

La Universidad del Zulia  
Facultad Experimental de Ciencias  
Departamento de Matemática y Computación

# Preparación de Documentos

## Técnicos con L<sup>A</sup>T<sub>E</sub>X

José H. Nieto

Maracaibo, 1999



# Prefacio

$\text{\LaTeX}$  es un sistema para producir con el computador documentos técnicos de alta calidad. Se basa en el sistema  $\text{\TeX}$ , desarrollado por Donald Knuth. En la actualidad  $\text{\TeX}$  y  $\text{\LaTeX}$  son el estándar para escribir artículos y libros de matemática y física, y su uso se está extendiendo rápidamente a todas las ramas de la ciencia y la ingeniería. Las razones de su éxito radican tanto en la excelencia de los resultados que se obtienen como en su portabilidad y adaptabilidad.  $\text{\TeX}$  y  $\text{\LaTeX}$  se hayan disponibles en forma gratuita para todo tipo de computadoras y sistemas operativos. Los documentos elaborados en un PC, por ejemplo, pueden ser vistos sin modificaciones en una Mac o en una estación de trabajo UNIX y viceversa. Además  $\text{\LaTeX}$  es extensible y hay cientos de paquetes disponibles y bien documentados para realizar todo tipo de tareas especializadas.

Este manual contiene los conocimientos básicos necesarios para comenzar a producir documentos de calidad con  $\text{\LaTeX}$ . También cubre el uso de algunos paquetes importantes —en especial los de la *American Mathematical Society*— e incluye numerosas tablas de referencia, así como una bibliografía para quienes deseen profundizar en alguno de los temas tratados o explorar tópicos más avanzados.

José H. Nieto  
Maracaibo, abril 1999.



# Índice General

<b>Prefacio</b>	<b>iii</b>
<b>1 Conceptos básicos</b>	<b>1</b>
1.1 T <sub>E</sub> X, L <sup>A</sup> T <sub>E</sub> X y programas relacionados . . . . .	1
1.1.1 T <sub>E</sub> X y su familia . . . . .	1
1.1.2 L <sup>A</sup> T <sub>E</sub> X . . . . .	3
1.2 Argumentos a favor y en contra . . . . .	3
1.3 Cómo iniciarse . . . . .	6
1.4 Un primer documento en L <sup>A</sup> T <sub>E</sub> X . . . . .	6
1.5 Estructura de un archivo fuente . . . . .	10
1.6 Elementos de L <sup>A</sup> T <sub>E</sub> X . . . . .	11
1.6.1 Caracteres . . . . .	11
1.6.2 Comandos . . . . .	12
1.6.3 Comentarios . . . . .	14
1.6.4 Ambientes . . . . .	15
1.6.5 Declaraciones . . . . .	15
1.6.6 Expresiones matemáticas . . . . .	16
1.7 Paquetes . . . . .	18
1.8 Ejercicios . . . . .	19
<b>2 Una visión más detallada</b>	<b>21</b>
2.1 Composición de textos . . . . .	21
2.1.1 Acentos y símbolos especiales . . . . .	21
2.1.2 Signos de puntuación . . . . .	22
2.1.3 Unidades de longitud . . . . .	22
2.1.4 Saltos de línea y de página . . . . .	24
2.1.5 Tipos y tamaños . . . . .	24
2.1.6 Notas al pie de página . . . . .	26
2.1.7 Notas marginales . . . . .	27
2.1.8 Espaciado . . . . .	27
2.1.9 Ajuste de textos . . . . .	30
2.1.10 Citas y poesía . . . . .	30
2.1.11 Edición textual . . . . .	31

2.1.12	Referencias cruzadas . . . . .	31
2.1.13	División de palabras . . . . .	32
2.2	El paquete Babel . . . . .	33
2.3	Listas . . . . .	35
2.4	Tablas . . . . .	36
2.5	Objetos flotantes . . . . .	37
2.6	Referencias bibliográficas . . . . .	39
2.7	Estilos de página . . . . .	40
2.8	Clases de documento estándar . . . . .	41
2.9	Opciones de <code>\documentclass</code> . . . . .	44
2.10	Contadores . . . . .	45
2.11	Definición de nuevos comandos . . . . .	47
2.12	Definición de nuevos ambientes . . . . .	48
2.13	Ejercicios . . . . .	49
<b>3</b>	<b>Matemática</b>	<b>51</b>
3.1	Ecuaciones . . . . .	51
3.2	Texto y espacios . . . . .	53
3.3	Delimitadores . . . . .	54
3.4	Operadores y relaciones . . . . .	55
3.5	Tamaños y Tipos . . . . .	56
3.6	Disposición vertical de símbolos . . . . .	57
3.7	Matrices y construcciones similares . . . . .	60
3.8	Definiciones, teoremas y demostraciones . . . . .	61
3.9	Ejercicios . . . . .	62
<b>4</b>	<b>L<sup>A</sup>T<sub>E</sub>X avanzado</b>	<b>65</b>
4.1	Cajas . . . . .	65
4.2	Gráficas sencillas . . . . .	68
4.3	Imágenes y colores . . . . .	71
4.4	Documentos grandes . . . . .	74
4.5	Diseño de la página . . . . .	75
4.6	Índices alfabéticos . . . . .	77
4.7	Palabras finales . . . . .	78
<b>A</b>	<b>Soluciones a los ejercicios</b>	<b>79</b>
<b>B</b>	<b>Tablas de símbolos matemáticos</b>	<b>83</b>

# Índice de Tablas

2.1	Acentos y símbolos especiales . . . . .	21
2.2	Unidades de $\text{T}_{\text{E}}\text{X}$ . . . . .	23
2.3	Tipos en el modo de texto . . . . .	25
2.4	Tamaños de los tipos . . . . .	25
2.5	Permisos para objetos flotantes . . . . .	38
2.6	Estilos de página predefinidos en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . . . . .	41
3.1	Tipos en el modo matemáticos . . . . .	56
3.2	Tamaños en modo matemático . . . . .	57
3.3	Acentos matemáticos . . . . .	57
B.1	Relaciones binarias . . . . .	83
B.2	Operadores binarios . . . . .	84
B.3	Símbolos de tamaño variable . . . . .	84
B.4	Delimitadores . . . . .	84
B.5	Delimitadores grandes . . . . .	84
B.6	Flechas . . . . .	85
B.7	Letras griegas minúsculas . . . . .	85
B.8	Letras griegas mayúsculas . . . . .	85
B.9	Símbolos diversos . . . . .	86
B.10	Relaciones binarias de la AMS . . . . .	86
B.11	Flechas de la AMS . . . . .	87
B.12	Relaciones y flechas negadas de la AMS . . . . .	87
B.13	Operadores binarios de la AMS . . . . .	88
B.14	Delimitadores de la AMS . . . . .	88
B.15	Símbolos griegos y hebreos de la AMS . . . . .	88
B.16	Símbolos diversos de la AMS . . . . .	88





# Capítulo 1

## Conceptos básicos

Este Capítulo cubre los conceptos básicos asociados a la preparación de documentos con  $\LaTeX$ . Luego de una introducción histórica y de un breve análisis de la filosofía de  $\LaTeX$ , usted aprenderá a crear, visualizar e imprimir documentos sencillos.

### 1.1 $\TeX$ , $\LaTeX$ y programas relacionados

#### 1.1.1 $\TeX$ y su familia

$\TeX$  es un sistema de composición tipográfica para computadoras desarrollado por Donald E. Knuth con la intención—expresada en el prefacio de su libro *The  $\TeX$ book* [10]—de crear *libros hermosos, especialmente libros que contengan mucha matemática*.

$\TeX$  se pronuncia *tej*, como la palabra griega  $\tau\epsilon\chi$ , que significa arte y también técnica. El origen del nombre escogido por Knuth para su sistema es ya un indicio del grado de sutileza y refinamiento que le es inherente. Además la ‘E’ ligeramente desplazada nos recuerda que  $\TeX$  no es un simple procesador de textos sino que va mucho más allá. *Para quien meramente desee producir documentos aceptables, legibles pero no realmente hermosos—dice Knuth—un sistema más simple será en general suficiente*.

Al preparar un documento con  $\TeX$  le damos al computador instrucciones precisas para transformar el manuscrito en páginas de una calidad tipográfica comparable a la de las mejores imprentas. El significado profundo de estas palabras es difícil de comprender para quienes no están familiarizados con el arte y la técnica de los tipógrafos e impresores, pero es suficiente saber que  $\TeX$  realizará para nosotros un trabajo comparable al de un experto cajista, ocupándose de sutiles detalles de espaciado, ligaduras, división de palabras, juegos de caracteres y composición de fórmulas matemáticas de acuerdo a los más exigentes criterios técnicos y estéticos. Sin embargo, el trabajo que debe realizar el usuario no es mucho mayor que el de mecanografiar el manuscrito en una máquina de escribir corriente.

Durante el desarrollo de  $\text{T}_{\text{E}}\text{X}$  Knuth elaboró también un sistema para desarrollar programas en forma integrada con su documentación (*web/literate programming*, vea [12] para más detalles) y puso el fruto de su trabajo a disposición de todo el mundo. En consecuencia  $\text{T}_{\text{E}}\text{X}$  se puede ejecutar en los más diversos sistemas (Linux, UNIX, VMS, Os2, Mac, DOS, Windows).

Una vez aparecida la versión 3.0 Knuth declaró que ya no haría modificaciones sustanciales a  $\text{T}_{\text{E}}\text{X}$ , exceptuando la eliminación de eventuales *bugs* (que por otra parte son muy raros). Después de cada corrección agregó un dígito de  $\pi$  al número de versión, siendo la actual la 3.14159.

Para realizar su trabajo  $\text{T}_{\text{E}}\text{X}$  solamente necesita información *métrica* sobre los caracteres utilizados: ancho, altura sobre la línea base, profundidad y algunos otros parámetros. En cambio para imprimir o visualizar en una pantalla un documento ya procesado se necesita información detallada sobre la *forma* exacta de cada caracter (lo que se conoce como el *glifo* del caracter). Con este fin Knuth escribió otro programa llamado METAFONT, el cual permite diseñar juegos completos de caracteres para ser usados con  $\text{T}_{\text{E}}\text{X}$  generando automáticamente toda la información métrica y gráfica necesaria (la versión actual de este programa lleva el número 2.718, ¿le sugiere esto algo?)

Los juegos de caracteres se conocen en inglés con el nombre de *fonts*, palabra proveniente del francés y que alude al método tradicional usado para crear los tipos en las imprentas, con metal fundido. En español existe la palabra *fundición* para designar un juego de tipos de metal correspondientes a una misma familia. Sin embargo en los ambientes informáticos se ha generalizado el uso de la palabra *fuentes* como traducción de *font*, lo cual nos parece semántica y etimológicamente incorrecto. En esta obra usaremos la palabra *tipo* o bien el término inglés *font* para referirnos a este concepto.

METAFONT tiene muchas características avanzadas. Por ejemplo los juegos de caracteres que genera se pueden hacer depender de un cierto número de parámetros, lo cual permite obtener familias diferentes a partir de un mismo diseño básico, modificando simplemente esos parámetros. Por ejemplo el propio Knuth obtuvo, a partir de sus *Computer Modern fonts* (los juegos de caracteres que preparó para ser usados por defecto con  $\text{T}_{\text{E}}\text{X}$ ), una nueva familia llamada *Concrete fonts*, usada para componer su libro *Concrete Mathematics* [7]. Esto explica el significado de la partícula *meta* en el nombre de METAFONT, ya que este programa más que familias de caracteres permite generar metafamilias.

Ahora bien, el diseño de un juego de caracteres es una tarea bastante delicada, propia de especialistas. El usuario corriente de  $\text{T}_{\text{E}}\text{X}$  por lo general utiliza alguno de los juegos disponibles, siendo el más común el *Computer Modern*. De todas maneras, si quiere saber más sobre este tema vea el libro *The METAFONT book* [11]. También es posible utilizar juegos de caracteres POSTSCRIPT con  $\text{T}_{\text{E}}\text{X}$  y  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Vea al respecto [6] y sobre todo [8].

### 1.1.2 L<sup>A</sup>T<sub>E</sub>X

T<sub>E</sub>X es esencialmente un procesador de instrucciones, que ofrece al usuario una alta capacidad de programación. Evidentemente esto lo hace difícil y complicado, en la misma medida que poderoso. Por esta razón el propio Knuth escribió un conjunto de macros que llamó “plain T<sub>E</sub>X” y que facilita las tareas más corrientes en la preparación de documentos. Un paso más allá fue dado por Leslie Lamport al desarrollar L<sup>A</sup>T<sub>E</sub>X. Uno de los postulados de L<sup>A</sup>T<sub>E</sub>X es que el autor de un documento debe concentrarse en la *estructura lógica* del mismo y no en los detalles de apariencia y presentación, de los cuales debe encargarse el sistema, hasta donde sea posible. Por esta razón L<sup>A</sup>T<sub>E</sub>X provee facilidades para crear varios tipos usuales de documentos (artículos, reportes, libros, cartas, transparencias), generar índices alfabéticos y tablas de contenidos, resúmenes, notas a pie de página, bibliografías, etc. Según el propio Lamport *LaTeX representa un balance entre funcionalidad y facilidad de uso*. A partir de la aparición de la versión L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (la que se trata en este manual) y con la continua adición de paquetes para realizar tareas especiales de todo tipo (como por ejemplo componer diagramas conmutativos, grafos, fórmulas químicas, diagramas de Feynman, partituras musicales y hasta posiciones de ajedrez, Go y otros juegos) este sistema se ha convertido en el estándar para la preparación de documentos científicos y técnicos. La referencia estándar sobre L<sup>A</sup>T<sub>E</sub>X es *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, de Leslie Lamport [9].

En la actualidad L<sup>A</sup>T<sub>E</sub>X es mantenido y actualizado, aproximadamente cada seis meses, por el equipo L<sup>A</sup>T<sub>E</sub>X3, coordinado por Frank Mittelbach.

Mientras Lamport desarrollaba L<sup>A</sup>T<sub>E</sub>X, aproximadamente entre 1982 y 1986, Michael Spivak trabajaba por encargo de la *American Mathematical Society* (AMS) en otro conjunto de macros para T<sub>E</sub>X, el cual se llamó *A<sub>M</sub>S-T<sub>E</sub>X* (ver [19]). Alrededor de 1987 se hizo evidente que L<sup>A</sup>T<sub>E</sub>X y *A<sub>M</sub>S-T<sub>E</sub>X* tenían características complementarias, por ejemplo mientras L<sup>A</sup>T<sub>E</sub>X era superior en cuanto a la estructuración y diseño lógico de los textos *A<sub>M</sub>S-T<sub>E</sub>X* sobresalía en la composición de expresiones matemáticas. Esto llevó a la AMS a desarrollar un conjunto de paquetes conocidos colectivamente como *A<sub>M</sub>S-L<sup>A</sup>T<sub>E</sub>X*, los cuales incorporan a L<sup>A</sup>T<sub>E</sub>X la funcionalidad de *A<sub>M</sub>S-T<sub>E</sub>X*. La mayor parte de este trabajo fue realizada por Frank Mittelbach y Rainer Schöpf entre 1989 y 1990.

## 1.2 Argumentos a favor y en contra

Entre las ventajas de T<sub>E</sub>X y L<sup>A</sup>T<sub>E</sub>X pueden citarse la calidad profesional de los documentos elaborados y su portabilidad. Los archivos de entrada son simples archivos de texto que pueden prepararse con cualquier editor, llevarse de un sistema a otro y enviarse por correo electrónico como mensajes

corrientes.

Además  $\text{T}_{\text{E}}\text{X}$  y  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  son extensibles. Si no fueran suficientes los cientos de paquetes que están disponibles gratuitamente, ya depurados y documentados, usted mismo puede definir sus propias extensiones, para realizar las tareas especializadas que requiera. Toda la documentación y el código fuente necesario se puede obtener libremente.

Posiblemente el principal inconveniente de  $\text{T}_{\text{E}}\text{X}$  y  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  sea que su aprendizaje se asemeja más al de un lenguaje de programación que al de un procesador de textos o paquete de *desktop publishing* convencional. Cuantificar el tiempo requerido es difícil, ya que depende del grado de maestría que se desee alcanzar y de otras variables; el aprendizaje será más rápido, por ejemplo, si se tiene experiencia en programación o en lenguajes como HTML. Usted debe decidir si se justifica o no dedicarle a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  el tiempo necesario. En nuestra opinión el tiempo invertido se justifica ampliamente para quienes planean escribir numerosos artículos técnicos o bien libros o manuales. Además de la calidad de los resultados obtenidos, en el mediano plazo el ciclo de preparación de los documentos se reducirá y su productividad aumentará.

Entre las críticas que se les hace a  $\text{T}_{\text{E}}\text{X}$  y  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  la más frecuente es que no son WYSIWYG. Para aclarar este concepto recordemos que con los primeros procesadores de palabras para microcomputadoras uno escribía (y veía en la pantalla) una cantidad de caracteres de control y símbolos especiales que solamente al imprimir el documento se convertían en acentos, negritas, tabulaciones, etc. Si el efecto obtenido no era el deseado había que reiniciar el ciclo modificando el documento y volviendo a imprimir, con gran pérdida de tiempo. La mejora en la velocidad y la capacidad gráfica de los equipos hizo posible visualizar en pantalla el aspecto final del documento impreso, al principio como una función especial (*preview*) y luego en forma casi simultánea con el proceso de edición. Esta última característica es la que se denomina WYSIWYG (siglas en inglés de *what you see is what you get*, lo que usted ve es lo que usted obtiene) y resulta bastante cómoda, aunque para documentos largos tiende a retardar considerablemente el proceso de edición (como lo sabe cualquiera que haya tratado de escribir un libro con Word). Ahora bien,  $\text{T}_{\text{E}}\text{X}$  no es WYSIWYG en primer lugar porque *no es un editor* (de hecho se puede usar cualquier editor para escribir un archivo de entrada para  $\text{T}_{\text{E}}\text{X}$ ). En segundo lugar, en realidad existe *software* que proporciona una interfaz más o menos WYSIWYG para  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , como Scientific Workplace (comercial) y LyX (público).

Ahora bien, es importante señalar que la *edición visual* (es decir, el mover texto de un lado para otro y cambiar sus características visuales hasta lograr un aspecto que nos parezca apropiado) propiciada por los sistemas WYSIWYG conduce a elaborar documentos que pueden ser satisfactorios estéticamente, pero que en general están pobremente estructurados. En este sentido la filosofía de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  es completamente diferente, ya que pone el énfasis en la

estructura antes que en los aspectos visuales. La razón para esto es que la apariencia final de un documento depende de muchas variables que suelen ser desconocidas cuando se comienza a escribir (como por ejemplo el tamaño de la página, los márgenes, el estilo y los tipos con que se va a imprimir la versión final) y la mayor parte del formato visual se pierde cuando esas variables deben ser modificadas. Lo mismo ocurre si decidimos realizar un cambio de notación sobre la marcha. Supongamos por ejemplo que estamos escribiendo un documento en el cual usamos la notación  $(u, v)$  para denotar el producto interno de dos vectores, y una vez finalizado nos damos cuenta de que la misma notación se usa también para los pares ordenados, lo cual podría ocasionar confusión. Decidimos entonces cambiar la notación para el producto interno a  $\langle u, v \rangle$ . Para hacer esto con un editor WYSIWYG, si el documento es extenso, tendremos que trabajar bastante ya que la sustitución de paréntesis por corchetes angulares depende del contexto y no puede automatizarse. Un conocedor de  $\text{\LaTeX}$  en cambio hubiese escrito algo así como `\p{u}{v}` cada vez que tuviese que referirse al producto interno de  $u$  y  $v$ , dejando los detalles concretos de la notación a ser utilizada a la definición del comando `\p` en el preámbulo del documento. Procediendo de esta forma, si se decide cambiar la notación para el producto interno solamente tendremos que modificar esa definición. Puede aún objetarse que para escribir `\p{u}{v}` hay que teclear mucho. Si realmente lo cree así haga un sencillo experimento: compare el tiempo que tarda en escribirla con el que lleva componer la expresión  $\langle u, v \rangle$  seleccionando con el ratón los símbolos especiales ‘ $\langle$ ’ y ‘ $\rangle$ ’ de un menú desplegable.

En cuanto a la lentitud del ciclo edición-compilación-visualización, que sin duda era una realidad años atrás, ya no lo es gracias a la velocidad de los equipos actuales. Con un sistema operativo multitareas es posible tener varias ventanas abiertas simultáneamente y utilizar una para editar, otra para compilar y una tercera para visualizar los resultados, con una rapidez y comodidad comparable o superior a la de cualquier sistema WYSIWYG convencional.

Hace algunos años también representaban un problema los grandes recursos computacionales necesarios para correr  $\text{\TeX}$  (memoria, espacio en disco y velocidad del procesador). Pero esto ya no es una limitación con la tecnología actual. Además  $\text{\TeX}$  consume recursos solamente mientras está procesando un documento, y no durante todo el ciclo de desarrollo, como lo hacen los procesadores de texto WYSIWYG.

Ahora bien, indudablemente que hay ciertas aplicaciones para las cuales programas como PageMaker, Quark-XPRESS o incluso Word pueden ser más apropiados que  $\text{\LaTeX}$ . El ejemplo típico son los anuncios publicitarios, los cuales no contienen expresiones matemáticas pero sí muchas imágenes en diversos formatos gráficos y texto compuesto con diversos *fonts*. Asimismo si necesita elaborar portadas y en general páginas en las cuales el diseño visual sea lo más importante, deberá recurrir al software especializado para

esas tareas.

### 1.3 Cómo iniciarse

Además de este manual usted necesitará obviamente acceso a un sistema de computación que cuente con una instalación razonablemente completa y actualizada de  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  y si es posible  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . Los sistemas tipo UNIX por lo general tienen  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  instalado, y usted no tendrá que preocuparse por este aspecto. De todas maneras todo lo necesario se puede bajar por `ftp` desde la red de servidores de archivos CTAN (*Comprehensive  $\text{T}_{\text{E}}\text{X}$  Archive Network*). Los sitios principales son `ftp.dante.de` en Alemania, `ftp.tex.ac.uk` en el Reino Unido y `ftp.shsu.edu` en los Estados Unidos de América. También se puede acceder a través del World Wide Web, en las direcciones `http://tug2.cs.umb.edu/CTAN/` y `http://www.tug.org/`.

Ahora bien, la instalación y configuración de un sistema completo no es trivial y requiere ciertos conocimientos. Lea la documentación y si es posible busque la ayuda de un experto. Para Windows lo más sencillo y recomendable es obtener la distribución MiKTeX [17], la cual viene en un solo archivo comprimido y ejecutable, que se descomprime e instala solo.

El inconveniente de bajar todo por Internet es el tiempo requerido (son muchos megabytes). Una alternativa es conseguir una distribución en CD, como la que distribuye TUG ( $\text{T}_{\text{E}}\text{X}$  Users Group, [22]).

Además de  $\text{T}_{\text{E}}\text{X}$   $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  y sus programas relacionados necesitará también un editor de texto. En los sistemas UNIX se puede usar por ejemplo `vi`, `emacs` o el sencillo `joe`. En ambiente Windows se puede usar NotePad, para archivos pequeños, o bien WordPad o Word, pero teniendo cuidado de grabar los resultados como archivos de texto. Sin embargo si piensa utilizar  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  intensivamente bajo Windows, es mejor conseguir un editor como TextPad [21] o WinEdit [23], los cuales se distribuyen como *shareware* y pueden ser bajados por Internet en versiones de demostración. Estos editores poseen facilidades de búsqueda y reemplazo avanzadas, permiten ejecutar programas (como  $\text{T}_{\text{E}}\text{X}$  y  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ) que procesen el archivo editado y son ampliamente configurables.

### 1.4 Un primer documento en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

Supongamos que usted ya tiene acceso a un computador con  $\text{T}_{\text{E}}\text{X}$  y  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  instalados y está ansioso por poner manos a la obra y crear su primer documento. Para ello el primer paso consiste en crear un *archivo de entrada* (también llamado archivo *fuentes*), que contenga el texto y las instrucciones necesarias para que  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  lo procese. Este archivo debe llevar la extensión `tex` y no es más que un archivo de texto ASCII, que puede ser creado con

cualquier editor. Inicie entonces su editor favorito y cree un archivo de texto con el nombre `prueba.tex` que contenga lo siguiente:

```
\documentclass{article}
\begin{document}
Esto es una
prueba de \LaTeXe.
\end{document}
```

No deje de teclear los caracteres ‘\’ (diagonal invertida o *backslash*), los cuales actúan como caracteres de escape para que T<sub>E</sub>X interprete la palabra siguiente como una instrucción y no como parte del texto. Ponga también atención a las mayúsculas y minúsculas alternadas en L<sup>A</sup>T<sub>E</sub>X<sub>e</sub>. En cambio el número exacto de espacios entre palabras no es importante (el ejemplo tiene un aspecto irregular a propósito).

El paso siguiente es procesar este archivo con L<sup>A</sup>T<sub>E</sub>X, operación a la que llamaremos *texear*. Por ejemplo bajo Linux ejecutamos el comando:

```
latex prueba
```

(en las versiones para Windows este comando debe darse por lo general en una ventana MSDOS). Observe que no es necesario escribir la extensión, L<sup>A</sup>T<sub>E</sub>X asume por defecto que la misma es `tex`. Una vez ejecutado el comando aparecerán varios mensajes en la pantalla. Los primeros nos informarán sobre las versiones de T<sub>E</sub>X y de L<sup>A</sup>T<sub>E</sub>X que estamos ejecutando, y posiblemente sobre algunas otras cosas como la versión del paquete Babel y los patrones cargados para dividir palabras (*hyphenation patterns*). Si todo va bien los mensajes finales dirán:

```
Output written on prueba.dvi (1 page, 380 bytes).
```

```
Transcript written on prueba.log.
```

La primera línea nos informa que la “salida” (*output*), es decir el resultado del trabajo realizado por T<sub>E</sub>X, se ha escrito en el archivo `prueba.dvi`. La extensión `dvi` significa *device independent* (independiente del dispositivo) ya que la información contenida en este archivo permite ver o imprimir el documento en una gran variedad de dispositivos. La segunda línea nos dice que la transcripción de la sesión (que en este manual llamaremos *bitácora*) ha sido escrita en `prueba.log`. Este archivo contiene todos los mensajes que aparecieron en la pantalla al texear (y que posiblemente pasaron muy rápido y no tuvimos tiempo de leer) más algunos datos adicionales. Otro archivo generado es `prueba.aux`, el cual contiene información auxiliar utilizada por T<sub>E</sub>X.

Si T<sub>E</sub>X encuentra algún error en nuestro archivo fuente entonces se detendrá con un mensaje y esperará que le indiquemos como proceder. Uno de los errores más comunes se produce si tipeamos mal el nombre del archivo

fuente o si corremos L<sup>A</sup>T<sub>E</sub>X desde un directorio diferente al que contiene a nuestro archivo, sin darle información sobre donde debe buscar. Por ejemplo si damos la orden

```
latex prueba
```

obtendremos el siguiente mensaje:

```
! I can't find file 'prueba.tex'.
<*> &latex prueba
```

```
Please type another input file name:
```

Como ve, tenemos la oportunidad de enmendar el error cometido e ingresar el nombre del archivo correctamente (alternativamente se puede abortar la ejecución tecleando ctrl-Z en Windows o ctrl-D en Unix).

Otro error común consiste en escribir mal, en el archivo de entrada, el nombre de algún comando. Supongamos por ejemplo que en la primera línea del archivo `prueba.tex` escribimos erróneamente `\documentclas{article}` (es decir, nos olvidamos de una 's'). Entonces obtendremos el mensaje

```
! Undefined control sequence.
1.1 \documentclas
      {article}
?
```

el cual nos dice que el comando (*control sequence*) '`\documentclas`' no está definido. En las dos líneas siguientes T<sub>E</sub>X nos dice lo que estaba leyendo cuando encontró el error y lo que debe leer a continuación. El signo de interrogación significa que está esperando que le digamos qué hacer. Si por nuestra parte le respondemos con un signo de interrogación entonces veremos las alternativas a nuestra disposición:

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
```

Probablemente la más recomendable para un principiante sea la H, pues así se obtiene información adicional sobre el error encontrado y tal vez alguna sugerencia para solucionarlo.

La opción X simplemente finaliza la sesión: T<sub>E</sub>X generará un archivo de salida con las páginas ya procesadas antes de encontrar el error.

La opción E finaliza la sesión y además, si hay un buen editor disponible y T<sub>E</sub>X ha sido adecuadamente configurado, abrirá el archivo fuente con el



editor y se ubicará en la línea donde se encontró el error para permitirnos corregirlo.

La opción `I` nos permite introducir caracteres por por el teclado. En el ejemplo anterior podríamos teclear `I\documentclass` para corregir el error, luego de lo cual T<sub>E</sub>X finalizará normalmente creando un archivo de salida. Sin embargo el error permanece en el archivo fuente, y si lo queremos corregir definitivamente tendremos que editarlo.

Si simplemente oprimimos la tecla `<return>`, T<sub>E</sub>X ignorará el error y seguirá adelante. Esto puede ser útil para un usuario experimentado que desee recoger información sobre varios errores antes de editar el archivo fuente para corregirlos. Sin embargo también puede causar gran confusión, ya que posiblemente obtendremos luego numerosos errores que no son sino las consecuencias del primero (vea lo que pasa en el ejemplo anterior si responde con `<return>` a cada mensaje de error).

La opción `S` hace que T<sub>E</sub>X ignore los errores siguientes sin detenerse; `R` y `Q` son variantes más fuertes de `S`. Ninguna de estas opciones es recomendable para un principiante.

Finalmente, una entrada numérica  $n$  hace que T<sub>E</sub>X ignore los  $n$  *tokens* siguientes en el archivo de entrada (cada caracter del texto y cada secuencia de control se cuentan como un *token*).

Ahora bien, una vez corregidos todos los errores y obtenido el archivo de salida, naturalmente querremos verlo en pantalla y posiblemente imprimirlo. Para esto necesitamos un programa (*driver*) que traduzca el formato `dvi` a un formato apropiado para el sistema de video o la impresora que estemos utilizando. Por ejemplo bajo Linux podemos iniciar el sistema X Window (si es que ya no estamos en él) y ejecutar el comando `xdvi prueba`. (no es necesario escribir la extensión pues `xdvi` asume que ésta será `dvi`). Bajo Windows utilizaremos el programa visualizador de archivos `dvi` que tengamos instalado (por ejemplo YaP (el visualizador de MiK<sub>T</sub>E<sub>X</sub> [17]), DVIScope [18] o `dviwin` [2]), Inmediatamente se abrirá una ventana en la cual veremos nuestro documento, en este caso una página con una sola línea de texto:

Esto es una prueba de L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

Observemos que T<sub>E</sub>X hizo caso omiso de los espacios extra que dejamos entre algunas palabras. Tampoco tuvo en cuenta que nuestro texto estaba escrito en dos líneas. En general, dos o más espacios consecutivos son considerados igual que uno solo y el caracter de fin de línea se considera equivalente a un espacio. Otro detalle importante es que las palabras que comienzan con el carácter `\` son interpretadas evidentemente como instrucciones. Por ejemplo `\LaTeXe` es un comando que indica a T<sub>E</sub>X que componga el logo L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

Los programas visualizadores de archivos `dvi` pueden también ejecutarse sin suministrarles nombre de archivo alguno, abriendo luego el que

deseemos. Seguramente en su sistema hay varios archivos `dvi` disponibles para ser examinados, con información sobre  $\text{\LaTeX}$ : busque en el directorio `texmf/doc`. Si su sistema es nuevo, es probable que vea correr por la pantalla varios mensajes informativos sobre la creación de *fonts*. Si esto ocurre no se preocupe, significa que los *fonts* necesarios para visualizar un documento no estaban presentes en su sistema y que METAFONT y programas relacionados los están creando automáticamente.

Si queremos imprimir el documento en una impresora laser debemos convertir el archivo `prueba.dvi` al formato POSTSCRIPT con el programa `dvips`. Para ello escribiremos en la línea de comandos:

```
dvips -o prueba.ps prueba.dvi
```

Esto crea un archivo `prueba.ps` que puede ser enviado a la impresora. Por ejemplo en Linux, si nuestra impresora se llama `lj1`, escribiremos:

```
lpr -h -Plj1 prueba.ps
```

El archivo `prueba.ps` también puede visualizarse con un programa llamado GHOSTVIEW, el cual es una interfaz gráfica de GHOSTSCRIPT [4]. Estos útiles programas nos permiten usar una gran variedad de impresoras, y también seleccionar las páginas del documento que queremos imprimir.

Bajo Windows, imprimir es aun más fácil, ya que los programas visualizadores por lo general incluyen esta función y pueden hacerlo en cualquier impresora que esté instalada.

## 1.5 Estructura de un archivo fuente

Todo archivo fuente para  $\text{\LaTeX} 2_{\epsilon}$  debe tener la estructura que se ilustra en la figura siguiente:

<code>\documentclass{clase}</code>
<code>:</code>
<code>  <i>preámbulo</i></code>
<code>:</code>
<code>\begin{document}</code>
<code>:</code>
<code>  <i> cuerpo del documento</i></code>
<code>:</code>
<code>\end{document}</code>

La parte del documento fuente comprendida entre `\documentclass` y `\begin{document}` se llama *preámbulo* y se utiliza para declarar los paquetes que se van a utilizar, dar valores a ciertos parámetros, definir nuevos comandos, etc.

Lo que sigue a `\begin{document}` es el *cuerpo del documento*, que finaliza con el comando `\end{document}`. LaTeX ignorará cualquier cosa que se escriba después de este comando.

Cuando procesa un archivo de entrada, lo primero que necesita saber L<sup>A</sup>T<sub>E</sub>X es el tipo de documento que el autor quiere crear. Esta información la suministra el argumento obligatorio *clase* de la instrucción `\documentclass`:

`\documentclass[opciones]{clase}`

Las clases de documento estándar de L<sup>A</sup>T<sub>E</sub>X son `article` (para escribir artículos), `report` (reportes), `book` (libros), `slide` (transparencias) y `letter` (cartas). Pueden definirse otras clases para propósitos especiales; por ejemplo muchas revistas científicas tienen su propia versión de la clase `article`, adaptada al estilo de la publicación, para que sea utilizada por los autores.

El argumento opcional *opciones* permite especificar algunas características de la clase de documento elegida. Algunas opciones comunes son `11pt` y `12pt` (para usar tipos de tamaño 11 y 12 puntos respectivamente, en vez del valor por defecto que es 10 puntos) y `twocolumn` para componer el documento a dos columnas. Pueden incluirse varias opciones simultáneamente, separadas con comas. Por ejemplo, un archivo de entrada que comience con el comando:

```
documentclass[11pt]{article}
```

le indica a L<sup>A</sup>T<sub>E</sub>X que componga el documento como un *artículo*, utilizando tipos del cuerpo 11, para ser impreso en papel tamaño carta (el valor por defecto). En el siguiente capítulo daremos una información más completa sobre clases y opciones.

## 1.6 Elementos de L<sup>A</sup>T<sub>E</sub>X

### 1.6.1 Caracteres

Los caracteres que pueden aparecer en un documento L<sup>A</sup>T<sub>E</sub>X incluyen las letras mayúsculas y minúsculas, los dígitos del 0 al 9 y los 16 caracteres:

. : ; , ? ! ‘ ’ ( ) [ ] - / \* @

Los *caracteres especiales*

\$ & % # \_ { } ~ ^ \

solamente se usan en comandos. Si los usamos en un texto corriente es muy probable que generen errores, no se impriman o fuercen a  $\text{\LaTeX}$  a hacer cosas que no deseamos.

Los caracteres

+ = | < >

se usan principalmente en expresiones matemáticas, aunque + y = pueden también utilizarse en texto ordinario. El carácter " (comillas dobles) es raramente usado, ya que para entrecomillar un texto lo correcto es usar comillas diferentes al comienzo y al final. En  $\text{\LaTeX}$  se usa el acento grave para abrir comillas y el apóstrofo para cerrarlas, Para obtener comillas dobles se duplican los acentos y los apóstrofos. En otras palabras si escribimos ‘simple’ obtendremos ‘simple’, y si escribimos ‘‘doble’’ obtendremos “doble”.

Hay también caracteres invisibles, como el espacio en blanco que se produce con la barra espaciadora del teclado, el carácter de fin de línea que se introduce con la tecla `<return>` o `<enter>`, y el carácter de tabulación. Todos ellos son tratados por  $\text{\TeX}$  como signos de espacio. Varios espacios consecutivos se tratan igual que uno solo (excepto dentro de ciertos ambientes especiales) y generalmente un espacio en blanco al comienzo de una línea se ignora.

Una o más líneas consecutivas completamente en blanco se interpretan como el final de un párrafo. El texto que se escriba a continuación aparecerá sangrado y en una nueva línea.

### 1.6.2 Comandos

Un archivo fuente para  $\text{\LaTeX}$  se compone básicamente de *texto* (que especifica el *contenido* del documento) y *comandos*, que especifican como debe ser compuesto ese contenido y además nos permiten ordenarle a  $\text{\LaTeX}$  que realice una multitud de tareas útiles para nosotros. El nombre de un comando en  $\text{\LaTeX}$  puede consistir en:

- un sólo carácter especial.
- un \ seguido de un carácter que no sea una letra.
- un \ seguido por una o más letras.

En los nombres de comandos  $\text{\LaTeX}$  distingue entre letras mayúsculas y minúsculas (aunque lo más común es usar sólo minúsculas). Ciertos comandos tienen una variante o forma especial que se obtiene agregando \* al final del nombre.

Algunos comandos aceptan argumentos, los cuales pueden ser obligatorios u opcionales. Los primeros se colocan a continuación del nombre del comando encerrados entre llaves, y los segundos entre corchetes.

Un ejemplo de comando del primer tipo es `$`, que se usa para entrar y salir del modo matemático en medio de un párrafo.

Como ejemplo de comandos del segundo tipo consideremos los siguientes, que nos permiten imprimir caracteres especiales (en este ejemplo y en los que sigan escribiremos el código a la izquierda y el resultado que se obtiene a la derecha):

```
\$ \& \% \# \_ \{ \}          $ & % # - { }
```

Otro ejemplo es el comando `\` el cual permite finalizar una línea pero sin iniciar un nuevo párrafo.

```
Luego de la coma,\ seguímos          Luego de la coma,
en una nueva línea.                  seguimos en una nueva línea.
```

Otros comandos interesantes son `\'`, `\'`, `\~` y `\"`, los cuales nos permiten poner los acentos agudos y graves, tildes y diéresis indispensables en castellano y otros idiomas (en la Tabla 2.1 del Capítulo siguiente se halla un muestrario completo de caracteres especiales y acentuados).

```
Lleg\ 'o la cig\ "ue\ ~na.          Llegó la cigüeña.
```

La letra ‘i’ requiere un tratamiento especial debido a su punto. El comando `\'i` genera *í* que si se mira atentamente tiene tanto el acento como el punto. Pero el comando `\i` genera *i* (¡una *i* sin punto!), por lo tanto `\'i` es la solución para generar una *í* acentuada. Las llaves son necesarias para que el espacio siguiente sea tomado en cuenta. Si todo esto le parece muy complicado no se desanime: con el paquete `babel` las vocales acentuadas y la *eñe* se pueden generar escribiendo simplemente `'a`, `'e`, `'i`, `'o`, `'u` y `'n` o `\~n`. Para ello debe incluir en el preámbulo del documento la línea

```
\usepackage[spanish,activeacute]{babel}
```

En los ejemplos que aparezcan de aquí en adelante supondremos siempre que se está haciendo esto. Si se dispone de un teclado en español y se van a escribir muchos documentos en este idioma (o en otros idiomas europeos diferentes del inglés) se puede cambiar la codificación de entrada con el paquete `inputenc`. Para ello en los sistemas UNIX se debe incluir en el preámbulo del documento la instrucción:

```
\usepackage[latin1]{inputenc}
```

En cambio bajo Windows se debe usar:

```
\usepackage[ansinew]{inputenc}
```

Con esto se pueden introducir por teclado de la manera acostumbrada (y ver en el editor) los caracteres acentuados y símbolos especiales del castellano.

Un comando del tercer tipo es `\TeX`, el cual genera el logo de  $\TeX$ . Esta es en realidad una macroinstrucción que se expande en un conjunto de comandos de bajo nivel, los cuales especifican exactamente cómo debe desplazarse la ‘E’. Los espacios en blanco que van tras este tipo de comandos son ignorados. Para introducir un espacio en blanco en estos casos debemos escribir a continuación del comando o bien `{ }` y un espacio, o bien un comando de espaciado. Con `{ }` se crea una separación lógica que fuerza a  $\LaTeX$  a tomar en cuenta los espacios siguientes. La alternativa es introducir un espacio explícitamente mediante el comando `\_` (donde el signo `_` representa un espacio en blanco). Observe el siguiente ejemplo:

```
\TeX{} se describe detalladamente      \TeX se describe detalladamente en
en ‘‘The \TeX book’’.                  “The  $\TeX$ book”.
```

Si se usa el paquete `babel` con la opción `activeacute` se puede presentar un problema similar con el apóstrofo, el cual se convierte en un carácter activo para generar letras acentuadas. Si en alguna ocasión no queremos que cumpla esta función y va seguido de una vocal o una n, debemos separarlo con un par de llaves (o con un espacio explícito) de la letra siguiente. Vea el siguiente ejemplo.

```
La ‘o’ es una vocal\\                 La ‘oés una vocal
La ‘o’{} es una vocal\\              La ‘o’ es una vocal
Jos’e ‘Pepe’ Nieves.\\               José ‘PepeÑieves.
Jos’e ‘Pepe’\ Nieves.                José ‘Pepe’ Nieves.
```

### 1.6.3 Comentarios

Si  $\LaTeX$  encuentra el carácter `%` (símbolo de porcentaje) en un archivo fuente entonces lo ignora, así como al resto de la línea en la cual se halle, incluido el carácter de fin de línea (esta regla tiene algunas excepciones, por ejemplo si `%` va precedido de `\` entonces se imprime como `%`). Esto permite introducir comentarios en el archivo fuente que no aparecerán en la versión impresa. También resulta útil para partir una palabra muy larga entre dos líneas sin introducir un espacio entre las dos partes (los nombres de comando, sin embargo, no se pueden partir). Analice el siguiente ejemplo:

```
% Ejemplo de comentario              Si quiere imprimir un símbolo de por-
Si quiere imprimir un s’imb%         centaje, precédalo de un “backslash”.
olo de porcentaje, prec’edalo        Esto es 100% seguro.
de un ‘‘backslash’’. Esto es
100%\% seguro.
```

### 1.6.4 Ambientes

Para componer textos con un propósito especial L<sup>A</sup>T<sub>E</sub>X define varios tipos de *ambientes* (*environments*, en inglés). La forma general de un ambiente es la siguiente:

<code>\begin{nombre}</code>
⋮
<i>contenido</i>
⋮
<code>\end{nombre}</code>

donde *nombre* es el nombre del ambiente. Los ambientes se pueden anidar, pero dos ambientes diferentes deben ser disjuntos o estar uno contenido en el otro.

Ya hemos visto un ejemplo de ambiente: `document`, el cual encierra el contenido de todo el documento. En los capítulos siguientes estudiaremos otros ambientes que permiten hacer listas y tablas, estructurar y dar formato a teoremas y demostraciones, etc.

También veremos como definir nuestros propios ambientes.

### 1.6.5 Declaraciones

Una *declaración* es un comando que no tiene argumentos pero afecta al texto que le sigue, hasta un cierto límite. La porción de texto afectada por una declaración se conoce como su *ámbito*. El ámbito de una declaración comienza inmediatamente después de la misma y finaliza al encontrar el primer `\end` o la primer `}` que se emparejen con un `\begin` o con una `{` anteriores a la declaración.

Por ejemplo, en los trabajos mecanografiados se acostumbra subrayar las palabras o el texto que se desea resaltar. En los libros impresos en cambio se utiliza un tipo de letra diferente, generalmente la llamada *bastardilla* o *itálica*. En L<sup>A</sup>T<sub>E</sub>X se usa la declaración `\em` para resaltar. Otra declaración, `\bfseries`, permite escribir en letra **negrita** (también llamada *negrilla*, y en inglés *boldface*).

Texto `{\em destacado}`, normal\\  
y `{\bfseries en negritas}`.

Texto *destacado*, normal  
y **en negritas**.

En general cada declaración tiene un comando equivalente al cual se le proporciona el texto a ser afectado como argumento. Por ejemplo la declaración `\em` equivale al comando `\emph`. Veamos su uso en el siguiente ejemplo:

<pre>{\em Si se destaca \emph{algo} en un texto ya resaltado, entonces se vuelve a utilizar letra \emph{romana} para destacarlo.}</pre>	<p><i>Si se destaca algo en un texto ya resaltado, entonces se vuelve a utilizar letra romana para destacarlo.</i></p>
---	--

### 1.6.6 Expresiones matemáticas

La composición de expresiones matemáticas es uno de los puntos fuertes de  $\text{T}_{\text{E}}\text{X}$ . Para ello utiliza un modo especial, diferente al modo de texto (en el cual se encuentra  $\text{TeX}$  cuando comenzamos a escribir un documento). Para escribir algo de matemática en medio de un párrafo de texto corriente debemos entrar al *modo matemático* escribiendo el carácter especial  $\$$ . Para continuar escribiendo texto normal debemos salir de este modo escribiendo nuevamente  $\$$ . Un primer ejemplo:

Si  $\$2x+1=7\$$  entonces  $\$x=3\$$ .                      Si  $2x + 1 = 7$  entonces  $x = 3$ .

Observe que en el modo matemático las letras se imprimen en un estilo inclinado. Por lo tanto si en medio de un texto hacemos referencia a una variable  $x$  debemos escribir  $\$x\$$ , para ser consistentes. Otro punto importante es que  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  tiene sus propios criterios sobre el espaciado en las fórmulas matemáticas, y no tomará en cuenta los espacios en blanco que nosotros escribamos. Por ejemplo:

<pre>\\$x+y=1\\$ se ve igual que \\$x + y = 1\\$.</pre>	<p><math>x + y = 1</math> se ve igual que <math>x + y = 1</math>.</p>
---	---

En el modo matemático se pueden usar los operadores  $_$  para generar subíndices y  $^$  para denotar superíndices (exponentes). Por ejemplo:

<pre>Sea \\$x_0=2\\$ y para \\$n&gt;0\\$ definamos \\$x_n=x_{n-1}^2 +1\\$.</pre>	<p>Sea <math>x_0 = 2</math> y para <math>n &gt; 0</math> definamos <math>x_n = x_{n-1}^2 + 1</math>.</p>
--	--

Observe que si el subíndice o superíndice se compone de más de un símbolo, como  $n - 1$  en el ejemplo anterior, debe encerrarse entre llaves. También es posible utilizar simultáneamente ambos tipos de índice.

Las *letras griegas* minúsculas se introducen como  $\backslash\alpha$ ,  $\backslash\beta$ , ... y las mayúsculas que sean diferentes de la correspondiente romana como  $\backslash\Gamma$ ,  $\backslash\Delta$  ... (vea las Tablas B.7 y B.8 en el Apéndice B para una lista completa).

<pre>El n'umero \\$\pi\\$ es aproximadamente 3.14159, mientras que el s'imbolo \\$Pi\\$ se usa para representar productos.</pre>	<p>El número <math>\pi</math> es aproximadamente 3.14159, mientras que el símbolo <math>\Pi</math> se usa para representar productos.</p>
--	---



Hay una enorme variedad de símbolos matemáticos para los cuales hay comandos predefinidos en L<sup>A</sup>T<sub>E</sub>X. Por ejemplo `\leq` y `\geq` generan los símbolos  $\leq$  y  $\geq$  respectivamente. El signo de sumatoria  $\sum$  se genera con `\sum`, y el de integral  $\int$  con `\int`. Los límites de sumación e integración se ponen con los operadores de subíndice y superíndice. El signo  $\infty$  se genera con `\infty`. Veamos algunos ejemplos:

Según Euler  $e^{i\pi} + 1 = 0$ .  
Además  $\gamma = \lim_{n \rightarrow \infty} (\sum_{k=1}^n 1/k - \ln n)$ .

Según Euler  $e^{i\pi} + 1 = 0$ . Además  $\gamma = \lim_{n \rightarrow \infty} (\sum_{k=1}^n 1/k - \ln n)$ .

Los nombres de ciertas funciones y operadores, como  $\ln$  y  $\lim$  en el ejemplo anterior, deben ir en letra redonda o romanilla, es decir en estilo vertical, no inclinado. Por eso es que usamos los comandos `\ln` y `\lim`. Si en cambio hubiésemos escrito `ln n`, el resultado hubiera sido *lnn*.

Cuando las fórmulas son muy complejas o largas no resulta apropiado escribirlas mezcladas con el texto. En estos casos se utiliza el modo matemático *desplegado*, el cual se inicia con `\[` y finaliza con `\]` (o alternativamente con `\begin{displaymath}` y `\end{displaymath}`). En este modo las expresiones se componen de manera más elegante, aprovechando la disponibilidad de espacio. Por ejemplo:

`\[ \gamma = \lim_{n \to \infty} ( \sum_{k=1}^n \frac{1}{k} - \ln n ) \]`

$$\gamma = \lim_{n \rightarrow \infty} \left( \sum_{k=1}^n \frac{1}{k} - \ln n \right)$$

Los comandos `\left` y `\right` hacen que los delimitadores que les siguen, en este caso paréntesis, adopten un tamaño apropiado para las expresiones que encierran. Observe además el uso del comando `\frac`, con dos argumentos, para generar la fracción  $\frac{1}{k}$ . Como vemos, esta instrucción puede emplearse también dentro de un párrafo, aunque en estos casos se prefiere la forma más simple  $1/k$ . Por supuesto que los argumentos de `\frac` pueden ser expresiones complejas, incluso otras fracciones, por ejemplo:

`\[ \frac{e^{e^x} + \frac{1 + \frac{x}{x-1}}{x^3+x+2}}{\frac{2x+3}{x^2+x+1} + \frac{x+1}{1 + \frac{x^2+1}{x-1}}} \]`

$$\frac{e^{e^x} + \frac{1 + \frac{x}{x-1}}{x^3+x+2}}{\frac{2x+3}{x^2+x+1} + \frac{x+1}{1 + \frac{x^2+1}{x-1}}}$$

En el capítulo 3 estudiaremos en mayor profundidad la composición de fórmulas matemáticas con L<sup>A</sup>T<sub>E</sub>X.

## 1.7 Paquetes

Como ya hemos dicho  $\text{\LaTeX}$  es extensible, y hay gran número de extensiones ya elaboradas que se conocen con el nombre de *paquetes* (en inglés *packages*). Para utilizar un paquete hay que incluir en el preámbulo el comando

`\usepackage[opciones]{paquete}`

donde *paquete* es el nombre del paquete y *opciones* es una lista de palabras clave que permiten controlar algunas características. Muchos paquetes son estándar y vienen incluidos en las distribuciones de  $\text{\LaTeX}$ . Otros se pueden obtener a través de CTAN.

En este manual solamente describiremos el uso de algunos paquetes (los que consideramos de uso más general). La fuente principal de información sobre este tema (además de la documentación que viene incluida con los mismos paquetes) es *The  $\text{\LaTeX}$  Companion* [5], que contiene descripciones de cientos de ellos.

## 1.8 Ejercicios

**Ejercicio 1.1.** Escriba lo necesario para obtener la frase siguiente:

En cada instalación de L<sup>A</sup>T<sub>E</sub>X debería haber una *Guía Local* para especificar las características particulares del sistema.

**Ejercicio 1.2.** Escriba lo necesario para obtener la frase siguiente:

Las acciones de Wilson & Sons llegaron a 20\$, lo cual representa un aumento del 10% respecto al cierre de la víspera.

**Ejercicio 1.3.** Escriba lo necesario para obtener las siguientes palabras francesas: château, élève, naïf.

**Ejercicio 1.4.** Componga la expresión  $ax + by \leq \sqrt{a^2 + b^2} \sqrt{x^2 + y^2}$ .

**Ejercicio 1.5.** Componga la expresión  $F_{n+1} = F_n + F_{n-1}$ .

**Ejercicio 1.6.** Componga la expresión:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$



## Capítulo 2

# Una visión más detallada

En este capítulo estudiaremos los aspectos más importantes de  $\text{\LaTeX}$  con excepción de lo relativo al modo matemático, al cual—dada su complejidad e importancia—le dedicaremos íntegramente el capítulo siguiente.

### 2.1 Composición de textos

En esta sección hemos reunido varios comandos y ambientes que son importantes para la composición de textos.

#### 2.1.1 Acentos y símbolos especiales

$\text{\LaTeX}$  permite el uso de caracteres acentuados y *símbolos especiales* de numerosos idiomas. La tabla 2.1 muestra los diversos tipos de acentos disponibles aplicados a la letra *o* (naturalmente pueden aplicarse a cualquier otra letra). También se incluyen letras escandinavas y otros símbolos especiales.

Tabla 2.1: Acentos y símbolos especiales

ó	<code>\'o</code>	ò	<code>\'o</code>	õ	<code>\~o</code>	ö	<code>\"o</code>
ō	<code>\=o</code>	ô	<code>\.o</code>	ô	<code>\^o</code>		
ǒ	<code>\u o</code>	ǒ	<code>\v o</code>	ǒ	<code>\H o</code>	ȝ	<code>\c o</code>
ȝ	<code>\d o</code>	ȝ	<code>\b o</code>	ōō	<code>\t oo</code>		
œ	<code>\oe</code>	Œ	<code>\OE</code>	æ	<code>\ae</code>	Æ	<code>\AE</code>
å	<code>\aa</code>	å	<code>\aa</code>	Å	<code>\AA</code>		
ø	<code>\o</code>	Ø	<code>\O</code>	ı	<code>\l</code>	Ł	<code>\L</code>
ı	<code>\i</code>	Ј	<code>\j</code>	ı	<code>!'</code>	ı	<code>?'</code>

---

Veamos algunos ejemplos:

<pre> “Paul Erd\“os” en h\‘ungaro\ se escribe Erd\H{o}s P\‘al.\ L’H\^opital, \v{C}ech, Gau\ss ,\ Mu\d hammad ibn M\^us\^a\ al-Khw\^arizm\^i, \AA ngstr\“om. </pre>	<pre> “Paul Erdős” en húngaro se escribe Erdős Pál. L’Hôpital, Čech, Gauß, Muḥammad ibn Mūsā al-Khwārizmī, Ångström. </pre>
--	---

### 2.1.2 Signos de puntuación

Como ya hemos visto las comillas simples se abren con el acento grave y se cierran con un apóstrofo. Para las comillas dobles simplemente se duplica el número de acentos y de apóstrofes.

$\text{\LaTeX}$  reconoce tres tipos de guiones, como se muestra en el ejemplo siguiente. El primer tipo (guión simple) se usa para palabras compuestas, el segundo para rangos numéricos y el tercero para la raya que se usa en los diálogos. Un guión en el modo matemático se interpreta como un signo de ‘menos’ y se compone en un tipo diferente:

<pre> psico-terap’eutico \ p’aginas 83--89\ Ven---me dijo--- \ 1 y \$-1\$ </pre>	<pre> psico-terapéutico páginas 83–89 Ven—me dijo— 1 y −1 </pre>
--	--

Para abrir signos de interrogación o admiración, se escribe el signo que se usa para cerrar seguido de un acento grave. Los puntos suspensivos (o elipsis) no se deben introducir con tres puntos normales, ya que el espaciado resultante no sería correcto. Se usa en cambio la instrucción `\ldots`.

<pre> No as’i ... sino as’i \ldots\ ?‘Est’a claro? !‘Muy bien! </pre>	<pre> No así ... sino así ... ¿Está claro? ¡Muy bien! </pre>
---	--

### 2.1.3 Unidades de longitud

En la Tabla 2.2 se muestran las unidades de medida usadas por  $\text{\LaTeX}$ . Numerosos comandos requieren como argumento una longitud, la cual debe indicarse mediante un número entero o decimal, posiblemente precedido de un signo y obligatoriamente seguido de una unidad. Incluso para la longitud nula es necesario especificar alguna unidad.

Los siguientes son algunos ejemplos de longitudes válidas: `0pt`, `15cm`, `1.6667pt`, `-1.3in`. Los puntos y las picas son unidades tradicionales de los tipógrafos, sobre todo en los países de habla inglesa, mientras que los didots y ciceros son usados en Europa continental. Las unidades `em` y `ex` varían

de acuerdo con los tipos y tamaños utilizados, y por esa razón son muchas veces más flexibles y útiles que las unidades absolutas.

Tabla 2.2: Unidades de T<sub>E</sub>X

---

mm	milímetro	
cm	centímetro	┌───┐
in	pulgada (1in= 2.54cm)	┌──────────┐
pt	punto (1in = 72.27pt)	
bp	punto grande (1in = 72bp)	
pc	pica (1pc = 12pt)	┌┐
dd	punto didot (1157dd= 1238pt)	
pc	cicero (1cc = 12dd)	┌┐
sp	punto escalado (65536sp = 1pt)	
em	aprox. el ancho de una ‘M’ en el tipo actual	┌┐
ex	aprox. la altura de una ‘x’ en el tipo actual	┐

---

En L<sup>A</sup>T<sub>E</sub>X es posible definir variables capaces de contener una longitud. El nombre de la variable puede ser cualquier secuencia de control válida (que no haya sido ya definida) y se crea con el comando

$$\boxed{\backslash\text{newlength}\{nombre\}}$$

Para asignar un valor a una de estas variables se usa el comando

$$\boxed{\backslash\text{setlength}\{nombre\}\{valor\}}$$

También es posible incrementar el valor actual en una magnitud dada, con el comando

$$\boxed{\backslash\text{addtolength}\{nombre\}\{valor\}}$$

En L<sup>A</sup>T<sub>E</sub>X hay variables predefinidas de este tipo, llamadas *parámetros de estilo*, que pueden ser modificadas mediante estos comandos. Por ejemplo `\textwidth` regula el ancho normal que el texto ocupa en la página. Si queremos que este valor sea 10 centímetros debemos escribir

$$\backslash\text{setlength}\{\text{width}\}\{10\text{cm}\}$$

en el preámbulo del documento (en la sección 4.5 se detallan todos los parámetros del diseño de página).

### 2.1.4 Saltos de línea y de página

El comando `\[longitud]` permite iniciar una nueva línea sin justificar la actual. Debe ser usado dentro de un párrafo. El argumento opcional puede usarse para especificar un espacio vertical a ser agregado antes de la nueva línea.

Dos líneas algo separadas.	<code>\[2mm]</code>	Dos líneas algo separadas.
-------------------------------	---------------------	-------------------------------

El comando `\newpage` se usa para comenzar una nueva página. Los comandos `\linebreak[n]` y `\nolinebreak[n]` hacen lo que sugieren sus nombres: comenzar o no una nueva línea. El argumento opcional  $n$  puede ser un valor entero entre cero y cuatro. Cuanto menor sea el valor de  $n$  mayor libertad se le deja a  $\text{\LaTeX}$  para ignorar la orden, en caso de que los resultados de su aplicación sean muy malos. De manera análoga los comandos `\pagebreak[n]` y `\nopagebreak[n]` sirven para comenzar o no una nueva página.

### 2.1.5 Tipos y tamaños

Con  $\text{\LaTeX} 2_{\epsilon}$  se introdujo un nuevo sistema de selección de tipos, diferente al usado en plain  $\text{\TeX}$  y en  $\text{\LaTeX} 2.09$ . El nuevo esquema se conoce como NFSS (*New Font Selection Scheme*), y permite seleccionar de manera independiente tres características de los tipos llamadas familia, forma y serie (en inglés *family*, *shape* y *series*, respectivamente),

Los tipos pertenecientes a una misma *familia* poseen rasgos estilísticos comunes, aunque pueden diferir en tamaño, peso y otros aspectos. Por ejemplo la familia ‘romana’ se caracteriza por tener *serifs*, pequeños trazos horizontales presentes en los extremos de los caracteres. Originalmente estos trazos eran producto del método utilizado por los antiguos romanos para grabar los caracteres en la piedra con un cincel, pero actualmente se cree que los *serifs* ayudan a la vista a pasar de una letra a otra durante la lectura, por lo cual se prefiere esta familia para componer textos extensos. La familia *sans serif* se caracteriza naturalmente por la ausencia de *serifs* y la familia *máquina* por sus caracteres monoespaciados semejantes a los producidos por las máquinas de escribir.

Los tipos de una familia como la romana pueden presentarse en la *forma* (en inglés *shape*) vertical, *oblicua*, *itálica* o *VERSALITA* (en inglés: *upright*, *slanted*, *italic* y *SMALL CAPS*, respectivamente).

La *serie* representa el grosor de los trazos constituyentes de los caracteres.

Las tablas 2.3 y 2.4 resumen los comandos disponibles para seleccionar familias, formas, series y tamaños en el modo de texto (en el modo matemático no se usan estas instrucciones sino las que se detallan en la tabla 3.1 del



Capítulo siguiente).

Tabla 2.3: Tipos en el modo de texto

Comando	Declaración	Ejemplo
<code>\textup</code>	<code>\upshape</code>	vertical
<code>\textit</code>	<code>\itshape</code>	<i>itálica</i>
<code>\textsl</code>	<code>\slshape</code>	<i>inclinada</i>
<code>\textsc</code>	<code>\scshape</code>	VERSALITA
<code>\textmd</code>	<code>\mdseries</code>	media
<code>\textbf</code>	<code>\bfseries</code>	<b>negrita</b>
<code>\textrm</code>	<code>\rmfamily</code>	redonda
<code>\textsf</code>	<code>\sffamily</code>	sans serif
<code>\texttt</code>	<code>\ttfamily</code>	máquina

Tabla 2.4: Tamaños de los tipos

<code>\tiny</code>	diminuta	<code>\Large</code>	más grande
<code>\scriptsize</code>	muy pequeña	<code>\LARGE</code>	muy grande
<code>\footnotesize</code>	bastante pequeña	<code>\huge</code>	enorme
<code>\small</code>	pequeña	<code>\Huge</code>	la mayor
<code>\normalsize</code>	normal		
<code>\large</code>	grande		

Idealmente las instrucciones de cambio de tipo no deberían usarse directamente sino como parte de comandos que definan elementos lógicos o estructurales del documento.  $\text{\LaTeX}$  hace esto al seleccionar automáticamente los tipos adecuados para los encabezados de capítulo, sección, subsección, notas al pie de página, etc. Si en un artículo decidimos usar versalitas para componer los nombres de los autores citados, es preferible definir un comando especial para hacerlo y llamarlo por ejemplo `\autcit` en vez de usar directamente `\textsc` (en la sección 2.11 veremos cómo definir nuestros propios comandos). De este modo, si cambiamos de idea respecto al estilo de los nombres solamente tendremos que redefinir el comando `\autcit`.

Las diversas familias, formas, series y tamaños se pueden combinar para obtener, por ejemplo, tipos sans serif inclinados y en negritas. Puede ocurrir sin embargo que los *fonts* necesarios para alguna combinación particular no existan en una determinada instalación, en cuyo caso  $\text{\LaTeX}$  tratará de aproximar el efecto deseado sustituyendo los tipos faltantes por otros que

sí estén disponibles.

Letras `\Large` grandes  
`\textit{it'alicas}` y `\`  
`\sffamily sans serif`  
`\slshape inclinadas` o  
`\bfseries negritas`.

Letras grandes *itálicas* y  
 sans serif *inclinadas* o **negritas**.

Por razones de compatibilidad con L<sup>A</sup>T<sub>E</sub>X 2.09 las clases estándar de L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> aceptan también las declaraciones `\bf`, `\it`, `\tt`, `\sc` y `\rm` para obtener negrillas, itálicas, máquina, versalitas y tipos romanos, respectivamente. Estas declaraciones sin embargo no son sensibles al contexto, y su uso no es recomendable,

Las instrucciones de tamaño también alteran el espaciado entre líneas, pero sólo si el párrafo termina dentro del ámbito de la orden de tamaño del tipo. Por ello, la llave de cierre `}` no debe aparecer antes de lo indicado. Compare la posición de la instrucción `\par` (equivalente a dejar una línea en blanco) en los dos ejemplos siguientes:

`\Large Este es un T'itulo Largo`  
`con Iniciales May'usculas\par}`

Este es un Título Largo  
 con Iniciales Mayúsculas

`\Large Este otro en cambio se`  
`ve Demasiado Apretado.}\par`

Este otro en cambio se ve  
 Demasiado Apretado.

### 2.1.6 Notas al pie de página

Con la instrucción

```
\footnote[num]{texto de la nota al pie}
```

se imprimirá una nota en el pie de la página actual. El argumento opcional *num*, si se usa, debe ser un entero positivo (aún cuando las notas se “numeren” con letras u otros símbolos). Si no se incluye, el valor del contador `footnote` se incrementará y el valor resultante será usado como *num*,

Algunos autores `\footnote{Entre`  
`los cuales no me incluyo}` son  
 muy afectos a usar notas al pie  
 de página.

Algunos autores<sup>a</sup> son muy afectos a  
 usar notas al pie de página.

---

<sup>a</sup>Entre los cuales no me incluyo

El parámetro `\footnotesep` regula la separación vertical entre las notas al pie de página.

### 2.1.7 Notas marginales

Con la instrucción

```
\marginpar{texto de la nota marginal}
```

se imprimirá una nota en el margen del documento, comenzando en la línea que contiene a la instrucción. La nota que aquí se ve se obtuvo con: *Esta es una nota marginal.*

```
\marginpar{\small\em Esta es una\\nota marginal.}
```

En las clases estándar las notas marginales aparecen a la derecha, o en el margen exterior si se usa impresión bilateral. En este último caso se puede usar un primer argumento opcional para especificar el contenido de la nota si cae en el margen izquierdo (el argumento obligatorio se usará para el margen derecho).

### 2.1.8 Espaciado

#### Espaciado entre líneas

La separación *normal* entre líneas está regulada por un parámetro de  $\text{\TeX}$  llamado `\baselineskip`. Su valor por defecto es 20% mayor que el tamaño de los tipos en uso (por ejemplo para tipos de 10pt `\baselineskip` será 12pt) pero la separación real entre líneas puede ser mayor, si alguna línea contiene objetos más altos de lo normal. Este parámetro puede modificarse directamente, pero no es conveniente hacerlo ya que el valor apropiado depende del tamaño y tipo de letras que se estén utilizando. Por ello  $\text{\LaTeX}$  proporciona el parámetro `\baselinestretch`, el cual permite cambiar el espaciado globalmente. Su valor multiplica a `\baselineskip` cada vez que se produce un cambio de tamaño. Por ejemplo si necesita imprimir un documento a doble espacio (posiblemente para facilitar la revisión de una versión preliminar) puede hacerlo incluyendo en el preámbulo la orden

```
\renewcommand{\baselinestretch}{1.67}
```

¿Por qué usamos 1.67 y no sencillamente 2? Pues porque *doble espacio* se entiende por lo general como *espacio entre líneas igual al doble del tamaño de los caracteres* y `\baselineskip` ya es 1.2 veces dicho tamaño.

Para situaciones más complejas se puede usar el paquete `doubleSPACE`, el cual proporciona un ambiente para modificar el espaciado localmente.

#### Párrafos

El parámetro `\parindent` regula la sangría al comienzo de cada párrafo, mientras que `\parskip` determina el espacio entre párrafos consecutivos.

Los valores más usuales de estos parámetros varían según el idioma y el país. Si es necesario se pueden modificar con la instrucción `\setlength`, por ejemplo si se escribe

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex}
```

en el preámbulo del documento, se obtendrán párrafos más separados y sin sangría.

Si quiere que un párrafo en particular no tenga sangría inícielo con el comando `\noindent`. El comando `\indent` por supuesto tiene el efecto contrario.

Observe que en las clases estándar de  $\text{\LaTeX}$  el primer párrafo de cada sección no se sangra, de acuerdo con el estilo anglosajón. Si lo desea esto puede cambiarse fácilmente con el paquete `indentfirst`.

## Espacios horizontales

$\text{\LaTeX}$  determina automáticamente las separaciones entre palabras y oraciones. Para producir otras separaciones horizontales utilice el comando:

 $\text{\hspace}{longitud}$ 

La variante `\hspace*` genera el espacio incluso al comienzo o al final de una línea.

Un espacio de `\hspace{1cm}` 1 cm.

Un espacio de 1 cm.

La instrucción

 $\text{\stretch}{n}$ 

produce una separación *elástica*, que se alarga hasta llenar el espacio que quede en la línea. Si dos instrucciones `\hspace{\stretch{n}}` aparecen en la misma línea entonces generan espacios proporcionales a sus argumentos numéricos.

`x\hspace{\stretch{1}}`  
`x\hspace{\stretch{3}}`x

x

x

x

## Espacios verticales

L<sup>A</sup>T<sub>E</sub>X determina de modo automático la separación entre dos párrafos, apartados o subapartados. En casos especiales se pueden forzar separaciones adicionales entre dos párrafos con la orden

`\vspace{longitud}`

Esta orden surte efecto *después* de la línea en que se encuentra. Cuando la separación se debe introducir aunque vaya al principio o al final de una página, entonces se usa la variante `\vspace*`.

Se puede utilizar la orden `\stretch` conjuntamente con `\pagebreak` para llevar texto al borde inferior de una página o para centrarlo verticalmente.

Algo de texto \ldots

```
\vspace{\stretch{1}}
```

Esto va en la 'ultima l'inea de la p'agina.

```
\pagebreak
```

También se pueden lograr espacios verticales convenientes con las instrucciones de T<sub>E</sub>X `\smallskip`, `\medskip` y `\bigskip`, que corresponden a 3pt, 6pt y 12pt respectivamente.

## Espacio entre palabras

Para conseguir un margen derecho recto en la salida, L<sup>A</sup>T<sub>E</sub>X introduce cantidades variables de espacios entre las palabras. Al final de una oración, introduce unos espacios algo mayores que favorecen la legibilidad del texto. L<sup>A</sup>T<sub>E</sub>X presupone que las frases acaban con un punto, un signo de interrogación o un signo de admiración. Pero si hay un punto tras una letra mayúscula, entonces esto no se considera el fin de una oración ya que los puntos tras las letras mayúsculas normalmente se utilizan para abreviaturas. El autor debe indicar cualquier excepción a estas reglas. El carácter `\` antes de un espacio en blanco produce un espacio en blanco que no se ensanchará. Un carácter `~` (tilde) genera un espacio que no se puede ensanchar y en el que no se puede producir ningún cambio de línea (pero tenga cuidado si está usando la opción `activeacute` del paquete `babel`, ya que si una tilde es seguida por una `'n'` o `'N'` se interpretará como ñ; en estos casos introduzca una separación lógica abriendo y cerrando llaves antes de la `n`). Si antes de un punto aparece la instrucción `\@`, significa que este punto acaba una oración, aunque se encuentre tras una letra mayúscula.

Yo tomo vitamina~C\@. ¿'Y t'u?

Yo tomo vitamina C. ¿Y tú?

Este tratamiento especial para los espacios al final de las oraciones se puede evitar con la instrucción `\frenchspacing`, la cual le indica a  $\text{\LaTeX}$  que no introduzca más espacios tras un punto que tras cualquier otro carácter. Esto es común en francés y otros idiomas.

### 2.1.9 Ajuste de textos

Los ambientes `flushleft` y `flushright` producen párrafos ajustados a la izquierda y a la derecha (sin nivelación de bordes). El ambiente `center` genera texto centrado. Si no se introduce `\` para dividir las líneas,  $\text{\LaTeX}$  lo hará automáticamente.

<pre>\begin{flushleft} En un texto ajustado a la izquierda \LaTeX\ no trata de que todas las l'ineas tengan igual longitud. \end{flushleft}</pre>	<p>En un texto ajustado a la izquierda <math>\text{\LaTeX}</math> no trata de que todas las líneas tengan igual longitud.</p>
---	---

<pre>\begin{center} Este es un ejemplo de\ texto centrado. \end{center}</pre>	<p>Este es un ejemplo de texto centrado.</p>
---	--

<pre>En el año <math>x^2</math> yo ten'ia <math>x</math> años de edad. \begin{flushright} A. De Morgan (1800--1871) \end{flushright}</pre>	<p>En el año <math>x^2</math> yo tenía <math>x</math> años de edad.</p> <p style="text-align: right;">A. De Morgan (1800–1871)</p>
--	--

### 2.1.10 Citas y poesía

El ambiente `quote` se usa para citas pequeñas. Para citas más extensas, que abarquen varios párrafos, se usa el ambiente `quotation`.

<pre>Como dijo Groucho Marx, \begin{quote} ‘‘Citadme diciendo que me han citado mal.’’ \end{quote}</pre>	<p>Como dijo Groucho Marx,</p> <p style="text-align: center;">“Citadme diciendo que me han citado mal.”</p>
--	---

El ambiente `verse` es apropiado para poemas, en los cuales el espacio entre líneas es esencial. Los versos se dividen con `\` y las estrofas con líneas en blanco.

```

\begin{flushleft}
\begin{verse}
Un primer verso corto,\ \
el segundo verso bastante m'as largo.
Un primer verso corto,
el segundo verso
bastante más largo.
Y la segunda estrofa.
\end{verse}
\end{flushleft}

```

### 2.1.11 Edición textual

El texto que se encuentre entre `\begin{verbatim}` y `\end{verbatim}` aparecerá textualmente, en caracteres monoespaciados (como si se hubiese compuesto con una máquina de escribir). Se respetarán todos los espacios y saltos de línea y los comandos de  $\text{\LaTeX}$  no serán interpretados.

```

\begin{verbatim}
10 PRINT "BUCLE INFINITO";
20 GOTO 10
\end{verbatim}

\begin{verbatim*}
La version con estrella de
el ambiente      verbatim
destaca los espacios en
el texto
\end{verbatim*}

```

Dentro de un párrafo se puede lograr el mismo efecto con

`\verb+texto+`

El `+` sólo es un ejemplo de carácter delimitador. Se puede usar cualquier carácter excepto letras, espacio y `*` (ya que esta instrucción admite también la variante con asterisco).

### 2.1.12 Referencias cruzadas

En los libros, informes y artículos técnicos se suelen emplear referencias a figuras, tablas, ecuaciones y porciones de texto que se encuentran en otros lugares del documento.  $\text{\LaTeX}$  proporciona las siguientes instrucciones para producir referencias cruzadas:

`\label{etiqueta}, \ref{etiqueta} y \pageref{etiqueta}`

donde *etiqueta* es un identificador elegido por el usuario.  $\LaTeX$  reemplaza  $\backslash\text{ref}$  por el número del apartado, subapartado, figura, tabla o teorema donde se introdujo la instrucción  $\backslash\text{label}$  correspondiente. La orden  $\backslash\text{pageref}$  imprime el número de página donde se produce la orden  $\backslash\text{label}$  con igual argumento.

Una referencia a este apartado $\backslash\text{label}\{\text{ej:ref}\}$ dir'ia, por ejemplo:	Una referencia a este apartado diría, por ejemplo: “vea el apartado 2.1.12 en la pági- na 32.”
‘‘vea el apartado~ $\backslash\text{ref}\{\text{ej:ref}\}$ en la p'agina~ $\backslash\text{pageref}\{\text{ej:ref}\}$ .’’	

Observe que aunque añada, quite o mueva texto y apartados las referencias cruzadas siempre mostrarán los números de sección y página correctos. A veces sin embargo deberá texear el documento dos veces para que  $\LaTeX$  pueda resolver correctamente todas las referencias (preste atención a los mensajes generados al texear el documento). Las referencias no resueltas se verán como signos de interrogación.

### 2.1.13 División de palabras

Al componer cada línea  $\TeX$  ajusta los espacios entre las palabras de manera que el texto quede alineado a la derecha, ni demasiado apretado ni demasiado separado. Si es necesario, introduce guiones para dividir las palabras que no encajen bien al final de una línea (por defecto lo hará de acuerdo a las reglas del idioma inglés, pero puede lograrse que lo haga para otros idiomas, vea la sección 2.2).

Si  $\TeX$  no puede alinear los bordes derechos de un párrafo satisfactoriamente, entonces dejará una línea demasiado larga y emitirá un mensaje de advertencia (*overfull box*) mientras procesa el archivo de entrada. Esto sucede en especial si no encuentra un lugar apropiado para introducir un guión entre las sílabas. Si se introduce el comando  $\backslash\text{sloppy}$ ,  $\LaTeX$  será menos severo en sus exigencias y evitará tales líneas largas, aumentando la separación entre las palabras. En este caso emitirá otro mensaje de advertencia (*underfull box*).

En algunos casos el algoritmo de división de palabras no produce resultados correctos. Esto puede ocurrir especialmente si usamos palabras compuestas, palabras en un idioma para el cual no tengamos patrones de silabeo, o palabras que incluyan caracteres especiales o acentuados (si se está usando el esquema de codificación  $\text{OT1}$ ). En esos casos resulta útil la instrucción  $\backslash-$ , la cual intercalada dentro de una palabra establece uno o más puntos donde colocar un guión si fuese necesario. Además, éstos se convierten en los *únicos* lugares donde se permite dividir esa palabra.



Y lamentablemente su  
`\emph{welt-an-schau-ung}`  
 era bastante decadente y  
`de-ci-mo-n'o-ni-ca`.

Y lamentablemente su *weltanschau-  
 ung* era bastante decadente y decimo-  
 nónica.

Cuando una palabra problemática se repite varias veces en el documento es posible declarar una sola vez cómo debe dividirse, con el comando

```
\hyphenation{lista de palabras}
```

Su efecto es que las palabras que se le pasan como argumento se puedan dividir únicamente en los lugares indicados con “-”. Este comando debe aparecer en el preámbulo del archivo de entrada y puede contener solamente palabras sin caracteres especiales. No hace distinciones entre letras mayúsculas y minúsculas. El ejemplo siguiente permite guionizar correctamente la palabra *weltanschauung* y además impide que la palabra “latex” (y sus variantes con mayúsculas) sean divididas:

```
\hyphenation{welt-an-schau-ung latex}
```

## 2.2 El paquete Babel

El paquete `babel` de Johannes L. Braams facilita la composición de textos en idiomas diferentes del inglés, o en los cuales se empleen dos o más idiomas diferentes. Para usarlo es necesario declarar el paquete en el preámbulo:

```
\usepackage[opciones]{babel}
```

Los idiomas a utilizar se deben incluir como argumentos opcionales. Por ejemplo

```
\usepackage[spanish,english]{babel}
```

nos permitirá usar español e inglés en el mismo documento. Para cambiar de un idioma a otro se usa la instrucción

```
\selectlanguage{idioma}
```

Para algunos idiomas, `babel` define instrucciones que simplifican la entrada de caracteres acentuados o especiales. En español, como ya hemos visto, se pueden generar las vocales acentuadas simplemente precediéndolas de un apóstrofo. Del mismo modo la ñ se genera escribiendo simplemente `~n`. Sin embargo como este carácter *activo* del apóstrofo y la tilde pueden interferir con otras cosas y producir resultados indeseados, en las versiones

más recientes de `babel` no está disponible por defecto sino que se debe activar incluyendo `activeacute` entre los argumentos opcionales.

Las reglas para dividir palabras dependen del idioma. Por ejemplo la palabra ‘separableén castellano se silabea se-pa-ra-ble pero en inglés, aunque se escribe igual, se silabea sep-a-ra-ble. Lamentablemente, no es suficiente usar `babel` y seleccionar un lenguaje para que  $\LaTeX$  guionice correctamente: también es necesario tener un *formato* que incluya los patrones de silabeo adecuados (un formato es un conjunto de macros precompiladas para que  $\TeX$  las pueda cargar rápidamente al iniciarse). El formato de  $\LaTeX$  se crea generalmente durante la instalación y por razones de eficiencia incluye solamente los patrones de silabeo de los idiomas que se van a utilizar. Es posible ver los patrones disponibles en el formato examinando las primeras líneas del archivo `.log` que se crea cada vez que texeamos un documento. Si se va a trabajar en un idioma para el cual el formato no tiene patrones, es necesario crear un nuevo formato que los incluya. Generalmente para esto hay que editar un archivo llamado `language.dat`, en el cual se especifican los idiomas que se van a usar y los nombres de los archivos que contienen los respectivos patrones. Generalmente ese archivo ya incluye líneas apropiadas para varios idiomas, pero comentadas con un carácter `%` al comienzo de las mismas. Lo que debe hacerse es descomentar las líneas que nos interesan (removiendo el `%` inicial) y posiblemente comentar las líneas correspondientes a idiomas que nunca vamos a usar. Luego se ejecuta el programa `initex`, con los parámetros adecuados, para crear el nuevo formato (en los sistemas tipo UNIX esto debe hacerlo el superusuario, así que deberá hablar con el administrador del sistema al respecto). Algunas distribuciones incluyen utilidades que facilitan esta tarea, como por ejemplo `texconfig` en  $\TeX$ . MikTeX [17] (para Windows) incluye también una utilidad para crear formatos (debe primero editar `language.dat`, por supuesto).

Ahora bien, aún con los patrones de silabeo apropiados es posible que  $\TeX$  no sea capaz de dividir correctamente algunas palabras. Esto ocurrirá sobre todo con las palabras acentuadas si estamos utilizando la codificación `OT1` (la que  $\LaTeX$  usa por defecto). La solución consiste en usar la codificación `T1` y los *fonts ec* (*European Computer Modern Fonts*), con los cuales cada letra acentuada es un carácter único y no una superposición de símbolos. Estos *fonts* se pueden bajar de CTAN [3], pero si usted tiene una distribución reciente de  $\LaTeX$  seguramente vienen incluidos. Para cambiar la codificación se usa el paquete `fontenc`, escribiendo en el preámbulo del documento la instrucción

```
\usepackage[T1]{fontenc}
```

Un posible inconveniente a tener en cuenta es que se reduzca la portabilidad de los documentos así creados, ya que la primera versión estable de los `ec` es de 1997 y en muchas instalaciones aún no están disponibles.

Otra característica útil de `babel` es que redefine los títulos generados por algunas instrucciones de `LATEX` como `\tableofcontents` y ambientes como `abstract` y `thebibliography`, para que aparezcan en el idioma correcto. Lo mismo hace con la fecha generada por la instrucción `\today`.

## 2.3 Listas

Hay un ambiente genérico denominado `list` con el cual se pueden construir listas de muchos tipos; sin embargo su uso es algo complicado y no lo trataremos aquí (vea [9] para más detalles). Pero hay tres ambientes fáciles de usar con los cuales se pueden construir los tipos de listas más comunes. Estos ambientes se pueden mezclar y anidar hasta una profundidad de cuatro niveles.

El ambiente `itemize` se emplea para listas sencillas, donde los items son precedidos por viñetas. El ambiente `description` se utiliza para descripciones, como muestra el ejemplo siguiente.

```

\begin{itemize}
\item Cada item de una lista tiene
una etiqueta, que puede ser:
\begin{itemize}
\item una viñeta.
\item una descripci'on.
\item un n'umero o una letra.
\end{itemize}
\item En las descripciones el
usuario debe proporcionar las
etiquetas. Por ejemplo:
\begin{description}
\item[gules] \textit{Blas}. rojo
\item[sinoples] \textit{Blas}. verde
\end{description}
\end{itemize}

```

- Cada item de una lista tiene una etiqueta, que puede ser:
  - una viñeta.
  - una descripción.
  - un número o una letra.
- En las descripciones el usuario debe proporcionar las etiquetas. Por ejemplo:
  - gules** *Blas.* rojo
  - sinoples** *Blas.* verde

Para esquemas numerados se utiliza el ambiente `enumerate`. Como se ve en el siguiente ejemplo `LATEX` numera primero con cifras arábigas y si hay listas anidadas utiliza letras minúsculas para el segundo nivel y números romanos para el tercero (si hubiese un cuarto nivel emplearía letras mayúsculas).

```

\begin{enumerate}
\item Conjuntos.
  \begin{enumerate}
\item Finitos.
\item Infinitos.
  \begin{enumerate}
\item Numerables.
\item No numerables.
  \end{enumerate}
  \end{enumerate}
\end{enumerate}
\item N'umeros.
  \begin{enumerate}
\item Racionales.
\item Irracionales.
  \end{enumerate}
\end{enumerate}

```

1. Conjuntos.
  - (a) Finitos.
  - (b) Infinitos.
    - i. Numerables.
    - ii. No numerables.
2. Números.
  - (a) Racionales.
  - (b) Irracionales.

El estilo de numeración puede modificarse redefiniendo ciertos comandos, pero es más sencillo hacerlo con el paquete `enumerate` de David Carlisle, el cual permite especificar el estilo deseado con un argumento opcional que se incluye inmediatamente a continuación de `\begin{enumerate}`. En dicho argumento los símbolos `a`, `A`, `i`, `I` y `1` se interpretan respectivamente como letras minúsculas, letras mayúsculas, números romanos en minúsculas, números romanos en mayúsculas y números arábigos. Para que alguno de estos caracteres no sea interpretado, debe colocarse entre llaves. Por ejemplo el argumento opcional `[(i)]` generará items numerados con (i), (ii), (iii), (iv), ... , mientras que el argumento `[{Inciso} 1]` generará las etiquetas Inciso 1), Inciso 2), Inciso 3) ...

## 2.4 Tablas

El ambiente `tabular` sirve para crear *tablas* con líneas horizontales y verticales según se desee.  $\text{\LaTeX}$  determina el ancho de las columnas de modo automático.

El argumento *especificaciones* de la instrucción

```

\begin{tabular}{especificaciones}

```

define el diseño de la tabla. Utilice `l` para una columna con texto justificado a la izquierda, `r` para justificar el texto a la derecha, `c` para texto centrado, `p{ancho}` para una columna que contenga texto con saltos de línea, y `|` para una línea vertical.

Dentro de un ambiente `tabular`, `&` salta a la próxima columna, `\\` separa las líneas y `\hline` introduce una línea horizontal.

```

\begin{tabular}{|r|r|r|}
\hline
$n$ & $B_n$ & $G_n$ \\
\hline\hline
1 & 1 & 1 \\
2 & 2 & 2 \\
3 & 5 & 4 \\
4 & 15 & 11 \\
5 & 52 & 34 \\
\hline
\end{tabular}

```

$n$	$B_n$	$G_n$
1	1	1
2	2	2
3	5	4
4	15	11
5	52	34

Pueden construirse tablas con estructura más compleja, utilizando las instrucciones `\multicolumn` y `\cline` para crear celdas y trazar líneas, respectivamente, que abarquen varias columnas (ver ejemplo siguiente). Además con la construcción `@{...}` se puede especificar el separador de columnas. Esta construcción elimina el espacio entre columnas y lo reemplaza con lo que se haya introducido entre los paréntesis. Por ejemplo se puede usar `@{}` para eliminar el espacio que antecede y precede a las líneas de una tabla. Otro uso posible es la alineación de una columna de números por la coma decimal. Para ello se usan dos columnas: una para la parte entera, alineada a la derecha y otra para los decimales, alineados a la izquierda. La instrucción `@{,}` en el argumento de `\begin{tabular}` reemplaza el espacio normal entre columnas con una coma, dando la apariencia de una única columna alineada por la coma decimal.

```

\begin{tabular}{|l||c|c|}
\hline
Ejemplo & & de una Tabla \\
\hline
con & divisiones & y \\
\multicolumn{2}{|c|}{complejas} & algunos \\
\hline
números & 2.57 & 0.123 \\
\hline
\end{tabular}

```

Ejemplo	de una Tabla	
con	divisiones	y
	complejas	algunos
números	2.57	0.123

## 2.5 Objetos flotantes

Algunos elementos de un documento, como las figuras y las tablas, no se pueden dividir entre páginas. Por lo tanto, cuando no caben en el espacio que queda en la página al momento de su aparición, hay que hacerlos *flotar* hasta una página posterior y rellenar lo que queda de la página actual con el texto del documento.  $\text{\LaTeX}$  ofrece dos ambientes para estos *objetos flotantes*: uno para las tablas y otro para las figuras.

Es importante entender cómo maneja  $\text{\LaTeX}$  la colocación de estos objetos flotantes para evitar frustraciones al ver que no quedan donde nosotros

Tabla 2.5: Permisos para objetos flotantes

Permiso	Explicación
<b>h</b>	( <i>here</i> ) aquí, próximo al lugar donde se ha introducido. Es útil sobre todo para objetos pequeños.
<b>t</b>	( <i>top</i> ) en la parte superior de una página.
<b>b</b>	( <i>bottom</i> ) en la parte inferior de una página.
<b>p</b>	( <i>page</i> ) en una página especial que sólo contenga objetos flotantes.
<b>!</b>	sin considerar la mayoría de los parámetros internos (como el número máximo de elementos flotantes en una página) que impedirían la colocación de este objeto.

quisiéramos.

Cualquier cosa que se incluya en un ambiente `figure` o `table` será tratada como objeto flotante. Ambos ambientes admiten un argumento opcional que se usa para indicarle a  $\text{\LaTeX}$  los lugares donde está permitido colocar el objeto flotante, y que consiste en una cadena de *permisos* que se detallan en la tabla 2.5.

Una tabla podría comenzar, por ejemplo, con la siguiente línea:

```
\begin{table}[!hbp]
```

El argumento `[!hbp]` le permite a  $\text{\LaTeX}$  colocar la tabla justamente aquí (**h**) o al final (**b**) de alguna página o en alguna página especial para elementos flotantes, y en cualquier parte si no queda tan bien (**!**). Si no se da ningún permiso entonces por defecto se asume `[tbp]`.

$\text{\LaTeX}$  colocará todos los objetos flotantes que encuentra según los permisos que haya indicado el autor. Si un objeto flotante no se puede colocar en la página actual entonces se aplaza su colocación, para lo cual se introduce en una cola tipo FIFO (*first in first out*, o “lo primero que entra es lo primero que sale”) de *tablas* o de *figuras*. Cuando se comienza una nueva página, lo primero que hace  $\text{\LaTeX}$  es confirmar si se puede construir una página especial con los objetos flotantes existantes en las colas. Si no es posible, entonces trata el primer objeto que se encuentra en las colas como si lo acabásemos de introducir.  $\text{\LaTeX}$  vuelve a intentar colocar el objeto según sus permisos de posición. Cualquier objeto flotante nuevo que aparezca se introduce en la cola correspondiente.  $\text{\LaTeX}$  mantiene estrictamente el orden original de apariciones de cada tipo de objeto flotante.

Esta es la razón por la que una ilustración que no se puede colocar desplaza al resto de las figuras al final del documento. Por lo tanto, si los objetos flotantes no aparecen como usted esperaba, puede deberse a un objeto que está obstruyendo una de las colas.

La instrucción

```
\caption{texto de título}
```

permite definir un título para el objeto flotante. L<sup>A</sup>T<sub>E</sub>X añadirá automáticamente la palabra “Figura” o “Tabla” y un número de secuencia.

Las dos instrucciones

```
\listoffigures y \listoftables
```

funcionan de modo análogo a la orden `\tableofcontents`, imprimiendo un índice de figuras o de tablas respectivamente. En estas listas se repetirán los títulos completos. Si Vd. tiende a utilizar títulos largos, debería tener una versión de estos títulos más cortos para introducirlos en estos índices. Esto se consigue dando la versión corta entre corchetes tras la orden `\caption`.

```
\caption[Unidades]{Unidades legales de longitud, masa y tiempo}
```

Con `\label` y `\ref` se pueden crear referencias a un objeto flotante dentro del texto.

En ciertas condiciones podría ser necesario usar la orden

```
\clearpage
```

la cual ordena que se coloquen inmediatamente todos los objetos flotantes que se hallen en las colas para comenzar después una nueva página.

## 2.6 Referencias bibliográficas

Con el entorno `thebibliography` se puede imprimir una bibliografía.

```
\begin{thebibliography}{ancho}
entradas
\end{thebibliography}
```

El argumento *ancho* es una cadena de caracteres cuya longitud establece el espacio que se reservará para colocar las etiquetas (por lo tanto debe ser aproximadamente igual a la mayor de ellas). Cada entrada se introduce con el comando `\bibitem`:

```
\bibitem[etiqueta]{clave}
```

El argumento opcional *etiqueta* se colocará frente a cada entrada en la bibliografía. Si no se incluye, éstas sencillamente se numeran. La *clave* se utiliza dentro del documento para citar la referencia respectiva, con el comando `\cite`:

```
\cite{clave}
```

Por ejemplo la entrada correspondiente a *The T<sub>E</sub>Xbook* en la bibliografía de este manual se compuso así:

```
\bibitem{texbook} Knuth, D.E. \emph{The \TeX book},
    Addison-Wesley, Reading, 1984.
```

y para citarla escribimos `\cite{texbook}`.

Un programa adicional muy útil es BIB<sub>T</sub>E<sub>X</sub>. Para usarlo primero debemos crear una base de datos con información sobre las obras que solemos citar. En el archivo fuente se debe indicar el nombre de la base de datos y el estilo que deseamos para la bibliografía. Al procesar nuestro archivo fuente con el programa BIB<sub>T</sub>E<sub>X</sub> éste generará un archivo con comandos para componer automáticamente la bibliografía, a partir de los comandos `\cite` que encuentre. Los detalles sobre el uso de este programa pueden verse en [9] o en la documentación que acompaña a BIB<sub>T</sub>E<sub>X</sub>.

## 2.7 Estilos de página

Cada página se compone de tres partes: la *cabecera*, el *cuerpo* y el *pie*. Por lo general la cabecera contiene un número de página y el título de la sección o del capítulo, aunque en ciertas páginas especiales la cabecera puede estar vacía y el número de página encontrarse en el pie. Por defecto los números de página son arábigos, pero puede especificarse el estilo deseado con el comando

```
\pagenumbering{estilo-num}
```

donde *estilo-num* puede ser `roman` para números romanos o `arabic` para numeración arábica. Este comando inicializan el contador de páginas a 1, comenzando con la página en la cual se encuentre. Para comenzar a numerar a partir del número *n* se puede usar el comando:

```
\setcounter{page}{n}
```

Existen cuatro combinaciones estándar de cabeceras y pies de página, a los cuales se les llama *estilos de página*. El argumento *estilo* del comando



```
\pagestyle{estilo}
```

define cuál se va a usar. La tabla 2.6 muestra los estilos de página predefinidos.

Tabla 2.6: Estilos de página predefinidos en L<sup>A</sup>T<sub>E</sub>X

---

<b>plain</b>	imprime los números de página en el centro del pie de las páginas. Este es el estilo que se utiliza por defecto para las clases <code>article</code> y <code>report</code> .
<b>empty</b>	deja las cabeceras y pies de página vacíos.
<b>headings</b>	en la cabecera de cada página se imprime el número de la misma más otra información adicional que depende de la clase del documento, mientras que el pie queda vacío.
<b>myheadings</b>	es similar a <b>headings</b> , pero nos permite especificar la información adicional a incluir en las cabeceras.

---

Con el estilo `myheadings`, se usa el comando

```
\markboth {cabecera izquierda}{cabecera derecha}
```

en el modo bilateral (*twosided*) para indicar el contenido de las cabeceras. En el modo unilateral se usa:

```
\markright {cabecera}
```

Es posible cambiar el estilo de la página actual con la instrucción

```
\thispagestyle{estilo}
```

El diseño de página, incluidas las cabeceras y pies de página, se pueden personalizar con el paquete `fancyhdr`.

## 2.8 Clases de documento estándar

Recordemos que L<sup>A</sup>T<sub>E</sub>X ofrece varias clases de documento estándar que cubren las necesidades más comunes. Estas clases proporcionan varios medios para estructurar adecuadamente los documentos. Algunos son comunes a todas o a varias clases y otros específicos de alguna de ellas. En esta sección describiremos las características más importantes de cada clase.

## Clase article

Se usa para escribir artículos para revistas, ponencias e informes breves.

El título, autor y fecha de creación del documento se definen con las instrucciones

```

\title{título}
\author{autor(es)}
\date{fecha}
```

El argumento de `\author` admite varios nombres de autores separados con la orden `\and`. Si no especifica la fecha entonces `LATEX` incluirá automáticamente la actual. Si no desea que aparezca fecha alguna puede escribir `\date{}`.

La instrucción

```
\maketitle
```

genera el título con la información suministrada por los comandos anteriores.título

Los artículos se pueden seccionar con los siguientes comandos:

```

\section      \subsection  \subsubsection
\paragraph   \subparagraph
```

Cada uno de ellos tiene un argumento obligatorio que determina el título de la sección y uno opcional que se utiliza para indicar una versión corta del título para la Tabla de Contenidos, cuando el título completo es muy largo. También hay versiones con estrella de estos comandos, que difieren en que la sección respectiva no se numera ni aparece en el índice general.

`LATEX` se encarga de numerar y componer los títulos de las secciones con tipos y tamaños adecuados a su importancia, además de espaciarlas de manera apropiada.

El comando `\appendix` cambia la forma de numerar las secciones que aparezcan después de él, de números a letras (lo cual se acostumbra en los apéndices).

`LATEX` puede crear un Índice General (o Tabla de Contenidos) a partir de los títulos de las unidades seccionales y los números de las páginas en que se encuentran. La instrucción

```
\tableofcontents
```

hace que se introduzca este índice en el sitio del documento donde la coloquemos. Para generar el índice  $\LaTeX$  utiliza un archivo con el mismo nombre del fuente pero con la extensión `toc`. Cada vez que se texea un documento, si la instrucción `\tableofcontents` está presente, se genera el índice a partir de la información contenida en el archivo `toc` (suponiendo que el mismo exista como producto de una corrida anterior) y luego se actualiza el mismo con la información recolectada en la corrida actual. Esto significa que si modificamos nuestro archivo fuente, los cambios se reflejarán en el índice general solamente después de texearlo dos veces.

### Clase report

Se usa para escribir reportes que consten de más de un capítulo, disertaciones y similares. En las clases `report` y `book` se pueden utilizar todas las instrucciones de seccionado vistas para la clase `article`, más dos instrucciones adicionales: `\part` y `\chapter`. El comando `\part` no influye en la numeración de los capítulos.

En estas clases el comando `\appendix` cambia el modo de numerar los capítulos (de números a letras).

### Clase book

Se usa para escribir libros.

Además de todas las instrucciones ya vistas, en la clase `book` están disponibles las tres siguientes:

<code>\frontmatter</code>
<code>\mainmatter</code>
<code>\backmatter</code>

Entre `\frontmatter` y `\mainmatter` se incluyen las páginas preliminares del libro, tales como páginas de título, tabla de contenidos, tablas de figuras, dedicatoria, prefacio, etc. Entre `\mainmatter` y `\backmatter` va la materia principal del libro, incluyendo los capítulos y apéndices. Después de `\backmatter` van las páginas posteriores del libro, las cuales contienen cosas como la bibliografía, un índice alfabético, agradecimientos, epílogo y colofón.

Las páginas preliminares se numeran con números romanos, y las demás con números arábigos.

Los capítulos incluidos en las páginas preliminares o en las posteriores no se numeran, pero aparecen en el índice general.

### Clase `slide`

Se usa para preparar transparencias. Esta clase usa tipos grandes sans serif para una mejor visibilidad. La preparación de buenas transparencias requiere mucho trabajo de diseño visual que no es uno de los puntos fuertes de  $\text{\LaTeX}$ . Sin embargo si debe componer muchas fórmulas matemáticas esta clase puede resultar muy útil. Cada transparencia individual se genera con un ambiente `slide`. Dentro del mismo pueden usarse los comandos ordinarios de  $\text{\LaTeX}$  exceptuando aquellos que no tendrían sentido en una transparencia, como los de seccionamiento, nueva página y ambientes para tablas y figuras.

### Clase `letter`

La clase `letter` se usa para escribir cartas. El nombre y la dirección del remitente se declaran generalmente en el preámbulo, como argumentos de los comandos `\signature` y `\address`. Varias cartas pueden generarse con un mismo archivo de entrada, colocando cada una de ellas en un ambiente `letter`, con el nombre y dirección del destinatario como argumento. La fecha se genera automáticamente. El saludo inicial se coloca como argumento del comando `\opening`, y el final como argumento de `\closing`. Las personas a las que se les envía copia se especifican con el comando `\cc`.

## 2.9 Opciones de `\documentclass`

A continuación describimos las opciones disponibles para las clases estándar de  $\text{\LaTeX}$ . Las mismas deben incluirse como un argumento opcional, separadas por comas y sin espacios en blanco. Cuando hay varias alternativas para una opción las escribimos separadas por el caracter '|', con el valor por defecto para la clase `article` en primer lugar (algunas opciones tienen un valor por defecto que depende de la clase).

`10pt|11pt|12pt` Establecen el tamaño normal de los tipos para todo el documento. El valor por defecto es `10pt`.

`letterpaper|legalpaper|executivepaper|a4paper|a5paper|b5paper`  
Define el tamaño del papel.

`fleqn` Dispone las ecuaciones hacia la izquierda en vez de centradas.

`leqno` Coloca el número de las ecuaciones a la izquierda en vez de a la derecha.

`notitlepage|titlepage` Indica si el título del documento (generado por `\maketitle`) y el resumen (incluido en el ambiente `abstract`) deben componerse cada uno en una página separada. Por defecto, se usan

páginas separadas en las clases `report` y `book` pero no en la clase `article`.

`onecolumn|twocolumn` Le dice a  $\LaTeX$  si debe componer el documento a una o a dos columnas.

`oneside|twoside` Especifica si el documento se debe componer a una o a dos caras. Por defecto, la clase `book` es a dos caras y las otras son a una cara.

`openright|openany` Esta opción solamente se aplica a las clases `book` y `report`, en las cuales su valor por defecto es `openright` y `openany` respectivamente, y determina si los capítulos comenzarán siempre en páginas a la derecha, o bien en la próxima página disponible.

`landscape` Compone el documento en formato apaisado, intercambiando el ancho y la altura del tipo de papel seleccionado.

`final|draft` Si se selecciona `draft` las líneas demasiado largas (*overflow* *hbox*) se señalan con un rectángulo negro bien visible.

`openbib` Hace que la bibliografía se componga en estilo abierto.

Por ejemplo: un archivo de entrada podría comenzar con

```
\documentclass[11pt,twoside]{article}
```

Esto le indica a  $\LaTeX$  que componga el documento como un *artículo* utilizando tipos del cuerpo 11, y que produzca un formato para impresión a doble cara en papel tamaño carta.

## 2.10 Contadores

Un *contador* es una variable capaz de almacenar un número entero. Cada número generado por  $\LaTeX$  tiene un contador asociado. El nombre del contador es generalmente el mismo del ambiente o comando que lo genera, pero sin el *backslash* inicial. En las clases estándar están definidos los siguientes contadores: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`, `page`, `equation`, `figure`, `table`, `footnote`, `mpfootnote`, `enumi`, `enumii`, `enumiii`, `enumiiii`. Sus nombres son autodescriptivos, excepto tal vez `mpfootnote` que cuenta las notas al pie de una minipágina (ver la sección 4.1) y los cuatro últimos, que corresponden a los cuatro posibles niveles de un ambiente `enumerate`. Además cada ambiente definido con `\newtheorem` (ver sección 3.8) tiene un contador del mismo nombre, a menos que se haya especificado una numeración subordinada.

El comando

$$\boxed{\backslash\text{setcounter}\{ctr\}\{num\}}$$

asigna el valor *num* a *ctr*. En cambio

$$\boxed{\backslash\text{addtocounter}\{ctr\}\{num\}}$$

incrementa el valor actual de *ctr* sumándole *num*. El comando

$$\boxed{\backslash\text{value}\{ctr\}}$$

produce el valor numérico de *ctr*.

Los siguientes comandos representan el valor de *ctr* en diversos estilos:

`\arabic{ctr}` : con números arábigos.

`\roman{ctr}` : con números romanos en minúsculas.

`\Roman{ctr}` : con números romanos en mayúsculas.

`\alph{ctr}` : con letras minúsculas ( $ctr < 27$ ).

`\Alph{ctr}` : con letras mayúsculas ( $ctr < 27$ ).

`\fnsymbol{ctr}` : con uno de los símbolos usados para las notas al pie de página, a saber: \* † ‡ § ¶ || \*\* †† ‡‡. Sólo puede usarse en el modo matemático y *ctr* debe ser menor que 10.

Cada contador tiene asociado un comando que imprime su valor. El nombre de este comando se forma anteponiendo `\the` al nombre del contador. Por ejemplo el contador `\page` tiene asociado el comando `\thepage`.

El comando

$$\boxed{\backslash\text{stepcounter}\{ctr\}}$$

incrementa en uno el valor de *ctr* y reinicializa a 0 cualquier contador subordinado. Por ejemplo en la clase `book` el contador `subsection` está subordinado a `section` y éste a su vez lo está a `chapter`. Cada vez que se inicia una nueva sección el contador `subsection` se reinicializa a 0, y lo mismo ocurre con `section` cada vez que se inicia un nuevo capítulo. El comando `\refstepcounter` hace lo mismo, pero además establece el valor corriente de `\ref` como `\thectr`.

Es posible crear un nuevo contador inicializado a 0 con el comando

$$\boxed{\backslash\text{newcounter}\{ctr\}[subord]}$$

El argumento *ctr* debe estar compuesto sólo por letras y no puede ser el nombre de un contador ya existente. El argumento opcional debe ser el nombre de otro contador ya definido. En caso de estar presente, cada vez que *subord* se incremente con las órdenes `\stepcounter` o `\refstepcounter`, el nuevo contador será reinicializado a 0. El comando `\thectr` para un contador recién creado imprime el valor en números arábigos, pero esto se puede modificar como veremos en la próxima sección.

## 2.11 Definición de nuevos comandos

L<sup>A</sup>T<sub>E</sub>X permite definir nuevas instrucciones para satisfacer necesidades especiales. Para ello se utiliza el comando

$$\backslash\text{newcommand}\{\textit{nombre}\}[\textit{args}][\textit{opc}]\{\textit{defin}\}$$

El primer argumento obligatorio establece el *nombre* de la instrucción que se va a definir, el cual no puede comenzar con los caracteres `end`. El argumento opcional *args* es un entero entre 1 y 9 que especifica el número de argumentos que requiere la nueva instrucción (si es que requiere alguno). El argumento *opc*, si se incluye, significa que el primero de los *args* argumentos es opcional y su valor por defecto es *opc*. Finalmente, *defin* es la definición del nuevo comando.

Como un primer ejemplo sencillo supongamos que la frase “espacio topológico de Hausdorff” se repite muchas veces en nuestro documento. Entonces podemos ahorrar tiempo y espacio definiendo un comando `\etH` que la genere:

<pre>\newcommand{\etH} {espacio topol'ogico de Hausdorff} % en el cuerpo del documento : Sea \$\$X\$ un \etH\ conexo\ldots Entonces para todo \etH\ compacto \$\$Y\$ se cumple que \ldots</pre>	<p>Sea <math>X</math> un espacio topológico de Hausdorff conexo ... Entonces para todo espacio topológico de Hausdorff compacto <math>Y</math> se cumple que ...</p>
---	--

En el siguiente ejemplo definimos el comando `\p` mencionado en la sección 1.2 (página 5). Observe que los *parámetros* `#1` y `#2` se substituyen por el primer y el segundo argumentos respectivamente.

<pre>% en el preambulo: \newcommand{\p}[2] {\langle #1,#2 \rangle} % en el cuerpo del documento: Si \$\p{u}{v}=0\$ entonces \$u\$ y \$v\$ se dicen \emph{ortogonales}.</pre>	<p>Si <math>\langle u, v \rangle = 0</math> entonces <math>u</math> y <math>v</math> se dicen <i>ortogonales</i>.</p>
--	---

Si el nombre elegido para una nueva instrucción ya está en uso  $\LaTeX$  generará un mensaje de error. Si realmente se desea redefinir un comando ya existente entonces hay que usar la instrucción `\renewcommand`, cuya sintaxis es idéntica a la de `\newcommand`. Obviamente la redefinición de comandos básicos puede tener toda clase de efectos secundarios indeseados, y no debe intentarse a menos que se sepa muy bien lo que se está haciendo.

Como ejemplo supongamos que necesitamos hacer una lista enumerada (usando el ambiente `enumerate`) pero queremos que los items del primer nivel se numeren (i), (ii), (iii), (iv), etc. Esto se puede hacer fácilmente con el paquete `enumerate`, pero si no lo tenemos o no queremos usarlo podemos redefinir los comandos usados por  $\LaTeX$ . El contador del primer nivel se llama `enumi` y el comando que genera el número es `\theenumi`. Lo redefiniremos para que genere números romanos en minúsculas con la orden `\roman`. La etiqueta `\labelenumi` también la redefiniremos para poner paréntesis alrededor del número. En definitiva escribiremos:

<code>\renewcommand{\theenumi}</code>	Si $X$ es finito y $f : X \rightarrow X$
<code>{\roman{enumi}}</code>	entonces:
<code>\renewcommand{\labelenumi}</code>	(i) $f$ es inyectiva
<code>{(\theenumi)}</code>	(ii) $f$ es sobre
Si $X$ es finito y $f : X \rightarrow X$	(iii) $f$ es biyectiva
entonces:	son equivalentes.
<code>\begin{enumerate}</code>	
<code>\item <math>f</math> es inyectiva</code>	
<code>\item <math>f</math> es sobre</code>	
<code>\item <math>f</math> es biyectiva</code>	
<code>\end{enumerate}</code>	
son equivalentes.	

## 2.12 Definición de nuevos ambientes

Es posible crear nuevos ambientes con la instrucción

`\newenvironment{nombre}[args]{prologo}{epilogo}`

Lo indicado en el *prologo* se procesa antes que lo que pongamos en el ambiente, y el *epilogo* se procesa al encontrar la instrucción `\end{nombre}`.

Por ejemplo supongamos que queremos crear un ambiente para componer títulos centrados, con tipos Sans Serif grandes. Podemos hacerlo así:



```

\newenvironment{titulo}
{\begin{center}
\Large \sffamily}
{\end{center}}
% en el cuerpo del documento:
\begin{titulo}
La Metafísica\de lo Imaginario
\end{titulo}

```

## La Metafísica de lo Imaginario

El argumento *args* se utiliza igual que en la instrucción `\newcommand` para indicar el número de argumentos, si los hay. Para redefinir un ambiente ya existente se usa la instrucción `\renewenvironment` (con la misma sintaxis de `\newenvironment`).

### 2.13 Ejercicios

**Ejercicio 2.1.** Componga la siguiente frase usando comandos:

**Horrible** mezcla DE estilos.

**Ejercicio 2.2.** Componga la siguiente frase usando declaraciones:

Diminuendo y *crescendo*.

**Ejercicio 2.3.** ¿Qué comando se puede usar para imprimir un documento a *espacio y medio*?

**Ejercicio 2.4.** Componga la siguiente tabla:

$\oplus$	0	1
0	0	1
1	1	0

**Ejercicio 2.5.** Defina un comando que haga una plana con la frase “Para aprender debo hacer todos los ejercicios.” repetida cien veces.



## Capítulo 3

# Matemática

En el apartado 1.6.6 del primer Capítulo ya vimos los conceptos básicos de la composición de expresiones matemáticas con  $\text{\LaTeX}$ . En este Capítulo nos ocuparemos con mayor profundidad del modo matemático y estudiaremos algunas de las útiles extensiones proporcionadas por los paquetes de la American Mathematical Society.

### 3.1 Ecuaciones

Recordemos que  $\text{\LaTeX}$  posee dos modos especiales para componer expresiones matemáticas. Dentro de un párrafo, estas expresiones se introducen entre  $\$$  y  $\$$ , o bien entre  $\backslash($  y  $\backslash)$  o entre  $\backslash\text{\begin{math}}$  y  $\backslash\text{\end{math}}$ . Las expresiones grandes y las ecuaciones quedan mejor *desplegadas*, para lo cual se ponen entre  $\backslash[$  y  $\backslash]$ , entre  $\text{\$}$  y  $\text{\$}$  o entre  $\backslash\text{\begin{displaymath}}$  y  $\backslash\text{\end{displaymath}}$ . Esto produce expresiones no numeradas. Si desea que  $\text{\LaTeX}$  las numere consecutivamente, se usa el ambiente `equation`. Con `\label` se puede asociar un nombre (o etiqueta) a una ecuación, para luego referenciarla con el comando `\ref`.

Por lo tanto  
`\begin{equation} \label{eq2}`  
 $x^2 - 5x + 6 = 0$   
`\end{equation}`  
`\ldots` y como hemos establecido  
en la ecuación (`\ref{eq2}`) `\ldots`

Por lo tanto  
$$x^2 - 5x + 6 = 0 \quad (3.1)$$
  
... y como hemos establecido en la  
ecuación (3.1) ...

Para los sistemas de ecuaciones se puede usar el ambiente `eqnarray`, el cual funciona como una tabla de tres columnas con alineación `{rcl}`. Las columnas se separan con el carácter `&` y las líneas con `\backslash`. Cada línea recibe un número, a menos que se coloque en ella el comando `\nonumber`. El ambiente `eqnarray*` es similar, excepto que no se numeran las líneas.

```

\begin{eqnarray*}
2x+y-4z & = & 1 \\
3x-2y+z & = & -5 \\
-x+4y+2z & = & 3
\end{eqnarray*}

```

$$\begin{array}{rcl}
2x + y - 4z & = & 1 \\
3x - 2y + z & = & -5 \\
-x + 4y + 2z & = & 3
\end{array}$$

Estos ambientes se pueden usar también para acomodar ecuaciones largas que no quepan en una sola línea. Veamos un ejemplo:

```

\begin{eqnarray}
e^x & = & 1+x+\frac{x^2}{2!} \\
& + & \frac{x^3}{3!} \\
& + & \frac{x^4}{4!} + \cdots
\end{eqnarray}

```

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \quad (3.2)$$

El grupo vacío {} que precede al término  $x^5/5!$  es para obtener un espaciado correcto alrededor del operador + (que de lo contrario podría interpretarse como un signo positivo para el término siguiente).

El paquete `amsmath` incluye varios ambientes para componer sistemas de ecuaciones y ecuaciones largas, con o sin alineación. El ambiente `split` se puede usar para dividir ecuaciones largas. Por sí mismo no genera numeración, ya que está pensado para usarlo dentro de otro ambiente como por ejemplo `\equation`. Se usa un caracter & para obtener alineación y \\ para separar líneas. Compare el próximo ejemplo con el anterior.

```

\begin{equation}
\begin{split}
e^x & = 1+x+\frac{x^2}{2!} \\
& + \frac{x^3}{3!} \\
& + \frac{x^4}{4!} + \cdots
\end{split}
\end{equation}

```

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \quad (3.3)$$

Otros ambientes proporcionados por el paquete `amsmath` son: `align`, `alignat` y `flalign` (con alineación); `gather` y `multline` (sin alineación). Todos ellos admiten variantes con asterisco, para suprimir la numeración automática. En los ejemplos siguientes se ilustran algunos de estos ambientes y el uso de las instrucciones `\tag` y `\notag` para sustituir o eliminar la numeración estándar (para más detalles vea el documento `amslldoc.dvi` incluido con el paquete `amsmath`).

```

\begin{multline}
a+b+c+d+e+f \\
+i+j+k+l+m \\
+n+o+p+q
\end{multline}

```

$$\begin{array}{l}
a + b + c + d + e + f \\
+ i + j + k + l + m \\
+ n + o + p + q
\end{array} \quad (3.4)$$

```
\begin{gather}
a_1=b_1+c_1 \notag \\
a_2=b_2+c_2-d_2+e_2 \\
\end{gather}
```

$$a_1 = b_1 + c_1$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (3.5)$$

```
\begin{align*}
a_{11}&=b_{11}& a_{11} &= b_{11} & a_{12} &= b_{12} & (*) \\
a_{12}&=b_{12} \tag{**} \\
a_{21}&=b_{21} & a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} \\
a_{22}&=b_{22}+c_{22} \\
\end{align*}
```

## 3.2 Texto y espacios

En el modo matemático las letras se componen en un tipo especial y son parecidas pero no iguales a las itálicas del modo de texto. En particular el espaciado entre letras es diferente, apropiado para símbolos matemáticos pero muy grande para texto corriente. Por tal razón cuando se necesita intercalar texto en el modo matemático hay que utilizar algún comando especial. En  $\text{\LaTeX}$  estándar se puede usar  $\text{\mbox}{texto}$ , pero es preferible el comando  $\text{\text}{texto}$  suministrado por el paquete  $\text{\amsmath}$ , ya que además de su nombre más descriptivo ajusta automáticamente el tamaño de los tipos.

El espaciado en modo matemático se determina a partir de la lógica de la expresión. Para introducir espacio adicional hay que usar instrucciones especiales como  $\text{\}$ , (un pequeño espacio),  $\text{\:}$  (un espacio medio),  $\text{\quad}$  (espacio correspondiente a la separación entre palabras, donde  $\text{\quad}$  representa el carácter blanco),  $\text{\;}$  (espacio grueso),  $\text{\quad}$  (un espacio algo mayor),  $\text{\quad}$  (un espacio aun mayor) y  $\text{\!}$  (un pequeño espacio negativo, que tiene el efecto de disminuir una separación). En general es preferible dejar que  $\text{\LaTeX}$  determine el espaciado adecuado. Sin embargo hay ciertos casos en los cuales necesita una pequeña ayuda de nuestra parte. El ejemplo más común lo proporcionan las integrales:

```
\[
\int\!\!\!\int\int_{\Omega} dx\,dy \quad \iint_{\Omega} dx dy \quad \text{mejor que} \quad \int \int_{\Omega} dx dy
\quad \text{\quad \text{mejor que} \quad \quad}
\int\int_{\Omega} dx dy
\]
```

(El paquete  $\text{\amsmath}$  proporciona los comandos  $\text{\iint}$ ,  $\text{\iiint}$  y  $\text{\iiiiint}$  para integrales dobles, triples y cuádruples, con espaciado adecuado tanto dentro de un párrafo como en el modo desplegado.)



para puntos verticales y `\ddots` para puntos en diagonal. En el apartado 3.7 podrá encontrar otro ejemplo.

```
\begin{displaymath}
x_{1}, \dots, x_{n} \quad \quad \quad x_{1} + \dots + x_{n}
x_{1} + \dots + x_{n}
\end{displaymath}
```

El signo de raíz *cuadrada* se introduce con `\sqrt`, y la raíz *n*-ésima con `\sqrt[n]`. L<sup>A</sup>T<sub>E</sub>X elige automáticamente el tamaño adecuado del signo.

```
\[
\sqrt{\sqrt{\sqrt{x}}} = \sqrt[8]{x}
\]
```

$$\sqrt{\sqrt{\sqrt{x}}} = \sqrt[8]{x}$$

### 3.4 Operadores y relaciones

T<sub>E</sub>X reconoce el rol de algunos signos matemáticos y proporciona el espacio adecuado en torno a ellos. Por ejemplo reconoce que `+` es una operación binaria mientras que `=` es una relación binaria.

Los nombres de algunas operaciones y funciones matemáticas se deben escribir en tipo romano (vertical). Para esto L<sup>A</sup>T<sub>E</sub>X proporciona las siguientes instrucciones:

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>	<code>\sup</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>

El paquete `amsmath` permite aplicar este mismo tratamiento a otros nombres de función con la instrucción

`\DeclareMathOperator{nombre del operador}`

La variante con asterisco de este comando permite definir operadores que admiten subíndices y superíndices en posición de límites, como ocurre con `lim`, `sup`, etc.

El operador *módulo* se genera con el comando `\bmod`, como en “*a mod b*”. Para escribir congruencias se utilizan los comandos `\equiv` y `\pmod`, como en el ejemplo siguiente:

```
$ x^p \equiv x \pmod{p} $
```

$$x^p \equiv x \pmod{p}$$

El siguiente ejemplo ilustra algunos de los comandos disponibles para trabajar con conjuntos:

```
\[
x \in X, \ y \not\in Y, \
A \cap (B \cup C), \ U \subset V
\]
```

$$x \in X, y \notin Y, A \cap (B \cup C), U \subset V$$

En el Apéndice B se encuentran tablas completas de los operadores y relaciones binarias disponibles.

### 3.5 Tamaños y Tipos

Existen varios tipos especiales para ser usados en el modo matemático. Algunos de ellos son estándar en L<sup>A</sup>T<sub>E</sub>X mientras que otros requieren paquetes adicionales. En la Tabla 3.1 se muestran los más importantes.

Tabla 3.1: Tipos en el modo matemáticos

<i>Orden</i>	<i>Ejemplo</i>	<i>Resultado</i>
<code>\mathrm</code>	<code>\mathrm{Supp}(f)</code>	$\text{Supp}(f)$
<code>\mathbf</code>	<code>\mathbf{x}=(x_1,x_2)</code>	$\mathbf{x} = (x_1, x_2)$
<code>\mathsf</code>	<code>\mathsf{G \times H}</code>	$G \times H$
<code>\mathtt</code>	<code>\mathtt{a,b,c}</code>	$a, b, c$
<code>\mathnormal</code>	<code>\mathnormal{f(x)=x^2+1}</code>	$f(x) = x^2 + 1$
<code>\mathit</code>	<code>\mathit{123}</code>	$123 = 123$
<code>\mathcal</code>	<code>\mathcal{A,H,L}</code>	$\mathcal{A}, \mathcal{H}, \mathcal{L}$
<code>\mathbb</code> <sup>a</sup>	<code>\mathbb{Q,R,Z}</code>	$\mathbb{Q}, \mathbb{R}, \mathbb{Z}$
<code>\mathfrak</code> <sup>a</sup>	<code>\mathfrak{Euler}</code>	$\mathfrak{Euler}$
<code>\mathscr</code> <sup>b</sup>	<code>\mathscr{A,H,L}</code>	$\mathcal{A}, \mathcal{H}, \mathcal{L}$

<sup>a</sup>Requiere el paquete `amsfonts`

<sup>b</sup>Requiere el paquete `mathrsfs`

El tamaño de los tipos en el modo matemático se selecciona automáticamente según el contexto: las expresiones desplegadas se componen con tipos mayores que las incluidas en un párrafo de texto, los subíndices y superíndices se componen en un tipo más pequeño, etc. Sin embargo, a veces es preciso indicarle a L<sup>A</sup>T<sub>E</sub>X el tamaño del tipo que se desea. La Tabla 3.2 detalla los tamaños disponibles en el modo matemático.



Tabla 3.2: Tamaños en modo matemático

<i>Declaración</i>	<i>Ejemplo</i>	<i>Resultado</i>
<code>\displaystyle</code>	<code>\$\$\displaystyle\int f(x)dx\$</code>	$\int f(x)dx$
<code>\textstyle</code>	<code>\$\$\textstyle\int f(x)dx\$</code>	$\int f(x)dx$
<code>\scriptstyle</code>	<code>\$\$\scriptstyle\int f(x)dx\$</code>	$\int f(x)dx$
<code>\scriptscriptstyle</code>	<code>\$\$\scriptscriptstyle\int f(x)dx\$</code>	$\int f(x)dx$

### 3.6 Disposición vertical de símbolos

Una de las dificultades para componer expresiones matemáticas proviene de que los símbolos no aparecen siempre en una secuencia lineal, de izquierda a derecha, sino que suelen presentarse apilados unos encima de otros. en diversas configuraciones.

Es muy común aplicar *acentos matemáticos* como por ejemplo puntos, pequeñas flechas o tildes a las variables. Para ello no se deben usar las instrucciones estudiadas en el capítulo precedente, sino las que aparecen en la tabla 3.3.

Tabla 3.3: Acentos matemáticos

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>	$\acute{a}$	<code>\acute{a}</code>
$\grave{a}$	<code>\grave{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\widehat{A}$	<code>\widehat{A}</code>	$\widetilde{A}$	<code>\widetilde{A}</code>

Notemos que `\widehat` y `\widetilde` generan acentos circunflejos y tildes extensibles, que pueden abarcar varios caracteres:

```
\[
\widehat{G\times H}, \quad \widehat{G\times H}, \quad \widetilde{abc}
\widetilde{abc}
\]
```

Para el símbolo de prima simplemente se coloca un apóstrofo:

```
$y=x^2+5x+6$
$y'=2x+5, \quad y''=2$
```

$$y = x^2 + 5x + 6$$

$$y' = 2x + 5, \quad y'' = 2$$

Los subíndices y superíndices se obtienen, como vimos en el primer capítulo, con los operadores `_` y `^` respectivamente. A veces es necesario

un índice de varias líneas: para estos casos `amsmath` incluye el comando `\substack`.

$$\begin{array}{l} \backslash [ \sum_{\substack{1 \leq d \leq n \\ d|n}} \phi(d) = n \backslash \\ \end{array} \qquad \sum_{\substack{1 \leq d \leq n \\ d|n}} \phi(d) = n$$

Otro comando disponible con `amsmath` es `\sideset`, que permite colocar índices tanto a la derecha como a la izquierda de un “operador grande”.

$$\backslash [ \sideset{a^b}{c^d} \prod \backslash \qquad \prod_a^b c^d$$

La colocación de los límites en sumatorias e integrales se puede modificar con las instrucciones `\limits` y `\nolimits`, como muestra el siguiente ejemplo:

$$\begin{array}{l} \backslash [ \int_0^1 f(x) \, dx = \\ \int \limits_0^1 f(x) \, dx, \\ \quad \quad \quad \backslash quad \sum \nolimits^* \backslash \\ \end{array} \qquad \int_0^1 f(x) \, dx = \int_0^1 f(x) \, dx, \quad \sum^*$$

Las flechas largas colocadas encima de varios caracteres se hacen con las instrucciones `\overrightarrow` y `\overleftarrow`:

$$\$ \vec{u} = \overrightarrow{AB} \$ \qquad \vec{u} = \overrightarrow{AB}$$

Las instrucciones `\overline` coloca una raya horizontal por encima de su argumento, mientras que `\underline` puede utilizarse para subrayar, tanto en el modo matemático como fuera de él.

$$\begin{array}{l} \$ \overline{a+b} = \\ \overline{a} + \overline{b} \$ \\ \$ x^{\underline{n}} = \\ x(x-1) \dots (x-n+1) \$ \end{array} \qquad \begin{array}{l} \overline{a+b} = \overline{a} + \overline{b} \\ x^{\underline{n}} = x(x-1) \dots (x-n+1) \end{array}$$

Las órdenes `\overbrace` y `\underbrace` crean *llaves horizontales* largas encima o debajo de una expresión, respectivamente.

$$\begin{array}{l} \$ \underbrace{999 \dots 999}_ \\ {99 \text{ nueves}} = 10^{100} - 1 \$ \end{array} \qquad \underbrace{999 \dots 999}_{99 \text{ nueves}} = 10^{100} - 1$$

A veces es necesario colocar un símbolo sobre una relación binaria, como por ejemplo el nombre de una función arriba de una flecha. En estos casos se puede usar la instrucción `\stackrel`, como en el siguiente ejemplo:

```
$\stackrel{\phi}{\longrightarrow} B, \quad x \stackrel{\Delta}{=} y$
```

Observe que el primer argumento de `\stackrel` (el que va arriba) se compone en un tamaño más pequeño (`\texttt{scriptsize}`). Esto puede modificarse si es necesario con los comandos de la Tabla 3.2.

Con `amsmath` es posible un tratamiento más general: los comandos `\overset` y `\underset` colocan *cualquier* símbolo encima o debajo de otro:

```
$\overset{*}{X} = \underset{\leftarrow}{\Psi}$
```

Una *fracción* se compone con la orden

```
\frac{numerador}{denominador}
```

Sin embargo para fracciones sencillas dentro de un párrafo de texto suele ser preferible el uso de la diagonal `/`, como en `1/2`.

El paquete `amsmath` proporciona un comando generalizado para fracciones:

```
\genfrac{del-i}{del-d}{grosor}{estilo}{num}{den}
```

Los argumentos `del-i` y `del-d` especifican los delimitadores que se usarán a la izquierda y la derecha, respectivamente (naturalmente se puede escoger un delimitador vacío); `grosor` es el espesor de la línea horizontal (puede indicarse `0pt` si no se desea línea alguna); `estilo` puede ser 0, 1, 2 o 3 que significan, respectivamente, `\displaystyle`, `\textstyle`, `\scriptstyle` y `\scriptscriptstyle`; `num` es el numerador de la fracción y `den` el denominador.

Por ejemplo podríamos definir un comando para generar números de Stirling de segunda clase (que se escriben como los coeficientes binomiales pero entre llaves en vez de paréntesis) así:

```
\newcommand{\Ss}[2]
{\genfrac{\}{\}{0pt}{\#1}{\#2}}
% y en el cuerpo del documento:
\[ \Ss{n}{n-1} = \binom{n}{2} \]
```

Observemos que el coeficiente binomial que aparece en el ejemplo anterior se podría haber generado con `\genfrac`, pero en realidad `amsmath` ya tiene predefinido el comando `\binom`.

### 3.7 Matrices y construcciones similares

Para componer matrices y estructuras semejantes se usa el ambiente `array`, el cual se asemeja al ambiente `tabular`. El caracter `&` se usa como tabulador y `\\` se emplea para iniciar una nueva línea.

```
\[
A= \left( \begin{array}{cccc}
x_{11} & x_{12} & \dots & x_{1n} \\
x_{21} & x_{22} & \dots & x_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \dots & x_{nn}
\end{array} \right)
```

$$A = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{pmatrix}$$

También se puede usar el ambiente `array` para componer expresiones de funciones que tengan definiciones distintas en intervalos separados. Esto se hace utilizando un delimitador derecho invisible, como en el siguiente ejemplo:

```
\begin{displaymath}
f(x) = \left\{ \begin{array}{ll}
1 & \mbox{si } x \in \mathbf{Q} \\
0 & \mbox{si } x \notin \mathbf{Q}
\end{array} \right.
```

$$f(x) = \begin{cases} 1 & \text{si } x \in \mathbb{Q} \\ 0 & \text{si } x \notin \mathbb{Q} \end{cases}$$

```
\end{array} \right.
\end{displaymath}
```

Pero con el paquete `amsmath` esto se logra más fácilmente usando el ambiente `cases`.

```
\begin{displaymath}
f(x) = \begin{cases}
0 & \text{si } x \in \mathbf{Q} \\
1 & \text{si } x \notin \mathbf{Q}
\end{cases}
```

$$f(x) = \begin{cases} 0 & \text{si } x \in \mathbb{Q} \\ 1 & \text{si } x \notin \mathbb{Q} \end{cases}$$

```
\end{cases}
\end{displaymath}
```

## 3.8 Definiciones, teoremas y demostraciones

Dada la importancia que las definiciones, teoremas y demostraciones tienen en matemática es natural que en  $\text{\LaTeX}$  haya ambientes especiales para este tipo de construcciones, sin olvidarse de los axiomas, postulados, lemas, proposiciones, corolarios, observaciones, etc. La instrucción básica, que se debe utilizar en el preámbulo, es

```
\newtheorem{clave}[contador]{nombre}[sección]
```

El argumento *clave* es una palabra corta que se utiliza para identificar el tipo de “teorema”. Generalmente se usan palabras como **teo**, **defi**, **prop**, **coro**, etc. El argumento *nombre* define el nombre del tipo de teorema que aparecerá impreso. Los argumentos opcionales se usan para especificar la forma de numerar. El argumento *contador* permite especificar el *nombre* de un “teorema” declarado previamente, para numerar al nuevo con la misma secuencia. El argumento *sección* permite numerar los teoremas dentro de la sección indicada (o capítulo, o subsección).

Cada `\newtheorem` que definamos activa un nuevo ambiente con *clave* como nombre. Este ambiente admite un argumento opcional para agregar algo al tipo de teorema y al número (que se generarán automáticamente). Si por ejemplo hemos definido en el preámbulo

```
\newtheorem{teo}[chapter]
```

Entonces en el cuerpo del documento se puede escribir:

<pre>\begin{teo}[Teorema de Euler] Si <math>(n,a)=1</math> entonces <math>a^{\phi(n)} \equiv 1 \pmod{n}</math>. \end{teo}</pre>	<p><b>Teorema 3.1 (Teorema de Euler).</b>  <i>Si <math>(n, a) = 1</math> entonces <math>a^{\phi(n)} \equiv 1 \pmod{n}</math>.</i></p>
---	---

El paquete `amsthm` ofrece la variante `\newtheorem*` para generar teoremas sin numeración.

El lector observará que el texto del enunciado de los teoremas aparece resaltado. Esto es bueno para teoremas, lemas, proposiciones y corolarios, pero no lo es para las definiciones y observaciones. El paquete `amsthm` ofrece una solución con el comando

```
\theoremstyle{estilo}
```

donde el argumento *estilo* puede tomar los valores `plain`, `definition` y `remark`. Cada estilo recibe un tratamiento tipográfico distinto. Solamente en el estilo `plain` se resalta el texto en itálicas. Para usar este comando divida los comandos `\newtheorem` en grupos y preceda cada grupo del comando `\theoremstyle` apropiado.

El paquete `amsthm` ofrece también el ambiente `proof` para escribir demostraciones. Este ambiente comienza con la palabra *Proof* (o su equivalente en el lenguaje que esté seleccionado, si se está usando `babel`), pero esto puede cambiarse por cualquier otro texto que se especifique como argumento opcional. Al finalizar la demostración se imprime el símbolo  $\square$ , el cual también puede cambiarse por otro si se desea redefiniendo `\qedsymbol` con `\renewcommand`.

<code>\begin{proof}</code>	<i>Demostración.</i> Es absolutamente obvia.	
Es absolutamente obvia.		$\square$
<code>\end{proof}</code>		

### 3.9 Ejercicios

En cada uno de los ejercicios siguientes se da una expresión matemática que usted debe componer en  $\text{\LaTeX}$ . Recorra a las Tablas del Apéndice B si necesita algún símbolo especial.

**Ejercicio 3.1.**

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x^2} = \frac{1}{2}$$

**Ejercicio 3.2.**

$$\sqrt[n]{\frac{1 \cdot 3 \cdots (2n-1)}{2 \cdot 4 \cdots 2n}}$$

**Ejercicio 3.3.**

$$\Gamma(z) = \frac{e^{-\gamma z}}{z} \prod_{n=1}^{\infty} \left(1 + \frac{z}{n}\right)^{-1} e^{z/n}$$

**Ejercicio 3.4.**

$$\int_S (\nabla \times \mathbf{F}) \cdot \mathbf{N} \, dS = \oint_{\partial S} \mathbf{F} \cdot d\mathbf{P}$$

**Ejercicio 3.5.**

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} + V(x)\Psi(x, t) = \tilde{H}\Psi(x, t)$$

**Ejercicio 3.6.**

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{\emptyset \neq J \subset \{1, 2, \dots, n\}} (-1)^{|J|-1} \left| \bigcap_{j \in J} A_j \right|$$

**Ejercicio 3.7.**

$$\begin{vmatrix} 1 & 1 & 1 \\ x & y & z \\ x^2 & y^2 & z^2 \end{vmatrix} = (x-y)(y-z)(z-x)$$

**Ejercicio 3.8.**

$$\begin{aligned} \nabla \cdot \mathbf{E} &= 4\pi\rho \\ \nabla \times \mathbf{E} &= \mathbf{0} \\ \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{B} &= \frac{4\pi}{c} \mathbf{J} \end{aligned}$$

**Ejercicio 3.9.**

$$\begin{array}{ccc} A & \xrightarrow{\phi} & B \\ \psi \downarrow & & \uparrow \xi \\ C & \xrightarrow{\theta} & D \end{array}$$

**Ejercicio 3.10.** Para este ejercicio probablemente necesitará el comando `\raisebox`, que se explica en el capítulo siguiente.

$$2 \uparrow \uparrow k \stackrel{def}{=} 2^{2^{\cdot^{\cdot^{\cdot^2}}}} \}^k .$$





## Capítulo 4

# L<sup>A</sup>T<sub>E</sub>X avanzado

### 4.1 Cajas

El concepto de *caja* es básico en T<sub>E</sub>X. En cierta forma, todo lo que T<sub>E</sub>X hace cuando procesa un documento es componer, alinear y apilar cajas, uniéndolas con un elemento transparente y elástico llamado *goma*. Todo esto ocurre “detrás del escenario” y el usuario promedio de L<sup>A</sup>T<sub>E</sub>X ni siquiera lo nota. Sin embargo en ciertas ocasiones, para lograr efectos especiales que escapan al ámbito de los comandos de alto nivel, es necesario manipular cajas directamente.

Las cajas tienen forma rectangular. T<sub>E</sub>X las trata como si fuesen un único carácter: no divide su contenido entre líneas ni entre páginas, ni las estira ni las comprime. Cada caja tiene un *punto de referencia* en su borde izquierdo que se usa para alinear las cajas. La *altura* de una caja se mide desde el punto de referencia hasta el borde superior. La distancia del punto de referencia al borde inferior se llama *profundidad*.

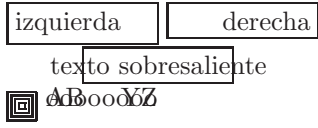
En L<sup>A</sup>T<sub>E</sub>X se puede crear una caja que no abarque más de una línea con el comando

```
\makebox[ancho][pos]{contenido}
```

cuyo primer argumento opcional especifica el ancho de la caja (si se omite, el ancho será el estrictamente necesario para acomodar el *contenido*). Por defecto el contenido aparecerá centrado en la caja, pero puede ajustarse a la izquierda o a la derecha usando *l* o *r*, respectivamente, como valor del segundo argumento opcional. Un valor de *s* hará que el espacio entre palabras se estire o comprima tratando de ajustar el texto al ancho especificado. El comando `\mbox` que mencionamos en el Capítulo 3 no es más que una versión abreviada de `\makebox`, sin argumentos opcionales. El comando `\framebox` es idéntico a `\makebox`, excepto que dibuja un marco alrededor de la caja. También tiene una variante sin argumentos opcionales llamada `\fbox`. El

ancho de los trazos del marco está regulado por el parámetro `\fboxrule` y la separación entre el marco y el borde de la caja por `\fboxsep`. El siguiente ejemplo ilustra el uso de los argumentos opcionales. Las dos primeras cajas tienen el mismo ancho y altura, pero diferente profundidad ya que la segunda debe acomodar la “patita” de la q.

```
\framebox[2cm][l]{izquierda}
\framebox[2cm][r]{derecha}\
\makebox[1cm]{}%
\framebox[2cm]{texto sobresaliente}\
\framebox[2cm]{texto sobresaliente}\
\makebox[0pt][l]{AB}\makebox[0pt][r]{YZ}
```



En la segunda línea hay una caja vacía de un centímetro de ancho (que es lo mismo que introducir un espacio de esa longitud) seguida de otra caja que muestra lo que ocurre cuando el contenido rebasa las dimensiones indicadas por el primer argumento. La tercera línea muestra un dibujo realizado con cuatro marcos y finaliza mostrando como se pueden usar cajas de ancho nulo para sobreimprimir.

Para crear cajas con más de una línea de contenido se puede usar el comando

```
\parbox[pos]{ancho}{contenido}
```

El argumento opcional *pos* puede tomar los valores *t* o *b*, los cuales hacen que el punto de referencia de la caja sea el de la primera línea (*top*) o el de la última (*bottom*). Por defecto la caja aparecerá centrada respecto a la línea en la cual se encuentre. El comando `\parbox` se usa para pequeñas porciones de texto. Para textos más extensos se usa el ambiente `minipage`.

```
\begin{minipage}[pos]{ancho}
contenido
\end{minipage}
```

Este ambiente acepta listas enumeradas y notas al pie, como se ve en el siguiente ejemplo.

```
\begin{minipage}{10pc}
Seg'un Arist'oteles\footnote{
\emph{Categor'ias}, cap'itulo VI}
la \emph{cantidad} puede ser
\begin{enumerate}
\item discreta, o
\item continua.
\end{enumerate}
\end{minipage}
```

Según Aristóteles<sup>a</sup> la *canti-*  
*dad* puede ser

1. discreta, o
2. continua.

---

<sup>a</sup>*Categorías*, capítulo VI

Una *regla* es una caja completamente negra, con el punto de referencia en su vértice inferior izquierdo. Se pueden usar para trazar líneas horizontales o verticales, o pequeños rectángulos negros como los que se suelen colocar al finalizar una demostración. Se generan con el comando

$$\boxed{\backslash\text{rule}[subir]{ancho}{altura}}$$

El argumento opcional debe ser una longitud y se usa para subir (si es positivo) o bajar (si es negativo) la regla respecto a la línea base. Una regla de ancho nulo se denomina *strut* y naturalmente es invisible. Sin embargo su altura puede usarse para obtener espacio vertical en ciertas situaciones. En el siguiente ejemplo la altura del marco y el desproporcionado tamaño de la raíz cuadrada son consecuencia de la presencia de un *strut*.

```

\rule{1cm}{.3pt}
\rule[-5pt]{3pt}{5pt}\rule{4pt}{5pt}
\rule[-15pt]{.5pt}{30pt}
\fbbox{\rule[-15pt]{0pt}{30pt}strut}
$\sqrt{\rule[-15pt]{0pt}{30pt}x}$

```

El comando

$$\boxed{\backslash\text{raisebox}[subir][altura][profundidad]{contenido}}$$

crea una caja y la coloca a una distancia *subir* por encima de la línea base (o por debajo si *subir* es negativa). Sus dos argumentos opcionales son para *hacer creer* a T<sub>E</sub>X que el resultado final tendrá la *altura* y *profundidad* indicadas.

```

Esto es una s\raisebox{.6ex}{i}%      Esto es una sinuoide.
\raisebox{1ex}{n}\raisebox{.7ex}{u}s%
\raisebox{-.7ex}{o}\raisebox{-1ex}{i}%
\raisebox{-.7ex}{d}e.

```

Una caja puede guardarse para un uso posterior. Para ello se le debe asignar un nombre mediante el comando

$$\boxed{\backslash\text{newsavebox}{cmd}}$$

donde *cmd* puede ser cualquier nombre válido de secuencia de control que no haya sido previamente definido. Luego se usa

$$\boxed{\backslash\text{savebox}{cmd}[ancho][pos]{contenido}}$$

que es análogo a `\makebox` pero no imprime nada, sino que guarda el resultado en *cmd*. Para usarlo, se emplea el comando

$$\boxed{\code{\usebox{cmd}}}$$

El comando `\sbox` es la forma abreviada de `\savebox`, sin argumentos opcionales. También existe un ambiente `lrbox` que es equivalente a `\savebox`, excepto que no toma en cuenta el espacio que se encuentre al comienzo y al final del contenido.

Los comandos `\width`, `\height`, `\depth` y `\totalheight` solamente pueden ser usados en el argumento *ancho* de `\makebox`, `\framebox` y `\savebox` y en los argumentos *subir*, *altura* y *profundidad* de `\raisebox`. Se refieren naturalmente a las dimensiones del contenido de estas instrucciones, a saber el ancho, la altura, la profundidad y la suma de altura y profundidad, respectivamente.

## 4.2 Gráficas sencillas

L<sup>A</sup>T<sub>E</sub>X tiene la posibilidad de generar algunas gráficas sencillas con el entorno `picture`. Las coordenadas y distancias usadas en este entorno utilizan la unidad de medida determinada por `\unitlength`. Por defecto esta unidad es igual a 1 punto (aproximadamente 1/72 de pulgada o 0,35 milímetros) pero puede cambiarse con el comando:

$$\boxed{\code{\setlength{\unitlength}{unidad}}}$$

El espesor de las líneas se especifica con los comandos `\thinlines` (que es el valor por defecto) y `\thicklines`, que produce un trazo más grueso.

El entorno `picture` requiere como argumento un par de números entre paréntesis para indicar las dimensiones de la figura. Un segundo par ordenado es opcional y se usa para especificar las coordenadas de la esquina inferior izquierda, que por defecto es (0,0). Los comandos que se pueden utilizar en el entorno `picture` son `\put`, `\multiput`, `\qbezier` y `\graphpaper`, además de declaraciones como `\thicklines`.

El formato del comando `\put` es el siguiente:

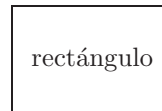
$$\boxed{\code{\put(x,y){objeto}}}$$

Este comando coloca el objeto especificado de modo tal que su punto de referencia tenga coordenadas  $(x,y)$ . El *objeto* puede ser simple texto o alguno de los siguientes:

## Cajas

Se generan con los comandos `\makebox`, `\framebox` y `\dashbox`, que producen cajas sin marco o con marco sólido o a rayas, respectivamente. Los dos primeros comandos requieren como primer argumento un par ordenado que especifica el ancho y la altura de la caja. El punto de referencia es el vértice inferior izquierdo. El segundo argumento (que va entre llaves) es el texto que va a contener la caja. Por defecto este texto va centrado, pero puede indicarse otra posición con los argumentos opcionales `t` (*top*, arriba), `b` (*bottom*, abajo), `l` (*left*, izquierda) y `r` (*right*, derecha). El comando `\dashbox` tiene un primer argumento adicional que especifica la longitud de cada rayita.

```
\begin{picture}(100,60)
\put(20,10){\framebox(60,40)%
{rect'angulo}}
\end{picture}
```



## Líneas rectas

Se generan con el comando

```
\line(a,b){longitud}
```

donde  $(a, b)$  se interpreta como un vector que especifica la dirección de la recta y *longitud* no es realmente la longitud de la recta sino la de su proyección horizontal (si  $a \neq 0$ ) o vertical (si  $a = 0$ ). Por ejemplo si el punto de referencia es  $(x, y)$  entonces el comando `\line{2,1}{10}` producirá una línea que va desde  $(x, y)$  hasta  $(x + 10, y + 5)$ . Lamentablemente las pendientes admitidas son muy pocas:  $a$  y  $b$  deben ser enteros entre  $-6$  y  $+6$  inclusive, sin factores comunes.

```
\begin{picture}(100,60)
\put(10,10){\line(3,4){27}}
\put(37,46){\line(4,-3){48}}
\put(10,10){\line(1,0){75}}
\put(10,10){%
\makebox(75,20){tri'angulo}}
\end{picture}
```



## Flechas

Se generan con el comando `\vector`, el cual trabaja igual que `\line` pero termina en una punta de flecha. Los valores de  $a$  y  $b$  están limitados al rango  $-4 \dots +4$ .

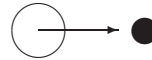
## Círculos

El comando

$$\boxed{\backslash\text{circle}\{radio\}}$$

dibuja una circunferencia con centro en el punto de referencia y *radio* como radio. La variante `\circle*` produce un disco. L<sup>A</sup>T<sub>E</sub>X dispone solamente de una colección fija de círculos y discos, por lo tanto dibujará el que más se aproxime al radio especificado. Las dimensiones máximas dependen de la implementación.

```
\begin{picture}(100,60)
\put(30,30){\circle{20}}
\put(30,30){\vector(1,0){30}}
\put(70,30){\circle*{10}}
\end{picture}
```



## Óvalos

Un óvalo es como un rectángulo con bordes redondeados. Se genera con el comando `\oval`, cuyo primer argumento es un par ordenado que determina las dimensiones (ancho y altura) del óvalo. También pueden dibujarse mitades y cuartos de óvalo usando una o dos de las letras `t`, `b`, `r` y `l` como argumentos opcionales.

## Curvas

Pueden dibujarse curvas de Bezier determinadas por tres puntos de coordenadas  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  con el comando:

$$\boxed{\backslash\text{qbezier}(x_1, y_1)(x_2, y_2)(x_3, y_3)}$$

```
\begin{picture}(100,60)
\qbezier(10,10)(70,60)(80,20)
\qbezier[30](10,10)(90,90)(80,20)
\end{picture}
```



El número de puntos a graficar se puede especificar como un argumento opcional.

Quien desee conocer los restantes comandos disponibles con el entorno `picture` puede consultar [9] y [5]. Sin embargo para realizar diagramas

de cierta complejidad es necesario utilizar paquetes gráficos más poderosos, como por ejemplo `pictex` o `xypic`, el cual permite generar complejos diagramas conmutativos y muchos tipos de gráficas.

### 4.3 Imágenes y colores

Las imágenes generadas con software gráfico (como por ejemplo `xfig` o `gnuplot` bajo linux, `CorelDraw` o `PhotoShop` para Windows) pueden incluirse en un documento  $\LaTeX$  de varias maneras. La más universal y generalizada consiste en el uso del formato `POSTSCRIPT` encapsulado (`eps`). Para imprimir documentos que contengan este tipo de gráficas no es imprescindible poseer una impresora láser, ya que con el programa `GHOSTSCRIPT` [4] puede hacerlo en una gran variedad de impresoras.

Uno de los paquetes que facilita la inclusión de gráficos `eps` en  $\LaTeX$  es `graphics` [1]. Este paquete debe cargarse en el preámbulo del documento con

```
\usepackage[driver]{graphics}
```

donde el argumento opcional *driver* es el nombre del visualizador de archivos `dvi` (o del conversor de `dvi` a `POSTSCRIPT`) que se esté utilizando (por ejemplo podría ser `dvips`, `xdvi`, `dviwin`, `entex`, `pctexwin`, `tcidvi`, `oztex`, `textures`, entre otros).

Ahora para incluir por ejemplo una imagen llamada `zebra.eps` usamos la instrucción `\includegraphics`, escribiendo simplemente:

```
\includegraphics{zebra.eps}
```

El comando `\includegraphics` admite dos argumentos opcionales, cada uno de ellos formado por un par de longitudes separadas por una coma. El primer par especifica las coordenadas del punto de la imagen que se tomará como punto de referencia para imprimirla (el origen de coordenadas

en la imagen es el vértice inferior izquierdo). El segundo par da las coordenadas de otro punto que junto con el primero determina el rectángulo que se desea imprimir. Por ejemplo:

```
\includegraphics*[20mm,26mm][40mm,37mm]{zebra.eps}
```

El `*` se usa para que solamente se vea el rectángulo especificado. Si no se emplea se imprimirá la figura completa, pero sin reservar espacio para ello (es decir sobreimprimiendo cualquier cosa que esté alrededor).

El paquete `\graphics` permite también escalar y rotar imágenes o texto. El comando

```
\scalebox{factor}[factor-v]{objeto}
```

permite especificar el *factor* de escala que se aplicará al *objeto*. El segundo argumento (opcional) *factor-v* se emplea si se desea especificar un factor diferente en dirección vertical.

```
Hola \scalebox{2}{Hola}
\scalebox{1}[2.5]{Hola}
```

Hola **Hola** Hola

El comando

```
\resizebox{ancho}{altura}{objeto}
```

es similar, pero permite especificar las dimensiones absolutas que se desean. Un signo `!` como uno de los argumentos se usa para preservar la razón de aspecto del *objeto*.

Finalmente el comando

```
\rotatebox{ángulo}{objeto}
```

permite rotar un objeto el número de grados indicado por el argumento *ángulo*. Por ejemplo:

```
Hola \rotatebox{45}{Subiendo}
\rotatebox{-30}{Bajando}
```

Hola Subiendo  
Bajando

Para generar texto coloreado se puede utilizar el paquete `color`. Al menos los siguientes colores están predefinidos: `red` (rojo), `green` (verde), `blue`



(azul), `cyan` (turquesa), `magenta` (morado), `black` (negro) y `white` (blanco). Otros colores pueden definirse con la instrucción

```
\definecolor{nombre}{modelo}{especificación}
```

El primer argumento es el *nombre* que vamos a dar al color definido, el *modelo* puede ser `rgb`, `cmymk`, `gray` o `named`. La *especificación* depende del modelo. En el modelo `rgb` consiste en tres números entre 0 y 1, separados por comas, que determinan la proporción de los componentes rojo (*red*), verde (*green*) y azul (*blue*), respectivamente, en el color definido. En el modelo `cmymk` se requieren cuatro números entre 0 y 1, separados por comas, que corresponden a las proporciones de los componentes turquesa (*cyan*), morado (*magenta*), amarillo (*yellow*) y negro (*black*). En el modelo `gray` se requiere solamente un número que representa el nivel de gris, entre 0 (negro) y 1 (blanco). En el modelo `named` cada color se identifica mediante un nombre. Por ejemplo:

```
\definecolor{verdemar}{rgb}{0.3,0.93,0.19}
```

La declaración

```
\color{color}
```

permite generar texto coloreado dentro de su ámbito. Alternativamente se puede usar el comando

```
\textcolor{color}{texto}
```

estos comandos también admiten la especificación directa del color indicando el modelo como un primer argumento opcional seguido de la especificación como argumento obligatorio.

El comando

```
\colorbox{color}{texto}
```

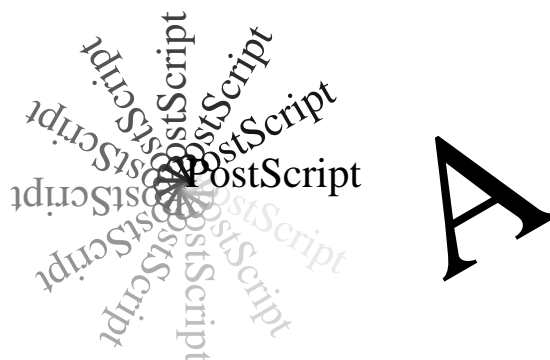
generará una caja del *color* que se le indique con el *texto* en su interior.

```
\textcolor{green}{Verde},
{\color{blue} azul} y\
\colorbox{yellow}
{\color{red}rojo en
una caja amarilla}
```

Verde, azul y  
rojo en una caja amarilla

Todos los comandos vistos en esta sección requieren apoyo de parte de su visualizador de archivos `dvi`, así que es posible que algunos de ellos no funcionen en su sistema. Es común por ejemplo que las imágenes incluidas se vean en pantalla como simples rectángulos vacíos. Sin embargo es probable que aun así se impriman correctamente. Pruebe a convertir el archivo `dvi` a `POSTSCRIPT` (con `dvips`) y examine el resultado con `GHOSTVIEW` [4].

En realidad L<sup>A</sup>T<sub>E</sub>X y `POSTSCRIPT` pueden trabajar juntos muy bien, al punto de que es posible incluir código nativo `POSTSCRIPT` en un documento fuente L<sup>A</sup>T<sub>E</sub>X, como muestran los ejemplos siguientes:



Estas imágenes fueron generadas con el siguiente código:

```
\special{" newpath /Times-Roman findfont 16 scalefont
setfont /texto (PostScript) def 144 72 translate
360 -30 0 { 0 0 moveto 360 div setgray
texto show -30 rotate } for}
```

```
\special{" /Times-Roman findfont 72 scalefont setfont
240 36 translate 30 rotate 0 0 moveto (A) show}
```

Si desea explorar las posibilidades de esta interacción le sugerimos examinar el paquete `pstricks`. Para una amplia y completa información sobre gráficas, inclusión de imágenes en L<sup>A</sup>T<sub>E</sub>X consulte [6].

## 4.4 Documentos grandes

Cuando se trabaja con documentos grandes muchas veces resulta conveniente dividir el archivo de entrada en varias partes. Por ejemplo al escribir un libro es una buena idea tener archivos separados para cada capítulo. La instrucción

```
\include{archivo}
```

utilizada en el cuerpo del documento permite insertar el contenido del archivo especificado en nuestro documento. Así por ejemplo podríamos tener un archivo raíz cuyo cuerpo consista solamente en instrucciones `\includeonly` para incluir el contenido real del documento, dividido en varios archivos.

La instrucción

```
\includeonly{archivo1,archivo2,... }
```

se puede incluir en el preámbulo para indicar que sólo queremos que se procesen los archivos especificados en ella. Sin embargo  $\text{\LaTeX}$  procederá como si los demás archivos estuvieran presentes, numerando las páginas consecutivamente y resolviendo las referencias cruzadas (suponiendo claro está que los archivos no incluidos no han sido modificados desde la última vez que fueron procesados).

Una limitación del comando `\includeonly` es que comienza una nueva página antes de procesar el texto insertado, y lo mismo hace con lo que siga a la instrucción (lo cual no es ningún problema si lo usamos para insertar cosas como capítulos enteros, o tablas que ocupen páginas completas).

La instrucción

```
\input{archivo}
```

hace que  $\text{\LaTeX}$  lea el archivo especificado e inserte su contenido en el punto donde se encuentra la instrucción, sin comenzar una nueva página. Además puede colocarse en cualquier parte del archivo raíz, incluso puede ser la primera instrucción del mismo y cargar otro archivo que contenga la instrucción `\documentclass`. Esto hace posible tener un archivo con las definiciones de comandos y ambientes que solemos usar y cargarlo en el preámbulo de cada documento que escribamos.

## 4.5 Diseño de la página

En base al tamaño del papel indicado en `\documentclass`,  $\text{\LaTeX}$  determina los valores apropiados para los márgenes y otros parámetros de la página. Pero estos valores pueden modificarse con las instrucciones `\setlength` y `\addtolength`. En la página siguiente (que se ha generado con el paquete `layout`) se muestran todos los parámetros que se pueden cambiar. Por ejemplo para añadir un centímetro al ancho del texto, basta incluir en el preámbulo del documento las instrucciones siguientes:

```
\addtolength{\hoffset}{-5mm}
\addtolength{\textwidth}{1cm}
```

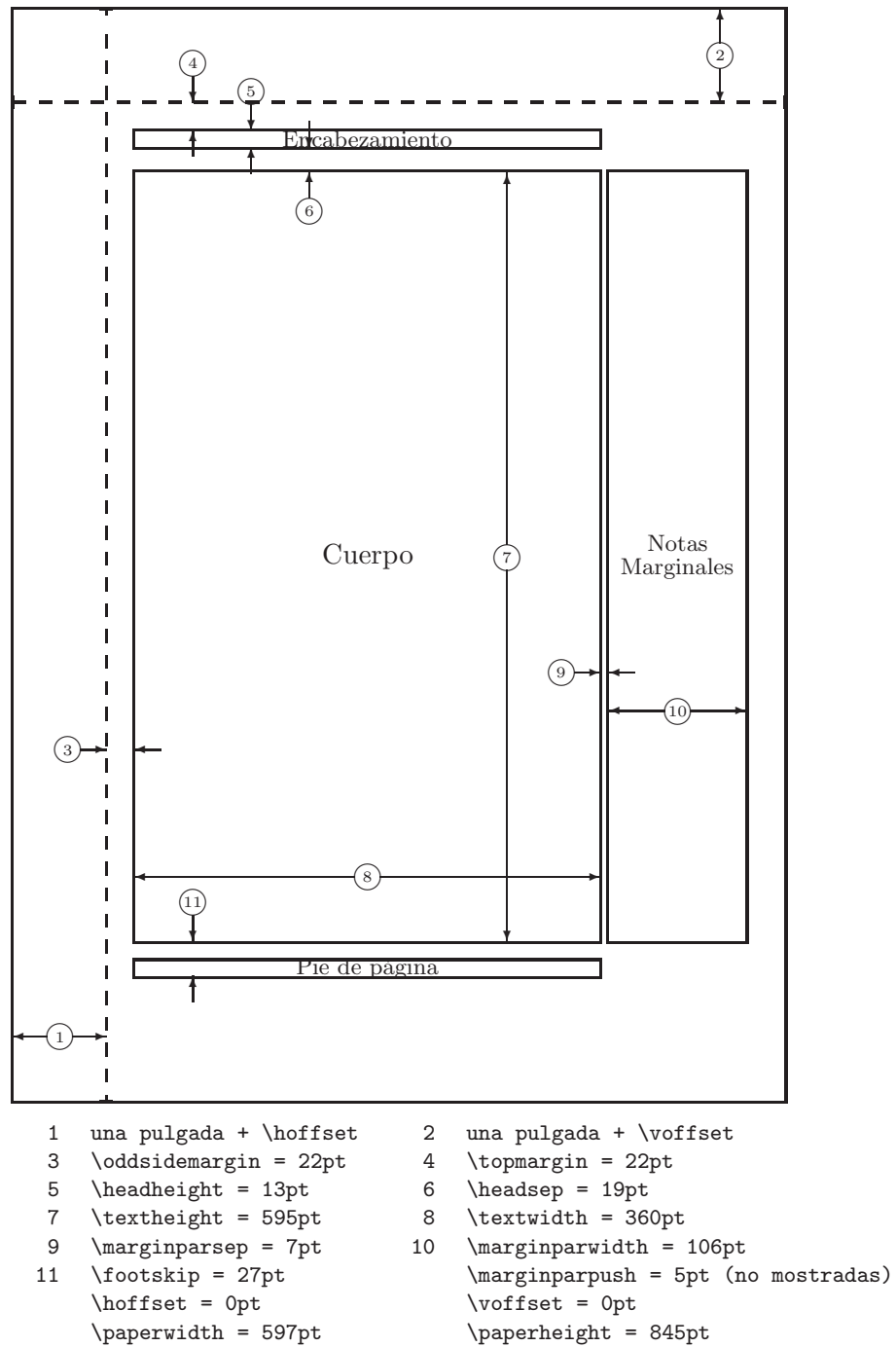


Figura 4.1: Parámetros del diseño de la página

## 4.6 Índices alfabéticos

El programa *MakeIndex* permite elaborar fácilmente un índice alfabético, lo cual es muy útil si se escriben libros. Para ello incluya las siguientes instrucciones en el preámbulo de su documento:

```
\usepackage{makeidx}
\makeindex
```

Para incluir algo en el índice se usa la instrucción

```
\index{entrada}
```

donde *entrada* especifica la forma que tendrá la entrada en el índice. El número de página se generará automáticamente. La sintaxis de este comando se explicará a través de ejemplos. En el caso más sencillo (entradas simples) escribiremos algo como

```
\index{espacio}
```

lo cual generará una entrada para la palabra ‘espacio’. Cuando el texto trate el tema ‘espacio vertical’, en vez de generar una entrada con ambas palabras es preferible generar una subentrada de la entrada principal ‘espacio’. Esto se hace así:

```
\index{espacio!vertical}
```

Es posible también generar sub-subentradas, por ejemplo

```
\index{modo!matem'atico!desplegado}
```

A menudo se desea incluir entradas en algún formato particular (por ejemplo en itálicas o negritas) o que contengan acentos o símbolos especiales. En estos casos se separa la clave que se usará para ordenar las entradas de su apariencia, como se indica a continuación:

```
\index{LaTeX@LaTeX}
```

Las entradas verdaderas generadas por estos ejemplos las puede ver en el índice de este mismo manual.

Cuando se procese el archivo de entrada (supongamos que se llame *milibro.tex*) cada instrucción `\index` escribirá en un archivo llamado *milibro.idx* la entrada en el índice y el número de la página actual. Este archivo debe procesarse con el programa *makeindex* (en algunos sistemas que no permiten nombres de archivo de más de 8 caracteres es posible que este programa se llame *makeidx* o algo semejante)

```
$ makeindex milibro.idx
```

con lo cual se generará un índice ordenado con el nombre `milibro.ind`. También se crea un archivo `milibro.ilg` el cual podemos examinar para ver si se produjeron errores y algunos otros datos, como el número de entradas generadas. Si se texea ahora nuevamente el archivo de entrada, el índice alfabético aparecerá donde se encuentre la instrucción

```
\printindex
```

Hay muchas otras posibilidades, por ejemplo se pueden indizar rangos de páginas, imprimir los números en negritas o bastardilla, incluir referencias cruzadas entre las entradas, etc. Para estos y otros detalles vea [9] o la documentación que acompaña al paquete `makeidx`.

Otro paquete, `showidx`, hace que se impriman todas las entradas del índice en el margen del texto. Esto es útil para las revisiones y para verificar el índice.

## 4.7 Palabras finales

Es posible aprender mucho más... Además de la documentación disponible en CTAN y la bibliografía de referencia ([9], [5]), otra fuente de información sobre T<sub>E</sub>X y L<sup>A</sup>T<sub>E</sub>X son los grupos de discusión de USENET, principalmente `comp.text.tex`. También existen grupos nacionales, como por ejemplo `es.comp.lenguajes.tex` en español, pero su tráfico es naturalmente menor. Si no dispone de un servidor de *news* puede acceder a estos grupos a través de DejaNews ([www.dejanews.com](http://www.dejanews.com)). También existen varias listas de correo a las cuales se puede suscribir, en particular `spanish-tex` en español.

# Apéndice A

## Soluciones a los ejercicios

- 1.1 En cada instalaci' on de `\LaTeX` deber' ia haber una `\emph{Gu'ia Local}` para especificar las caracter' isticas particulares del sistema.
- 1.2 Las acciones de Wilson & Sons llegaron a 20\$, lo cual representa un aumento del 10% respecto al cierre de la v' ispera.
- 1.3 `ch\^ateau, \el\`eve, na\{"i}f.`
- 1.4 `$ax+by \leq \sqrt{a^2+b^2}\sqrt{x^2+y^2}$`
- 1.5 `$F_{n+1} = F_n + F_{n-1}$`
- 1.6 `\[ \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \]`
- 2.1 `\textbf{Horri\textsl{ble}} mez\textit{cla} \textsc{de}`  
`\textsf{est}\texttt{ilos}.`
- 2.2 `{\Huge Di\huge m\LARGE i\Large n\large u\normalsize e`  
`\small n\footnotesize d\scriptsize o \tiny y \scriptsize c%`  
`\footnotesize r\small e\normalsize s\large c\Large e\LARGE n%`  
`\huge d\Huge o}.`
- 2.3 `\renewcommand{\baselinestretch}{1.25}`
- 2.4 `\begin{tabular}{|c||c|c|}`  
`\hline`  
 `$\oplus$ & 0 & 1 \\\`  
`\hline\hline`  
 `0 & 0 & 1 \\\ \hline`  
 `1 & 1 & 0 \\\ \hline`  
`\end{tabular}`
- 2.5 Existen muchas soluciones posibles, una de ellas es:

```

\newcommand{\uno}{Para aprender debo hacer todos los%
ejercicios.\\}
\newcommand{\cinco}{\uno\uno\uno\uno\uno}
\newcommand{\veinte}{\cinco\cinco\cinco\cinco}
\newcommand{\cien}{\veinte\veinte\veinte\veinte\veinte}

```

Pero usando el paquete ifthen se puede hacer de manera más elegante:

```

\newcounter{lin}
\whiledo{\value{lin}<100}{\stepcounter{lin}%
Para aprender debo hacer todos los ejercicios.\\}

```

- 3.1 \[  

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x^2} = \frac{1}{2}$$
\]
- 3.2 \[  

$$\sqrt[n]{\frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot \dots \cdot 2n}}$$
\]
- 3.3 \[  

$$\Gamma(z) = \frac{e^{-\gamma z}}{z} \prod_{n=1}^{\infty} \left(1 + \frac{z}{n}\right)^{-1} e^{z/n}$$
\]
- 3.4 \[  

$$\int_S (\nabla \times \mathbf{F}) \cdot \mathbf{N} \, dS = \oint_{\partial S} \mathbf{F} \cdot d\mathbf{P}$$
\]
- 3.5 \[  

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} + V(x) \Psi(x, t) = \tilde{H} \Psi(x, t)$$
\]
- 3.6 \[  

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{\substack{J \subseteq \{1, 2, \dots, n\} \\ |J| \geq 1}} (-1)^{|J|-1} \left| \bigcap_{j \in J} A_j \right|$$
\]
- 3.7 \[  

$$\left| \begin{array}{ccc} 1 & 1 & 1 \\ x & y & z \end{array} \right|$$
\]



---

```

x^2 & y^2 & z^2
\end{array} \right| = (x-y)(y-z)(z-x)
\]
```

```

3.8 \begin{eqnarray*}
  \nabla\cdot \{\mathbf{E}\} &=& 4\pi\rho\\
  \nabla\times \{\mathbf{E}\} &=& \{\mathbf{0}\}\\
  \nabla\cdot \{\mathbf{B}\} &=& 0\\
  \nabla\times \{\mathbf{B}\} &=& \frac{4\pi}{c} \{\mathbf{J}\}
\end{eqnarray*}
```

```

3.9 \[
\begin{array}{ccc}
A & \stackrel{\{\phi\}}{\longrightarrow} & B \\
\scriptstyle\{\psi\} \downarrow & & \\
\uparrow \scriptstyle\{\xi\} & & \\
C & \stackrel{\{\theta\}}{\longrightarrow} & D
\end{array}
\]
```

```

3.10 \[
  2\uparrow\uparrow k\stackrel{\text{def}}{=}
  2^{2^{2^{\cdots^{\cdots^{\cdots^2}}}}}}
  \raisebox{\depth}{\$\Bigr\} \ , \ \scriptstyle k$.
\]
```



## Apéndice B

# Tablas de símbolos matemáticos

Las Tablas reunidas en este Apéndice muestran la gran variedad de símbolos que se pueden utilizar con L<sup>A</sup>T<sub>E</sub>X en el modo matemático. Algunos símbolos requieren el paquete latexsym (se indica en cada caso). Los de las Tablas B.14–B.12 requieren el paquete amssymb.

Tabla B.1: Relaciones binarias

Puede realizar las negaciones correspondientes a estos símbolos añadiéndoles una orden `\not` como prefijo a las instrucciones siguientes.

$<$	<code>&lt;</code>	$>$	<code>&gt;</code>	$=$	<code>=</code>
$\leq$	<code>\leq</code> o <code>\le</code>	$\geq$	<code>\geq</code> o <code>\ge</code>	$\equiv$	<code>\equiv</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\doteq$	<code>\doteq</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubset$	<code>\sqsubset</code> <sup>a</sup>	$\sqsupset$	<code>\sqsupset</code> <sup>a</sup>	$\bowtie$	<code>\Join</code> <sup>a</sup>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code> , <code>\owns</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>
$\mid$	<code>\mid</code>	$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>
$\smile$	<code>\smile</code>	$\frown$	<code>\frown</code>	$\asymp$	<code>\asymp</code>
$:$	<code>:</code>	$\notin$	<code>\notin</code>	$\neq$	<code>\neq</code> o <code>\ne</code>

<sup>a</sup>Requiere latexsym o amssymb

Tabla B.2: Operadores binarios

$+$	<code>+</code>	$-$	<code>-</code>	$\triangleleft$	<code>\triangleleft</code>
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\triangleright$	<code>\triangleright</code>
$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>	$\star$	<code>\star</code>
$\times$	<code>\times</code>	$\setminus$	<code>\setminus</code>	$\ast$	<code>\ast</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$\circ$	<code>\circ</code>
$\sqcup$	<code>\sqcup</code>	$\sqcap$	<code>\sqcap</code>	$\bullet$	<code>\bullet</code>
$\vee$	<code>\vee</code> , <code>\lor</code>	$\wedge$	<code>\wedge</code> , <code>\land</code>	$\diamond$	<code>\diamond</code>
$\oplus$	<code>\oplus</code>	$\ominus$	<code>\ominus</code>	$\uplus$	<code>\uplus</code>
$\odot$	<code>\odot</code>	$\oslash$	<code>\oslash</code>	$\amalg$	<code>\amalg</code>
$\otimes$	<code>\otimes</code>	$\bigcirc$	<code>\bigcirc</code>	$\dagger$	<code>\dagger</code>
$\triangle$	<code>\bigtriangleup</code>	$\nabla$	<code>\bigtriangledown</code>	$\ddagger$	<code>\ddagger</code>
$\triangleleft$	<code>\lhd</code> <sup>a</sup>	$\triangleright$	<code>\rhd</code> <sup>a</sup>	$\wr$	<code>\wr</code>
$\trianglelefteq$	<code>\unlhd</code> <sup>a</sup>	$\trianglerighteq$	<code>\unrhd</code> <sup>a</sup>		

<sup>a</sup>Requiere latexsym o amssymb

Tabla B.3: Símbolos de tamaño variable

$\sum$	<code>\sum</code>	$\bigcup$	<code>\bigcup</code>	$\bigvee$	<code>\bigvee</code>	$\bigoplus$	<code>\bigoplus</code>
$\prod$	<code>\prod</code>	$\bigcap$	<code>\bigcap</code>	$\bigwedge$	<code>\bigwedge</code>	$\bigotimes$	<code>\bigotimes</code>
$\coprod$	<code>\coprod</code>	$\bigsqcup$	<code>\bigsqcup</code>			$\bigodot$	<code>\bigodot</code>
$\int$	<code>\int</code>	$\oint$	<code>\oint</code>			$\biguplus$	<code>\biguplus</code>

Tabla B.4: Delimitadores

$($	<code>(</code>	$)$	<code>)</code>	$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
$[$	<code>[</code> o <code>\lbrack</code>	$]$	<code>]</code> o <code>\rbrack</code>	$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\{$	<code>\{</code> o <code>\lbrace</code>	$\}$	<code>\}</code> o <code>\rbrace</code>	$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\langle$	<code>\langle</code>	$\rangle$	<code>\rangle</code>	$ $	<code> </code> o <code>\vert</code>	$\ $	<code>\ </code> o <code>\Vert</code>
$\lfloor$	<code>\lfloor</code>	$\rfloor$	<code>\rfloor</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
$/$	<code>/</code>	$\backslash$	<code>\backslash</code>	.	(nada)		

Tabla B.5: Delimitadores grandes

$\left($	<code>\left(</code>	$\right)$	<code>\right)</code>	$\left\{$	<code>\left\{</code>	$\right\}$	<code>\right\}</code>
$\left $	<code>\left </code>	$\right $	<code>\right </code>	$\left\ $	<code>\left\ </code>	$\right\ $	<code>\right\ </code>

Tabla B.6: Flechas

$\leftarrow$	<code>\leftarrow</code> o <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\rightarrow$	<code>\rightarrow</code> o <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Lleftarrow$	<code>\Lleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$	<code>\iff</code> (bigger spaces)	$\leadsto$	<code>\leadsto</code> <sup>a</sup>

<sup>a</sup>Requiere `latexsym` o `amssymb`

Tabla B.7: Letras griegas minúsculas

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$\omicron$	<code>\omicron</code>	$\upsilon$	<code>\upsilon</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\varphi$	<code>\varphi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\chi$	<code>\chi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\psi$	<code>\psi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\omega$	<code>\omega</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>		
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>	$\tau$	<code>\tau</code>		

Tabla B.8: Letras griegas mayúsculas

$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

Tabla B.9: Símbolos diversos

$\dots$	<code>\dots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\hbar$	<code>\hbar</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	$\ell$	<code>\ell</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\mho$	<code>\mho</code> <sup>a</sup>	$\partial$	<code>\partial</code>
$'$	<code>'</code>	$\prime$	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	$\infty$	<code>\infty</code>
$\nabla$	<code>\nabla</code>	$\triangle$	<code>\triangle</code>	$\square$	<code>\Box</code> <sup>a</sup>	$\diamond$	<code>\Diamond</code> <sup>a</sup>
$\perp$	<code>\bot</code>	$\top$	<code>\top</code>	$\sphericalangle$	<code>\angle</code>	$\surd$	<code>\surd</code>
$\diamond$	<code>\diamondsuit</code>	$\heartsuit$	<code>\heartsuit</code>	$\clubsuit$	<code>\clubsuit</code>	$\spadesuit$	<code>\spadesuit</code>
$\neg$	<code>\neg</code> o <code>\lnot</code>	$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>

<sup>a</sup>Requiere latexsym o amssymb

Tabla B.10: Relaciones binarias de la AMS

$\triangleleft$	<code>\lessdot</code>	$\triangleright$	<code>\gtrdot</code>	$\doteq$	<code>\doteqdot</code> o <code>\Doteq</code>
$\leqslant$	<code>\leqslant</code>	$\geqslant$	<code>\geqslant</code>	$\risingdotseq$	<code>\risingdotseq</code>
$\leqslantless$	<code>\leqslantless</code>	$\leqslantgtr$	<code>\leqslantgtr</code>	$\fallingdotseq$	<code>\fallingdotseq</code>
$\leqq$	<code>\leqq</code>	$\geqq$	<code>\geqq</code>	$\eqcirc$	<code>\eqcirc</code>
$\lll$ o $\llless$	<code>\lll</code> o <code>\llless</code>	$\ggg$ o $\gggtr$	<code>\ggg</code> o <code>\gggtr</code>	$\circeq$	<code>\circeq</code>
$\lesssim$	<code>\lesssim</code>	$\gtrsim$	<code>\gtrsim</code>	$\trianglelefteq$	<code>\trianglelefteq</code>
$\lessapprox$	<code>\lessapprox</code>	$\gtrapprox$	<code>\gtrapprox</code>	$\bumpeq$	<code>\bumpeq</code>
$\lessgtr$	<code>\lessgtr</code>	$\gtrless$	<code>\gtrless</code>	$\Bumpeq$	<code>\Bumpeq</code>
$\lesseqgtr$	<code>\lesseqgtr</code>	$\gtreqless$	<code>\gtreqless</code>	$\thicksim$	<code>\thicksim</code>
$\lesseqqgtr$	<code>\lesseqqgtr</code>	$\gtreqqless$	<code>\gtreqqless</code>	$\thickapprox$	<code>\thickapprox</code>
$\preccurlyeq$	<code>\preccurlyeq</code>	$\succcurlyeq$	<code>\succcurlyeq</code>	$\approxeq$	<code>\approxeq</code>
$\curlyeqprec$	<code>\curlyeqprec</code>	$\curlyeqsucc$	<code>\curlyeqsucc</code>	$\backsim$	<code>\backsim</code>
$\precsim$	<code>\precsim</code>	$\succsim$	<code>\succsim</code>	$\backsimeq$	<code>\backsimeq</code>
$\precapprox$	<code>\precapprox</code>	$\succapprox$	<code>\succapprox</code>	$\vDash$	<code>\vDash</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\Vdash$	<code>\Vdash</code>
$\Subset$	<code>\Subset</code>	$\Supset$	<code>\Supset</code>	$\Vvdash$	<code>\Vvdash</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupset$	<code>\sqsupset</code>	$\backepsilon$	<code>\backepsilon</code>
$\therefore$	<code>\therefore</code>	$\because$	<code>\because</code>	$\varpropto$	<code>\varpropto</code>
$\shortmid$	<code>\shortmid</code>	$\shortparallel$	<code>\shortparallel</code>	$\between$	<code>\between</code>
$\smallsmile$	<code>\smallsmile</code>	$\smallfrown$	<code>\smallfrown</code>	$\pitchfork$	<code>\pitchfork</code>
$\vartriangleleft$	<code>\vartriangleleft</code>	$\vartriangleright$	<code>\vartriangleright</code>	$\blacktriangleleft$	<code>\blacktriangleleft</code>
$\trianglelefteq$	<code>\trianglelefteq</code>	$\trianglerighteq$	<code>\trianglerighteq</code>	$\blacktriangleright$	<code>\blacktriangleright</code>

Tabla B.11: Flechas de la AMS

$\dashleftarrow$	<code>\dashleftarrow</code>	$\dashrightarrow$	<code>\dashrightarrow</code>	$\multimap$	<code>\multimap</code>
$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>	$\rightrightarrows$	<code>\rightrightarrows</code>	$\Uparrow$	<code>\upuparrows</code>
$\leftrightarrows$	<code>\leftrightarrows</code>	$\rightleftarrows$	<code>\rightleftarrows</code>	$\Downarrow$	<code>\downdownarrows</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>	$\Uparrow$	<code>\upharpoonleft</code>
$\twoheadleftarrow$	<code>\twoheadleftarrow</code>	$\twoheadrightarrow$	<code>\twoheadrightarrow</code>	$\Uparrow$	<code>\upharpoonright</code>
$\leftarrowtail$	<code>\leftarrowtail</code>	$\rightarrowtail$	<code>\rightarrowtail</code>	$\Downarrow$	<code>\downharpoonleft</code>
$\leftrightharpoons$	<code>\leftrightharpoons</code>	$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\Downarrow$	<code>\downharpoonright</code>
$\Lsh$	<code>\Lsh</code>	$\Rsh$	<code>\Rsh</code>	$\rightsquigarrow$	<code>\rightsquigarrow</code>
$\looparrowleft$	<code>\looparrowleft</code>	$\looparrowright$	<code>\looparrowright</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>
$\curvearrowleft$	<code>\curvearrowleft</code>	$\curvearrowright$	<code>\curvearrowright</code>		
$\circlearrowleft$	<code>\circlearrowleft</code>	$\circlearrowright$	<code>\circlearrowright</code>		

Tabla B.12: Relaciones y flechas negadas de la AMS

$\nless$	<code>\nless</code>	$\ngtr$	<code>\ngtr</code>	$\varsubsetneqq$	<code>\varsubsetneqq</code>
$\lneq$	<code>\lneq</code>	$\gneq$	<code>\gneq</code>	$\varsupsetneqq$	<code>\varsupsetneqq</code>
$\nleq$	<code>\nleq</code>	$\ngeq$	<code>\ngeq</code>	$\nsubseteqeq$	<code>\nsubseteqeq</code>
$\nleqslant$	<code>\nleqslant</code>	$\ngeqslant$	<code>\ngeqslant</code>	$\nsupseteqq$	<code>\nsupseteqq</code>
$\lneqq$	<code>\lneqq</code>	$\gneqq$	<code>\gneqq</code>	$\nmid$	<code>\nmid</code>
$\lvertneqq$	<code>\lvertneqq</code>	$\gvertneqq$	<code>\gvertneqq</code>	$\nparallel$	<code>\nparallel</code>
$\nleqq$	<code>\nleqq</code>	$\ngeqq$	<code>\ngeqq</code>	$\nshortmid$	<code>\nshortmid</code>
$\lnsim$	<code>\lnsim</code>	$\gnsim$	<code>\gnsim</code>	$\nshortparallel$	<code>\nshortparallel</code>
$\lnapprox$	<code>\lnapprox</code>	$\gnapprox$	<code>\gnapprox</code>	$\nsim$	<code>\nsim</code>
$\nprec$	<code>\nprec</code>	$\nsucc$	<code>\nsucc</code>	$\ncong$	<code>\ncong</code>
$\npreceq$	<code>\npreceq</code>	$\nsucceq$	<code>\nsucceq</code>	$\nvdash$	<code>\nvdash</code>
$\nprecneqq$	<code>\nprecneqq</code>	$\nsuccneqq$	<code>\nsuccneqq</code>	$\nvDash$	<code>\nvDash</code>
$\nprecnsim$	<code>\nprecnsim</code>	$\nsuccnsim$	<code>\nsuccnsim</code>	$\nVdash$	<code>\nVdash</code>
$\nprecnapprox$	<code>\nprecnapprox</code>	$\nsuccnapprox$	<code>\nsuccnapprox</code>	$\nVDash$	<code>\nVDash</code>
$\subsetneq$	<code>\subsetneq</code>	$\supsetneq$	<code>\supsetneq</code>	$\ntriangleleft$	<code>\ntriangleleft</code>
$\varsubsetneq$	<code>\varsubsetneq</code>	$\varsupsetneq$	<code>\varsupsetneq</code>	$\ntriangleright$	<code>\ntriangleright</code>
$\nsubseteq$	<code>\nsubseteq</code>	$\nsupseteq$	<code>\nsupseteq</code>	$\ntrianglelefteq$	<code>\ntrianglelefteq</code>
$\subsetneqq$	<code>\subsetneqq</code>	$\supsetneqq$	<code>\supsetneqq</code>	$\ntrianglerighteq$	<code>\ntrianglerighteq</code>
$\nleftarrow$	<code>\nleftarrow</code>	$\nrightarrow$	<code>\nrightarrow</code>	$\nleftrightharrow$	<code>\nleftrightharrow</code>
$\nLeftarrow$	<code>\nLeftarrow</code>	$\nRightarrow$	<code>\nRightarrow</code>	$\nLeftrightarrow$	<code>\nLeftrightarrow</code>

Tabla B.13: Operadores binarios de la AMS

$\dot{+}$	<code>\dotplus</code>	$\cdot$	<code>\centerdot</code>	$\intercal$	<code>\intercal</code>
$\ltimes$	<code>\ltimes</code>	$\rtimes$	<code>\rtimes</code>	$\div$	<code>\divideontimes</code>
$\cup\cup$	<code>\Cup o \doublecup</code>	$\cap\cap$	<code>\Cap o \doublecap</code>	$\smallsetminus$	<code>\smallsetminus</code>
$\veebar$	<code>\veebar</code>	$\bar{\wedge}$	<code>\barwedge</code>	$\bar{\bar{\wedge}}$	<code>\doublebarwedge</code>
$\boxplus$	<code>\boxplus</code>	$\boxminus$	<code>\boxminus</code>	$\ominus$	<code>\circleddash</code>
$\boxtimes$	<code>\boxtimes</code>	$\boxdot$	<code>\boxdot</code>	$\odot$	<code>\circledcirc</code>
$\leftthreetimes$	<code>\leftthreetimes</code>	$\rightthreetimes$	<code>\rightthreetimes</code>	$\otimes$	<code>\circledast</code>
$\curlyvee$	<code>\curlyvee</code>	$\curlywedge$	<code>\curlywedge</code>		

Tabla B.14: Delimitadores de la AMS

$\ulcorner$	<code>\ulcorner</code>	$\urcorner$	<code>\urcorner</code>	$\llcorner$	<code>\llcorner</code>	$\lrcorner$	<code>\lrcorner</code>
-------------	------------------------	-------------	------------------------	-------------	------------------------	-------------	------------------------

Tabla B.15: Símbolos griegos y hebreos de la AMS

$\digamma$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beth$	<code>\beth</code>	$\daleth$	<code>\daleth</code>	$\gimel$	<code>\gimel</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	-----------	----------------------	----------	---------------------

Tabla B.16: Símbolos diversos de la AMS

$\hbar$	<code>\hbar</code>	$\hslash$	<code>\hslash</code>	$\Bbbk$	<code>\Bbbk</code>
$\square$	<code>\square</code>	$\blacksquare$	<code>\blacksquare</code>	$\textcircled{S}$	<code>\circledS</code>
$\vartriangle$	<code>\vartriangle</code>	$\blacktriangle$	<code>\blacktriangle</code>	$\complement$	<code>\complement</code>
$\nabla$	<code>\triangledown</code>	$\blacktriangledown$	<code>\blacktriangledown</code>	$\Game$	<code>\Game</code>
$\lozenge$	<code>\lozenge</code>	$\blacklozenge$	<code>\blacklozenge</code>	$\bigstar$	<code>\bigstar</code>
$\angle$	<code>\angle</code>	$\measuredangle$	<code>\measuredangle</code>	$\sphericalangle$	<code>\sphericalangle</code>
$\diagup$	<code>\diagup</code>	$\diagdown$	<code>\diagdown</code>	$\backprime$	<code>\backprime</code>
$\nexists$	<code>\nexists</code>	$\Finv$	<code>\Finv</code>	$\varnothing$	<code>\varnothing</code>
$\eth$	<code>\eth</code>	$\mho$	<code>\mho</code>		



# Bibliografía

- [1] Carlisle, D. *Packages in the 'graphics' bundle*, disponible como `grfguide.dvi` con la documentación del conjunto `graphics`.
- [2] Costella, J. <http://www.ph.unimelb.edu.au/~jpc/#dviview> (en esta página hay una versión de `dviwin` empaquetada con `METAFONT`).
- [3] CTAN (*Comprehensive T<sub>E</sub>X Archive Network*). Los sitios principales son <ftp.dante.de> en Alemania, <ftp.tex.ac.uk> en el Reino Unido y <ftp.shsu.edu> en los Estados Unidos de América. También se puede acceder a través de las direcciones <http://tug2.cs.umb.edu/ctan/> y <http://www.tug.org/>.
- [4] GhostScript y GhostView, <http://www.cs.wisc.edu/~ghost/>
- [5] Goossens, M., Mittelbach, F., Samarin, A. *The L<sup>A</sup>T<sub>E</sub>X Companion*, Addison-Wesley, Reading, 1994.
- [6] Goossens, M., Rahtz, S., Mittelbach, F. *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion*, Addison-Wesley, Reading, 1997.
- [7] Graham, R. L. Knuth, D. E., Patashnick, O. *Concrete Mathematics*, 2nd ed., Addison Wesley, 1995.
- [8] Hoening, A., Hoenig, A. *T<sub>E</sub>XUnbound : L<sup>A</sup>T<sub>E</sub>X and T<sub>E</sub>X Strategies for Fonts, Graphics, & More*, Oxford University Press, 1998.
- [9] Lamport, L. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, 2nd ed., Addison-Wesley, Reading, 1994.
- [10] Knuth, D.E. *The T<sub>E</sub>Xbook*, Addison-Wesley, Reading, 1984.
- [11] Knuth, D.E. *The METAFONT book*, Addison-Wesley, Reading, 1986.
- [12] Knuth, D.E. *Literate Programming*, CSLI Lecture Notes No. 27, Stanford University, 1992.
- [13] Krantz, S.G. *A Primer of Mathematical Writing*, American Mathematical Society, Providence, 1996.

- 
- [14] L<sup>A</sup>T<sub>E</sub>X3 Project Team, *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for authors*, viene con la distribución de L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> como `usrguide.dvi`.
  - [15] L<sup>A</sup>T<sub>E</sub>X3 Project Team, *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for Class and Package writers*, viene con la distribución de L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> como `clsguide.dvi`.
  - [16] L<sup>A</sup>T<sub>E</sub>X3 Project Team, *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Font selection*, viene en la distribución de L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> como `fntguide.dvi`.
  - [17] MiKTeX, <http://www.miktex.de/>
  - [18] Personal T<sub>E</sub>X Inc., *DVIscope*, <http://www.pctex.com/dviscope.htm>
  - [19] Spivak, M. *The Joy of T<sub>E</sub>X*, American Mathematical Society, 1990.
  - [20] Steenrod, N.E., Halmos, P.R., Schiffer, M.M., Dieudonné, J.A. *How to Write Mathematics*, American Mathematical Society, 1973.
  - [21] *TextPad*, Hellos Software, <http://www.textpad.com>
  - [22] *TUG (T<sub>E</sub>X Users Group)*, <http://www.tug.org>
  - [23] *WinEdit*, Wilson WindowWare, <http://www.winware.com>

# Índice de Materias

`\!`, 53  
`\"`, 13  
`\#` (`#`), 13  
`\$` (`$`), 13  
`$$`, 51  
`\%` (`%`), 13  
`&`, 36, 60  
`\&` (`&`), 13  
`\'`, 13, 57  
`\(`, 51  
`\)`, 51  
`\,`, 53  
– (signo negativo), 22  
- (guión), 22  
`\-`, 32  
– (guión largo), 22  
— (raya), 22  
`/`, 59  
`\:`, 53  
`\;`, 53  
`\@`, 29  
`@`, 37  
`\[`, 17, 51  
`\]`, 13, 24, 30, 36  
`\]`, 17, 51  
`ˆ`, 16, 57  
`˘`, 16, 57  
`\_`, 13  
`\‘`, 13  
`|`, 36  
`\{`, 13  
`\}`, 13  
`\~` (tilde), 13  
`˜`, 29  
`abstract`, 35, 44  
acento  
    agudo, 13  
    grave, 13  
    matemático, 57  
    ortográfico, 21  
`activeacute`, 34  
`\address`, 44  
`\addtocounter`, 46  
`\addtolength`, 23, 75  
`align`, 52  
`alignat`, 52  
`\Alph`, 46  
`\alph`, 46  
ambientes, 15  
*AMS-LATEX*, 3  
`amsmath`, 52, 53, 55, 58, 59  
`amssymb`, 83  
*AMS-TEX*, 3  
`amsthm`, 61  
`\and`, 42  
`\appendix`, 42, 43  
`\arabic`, 46  
archivo  
    log, 7  
    auxiliar (`aux`), 7  
    de salida (`dvi`), 7  
    fuente (`tex`), 6  
    idx, 77  
    ilg, 78  
    ind, 78  
argumentos, 12  
    obligatorios, 12  
    opcionales, 12

- array, 60
- article, 42
- artículos, 42
- \author, 42
  
- babel, 13, 33
- \backmatter, 43
- \baselineskip, 27
- \baselinestretch, 27
- bastardilla, 15
- \begin, 15
- \bf, 26
- \bfseries, 15, 25
- \bibitem, 39
- BIBTEX, 40
- \Big, 54
- \big, 54
- \Bigg, 54
- \bigg, 54
- \bigskip, 29
- \binom, 60
- bitácora, 7
- \bmod, 55
- book, 43
  
- caja, 65
- \caption, 39
- caracteres, 11
  - especiales, 11
  - invisibles, 12
- cartas, 44
- cases, 60
- \cc, 44
- \cdots, 54
- center, 30
- \chapter, 43
- \circle, 70
- \circle\*, 70
- \cite, 40
- \clearpage, 39
- \cline, 37
- \closing, 44
- coeficiente binomial, 60
- color, 72
- \color, 73
- \colorbox, 73
- comandos, 12
  - definición de, 47
- comentarios, 14
- Computer Modern, 2
- Concrete, 2
- contador, 45
- CTAN, 6
- cuerpo del documento, 11
  
- \dashbox, 69
- \date, 42
- \ddots, 55
- declaración, 15
  - ámbito de una, 15
- \DeclareMathOperator, 55
- \definecolor, 73
- \depth, 68
- description, 35
- diéresis, 13
- diseño de la página, 75
- displaymath, 51
- \displaystyle, 57
- \documentclass, 11
  - opciones de, 44
- dos columnas, 45
- \dots, 54
- double space, 27
- driver*, 9
- dvi, 7
- dvips, 10
  
- ec fonts*, 34
- ecuaciones largas, 52
- elipsis, 22
- \em, 15
- em, 22
- \emph, 15
- \end, 15
- enumerate, 36, 48
- enumerate, 35, 45, 48
- enumi, 48

- eqnarray, 51
- equation, 51
- \equiv, 55
- espacio, 12
  - doble, 27
  - en modo matemático, 53
  - entre líneas, 27
  - horizontal, 28
  - tras un comando, 14
  - tras un punto, 29
  - vertical, 29
- español, 33
- etiqueta, 51
- \evensidemargin, 75
- ex, 22
  
- familia, 24
- fancyhdr, 41
- \fbox, 65
- \fboxrule, 66
- \fboxsep, 66
- figuras, 37
- figure, 38
- flalign, 52
- fleqn, 44
- flushleft, 30
- flushright, 30
- \fnsymbol, 46
- font, 2
- fontenc, 34
- \footnote, 26
- \footnotesep, 26
- \footnotesize, 25
- \footskip, 75
- forma, 24
- formato, 34
- \frac, 17, 59
- fracción, 59
- \framebox, 65, 68, 69
- \frenchspacing, 30
- \frontmatter, 43
- fuelle, 2
- funciones, 55
- fundición, 2
  
- gather, 52
- \genfrac, 59, 60
- \geq, 17
- GHOSTSCRIPT, 10, 71
- GHOSTVIEW, 10, 74
- glifo, 2
- goma, 65
- graphics, 71
- \graphpaper, 68
- guiones, 22
  
- \headheight, 75
- \headsep, 75
- \height, 68
- \hline, 36
- \hoffset, 75
- \hspace, 28
- \hspace\*, 28
- \Huge, 25
- \huge, 25
- \hyphenation, 33
  
- \i, 13
- idiomas, 33
- ifthen, 80
- \iiiint, 53
- \iiint, 53
- \iint, 53
- inclinada, 25
- \include, 74
- \includegraphics, 71
- \includeonly, 75
- \indent, 28
- indentfirst, 28
- \index, 77
- índice
  - alfabético, 77
  - general, 42
- \infty, 17
- \input, 75
- inputenc, 13
- \int, 17
- \it, 26
- itálica, 15, 24, 25

- \item, 35
- itemize, 35
- \itshape, 25
- Knuth, Donald E., 1
- \label, 31, 51
- \labelenumi, 48
- Lamport, Leslie, 3
- \langle, 54
- \LARGE, 25
- \Large, 25
- \large, 25
- L<sup>A</sup>T<sub>E</sub>X, 3
- L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, 3
- \LaTeXe, 9
- latexsym, 83
- layout, 75
- \ldots, 22, 54
- \left, 17, 54
- \leq, 17
- leqno, 44
- letras, 11
  - escandinavas, 21
  - griegas, 16
- letter, 44
- letter, 44
- libros, 43
- \lim, 17
- \limits, 58
- \line, 69
- \linebreak, 24
- list, 35
- \listoffigures, 39
- \listoftables, 39
- \ln, 17
- lrbox, 68
- máquina, 24
- márgenes, 75
- módulo, 55
- \mainmatter, 43
- \makebox, 65, 68, 69
- makeidx, 77
- \makeindex, 77
- \maketitle, 42, 44
- \marginpar, 27
- \marginparpush, 75
- \marginparsep, 75
- \marginparwidth, 75
- \markboth, 41
- \markright, 41
- math, 51
- \mathbb, 56
- \mathbf, 56
- \mathcal, 56
- \mathfrak, 56
- \mathit, 56
- \mathnormal, 56
- \mathrm, 56
- \mathscr, 56
- \mathsf, 56
- \mathtt, 56
- \mbox, 53, 65
- \mdseries, 25
- \medskip, 29
- METAFONT, 2
- minipage, 66
- Mittelbach, Frank, 3
- modo
  - de texto, 16
  - matemático, 16
  - desplegado, 17
- \multicolumn, 37
- \multirow, 68
- multline, 52
- negrita, 15, 25
- \newcommand, 47
- \newcounter, 46
- \newenvironment, 48, 49
- \newlength, 23
- \newpage, 24
- \newsavebox, 67
- \newtheorem, 45, 61
- \newtheorem\*, 61
- NFSS, 24
- \noindent, 28
- \nolimits, 58

- `\nolinebreak`, 24
- `\nonumber`, 51
- `\nopagebreak`, 24
- `\normalsize`, 25
- `\notag`, 52
- `notitlepage`, 45
- número de página, 40
  
- objetos flotantes, 37
  - permisos para, 38
- oblicua, 24
- `oneside`, 45
- opciones, 11
- `openany`, 45
- `\opening`, 44
- `openright`, 45
- OT1, 34
- `\oval`, 70
- `\overbrace`, 58
- overflow box*, 32
- `\overleftarrow`, 58
- `\overline`, 58
- `\overrightarrow`, 58
- `\overset`, 59
  
- `\page`, 46
- `\pagebreak`, 24
- `\pagenumbering`, 40
- `\pageref`, 31
- `\pagestyle`, 41
- página
  - cabecera de, 40
  - cuerpo de, 40
  - estilo de, 40
  - número de, 40
  - pie de, 40
- papel (tamaño), 44
- `\paperheight`, 75
- `\paperwidth`, 75
- paquetes, 18
- parámetro
  - de estilo, 23
  - de un comando, 47
- `\parbox`, 66
  
- `\parindent`, 27
- `\parskip`, 27
- `\part`, 43
- pica, 22
- `pictex`, 71
- `picture`, 68
- `\pmod`, 55
- POSTSCRIPT, 2, 10, 74
  - encapsulado, 71
- preámbulo, 11
- prima, 57
- `\printindex`, 78
- `proof`, 62
- `pstricks`, 74
- punto, 22
- punto de referencia, 65, 68
- puntos suspensivos, 22, 54
  - en diagonal, 55
  - horizontales, 55
  - verticales, 55
- `\put`, 68
  
- `\qBezier`, 68, 70
- `\qedsymbol`, 62
- `\qqquad`, 53
- `\quad`, 53
- quotation, 30
- quote, 30
  
- raíz cuadrada, 55
- `\raisebox`, 63, 67, 68
- `\rangle`, 54
- redonda, 25
- `\ref`, 31, 46, 51
- `\refstepcounter`, 46, 47
- regla, 67
- `\renewcommand`, 48
- `\renewenvironment`, 49
- report, 43
- `\resizebox`, 72
- `\right`, 17, 54
- `\rm`, 26
- `\rmfamily`, 25
- `\Roman`, 46

- `\roman`, 46, 48
- romana, 24
- `\rotatebox`, 72
- `\rule`, 67
- sans serif, 24, 25
- `\savebox`, 67, 68
- `\sbox`, 68
- `\sc`, 26
- `\scalebox`, 72
- Schöpf, Rainer, 3
- `\scriptscriptstyle`, 57
- `\scriptsize`, 25
- `\scriptstyle`, 57
- `\scshape`, 25
- `\selectlanguage`, 33
- serie, 24
- serifs, 24
- `\setcounter`, 40, 46
- `\setlength`, 23, 68, 75
- `\sffamily`, 25
- showidx, 78
- `\sideset`, 58
- `\signature`, 44
- símbolos
  - especiales, 21
  - matemáticos, 83
- sistema de ecuaciones, 51
- slide, 44
- `\sloppy`, 32
- `\slshape`, 25
- `\small`, 25
- `\smallskip`, 29
- Spivak, Michael, 3
- split, 52
- `\sqrt`, 55
- `\stackrel`, 59
- `\stepcounter`, 46, 47
- `\stretch`, 28
- strut*, 67
- subíndices, 16, 57
- `\substack`, 58
- `\sum`, 17
- superíndices, 16, 57
- título, 42
- T1, 34
- tablas, 36, 37
- table, 38
- `\tableofcontents`, 35, 42, 43
- tabular, 36
- `\tag`, 52
- tamaño
  - de los tipos, 25
  - del documento, 44
  - en modo matemático, 56
  - del papel, 44, 75
- TeX, 1
- `\TeX`, 14
- texear, 7
- `\text`, 53
- `\textbf`, 25
- `\textcolor`, 73
- `\textheight`, 75
- `\textit`, 25
- `\textmd`, 25
- texto
  - ajustado, 30
  - centrado, 30
  - modo de, 16
- `\textrm`, 25
- `\textsc`, 25
- `\textsf`, 25
- `\textsl`, 25
- `\textstyle`, 57
- `\texttt`, 25
- `\textup`, 25
- `\textwidth`, 23, 75
- `\the`, 46
- thebibliography, 35, 39
- `\theenumi`, 48
- `\theoremstyle`, 61
- `\thicklines`, 68
- `\thinlines`, 68
- `\thispagestyle`, 41
- tilde, 13, 29, 33, 57
- `\tiny`, 25
- tipo, 24
- `\title`, 42



---

titlepage, 45  
\today, 35  
\topmargin, 75  
\totalheight, 68  
transparencias, 44  
\tt, 26  
\ttfamily, 25  
twocolumn, 45  
twoside, 45  
  
\underbrace, 58  
*underfull box*, 32  
\underline, 58  
\underset, 59  
unidades, 23  
\unitlength, 68  
\upshape, 25  
\usebox, 68  
\usepackage, 18  
  
\value, 46  
\vdots, 54  
\vector, 69  
\verb, 31  
verbatim, 31  
versalita, 24, 25  
vertical, 24  
\voffset, 75  
\vspace, 29  
  
\widehat, 57  
\width, 68  
WYSIWYG, 4  
  
xypic, 71