilador `TeX` no soporta en los nombres de ficheros espacios en blanco o caracteres no anglosajones (eñes o vocales acentuadas). Ahora bien: si estos caracteres están en la ruta de acceso pero no en el nombre del fichero propiamente dicho, entonces la compilación no tiene por qué generar errores si el compilador es llamado *desde dentro* del directorio en cuyo nombre existen caracteres no admitidos.

Pero Emacs pasa al compilador, como parámetro, el nombre de fichero que se haya incluido en esta variable; en consecuencia, si en ella figura la ruta de acceso y en dicha ruta hay caracteres no acentuados, se provocará un error que no se habría provocado en el caso de haber compilado manualmente (desde la shell). La forma de evitar dicho error es no escribir, como valor de «`TeX—master`» la ruta de acceso si esta contiene caracteres de los no soportados por `TeX`<sup>32</sup>.

*El nombre del documento maestro viene controlado por la variable «`TeX—master`». Por defecto Emacs pregunta por el valor de la variable cuando se abre un documento nuevo y la guarda automáticamente como variable de fichero. Pero podemos establecerla como valor fijo, lo cual sólo tiene sentido si en todos nuestros proyectos el documento maestro tiene siempre el mismo nombre.*

*Por ejemplo, si queremos establecer, como nombre fijo del documento maestro, el de “principal”, deberíamos escribir en nuestro fichero `.emacs` la siguiente sentencia:*

```
(setq-default TeX-master "principal")
```

*Pero eso nos obligaría a dar el nombre de “principal.tex” al documento maestro de todos nuestros proyectos.*

*En el caso de que «`TeX—master`» tenga el valor “t” Emacs asume que el documento actual es el documento maestro, y en el caso de que tenga el valor “nil” siempre que el documento sea abierto se preguntará por su documento maestro.*

*La variable «`TeX—one—master`» contiene una expresión regular que se utiliza para asignar automáticamente una variable de fichero «`TeX—master`» a todos los ficheros cuyo nombre coincida con dicha expresión regular.*

*Emacs mantiene información sobre los macros, entornos etiquetas y estilos utilizados en un determinado documento. Para hacer esto cuando se trabaja con varios documentos, Emacs genera un directorio llamado “auto” en el que guarda la información necesaria (véase la sección 2.5, pág. 19).*

---

<sup>32</sup>Si, por error, hemos permitido que Emacs automáticamente genere el nombre del fichero con caracteres acentuados en la ruta de acceso, podemos reescribir el valor de la variable, y ejecutar el mandato «`TeX—normal—mode`» (“`C-c #`”) para obligar a Emacs a releer el valor de la variable

## 8.2. Mandatos para trabajar con varios documentos cuyo documento maestro es común.

Emacs permite en todas las órdenes externas dirigidas a ejecutar el compilador T<sub>E</sub>X o alguna de sus herramientas relacionadas, indicar si queremos ejecutar la orden sobre el documento que se está editando actualmente o sobre el documento maestro del que este depende. La forma de hacer eso se verá cuando se hable de los mandatos externos ejecutables desde Emacs (sección 10.2, pág. 79).

Además, se proporcionan los siguientes dos mandatos utilizables cuando se está trabajando con varios documentos relacionados entre sí mediante un documento maestro común:

Teclado	Mandato	Acción
C-c ^	TeX-home-buffer	Abre el documento maestro.
C-c C-d	TeX-save-document	Guarda en disco todos los documentos asociados que hayan sido modificados.

En realidad la segunda orden no es extremadamente útil, puesto que su función la cumple también la orden estándar de Emacs “C-x s”. En todo caso, cuando se trabaja con varios documentos hay que tener en cuenta que las utilidades externas a Emacs siempre trabajan sobre el contenido de los ficheros y no sobre el contenido del *buffer* de edición de Emacs. Por ello cuando una orden de estas es ejecutada, Emacs comprueba si ha habido modificaciones en el *buffer* desde el que dicha orden es llamada, para grabarlo en disco; pero no extiende tal comprobación al resto de *buffers*, con lo que pueden producirse errores o compilaciones inadecuadas, al no actualizarse a disco todos los ficheros implicados. Para evitar esto es para lo que se ha implementado en Emacs el mandato “C-c C-d”, aunque, como ya he dicho, en su lugar podemos utilizar el estándar “C-x s”.

*La variable «TeX—save—query» controla si antes de guardar los cambios de un determinado fichero debe preguntarse al usuario si quiere hacerlo. Los cambios se guardarán automáticamente —sin preguntar— si esta variable tiene un valor de “nil”.*

## 9. Trabajo con ficheros BibTeX

Las bases de datos de BibTeX son ficheros de texto en los que cada entrada se somete a unas rígidas reglas de sintaxis de tal manera que posteriormente el programa BiBTeX podrá extraer de la base de datos la información necesaria para confeccionar la bibliografía de un documento L<sup>A</sup>T<sub>E</sub>X.

Se trata de ficheros de texto, y por lo tanto pueden ser manejados por cualquier editor de textos, aunque en general las aplicaciones específicas orientadas a

bases de datos los manejan mejor. Personalmente me gusta el manejo que de estos ficheros se consigue con la aplicación gráfica gBib (diseñada para el entorno de escritorio GNome).

Pero entre las aplicaciones que gestionan estos ficheros *como ficheros de texto* sin duda que Emacs es de las que mejor lo hace, a pesar de las limitaciones que este tipo de manejo supone en bases de datos amplias.

Desde su versión 20.4 Emacs incorpora un modo mayor denominado “bibtex” que se activa automáticamente al cargar un fichero de extensión bib, y que nos permite:

- Insertar con facilidad diferentes tipos de registros.
- Mantener ordenada la base de datos atendiendo a la etiqueta asignada a cada registro.
- Asegurarse en el momento de la introducción de los registros de que no se asigna la misma etiqueta a dos registros distintos.
- Comprobar que no se deje de rellenar ningún campo obligatorio en los registros.

A lo que hay que añadir que el paquete RefTeX también trabaja con ficheros bib, por lo que permite buscar en ellos la referencia de una cita y, viveversa, buscar en un fichero tex, las citas concretas que se hacen de una entrada en el fichero bib (véanse las secciones [7.2.1](#) y [7.4](#)).

### 9.1. Inserción de registros

En las bases de BibTeX existen distintos tipos de registros, y cada uno de ellos requiere un conjunto de campos obligatorios y admite otro conjunto de campos opcionales.

Los mandatos de Emacs para la inserción de registros conocen los distintos tipos de registros y saben qué campos requiere cada uno de ellos. Ejecutando el mandato se insertará un registro vacío que contendrá una especie de plantilla del tipo de registro solicitado con el nombre de los campos ya escrito y preparada para simplemente escribir el valor asignado a cada uno de ellos.

El nuevo registro, por otra parte se insertará:

- En el lugar donde se encuentre el cursor, si este se encuentra entre dos registros.
- Detrás del registro actual, si en el momento de ejecutar el mandato de inserción, el cursor estaba *dentro* de un registro.

- En el lugar que alfabéticamente le corresponda si está activada la opción de mantenimiento automático del orden alfabético.

Existen dos procedimientos para insertar un nuevo registro. El primero consiste en ejecutar el mandato “C-c C-b” («bibtex—entry») que nos preguntará en el *minibuffer* por el tipo de registro deseado, pudiendo aquí utilizar las funciones de auto-completado propias de Emacs para los tipos de registro estándar de BibTeX: Article, Book, Booklet, InBook, InCollection, InProceedings, Manual, MasterThesis, Misc, PhdThesis, Proceedings, TechReport y Unpublished.

El segundo procedimiento consiste en ejecutar alguno de los mandatos existentes para insertar tipos de registros específicos. Así disponemos de los siguientes mandatos:

Teclado	Mandato	Tipo de registro
C-c C-e C-a	bibtex-Article	Artículo
C-c C-e C-b	bibtex-InBook	En libro
C-c C-e b	bibtex-Book	Libro (con editorial conocida)
C-c C-e B	bibtex-Booklet	Libro (sin editora conocida)
C-c C-e C-c	bibtex-InCollection	Parte autónoma de un libro
C-c C-e C-m	bibtex-Manual	Documentación técnica
C-c C-e m	bibtex-MasterThesis	Tesina de licenciatura
C-c C-e M	bibtex-Misc	Cualquiera que no encaje en las demás
C-c C-e C-p	bibtex-InProceedings	Artículo en congreso
C-c C-e p	bibtex-Proceedings	Actas de un congreso
C-c C-e P	bibtex-PhdThesis	Tesis doctoral
C-c C-e M-p	bibtex-Preamble	Preámbulo del fichero
C-c C-e C-s	bibtex-String	Cadena de abreviatura
C-c C-e C-t	bibtex-TechReport	Informe técnico
C-c C-e C-u	bibtex-Unpublished	No publicado

Como se observará las combinaciones de teclas son fáciles de recordar, porque todas comparten el prefijo “C-c C-e” y a continuación llevan una tercera tecla que consiste siempre en la inicial del tipo de referencia a insertar. Pero como en algunos casos tipos distintos de referencias comparten la misma inicial, se adopta la siguiente convención:

- Con la tecla Control y en minúsculas.
- Pulsada sola y en minúsculas.
- Pulsada sola y en mayúsculas.
- Con la tecla Meta y en minúsculas.

El elegir una u otra posibilidad depende de que se trate de un tipo de referencia más o menos usado (según los diseñadores del paquete). La primera posibilidad (tecla Control y minúsculas) se usa para las referencias más habitualmente usadas, y la última (Meta y minúsculas) para las más raramente usadas.

Tras insertar el nuevo registro, Emacs ejecutará la función «*bibtex—add—entry—hook*», la cual puede haber sido redefinida en nuestro fichero *.emacs*.

De otro lado, el contenido de cada tipo de registro viene determinado por «*bibtex—entry—field—alist*». Se trata de una lista asociativa de Emacs, para cuya explicación puede consultarse la ayuda HTML del paquete BibTeX de Emacs, al tiempo que otras variables controlan otros aspectos del registro que se insertará, y así:

«*bibtex—field—delimiters*» Determina que delimitadores se usarán para los campos. Puede ser braces o double-quotes. Por defecto vale braces, es decir: llaves.

«*bibtex—entry—delimiters*» Determina los delimitadores del registro propiamente dicho. Puede valer braces o parentheses. Por defecto vale braces.

«*bibtex—comma—after—last—field*» Determina si después del último campo se incluirá o no una coma.

Cuando está activada la opción «*global—font—lock—mode*» y Emacs se ejecuta en un entorno gráfico, en los registros recién insertados, los campos se muestran en dos colores diferentes: en la parte superior están los campos obligatorios; en la parte inferior, y en otro color, los campos optativos. Asimismo algunos nombres de los campos obligatorios van precedidos del prefijo “ALT” y todos los nombres de los campos opcionales van precedidos del prefijo “OPT”.

Estos prefijos sirven para recordarnos si un campo es *alternativo* u *opcional*. El campo alternativo implica que es *obligatorio* rellenar uno de los dos, pero no los dos. Los campos opcionales pueden o no rellenarse.

## 9.2. Edición de los registros

Las acciones de edición para las que existe algún mandato especial son las siguientes:

### 9.2.1. Movimiento del cursor

Por supuesto que podemos usar el sistema normal de movimiento del cursor en el fichero BibTeX; pero también podemos utilizar los mandatos específicos proporcionados por el paquete “*bibtex*”. Estos son los siguientes:

Teclado	Nombre	Acción
M-C-a	<i>bibtex—beginning—of—entry</i>	Va al principio del registro,
M-C-e	<i>bibtex—end—of—entry</i>	Va al final del registro.
TAB	<i>bibtex—find—text</i>	Va al final del campo actual.
C-j	<i>bibtex—next—fill</i>	Va al próximo campo.

El mandato TAB precedido de un prefijo numérico muestra en la línea de eco un pequeño mensaje con información sobre la utilidad del campo en el que esté situado el cursor; efecto que también se obtiene con el mandato “C-c ?”.

Además en un fichero BibTeX se asume que cada registro constituye un párrafo. Por lo tanto “C-Abajo” y “C-Arriba” que saltan, respectivamente, al párrafo siguiente o al anterior, nos permitirán desplazarnos de registro en registro.

### 9.2.2. Añadir, borrar y modificar campos

Las tareas que en este punto están asociadas a mandatos son las siguientes:

Teclado	Nombre	Acción
C-c "	bibtex-remove-delimiters	Elimina las llaves de delimitación del campo
C-c C-d	bibtex-empty-field	Borra el contenido del campo
C-c C-f	bibtex-make-field	Inserta un nuevo campo
C-c C-q	bibtex-text-indentation	Reformatear registro
C-c C-o	bibtex-remove-OPT-or-ALT	Elimina, del nombre de los campos, los prefijos OPT y ALT

Los campos están normalmente delimitados por llaves; pero en algunos casos es preciso eliminar esos delimitadores (cuando el contenido del campo esté asociado a una *abreviatura* o “string”), y en otros casos el uso de los delimitadores es voluntario (campos cuyo contenido sean números). Para eliminar los delimitadores funcionan, además de las teclas indicadas en la tabla superior, “C-c }” y “C-c {”.

En cuanto a los prefijos, su función es simplemente recordarnos de qué tipo es cada campo. Pero deben ser eliminados antes de procesar el fichero con el programa *BibTeX*, ya que en caso contrario no se identificará el nombre correcto de los campos. Para eliminar los prefijos disponemos de “C-c C-o”, aunque los prefijos también desaparecen cuando se ordena que el registro sea *limpiado* (véase 9.2.4). Este último es el mejor procedimiento ya que si nos acostumbramos a hacerlo así, la presencia de los prefijos servirá para recordarnos que el registro aún no ha sido *limpiado*.

### 9.2.3. Eliminar y recuperar campos y registros

Las operaciones normales de eliminar (“C-k”, “C-w”), copiar (“M-w”), y recuperar (“C-y”, “M-y”) datos, pueden precederse del prefijo “C-c”, en cuyo caso se asume que cada campo es una línea y los registros son el equivalente a la región. Es decir: “C-c C-w” eliminará todo el registro, mientras que “C-c C-k” eliminará el

campo actual. En concreto los mandatos relativos a la eliminación y recuperación<sup>33</sup> son los siguientes:

Teclado	Nombre	Acción
M-C-h	bibtex-mark-entry	Selecciona todo el registro
C-c C-w	bibtex-kill-entry	Elimina todo el registro (esté o no seleccionado)
C-c M-w	bibtex-copy-entry-as-kill	Copia en memoria el registro como si hubiera sido eliminado
C-c C-k	bibtex-kill-field	Eliminar todo el campo
C-c M-k	bibtex-copy-field-as-kill	Copia en memoria el campo como si hubiera sido eliminado
C-c C-y	bibtex-yank	Recuperar última eliminación
C-c M-y	bibtex-yank-pop	Recuperar eliminación anterior

*Estas operaciones de eliminación y recuperación, actúan sobre sus propios bloques de memoria, los cuales no interfieren con los bloques de memoria asociados a las operaciones ordinarias de eliminación y recuperación. Es decir: si eliminamos algo con uno de estos mandatos, hay que recuperarlo exclusivamente con estos mandatos; los mandatos ordinarios de recuperación (“C-y” y “M-y”) no pueden recuperar bloques de texto eliminados con estos mandatos.*

*La variable «bibtex—field—kill—ring—max» determina el número de eliminaciones que podemos acumular en memoria. Por defecto vale 20.*

#### 9.2.4. Limpiar el registro

El mandato “C-c C-c” («bibtex—clean—entry») realiza varias acciones sobre el registro en donde sea ejecutado y así:

- Elimina los prefijos OPT y ALT.
- Elimina los campos optativos o alternos no utilizados.
- Elimina los delimitadores en los campos cuyo valor sea puramente numérico.
- Si no se ha asignado ninguna etiqueta identificativa al registro, o el mandato ha sido llamado con un prefijo numérico, se genera una etiqueta y se pide al usuario que la acepte o modifique. Véase la sección 9.3.3.
- Si está activado el mantenimiento del orden alfabético (véase la sección 9.3.2), y la etiqueta que determina el orden ha sido modificada, coloca el registro en la posición que le corresponda de acuerdo con el nuevo contenido de la etiqueta.

No debe pues ser utilizado hasta que se haya escrito el valor de los campos y el de la etiqueta identificativa del registro. Además, «bibtex—clean—entry» realiza

<sup>33</sup>Para usuarios poco acostumbrados a la terminología de Emacs, en ella *eliminar* equivale a lo que en otras aplicaciones se denomina *cortar*, es decir: borrar el texto seleccionado, pero enviándolo a una zona de la memoria desde la que se pueda recuperar con facilidad. A la operación de recuperar el texto previamente eliminado se le denomina, en otras aplicaciones, *pegar*.



una comprobación sintáctica del registro antes de proceder a su limpieza, y sólo actúa en el caso de que todo esté bien. Se entiende que no está todo bien si en la etiqueta identificativa se han incluido caracteres no anglosajones o espacios en blanco, o si se ha dejado de rellenar algún campo obligatorio.

La acción exacta de «*bibtex—clean—entry*» depende del contenido de la variable «*bibtex—entry—format*». Además después de ejecutarse este mandato llama a las tareas definidas en «*bibtex—clean—entry—hook*».

«*bibtex—entry—format*» consiste en una lista de aspectos que se pueden ajustar en un campo. Estos aspectos son los siguientes:

**opts-or-alts:** Determina si se deben o no eliminar los prefijos *OPT* y *ALT* del nombre de los campos. Por defecto está activada.

**numerical-fields:** Determina si se deben eliminar o no los delimitadores en los campos de contenido numérico. Por defecto está activada.

**page-dashes:** Determina si en los campos que contienen números de páginas deben o no sustituirse los dobles guiones que indican un rango, por guiones simples. Por defecto esta opción está desactivada.

**interhit-booktitle:** Si esta opción está activada (que por defecto no lo está) se intentará llenar los campos título del libro vacíos con el contenido del campo título del registro determinado por el contenido del campo *crossref*. Esta utilidad sólo funciona si está activado el mantenimiento automático del orden alfabético (véase la sección 9.3.2).

**realign:** Determina si al limpiar el registro debe o no reformatearse, es decir: recalcular el lugar en donde deben ir los saltos de línea. Por defecto está desactivada, aunque me parece una buena candidata a ser activada.

**last-comma:** Determina si debe comprobarse la existencia de comas después del último campo. En tal caso las comas se añadirán o se suprimirán dependiendo del valor de la variable «*bibtex—comma—after—last—field*». Por defecto está desactivada.

**delimiters:** Determina si hay o no que cambiar los delimitadores para ajustarlos a los previstos por defecto; es decir: si para un campo hemos cambiado los delimitadores por defecto, ¿hay que reajustarlos? La opción en principio está desactivada.

**unify-case:** Determina si hay que reajustar el nombre de los campos para que estén en mayúsculas o minúsculas exactamente tal y como aparecen definidos en «*bibtex—entry—field—alist*».

Sobre cuestiones relacionadas con las tres últimas opciones se puede encontrar más información en la sección 9.1.

### 9.3. Edición avanzada

#### 9.3.1. Análisis sintáctico de los registros

El mandato «`bibtex—validate`» comprueba si hay o no errores sintácticos en los registros. Se entiende por error sintáctico, la presencia de algún carácter no autorizado en la etiqueta (espacios en blanco, eñes y vocales acentuadas), o la ausencia de algún delimitador cuando sea preciso (llaves, comillas, etc). Si el mandato es llamado con un prefijo numérico también se comprueba la presencia en los registros de todos los campos obligatorios, así como si ciertos campos contienen un mes “*cuestionable*”.

*En principio los meses se pueden insertar mediante abreviaturas de sus nombres en inglés (definidas en la variable «`bibtex—predefined—month—strings`»). Pues bien: se considera que el nombre de un mes es “sospechoso” cuando su contenido forma parte de dicha variable, pero el texto ha sido encerrado entre comillas o delimitadores.*

*¿Por qué es sospechoso ese nombre? Porque las abreviaturas usadas coinciden con las conocidas por el programa BibTeX, cuyos estilos son los que determinan cómo escribir el nombre de los meses. Pero si el contenido del campo va entrecomillado o entre llaves, BiBTeX lo tratará como un literal, y será incapaz de manipularlo.*

*Por ejemplo: si nuestro estilo de bibliografía define como idioma el español, el nombre de los meses aparecerá en español, siempre que en la base de datos se hayan escrito con las abreviaturas conocidas por BiBTeX, o como cadenas literales (en español).*

*De otro lado si modificamos la variable «`bibtex—predefined—month—strings`» (cosa que no se recomienda por lo que se acaba de decir), hay que modificar también la variable «`bibtex—predefined—strings`» que contiene todas las cadenas que Emacs tratará como abreviaturas que no requieren ir entrecomilladas.*

#### 9.3.2. Orden alfabético de los registros

En cualquier momento podemos ejecutar el mandato «`bibtex—sort—buffer`» para ordenar alfabéticamente el contenido de nuestra base de datos. Este mandato establece el orden basándose en las etiquetas identificativas, y además no afecta al contenido del *buffer* que no constituya un registro.

Pero mejor que ordenar el contenido del *buffer* de vez en cuando, es conseguir que se mantenga automáticamente ordenado, lo cual garantiza el buen funcionamiento de varias funciones como la detección de etiquetas duplicadas, o la generación de prefijos o de etiquetas.

Para activar el mantenimiento automático del orden hay que realizar las siguientes acciones:

- Comprobar que el *buffer* está libre de errores sintácticos (ver apartado anterior).
- Ordenar manualmente el *buffer*.
- Asignar a la variable «`bibtex—maintain—sorted—entries`» el valor de “t”.

Una vez activado el mantenimiento automático del orden, las macros que generan nuevas entradas en la base de datos (véase la sección 9.1), empezarán por preguntar la etiqueta para el nuevo registro, ya que antes de generar el registro hay que saber en qué lugar insertarlo, y este viene determinado por el orden alfabético de las etiquetas. Aquí podremos usar la función de auto-completado, ofreciéndonos como posibilidades las etiquetas y campos *crossref* ya definidas en el *buffer*. Finalmente el registro se insertará; y podemos editarlo normalmente, aunque en el caso de que modifiquemos la etiqueta que determina su lugar alfabético, se hará preciso ejecutar el mandato “C-c C-c” («`bibtex—clean—entry`») para que el registro sea reubicado en el *buffer*.

*Como se ha dicho la ordenación alfabética sólo afecta a los registros. Aunque puede afectar también a las definiciones de cadenas con la orden “@STRING”. Ello depende del valor de la variable «`bibtex—sort—ignore—string—entries`». Cuando se le asigna el valor “t” estas entradas se ignoraran, pero con el valor “nil” no serán ignoradas.*

*Esta variable se puede establecer localmente para cada fichero. Si no sabe cómo hacerlo, consulte el apéndice sobre la personalización de Emacs.*

Si pensamos utilizar referencias cruzadas entre las entradas de la base de datos, es buena idea que a la entrada que será llamada por los campos “`crossref`” del fichero bibtex, se le asigne una etiqueta que empiece por “zz” para asegurarnos así de que cuando la base sea ordenada alfabéticamente, estas entradas se coloquen al final, ya que las referencias entre registros de bibtex exigen que la obra referenciada esté *después* que la referenciadora, pero sin embargo Emacs no tiene en cuenta esta circunstancia al ordenar alfabéticamente.

### 9.3.3. Generación automática de etiquetas

Si «`bibtex—clean—entry`» es ejecutado con un prefijo numérico, o si es ejecutado en un registro en el que no se ha llenado el campo para la etiqueta, automáticamente se calcula una etiqueta para el registro, la cual se insertará automáticamente en su lugar o se permitirá su edición por el usuario antes de la inserción, dependiendo del valor de la variable «`bibtex—autokey—edit—before—use`».

*En principio la etiqueta se calcula partiendo del contenido del campo autor o editor (si no se proporciona un autor). No obstante podemos personalizar la forma en la que se calcula la etiqueta, de tal manera que esta puede tener un prefijo, una parte*

*basada en el nombre del autor, otra parte basada en el año y otra basada en el campo título. Podemos además determinar el modo en que se concatenarán todas las partes.*

*Una descripción de todas las variables implicadas, convertiría a este apartado en más largo de lo que tenía pensado. Por lo tanto, para ver cuáles son y cuál es su significado, además de leer la ayuda html bibtex (el paquete de extensión de Emacs, no el programa), podemos también acceder a dicha información abriendo un buffer de configuración para el grupo de opciones llamado “Bibtex Autokey”. Allí, además de recogerse todas las variables, se indica para qué sirve cada una de ellas.*

## 10. Uso del compilador $\LaTeX$ y de otras utilidades externas relacionadas

Emacs dispone del mandato “M-!” («shell—command») para comunicarse con la shell y ejecutar en ella mandatos del sistema operativo o aplicaciones. AUC  $\TeX$  lo que hace es aprovechar esta utilidad para *preconfigurar* la ejecución desde Emacs del compilador  $\TeX$  y de otras herramientas externas asociadas, con lo que, al permitir que desde una misma aplicación (Emacs) podamos crear los ficheros, modificarlos, imprimirlos, convertirlos y visualizarlos, no es inexacto denominar a la combinación Emacs-AUC  $\TeX$  *entorno integrado* para la creación de ficheros en formato  $\TeX$ .

Es al estudio de estos mandatos externos al propio Emacs a lo que se dedica la presente sección.

### 10.1. Mandatos externos que se pueden ejecutar

Mediante los mandatos externos al propio Emacs implementados por AUC  $\TeX$  podremos:

**Compilar el documento:** Los mandatos que Emacs reconoce y van dirigidos a la compilación son “AmSTeX”, “LaTeX”, “LaTeX Interactive”, “LaTeX PDF”, “LaTeX2e”, “Makeinfo”, “Makeinfo HTML”, “TeX” y “TeX Interactive”. La diferencia entre unos y otros está en las convenciones de formato o sintaxis del documento a compilar ( $\TeX$  puro,  $\LaTeX$ , Info o AmsTeX), en el tipo de fichero de salida que generarán (dvi, pdf o html) y en si la compilación se hará o no de modo interactivo. Yo no le recomiendo el modo interactivo a nadie, ya que veo preferible terminar la compilación y luego comprobar si ha habido o no errores. Pero la cosa va en gustos.

**Ejecutar herramientas adicionales de  $\LaTeX$ :** En concreto podemos usar el mandato “BibTeX” para ejecutar el programa gestor de bases de datos bibliográficos del mismo nombre, “Check” para ejecutar “lacheck”, una utilidad que analiza sintácticamente un fichero  $\LaTeX$ <sup>34</sup>, e “Index” que lanzará la ejecución de makeindex, utilidad para generar índices terminológicos.

**Visualizar e imprimir el fichero de salida:** Para ello Emacs incorpora los mandatos “Print”, “Queue” y “View”. El primero envía el fichero resultado de la compilación (dvi, pdf o html) a la impresora; el segundo lo pone en la cola de impresión, y el tercero arranca un programa visor que nos permita ver en pantalla dicho fichero. En una instalación Linux estándar, se utiliza como visor predeterminado “xdvi”.

**Otras acciones posibles:** Además de los mandatos examinados, se incluyen los siguientes tres mandatos:

**File:** Hay que decir que los diseñadores de AUC  $\TeX$  no fueron muy afortunados al bautizar este mandato, ya que el nombre no es especialmente claro. En realidad este mandato lo que hace es lanzar la utilidad “dvips” que convierte un fichero DVI al formato Post-Script.

**Spell:** Ejecuta “ispell” sobre el fichero. Aquí debo decir que como corrector ortográfico ispell no me gusta, ya que es excesivamente anglosajón y considera erróneas todas las palabras con tilde, incluso aunque esté cargado un diccionario de español. Yo suelo usar aspell en lugar de ispell.

**Other:** Este último mandato se incluye para poder ejecutar desde Emacs cualquier mandato. Cuando ejecutamos el mandato Other, en el minibuffer se nos pregunta por la cadena de texto correspondiente al mandato que hay que ejecutar.

*Los mandatos ejecutables vienen definidos por la variable «TeX—command—list», que consiste en una lista en la que cada entrada tiene a su vez los siguientes cinco elementos:*

1. *El primer elemento contiene el nombre del mandato tal y como el usuario debe utilizarlo. Por ejemplo “TeX” o “LaTeX Interactive”.*

---

<sup>34</sup>Permítaseme unas palabras sobre esta utilidad que tal vez no sea tan conocida como debiera: con ella pueden detectarse no sólo posibles futuros errores de compilación, sino, sobre todo, errores de estilo que el compilador no detecta, tales como, por ejemplo, espacios en blanco *antes* de un signo de puntuación. Estos *errores* los muestra lacheck en una ventana adicional que se genera en la parte inferior del marco de Emacs, y podemos navegar entre ellos pulsando “C-x ‘”, que funciona de modo similar a como más adelante se explica que lo hace la función «TeX—next—error». Otra utilidad que analiza sintácticamente ficheros  $\LaTeX$  es “chktx”. Si la instalamos, podemos ampliar la lista de mandatos ejecutables por Emacs (en seguida se explica cómo) e incluirlo. Es más configurable que “lacheck”, pero detecta menos *errores* que éste último.

2. El segundo elemento contiene la cadena de texto que hay que pasar a la shell del sistema operativo para que el mandato se ejecute. Esta cadena puede contener —si así lo deseamos— la ruta de acceso al mandato, o las opciones de línea de mandato que queramos. Por ejemplo, cuando Emacs ejecuta los mandatos “TeX” y “TeX Interactive”, en ambos casos se llama al compilador de TeX. La diferencia está en que en el primer caso lo hace con la opción “nonstopmode” y en el segundo caso no.
3. El tercer elemento de la lista es algo más complejo: indica qué función de Emacs (escrita en emacs-lisp) lanzará el proceso. Podemos elegir entre alguna de las que a continuación se detallan o escribir nuestra propia función. A priori Emacs incluye las siguientes funciones:
  - «TeX—run—command»: inicia el proceso y muestra la salida en un buffer separado. Comprueba que no haya dos mandatos ejecutándose para el mismo fichero. Devuelve el proceso objeto.
  - «TeX—run—format»: similar al anterior, pero asume que la salida es creada por un paquete de macros.
  - «TeX—run—TeX»: para salida de TeX.
  - «TeX—run—LaTeX»: para salida de  $\LaTeX$ .
  - «TeX—run—Interactive»: ejecuta TeX o  $\LaTeX$  de modo interactivo.
  - «TeX—run—BibTeX»: para salida de BibTeX.
  - «TeX—run—compile»: usa la opción “compile” para ejecutar el proceso.
  - «TeX—run—shell»: usa la función «shell—command» para ejecutar el proceso.
  - «TeX—run—discard»: inicia el proceso en segundo plano, descartando su salida.
  - «TeX—run—background»: inicia el proceso en segundo plano, pero la salida se muestra en otra ventana (en el sentido que Emacs da al término ventana).
  - «TeX—run—silent»: inicia el proceso en el segundo plano.
  - «TeX—run—dviout»: opción especial para el visualizador dviout PC-9801 para idioma japonés.

Para escribir nuestra propia función, ésta debe admitir tres argumentos: el nombre del mandato, la cadena del mandato (los dos primeros elementos de la lista) y el nombre del fichero a procesar.
4. El cuarto elemento indica si antes de lanzar el proceso debe permitírsele al usuario que modifique la cadena de texto que lo ejecuta. Por defecto vale “nil”, lo que significa que no se dará al usuario dicha oportunidad.
5. El quinto elemento está obsoleto y es ignorado.

En consecuencia, modificando esta lista podemos añadir nuevos mandatos, o cambiar el modo en el que alguno de los mandatos se ejecuta. Podríamos, por ejemplo, añadir un mandato para generar un fichero HTML, o añadir mandatos para visualizar formatos adicionales al DVI, o cambiar el programa visor predeterminado (me gusta más kdvi que el usado por defecto xdvi), o —algo muy recomendable— cambiar el uso de ispell por el de aspell como corrector ortográfico... Las posibilidades son tantas como nuestra imaginación.

## 10.2. Cómo ejecutar los mandatos externos

Emacs dispone de tres mandatos para lanzar un proceso externo:

- «TeX—command—master» (“C-c C-c”)
- «TeX—command—buffer» (“C-c C-b”)
- «TeX—command—region» (“C-c C-r”)

La diferencia entre ellos está en a qué se aplica el mandato: “C-c C-c” aplicará el mandato al documento maestro, “C-c C-b” lo aplicará al *buffer* actualmente objeto de edición —sea o no el documento maestro— y “C-c C-r” lo aplicará a la región actualmente seleccionada.

Por lo tanto si no estamos en un proyecto multidocumento, o se trata de un proyecto multidocumental pero estamos editando el documento maestro, no hay diferencias entre utilizar “C-c C-c” o “C-c C-b”. Además en muchos mandatos la noción de *documento maestro* es irrelevante. Por ejemplo, el mandato BibTeX.

Por otra parte, pudiera pensarse que algunos mandatos solo tiene sentido ejecutarlos sobre el documento principal. Fundamentalmente los mandatos de compilación ya que en un documento L<sup>A</sup>T<sub>E</sub>X cualquier intento de compilar un documento o una zona del documento *que no incluya el encabezado* generará un error, pues la información del encabezado es necesaria para la compilación.

Curiosamente este problema está resuelto para la compilación de una región, pero no para la compilación de un *buffer*<sup>35</sup>.

Cuando se ordena que se compile una región, Emacs genera un documento temporal en el que se copia el encabezado del documento maestro, el texto a compilar, y *la cola* del documento maestro. Ese fichero transitorio, que por defecto se denomina *region.tex* es el que se compilará. Pueden seguir produciéndose errores —si por ejemplo una definición no se ha incluido en el encabezado sino en una parte del documento que no está en la parte seleccionada que vamos a compilar—, pero al menos con ello se reducen sensiblemente las posibilidades de error.

*El nombre del fichero en el que se guarda temporalmente la región para aplicar el mandato «TeX—command—region», se almacena en la variable «TeX—region». Se considera que el encabezado es todo lo que hay entre el principio del documento maestro y la línea que coincida con la expresión regular almacenada en la variable «TeX—header—end». Se considera que la cola o parte final del documento es lo*

---

<sup>35</sup>Es cierto que el mandato «TeX—command—buffer» no está documentado en la ayuda de AUC T<sub>E</sub>X, por lo que es posible que aún no esté totalmente desarrollado, o que sea un residuo de versiones anteriores; cualquiera de las dos posibilidades explicaría esta aparente contradicción consistente en que sea posible compilar sólo un fragmento de un fichero sin error, pero no lo sea compilar todo el contenido de un fichero que no sea el principal.

que hay desde la línea que coincida con la expresión regular almacenada en `«TeX—trailer—start»` y el final del fichero.

Estas variables las podemos establecer como variables de fichero (véase el apéndice sobre cómo personalizar Emacs, en pág. 84). Pero también las podemos definir con carácter general. Para ello hay que modificar `«TeX—mode—hook»`, `«plain—TeX—mode—hook»` o `«LaTeX—mode—hook»`, según queramos que la modificación afecte en general a ficheros TeX, plain-TeX o LaTeX.

Cuando se ordena la ejecución de cualquier mandato externo, lo primero que se comprueba es si el *buffer* actual ha cambiado desde la última modificación. En caso afirmativo se empieza por preguntar si queremos guardar los cambios.

*Asimismo se puede comprobar las posibles modificaciones de otros ficheros relacionados, para lo cual hay que establecer la variable `«TeX—check—path»` que por defecto tiene el valor de “nil”.*

A continuación se analizan los ficheros existentes en el directorio de trabajo actual para intentar deducir qué mandato es más normal que el usuario quiera ejecutar, y así si el fichero fuente (de extensión tex) fue modificado con posterioridad al fichero resultado de la compilación, ello significa que el fichero compilado (DVI) no está actualizado, por lo que se nos sugerirá, como mandato por defecto, el que llame al compilador adecuado para el tipo de documento (TeX,  $\LaTeX$ , AmsTeX...). Si por el contrario el fichero DVI está actualizado, Emacs supondrá que queremos verlo, y sugerirá como mandato por defecto: View.

*En ficheros  $\LaTeX$  se considera por defecto que la compilación normal se hará ejecutando “LaTeX” y no con “LaTeX Pdf”. Este comportamiento lo podemos cambiar para un fichero concreto estableciendo en él la variable de fichero `«TeX—command—default»` con el valor correspondiente a alguna entrada de la lista almacenada en `«TeX—command—list»`. Si queremos alterar el mandato por defecto con carácter general, hay que incluir `«TeX—command—default»` en `«LaTeX—mode—hook»` (o, si trabajamos con ficheros plain-tex o con ficheros tex puros, en `«plain—TeX—mode—hook»` o `«TeX—mode—hook»`).*

Los distintos mandatos externos que Emacs ejecuta, se pueden a su vez agrupar en dos categorías: aquellos que son lanzados directamente, y aquellos otros en los que antes de ejecutarse se solicita confirmación del mandato que se ejecutará. Esa confirmación se solicita en el minibuffer donde se muestra la cadena de texto que contiene el mandato a ejecutar. En ese instante podemos —si queremos— cambiar la cadena de texto y ejecutar un mandato distinto.

En este segundo grupo se incluyen los mandatos View, Print, File y Other. Así podríamos, por ejemplo, ejecutar el visor de documentos no sobre el fichero DVI fruto de la compilación con  $\LaTeX$ , sino sobre el fichero PDF fruto de la compilación con PDF $\LaTeX$ , o usar como visor de documentos DVI en lugar del predefinido por Emacs (“xdvi”), “kdvi” que ofrece bastantes más posibilidades ya



que permite búsquedas de texto dentro del fichero DVI, o enviarlo a la impresora, convertirlo a PS o PDF, etc.

### 10.3. Comprobación de errores

Tras ejecutar T<sub>E</sub>X o L<sup>A</sup>T<sub>E</sub>X para compilar nuestro documento, Emacs nos informará, en el minibuffer, si ha habido errores o, al menos, advertencias y, si la compilación pudo realizarse, cuántas páginas se han generado.

En el caso de que haya habido errores conviene comprobarlos, para lo cual Emacs nos ofrece las siguientes facilidades para realizar tal comprobación:

«**TeX—next—error**»: Este mandato, asociado a “C-c ‘”<sup>36</sup>, desplaza el cursor a la línea en la que se ha producido el siguiente error (o el primero si acabamos de compilar) y en una ventana que se abre en la parte inferior del marco de Emacs se informa (en inglés) sobre la naturaleza del error, así como sobre sus posibles causas. Es por esta utilidad por la que resulta utilísimo en documentos L<sup>A</sup>T<sub>E</sub>X activar el modo menor de Emacs “auto-fill” para conseguir que cada línea de texto realmente termine con un salto de línea, así el cursor se colocará lo más cerca posible del error que sea<sup>37</sup>.

«**TeX—toggle—debug—bad—boxes**»: Normalmente Emacs no cataloga como error el que alguna caja sea incorrecta (se salga de los márgenes), pero ejecutando este mandato (asociado a las teclas “C-c C-w”) podemos cambiar dicho comportamiento.

Este mandato funciona como un interruptor: es decir si la detección de cajas erróneas estaba activada la desactiva y viceversa.

*La variable «TeX—display—help» determina si cuando se muestra un error, junto con el error propiamente dicho, hay que ofrecer o no algún tipo de ayuda. Así se hará cuando esta variable no valga “nil”. No se hará en caso contrario.*

### 10.4. Control de procesos externos

*Mientras se está ejecutando alguna de las aplicaciones externas que hemos visto en esta sección, es posible controlar el proceso correspondiente a la misma sin salir de Emacs, utilizando los siguientes mandatos:*

---

<sup>36</sup>Se trata de “C-c” y luego el acento grave, el cual en un teclado español requiere, que primero se pulse la tecla del acento grave y después la barra espaciadora.

<sup>37</sup>Aunque eso depende del error; porque algunos errores no se detectan por T<sub>E</sub>X hasta el final de un párrafo, o incluso hasta el final del documento. Eso ocurre, sobre todo, en el caso de grupos (o comillas tipográficas) sin cerrar. Aunque en estos casos la información de ayuda sobre el error nos indica en qué línea se abrió el grupo no cerrado, lo cual es una pista importante. Mediante el mandato «goto—líne» podemos saltar directamente a la línea en cuestión.

- «**TeX—kill—job**»: Esta función (asociada a la combinación de teclas “C-c C-1”) permite matar la aplicación externa a Emacs que se haya iniciado mediante las órdenes vistas en esta sección ( «*TeX—command—master*», «*TeX—command—buffer*» o «*TeX—command—region*»).
- «**TeX—recenter—output—buffer**»: Este mandato (“C-c C-1” abre una ventana en el marco de Emacs para mostrar en ella un buffer con la salida del último mandato ejecutado (que haya producido una salida) en la sesión de trabajo actual.

# APÉNDICES

## A. Modificar variables de Emacs para la personalización

Emacs es tan personalizable, que explicar con detalle cómo hacerlo llevaría casi tanto espacio como explicar su funcionamiento estándar. Por ello en las páginas que siguen me concentraré exclusivamente en el establecimiento de *variables* para la personalización de Emacs.

Internamente Emacs ejecuta mandatos que son en realidad funciones escritas en el lenguaje Emacs-Lisp, aunque algunas de sus funciones están escritas en C. Esas funciones utilizan el valor de ciertas variables externas para determinar su funcionamiento. A esas variables externas, modificables por el usuario, se las denomina *opciones de usuario* ya que es cambiándoles el valor como podemos cambiar el funcionamiento predeterminado de Emacs.

El valor de una *opción de usuario* puede modificarse:

- Para la sesión de trabajo actual.
- Para siempre.
- Para cierto fichero.

En los dos primeros casos la modificación de la variable exige que escribamos una sentencia en formato Emacs-Lisp y, por lo tanto, se hace preciso conocer dicho formato, lo cual, obviamente, excede el propósito de este documento. Podemos, no obstante, introducir cambios sin saber nada de Elisp usando los propios mandatos que Emacs incluye para su personalización. Véase al respecto más adelante (página 89).

Por defecto el cambio en un valor afecta a la sesión de trabajo actual. Para que sea recordado entre distintas sesiones, hay que almacenar dicha modificación en el fichero de configuración de Emacs. Para que una variable asuma cierto valor al cargar un fichero, hay que guardarla como *variable de fichero*.

Por los demás los valores de las variables se ajustan a las siguientes convención:

- El valor "t" se interpreta —en variables de tipo lógico— como *verdadero*.
- El valor "nil" se interpreta como falso.
- Los valores alfabéticos (distintos de "t" y "nil") deben ir encerrados entre comillas dobles ("").
- Los valores numéricos no deben ir encerrados entre comillas.

## El fichero de configuración de Emacs

Cuando se arranca Emacs lo primero que hace es leer su fichero de configuración. En él encontrará la información pertinente para ajustar su comportamiento a las indicaciones que se le hayan dado.

El contenido de este fichero es siempre el de un programa escrito en Elisp. En consecuencia para almacenar un cambio permanente de alguna variable basta con escribir en el fichero la sentencia Elisp adecuada.

Enseñar Elisp está fuera de mi intención al escribir este documento. Por lo tanto no entraré en cómo escribir sentencias de asignación de valores en Elisp.

En sistemas Unix-Linux el fichero de configuración se denomina `.emacs` y se encuentra en el directorio "*home*" de cada usuario del sistema. Al estar su nombre precedido por un punto, el fichero es un *fichero oculto* que no se muestra con la orden `ls` normal de Unix. Puede no obstante editarse como cualquier otro fichero de texto normal, con la salvedad de que sus cambios no serán tenidos en cuenta por Emacs hasta que el programa se reinicie.

A veces, sin embargo, en el fichero `.emacs` no encontraremos toda la configuración, sino que el propio `.emacs` puede cargar otros ficheros con configuraciones individualizadas. Esto se puede ver en el propio fichero `.emacs`. Si hay en él alguna sentencia del tipo "(load *fichero*)" significa que ese fichero será leído y cargada su configuración. En mi configuración, por ejemplo se cargan siempre otros dos ficheros ocultos: `".gnu-emacs"` y `".gnu-emacs-custom"`; ambos en el directorio *home*. El último fichero, además, es el que el propio Emacs genera cada vez que hacemos uso de alguna de los mandatos «customize». Los cambios se guardan ahí y no en `.emacs`, y de esta manera Emacs puede saber qué variables han sido personalizadas mediante sus propias funciones de personalización.

Pero la mayor parte de la configuración de mi `.emacs` está en `".gnu-emacs"`. Ahí es donde se encuentran, por ejemplo, los cambios que en la configuración general de Emacs introduce AUC  $\text{\TeX}$ .

No obstante, para lo que a nosotros nos interesa, lo mejor es escribir la configuración en `.emacs`, ya que en caso de error podemos simplemente borrarlo de manera que la próxima ejecución de Emacs generará un fichero de configuración nuevo y vacío, con todas las opciones por defecto. Lo único que hay que hacer es asegurarnos de que nuestras órdenes de configuración se encuentren detrás de las órdenes "load" por las que se leen otros ficheros.

## Establecimiento de variables de documento

En un documento se pueden especificar valores para variables locales. Estos valores serán usados por Emacs.

Aunque es posible especificar variables de fichero en una sola línea, lo más corriente es hacerlo en varias siguiendo la siguiente convención:

- La lista de variables locales debe encontrarse al final del fichero, es decir: entre el final de la lista y el final del fichero sólo puede haber líneas en blanco, espacios en blanco o tabuladores (caracteres todos ellos que en general son ignorados al final de un fichero).

- La lista de variables debe empezar con una línea que contenga el texto «Local Variables:». La última línea de la lista debe contener la cadena de texto «End:». No importa si antes de dichas cadenas de texto se contienen otros caracteres. En particular puede y suele colocarse una marca de comentario para que el compilador ignore esta lista.

De hecho los caracteres escritos antes del texto «Local Variables:» se identificarán como *prefijo* y si se repiten en los restantes elementos de la línea serán ignorados siempre que se encuentren en la misma posición. De la misma manera lo que se escriba detrás de «Local Variables:» se considerará un *sufijo* y será ignorado en todos los elementos de la lista, siempre y cuando se repita en la misma posición.

- Cada línea entre la primera y la última dará valor a alguna variable de acuerdo con el siguiente formato:

```
[Prefijo] Nombre: valor [Sufijo]
```

El prefijo y el sufijo son opcionales (por eso los he puesto entre corchetes) y requieren que la primera línea de la lista los contenga. Entre el nombre de la variable y su valor debe haber un signo de dos puntos. Para el valor de la variable se aplican las reglas generales (“t”, “nil”, etc...)

Habitualmente además la primera variable establecida recoge el modo mayor en el que Emacs debe editar el documento. Así, en el siguiente ejemplo:

```
%%% Local Variables: ***
%%% mode: latex ***
%%% TeX-master: t ***
%%% End: ***
```

De hecho este ejemplo es tan corriente que casi habría que incorporarlo a cualquier documento  $\LaTeX$  que escribamos en Emacs. En él se da valor a dos variables: la variable *mode*, que almacena el modo mayor del documento, y el nombre del documento maestro del que este depende. Sobre la utilidad de estos *documentos maestros*, consulte la sección 8.1, en pág. 64.

Pero además de estas variables, por este procedimiento podemos establecer muchas otras variables. Así el siguiente ejemplo:

```
%%% Local Variables:
```

```
%%% mode: latex
%%% TeX-master: "principal"
%%% TeX-auto-regexp-list: TeX-auto-full-regexp-list
%%% TeX-auto-parse-length: 5000
%%% End:
```

Además de indicar que el documento está escrito para  $\text{\LaTeX}$  y que depende de otro documento llamado `principal.tex` (no hay que indicar en el valor de la variable la extensión), se da valor a otras dos variables que le indican a Emacs que, para ese documento, realice una completa comprobación de sintaxis durante los primeros 5000 caracteres.

### Usar un *buffer de personalización* para modificar el valor de variables

Mediante el mandato «M-x customize» podemos crear un *buffer de personalización*. Se trata de un *buffer* especial en el que podemos desplazarnos entre las numerosas opciones de usuario de Emacs y alterar su valor, determinando si queremos que la modificación esté disponible sólo para la sesión actual o si, por el contrario, deseamos que la modificación se almacene para futuras sesiones, en cuyo caso el propio Emacs modificará nuestro fichero de configuración.

Aunque la apariencia del *buffer de personalización* es muy diferente según estemos ejecutando Emacs en modo gráfico o en modo de sólo texto, en ambos casos podemos movernos por él mediante las teclas TAB y MAYUS-TAB. La primera mueve el cursor a la próxima opción, y la segunda mueve el cursor a la opción anterior. Estando el cursor sobre una opción concreta la seleccionaremos pulsando RET. En el modo gráfico también podemos hacer click con el ratón sobre las opciones (que se muestran con apariencia de botones gráficos).

Este *buffer*, por otra parte, está organizado en forma de árbol. Las distintas opciones de personalización forman grupos, los cuales a su vez pueden tener otros grupos anidados, etc. Cuando ejecutamos el mandato «customize» saltamos al nivel inicial, en el que se encuentran los grupos principales. Esto significa que cuando queremos modificar simplemente una variable, para localizarla debemos saber en qué grupo se ubica y navegar hacia ella mediante las teclas TAB y RET, lo que puede ser lento —lentísimo si no estamos seguros de en qué grupo se ubica la variable—

Para estos casos es preferible ejecutar el mandato «M-x customize—option». Tras introducir el nombre de la variable a modificar, saltaremos directamente al grupo que la contiene.

En todo caso, para ficheros  $\text{\LaTeX}$ , la mayoría de las opciones que nos puede interesar modificar estarán en alguno de los grupos que forman parte del grupo “TeX”, el cual, a su vez, está incluido en el grupo de primer nivel “wp”. El grupo “TeX”, por su parte, contiene dentro de sí a los grupos “Font LaTeX”, “AUC TeX”, “RefTeX”, “BibTeX”, “Tex File”, “Tex Run” y “Tex View”.

Otras opciones posibles son:

- «customize—group» Abre un *buffer* de personalización para un grupo determinado de opciones.
- «customize—apropos» Abre un *buffer* de personalización que incluirá todos los grupos cuyo nombre coincida con la expresión regular que se indique.
- «customize—changed—options» Abre un *buffer* de personalización que incluye todas aquellas variables cuyo valor ha cambiado desde que se instaló Emacs (o el paquete que incluyó dichas variables).
- «customize—customized» Similar al anterior, pero ahora se agrupan, no las variables cuyo valor haya simplemente cambiado, sino exclusivamente aquellas que, después del cambio, han asumido un valor diferente del inicial.

Mediante la ejecución de cualquiera de estos mandatos, una vez hayamos llegado a la variable cuyo valor pretendemos modificar, en el *buffer* podremos ver junto con el nombre de la variable, una serie de botones o campos (según estemos o no en modo texto) y tal vez el valor de la variable. Estos campos nos informan sobre varios aspectos. En concreto el estado actual de la variable, y si este estado ha cambiado desde la instalación. También puede ofrecerse ayuda sobre la utilidad de la variable y los valores admisibles.

Junto a la variable puede haber los siguientes campos especiales (que en modo gráfico se muestran como botones):

- State: ofrece información sobre el estado actual de la variable.
- Show: se encuentra cuando el valor de la variable ha sido *ocultado*. Debemos hacer RET sobre él (o click) para mostrar dicho valor.
- Hide: podemos hacer click sobre él para ocultar el valor de la variable.
- More: haciendo click sobre él se nos muestra información sobre la variable en cuestión.
- Toggle: se encuentra en las variables lógicas cuyo valor puede ser “t” (verdadero) o “nil” (falso). Haciendo click sobre él cambiaremos el valor.
- Ins, Del: estos campos se encuentran en las variables de tipo lista. Mediante el primero insertaremos un nuevo valor en la lista, y mediante el segundo eliminaremos de la lista el valor a la derecha.

En las variables cuyo valor consiste simplemente en un número o en un texto, deberemos escribirlo en el campo donde se encuentra el valor actual.

Una vez que hayamos establecido el valor deseado para la variable, debemos ir al principio del *buffer*. Allí encontraremos los siguientes campos:

- Set for Current Session: haciendo click o RET sobre él cambiaremos el valor de la variable exclusivamente para la sesión actual. El efecto es similar al que se obtendría ejecutando el mandato «M-x set—variable» seguido del nombre de la variable y del nuevo valor.

- Save for future Sessions: haciendo click sobre este campo, el cambio hecho en la variable se almacenará para el futuro (en el fichero de inicialización).
- Reset: deja las cosas en el estado en el que estaban antes de que entráramos en el *buffer* de personalización.
- Reset to Saved: deja las cosas en el estado en el que se encuentran en nuestro fichero de configuración.
- Erase Customization: devuelve a todas las variables mostradas en el *buffer* su valor inicial (el que tenían cuando se instaló Emacs).
- Finish: termina la edición del *buffer* de personalización.



## B. Resumen de mandatos

El presente apéndice pretende ser un resumen rápido de los mandatos ordenados por tareas. Al respecto téngase en cuenta que 1) No se incluyen todos los mandatos. 2) Los detalles sobre su funcionamiento hay que buscarlos en el lugar donde se explica cada uno de ellos. 3) En concreto hay mandatos que sólo funcionarán si están activados los modos bibtex, outline o reftex.

### Utilidades generales

- **C-c TAB** cargar la página Info de AUC T<sub>E</sub>X.
- **C-c ?** mostrar en una ventana todos los mandatos incorporados a AUC T<sub>E</sub>X.
- **C-c \*** selecciona la sección actual.
- **C-c .** selecciona el entorno actual.
- **C-c =** cargar en ventana aparte el índice sistemático del documento.
- **C-c ^** abrir el documento maestro.
- **C-c C-d** guardar el documento actual y todos los relacionados con él que estén abiertos.

### Reformateado del texto fuente

- **C-c C-q C-p** reformatear párrafo.
- **M-q** reformatear párrafo.
- **C-c C-q C-e** reformatear entorno.
- **C-c C-q C-s** reformatear sección.
- **C-c C-q C-r** reformatear región.
- **N-g** reformatear región.
- **C-u** seguido por cualquiera de los mandatos anteriores provoca que el reformateado justifique las líneas a la derecha.

### Caracteres especiales

- **C-c ;** marca como comentario la región.
- **C-c %** marca como comentario el párrafo.
- **C-c '**  marca como comentario el párrafo.
- **C-u C-c ;** elimina marcas de comentario de la región.
- **C-c** elimina marcas de comentario de la región.
- **C-c "** elimina marcas de comentario del párrafo.
- **C-c {** inserta simultáneamente llaves de apertura y cierre.

### Introducción general de mandatos de L<sup>A</sup>T<sub>E</sub>X

- **M-TAB** completa la macro de L<sup>A</sup>T<sub>E</sub>X que haya empezado a introducirse.
- **C-c C-m** inserta una macro de L<sup>A</sup>T<sub>E</sub>X.
- **C-c RET** inserta una macro de L<sup>A</sup>T<sub>E</sub>X.

### Especificación del formato de fuente

- **C-c C-f C-r** inserta texto roman.
- **C-c C-f C-f** inserta texto sanserif.
- **C-c C-f C-t** inserta texto tipo máquina de escribir.
- **C-c C-f C-b** inserta texto en negrita.
- **C-c C-f RET** inserta texto sin negrita.
- **C-c C-f C-i** inserta texto en cursiva.
- **C-c C-f C-s** inserta texto en oblicua.
- **C-c C-f C-e** inserta texto enfatizado.
- **C-c C-f C-u** inserta texto de perfil recorto.
- **C-c C-f C-c** inserta texto en versales.
- **C-c C-f C-n** inserta texto normal.
- **C-u** antes de cualquiera de los mandatos anteriores, en lugar de insertar un nuevo formato sustituye el formato que hubiera por el que se especifique.
- **C-c C-f C-d** borra la especificación de formato.

### Trabajo con unidades estructurales

**NOTA** Las funciones de *movimiento* entre títulos requieren activar el modo menor outline.

- **C-c \*** selecciona la sección actual.
- **C-c C-q C-s** reformatear sección.
- **C-c C-s** insertar unidad estructural.
- **C-c C-o C-d** ocultar el contenido de la sección.
- **C-c C-o C-s** mostrar el contenido de la sección.
- **C-c C-o C-n** ir al próximo título.
- **C-c C-o C-p** ir al título anterior.
- **C-c C-o C-f** ir al próximo título del mismo nivel que el actual.

- **C-c C-o C-b** ir al título anterior del mismo nivel que el actual.
- **C-c C-o C-u** ir al título de nivel inmediatamente superior.
- **C-c C-r** realizar indexación global para las frases seleccionadas en el fichero de frases.
- **C-c C-a** realizar indexación global para todo el fichero de frases.

### Trabajo con entornos

- **C-c** . selecciona el entorno actual.
- **C-c C-q C-e** reformatear entorno.
- **C-c C-z** ocultar el contenido del entorno.
- **C-c C-x** mostrar el contenido del entorno.
- **M-C-a** ir a la línea de apertura del entorno.
- **M-C-e** ir a la línea de cierre del entorno.
- **C-c ]** cerrar entorno.
- **C-c C-e** insertar entorno.
- **C-u C-c C-e** sustituir entorno.
- **C-c ~** activar modo menor para entorno matemático. En este modo la tecla "acento grave" funciona como tecla *prefijo* para la introducción rápida de macros.

### Etiquetas y referencias cruzadas

#### Mandatos generales

- **C-c (** insertar etiqueta.
- **C-c )** hacer referencia a una etiqueta (Insertar mandato `\ref`).
- **C-c [** insertar cita bibliográfica.
- **C-c /** insertar entrada de índice con generación automática de la etiqueta.
- **C-c <** insertar entrada de índice con edición de la etiqueta.
- **C-c >** ver y editar el índice.
- **C-c &** ver una referencia cruzada.
- **Mandatos relativos a la generación de un índice desde un fichero de frases**
- **C-c |** cargar en un *buffer* el fichero de frases para generación de índice. Si no existe, crearlo.
- **C-c \** insertar palabra o frase en el fichero de frases.
- **C-c C-c** guardar cambios en fichero de frases y volver a documento en edición.
- **C-c C-s** ordenar alfabéticamente las frases del fichero de frases.
- **C-c C-i** informar sobre la frase actual del fichero de frases.
- **C-c C-t** comprobar entradas dobles o superpuestas en fichero de frases.
- **C-c C-x** realizar indexación global para la frase actual.

### Ejecutar mandatos externos

- **C-c C-c** ejecutar mandato sobre fichero maestro.
- **C-c C-b** ejecutar mandato sobre *buffer* actual.
- **C-c C-r** ejecutar mandato sobre la región actual.
- **C-c C-'** ver próximo error del compilador.

### Ficheros bibliográficos

#### Inserción de registros

- **C-c C-b** insertar nuevo registro bibliográfico.
- **C-c C-e C-a** insertar nuevo artículo.
- **C-c C-e C-b** insertar nueva parte de un libro.
- **C-c C-e b** insertar nuevo libro (con editorial conocida).
- **C-c C-e B** insertar nuevo libro (sin editorial conocida).
- **C-c C-e C-c** insertar nueva parte autónoma de un libro.
- **C-c C-e C-m** insertar nueva documentación técnica.
- **C-c C-e m** insertar nueva Tesina de licenciatura.
- **C-c C-e C-p** insertar nuevo artículo en congreso.
- **C-c C-e p** insertar nuevas actas de congreso.
- **C-c C-e P** insertar nueva tesis doctoral.
- **C-c C-e C-t** insertar nuevo informe técnico.
- **C-c C-e C-u** insertar nueva obra sin publicar.
- **C-c C-e M** insertar referencia que no encaje en ninguna de las anteriores.
- **C-c C-e M-p** insertar nuevo preámbulo de fichero.
- **C-c C-e C-s** insertar nueva abreviatura.
- **Movimiento por el fichero**
- **M-C-a** ir al principio del registro.
- **M-C-e** ir al final del registro.
- **TAB** ir al final del campo actual.
- **C-j** ir al próximo campo.
- **Edición y manipulación de campos y registros**
- **C-c "** eliminar llaves delimitadoras del campo.

## Apéndice: Resumen de mandatos

---

- **C-c {** eliminar llaves delimitadoras del campo.
- **C-c }** eliminar llaves delimitadoras del campo.
- **C-c C-d** borrar contenido del campo.
- **C-c C-f** insertar nuevo campo.
- **C-c C-q** reformatear registro.
- **C-c C-o** eliminar prefijos de los nombres de campos.
- **C-c C-c** limpiar registro
- **Seleccionar, eliminar y recuperar**
- **M-C-h** seleccionar registro completo.
- **C-c C-w** eliminar todo el registro.
- **C-c M-w** copiar todo el registro
- **C-c C-k** eliminar todo el campo.
- **C-c M-k** copiar todo el campo.
- **C-c C-y** insertar registro o campo eliminado o copiado.
- **C-c M-y** cambiar registro o campo insertado por otro eliminado en una ocasión anterior.
- **Otras acciones**
- **C-c &** ver las citas a la referencia actual en nuestro documento .tex.

## C. GNU Free Documentation License (Licencia GNU para Documentación Libre)

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To

**"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ." according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
  - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section



unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include

the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## **ADDENDUM: How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## Apéndice: Índice de nombres

---

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

## D. Índice de nombres de órdenes y opciones de usuario

En este índice pueden encontrarse los nombres correspondientes a las órdenes y opciones de usuario de Emacs tratadas en este documento, junto con una indicación de la página en la que son mencionadas.

<b>B</b>	
bibtex-add-entry-hook (opción) .....	70
bibtex-Article (mandato) .....	69
bibtex-autokey-edit-before-use (opción) .....	75
bibtex-beginning-of-entry (mandato) .....	70
bibtex-Booklet (mandato) .....	69
bibtex-Book (mandato) .....	69
bibtex-clean-entry-hook (opción) .....	73
bibtex-clean-entry (mandato) .....	72, 73, 75
bibtex-comma-after-last-field (opción) .....	70, 73
bibtex-copy-entry-as-kill (mandato) .....	72
bibtex-copy-field-as-kill (mandato) .....	72
bibtex-empty-field (mandato) .....	71
bibtex-end-of-entry (mandato) .....	70
bibtex-entry-delimiters (opción) .....	70
bibtex-entry-field-alist (opción) .....	70, 73
bibtex-entry-format (opción) .....	73
bibtex-entry (mandato) .....	69
bibtex-field-delimiters (opción) .....	70
bibtex-field-kill-ring-max (opción) .....	72
bibtex-find-text (mandato) .....	70
bibtex-InBook (mandato) .....	69
bibtex-InCollection (mandato) .....	69
bibtex-InProceedings (mandato) .....	69
bibtex-kill-entry (mandato) .....	72
bibtex-kill-field (mandato) .....	72
bibtex-maintain-sorted-entries (opción) .....	75
bibtex-make-field (mandato) .....	71
bibtex-Manual (mandato) .....	69
bibtex-mark-entry (mandato) .....	72
bibtex-MasterThesis (mandato) .....	69
bibtex-Misc (mandato) .....	69
bibtex-next-fill (mandato) .....	70
bibtex-PhdThesis (mandato) .....	69
bibtex-Preamble (mandato) .....	69
bibtex-predefined-month-strings (opción) .....	74
bibtex-predefined-strings (opción) .....	74
bibtex-Proceedings (mandato) .....	69
bibtex-remove-delimiters (mandato) .....	71
bibtex-remove-OPT-or-ALT (mandato) .....	71
bibtex-sort-buffer (mandato) .....	74
bibtex-sort-ignore-string-entries (opción) .....	75
bibtex-String (mandato) .....	69
bibtex-TechReport (mandato) .....	69
bibtex-text-indentation (mandato) .....	71
bibtex-Unpublished (mandato) .....	69
bibtex-validate (mandato) .....	74
bibtex-yank-pop (mandato) .....	72
bibtex-yank (mandato) .....	72
<b>C</b>	
customize-afropos (mandato) .....	87
customize-changed-options (mandato) .....	87
customize-customized (mandato) .....	87
customize-group (mandato) .....	86
<b>D</b>	
describe-bindings (mandato) .....	12
describe-mode (mandato) .....	12, 13
<b>F</b>	
fill-column (opción) .....	14
font-lock-highlighting-faces (opción) .....	18
font-lock-maximum-decoration (opción) .....	18
font-lock-maximum-size (opción) .....	18
<b>G</b>	
global-font-lock-mode (opción) .....	18, 70
goto-line (mandato) .....	81
<b>H</b>	
hide-body (mandato) .....	36
hide-entry (mandato) .....	36
hide-leaves (mandato) .....	36
hide-other (mandato) .....	36
hide-sublevels (mandato) .....	36
hide-subtree (mandato) .....	36
<b>I</b>	
info (mandato) .....	13
<b>L</b>	
LaTeX-brace-indent-level (opción) .....	15
LaTeX-close-environment (mandato) .....	42
LaTeX-default-environment (opción) .....	43, 44
LaTeX-environment (mandato) .....	43, 44
LaTeX-figure-label (opción) .....	45
LaTeX-fill-environment (mandato) .....	16
LaTeX-fill-paragraph (mandato) .....	16
LaTeX-fill-region (mandato) .....	16
LaTeX-fill-section (mandato) .....	16
LaTeX-find-matching-begin (mandato) .....	42
LaTeX-find-matching-end (mandato) .....	42
LaTeX-float (opción) .....	45
LaTeX-hide-environment (mandato) .....	42
LaTeX-indent-level (opción) .....	15
LaTeX-indent-line (mandato) .....	17
LaTeX-insert-item (mandato) .....	44
LaTeX-item-indent (opción) .....	15
LaTeX-mark-environment (mandato) .....	17
LaTeX-mark-section (mandato) .....	17
LaTeX-math-abbrev-prefix (opción) .....	46
LaTeX-math-list (opción) .....	46
LaTeX-math-mode (mandato) .....	46
LaTeX-mode-hook (opción) .....	12
LaTeX-section-hook (opción) .....	32
LaTeX-section-label (opción) .....	33
LaTeX-section (mandato) .....	32

## Apéndice: Índice de nombres

---

LaTeX-show-environment (mandato) . . . . .	42	reftex-select-mark-and (mandato) . . . . .	54
LaTeX-table-label (opción) . . . . .	45	reftex-select-mark-comma (mandato) . . . . .	54
<b>O</b>			
outline-backward-same-level (mandato) . . . . .	37	reftex-select-mark-to (mandato) . . . . .	54
outline-forward-same-level (mandato) . . . . .	37	reftex-select-mark (mandato) . . . . .	54
outline-minor-mode-prefix (opción) . . . . .	35	reftex-select-next-heading (mandato) . . . . .	51
outline-minor-mode (mandato) . . . . .	35	reftex-select-next (mandato) . . . . .	51
outline-next-visible-heading (mandato) . . . . .	37	reftex-select-previous-heading (mandato) . . . . .	52
outline-previous-visible-heading (mandato) . . . . .	37	reftex-select-previous (mandato) . . . . .	51
outline-up-heading (mandato) . . . . .	37	reftex-select-quit (mandato) . . . . .	51
<b>P</b>			
plain-TeX-auto-regex-list (opción) . . . . .	21	reftex-select-read-label (mandato) . . . . .	52
plain-TeX-mode-hook (opción) . . . . .	12, 80	reftex-select-show-insertion-point (mandato) . . . . .	52
<b>R</b>			
reftex-abbrev-parameters (opción) . . . . .	49	reftex-select-toggle-fancyref (mandato) . . . . .	52
reftex-citation (mandato) . . . . .	55, 56	reftex-select-toggle-follow (mandato) . . . . .	52
reftex-cite-format (opción) . . . . .	56	reftex-select-toggle-varioref (mandato) . . . . .	52
reftex-derive-label-parameters (opción) . . . . .	49	reftex-select-unmark (mandato) . . . . .	54
reftex-display-index (mandato) . . . . .	61	reftex-string-to-label-function (opción) . . . . .	49
reftex-enable-partial-scans (opción) . . . . .	40, 53, 63	reftex-toc-display-index (mandato) . . . . .	40
reftex-format-label-function (opción) . . . . .	49	reftex-toc-external (mandato) . . . . .	39
reftex-format-ref-function (opción) . . . . .	53	reftex-toc-goto-line-and-hide (mandato) . . . . .	39
reftex-fref-is-default (opción) . . . . .	53	reftex-toc-goto-line (mandato) . . . . .	39
reftex-guess-label-type (opción) . . . . .	53	reftex-toc-jump (mandato) . . . . .	39
reftex-idle-time (opción) . . . . .	64	reftex-toc-max-level (mandato) . . . . .	40
reftex-index-default-macro (opción) . . . . .	56	reftex-toc-next-heading (mandato) . . . . .	39
reftex-index-edit-attribute (mandato) . . . . .	62	reftex-toc-next (mandato) . . . . .	39
reftex-index-edit-key (mandato) . . . . .	62	reftex-toc-previous-heading (mandato) . . . . .	39
reftex-index-edit-visual (mandato) . . . . .	62	reftex-toc-previous (mandato) . . . . .	39
reftex-index-edit (mandato) . . . . .	62	reftex-toc-quit-and-kill (mandato) . . . . .	38
reftex-index-find-next-conflict-phrase (mandato) . . . . .	58	reftex-toc-quit (mandato) . . . . .	38
reftex-index-level-down (mandato) . . . . .	62	reftex-toc-rescan (mandato) . . . . .	40
reftex-index-level-up (mandato) . . . . .	62	reftex-toc-show-calling-point (mandato) . . . . .	39
reftex-index-phrase-selection-or-word (mandato) . . . . .	57	reftex-toc-show-help (mandato) . . . . .	38
reftex-index-phrases-info (mandato) . . . . .	58	reftex-toc-toggle-context (mandato) . . . . .	40
reftex-index-phrases-save-and-return (mandato) . . . . .	57	reftex-toc-toggle-file-boundary (mandato) . . . . .	39
reftex-index-selection-or-word (mandato) . . . . .	56	reftex-toc-toggle-follow (mandato) . . . . .	39
reftex-index-show-help (mandato) . . . . .	61	reftex-toc-toggle-index (mandato) . . . . .	40
reftex-index-sort-phrases (mandato) . . . . .	58	reftex-toc-toggle-labels (mandato) . . . . .	40
reftex-index-toggle-range-beginning (mandato) . . . . .	62	reftex-toc-view-line (mandato) . . . . .	39
reftex-index-toggle-range-end (mandato) . . . . .	62	reftex-toc (mandato) . . . . .	38, 39
reftex-index-visit-phrases-buffer (mandato) . . . . .	57	reftex-translate-to-ascii-function (opción) . . . . .	49
reftex-index (mandato) . . . . .	56	reftex-view-crossref (mandato) . . . . .	63
reftex-insert-label-flags (opción) . . . . .	49	reftex-vref-is-default (opción) . . . . .	53
reftex-label-alist (opción) . . . . .	53	reindent-then-newline-and-indent (mandato) . . . . .	17
reftex-label-illegal-re (opción) . . . . .	49	revert-buffer (mandato) . . . . .	40, 63
reftex-label-menu-flags (opción) . . . . .	53	<b>S</b>	
reftex-label (mandato) . . . . .	48	shell-command (mandato) . . . . .	76
reftex-level-indent (opción) . . . . .	53	show-all (mandato) . . . . .	36
reftex-multiref-punctuation (opción) . . . . .	53	show-branches (mandato) . . . . .	36
reftex-reference (mandato) . . . . .	50, 52	show-children (mandato) . . . . .	36
reftex-select-accept (mandato) . . . . .	52	show-entry (mandato) . . . . .	36
reftex-select-callback (mandato) . . . . .	52	show-subtree (mandato) . . . . .	36
reftex-select-help (mandato) . . . . .	51	<b>T</b>	
reftex-select-jump-to-previous (mandato) . . . . .	52	TeX-auto-empty-regex-list (opción) . . . . .	20
reftex-select-jump (mandato) . . . . .	52	TeX-auto-full-regex-list (opción) . . . . .	21
		TeX-auto-label-regex-list (opción) . . . . .	20
		TeX-auto-minimal-regex-list (opción) . . . . .	20
		TeX-auto-parse-length (opción) . . . . .	20
		TeX-auto-regex-list (opción) . . . . .	20
		TeX-auto-save (opción) . . . . .	20
		TeX-auto-untabify (opción) . . . . .	17
		TeX-check-path (opción) . . . . .	80
		TeX-close-quote (opción) . . . . .	23

## Apéndice: Índice de nombres

---

TeX-command-buffer (mandato) . . . . .	79, 82	TeX-kill-job (mandato) . . . . .	82
TeX-command-default (opción) . . . . .	80	TeX-master (opción) . . . . .	65, 66
TeX-command-list (opción) . . . . .	77, 80	TeX-mode-hook (opción) . . . . .	12, 80
TeX-command-master (mandato) . . . . .	79, 82	TeX-next-error (mandato) . . . . .	77, 81
TeX-command-region (mandato) . . . . .	79, 82	TeX-normal-mode (mandato) . . . . .	20, 66
TeX-comment-paragraph (mandato) . . . . .	21, 22	TeX-one-master (opción) . . . . .	66
TeX-comment-region (mandato) . . . . .	21	TeX-open-quote (opción) . . . . .	23
TeX-complete-symbol (mandato) . . . . .	25, 26	TeX-outline-extra (opción) . . . . .	37
TeX-default-macro (opción) . . . . .	27	TeX-parse-self (opción) . . . . .	19
TeX-default-mode (opción) . . . . .	11, 12	TeX-recenter-output-buffer (mandato) . . . . .	82
TeX-display-help (opción) . . . . .	81	TeX-region (opción) . . . . .	79
TeX-electric-escape (opción) . . . . .	27	TeX-save-document (mandato) . . . . .	67
TeX-electric-macro (mandato) . . . . .	26, 27	TeX-save-query (opción) . . . . .	67
TeX-font-list (opción) . . . . .	31	TeX-toggle-debug-bad-boxes (opción) . . . . .	81
TeX-font (mandato) . . . . .	31	TeX-trailer-start (opción) . . . . .	80
TeX-force-default-mode (opción) . . . . .	11, 12	TeX-un-comment-region (mandato) . . . . .	22
TeX-goto-info-page (mandato) . . . . .	12	TeX-un-comment (mandato) . . . . .	22
TeX-header-end (opción) . . . . .	79		
TeX-home-buffer (mandato) . . . . .	67		
TeX-insert-braces (mandato) . . . . .	24		
TeX-insert-braces (opción) . . . . .	27		
TeX-insert-dolar (mandato) . . . . .	24		
TeX-insert-macro (mandato) . . . . .	26, 27		
TeX-insert-quote (mandato) . . . . .	23		
		<b>U</b>	
		up-list (mandato) . . . . .	25
		<b>W</b>	
		what-cursor-position (mandato) . . . . .	20