

TEX/L^ATEX
Manual de iniciación

Pedro Crespo

Junio 1999

Índice

1. Introducción	5
1.1. T _E X y su descendencia	5
1.2. Breve noticia histórica	7
1.3. Proceso de un documento	9
1.4. Controles y entornos	11
1.5. Empezar a escribir	13
1.6. La cuestión del idioma	17
1.7. Estructura general de un documento	20
1.8. Documentos extensos	21
1.9. Depuración de errores	23
1.10. Unidades de medida	25

Índice de figuras

1.1. Proceso de archivos \TeX	10
--------------------------------------------------	----

Índice de cuadros

1.1. Dimensiones del papel	14
1.2. Unidades	25

Prefacio

Pero yo me veo a mí mismo como un niño
que juega en la orilla de una playa
y que a veces encuentra una piedra
más hermosa que otras, mientras la vastedad
del océano se extiende insondable a su lado.

Isaac Newton

Cuentan que en un afamado club de golf se presentó en una ocasión un caballero de aspecto distinguido y serio. Tras pedir la ayuda de un caddie, se dirigió en su compañía al campo.

—Por favor, alcánceme el hierro, ese de platino con incrustaciones de jade—fue su primera petición, y aclaró

—Es un regalo de mi entrañable amigo, el difunto zar de todas las Rusias.

Ensayó el swing varias veces en vacío, y finalmente se decidió a golpear la bola. La tierra levantada salpicó los ojos del caddie, que ni remotamente esperaba una cosa así.

El lanzamiento siguiente terminó contra la rama de un árbol.

—Suerte que no juego en medal-play, porque esto me hubiese costado varios golpes. Los jugadores de golf creen, como los físicos, que los árboles están hechos fundamentalmente de vacío. Pero yo he constatado que eso no pasa de ser un mito más de los que oscurecen este juego— comentó nuestro hombre.

Por fin, tuvo la bola en el green.

—Palo de pat, el corto de madera con adornos de marfil, por favor. Regalo de mi amigo el llorado rey Faruk, por cierto.

Ante el asombro del ayudante, la bola terminó en la laguna cercana. De regreso a la casa club, el muchacho se atrevió a preguntar:

—¿Hace mucho que juega, señor?

—Cincuenta años—respondió el caballero. Y añadió —Incluso tengo escrito un libro sobre el golf.

—¿Cómo se titula, señor?— fue la natural pregunta del caddie.

—Pues ¡El golf y la madre que lo parió!

El cuento —chiste para el que prefiera verlo así— puede aplicarse, mutatis mutandis, al mundo del T_EX, tanto que hemos tenido que refrenar la tentación de bautizar a este manual con el título «El T_EX y la madre que lo parió». Cuesta algo de esfuerzo empezar a tener un cierto dominio sobre el mismo, y no se termina de aprender nunca por completo. Pero, como le ocurre a muchos con el golf a partir de un relativo conocimiento del juego, puede compensar el esfuerzo.

La cita de Newton que encabeza este prefacio no fue dicha naturalmente a propósito del T_EX, sino acerca de las ciencias de la naturaleza, pero puede

aplicarse al mismo, ya que el T_EX encierra en sus entrañas el conocimiento acumulado de generaciones de tipógrafos. Los usuarios corrientes de T_EX manejan una cómoda sintaxis de alto nivel, por lo que en eso se asemejan al niño que juega en la orilla de la playa aludido en la cita. De todos modos, el horizonte estará siempre abierto para que los que así lo deseen se aventuren en las aguas sin límite de este poderoso recurso, y se conviertan de ese modo en verdaderos «T_EXpertos».

Una advertencia, para terminar. La escasa experiencia del autor en la materia comporta necesariamente una consecuencia, y es la de que este manual deba ser considerado meramente una herramienta de iniciación, sin que quepa esperar del mismo un repertorio exhaustivo de explicaciones, instrucciones, posibilidades y soluciones a todos los problemas de la composición de escritos. Pretende, como mucho, ser solamente un vehículo para que los recién llegados al mundo del T_EX se adentren en adelante por mares más prometedores.

Barcelona, Junio 1999.

Nota: *Los cuatro primeros capítulos de este documento han permanecido durante más de un año en estado de hibernación, por lo que ciertas referencias a versiones u otras características evolutivas pueden estar desactualizadas.*

Barcelona, Noviembre 2000.

Mapa de atajos para apresurados

Al par que la rueda, la cama y el calzador, el T_EX quedará para la Historia como uno de esos inventos destinados a formar parte del patrimonio humano. Es por eso que en este manual se dedican algunas páginas de atención a su génesis y a las principales características que inspiran su diseño. Cuando esté por acabarse el siglo XXI, casi con seguridad que estarán extinguiéndose también o lo habrán hecho ya, aparte de los buenos modales, el jaguar y la ballena, de los que si acaso quedarán los respectivos genomas secuenciados, pero seguirán ahí la rueda, la cama, el calzador —no poluciona y no necesita manual de instrucciones, consume poca energía y su forma de uso se transmite por simple observación—, y el T_EX.

De todos modos, y para los que piensan que una herramienta nunca es un fin en sí misma, y también para los que necesitan emplear ésta con urgencia, indicamos aquí los puntos que creemos necesarios —y que esperamos que resulten suficientes— para empezar a escribir con el L^AT_EX, la variante del T_EX que se presenta aquí:

1. Si dispone ya del entorno de trabajo adecuado, es decir
 - Un editor de texto, que le permita escribir, guardar su documento y demás funciones
 - Un procesador de L^AT_EX, en relación con cuya sintaxis está elaborado este manual
 - Un visualizador del documento producido (`.dvi`) y que permita, directa o indirectamente, imprimirlo.
 entonces pase al punto 2. En otro caso, puede optar por
 - a Seguir las indicaciones del apéndice ??, que le darán ideas acerca de cómo instalarlo.
 - b Recurrir a alguien que conozca el asunto y se brinde a instalarle uno.
2. Eche un vistazo a la sección 1.5, que le instruirá acerca de los controles necesarios para empezar a escribir.
3. Si el idioma en que va a escribir es el inglés, puede pasar al punto 4. En otro caso, habrá de hacer frente a problemas como el de los caracteres que no pertenecen al alfabeto inglés, el de la división automática de las palabras, que si no se indica otra cosa sigue las reglas de la gramática inglesa, y otros por el estilo. Las opciones son dos:
 - Si el documento es corto, puede solucionar los problemas aludidos con la ayuda de las instrucciones de los cuatro primeros apartados de la sección 1.6 de la página 17.

- Si el documento es extenso, lo recomendable es instalar las ayudas sugeridas en la sección 1.6 de la página 17.
4. Lea el preámbulo del capítulo ???. Se refiere al reconocimiento de los cambios de párrafo, signos de puntuación y otros elementos.
 5. Repase la sección 1.9, que le ayudará a interpretar los diagnósticos de error con los que posiblemente se tendrá que enfrentar.
 6. Ya está en condiciones de empezar a escribir. Utilice este manual como de consulta, buscando las fórmulas de solución a los posibles planteamientos o requisitos que vayan surgiendo. El capítulo ?? está dedicado a la modalidad de escritura de texto sin más. El capítulo ?? presta atención a las estructuras de las expresiones de tipo matemático. El ?? trata de la incorporación de gráficos e imágenes al documento. El capítulo ?? atiende a las estructuras que afectan al documento en general, tales como las referencias cruzadas, el índice, etc. En cualquier caso, el *Índice* y el *Índice de palabras* (página ??) le ayudarán a encontrar el concepto sobre el cual necesite aclaración.

Capítulo 1

Introducción

Para percibir la distancia que hay entre lo divino y lo humano, basta comparar estos rudos símbolos trémulos que mi falible mano garabatea en la tapa de un libro, con las letras orgánicas del interior: puntuales, delicadas, negrísimas, inimitablemente simétricas.

Jorge Luis Borges,
«La biblioteca de Babel»

1.1. T_EX y su descendencia

El T_EX es una aplicación informática para la composición tipográfica, especialmente preparado para la edición de fórmulas y expresiones matemáticas. La palabra T_EX deriva de la raíz griega de la palabra tecnología, que también significaba «*arte*» para los griegos, y que empieza con las letras $\tau\epsilon\chi$, y es por eso que los puritanos, y también los que no siéndolo profesan la religión T_EX, lo pronuncian de modo reverente «tej» en español, «tecchhh» en inglés, o «tech» en alemán.¹

En la composición mediante ordenador son corrientes desde hace tiempo los programas editores de tipo WYSIWYG, («**What you see is what you get**»), es decir los que van presentando en pantalla, a medida que se escribe, el mismo formato (tipos de letra, márgenes, etc.) que se obtendrá al imprimir el documento. En el caso del T_EX, a diferencia de dichos editores, se escribe en forma de texto llano, y su formato se gobierna mediante determinadas secuencias de control o mandatos. Esto requiere el proceso del archivo de texto mediante un módulo adecuado que le otorga el formato definitivo, proceso semejante al de la compilación de un programa. Lo cual tiene el inconveniente de que en general el usuario no descubre el aspecto final de

¹Donald E. Knuth, «**The T_EXbook**», pág. 1.

su documento hasta haberlo sometido a dicha operación.² Pero esto es una desventaja solamente en apariencia, porque en cuanto se adquiere un cierto dominio del lenguaje \TeX , dicho inconveniente queda compensado con creces por el hecho de que el autor del documento se despreocupa por completo del cuidado del aspecto tipográfico, para concentrarse solamente en el contenido. Los editores de tipo WYSIWYG, por otra parte, tienen el inconveniente de que deben adoptar decisiones de tipo local en relación con el formato del texto, mientras el \TeX (en realidad, todos los programas que se basan en procesar un documento origen en el que además del texto figuran controles para la composición) decide en cuanto al documento en su conjunto, mediante cálculos de tipo global.

El \TeX es multiplataforma, es decir que está pensado para poder trabajar en una variedad de ordenadores equipados con diferentes sistemas operativos. El resultado del proceso de un documento mediante el \TeX es un archivo de tipo estándar, independiente de dispositivo, que se interpreta en el momento de presentarlo o de imprimirlo en un hardware determinado. Aquí se harán las mínimas referencias a una plataforma particular; las que se hagan, ya que siempre conviene proporcionar una idea de la operativa general, se referirán al entorno del PC bajo el sistema operativo DOS o WINDOWS.

La modularidad del \TeX está basada en los **archivos de estilo** (*style files*) y en los **conjuntos de macros** (*macro packages*). Los primeros son definiciones de estilos tipográficos particulares que instruyen al \TeX acerca de cómo confeccionar la presentación de determinadas partes del documento (páginas de título, teoremas, definiciones, etc.). Una **macro** es la definición de un mandato de \TeX en términos de otras elementales o previamente definidas. Las macros pueden ser definidas por el propio usuario, y existen por otra parte potentes colecciones de macros (*macro packages*) para aplicaciones particulares (diagramas en árbol, diagramas de Feynmann para la literatura relativa a la física de partículas, escritura de música, etc.) e incluso para idiomas como el chino o el árabe.

La idea de la modularidad que preside el diseño del \TeX permite que el único cambio a operar en el caso de que un documento se haya de imprimir en una revista de una institución determinada que imponga un estilo basado en un formato específico propio, por ejemplo, consista simplemente en sustituir el archivo de estilo original por el correspondiente a dicha revista, sin que haya que modificar el documento origen. Esta es otra prueba de la gran ductilidad intrínseca propia del diseño del \TeX .

Por otra parte, la familia \TeX forma parte del software gratuito (*freeware*), lo que naturalmente favorece su difusión. La historia del \TeX y de sus descendientes, como el \LaTeX , el \AMS-TeX , etc., al igual que la del **Linux**,

²Existen programas, en sistemas operativos gráficos, que permiten mostrar simultáneamente el resultado en una ventana separada, es decir mientras se escribe el documento *origen* o *fuentes* (*source* según la terminología inglesa).

pertenece a uno de esos fenómenos de resonancia mundial, de característica romántica, donde muchos colaboran con su esfuerzo generoso para bien de una comunidad, en este caso de ámbito planetario.

1.2. Breve noticia histórica

T_EX

El programa T_EX fue diseñado por Donald E. Knuth, profesor de *Computer Science* en la universidad de Stanford³, a finales de la década de los setenta y principios de los años ochenta, a requerimiento de la *American Mathematical Society* (abreviada en adelante como *AMS*). Se cuenta que fue el propio Knuth el que provocó esa petición, como consecuencia de sus críticas a la calidad de las publicaciones de la *AMS*; durante un cierto periodo del año 1974 declinó el presentar trabajos a dicha institución debido al padecimiento que le producía la contemplación del deplorable resultado. Knuth empezó a trabajar en el proyecto T_EX en 1977, y la primera versión estuvo disponible el año 1982. El acierto de su diseño modular y lo ambicioso del proyecto⁴ ha recibido una franca aceptación y ha hecho que se adopte, en sus versiones más evolucionadas, a escala mundial. El número de versión del programa converge hacia el número π , siendo 3,14159 el último valor del que tenemos noticia.

Plain T_EX

La primera versión del T_EX, aunque potente desde el punto de vista tipográfico, constaba de unas 300 primitivas o mandatos de composición elementales, y era por ende un lenguaje de bajo nivel, que requería de mucho trabajo para obtener resultados finales satisfactorios. Al poco tiempo, el mismo Knuth desarrolló una versión de nivel más elevado, el Plain T_EX, que añadía al anterior unas 600 macros, definiciones de mandatos basadas directamente en las primitivas o bien en las propias macros de ellas derivadas.

A continuación se pasa revista a los representantes más conspicuos de los programas descendientes del T_EX. Todos utilizan el T_EX como módulo encargado de la composición tipográfica, es decir en la función de «*motor tipográfico*» («typesetting engine»). Así pues, más que como nuevos programas inspirados en todo caso en el T_EX, estos descendientes deben ser vistos como dialectos de alto nivel basados fundamentalmente en el T_EX, dado que el T_EX sigue siendo la componente nuclear de los mismos.

³Y autor de la monumental obra «*The Art of Computer Programming*», enciclopedia del saber informático de su época, muchos de cuyos capítulos no han perdido vigencia.

⁴Tipográficamente no reconoce límites prácticos: baste decir que la unidad de composición es del orden de la centésima de la longitud de onda de la luz visible.

L^AT_EX

En 1985, Leslie Lamport amplió el Plain T_EX mediante macros y formatos (conjuntos de macros que definen diseños tipográficos), para constituir un lenguaje que permitiera producir con facilidad documentos estructurados, tales como artículos o libros. Esta versión se llamó L^AT_EX,⁵ palabra en la que el prefijo «La» alude a *Lamport*. El L^AT_EX estructura los documentos mediante capítulos, secciones, subsecciones, etc.; permite las *referencias cruzadas*, es decir que brinda la posibilidad de etiquetar no importa qué punto del documento para referirse al mismo en cualquier otra parte sin necesidad de llevar el control de la sección o la página en que se encuentra. Facilita igualmente la numeración de las ecuaciones matemáticas, la formación del Índice (la *Table of contents* o «tabla de contenido», nombre que con más propiedad se le da en inglés), la construcción de índices de palabras (el *Index* o «Índice» de los libros en inglés), de secciones de bibliografía, la inclusión de tablas, figuras, imágenes, acompañando los índices de las mismas, etc. Se comprende así que el L^AT_EX haya tenido una gran difusión en el ámbito científico desde los mismos inicios de su presentación. Actualmente hay libros y revistas científicas escritos en L^AT_EX, y son muchas las imprentas preparadas para obtener los fotolitos directamente a partir del documento original en formato L^AT_EX. Este manual está escrito con la ayuda del L^AT_EX, de modo que, siempre que la copia que esté leyendo se haya obtenido con una impresora de cierta calidad, pueden servir de muestra de algunos de los recursos que ofrece este programa.

L^AT_EX se pronuncia «lay-tecchhh» en inglés («lei-tej» en español).

A_MS-T_EX

Por las mismas fechas que el L^AT_EX, el A_MS-T_EX fue desarrollado, también a partir del T_EX, por Michael Spivak, comisionado a tal efecto por la AMS, y enfocado a las publicaciones de libros, revistas, artículos, etc. de esa entidad. El formato estándar está basado en el «*Journal of the American Mathematical Society*», en lo que hace al tamaño de las hojas y demás convenios tipográficos. El A_MS-T_EX, como se comprende, está especialmente orientado al dominio de la literatura matemática.

L^AT_EX 2_ε

El L^AT_EX dio lugar a otros desarrollos que trataban de ampliarlo en algunos aspectos, y que provocaron problemas de compatibilidad entre las diversas versiones. Para tratar de establecer un estándar se creó en 1994 el equipo *Latex3*, bajo la dirección de Frank Mittelbach. Se trataba de incorporar los requisitos solicitados desde hacía tiempo, además de reunir en una

⁵Sí, en efecto, se presta a chistes fáciles e incluso groseros.

misma versión las aparecidas a partir de la edición \LaTeX 2.09.

El resultado de los trabajos del equipo mencionado es el llamado $\text{\LaTeX}2_{\epsilon}$ (se pronuncia en inglés como «Lay-tecchhh two e», y en fonética castellana como «lei-tej tu i»), aparecido en Junio de 1994 y aceptado actualmente como el nuevo estándar. La versión 3 se anuncia como de próxima aparición.

Una de las características principales que distinguen al $\text{\LaTeX}2_{\epsilon}$ de la versión 2.09 del \LaTeX es el modo como se seleccionan los tipos de letra. En el caso del \LaTeX 2.09, los *Computer Modern Fonts* están contruidos dentro del formato \LaTeX , de modo que personalizar otros tipos de letra resulta muy difícil. En el $\text{\LaTeX}2_{\epsilon}$, por el contrario, son muy pocos los tipos de letra contruidos dentro del formato, y se dispone de mandatos para cargar nuevos tipos de letra, tanto para el modo texto como para el modo matemático. Así, *packages* como *times* y *latexsym* permiten el acceso a dichos tipos de letra. $\text{\LaTeX}2_{\epsilon}$ incluye como estándar la norma NFSS (*New Font Selection Scheme*, «esquema de selección de nuevos tipos de letra», anunciada en 1989) en su versión 2.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ y \LaTeX son complementarios, teniendo el primero mayores posibilidades para la escritura de las expresiones matemáticas, y siendo el segundo más capaz para la elaboración de variadas estructuras de documentos. Esto produjo desde la aparición de ambos productos, que fue casi conjunta, una cierta esquizofrenia entre los potenciales usuarios. Para resolver este estado de cosas, tratando por tanto de combinar la potencia específica del $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ para la literatura matemática con la facilidad y la generalidad del \LaTeX , la *AMS* comisionó de nuevo un proyecto para proporcionar a los usuarios de tales productos la capacidad reunida de ambos, con el resultado del $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$, un híbrido de los dos programas que lo inspiran.

La familia \TeX es utilizada hoy en día a escala planetaria, sobre todo por matemáticos, físicos, ingenieros y todos aquellos científicos que desean presentar sus informes de modo estructurado, en particular si han de expresar parte de la información en forma matemática.

1.3. Proceso de un documento

El documento original (usualmente con extensión **.tex**, aunque no es obligatorio) se escribe en modo texto ASCII⁶ y, tal como indica la figura 1.1, se procesa mediante el correspondiente módulo (**tex**, **latex**, etc.), para producir un archivo de extensión **.dvi** (**d**evice **i**ndependent, es decir independiente del dispositivo destinado a mostrarlo o a imprimirlo). Este paso,

⁶ *American Standard C*ode for *I*nformation *I*nterchange («Código americano estándar para el intercambio de información»).

si se ejecuta en modo de mandato, es del estilo

```
latex documento
```

donde se supone que *documento.tex* es el nombre del documento (si la extensión es *.tex*, como se supone que ocurre en este caso, se puede omitir; en otro caso hay que escribirla, o terminar el nombre del archivo con un punto si carece de ella).

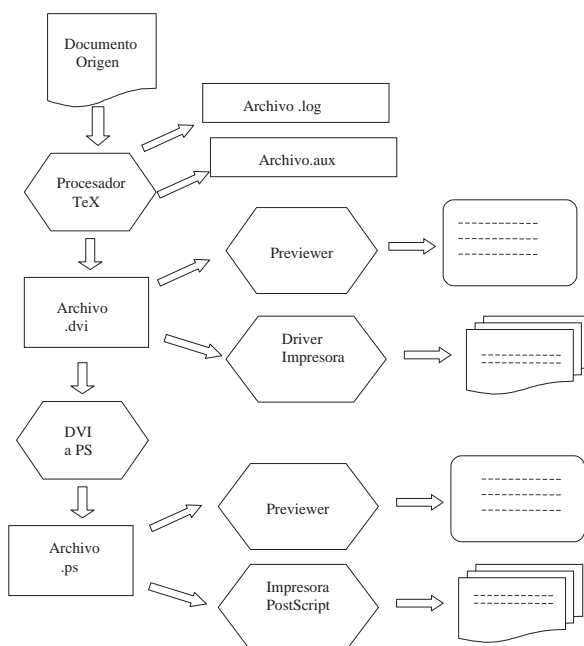


Figura 1.1: Proceso de archivos \TeX .

Los programas adecuados permiten que el archivo *.dvi* resultante se pueda ver por pantalla⁷ o que se escriba por impresora. En el primer caso, el proceso se conoce en general como *preview* («visión previa»), y los módulos o programas que intervienen, basados en los *drivers* de pantalla, se llaman *previewers*. Los módulos que intervienen en el segundo caso manejan los *drivers* correspondientes a la impresora de que se trate.

La independencia del archivo *.dvi* de los dispositivos finales hace posible que dicho archivo pueda transferirse desde el ordenador en el cual se ha preparado a otro cualquiera, en el que se podrá tratar con los módulos adecuados para ver o imprimir el documento.

Otros archivos que se producen típicamente en el proceso del texto origi-

⁷Siempre que el conjunto formado por el adaptador y el monitor tenga capacidad gráfica, configuración que actualmente casi no tiene excepciones.

nal son los de extensión `.log` («registro», o «cuaderno de bitácora»), y `.aux` («auxiliar»). El primero registra la traza de los pasos del proceso de conversión del archivo original, incluyendo los mensajes de aviso o de error si los hubiera; en este caso acudir al citado archivo puede resultar útil para analizar los problemas habidos. El segundo es un archivo de trabajo del propio procesador L^AT_EX. Se trata de archivos tipo texto.

Siempre es posible obtener el archivo resultante del documento en formato *PostScript*, con extensión `.ps`, para ser resuelto directamente mediante dispositivos tipo *PostScript*. En modalidad de mandato el documento se transforma mediante un paso del estilo

```
dvips documento
```

proceso que lee el archivo `documento.dvi` y produce como resultado el nuevo archivo `documento.ps`. Una impresora tipo *PostScript* puede imprimir directamente este archivo. En todo caso, el software adecuado permite tratarlo, tanto si es para mostrarlo por pantalla como para imprimirlo mediante la preparación que corresponda aun cuando la impresora carezca del dispositivo adaptador *PostScript*.

1.4. Controles y entornos

Cuando se escribe para el L^AT_EX, tanto si se trata de texto en general como de expresiones de tipo matemático, se está alternativamente en una de dos modalidades: la de escribir *texto literal*, por una parte, y la de escribir *controles* o *mandatos* para el gobierno del formato del documento. En cuanto a estos últimos, destinados a ser interpretados por el programa que procesa al texto original, son reconocibles porque comienzan con el símbolo `\` en el papel de prefijo marcador. Conviene tener clara la distinción entre los dos tipos o formatos principales:

palabras de control que se definen como la secuencias que constan de la barra `\` seguida de cualquier combinación de letras del alfabeto básico (no acentos, ni cifras numéricas, ni caracteres especiales, aunque se distingue entre las minúsculas y las mayúsculas⁸). Las palabras de control, por tanto, terminan con un símbolo cualquiera que no sea una letra del alfabeto, como por ejemplo un espacio o una cifra. En el caso particular en que la palabra de control termina con un espacio, se ignora dicho espacio y los que puedan venir a continuación; en los casos en que se quiere imponer un espacio inmediatamente después de una palabra de control, hay que utilizar el símbolo `\` seguido de un

⁸La palabra de control es sensible al cambio de una letra minúscula por su correspondiente versión en mayúscula. Para las palabras de control estándar se emplean en general las letras minúsculas, lo cual disminuye el riesgo de confusiones.

espacio, o bien la pareja de llaves `{ }` seguida también de un espacio en blanco. Así, siendo `\TeX` y `LaTeX` las palabras de control que dan como resultado las palabras `TeX` y `LaTeX` respectivamente, para obtener la frase

El `LaTeX` es una maravilla. También su motor el `TeX`, claro.

hay que escribir

```
El \LaTeX\ es una maravilla.
También su motor el \TeX, claro.
```

Después de la palabra `LaTeX` hay que introducir un espacio; luego de la palabra `TeX` no es necesario, sin embargo, porque el procesador reconoce el final de la palabra de control al encontrar la coma.

Si se escribe

```
Según Knuth, existen \TeX nicos y \TeX{}pertos.
```

resultará, de acuerdo con las normas anteriores

Según Knuth, existen `TeXnicos` y `TeXpertos`.

Como se explicará, las palabras de control admiten a veces *argumentos*, lo que potencia grandemente las posibilidades de composición.

símbolos de control formados por la barra `\` seguida de cualquier símbolo no alfabético. Los símbolos de control sirven para escribir símbolos especiales, o para instrucciones de control elementales. A diferencia de lo que ocurría en el caso anterior, el `LaTeX` distingue el carácter que viene inmediatamente después de un símbolo de control, aunque se trate de un espacio; en este último caso se ignoran los espacios que puedan seguir a continuación.

Como ejemplo tenemos el símbolo de control `\%`, empleado para obtener el signo de porcentaje en la frase

Se aplica un descuento del 12 % sobre el precio marcado.

escrita como sigue:

```
Se aplica un descuento del 12\% sobre el precio marcado.
```

Naturalmente, tanto las palabras como los símbolos de control han de pertenecer al repertorio reconocido por el `LaTeX`, o de otro modo el correspondiente mensaje advertirá del error.

Los entornos

Otro concepto importante del que conviene hablar ya porque hará su aparición en breve es el de **entorno**. Los entornos proporcionan al L^AT_EX potentes recursos, que le permiten manejar separadamente bloques del texto para determinados propósitos (citas, diversos alineamientos), así como construir listas, tablas, formar columnas, insertar figuras, escribir ecuaciones matemáticas, etc. Los entornos comienzan con el control

```
\begin{nombre-entorno}
```

y terminan con el control

```
\end{nombre-entorno}
```

representando **nombre-entorno** la palabra específica del entorno de que se trate, de modo que ambos actúan como marcas de inicio y final de un grupo que puede contener a su vez otros entornos anidados en su interior, con el resultado de que todos los mandatos actúan localmente en el interior de cada entorno, sin afectar al texto exterior al mismo. Aunque no sea obligado, en aras a una mejor legibilidad del documento origen se aconseja escribir las marcas de inicio y de fin en líneas separadas del resto del texto.

Los argumentos

Tanto las palabras de control como los entornos admiten argumentos, es decir ciertas claves o valores que hacen el papel de parámetros que modulan su función. Los argumentos pueden en general ser de dos tipos: *optativos* u *obligatorios*, según que puedan obviarse —en cuyo caso el sistema adoptará los valores por omisión que tenga previstos— o que deban indicarse necesariamente. En este manual, cuando se escriba

```
\control[xxx]{yyy}
```

(donde **control** hace el papel de una palabra de control dada), deberán considerarse optativos los argumentos encerrados entre corchetes y obligatorios los que van entre llaves. En cada caso, naturalmente, se indicarán los valores pertinentes de los parámetros.

1.5. Empezar a escribir

Para que un texto sea inteligible para el L^AT_EX, es preciso antes que nada declarar las características del documento que se va a escribir, lo que se hace mediante el control

```
\documentclass[opciones]{tipo-documento}
```

en donde

tipo-documento se refiere al estilo de documento que se quiere producir, y debe indicarse siempre. Los tipos de documento básicos son

- **article** («artículo»), para comunicaciones a publicar en revistas científicas, informes breves, invitaciones, etc.
- **report** («informe»), para escritos algo más extensos conteniendo varios capítulos, panfletos, tesis doctorales, y otros documentos por el estilo.
- **book** («libro»), para libros en el sentido más completo.
- **letter** («carta»), previsto para configurar la estructura de una o varias cartas.
- **slide** («transparencia») para producir material con destino a presentaciones, tales como transparencias⁹.

opciones constituye una lista de características optativas, separadas por comas, del tipo de las siguientes:

- las que indican el tamaño del papel, que por omisión corresponde al formato americano **letterpaper** (carta). Las opciones posibles son:
a4paper, **a5paper**, **b5paper**, **letterpaper**, **legalpaper**, **executivepaper**.

La tabla 1.1 muestra las dimensiones en milímetros de cada uno de los diversos tamaños de papel antes indicados.

tamaño	ancho mm.	largo mm.
a4paper (DIN A4)	210	297
a5paper (DIN A5)	148	210
b5paper	176	250
letterpaper	215'9	279'4
executivepaper	184'15	266'7
legalpaper	215	355'6

Cuadro 1.1: Dimensiones del papel

- las que establecen el tamaño de la letra, tal como **11pt** o **12pt**, si se desea elegir un tamaño de letra distinto de **10pt**, que es el valor que se adopta por omisión. En lo anterior **pt** es la unidad de medida **punto**, descrita en la sección 1.10.

⁹Puede ser papel, si se va a utilizar un proyector por reflexión, o se van a mostrar las hojas directamente.

- las que distinguen el formato de las páginas pares del de las impares. Si el tipo de documento es *article* o *report*, por omisión se adopta **oneside**, que no hace distinción de formato entre las páginas pares y las impares. En el caso de que se elija *book* para el tipo de documento, el valor por omisión es **twoside**, que distingue en cuanto a sus márgenes las páginas pares de las impares. La opción **twoside** afecta solamente al formato de las páginas, y no comporta ninguna orden al *driver* de la impresora para que imprima por ambas caras, como cabría pensar; esta es una decisión que el usuario deberá adoptar al imprimir. Si la impresora no es capaz de imprimir ambas caras a un tiempo, siempre cabe la posibilidad de hacerlo en dos pasadas, una dedicada a las páginas impares y otra a las pares, cuidando de realimentar adecuadamente el conjunto de hojas luego de la primera pasada.
- la que indica si se escribe en una columna (**onecolumn**) o en dos (**twocolumn**), al estilo de una revista. Por omisión se adopta **onecolumn**.
- La que controla si los capítulos comienzan en la página de la derecha (**openright**) o si ello es indiferente, de modo que el inicio de los mismos tendrá lugar en la siguiente página disponible (**openany**). Esta opción no se aplica al tipo de documento *article*, dado que no puede estructurarse en capítulos. Si el estilo es *book*, se adopta por omisión **openright**, y si es *report*, **openany**.
- Las opciones **titlepage** y **notitlepage** indican si después del título del documento se debe continuar en una página nueva o no, respectivamente. El estilo *article* tiene **notitlepage** como opción por omisión, mientras que **titlepage** se aplica si no se dice nada en contra a los estilos *report* y *book*.
- Las que fijan la orientación del papel, con las opciones **portrait** («retrato», o vertical), que es el valor por omisión, y **landscape** («paisaje», horizontal o apaisado).
- La opción **draft** («borrador») señala con un trazo vertical al margen los puntos en los que se ha producido una condición de desbordamiento (*overflow*) de los márgenes del texto. En el caso de que se hayan incluido gráficos o imágenes, se muestra solamente el marco de los mismos, lo que ahorra bastante tinta al imprimir pruebas, y, si el visualizador es lento, tiempo en el proceso de presentación. La opción **final** es la opuesta a la anterior, y la que se adopta por omisión.
- La opción **fleqn** (*flushleft equation*) hace que las fórmulas matemáticas escritas en la modalidad del entorno *displaymath* («presentación matemática», ver sección ?? en la página ??) se escri-

ban alineadas a la izquierda, en lugar de centradas (formato por omisión).

- La opción `leqno` (*left equation number*) hace que, de numerarse las ecuaciones (ver sección ?? en la página ??), el número de la fórmula se escriba a la izquierda en lugar de a la derecha de la misma (formato por omisión).

Ninguna de las opciones es obligatoria, porque cada una tiene definido su correspondiente valor por omisión.

El presente documento utiliza la declaración

```
\documentclass[a4paper,11pt]{report}
```

Una vez declarado el estilo del documento, sólo se necesita indicar el inicio y el final del mismo, mediante los controles del entorno *document*:

```
\begin{document}
```

y

```
\end{document}
```

y colocar el escrito entre ambas marcas. Lo escrito en el interior del entorno *document* se llama **cuerpo** del documento; las líneas que se escriben entre la declaración del tipo de documento (`\documentclass`) y el mandato de inicio del entorno *document* (`\begin{document}`) constituye lo que se llama el **preámbulo** del documento. Como veremos, ciertas declaraciones se incluyen en esta zona. Lo escrito después del entorno *document*, es decir a partir del control `\end{document}`, es ignorado por el procesador.

Si se escribe en inglés, esto basta, en principio, para producir un documento de tipo narrativo. Una estructura mínima aunque suficiente podría responder al esquema del ejemplo siguiente:

```
\documentclass{book}
\begin{document}
Once upon a time, in a no-space no-time place, a fluctuation of
the vacuum exceeded its critical value, and the entire world was
created from nothing. This is actually the only neverending tale.
\end{document}
```

Pero si dicho documento se escribe en un idioma distinto, se presentan algunas dificultades, como enseguida examinaremos; para facilitar las cosas convendrá introducir controles relativos al idioma, o acudir a *packages* auxiliares, lo cual se considera también a continuación.

1.6. La cuestión del idioma

Cuando se escribe en un idioma distinto del inglés, que es el nativo del T_EX, surgen varios aspectos que dificultan la escritura:

Los caracteres especiales

Los idiomas como el español, el catalán, el francés, etc. comparten con el inglés el alfabeto básico, pero lo amplían —por así decirlo— con caracteres especiales tales como las vocales acentuadas, la letra ñ, la ç, etc. El L^AT_EX permite obtener dichos caracteres especiales mediante combinación de otros símbolos, es decir mediante secuencias de control del tipo

`\’a`

que en este caso produce la letra a acentuada. La sección ?? muestra de forma más completa el modo de obtener tales caracteres acentuados y otros especiales. Como se comprende, esto no evita que resulte incómodo a la vez que tedioso tener que componer de tal modo esos caracteres, en general bastante frecuentes en el texto. Por otra parte, el módulo de división de palabras no funciona bien con las palabras escritas de modo ampliado.

El paquete *inputenc*

Una ayuda consiste en el uso del package *inputenc*, mediante una declaración del estilo de la siguiente:

`\usepackage[latin1]{inputenc}`

donde `inputenc.sty` es el archivo que define al package, y `latin1` es la opción adecuada para escribir en español (`latin3` correspondería al catalán).

Este package facilita la escritura de las letras acentuadas, en el sentido de que la a acentuada se escribirá ahora mediante la secuencia `\’a`. En particular, para obtener las vocales acentuadas, este sistema requiere las mismas pulsaciones que con el sistema ordinario, el cual exige pulsar la tecla muerta del acento seguida de la que corresponde a la vocal de que se trate. Por otra parte, signos como los de apertura de interrogación (`\?`) y de admiración (`\!`) pueden escribirse directamente del teclado.

No obstante, este paquete presenta un inconveniente, debido a que las palabras que se entregan al módulo que calcula la oportunidad de dividir las por cambio de línea —y que las divide efectivamente si procede— tienen la forma ampliada en cuanto a los caracteres especiales y letras acentuadas; así, la palabra *crónica* se presenta a la entrada de dicho módulo como `cr\’onica`, lo cual impide que se pueda dividir de la forma correcta.

El paquete *tlenc*

Una alternativa al paquete anterior, y que representa una solución cómoda al problema que consideramos, es el *tlenc*, que se declara como

```
\usepackage{tlenc}
```

donde `tlenc.sty` es el archivo correspondiente al paquete. Este paquete permite utilizar directamente las letras acentuadas, así como la letra ñ y otros caracteres especiales, lo cual hace posible que la división de las palabras por cambio de línea se lleve a cabo adecuadamente (ver punto ... de Babel). Un inconveniente, aunque comparativamente pequeño, es que en este caso los signos de apertura de interrogación (¿) y de admiración (!) han de componerse en la forma `?` y `!` respectivamente. Se añade también otro inconveniente que el usuario sufre sólo indirectamente: el paquete *tlenc* utiliza un tipo de letra distinto del estándar *OT1*, el *T1*, y eso provoca que el *previewer* y el propio procesador necesiten cargar el nuevo tipo, aunque se trata de una operación que se lleva a cabo solamente la primera vez en que tiene lugar dicho proceso.

Este manual se ha escrito con la ayuda del paquete *tlenc*, lo que de todos modos no ha de tomarse como una recomendación declarada.

La división de las palabras

Al justificar los párrafos, es decir al ajustar las líneas del texto a los márgenes definidos para el mismo, se impone a veces partir las palabras al cambio de línea, de acuerdo con reglas que son específicas del idioma de que se trate. El \LaTeX proporciona el medio de señalar cómo deben dividirse con guiones las palabras que el usuario desee indicar; se emplea para ello el control `\hyphenation`, tal como en

```
\hyphenation{tre-nes Bar-ce-lo-na Za-ra-go-za}
```

Las palabras que se incluyen en la lista son entendidas por el \LaTeX con independencia del empleo de letras mayúsculas o minúsculas, de modo que si se escribe `lu-na` se dividirán del mismo modo las palabras *luna*, *Luna*, *LUNA*, etc. Por otra parte, si no se quiere que la palabra *átomo*, por ejemplo, quede dividida, bastará incluirla en la lista en la forma `átomo`, sin ningún guión intermedio.

Como es natural, bastará (y es lo aconsejable, por otra parte) con hacer estas indicaciones —para las palabras que han quedado mal divididas— en la etapa final de la preparación del documento, es decir en la fase equivalente a la revisión de galeras, hasta dejar el documento definitivamente compuesto. Pero, aparte de representar un trabajo no despreciable, obliga a conocer bien las reglas para dividir las palabras. Este problema, y el de los

textos automáticos que se examina a continuación, se resuelve mediante el paquete **babel**.

Los nombres de las partes del documento

EL procesador incorpora en ciertas ocasiones al documento textos producidos automáticamente. Así, por ejemplo, y si no se indica otra cosa, el procesador nombra las diversas partes del documento con la terminología inglesa: *Chapter* para Capítulo, *Section* para Sección, *Table of Contents* para el Índice, etc. Para obtener la traducción correspondiente, se pueden emplear controles del estilo de los siguientes:

```
\def\chaptername{Cap\'}\i tulo}
\def\sectionname{Secci\'}on}
\def\appendixname{Ap\'}endice}
\def\contentsname{\'}Indice}
\def\indexname{\'}Indice de palabras}
```

De todos modos, y como se ha dicho más arriba, el paquete **babel** contempla este problema, haciendo innecesario el tener que introducir definiciones del tipo de las anteriores.

El paquete *babel*

El paquete multilanguage **babel** de Johannes Braams, ofrece una solución estándar a los mencionados problemas de la división de las palabras y de los textos incorporados automáticamente para idiomas distintos del inglés. Da soporte a una treintena de idiomas, entre otros el español, el catalán y el gallego, y permite incluso alternar en un mismo documento idiomas distintos. En cada caso aplica las reglas adecuadas para la división de las palabras por razón del cambio de línea (siempre que las palabras se entreguen al módulo de división —*hyphenation*— escritas en su forma definitiva); estas reglas son distintas según el idioma, y representaría como ya hemos dicho un cierto trabajo el ocuparse de dicho aspecto. Por otra parte, los textos incorporados automáticamente, tales como los nombres de las partes del documento (Capítulo, Sección, Índice, etc.) se establecen directamente en el idioma elegido.

Ver apéndice C para la información acerca de cómo activar el paquete **babel**. Cuando se adopta dicha solución, la declaración del documento, queda como sigue (el ejemplo supone que se escribe en español):

```
\documentclass[a4paper,11pt]{report}
\usepackage{t1enc}
\usepackage[spanish]{babel}
\begin{document}
```

```

.....
.....
aquí el texto del documento ....
.....
.....
\end{document}

```

Los controles anteriores bastan para empezar a escribir un documento, lo que permitirá comprobar las posibilidades de composición que se muestran en los capítulos que siguen, así como ensayar las variantes que puedan interesar. La sección siguiente enseña a estructurar el documento de modo jerárquico, lo que puede dejarse para una etapa posterior si se prefiere.

1.7. Estructura general de un documento

Un documento no es casi nunca un texto sin soluciones de continuidad, sino que consta de partes diferenciadas, como son los capítulos o las secciones, aparte de otras componentes como la página de título, el índice, etc. Consideramos a continuación estas primeras componentes de la estructura general de un documento. En cuanto a la página de título, el índice, etc., se tratan más adelante, en el capítulo ??.

Los documentos pueden estructurarse según las siguientes divisiones de jerarquía:

```

\part (parte)

\chapter (capítulo)

\section (sección)

\subsection (subsección)

\subsubsection (sub-subsección)

\paragraph (párrafo)

\subparagraph (subpárrafo)

```

Cada uno de los controles anteriores tiene como argumento el nombre que encabeza la parte, el capítulo, la sección, etc. El propio L^AT_EX establece los adecuados tamaño de la letra, espaciado, numeración, etc. de las diversas divisiones. Si no se desea que la división quede numerada (es lo que se ha hecho para las subsecciones de la sección 1.6), basta con anteponer un asterisco al argumento, como en

```
\subsection*{subsección no numerada}
```

En el caso de los capítulos (*report* y *book*), el L^AT_EX destina una página nueva al inicio de cada uno. En cuanto a la numeración, se lleva a cabo de forma jerarquizada, de modo que la primera sección del segundo capítulo, por ejemplo, se numerará como 2.1, y su primera subsección, de existir, llevaría el número 2.1.1, etc.

1.8. Documentos extensos

Un documento puede ser escrito mediante un solo archivo de texto, siempre que se disponga de un editor capaz de manejarlo. Aun así, no obstante, si un documento tiene una extensión algo considerable —más de unas cincuenta páginas, por ejemplo—, puede resultar conveniente separarlo en partes (los diversos capítulos, por ejemplo), y proceder a escribir y procesar para la comprobación cada parte por separado. Esto puede irse controlando, como enseguida veremos, mediante un documento *maestro* o *raíz*, que organice el conjunto y permita indicar en cada etapa cuál es el documento parcial o la combinación de documentos parciales sometidos al proceso, y que por supuesto trate fácilmente el conjunto global en la fase final de elaboración. Con este método se obtienen dos ventajas principales:

- Cada texto parcial es más manejable, resultando más fáciles las búsquedas que en ocasiones hay que realizar en el mismo cuando se llevan a cabo modificaciones o correcciones.
- Cada vez que se procesa el texto que se está escribiendo, con el fin de hacer las correspondientes comprobaciones, el tiempo de proceso es menor que si se trabajase con la totalidad del documento, dado que el L^AT_EX no tiene que trabajar toda la extensión del documento completo.

Dado que se va a aludir pronto al mismo, adelantamos aquí el papel del símbolo %, (símbolo de comentario) que se emplea usualmente para anotaciones internas, y del que se tratará especialmente en la sección ???. Este carácter provoca que el L^AT_EX ignore el resto de la línea al procesar el archivo de entrada. Así, si se escribe

```
se aplica un descuento del 12% sobre el precio marcado.
```

```
resultará la expresión
```

```
se aplica un descuento del 12
```

Supongamos ahora que se quiere escribir un documento cuyo nombre es, por ejemplo, `nomb.tex`, y que consta de varios capítulos. En tal caso, un enfoque operativo consiste en escribir cada capítulo en un archivo separado, `nombcap1`, `nombcap2`, etc. El documento maestro, que contendrá todos los documentos parciales por preparar, será un texto del tipo siguiente:

```

\documentclass[a4paper,11pt,twoside]{report}
\usepackage{t1enc} % acentos y otros símbolos especiales
\usepackage[spanish]{babel} % idioma español
\usepackage{graphicx} % para incluir gráficos e imágenes
\begin{document}
%\input{nombpref} %prefacio
%\input{nombcap1} % introducción
%\input{nombcap2} % que trata de ...
...
%\input{nombcapN} % último capítulo
% Los apéndices empiezan aquí
\appendix
%\input{nombcapA}% apéndice A
...
%\input{nombcapD}% apéndice D
\end{document}

```

en donde cada capítulo se integra en el conjunto mediante el control

```
\input{nombre del capítulo}
```

(el nombre sin la extensión `.tex`). Este control incluye el archivo nombrado en el texto global, en el mismo punto en el que se le llama. Como puede verse, aparecen comentados (el símbolo `%` al inicio de una línea la desactiva para el proceso, como se ha explicado más arriba), y por tanto sin efecto, las diversas partes constituyentes del documento global: `nombpref`, `nombcap1`, `nombcap2` A medida que se escribe o que se trabaja con cada una de las partes o capítulos, bastará con ir activando (eliminando el símbolo `%` de comentario de la posición inicial) o suspendiendo (comentando, mediante el símbolo `%` colocado al inicio de la línea) cada una de dichas partes, para que el proceso del documento maestro la incluya o la ignore. De este modo tenemos de un modo sencillo un control sobre la parte del documento —o la combinación de partes— con la que deseamos trabajar. En esta etapa, naturalmente, no se resolverán detalles que dependan del conjunto en su totalidad, como por ejemplo las referencias cruzadas, lo cual por otra parte no tiene mayor importancia (aparecen en el documento resultante señaladas mediante la secuencia `??`) ya que de todos modos no adoptarán su expresión definitiva hasta el final del proceso completo.

Otra alternativa para conseguir trabajar con el documento o los documentos parciales que se desee, procesando siempre el documento maestro como principal (el mandato de proceso sería por tanto siempre `latex nomb`), consiste en incorporar cada capítulo mediante el control

```
\include{nombre del capítulo}
```

(de nuevo el nombre sin la extensión `.tex`). No es necesario desactivar o activar mediante el símbolo de comentario (`%`) ninguna de dichas inclusiones,

ya que la selección se hace mediante un control colocado en el preámbulo del documento, el cual indica los documentos parciales que se desean incluir en el proceso:

```
\includeonly{nombcapa,nombcapb, ... ,nombcapn}
```

que como se ve lista, separados mediante comas, los archivos parciales a procesar, nombrados también aquí sin su extensión.

El mandato `\include`, a diferencia del `\input` antes mencionado, provoca un avance de página antes de tratar el documento incluido. Esto conviene a veces, porque mantiene más estable la presentación resultante de cada documento parcial, ya que no se altera la paridad de su página inicial.

Cuando todos los capítulos estén escritos y comprobados individualmente, es llegado el momento de la composición final. Entonces bastará con activar todos los documentos parciales, de modo que al procesar el documento maestro se obtenga el documento global. En esta etapa final deberá comprobarse si se han resuelto bien las referencias cruzadas entre elementos de las partes inicialmente separadas. Por otra parte, si por ejemplo se había elegido al comienzo un tamaño de letra pequeño por razones de economía de papel en el caso de imprimir para revisar el documento, esta es la ocasión de restituir el tamaño elegido como definitivo. Ahora conviene preocuparse también de la página de título, del índice, del índice de palabras, etc., temas de los que se trata en el capítulo ???. Por otra parte, este es también el momento en que se puede decidir imponer un avance de página en aquellos casos en que el propio L^AT_EX haya producido un formato que por alguna razón no nos resulte estético o suficientemente claro, como es el caso en que quedan «*líneas viudas*» al final de una página que resultan mejor al inicio de la siguiente, etc. Este trabajo debe hacerse de forma incremental desde el inicio del documento hacia adelante, para que los nuevos arreglos no incidan negativamente en lo ya conseguido; y únicamente en la fase de corrección definitiva, o de otro modo cualquier cambio puede alterar lo ya ajustado, obligando a repetir el trabajo.

En el caso en que se quiera elaborar un *índice de palabras* —ver sección ?? en la página ??— las que se marcan con destino al mismo pueden irse indicando mientras se escribe el documento, pero la experiencia aconseja dejar esa labor para el final, porque solamente entonces se reconoce mejor cuáles deben marcarse de forma estructurada, aparte de que se ahorra tiempo llevando a cabo de modo sistemático las necesarias búsquedas a lo largo de todo el documento.

1.9. Depuración de errores

Al procesar el archivo de texto origen `.tex`, los mensajes de error o de aviso se presentan por pantalla, con indicación del diagnóstico, y del número

de la línea del archivo de texto en el que ha tenido lugar el conflicto. Si escribimos, por ejemplo,

```
Tendremos un \error luego de la palabra un
```

al procesar el archivo obtendremos un mensaje de error (se supone que `\error` no se ha definido como macro) del estilo del siguiente:

```
! Undefined control sequence.
1.719 Tendremos un \error
      luego de la palabra un
```

el cual nos indica

- El diagnóstico (precedido por el signo !), que en este caso advierte que una pretendida palabra de control (`\error`) no se reconoce como definida.
- Un número, 1.719 por ejemplo, que indica el número de línea del texto original donde ha sido detectado el problema. En este caso la línea señalada es la 719 del archivo de texto original (hay que prescindir del prefijo 1.).
- El punto en el cual se ha detectado el problema, ya que la línea conflictiva queda separada en dos a la altura de dicho punto. En el caso del ejemplo, la línea se divide en el lugar en el que se encuentra el falso mandato `\error`.

El mensaje de error se detiene con el cursor situado a continuación de un signo de interrogación, indicando que se solicita una respuesta por parte del usuario. Si se contesta con un signo de interrogación (?), el procesador muestra las respuestas aceptables (valen las minúsculas):

<Intro> hace que se ignore el error y continúe el proceso.

S continúa el proceso, ignorando el error presente así como los que pueda hallar en adelante.

R sigue con el proceso sin detenerse.

Q continúa el proceso sin mostrar los mensajes, ignorando también los errores.

I permite insertar la versión correcta de la secuencia que ha provocado el error. Así, si en el ejemplo anterior se contesta con `I\%` (que como ya hemos visto sirve para escribir el signo %), la frase resultante será

```
Tendremos un % luego de la palabra un
```

lo que no significa, sin embargo, que se corrija el texto en el documento original. Para ello será necesario editarlo y llevar a cabo la oportuna corrección en el mismo.

E edita el documento original.

H (de *help* indica más in extenso el origen del error y cómo subsanarlo.

X termina abruptamente el proceso.

En principio, no se necesita más para encontrar y depurar el error producido. De todos modos, cuando se empieza a utilizar el \TeX o cualquier otro de los procesadores de la familia, el bregar con los errores suele ser una dura tarea. La experiencia desarrolla un olfato especial para el reconocimiento de los problemas, que en muchas ocasiones hace innecesario incluso la lectura del diagnóstico. El síndrome de la depuración de errores del \TeX lo reconocerán pronto los que hayan programado alguna vez en lenguajes procedurales, porque es un proceso similar.

1.10. Unidades de medida

El \LaTeX maneja diversas unidades para las dimensiones lineales, que se muestran en la tabla 1.2, y a las que se harán referencias en este manual.

cm	centímetro	
mm	milímetro	1 cm = 10 mm
in	inch (pulgada)	1 in = 25.4 mm
pt	point (punto)	1 in = 72.27 pt
bp	big point (punto grueso)	1 in = 72 pt
pc	pica	1 pc = 12 pt
dd	didot point (punto didot)	1157 dd = 1238 pt
cc	cicero (cíceros)	1 cc = 12 dd
sp	scaled point (punto escalado)	65536 (= 2^{16}) sp = 1 pt

Cuadro 1.2: Unidades

\TeX reconoce también dos unidades de medida relativas al tipo de letra que se utiliza:

El **quad** («*cuadrado*») es una de las medidas que provienen del mundo de la imprenta mecánica, y viene a ser equivalente al cuadrado que define la caja de la M del tipo de letra de que se trate, dado que es una medida de carácter relativo.

Otras:

em es la anchura de un **quad** (del tipo actual de letra)

ex es la unidad de altura de la letra x (x-height) (del tipo actual de letra)

En consecuencia, **em** es una unidad de dimensión lineal que vale la anchura del **quad**, y **ex** es la unidad de dimensión lineal que vale por la altura de la letra x (repetimos: del tipo y tamaño de letra que se utiliza). En el T_EX y con los nuevos caracteres *Computer Modern Fonts* estas unidades se han precisado, de forma que para el Roman 10pt se tiene $1\text{em} = 10\text{pt}$ y $1\text{ex} \approx 4.3\text{pt}$. En general, **em** se aplica a medidas horizontales y **ex** a las verticales, y sus valores absolutos dependen como queda dicho del tipo de letra.

La unidad más pequeña es, como puede verse en la tabla, el **sp**, que equivale aproximadamente a unos $5 \cdot 10^{-7}\text{ cm}$. A efectos comparativos, digamos que las longitudes de onda del espectro de la luz visible están comprendidas en el margen que va de los $4 \cdot 10^{-5}\text{ cm}$, a los $7 \cdot 10^{-5}\text{ cm}$, lo que significa que son del orden de un centenar de **sp**.

Capítulo 2

Modo texto

Polonio —¿Qué estáis leyendo, señor?

Hamlet —Palabras, palabras, palabras

William Shakespeare,

«Hamlet», acto II, esc. II

Al escribir texto ordinario para el L^AT_EX no hay que preocuparse para nada del formato del mismo, porque el propio procesador se encarga de resolver este aspecto. Quiere esto decir que no es necesario estar pendiente de que la separación entre palabras sea exactamente de un espacio, ni de ajustar los márgenes, por ejemplo. De hecho, una entera narración podría escribirse sin atender en absoluto a las cuestiones de formato. Pero, en general, eso no vale para un documento de tipo científico o técnico, en el cual es corriente tener que hacer uso de listas, tablas, etc., por no hablar todavía de la cuestión de las expresiones de tipo matemático. Así pues, conviene conocer las posibilidades que ofrece el L^AT_EX para controlar ciertas particularidades del formato del texto.

Este capítulo se ocupa de algunos aspectos que se refieren a la modalidad de texto convencional, en la que no intervienen fórmulas matemáticas.

2.1. Reglas generales

Antes de examinar los aspectos de tipo estructural, la sección siguiente pasa revista a ciertas consideraciones que deben ser tenidas en cuenta con carácter general.

Los espacios en blanco: Los espacios en blanco, incluidos los indicados por los tabuladores, se consideran del mismo modo. Varios espacios en blanco consecutivos equivalen a uno solo. El final de una línea equivale asimismo a un espacio en blanco. Los espacios en blanco al inicio de

una línea se ignoran en general, así que la primera palabra de una línea queda separada por un espacio de la última de la línea precedente.

El punto y seguido: El L^AT_EX reconoce el final de una frase (hablamos del punto y seguido), por el punto (.), el signo de cierre de la interrogación (?) y el que cierra la admiración (!), seguidos en cada caso de un espacio en blanco. Hay una excepción, y se da en el caso en que el punto está situado a continuación de una letra mayúscula, como ocurre por ejemplo en *Donald E. Knuth*, caso en que no procede dar por terminada la frase tras la abreviatura E.; a pesar de todo, si la frase debe terminar después de una mayúscula, como por ejemplo si ha de acabar con las palabras *vitamina C*, hay que forzar el punto mediante el símbolo de control \@, escribiendo en ese caso

```
vitamina C\@.
```

En inglés, el espacio que termina una frase es algo mayor que el que sigue al punto correspondiente a una abreviatura, a diferencia de lo que ocurre en otros idiomas, entre otros el español. El mandato

```
\frenchspacing
```

en el preámbulo del documento elimina dicho espacio extra.

Las líneas vacías: Una línea vacía o en blanco, situada entre dos líneas del texto, es señal para el L^AT_EX de que termina un párrafo y comienza otro nuevo, el cual se inicia suprimiendo dicha línea. Varias líneas en blanco consecutivas equivalen a una sola, de modo que no pueden emplearse como recurso para establecer espacios verticales en el texto; más adelante se contempla cómo establecer y graduar espacios a voluntad, tanto en la dirección vertical como en la horizontal.

Las comas: En cuanto a las comas, hay que cuidar de dejar un blanco a continuación de las mismas siempre que sea pertinente, porque el procesador no lo incluye de modo espontáneo.

Otros detalles, también de tipo general, como la escritura de ciertos caracteres especiales, de los guiones, de los puntos suspensivos, etc. se tratan en las próximas secciones.

2.2. Cambios de tipo de letra

El tipo de letra, que por omisión es el llamado Roman, puede modificarse a lo largo del texto, alternando tipos diversos. Los controles son los siguientes:

- `\rm` para tipo Roman:

Este texto está escrito con caracteres tipo Roman.

- `\sl` para tipo Roman inclinado (`slanted`):

Este texto está escrito con caracteres tipo Slanted, cuyo estilo es el anterior con inclinación.

- `\it` para tipo itálica:

Este texto está escrito en itálica, o cursiva. Aunque se asemeja por lo inclinado, puede comprobarse que no es exactamente igual al del ejemplo precedente.

- `\sf` para tipo Sans serif:

Este texto está escrito en estilo Sans Serif.

- `\sc` para tipo capital:

ESTE TEXTO ESTÁ ESCRITO EN ESTILO CAPITAL. COMPÁRESE CON EL RESULTADO DE ESCRIBIR CON MAYÚSCULAS.

- `\tt` para tipo máquina de escribir (`typewriter`):

Este texto está escrito en estilo typewriter, semejante al que produce una máquina de escribir convencional.

- `\bf` para tipo negritas (`boldface`):

Este texto está escrito en negritas

2.2.1. Ambito del cambio

A no ser que se desee que la modificación resulte permanente, hay que determinar el ámbito de validez del tipo de letra al que se cambia, para lo cual se inicia el cambio mediante un paréntesis izquierdo del tipo llave (símbolo `{`) antes del control, y se termina mediante un paréntesis derecho también del tipo llave (símbolo `}`). El ejemplo siguiente muestra cómo se escribe en el texto origen una frase en negritas intercalada en un párrafo escrito con letra corriente:

Este es un ejemplo de un párrafo con tipo de letra estándar, `{\bf con esta frase en negritas}`, de modo que pueda comprobarse el cambio de tipo de letra.

que adoptará al procesarse la apariencia siguiente:

Este es un ejemplo de un párrafo con tipo de letra estándar, **con esta frase en negritas**, de modo que pueda comprobarse el cambio de tipo de letra.

2.2.2. Corrección de itálicas

El \LaTeX dispone de un control para corregir la estrechez de espaciado que se produce al pasar de un tipo de letra con inclinación (como la *Itálica* o la *Slanted* a otro no inclinado, debido a que aquéllos ‘invaden’ el espacio inicial de éste. Se trata del control `\/`. En el párrafo del ejemplo siguiente se puede apreciar la diferencia que hay entre el uso de dicho control y la omisión del mismo:

Este párrafo ilustra el paso de un tipo de letra *itálica* a un tipo estándar que:

- *si no se corrige, resulta mal* la separación en el cambio
- *si se corrige, no resulta mal* la separación en el cambio

La frase última, tipográficamente mejor resuelta, se ha escrito como:

`{\it si se corrige, no resulta mal\/}` la separación

Esta corrección se conoce como *corrección de itálicas* (*italic correction*).

2.2.3. Enfatizado en general

El \LaTeX proporciona también un control para destacar o **enfatizar** palabras en el texto, sin que se haya de tener en cuenta el tipo de letra que se está empleando; esto permite que si más tarde se decide cambiar el tipo de letra del documento, por ejemplo, dichas palabras continúen siendo enfatizadas. Se trata del control

`\em`

(del inglés *emphasized*), que pasa la palabra o conjunto de palabras a itálicas si el texto está escrito con tipo Roman corriente, o a Roman si dicho texto circundante está en itálicas. Al igual que con el cambio de tipo de letra, el control debe precederse del símbolo `{`, y terminarse el grupo de palabras a enfatizar mediante el símbolo de cierre `}`. Así, para enfatizar la frase «sólo tres palabras» emplearemos el control que sigue:

Se lo diré con sólo tres palabras: `{\em todo está perfecto}`. `{\it Si el texto está en itálicas, entonces sólo tres palabras: {\em todo sigue bien}}`.

que producirá el resultado siguiente:

Se lo diré con sólo tres palabras: *todo está perfecto*. *Si el texto está en itálicas, entonces sólo tres palabras: todo sigue bien*.

El control `\emph` produce el mismo efecto que el `\em`, pero se cuida además de la corrección de itálicas, permitiendo por lo tanto despreocuparse de esa particularidad.

2.3. Cambios de tamaño de letra

Los cambios de tipo de letra se hacen sin cambio de tamaño. El tamaño definitivo que se obtiene para la letra depende del tamaño fundamental que se ha elegido en la opción inicial, es decir en la cláusula del tipo

```
\documentclass[a4paper,11pt]{report}
```

Como ya se ha dicho, los tamaños posibles son 10pt, 11pt, 12 pt., y si no se indica ninguno, se adopta 10pt por omisión. En relación con este tamaño elegido inicialmente se ajusta el tamaño al cual se cambia, resultando un tamaño final que viene dado por la tabla ??.

Tamaño de la letra resultante			
Elegido	10pt (omisión)	11pt	12 pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Cuadro 2.1: Cambios de tamaño de letra

A guisa de ejemplos tenemos (el tamaño elegido como fundamental es 11pt):

Esto está escrito en tamaño **`\tiny`**.

Esto está escrito en tamaño **`\scriptsize`**.

Esto está escrito en tamaño **`\footnotesize`**.

Esto está escrito en tamaño **`\small`**.

Esto está escrito en tamaño **`\normalsize`**.

Esto está escrito en tamaño **`\large`**.

Esto está escrito en tamaño **`\Large`**.

Esto está escrito en tamaño **`\LARGE`**.

Esto está en tamaño **`\huge`**.

Esto está en tamaño **`\Huge`**.

2.4. Caracteres especiales

Los caracteres siguientes:

\$ % & _ { }

tienen un significado específico para los procesadores de \TeX , de modo que no se pueden emplear directamente como tales. En su lugar hay que emplear los controles

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

2.5. Entrecorillado

Para que el \LaTeX distinga entre las corillas de apertura y las de cierre, conviene emplear el carácter ‘ para la primera y el ’ para la segunda, como en la frase ‘entrecorillado sencillo’ si se trata de corillas simples, o en “entrecorillado doble” si se emplean dobles corillas. En lugar de la corilla de apertura puede emplearse también el control `\lq` (*left quote*, corilla izquierda), y en lugar de la de cierre el control `\rq` (*right quote*, corilla derecha).

Conviene recordar aquí la distinción entre las *corillas latinas* (« ») y las *corillas inglesas* (“ ”). Para citas, significados, destacar palabras, títulos de libros o de obras, etc., se emplean en principio en español las corillas latinas, a las que se subordinan las inglesas (por ejemplo, dentro de un entrecorillado con corillas latinas). Las corillas simples inglesas (‘ ’) se emplean en español para abarcar significados en lingüística, o para incluir una palabra o en general una expresión dentro de un texto escrito entre corillas inglesas (lo que significa que éstas ya van abrazadas entre corillas latinas).

2.6. Guiones

Los guiones se forman mediante uno o varios guiones simples, como en las frases

Es de origen franco-prusiano. En las páginas 78--96 encontrarán la biografía de Maxwell (1831--1879). Pasen ustedes ---invitó el conserje amablemente---.

que tomarán la apariencia final:

Es de origen franco-prusiano
En las páginas 78–96 encontrarán la biografía de Maxwell (1831–1879).
Pasen ustedes —invitó el conserje amablemente—.

Los tres tipos de guiones anteriores se distinguen en inglés con los nombres de *hyphen*, *en-dash* y *em-dash*, respectivamente.

El guión simple se emplea también, como se verá en su momento, para indicar el símbolo *menos* en las expresiones matemáticas.

2.7. Puntos suspensivos

Los puntos suspensivos escritos sin más en modo texto tienen el inconveniente de que dejan entre sí un espacio muy reducido, con una apariencia defectuosa. Esto se evita acudiendo al mandato `\ldots`, como se aprecia si se escribe

```
No es lo mismo escribir tres puntos ... que usar el mandato
\ldots\ apropiado para los puntos suspensivos.
```

que resulta en la frase

No es lo mismo escribir tres puntos ... que usar el mandato ... apropiado para los puntos suspensivos.

2.8. Acentos y otros caracteres especiales

El L^AT_EX, por su diseño independiente de plataforma, trabaja con la gama baja del código ASCII¹, de modo que las letras acentuadas, al igual que la mayoría de símbolos especiales, deben obtenerse mediante controles. En general, las letras acentuadas se obtienen mediante la secuencia formada por la barra inclinada `\` seguida del carácter indicativo del tipo de acento —como veremos a continuación— y de la letra correspondiente. Así,

- los acentos agudos se obtienen mediante la comilla de cierre, como en `p\'agina`, que produce la palabra **página**.
- los acentos graves se obtienen mediante la comilla de apertura, como en `alg\`ebre`, que produce la palabra **algèbre**.
- los acentos circunflejos se obtienen mediante el mismo símbolo, como ocurre en `Vend\ome`, que produce la palabra **Vendôme**.
- La diéresis se obtiene mediante la doble comilla, como en la secuencia `ambig\"uedad`, que resulta en **ambigüedad**.

¹El código ASCII (pronúnciese «aski») o **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, se diseñó en el tiempo de los primeros ordenadores. Aunque más tarde se amplió a 8 *bits* (256 caracteres), el T_EX se apoya en el código primitivo por razones de portabilidad, dado que los requisitos del idioma inglés se satisfacen a pesar de lo exiguo del código.

- La tilde² (~) sobre una letra se obtiene mediante dicho carácter, tal como en `mu\~neca`, que forma **muñeca**.
- La cedilla se obtiene con una `c`, tal como en `gar\c{c}on`, que da **garçon**.

Hay que hacer una consideración especial cuando se trata de acentuar letras como la *i* o la *j*, porque hay que suprimir el punto antes de acentuar, lo que se consigue mediante el control `\i` o `\j`. Así hay que escribir `\'indice` si se quiere obtener la palabra índice.

mandato	acento	significado
<code>\‘o</code>	ò	acento grave
<code>\’o</code>	ó	acento agudo
<code>\^o</code>	ô	acento circunflejo (sombrero)
<code>\"o</code>	ö	diéresis
<code>\~o</code>	õ	tilde
<code>\=o</code>	ō	barra superior
<code>\.o</code>	ô	acento de punto
<code>\u o</code>	ů	acento breve
<code>\v o</code>	ǔ	háček (checo)
<code>\H o</code>	ő	diéresis húngara
<code>\t oo</code>	öo	acento de enlace posterior
<code>\c o</code>	ç	cedilla
<code>\d o</code>	ð	acento de punto inferior
<code>\b o</code>	ð	barra inferior

mandato	letra	significado
<code>\oe, \OE</code>	œ, Æ	Ligadura francesa
<code>\ae, \AE</code>	æ, Æ	Ligadura latina y escandinava AE
<code>\aa, \AA</code>	å, Å	A con círculo escandinava
<code>\o, \O</code>	ø, Ø	O barrada escandinava
<code>\l, \L</code>	ł, Ł	L suprimida polaca
<code>\ss</code>	ß	«es-zet» o doble s alemana
<code>\i</code>	ı	i sin punto
<code>\j</code>	Ј	j sin punto
<code>!‘</code>	¡	apertura de admiración
<code>?‘</code>	¿	apertura de interrogación
<code>\dag</code>	†	daga u obelisco
<code>\ddag</code>	‡	doble daga
<code>\S</code>	§	signo de número de sección
<code>\P</code>	¶	signo de párrafo
<code>\pounds</code>	£	Libra esterlina
<code>\copyright</code>	©	marca registrada

²En un PC, por combinación de la tecla *AltGr* con la 4/\$

2.9. Ligaduras

2.9.1. Ligadura de palabras

Hay ocasiones en las que se quiere evitar el cambio de línea entre dos palabras dadas, como ocurre por ejemplo en el caso de las palabras **Donald** y **E.** en el nombre Donald E. Knuth. Para ello sirve la tilde, de forma que si se escribe

Donald~E. Knuth

el L^AT_EX considerará la tilde como un espacio en blanco en el cual evitará por otra parte efectuar el cambio de línea. Casos típicos en que conviene recordar el uso de la ligadura son aquellos en que se hace referencia a un capítulo, sección, etc., a una figura, a una tabla, a un teorema o a una fórmula matemática, etc.

Al tratar de las referencias cruzadas, en la sección ??, se volverá sobre este punto.

2.9.2. Evitar el guionizado

Hay veces también en que no se desea que una palabra quede guionizada por cambio de línea. Entonces hay que emplear el control

`\mbox`

Si escribimos

Si no deseamos que una palabra, como la que viene a continuación: Nabucodonosor quede separada en dos debido a un cambio de línea, debemos escribirla utilizando el control `mbox`.

obtenemos el párrafo siguiente en el cual podemos apreciar que la palabra *Nabucodonosor* ha resultado guionizada:

Si no deseamos que una palabra, como la que viene a continuación: Nabucodonosor quede separada en dos debido a un cambio de línea, debemos escribirla utilizando el control `mbox`.

Pero si, siguiendo el consejo anterior, escribimos

Si no deseamos que una palabra, como la que viene a continuación: `\mbox{Nabucodonosor}` quede separada en dos debido a un cambio de línea, debemos escribirla utilizando el control `mbox`.

obtenemos el párrafo siguiente:

Si no deseamos que una palabra, como la que viene a continuación: Nabucodonosor quede separada en dos debido a un cambio de línea, debemos escribirla utilizando el control `mbox`.

en el que puede comprobarse que la palabra *Nabucodonosor* no se ha guionizado, aun a costa de desbordar el límite impuesto al formato del margen de la página.

2.10. Cambio de línea

L^AT_EX ajusta el formato de los párrafos, y por ende decide acerca del cambio de línea. Si, no obstante, se desea forzar dicho cambio, se dispone para ello del control

`\newline`

que también puede abreviarse en la forma

`\\`

Así, el texto

Deseamos que esta línea termine aquí, `\\` de modo que esta otra se escribirá como una línea nueva, y en adelante, a no ser que se intervenga específicamente para impedirlo, el justificado de las líneas se llevará a cabo automáticamente, aunque nos hayamos despreocupado olímpicamente de ello.

quedará en la forma

Deseamos que esta línea termine aquí,
de modo que esta otra se escribirá como una línea nueva, y en adelante, a no ser que se intervenga específicamente para impedirlo, el justificado de las líneas se llevará a cabo automáticamente, aunque nos hayamos despreocupado olímpicamente de ello.

Nota: El control `\newline` de cambio de línea afecta solamente a líneas efectivamente escritas, y no tiene efecto para crear espaciados entre párrafos. La creación de espacio vertical en el texto se controla mediante un mandato distinto (ver sección ??).

2.11. Sangrado de párrafo

El L^AT_EX sangra la primera línea de cada párrafo —a excepción del primero de un capítulo, sección, etc.—, es decir desplaza ligeramente a la derecha su margen inicial, si no se le indica lo contrario. El control

`\noindent`

antes de un párrafo cancela el sangrado de su primera línea.

2.12. Espaciado entre párrafos

Normalmente, el inicio de un párrafo se destaca del anterior mediante el sangrado de su primera línea, pero ésta se separa verticalmente de la última del párrafo anterior por el espacio de interlineado normal. Si se desea conseguir un efecto más pronunciado, haciendo que esa separación sea mayor, puede utilizarse el control `\parskip`, con valores del estilo de los siguientes:

`\parskip=.2ex plus.1ex minus.05ex`

que fijan (relativamente al tamaño de la letra) la separación de la primera línea de un párrafo con respecto a la última del párrafo anterior, y los valores permitidos al \LaTeX para variar esa distancia en más y en menos, con el fin de permitirle cierta flexibilidad al componer la página.

2.13. Espaciados horizontal y vertical

Los controles

`\hspace{nn uu}`

y

`\vspace{nn uu}`

permiten controlar el desplazamiento horizontal y vertical del texto, respectivamente. El parámetro `nn` indica la magnitud del desplazamiento, y `uu` se refiere a la unidad en que se indica, sea ésta milímetros (mm), centímetros (cm), pulgadas (in) u otras unidades de \LaTeX . Así, escribiendo

`\hspace{40 mm}` esta línea desplazada 4 cm.

obtenemos

esta línea desplazada 4 cm.

y escribiendo

`\vspace{1 cm}` esta línea hace de referencia para que podamos comprobar que la siguiente se escribe un centímetro más abajo.

resultará

esta línea hace de referencia para que podamos comprobar que la siguien-

te se escribe un centímetro más abajo.

Los valores del parámetro `nn` que indica la magnitud del desplazamiento pueden ser negativos. Así, si se escribe

```
\hspace{-15 mm}esta línea desplazada centímetro y medio hacia la
izquierda del margen izquierdo.
```

se obtiene

esta línea desplazada centímetro y medio hacia la izquierda del margen izquierdo.

y se aprecia que la línea invade efectivamente la zona a la izquierda de la frontera marcada por el margen izquierdo del texto.

2.14. Comentarios

A veces resulta conveniente insertar en medio del texto anotaciones que por otra parte no se desea que aparezcan en el documento procesado. En tal caso puede utilizarse el símbolo `%`, de forma que lo que se escribe a continuación del mismo es considerado un comentario por el `LATEX` y se ignora en cuanto al proceso tipográfico. Así, si escribimos

```
Los comentarios % Hay que crear una sección
% para tratar del espaciado vertical
no aparecen en el documento resultante...
```

obtendremos

Los comentarios no aparecen en el documento resultante...

El signo de comentario hace que el `LATEX` ignore el resto de la línea, incluido su final, y los espacios en blanco que puedan hallarse al inicio de la línea siguiente. Esto hace útil a dicho símbolo para dividir líneas extensas cuando no se puede o no se desea emplear espacios en blanco. Así, por ejemplo, si se escribe

```
Todo aquello ocu%
    rrió en Babi%
lonia.
```

se leerá

Todo aquello ocurrió en Babilonia.

2.15. Notas a pie de documento

Las notas a pie de documento se incluyen muy fácilmente recurriendo al control

```
\footnote
```

que se encarga de situarlas en la parte inferior de la página, con un tamaño menor de letra³, y separadas del cuerpo del texto principal⁴ mediante una línea horizontal. El ejemplo de la nota anterior se ha obtenido escribiendo

```
...principal\footnote{Ejemplo de nota al pie.} mediante...
```

El \LaTeX numera⁵ automáticamente las notas a pie de documento, y se encarga de pasarlas parcial o completamente a la siguiente página si ocurre que se desborda el espacio destinado a las mismas.

2.16. Entornos del modo texto

De los entornos se ha hablado en el punto 1.4. A continuación se pasa revista a entornos que se aplican en la modalidad texto.

2.16.1. Enfatizar texto (entorno `em`)

El control `\em`, que sirve para enfatizar texto, se ha visto en el punto ?? (página ??). Allí se comentó que era necesario indicar su ámbito de validez, para evitar que el cambio se imponga sobre todo el resto del documento. Cuando el texto a enfatizar es extenso, entonces conviene destacar de modo más claro el inicio y el fin del cambio, por lo que resulta aconsejable emplear el entorno `em`. Como un ejemplo tenemos el siguiente, donde se escribe

```
\begin{em}
```

Este párrafo se escribirá enfatizado, es decir en itálicas si el texto principal estaba en Roman, o en Roman si el texto principal se hallaba escrito en modo itálicas. Si en un momento dado el estilo del texto principal se cambia (de Roman a itálicas o al revés), este párrafo continuará destacándose del resto por el tipo de letra.

```
\end{em}
```

para obtener el resultado que se muestra a continuación:

Este párrafo se escribirá enfatizado, es decir en itálicas si el texto principal estaba en Roman, o en Roman si el texto principal se hallaba escrito en

³El tamaño *footnotesize*, que debe su nombre a esta circunstancia precisamente.

⁴Ejemplo de nota al pie.

⁵Otra nota al pie, obtenida mediante el mismo control

itálicas. Si en un momento dado el estilo del texto principal se cambia (de Roman a itálicas o al revés), este párrafo continuará destacándose del resto por el tipo de letra.

2.16.2. Citas (entornos `quote` y `quotation`)

Este entorno destaca una frase o bien uno o varios párrafos del resto del texto, mediante el correspondiente sangrado. Puede utilizarse para citas y en general siempre que se desee separar parte del texto del resto del mismo. Si escribimos

```
\begin{quote}
\em <<La vida no es más que una sombra pasajera, un pobre cómico
que se pavonea y agita una hora sobre la escena, y luego no se le
oye más: es un cuento narrado por un idiota, lleno de ruido y
aparato, que nada significa.>>

\hspace{6em} {\em William Shakespeare}(<<La tragedia de Macbeth>>)

\end{quote}
```

obtenemos:

*«La vida no es más que una sombra pasajera, un pobre cómico
que se pavonea y agita una hora sobre la escena, y luego no se
le oye más: es un cuento narrado por un idiota, lleno de ruido y
aparato, que nada significa.»*

William Shakespeare(*«La tragedia de Macbeth»*)

Como puede observarse, para el control de enfatizado `\em` al inicio del grupo ‘quote’ no ha sido necesario limitar su ámbito de validez, porque su efecto no se ejerce más allá de los límites de dicho grupo.

El sangrado de márgenes ha corrido a cargo del entorno `quote`. El tipo de letra depende del que elija el usuario.

El entorno `quotation` se utiliza como el `quote`, pero está pensado para el caso en que se emplean largas citas de varios párrafos.

Los entornos `quote` y `quotation` se encargan de disponer el adecuado espacio vertical antes y después del grupo, de forma que también en este caso puede desatenderse dicho aspecto.

2.16.3. Poemas (entorno `verse`)

Está pensado para la inclusión de poemas en el texto. Cada línea debe separarse de la siguiente en la forma habitual, es decir mediante el empleo del control `\newline`, o su forma más corta `\\`. Si escribimos:

```

\begin{verse}
Si rodaran las lunas \\
como suelen rodar, \\
cuando tu luna toque \\
nacerás, para inútil \\
juguete entre dos playas \\
de arena hecha de ceros.

Si arrastrases la herencia \\
de nudos de mi frente, \\
si trajeras del sueño \\
mis cargas de ansiedad, \\
serán desde algún día, \\
en antítesis grave, \\
la vida y la muerte, \\
compañeras absurdas, \\
tus enemigas fieras.

{\em El Autor} (\sf <<Nana anticipada>>)
\end{verse}

```

tendremos como resultado:

Si rodaran las lunas
como suelen rodar,
cuando tu luna toque
nacerás, para inútil
juguete entre dos playas
de arena hecha de ceros.

Si arrastrases la herencia
de nudos de mi frente,
si trajeras del sueño
mis cargas de ansiedad,
serán desde algún día,
en antítesis grave,
la vida y la muerte,
compañeras absurdas,
tus enemigas fieras.

El Autor («Nana anticipada»)

2.16.4. Centrado de texto (entorno center)

Este entorno centra una o varias líneas consecutivas de texto con respecto a los márgenes del mismo. A excepción de la última, cada una de las líneas debe terminar con el control `\newline` (o `\\`).

El entorno **center** se encarga de disponer el adecuado espacio vertical antes y después del grupo. Así, escribiendo

```
\begin{center}
Esta línea se situará en el centro \\
seguida de esta otra, que también quedará centrada
\end{center}
```

resultará

Esta línea se situará en el centro
seguida de esta otra, que también quedará centrada

Hemos de dejar una línea en blanco después del entorno **center**, porque si no se reconocerá como un nuevo párrafo y no se le aplicarán las reglas habituales de sangrado.

2.16.5. Desplazamientos del texto (entornos **flushright** y **flushleft**)

El entorno **flushright** ajusta el último carácter de una línea al margen derecho, sin forzar la línea a adecuarse al ancho normal del texto. Análogamente sucede con el entorno **flushleft**, que alinea al margen izquierdo el primer carácter de la línea, sin alterarla por lo demás.

Así, por ejemplo, al escribir:

```
Podemos conseguir que las líneas no queden ajustadas a ambos
márgenes, sino al que indiquemos,
\begin{flushright}
bien se trate del margen derecho \\
lo que hacemos con el entorno {\tt flushright}
\end{flushright}
\begin{flushleft}
o bien del margen izquierdo \\
para lo cual empleamos el entorno {\tt flushleft}
\end{flushleft}
```

resultará

Podemos conseguir que las líneas no queden ajustadas a ambos márgenes, sino al que indiquemos,

bien se trate del margen derecho
lo que hacemos con el entorno **flushright**

o bien del margen izquierdo
para lo cual empleamos el entorno **flushleft**

2.16.6. Texto literal (entorno `verbatim`)

El entorno `verbatim` (palabra que en inglés equivale a ‘literalmente’, o sea ‘palabra por palabra’), se comporta como el control `\tt`, en el sentido de que el tipo de letra imita al de la máquina de escribir; pero además lo escrito dentro de este entorno se respeta tal como se ha dispuesto, en cuanto a espacio, separaciones de líneas, símbolos especiales, (¡también la invasión del margen derecho!) etc., con la excepción de la secuencia `\end{verbatim}` que da por terminado el grupo. Así, escribiendo

```
\begin{verbatim}
esto está escrito dejando aquí      cinco espacios,
    empezando esta línea algo desplazada a la derecha,
        con esta otra invadiendo el espacio del margen derecho
y escribiendo los caracteres especiales $, %, y \ tal
cual.
\end{verbatim}
```

resultará

```
esto está escrito dejando aquí      cinco espacios,
    empezando esta línea algo desplazada a la derecha,
        con esta otra invadiendo el espacio del margen derecho
y escribiendo los caracteres especiales $, % y \ tal cual.
```

El control `\verb` puede utilizarse como el entorno `verbatim` para frases cortas. No admite argumento, dado que es posible que se quiera emplear texto incluido entre llaves `{}`. Para delimitar su ámbito se recurre a un carácter que no aparezca en el texto literal, tal como `@`, `!`, etc., y se abre y se cierra el ámbito con dicho carácter. Así, la frase

```
\verb@Para obtener el signo $ se emplea \$@
```

dará como resultado la línea

```
Para obtener el signo $ se emplea \$
```

2.16.7. Listas (entornos `itemize`, `enumerate`, `description`)

El \LaTeX gestiona tres tipos de listas mediante los entornos correspondientes: no numeradas (`itemize`), numeradas (`enumerate`) y para descripción de elementos (`description`).

En todos los casos los elementos de la lista se indican mediante el control `\item[etiqueta]`, que inicia un nuevo elemento de la lista y da por terminado el anterior. Este control admite como se ve un argumento, que puede emplearse para etiquetar el encabezamiento de cada elemento; si no se utiliza, el \LaTeX proporciona:

- Para `itemize` un punto grueso, como el de la izquierda
- Para `enumerate`, la correspondiente numeración en secuencia
- Para `description` nada, porque se deja a la elección del usuario

itemize

Veamos varios ejemplos. Escribiendo

```
\begin{itemize}
\item Primer elemento de la lista
\item Segundo elemento de la lista
\item Tercer elemento de la lista
\end{itemize}
```

obtendremos

- Primer elemento de la lista
- Segundo elemento de la lista
- Tercer elemento de la lista

Si deseamos cambiar la etiqueta por un asterisco, por ejemplo, escribi-
ríamos

```
\begin{itemize}
\item[*] Primer elemento de la lista
\item[*] Segundo elemento de la lista
\item[*] Tercer elemento de la lista
\end{itemize}
```

para tener

- * Primer elemento de la lista
- * Segundo elemento de la lista
- * Tercer elemento de la lista

enumerate

Análogamente, mediante

```
\begin{enumerate}
\item Primer elemento de la lista
\item Segundo elemento de la lista
\item Tercer elemento de la lista
\end{enumerate}
```

resultará la lista numerada

1. Primer elemento de la lista
2. Segundo elemento de la lista
3. Tercer elemento de la lista

description

y finalmente, con

```
\begin{description}
\item Primer elemento de la lista
\item Segundo elemento de la lista
\item Tercer elemento de la lista
\end{description}
```

se obtiene simplemente

Primer elemento de la lista

Segundo elemento de la lista

Tercer elemento de la lista

En este último caso de lista descriptiva, puede utilizarse el argumento para resaltar el elemento a describir, como por ejemplo en

```
\begin{description}
\item[Primer] elemento de la lista
\item[Segundo] elemento de la lista
\item[Tercer] elemento de la lista
\end{description}
```

que producirá

Primer elemento de la lista

Segundo elemento de la lista

Tercer elemento de la lista

El tipo de letra para la etiqueta puede elegirse a voluntad, como en el caso en que se escribe

```

\begin{description}
\item[\it Primer] elemento de la lista
\item[\it Segundo] elemento de la lista
\item[\it Tercer] elemento de la lista
\end{description}

```

que produce como resultado:

Primer elemento de la lista

Segundo elemento de la lista

Tercer elemento de la lista

Como se observa, el ámbito del tipo de letra utilizado para la etiqueta queda restringido a la misma, por lo que no hace falta establecerlo mediante llaves.

Listas anidadas

Como ocurre en general con los entornos, las listas pueden anidarse. Así

```

\begin{enumerate}
\item Primer elemento de la lista
\item Segundo elemento de la lista
  \begin{itemize}
    \item primer ítem del segundo elemento
    \item segundo ítem del segundo elemento
    \item ....
  \end{itemize}
\item Tercer elemento de la lista
\end{enumerate}

```

nos dará como resultado:

1. Primer elemento de la lista
2. Segundo elemento de la lista
 - primer ítem del segundo elemento
 - segundo ítem del segundo elemento
 -
3. Tercer elemento de la lista

2.16.8. Tabulado (entorno `tabbing`)

Este entorno simula el tabulado de las máquinas de escribir clásicas, función que permite escribir en columnas. Con una máquina de escribir, antes de establecer los límites del tabulado se estudian las palabras más largas de cada columna, y a continuación se fijan los tabuladores de acuerdo con las correspondientes dimensiones. Con el \LaTeX se procede del mismo modo:

- En la línea en que se fijan los tabuladores se escriben las palabras que necesitan más espacio correspondientes a cada columna, y se separan mediante el control `\=`, que establece los tabuladores. La línea se termina casi siempre con el control `\kill`, para indicar que no debe aparecer en el escrito, a no ser que por casualidad esa primera línea establezca precisamente todos los tabuladores.
- A continuación se escribe cada línea separando los elementos (palabras o frases) de cada columna mediante el control `\>`, que hace el papel de la tecla de tabulación.

Si, por ejemplo, se escribe

```
\begin{tabbing}
Hematites \= Amarillo \= Densidad \kill
\bf Mineral \> \bf Color \> \bf Densidad \\\
Oro \> Amarillo \> 19\\
Plata \> Blanco \> 11\\
Galena \> Gris \> 7.5\\
Hematites \> Pardo \> 5.3
\end{tabbing}
```

obtendremos

Mineral	Color	Densidad
Oro	Amarillo	19
Plata	Blanco	11
Galena	Gris	7.5
Hematites	Pardo	5.3

Obsérvese que se ha utilizado el control de cambio de línea después de cada una de ellas, a excepción de la última. Naturalmente, si se desea una mayor amplitud para las columnas, nada impide fijar los tabuladores mediante palabras más largas, ya que el control `\kill` elimina la línea de declaración. De este modo, escribiendo

```

\begin{tabbing}
xxxxxxxxxxxxx \= xxxxxxxxxxxx \= Densidad \kill
\bf Mineral \> \bf Color \> \bf Densidad \\\
Oro \> Amarillo \> 19\\
Plata \> Blanco \> 11\\
Galena \> Gris \> 7.5\\
Hematites \> Pardo \> 5.3
\end{tabbing}

```

se obtienen columnas con mayor separación entre sí:

Mineral	Color	Densidad
Oro	Amarillo	19
Plata	Blanco	11
Galena	Gris	7.5
Hematites	Pardo	5.3

Dejando en blanco espacios entre tabulados pueden conseguirse algunos efectos, como las separaciones que se obtienen al escribir

```

\begin{tabbing}
Tiranosaurioxxx \= Cretácicoxxxx \= Añosx \= (millones) \kill
\bf Nombre \> \bf Periodo \> \bf Años \> \bf (millones)\\
Ictiosaurio \> Jurásico \> \>135\\
Diplodocus \> Jurásico \> \>145\\
\> \> \> \\\
Tricerátopo \> Cretácico \> \>70\\
Tiranosaurio \> Cretácico \> \> 70
\end{tabbing}

```

y que aparecen como sigue

Nombre	Periodo	Años (millones)
Ictiosaurio	Jurásico	135
Diplodocus	Jurásico	145
Tricerátopo	Cretácico	70
Tiranosaurio	Cretácico	70

Los tabuladores pueden redefinirse dinámicamente, es decir luego de haber escrito una serie de líneas, y ello tantas veces como se desee.

2.16.9. Tablas (entorno tabular)

Este entorno admite argumentos, cuya función es la del ajuste de las entradas de cada columna de la tabla, que en relación con la casilla correspondiente podrán situarse:

- a la izquierda, con el argumento **l** (de left, izquierda)
- en el centro, con el argumento **c** (de c, center)
- a la derecha, con el argumento **r** (de right, derecha)

Para separar en una línea los elementos que pertenecen a cada columna se emplea el símbolo de separación **&**. Por otra parte, cada línea a excepción de la última debe terminarse con el control de cambio de línea ****.

Como ejemplo escribimos

```
\begin{tabular}{llr}
\bf Mineral & \bf Color & \bf Densidad \\
Oro & Amarillo & 19\\
Plata & Blanco & 11\\
Galena & Gris & 7.5\\
Hematites & Pardo & 5.3
\end{tabular}
```

para obtener

Mineral	Color	Densidad
Oro	Amarillo	19
Plata	Blanco	11
Galena	Gris	7.5
Hematites	Pardo	5.3

Como vemos, no hace falta terminar la última fila con el control de separación de línea. Si se desean trazos verticales en la tabla, se indicará el carácter de trazo vertical **|** en la posición correspondiente de los argumentos. Los trazos horizontales, por otra parte, se consiguen mediante el control **\hline** (**h**orizontal **l**ine, línea horizontal) colocado en cada una de las líneas para las que quiera obtenerse dicho efecto. La tabla anterior se escribe así como sigue:

```
\begin{tabular}{|l|l|r|}
\hline
\bf Mineral & \bf Color & \bf Densidad \\
\hline
Oro & Amarillo & 19\\
\hline
Plata & Blanco & 11\\
\hline
```



```

\hline
Galena & Gris & 7.5\\
\hline
Hematites & Pardo & 5.3\\
\hline
\end{tabular}

```

lo cual produce

Mineral	Color	Densidad
Oro	Amarillo	19
Plata	Blanco	11
Galena	Gris	7.5
Hematites	Pardo	5.3

Obsérvese que ahora sí hemos tenido que rematar la última línea con el control de cambio de línea, para que actúe después de ella el control `\hline`.

Si se quieren establecer celdas que ocupen el espacio de varias de ellas en la tabla, se debe emplear el control

```
\multicolumn {n}{ajuste}{elemento}
```

donde

`n` indica el número de columnas del formato origen que han de cubrirse.

`ajuste` es el parámetro de alineamiento del elemento dentro de la celda, con valores `l` (izquierda), `c` (centrado), o `r` (derecha).

`elemento` es la palabra o frase a introducir en la casilla.

Así, mediante

```

\begin{tabular}{|l|l|l|r|} \hline
\multicolumn{3}{|c|}{Las primeras lunas de Júpiter}\\
\hline \hline \bf Nombre & \bf Descubridor & \bf Año \\ \hline Io
& Galileo & 1610\\ \hline Europa & Galileo & 1610\\ \hline
Ganímedes & Galileo & 1610\\ \hline Calixto & Galileo & 1610 \\
\hline Amaltea & E. Barnard & 1892 \\ \hline
\end{tabular}

```

se obtiene la tabla

Las primeras lunas de Júpiter		
Nombre	Descubridor	Año
Io	Galileo	1610
Europa	Galileo	1610
Ganímedes	Galileo	1610
Calixto	Galileo	1610
Amaltea	E. Barnard	1892

en la que cabe observar que se ha obtenido una separación de la cabecera, mediante un segundo control de línea horizontal, y además un doble trazo lateral, sin más que indicarlo en los argumentos.

2.16.10. Figuras y tablas (entornos `figure` y `table`)

Las tablas y las figuras proporcionan un marco para tablas propiamente dichas (entorno `tabular`), y para gráficos, imágenes, etc. (ver capítulo ??). Las tablas y las figuras ocupan siempre lugar en una página, sin cabalgar entre dos de ellas. Esto quiere decir que el propio L^AT_EX determina la ubicación en que emplazarlas, sin garantizar que dicho sitio corresponda al punto del texto original en que se las ha declarado. Los entornos `figure` y `table`, sin embargo, permiten cierto dominio sobre dicho emplazamiento, aparte de disponer formato para otras características de tales objetos.

El parámetro de estos entornos, que es optativo, consiste en una combinación (no exhaustiva) de las letras **h**, **t**, **b** y **p**, cuya función es la siguiente:

h (de *here*, aquí), indica que el objeto (tabla o figura) debe ser colocada donde se declara, si hay sitio disponible.

t (de *top*, parte superior), solicita que el objeto se coloque en la parte superior de la página, aunque nunca antes de la página actual.

b (de *bottom*, parte inferior), solicita que el objeto se sitúe en la parte inferior de la página, aunque no antes de la actual.

p (de *page*, página), indica que el objeto se debe colocar en una página separada (*page of floats* o «página de flotación»), dedicada exclusivamente a tablas y figuras.

Si se combinan varios de los anteriores valores de los parámetros, el orden de preferencia de los mismos es su orden de escritura. El valor por omisión del parámetro es **tbp** (para las clases de documento *article* y *report*), y una buena elección explícita pudiera ser **htbp**, en el caso en que se desee que, de ser posible, la figura o tabla se sitúen lo más cerca al punto en que se les declara en el documento.

Los entornos `figure` y `table` proporcionan el formato para numerar y para titular el objeto correspondiente, sea éste una tabla o una figura. Para ello tenemos el control

```
\caption{título del objeto}.
```

Por otra parte, el objeto se puede etiquetar mediante el control

```
\label{etiqueta del objeto},
```

que identificará la tabla o figura, lo cual permitirá referirse a la misma desde cualquier otra parte del texto (ver la sección ?? en la página ??).

Así, por ejemplo, al escribir

```
\begin{table}[htbp]
\centering
\begin{tabular}{|l|l|l|r|} \hline
\multicolumn{3}{|c|}{ Las primeras lunas de Júpiter} \\ \hline \bf
Nombre & \bf Descubridor & \bf Año \\ \hline Io & Galileo & 1610 \\
\hline Europa & Galileo & 1610 \\ \hline Ganímedes & Galileo &
1610 \\ \hline Calixto & Galileo & 1610 \\ \hline Amaltea & E.
Barnard & 1892 \\ \hline
\end{tabular} \\
\caption{\small Ejemplo de tabla.} \label{tabJupiter}
\end{table}
```

obtenemos como resultado la tabla ?? que, como se puede ver, se ha numerado y titulado. Por otra parte, la etiqueta *tabJupiter* permitirá referirse a ella en cualquier punto del texto. Se ha empleado el control `\centering` para colocar la tabla centrada en relación con el texto, pues de otro modo hubiese quedado alineada al margen izquierdo.

Las primeras lunas de Júpiter		
Nombre	Descubridor	Año
Io	Galileo	1610
Europa	Galileo	1610
Ganímedes	Galileo	1610
Calixto	Galileo	1610
Amaltea	E. Barnard	1892

Cuadro 2.2: Ejemplo de tabla

2.17. Notas al margen

Nota escrita al margen, con el fin de servir de ejemplo.

Es posible insertar notas al margen mediante el control `\marginpar{texto de la nota}`, donde *texto de la nota* se sitúa:

- en el margen derecho si se escribe en modo *un lado* (oneside)
- en el margen externo si se escribe en modo *dos lados* (twoside)
- en el margen más próximo para el modo *dos columnas* (twocolumn).

La primera línea se sitúa a la altura de la línea de texto en la que se ha insertado el control `\marginpar`. El mandato `\reversemarginpar` obliga a la nota al margen a colocarse en el margen opuesto (interior). La primera palabra no se suele guionizar. Se puede permitir el guionizado de la misma situando delante de ella el control `\hspace{0pt}`.

Capítulo 3

Modo matemático

Debajo de las multiplicaciones
hay una gota de sangre de pato;
debajo de las divisiones
hay una gota de sangre de marinero;
debajo de las sumas, un río de sangre tierna.

Federico García Lorca
«Poeta en Nueva York»

Las escritura de expresiones matemáticas puede realizarse según dos variantes, que corresponde a su vez a dos nuevos entornos. En la que llamaremos «*en línea de texto*» (entorno `math`), las expresiones se escriben como parte del texto ordinario de un párrafo; en la variante de «*presentación matemática*» (entorno `displaymath`), las expresiones o fórmulas se destacan claramente del texto, con el fin de disponer del espacio necesario para su tipografía. A continuación se explican ambas modalidades.

3.1. Modalidad «*en línea de texto*» (entorno `math`)

El símbolo `$` indica al `LATEX` que se cambia al modo matemático, y es el mismo que se utiliza para pasar nuevamente al modo texto ordinario. En su lugar pueden emplearse también los controles `\(`, para el inicio del modo matemático, y `\)`, para el retorno al modo texto; este último resulta más claro en cuanto a que se distingue entre la apertura y el cierre de la modalidad de escritura. Si se desea mayor claridad en las declaraciones, pueden utilizarse los controles más explícitos del entorno `math`, `\begin{math}` y `\end{math}`.

Si escribimos

Para las expresiones matemáticas, tales como $z=3xy-17x$, conviene establecer el modo matemático, tal como se hace a continuación: `$z=3xy-17x$`, que como vemos produce

una expresión más apropiada al estilo matemático.

obtendremos el párrafo

Para las expresiones matemáticas, tales como $z=3xy-17x$, conviene establecer el modo matemático, tal como se hace a continuación: $z = 3xy - 17x$, que como vemos produce una expresión más apropiada al estilo matemático.

Y escribiendo

En general, cuando intervienen símbolos o expresiones genuinamente matemáticas, como en $\forall x \in A$, $\exists y \in B$ tal que $x \rightarrow y$, el modo matemático se hace obviamente imprescindible.

resultará lo siguiente:

En general, cuando intervienen símbolos o expresiones genuinamente matemáticas, como en $\forall x \in \mathcal{A}, \exists y \in \mathcal{B}$ tal que $x \xrightarrow{f} y$, el modo matemático se hace obviamente imprescindible.

3.2. Modalidad «*presentación matemática*» (entorno `displaymath`)

En el modo matemático antes referido el procesador de \LaTeX trata de ir escribiendo las expresiones de modo que formen líneas de un párrafo, aunque procura en la medida de lo posible no guionizar las palabras que intervienen. De todos modos, en la escritura de fórmulas matemáticas es frecuente que se desee controlar el ámbito de las expresiones, así como disponer de una mayor libertad de espacio para la composición. Para ello recurrimos al modo “presentación matemática”, que se inicia con la secuencia `$$`, y se cierra con la misma `$$`. Más clara resulta la secuencia `\[` para la apertura y `\]` para el cierre de esta modalidad. Y aún más explícitos resultan los controles del entorno `displaymath`: `\begin{displaymath}` y `\end{displaymath}`, para definir el inicio y el fin de este entorno.

En el caso en que se desee que el \LaTeX se encargue de numerar las fórmulas, se utilizarán los controles del entorno `equation`: `\begin{equation}` y `\end{equation}`.

Si escribimos

Para resolver el problema, será necesario hallar las soluciones de $f(x)=0$ descartando en todo caso las que posean valores imaginarios.

resultará:

Para resolver el problema, será necesario hallar las soluciones de

$$f(x) = 0$$

descartando en todo caso las que posean valores imaginarios.

De haber escrito

Para resolver el problema, será necesario hallar las soluciones de `\begin{equation}f(x)=0\end{equation}` descartando en todo caso las que posean valores imaginarios.

obtendríamos la ecuación numerada automáticamente:

Para resolver el problema, será necesario hallar las soluciones de

$$f(x) = 0 \tag{3.1}$$

descartando en todo caso las que posean valores imaginarios.

La numeración de las ecuaciones se establece, como se ve, con referencia al capítulo.

3.3. Símbolos matemáticos

A continuación se pasa revista al conjunto de símbolos que forman parte de las expresiones matemáticas. Son los constituyentes elementales de tales expresiones.

3.3.1. Símbolos inmediatos

Directamente del teclado podemos escribir los símbolos:

`+ - = < > | / () [] *`

Aparte tenemos, claro está, las letras del alfabeto inglés (a-z y A-Z), y las cifras numéricas del 0 al 9.

3.3.2. Símbolos generales

La tabla ?? muestra una variedad de símbolos de propósito general.

3.3.3. Letras de tipo caligráfico

Para nombrar objetos matemáticos con letras mayúsculas que destaquen caligráficamente, disponemos del control `\cal`, que también puede escribirse `\mathcal`. Funciona como un control de cambio de estilo, por lo que debemos delimitar el ámbito de aplicación del mismo. Al escribir

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>		
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\triangle	<code>\triangle</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>
\wp	<code>\wp</code>	\perp	<code>\bot</code>	\sharp	<code>\sharp</code>	\diamond	<code>\diamondsuit</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>

Cuadro 3.1: Símbolos generales

.. supongamos un conjunto \mathcal{Z} en el cual...

obtenemos

.. supongamos un conjunto \mathcal{Z} en el cual...

Se aplica a las letras mayúsculas del alfabeto inglés, y el aspecto del juego completo es el siguiente:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

3.3.4. Letras griegas

El alfabeto griego se obtiene mediante los controles que se indican en la tabla ???. Para obtener el alfabeto griego en mayúsculas basta con escribir en mayúsculas la inicial del nombre de la letra, tal como muestra la tabla ???; para las letras que faltan se empleará el tipo *Roman* correspondiente. Se puede ver que las letras griegas minúsculas se italicizan, mientras que las mayúsculas se escriben verticales, aunque pueden italicizarse mediante el control `\it`.

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>
π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>
ω	<code>\omega</code>						

Cuadro 3.2: Letras griegas minúsculas

3.3.5. Operadores binarios

Aparte de los operadores $+$, $-$, y $*$, accesibles directamente por medio del teclado, la tabla ??? muestra los controles mediante los que se obtienen

Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>
Ξ	<code>\Xi</code>	Π	<code>\Pi</code>	Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>
Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>		

Cuadro 3.3: Letras griegas mayúsculas

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangleup	<code>\bigtriangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	∇	<code>\bigtriangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\bigcirc	<code>\bigcirc</code>
\dagger	<code>\dagger</code>	\setminus	<code>\setminus</code>	\amalg	<code>\amalg</code>	\circ	<code>\circ</code>
\ddagger	<code>\ddagger</code>	\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\bullet	<code>\bullet</code>

Cuadro 3.4: Operadores binarios

otros operadores de tipo binario, es decir los que implican dos argumentos. Si por ejemplo se escribe

`$$\cal A-(B\cap C)=(A-B)\cup (A-C)$$`

tendremos

$$\mathcal{A} - (\mathcal{B} \cap \mathcal{C}) = (\mathcal{A} - \mathcal{B}) \cup (\mathcal{A} - \mathcal{C})$$

3.3.6. Operadores de relación

En este caso también algunos símbolos se obtienen directamente del teclado: $<$, $>$, $=$ y $|$. La tabla ?? muestra un conjunto de operadores de relación binaria y sus controles.

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	$ $	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\Join	<code>\Join</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\neq	<code>\neq</code>	\in	<code>\in</code>	\notin	<code>\notin</code>	\ni	<code>\ni</code>
\propto	<code>\propto</code>	\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>		

Cuadro 3.5: Operadores de relación

Escribiendo

Así pues, si `$$\forall x\in\cal A$ se tiene $x\in\cal B$ $$`, se escribirá `$$\cal A\subset\cal B$`.

resulta

Así pues, si $\forall x \in \mathcal{A}$ se tiene $x \in \mathcal{B}$, se escribirá $\mathcal{A} \subset \mathcal{B}$.

3.3.7. Símbolos de flechas

La tabla ?? deja ver los símbolos de flechas que se pueden producir mediante las correspondientes palabras de control.

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</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>
\lhookrightarrow	<code>\lhookrightarrow</code>	\rhookrightarrow	<code>\rhookrightarrow</code>	\swarrow	<code>\swarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\nwarrow	<code>\nwarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>		
\rightleftharpoons	<code>\rightleftharpoons</code>				

Cuadro 3.6: Símbolos de flechas

Si se escribe

```

$$
A = B \text{ \mbox{ si se cumple } \forall x \in A}
\Longleftrightarrow \text{ , } x \in B
$$

```

se obtiene

$$A = B \text{ si se cumple } \forall x \in A \iff x \in B$$

3.3.8. Acentos

La tabla ?? muestra los controles utilizados para acentuar cuando se escribe en modo matemático. El ejemplo está referido a la letra *a*, pero se aplica a cualquier otra. En el caso particular de las letras *i* y *j*, la eliminación de sus puntos antes de acentuar se hace escribiéndolas como `\imath` y `\jmath`.

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

Cuadro 3.7: Acentos en modo matemático

3.3.9. Símbolos de tamaño variable

Hay símbolos de operadores que deben escalarse de distinto modo si se escriben en modo *línea de texto* (entorno `math`), donde es necesario que queden comprimidos en sentido vertical, o en modo *presentación matemática* (entorno `displaymath`), en cuyo caso disponen de espacio suficiente para indicar sus límites, etc.

La tabla ?? muestra varios de estos operadores de tamaño variable según del modo de escritura (*en línea de texto* o *presentación matemática*).

Σ	\sum	\bigcap	\bigcap	\bigodot	\bigodot
\prod	\prod	\bigcup	\bigcup	\bigotimes	\bigotimes
\coprod	\coprod	\bigsqcup	\bigsqcup	\bigoplus	\bigoplus
\int	\int	\bigvee	\bigvee	\biguplus	\biguplus
\oint	\oint	\bigwedge	\bigwedge		

Cuadro 3.8: Símbolos de tamaño variable

El texto que viene a continuación deja ver la distinta forma que adquiere la escritura de dichos operadores según el entorno matemático empleado:

En la modalidad de *línea de texto* los operadores de tamaño variable deben estrecharse verticalmente para encajar mejor en los espacios entre líneas, y los límites se situarán también de forma que ocupen el menor espacio vertical posible, tal como cuando se escribe $\sum_{i=1}^N$ o \int_0^∞ . Distinto es el caso en que tales símbolos se escriben en la modalidad de *presentación matemática*, donde disponen de la suficiente libertad de espacio, tal como ocurre en

$$\sum_{i=1}^N \text{ o también en } \int_0^\infty$$

En todos los casos, las expresiones se han escrito del mismo modo, es decir

`\sum_{i=1}^N` y `\int_0^\infty`

Aun así, el \LaTeX permite forzar el espacio para escalar dichos símbolos a su tamaño tipo *presentación matemática* aun cuando estemos escribiendo en modo *línea de texto*, como ocurre en el caso en que escribimos $\sum_{i=1}^N$ o \int_0^∞ , lo cual se consigue mediante el control `\displaystyle` antepuesto al símbolo correspondiente.

La asignación de límites a los operadores se considera en la sección ??.

3.4. Índices

Para afectar una expresión con subíndices emplearemos el control `_` (guión inferior), y para hacerlo con índices superiores (o superíndices) el control `^` (sombrero).

El subíndice o el índice superior se aplica al grupo que lo precede. Esto quiere decir que si escribimos x^n mediante la secuencia `$\text{\$x^n\$}$` , el exponente n se aplica obviamente a la letra x , y al escribir $(a+b)^n$ (obtenida mediante la secuencia `$\text{\$(a+b)^n\$}$` , el exponente n afecta solamente al último elemento de la expresión $(a+b)$, es decir al símbolo del paréntesis derecho ')'. Para que el exponente se aplique a toda la expresión, hay que establecerla como un grupo, tal como ocurre si se escribe la secuencia `$\text{\$\{(a+b)\}^n\$}$` , que produce como resultado la expresión matemática $(a+b)^n$, en la cual se aprecia que el exponente se coloca algo más elevado que en el caso anterior. Es cuestión de gustos emplear un sistema o el otro, dado que hay quien prefiere, por razones de legibilidad, que los exponentes o en general los índices superiores se hallen todos al mismo nivel de elevación.

Cuando el índice o el subíndice están formados por expresiones, deben agruparse entre llaves para delimitarlos. Así, la expresión

$$\Pi_{\alpha\beta}^{\gamma\delta}$$

se escribirá mediante la secuencia

```
\Large
\$\Pi_{\alpha\beta}^{\gamma\delta}\$
\normalsize
```

También es posible controlar la situación de los índices mediante los oportunos grupos de llaves vacíos, como en la expresión

```
\Large
\$\cal{A}_{\alpha\beta}^{\gamma}_{\delta}\$
\normalsize
```

que se presentará en la forma

$$\mathcal{A}_{\alpha\beta}^{\gamma}{}_{\delta}$$

Los índices pueden preceder las expresiones, como en

$${}_i^j\mathcal{P}_\alpha^\beta$$

escrito mediante la expresión

```
\Large
$$_i^j\cal{P}_\alpha^\beta$$
\normalsize
```

En las tres últimas expresiones se ha cambiado el tamaño de letra para facilitar la lectura de las mismas.

3.5. Primas

Las comillas o *primas* que se asocian a ciertas letras (para indicar las derivadas de una función, por ejemplo), se obtienen simplemente mediante la comilla de cierre ', siempre naturalmente que se esté en entorno matemático. El control `\prime` produce el mismo símbolo, pero hay que tener en cuenta el situarlo como índice superior; según esto, es lo mismo escribir

```
$$
f' , f'', f''', ...
$$
```

que escribir

```
$$
f^{\prime} , f^{\prime\prime}, f^{\prime\prime\prime}, ...
$$
```

para obtener

$$f', f'', f''', \dots$$

3.6. Fracciones

En el caso de expresiones sencillas puede utilizarse, siempre en modo matemático, el símbolo de la barra inclinada /, como por ejemplo en la expresión $1/\pi$, que se escribe simplemente como `$1/\pi$`. Pero en general las fracciones se generan mediante el control

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

donde los argumentos `numerador` y `denominador` son las expresiones que corresponden a dichos términos, y que pueden adoptar toda la generalidad que se desee, conteniendo fracciones a su vez, en particular. Así, la secuencia

```
$$
\frac{\int_0^{\frac{\pi}{2}} \{x^2 \, dx\}}{x^3}
$$
```

resulta en la expresión

$$\frac{\int_0^{\frac{\pi}{2}} x^2 dx}{x^3}$$

y análogamente mediante $\frac{\partial y}{\partial x}$ se tiene el símbolo de la derivada parcial

$$\frac{\partial y}{\partial x}$$

3.7. Raíces

Las raíces se escriben mediante el control

`\sqrt[n]{expresión}`

en el que

`n` es el índice de la raíz. En el caso de las raíces cuadradas no es necesario indicarlo.

`expresión` es la expresión bajo la raíz.

Valga como ejemplo la secuencia

`\displaystyle \sqrt{(x+y)^m} \ \mbox{ y } \ \sqrt[3]{x^2+y^2}`

que produce las expresiones $\sqrt{(x+y)^m}$ y $\sqrt[3]{x^2+y^2}$

respectivamente.

3.8. Puntos suspensivos

Los puntos suspensivos que se emplean en medio o al final de sumatorios y expresiones semejantes se escriben con los controles `\ldots` (puntos ajustados a la izquierda, nivel bajo) y `\cdots` (puntos centrados, nivel medio). Así, las secuencias

`$$`
`x+x^2+x^3+\cdots+x^n`
`$$`

y

`$$`
`(a_1,\ldots,a_n)`
`$$`

dan como resultado las expresiones

$$x + x^2 + x^3 + \cdots + x^n$$

y

$$(a_1, \dots, a_n)$$

respectivamente.

3.9. Texto en expresiones matemáticas

Para insertar texto corriente en una expresión matemática se utiliza el control

`\mbox`

Los espacios iniciales o finales del texto deben incluirse en el mismo, dado que el modo matemático los ignora. También pueden utilizarse los controles de espaciado (ver sección ?? para ajustar espacios en modo matemático.

Según lo dicho, para obtener

$$S_N = \sum_1^N x_i \text{ donde } N \text{ es menor que } 100$$

escribiremos

`$$`

`S_N = \sum_1^N x_i \mbox{ donde } N \text{ es menor que } 100\}`

`$$`

3.10. Nombres en expresiones matemáticas

Existen nombres de funciones y símbolos de operaciones que se suelen escribir en forma de texto normal (tipo Roman) en las expresiones matemáticas, tal como ocurre con las abreviaturas empleadas para las funciones seno, logaritmo, etc. Si dichos nombres se escriben sin más en modo matemático, resultarían italicizados. Por otra parte, el recurso al control `\mbox` (sección ??) obligaría a corregir defectos de espaciado. El \LaTeX proporciona un conjunto de palabras de control para los nombres de funciones del tipo a que nos referimos, controles que se presentan en la tabla

Como ejemplo tenemos:

$$f(x) = \cos x + \log(x^3)$$

obtenida escribiendo

<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>

Cuadro 3.9: Nombres de funciones

`$$f(x)=\cos x + \log(x^3)$$`

Si se escribe en un idioma distinto del inglés, pueden existir funciones que se designen con una palabra distinta que la prevista en la tabla. Esto ocurre en español por ejemplo con la que designa la función seno (sin en inglés). Para obtener la correspondiente abreviatura en español (sen), deberemos utilizar el control `\mbox`, escribiendo así `$$\mbox{sen}\backslash,x$` para obtener $\text{sen } x$, donde el control `\`, se ha utilizado para insertar algo de espacio (ver sección ??) entre la palabra sen y la x .

3.11. Delimitadores de expresiones

Son muchas las ocasiones en que una expresión matemática se escribe encerrada entre llaves, corchetes u otros símbolos delimitadores, que a veces la abrazan solamente por la izquierda o por la derecha. El L^AT_EX permite escalar tales símbolos al tamaño conveniente mediante los controles `\left` y `\right`, que se anteponen al símbolo delimitador, el primero para delimitar por la izquierda y el segundo cuando se delimita por la derecha de la expresión.

La tabla ?? ofrece un repertorio de símbolos delimitadores. Obsérvese que el símbolo de llave, tanto el de apertura `{` como el de cierre `}`, se debe escribir precedido de la barra inclinada `\`.

<code>(</code>	<code>(</code>	<code>)</code>	<code>)</code>	\uparrow	<code>\uparrow</code>
<code>[</code>	<code>[</code>	<code>]</code>	<code>]</code>	\downarrow	<code>\downarrow</code>
<code>{</code>	<code>\{</code>	<code>}</code>	<code>\}</code>	\updownarrow	<code>\updownarrow</code>
<code>\lfloor</code>	<code>\lfloor</code>	<code>\rfloor</code>	<code>\rfloor</code>	\Uparrow	<code>\Uparrow</code>
<code>\lceil</code>	<code>\lceil</code>	<code>\rceil</code>	<code>\rceil</code>	\Downarrow	<code>\Downarrow</code>
<code>\langle</code>	<code>\langle</code>	<code>\rangle</code>	<code>\rangle</code>	\Updownarrow	<code>\Updownarrow</code>
<code>/</code>	<code>/</code>	<code>\</code>	<code>\backslash</code>		
<code> </code>	<code> </code>	<code>\ </code>	<code>\ </code>		

Cuadro 3.10: Delimitadores

Cuando se emplea solamente un símbolo, el de apertura o el de cierre, debe indicarse igualmente el «*delimitador vacío*» en el otro extremo de la expresión: se escribirá `\left.` para la parte izquierda si ésta queda sin delimitador, y `\right.` para la derecha en su caso.

Así, para obtener la expresión

$$|x| = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$$

escribiremos

```

$$
\left|x\right| =
\left\{
\begin{array}{rl}
+ x & \text{\mbox{si } } x \geq 0 \\
- x & \text{\mbox{si } } x < 0
\end{array}
\right.

```

y para obtener esta otra

$$\left. \begin{array}{l} a_1^1 x^1 + a_1^2 x^2 + \cdots + a_1^n x^n = c_1 \\ a_2^1 x^1 + a_2^2 x^2 + \cdots + a_2^n x^n = c_2 \\ \quad \quad \quad \cdots \\ a_n^1 x^1 + a_n^2 x^2 + \cdots + a_n^n x^n = c_n \end{array} \right\}$$

se escribirá

```

$$
\left.
\begin{array}{c}
a_1^1 x^1 + a_1^2 x^2 + \cdots + a_1^n x^n = c_1 \\
a_2^1 x^1 + a_2^2 x^2 + \cdots + a_2^n x^n = c_2 \\
\cdots \\
a_n^1 x^1 + a_n^2 x^2 + \cdots + a_n^n x^n = c_n
\end{array}
\right\}

```

3.12. Trazos superiores e inferiores

Para trazar una raya en la parte inferior o superior de una expresión se dispone de los controles

`\underline`

y

`\overline`

respectivamente. Ambos admiten una expresión como argumento, pero hay que tener en cuenta que no se establecen líneas de párrafo automáticamente, lo que significa que en el caso de que se desee subrayar (o superrayar) una frase habrá que cuidar de la separación de líneas.

Así, para subrayar esta frase hemos escrito

`\$\underline{\mbox{para subrayar esta frase}}\$`.

y para producir la expresión $\overline{a + bi} = a - bi$

`$$\overline{a+bi}=a-bi$`

Los controles anteriores se pueden repetir, produciendo en tal caso rayas múltiples, como en el ejemplo $\overline{\overline{a + bi}} = a + bi$, para obtener el cual se ha escrito

`$$\overline{\overline{a+bi}}=a+bi$$`.

3.13. Llaves superiores e inferiores

Los controles que trazan llaves horizontales en la parte inferior o superior de una expresión son

`\underbrace`

y

`\overbrace`

respectivamente, donde la expresión es el argumento. Es posible etiquetar las llaves mediante los controles de subíndice (`_`) para el caso de la llave inferior o de superíndice (`^`) para el de la llave superior, como en los ejemplos

$$S_n = \underbrace{a + a + a + \dots + a}_{n \text{ términos}}$$

y

$$\Pi_n = \overbrace{a \times a \times \dots \times a}^{n \text{ términos}}$$

para los cuales se ha escrito respectivamente

`$$`

`S_n=\underbrace{a+a+a+\ldots+a}_{\mbox{n términos}}`

`$$`

y

`$$`

`\Pi_n=\overbrace`

`{a\times a\times\ldots\times a}^{\mbox{n términos}}`
`$$`

Algo más elaborado es el ejemplo

$$e^{-\alpha x} y = \underbrace{\int dx \int dx \int \cdots \int}_{\alpha \text{ integraciones}} e^{-\alpha x} \varphi(x) dx$$

producido mediante la expresión

`$$`
`e^{-\alpha x} y = \underbrace{\int dx \int dx \int \cdots \int}_{\alpha \text{ integraciones}} e^{-\alpha x} \varphi(x) dx`
`$$`

3.14. Apilar símbolos

El control

`\stackrel{expresión superior}{expresión inferior}`

permite colocar un símbolo (o una expresión) encima de otro (o de otra expresión). Tiene dos argumentos, de los cuales la expresión que corresponde al primero se escribe sobre la del que corresponde al segundo. Así, escribiendo

`$$\mbox{La aplicación }X\stackrel{f}{\rightarrow}Y$$$`

obtenemos

$$\text{La aplicación } X \xrightarrow{f} Y$$

3.15. Expresiones matriciales

El entorno `array` permite escribir matrices y expresiones de tipo matricial. El inicio del entorno es el siguiente

`\begin{array}[posicion]{ajuste columnas}`

que dispone de dos argumentos: el primero, que es optativo, sirve para alinear la parte superior (`t`), central (`c`) o inferior (`b`) de la tabla con la línea, adoptándose `c` si no se indica nada. El segundo argumento, que indica cómo se han de ajustar las entradas de las columnas, funciona como en el caso del entorno `tabular` en el modo texto: un conjunto de combinaciones de `l` (izquierda), `c` (centro) y `r` (derecha). Para la disposición de los elementos sirven los mismos controles que en el entorno `tabular`: el símbolo `&` separa un elemento de una columna del siguiente en la misma fila, y el control de cambio de línea (`\\`) termina una fila e inicia la siguiente (no es necesario en

la última). El entorno `array` se limita a colocar tabularmente los elementos, sin establecer delimitador alguno. Así, si se escribe

```


$$\begin{array}{rrr}
123 & 12 & 24 \\
35 & 1 & 17 \\
3 & 112 & 424
\end{array}$$


```

obtendremos simplemente

$$\begin{array}{rrr}
123 & 12 & 24 \\
35 & 1 & 17 \\
3 & 112 & 424
\end{array}$$

de modo que si se desean los delimitadores habrá que incluirlos, como en

```


$$\left( \begin{array}{rrr}
123 & 12 & 24 \\
35 & 1 & 17 \\
3 & 112 & 424
\end{array} \right)$$


```

que produce la matriz

$$\left(\begin{array}{rrr}
123 & 12 & 24 \\
35 & 1 & 17 \\
3 & 112 & 424
\end{array} \right)$$

El que el entorno `array` no imponga delimitadores permite utilizarlo para expresiones en las cuales se desea disponer ciertos elementos de modo equilibrado, como ocurre en

$$\phi(x) = \begin{cases} 1 & \text{para } x > 0 \\ 0 & \text{para } x \leq 0 \end{cases}$$

obtenida escribiendo

```


$$\phi(x) = \left\{ \begin{array}{c} 1 \\ \text{para } x > 0 \end{array} \right.$$


```

```

0 & \mbox{para }x\le 0
\end{array}
\right.
$$

```

3.16. Ajuste de espacios

Para llevar a cabo ajustes de espaciado horizontal entre elementos de una expresión se dispone de los controles siguientes:

<code>\,</code>	delgado	<code>\:</code>	medio
<code>\!</code>	delgado negativo	<code>\;</code>	grueso
<code>\quad</code>	un «quad»	<code>\qquad</code>	dos «quads»

Los controles `\,`, `\quad` y `\qquad` son válidos también en modo texto. Como ejemplo puede valer el caso de la expresión

$$\int f(x)dx$$

escrita del modo

```

$$\int f(x)dx$$

```

y en la cual se aprecia la conveniencia de separar dx de $f(x)$, lo cual se consigue escribiendo

```

$$\int f(x)\,dx$$

```

con lo que se obtiene

$$\int f(x) \, dx$$

El símbolo de control `\!` produce un espacio negativo, es decir que suprime espacio. Su empleo es conveniente en ciertos casos, como por ejemplo en el de las integrales múltiples, donde la separación estándar es excesiva. Así, las expresiones

```

$$\int\int_0^{\pi}$ y $\int\int\int_0^{\infty}$$

```

se traducen en

$$\int \int_0^{\pi} \text{ y } \int \int \int_0^{\infty}$$

respectivamente, mientras que, suprimiendo espacios escribiremos

```

$$\int\!\!\!\!\!\int_0^{\pi}$ y $\int\!\!\!\!\!\int\!\!\!\!\!\int_0^{\infty}$

```

para obtener

$$\iint_0^{\pi} \text{ y } \iiint_0^{\infty}$$

de más correcta factura.

3.17. Alinear expresiones

Para alinear expresiones podemos utilizar, como hemos visto, el entorno `array`. Si, al igual que cuando se utiliza el entorno `equation`, se quiere a la vez numerar las expresiones, se empleará el entorno `eqnarray`, similar a los entornos `tabular` y `array`, con la diferencia de que aquí no hay que declarar el número y el ajuste de las columnas. Así, por ejemplo

```
\begin{eqnarray}
(a+b)(a+b)&=&a^2+2ab+b^2\\
(a+b)(a-b)&=&a^2-b^2
\end{eqnarray}
```

dará como resultado la expresión

$$(a+b)(a+b) = a^2 + 2ab + b^2 \quad (3.2)$$

$$(a+b)(a-b) = a^2 - b^2 \quad (3.3)$$

Se observará que se numeran las líneas, porque se supone que cada una forma una ecuación, lo que será el caso más probable. Si una ecuación ocupa más de una línea, habrá que recurrir a un entorno separado para numerarla individualmente.

El entorno `eqnarray*` equivale al anterior, con la salvedad de que no numera las expresiones.

Ciertos alineamientos especiales pueden conseguirse dejando en blanco las entradas que convenga, como en

```
\begin{eqnarray*}
{\cal L}[x^n] &= & \int_0^\infty x^n e^{-px} dx \\
&= & \frac{1}{p^{n+1}} \int_0^\infty \zeta^{(n+1)-1} e^{-\zeta} d\zeta \\
&= & \frac{\Gamma(n+1)}{p^{n+1}}
\end{eqnarray*}
```

que da como resultado

$$\begin{aligned} \mathcal{L}[x^n] &= \int_0^\infty x^n e^{-px} dx \\ &= \frac{1}{p^{n+1}} \int_0^\infty \zeta^{(n+1)-1} e^{-\zeta} d\zeta \\ &= \frac{\Gamma(n+1)}{p^{n+1}} \end{aligned}$$

3.18. Teoremas y estructuras similares

Las estructuras semejantes a las que se utilizan normalmente en el enunciado de los teoremas se pueden declarar en forma personalizada del modo siguiente

`\newtheorem{nombre_entorno}{título}`

donde

- `nombre_entorno` es el nombre adjudicado por el usuario al entorno con el que se va a manejar la estructura.
- `título` es la palabra o frase con la que se titulará la estructura.

Ejemplos posibles son:

```
\newtheorem{teor}{Teorema}
\newtheorem{coro}{Corolario}
\newtheorem{defi}{Definición}
\newtheorem{prop}{Proposición}
\newtheorem{conj}{Conjetura}
\newtheorem{lemp}{Lema principal}
```

para ser utilizadas más adelante en el texto en expresiones del tipo

```
\begin{teor}
Si  $\forall x \in A$  se tiene que  $x \in B$ , entonces  $\ldots$ 
\end{teor}
```

que da lugar a la expresión

Teorema 1 *Si $\forall x \in A$ se tiene que $x \in B$, entonces...*

Las expresiones definidas mediante `\newtheorem` se numeran independientemente según el tipo (teoremas, definiciones, etc.). Si se escribe con destino a revistas de instituciones, hay que procurar no manipular las características de la letra del enunciado de las estructuras de este tipo, porque es corriente que dichas revistas tengan sus propios estilos definidos para estas componentes y aplicables en el caso de que las mismas no contengan modificaciones de usuario.

Capítulo 4

Gráficos

...¿y para qué sirve un libro —pensó Alicia—
sin dibujos ni conversaciones?

Lewis Carroll,
«Alicia en el país de las maravillas»

4.1. El entorno `picture`

El entorno `picture` permite crear gráficos con líneas, flechas, círculos, e incluso texto. Su declaración es de la forma:

```
\begin{picture}(ancho,alto)(x offset,y offset)
...
controles del entorno picture
...
\end{picture}
```

Este entorno tiene un argumento obligado, que define las dimensiones del gráfico como un rectángulo cuya anchura y altura son las especificadas en forma de coordenadas. Las coordenadas son números que admiten decimales y pueden tener signo negativo (por ejemplo 12, 4.5, -2.765), y que especifican longitudes en múltiplos de la unidad de longitud `\unitlength`, cuyo valor se modifica mediante el control

```
\setlength{\unitlength}{longitud}
```

donde `longitud` puede ser una cualquiera de las reconocidas por el `LATEX` (ver sección 1.10).

El entorno `picture` tiene además un argumento optativo, que permite indicar los desplazamientos horizontal y vertical del vértice inferior izquierda del rectángulo que encierra al gráfico. Hay que tener en cuenta que los

desplazamientos son positivos hacia la izquierda horizontalmente (valor `x offset`) y hacia abajo verticalmente (valor `y offset`). Así, si `\unitlength` se ha establecido como 1 mm, mediante el mandato

```
\setlength{\unitlength}{1mm}
```

entonces la declaración

```
\begin{picture}(40,30)(-20,-10)
```

dará como resultado un gráfico de 40 mm de ancho y 30 de alto, con su vértice inferior izquierdo desplazado al punto de coordenadas (20,10) relativo a su posición inicial.

Las dimensiones del gráfico establecida por el argumento obligatorio no impiden que los objetos que se dibujen en el mismo rebasen esos límites. La declaración de dichas dimensiones sirve para que el L^AT_EX prepare el espacio necesario previsto para el gráfico.

Los objetos que se incluyen en el gráfico obedecen al control `\put`:

```
\put (coordx,coordy){objeto}
```

que coloca el objeto especificado por el argumento `objeto` en el gráfico, con su punto de referencia —punto que depende del objeto— en las coordenadas (coordx,coordy) relativas al vértice inferior izquierda del rectángulo previsto para el gráfico.

Se citan a continuación los controles del entorno `picture`, apuntando someramente la función que llevan a cabo. La sección ?? del apéndice ?? pasa revista de modo más completo a la funcionalidad y los argumentos de los mismos.

`\circle` Un círculo.

`\dashbox` Un rectángulo a trazos.

`\frame` Un recuadro alrededor de un objeto.

`\framebox` (picture) Un rectángulo con un recuadro alrededor.

`\line` Un segmento.

`\linethickness` Establece el grueso de la línea.

`\makebox` (picture) Un rectángulo del tamaño especificado.

`\multiput` Produce varias copias de un objeto.

`\oval` Un recuadro con los vértices redondeados.

`\put` Coloca un objeto en un lugar especificado.

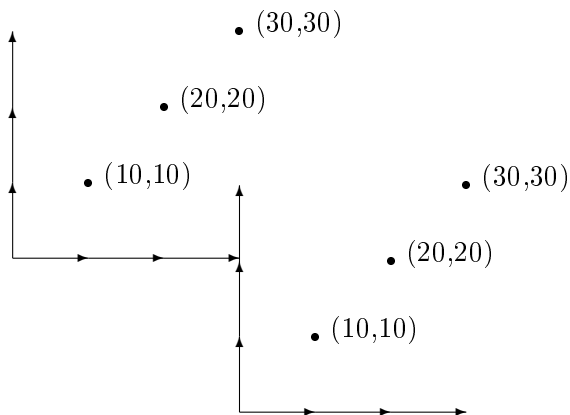
`\shortstack` Apila objetos.

`\vector` Segmento rematado por una flecha.

Como ejemplo, puede verse el resultado de declarar dos esquemas, el primero mediante los controles

```
\begingroup
\setlength{\unitlength}{1mm}
\begin{picture}(80,40)(0,0)
\put(0,0){\vector(1,0){10}}\put(0,0){\vector(0,1){10}}
\put(10,0){\vector(1,0){10}}\put(0,10){\vector(0,1){10}}
\put(20,0){\vector(1,0){10}}\put(0,20){\vector(0,1){10}}
\put(10,10){\circle*{1}}
\put(20,20){\circle*{1}}
\put(30,30){\circle*{1}}
\put(12,10){(10,10)}
\put(22,20){(20,20)}
\put(32,30){(30,30)}
\end{picture}
\endgroup
```

y el segundo repitiendo los mismos mandatos, pero con un desplazamiento para el entorno `picture` de $(-30,-20)$, en lugar del $(0,0)$ del primero.



Existe un mandato del entorno `picture` al que se le puede sacar partido si se dispone de recursos de programación. Se trata del control `\bezier`, que está determinado por tres puntos, P_1, P_2, P_3 , y traza un arco parabólico (*curva cuadrática Bezier*) que enlaza los puntos P_1 y P_3 , siendo la línea

definida por P_1, P_2 tangente a la curva en P_1 , y la definida por P_2, P_3 tangente en P_3 . La curva no pasa por el punto P_2 , a no ser que los tres puntos sean colineales. El formato del control es el siguiente:

`\bezier {n}(x_1, y_1)(x_2, y_2)(x_3, y_3)`

donde n es el número de puntos de que consta el arco. Unos cincuenta por centímetro lineal se consideran suficientes para generar una curva de apariencia continua.

Se comprende que una secuencia de mandatos **bezier**, cada uno de los cuales se inicia con el punto final del anterior, permite dibujar una determinada función, siempre que el intervalo de la misma se divida en un número suficiente de puntos. Mediante un programa es sencillo producir un archivo de texto cuyas líneas sean tales mandatos, dividiendo el ámbito de definición de la función en trozos suficientemente pequeños. La curva senoidal de la figura ?? se ha producido de la forma descrita. El programa auxiliar produce el archivo de texto **f seno.tex**, cada una de cuyas líneas es un mandato **bezier**. El intervalo $(0, 4\pi)$ se ha dividido en doscientas partes, y en cada parte se han incluido 10 puntos, de modo que la secuencia de mandatos tiene el aspecto siguiente:

```
\bezier{10}(0.00,30.00)(0.50,30.50)(1.50,31.49)
\bezier{10}(1.00,31.00)(1.50,31.49)(2.50,32.46)
\bezier{10}(2.00,31.98)(2.50,32.46)(3.50,33.39)
...
```

En el apéndice ?? puede verse el código *pascal* del módulo. El archivo se incluye sin más en el entorno **picture**, tal como se indica a continuación

```
\begin{figure}
\setlength{\unitlength}{1mm}
\begin{picture}(120,70)
\input{fseno}
\put(0,30){\vector(1,0){110}}
\put(0,30){\vector(0,1){30}}
\put(38,5){\framebox(40,10){Función  $y=\text{sen } x$  }}
\end{picture}
\label{fig:fseno}
\caption{\small{Senoide obtenida con
el mandato {\tt \$\backslash$bezier}.}}
\end{figure}
```

Curvas para las cuales a un valor de x corresponda más de uno de y pueden trazarse por el mismo método si se dispone de su ecuación en polares, o de la expresión paramétrica de dicha curva. De este modo ha resultado

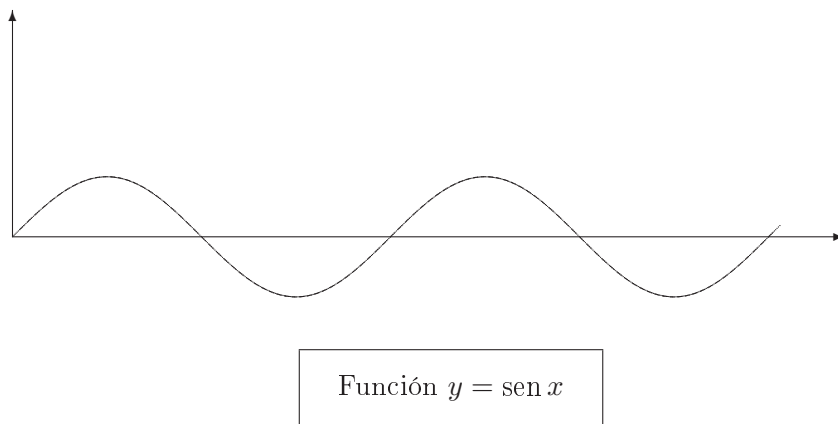


Figura 4.1: Senoide obtenida con el mandato `\bezier`.

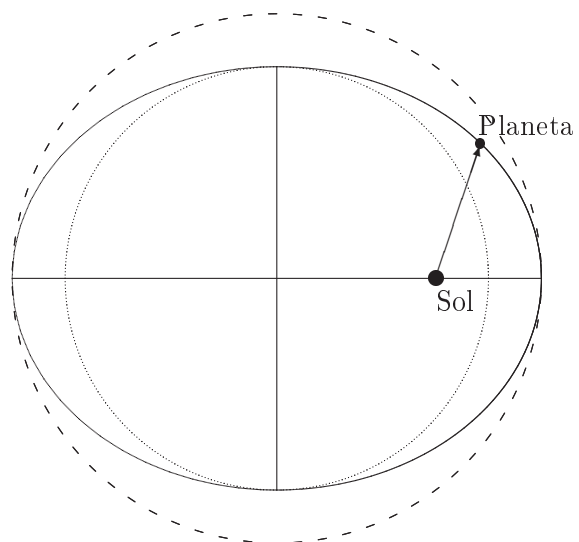


Figura 4.2: Cónicas obtenidas con el mandato `\bezier`.

también sencillo trazar la elipse y las dos circunferencias de la figura ???. La elipse tiene una excentricidad de $3/5$, valor que, aunque exagerado para representar fielmente en esquema una órbita planetaria, permite apreciar con facilidad la naturaleza ovalada de la curva. La circunferencia de radio el semieje menor de la elipse se ha trazado punteada, por el procedimiento de introducir pocos puntos «*bezier*» por tramo; la circunferencia se ha dividido en 400 tramos, y se han asignado 4 puntos por tramo. En cuanto a la externa, cuyo radio es el semieje mayor de la elipse, se ha dividido igualmente en 400 tramos. Para obtener los trazos se ha procedido a ignorar alternativamente tramos. Los que se dibujan tienen asignados 20 puntos intermedios «*bezier*». Los mandatos que corresponde a esta figura son los que siguen:

```
\begin{figure}
\setlength{\unitlength}{0.7mm}
\begin{picture}(100,80)(-40,-10)
\input{circulos} % circulos (trazos y puntos)
\input{elipse} %elipse excentricidad 3/5
\put(80,40){\circle*{3}}
\put(88.34,65.67){\circle*{2}}
\put(0,40){\line(1,0){100}}
\put(50,40){\line(0,1){40}}
\put(50,40){\line(0,-1){40}}
\put(80,40){\vector(1,3){8.4}}
\put(80,34){Sol}
\put(88,67){Planeta}
\end{picture}
\label{fig:elipse}
\caption{\small{Cónicas obtenidas con el
mandato {\tt \$\backslash$bezier}.}}
\end{figure}
```

4.2. Gráficos externos

Los drivers PostScript incorporados a la familia T_EX permiten importar gráficos definidos en archivos externos, haciendo posible además otras funciones tales como el escalado o la rotación de los mismos, modificaciones en relación con el color, etc.

Dos son los *packages* de mandatos estándar para el gobierno de los drivers aludidos:

- **graphics**, o el más avanzado **graphicx**, para el manejo de gráficos
- **color** para el gobierno de las características relacionadas con el color

Los ejemplos de incorporación de gráficos a partir de archivos externos que se presentan en el resto de este capítulo suponen que se ha declarado en

el preámbulo del documento la inclusión del *package* avanzado de gráficos, mediante el control

```
\usepackage{graphicx}
```

El *package graphicx* permite incorporar gráficos externos mediante el control

```
\includegraphics[clave=valor, ...]{nombre_archivo}
```

En cuanto a las palabras clave, hay que distinguir dos tipos:

- **variables numéricas**, a las que hay que asignar un valor
- **booleanas** (o **flags**) que pueden tomar los valores *cierto* o *falso*. El mero hecho de declararlas hace que adopten el valor *cierto*.

A continuación se repasan las palabras clave del mandato:

scale = *ratio* donde *ratio* fija la proporción por la que se escala en tamaño el gráfico, en relación con su tamaño original (el definido en el archivo).

width = *longitud* donde *longitud* establece la anchura del gráfico. Si no se especifica la altura (mediante **height**), ésta se fija guardando la proporción original del gráfico.

height = *longitud* donde *longitud* establece la altura del gráfico. Si no se indica la anchura del mismo (mediante la clave **width**, el factor de escalado será el que se derive de la altura, para mantener la proporción original.

totalheight = *longitud* donde *longitud* establece la altura y la profundidad (*depth*) del gráfico. Se recomienda emplear esta clave en lugar de **height** en el caso de que se haya rotado el gráfico.

keepaspectratio (*=true/false*) ; si se han especificado tanto **width** como **height**, esta clave garantiza que se mantiene la proporción original de la imagen, cuidando además que la figura no sobrepase ninguna de las dos dimensiones indicadas.

angle = *ángulo* donde *ángulo* es el valor en grados (y en sentido contrario al de las agujas del reloj) del ángulo por cuya medida se gira la figura. También se giran las dimensiones de las claves **width** y **height**; la anchura pasa a ser la altura (ángulos positivos) o la profundidad (ángulos negativos), y la altura se convierte en la anchura.

origin = *posición* donde *posición* establece el punto alrededor del cual gira la figura. El valor por omisión es **bl** (*bottom-left*, o sea el vértice inferior izquierda). Se interpretan asimismo

c *center* para el centro
t *top* para la parte superior
r *right* para la parte derecha
B *Baseline* para la línea de base
 combinaciones compatibles de los valores anteriores, como **br** para el vértice inferior derecha, por ejemplo.

draft = (*=true/false*) establece la modalidad *borrador*, de modo que el archivo del gráfico no se carga, y en su lugar se imprime un recuadro del tamaño calculado para la figura, en el que se incluye el nombre del archivo a efectos de comprobación. La opción **draft** («borrador») puede emplearse con validez general, como una de las opciones de la declaración **documentclass**. En ese caso extiende su efecto al documento en su totalidad, lo que puede ser ventajoso cuando se presta atención solamente al texto, por ejemplo.

clip = (*=true/false*) suprime la impresión de cualquier gráfico externo al marco de la figura.

bb = *llx lly urx ury* permite indicar las coordenadas del marco de la figura, por si éstas faltaran en el archivo original o fuesen incorrectas. Se trata de cuatro valores de longitud separados por espacios en blanco. Se pueden indicar las unidades; de no hacerlo, se adopta la unidad **bp**.

viewport = *llx lly urx ury* indica las coordenadas del marco de la figura, pero relativas al vértice inferior izquierda de la que señala el archivo. Permite (combinado con **clip**) seleccionar una porción de la figura completa.

trim = *dllx dlly durx dury* provoca la reducción del marco original de la figura según los valores especificados.

hiresbb = (*=true/false*) lee la información acerca del marco de la imagen a partir de la línea `%%HiResBoundingBox` del archivo gráfico.

Las claves son optativas, y pueden escribirse en cualquier orden, separándose mediante comas unas de otras. Las siguientes secciones muestran ejemplos del uso del mandato `\includegraphics`.

4.2.1. Gráficos vectoriales

El gráfico mostrado en ?? se realizó en un PC con el programa PowerPoint¹, y se grabó en formato **.wmf** (**W**indows **M**eta**F**ile), ocupando unos 45 Kbytes de espacio en disco. Con el convertidor **wmf2eps** se pasó a formato

¹Copyright© Microsoft Corporation.

.eps, (**E**ncapsulated **P**ost**S**cript), con una ocupación definitiva de unos 113 Kbytes.

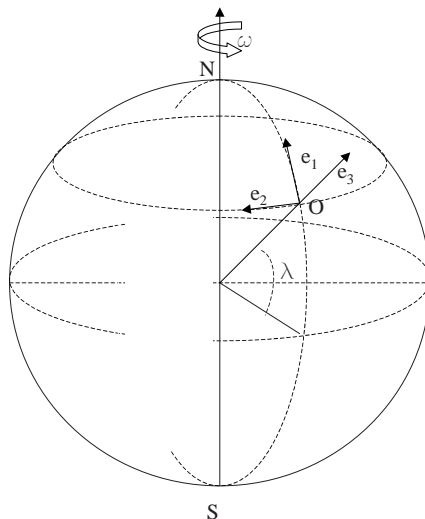


Figura 4.3: Péndulo de Foucault

4.2.2. Imágenes

Existen convertidores de imágenes digitales (tipo «*bitmap*» al formato .eps, a partir de formatos diversos.

La imagen de la figura ?? se ha incorporado mediante las sentencias

```
\begin{figure}[hbt]
\begin{center}
\colorbox{yellow}{\includegraphics[width=80 mm,
keepaspectratio]{venecia.eps}}
\end{center}
\caption{Venecia. Paisaje.}
\label{fig:venecia}
\end{figure}
```

que definen un ancho de 80 mm para la imagen, conservando sus propias proporciones (*keepaspectratio*). La imagen **venecia.eps** proviene de una originalmente en formato ordinario .jpg, que se ha convertido al formato .eps mediante un módulo convertidor jpeg2ps (en inglés, el 2 se lee como “to”).

La conversión se hace sin más que indicar en línea de mandatos (entorno PC/DOS):

```
jpeg2ps venecia.jpg >venecia.eps
```




Figura 4.4: Venecia. Paisaje.



Figura 4.5: Venecia. Imagen refleja.

La declaración siguiente hará que la imagen ocupe la amplitud horizontal del texto:

```
\begin{figure}[hbt]
\begin{center}
\includegraphics[width=\textwidth]{venecia.eps}
\end{center}
\caption{Venecia (ancho del texto)}
\end{figure}
```

La figura ?? presenta la imagen simétrica de la anterior con respecto al borde vertical izquierdo. El mandato utilizado proviene del *package* `graphics`, cuya sintaxis es en general compatible con la del `graphicx`. La declaración de la figura es la siguiente:

```
\begin{center}
\reflectbox{\includegraphics[width=80 mm]{venecia.eps}}
\end{center}
\caption{Venecia. Imagen refleja.}
\label{fig:veneciasim}
```

Las imágenes se pueden disponer rotadas un determinado ángulo, como ocurre al declarar:

```
\begin{figure}[hbt]
\begin{center}
\includegraphics[width=60 mm, angle=90]{venecia.eps}
\end{center}
\caption{Venecia (girada 90 grados)}
\label{fig:venecia90}
\end{figure}
```

que da lugar a la disposición girada 90° que muestra la figura ??.

En la figura ?? se presentan dos imágenes agrupadas mediante el control `\hbox` (*‘caja’ horizontal*). Cada una de ellas, por otra parte, se ha girado: la de la izquierda (*Daumier, «Don Quijote llegando al valle»*) en un ángulo positivo de 26 grados; la segunda (*la torre de Pisa*) en un ángulo negativo de 11 grados, en un intento virtual de restituir su verticalidad. Los siguientes mandatos corresponden a este efecto:

```
\begin{figure}[thb]
\hspace{20mm}
\hbox{
\includegraphics[width=30mm, angle=26]{daumier.eps}
\includegraphics[width=20mm, angle=-11]{torrepisa.eps}
}
```



Figura 4.6: Venecia (girada 90 grados)

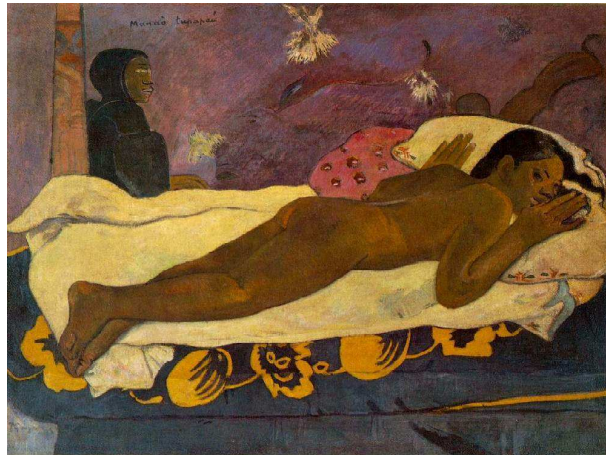


Figura 4.7: Gauguin.bmp



Figura 4.8: Paul Gauguin, « *Manae tupapau* » (1893) («*El espíritu de los muertos vela*»), 1893. Colección A. Conger Goodyear, N.Y.

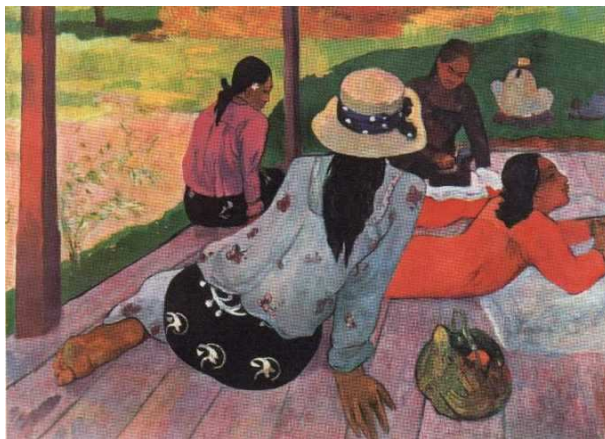


Figura 4.9: Paul Gauguin, « *La siesta* » (1893)



Figura 4.10: Paul Gauguin, «*Otahi: sola*» (1893)

```
\caption{Enderezando entuertos.}
\label{fig:enderezando}
\end{figure}
```

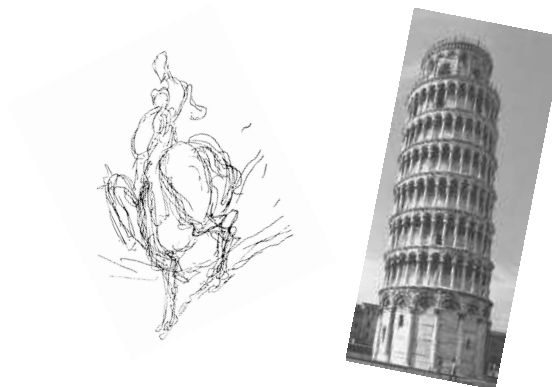


Figura 4.11: Enderezando entuertos.

Compárese la figura ?? con la ?? en cuanto a la calidad conseguida mediante escanear ...

4.3. Programas para gráficos en UNIX/LINUX

XFIG Es una especie de Paint-Brush para Linux. Es muy sencillo y puede generar archivos *.ps para incluir en el archivo LaTeX.

XPLOT, GNUPLOT Programa para generar funciones matemáticas. También genera archivos *.ps

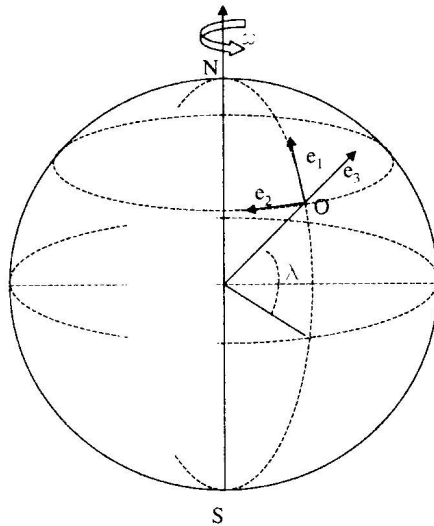


Figura 4.12: Figura ?? escaneada

PAW Es una aplicación propia del CERN. Sirve para tratamiento estadístico y representaciones "físicas". Es bastante liado, pero tengo manuales y yo lo he usado bastante.

De la WEB En <http://linuxapps.com> se encuentran apartados sobre gráficos, y gráficos científicos (plots, math (GEG))

Capítulo 5

Generalidades

Los escritores de los que me hablas eran hombres
como tú y como yo. Lo que ellos hicieron, puedes
hacerlo tú. Escribe libros, y podrás leerlos luego.

Jules Renard,
«Piel de zanahoria»

5.1. Referencias cruzadas

Las referencias que se hacen a capítulos, secciones, tablas, figuras, ecuaciones y otros elementos del documento se basan en la identificación de dichos elementos, la cual se lleva a cabo mediante el control

`\label{etiqueta}`

donde *etiqueta* es el nombre que se ha dado al punto al que deseamos referirnos desde otro lugar del documento, nombre que puede contener letras, números y símbolos, con distinción entre las letras mayúsculas y las minúsculas. Si se etiqueta un punto en medio de un texto, la referencia citará el número de la sección correspondiente; si la etiqueta se coloca en un entorno numerado, como por ejemplo una tabla o una figura, la referencia apuntará al número que corresponda a dicho entorno. Es habitual, para evitar repetir nombres, iniciar las etiquetas con un sufijo alusivo al elemento etiquetado:

`cap:` para los capítulos

`sec:` para las secciones

`tab:` para las tablas

`fig:` para las figuras

`ecu:` para las ecuaciones

etc.

de forma que la etiqueta que identifica una figura será del tipo

```
\label{fig:esquema1},
```

por ejemplo

Para la referencia se emplea el control:

```
\ref{etiqueta}
```

Así, por ejemplo, si un punto del documento, que puede ser por ejemplo el encabezamiento de la presente sección, se etiqueta con el nombre *sec:refcruzadas*, como efectivamente se ha hecho en la misma línea de encabezamiento de la sección del modo que sigue:

```
\section{Referencias cruzadas}\label{sec:refcruzadas}
```

entonces, para referirnos a la sección (esta misma en el caso del presente ejemplo), lo haremos mediante una frase del estilo de la siguiente:

Las referencias cruzadas se tratan en la
sección~\ref{sec:refcruzadas}, en la cual se consideran...

Las referencias cruzadas se tratan en la sección ??, en la cual se consideran...

Obsérvese que se ha recurrido a la “ligadura” de palabras (mediante la tilde) para evitar que se separen, debido a un posible cambio de línea, en el texto resultante la palabra *sección* del número de la misma.

Si se desea indicar también la página en la que se encuentra el elemento aludido por la referencia, el control a emplear es

```
\pageref{etiqueta}
```

de modo que en el ejemplo antes presentado la referencia que incluye la página se establecerá del modo siguiente:

Las referencias cruzadas se tratan en la
sección~\ref{sec:refcruzadas} de
la página~\pageref{sec:refcruzadas}
en la cual se consideran...

y el resultado tendrá la apariencia:

Las referencias cruzadas se tratan en la sección ?? de la página ?? en la cual se consideran...

Si en el texto de una sección se ha colocado una etiqueta en un punto cualquiera, el control `\ref` apunta como se ha dicho al número de la sección,

pero `\pageref` señalará el número de la página del texto donde se ha puesto la etiqueta, que puede no coincidir con la que corresponde al inicio de la sección.

Debe tenerse en cuenta asimismo que es frecuente que tenga que procesarse dos veces al menos el documento origen para que queden resueltas las referencias cruzadas. Las referencias no resueltas aparecen normalmente indicadas mediante la secuencia `??`. Se obtienen también, por otra parte, los mensajes de aviso correspondientes para el caso de que no se hayan podido resolver algunas referencias cruzadas.

5.2. Página de título

La página de título del documento se obtiene mediante los controles siguientes:

`\title` {título del documento}

`\author` {autor}

`\date` {fecha, por ej. 12 mayo 1999}

`\titlepage`, si se desea que el título figure en una página separada.

`\maketitle`, para producir efectivamente el título.

Excepto el control `\maketitle`, los otros pueden estar en el preámbulo (antes de `\begin{document}`).

Títulos extensos

No representa ningún problema el que un título necesite de varias líneas, ya que el texto del argumento puede contener cambios de línea, de tipo o de tamaño de letra si se desea (ver el ejemplo que se presenta más abajo).

Varios autores

El control `\and` permite separar los nombres de varios autores dentro del mismo campo del argumento de `\author`. También es posible utilizar varias líneas, para indicar la dirección del autor o la institución a la que pertenece.

Ejemplo

El ejemplo siguiente muestra cómo escribir una página de título con doble línea para el título, y varios autores con la indicación de sus instituciones.

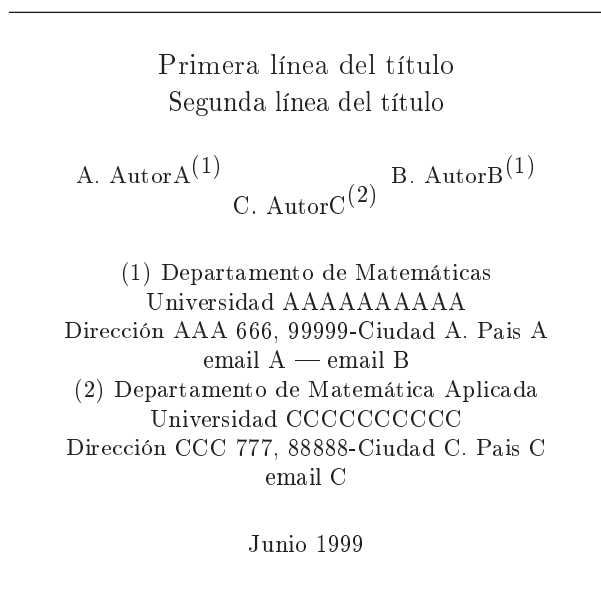


Figura 5.1: Esquema de una página de título

```

\title{\huge Primera línea del título\\
\Large Segunda línea del título}\\
\normalsize
\author{A.~AutorA$\sim\mbox{(1)}$ \and B.~AutorB$\sim\mbox{(1)}$
\and C.~AutorC$\sim\mbox{(2)}$}\ {} (1) Departamento de
Matemáticas \\ Universidad AAAAAAAAAA \\ Dirección AAA 666,
99999-Ciudad A. Pais A \\ email A --- email B \\ (2) Departamento
de Matemática Aplicada \\ Universidad CCCCCCCCCC \\
Dirección CCC 777,
88888-Ciudad C. Pais C \\ email C}
\date{Junio 1999}
\titlepage
\begin{document}
\maketitle
%\thanks{Si hay agradecimientos o comentarios}
\begin{abstract} En el presente informe se describen de forma
esquemática las estructuras derivadas ...
\end{abstract}

```

La figura ?? deja ver en esquema el resultado de la página de título que se acaba de confeccionar. El control `\thanks`, pensado en principio para el párrafo de agradecimientos, permite cualquier comentario que proceda. Un resumen (*abstract*) preliminar se coloca de modo adecuado mediante el control `\abstract`.

5.2.1. Índice

La confección del índice (*Table of Contents*, es decir «Tabla del contenido» en la terminología inglesa), se lleva a cabo mediante el control

```
\tableofcontents
```

situado al igual que el anterior en el cuerpo del documento. Lo corriente es elegir un tamaño de letra algo menor para esta sección. Ya se ha visto cómo se cambia el tipo y el tamaño de letra (secciones ?? y ??). Un ejemplo de cómo conseguir este efecto, así como continuar en página separada, sería el siguiente:

```
\footnotesize
\tableofcontents
\normalsize
\newpage
```

donde:

`\footnotesize` cambia el tamaño de letra al empleado en las notas a pie de página, que es menor que el estándar.

`\tableofcontents` produce el índice propiamente dicho, como ya se ha comentado antes.

`\normalsize` restituye el tamaño de letra al valor establecido como estándar.

`\newpage` provoca que el texto que sigue a continuación comience en una página separada.

Si se desea cambiar el encabezamiento que el L^AT_EX proporciona automáticamente para el índice (por ejemplo, *Índice General*), puede emplearse el control

```
\def\contentsname{\'Indice}
```

y el nuevo encabezamiento será ahora *Índice*. El citado control debe colocarse en el cuerpo del documento, como por ejemplo inmediatamente antes del control para elaborar el índice (`\tableofcontents`).

5.3. El índice de palabras

Incluir palabras

Las palabras a incluir en el índice de palabras se indican mediante el control

```
\index{palabra}
```

En el caso de que se desee crear una palabra de entrada que hace de encabezamiento de otras, como ocurre por ejemplo con entradas del tipo

```
Coulomb
```

```
biografía
```

```
ley de
```

se incluirán las palabras en la forma

```
\index{Coulomb! biografía} y \index{Coulomb! ley de}
```

cada una donde corresponda.

Formación del índice de palabras

El índice de palabras (*Index* simplemente, en inglés) contiene, clasificadas por orden alfabético, las palabras que se hayan designado expresamente para aparecer en dicho apartado, con la indicación de la página en la que se encuentran. Se forma mediante el control

```
\makeindex
```

colocado antes del control de inicio del documento (`\begin{document}`). Dicho control hará que el \LaTeX tome en consideración las palabras señaladas por el control `\index`, y produzca el archivo intermedio `nombre.idx`, donde *nombre* indica el nombre del documento. Lo normal es que se necesite un programa separado para obtener el formato definitivo del índice de palabras. Así, si este programa se llama `makeindex.exe` (como ocurre en el MikTeX) por ejemplo, el mandato

```
makeindex nombre
```

tomará el archivo *nombre.idx* para producir el *nombre.ind*, el cual es el que se incluirá en el documento en el lugar que se decida como apropiado (usualmente al final del mismo). La inclusión, en el punto donde se desee colocar el índice de palabras, se hace mediante el control:

```
\include{nombre.ind}
```

donde *nombre* es el supuesto nombre del documento.

Si se desea cambiar el encabezamiento que el \LaTeX proporciona automáticamente para el índice de palabras (por ejemplo, *Índice de palabras*), puede emplearse el control

```
\def\indexname{\'Indice de materias}
```

que establece *Índice de materias* como nuevo encabezamiento.

5.4. La bibliografía

La sección de bibliografía se establece mediante el entorno

```
\begin{thebibliography}{amplitud}
\bibitem[etiqueta]{clave_de_cita}
.
.
.
\end{thebibliography}
```

En el tipo de documento *article*, el encabezamiento para la bibliografía es *References* en inglés. Los tipos de documento *report* y *book* se encabezan con *Bibliography*.

amplitud: Este argumento es un texto que, una vez impreso, es aproximadamente tan ancho como la más ancha de las etiquetas de elementos del control `\bibitem`.

\bibitem: Especifica los elementos bibliográficos.

\cite: Se refiere a un elemento bibliográfico.

\nocite: Incluye un elemento en la bibliografía.

Bibliografía

- [1] Rocky and Bullwinkle, Open problems, in *Mr. Know-it-all's Rock Encyclopedia*.
- [2] Bullwinkle, Getting things out of hats, *Annals* **1** (1990BC), 1–2.

Bibliografía

- [AFG] *AMSTeX* Amer. Math. Soc., Providence, RI, 1994.
- [AH] *AMS author handbook*, Amer. Math. Soc., Providence, RI, 1996.
- [ALG] *amstex version*, Amer. Math. Soc., Providence, RI, 1994.
- [ASMR] *Abbreviations of names of serials reviewed in Mathematical Reviews*, Amer. Math. Soc., Providence, RI, revised annually.
- [GMS] Michel Goossens, Frank Mittelbach, and Alexander Samarin, *The L^AT_EX companion*, Addison-Wesley Co., Reading, MA, 1994.
- [Joy] M. D. Spivak, *The joy of T_EX*, 2nd revised ed., Amer. Math. Soc., Providence, RI, 1990.
- [LM] Leslie Lamport, *L^AT_EX: A document preparation system*, 2nd revised ed., Addison-Wesley, Reading, MA, 1994.
- [MIT] Ellen E. Swanson, *Mathematics into type*, Amer. Math. Soc., Providence, RI, 1979.
- [ML] George Grätzer, *Math into L^AT_EX: An introduction*
- [NJH] Nicholas J. Higham, *Handbook of writing for the mathematical sciences*, SIAM, Philadelphia, PA, 1993.
- [NM] Norman Walsh, *Making T_EX Work*, O'Reilly & Associates, Inc., Sebastopol, CA, 1994.
- [SHSD] Norman E. Steenrod, Paul R. Halmos, Menahem M. Schiffer, and Jean A. Dieudonné, *How to write mathematics*, 4th printing 1993, Amer. Math. Soc., Providence, RI, 1973.
- [TB] Donald E. Knuth, *The T_EXbook*, Addison-Wesley, Reading, MA, 1984.

Using BibTeX: Automatic generation of bibliographies.

```
\bibitem
\bibitem[label]{cite_key}
```

The `\bibitem` command generates an entry labelled by `label`. If the `label` argument is missing, a number is generated as the label, using the `enumi` counter. The `cite_key` is any sequence of letters, numbers, and punctuation symbols not containing a comma. This command writes an entry on the `‘.aux’` file containing `cite_key` and the item’s label. When this `‘.aux’` file is read by the `\begin{document}` command, the item’s label is associated with `cite_key`, causing the reference to `cite_key` by a `\cite` command to produce the associated label.

```
\cite
```

```
\cite[text]{key_list}
```

The `key_list` argument is a list of citation keys. This command generates an in-text citation to the references associated with the keys in `key_list` by entries on the `‘.aux’` file read by the `\begin{document}` command.

The optional `text` argument will appear after the citation, i.e. `\cite[p. 2]{knuth}` might produce `‘[Knuth, p. 2]’`.

```
\nocite
```

```
\nocite{key_list}
```

The `\nocite` command produces no text, but writes `key_list`, which is a list of one or more citation keys, on the `‘.aux’` file.

Using BibTeX

If you use the BibTeX program by Oren Patashnik (highly recommended if you need a bibliography of more than a couple of titles) to maintain your bibliography, you don’t use the `thebibliography` environment. Instead, you include the lines

```
\bibliographystyle{style}
```

`\bibliography{bibfile}`

where `style` refers to a file `style.bst`, which defines how your citations will look. The standard styles distributed with BibTeX are:

`alpha` Sorted alphabetically. Labels are formed from name of author and year of publication. `plain` Sorted alphabetically. Labels are numeric. `unsrt` Like `plain`, but entries are in order of citation. `abbrv` Like `plain`, but more compact labels. In addition, numerous other BibTeX style files exist tailored to the demands of various publications.

The argument to `\bibliography` refers to the file `bibfile.bib`, which should contain your database in BibTeX format. Only the entries referred to via `\cite` and `\nocite` will be listed in the bibliography.

5.5. Diseño del documento maestro

```
\documentclass[a4paper,11pt,twoside]{report}
\usepackage{t1enc}
\usepackage[spanish]{babel} % idioma español
\usepackage{graphicx} % para incluir gráficos e imágenes
\makeindex
\begin{document}
\pagenumbering{roman} % las primeras páginas en números romanos
\title{Título del documento}
\normalsize % tamaño de letra estándar
\author{Autor del documento}
\date{fecha del documento}
\titlepage
\maketitle % página de título
\footnotesize % tamaño de letra pequeño
\def\contentsname{Índice} % cambiar el encabezamiento
    %del índice
\tableofcontents % para formar el índice
\normalsize
\newpage
\thispagestyle{empty} % página sin numerar
\pagenumbering{arabic} % números ordinarios de
    % página
%\include{nombpref} %prefacio
```

```
%\include{nombcap1} % introducción
%\include{nombcap2} % que trata de ...
...
%\include{nombcapN} % capítulo final
% Los apéndices empiezan aquí
\appendix
%\include{nombcapA}% apéndice A
...
%\include{nombcapD}% apéndice D
\newpage
%\include{mat.ind} % para el índice de palabras
\end{document}
```

Capítulo 6

Maravillas, titi

6.1. Espacios y cajas

uno de uno **I** uno de dos
dos de uno dos de dos

```
\hbox{\vbox{\hbox{uno de uno}\hbox{dos de uno}}
\kern2cm \vrule width4pt height20pt depth0pt\kern2cm
\vbox{\hbox{uno de dos}\hbox{dos de dos}}}
\mbox
```

ahora trazamos un cuadro tal como $\square\Phi$ o bien \square^Φ este que no hay manera de bajarlo un poco. Comparar con $E\Phi$ o bien E^Φ , y con

```
\def\cuadro{
\hbox{\lower.5ex
\vbox{\hrule width10pt
\hbox{\vrule width0.4pt height10pt depth0pt
\kern9.25pt \vrule width0.4pt height10pt depth0pt}
\hrule width10pt}\kern1pt}}
{}}
ahora trazamos un cuadro tal como \cuadro$\Phi$ o bien $\cuadro^\Phi$
este que no hay manera de bajarlo un poco. Comparar con $E^\Phi$.
{}}\
```

All the predefined length parameters See Predefined lengths can be used in the arguments of the box-making commands.

* Menu:

Horizontal space
\dotfill Stretchable horizontal dots.
\hfill Stretchable horizontal space.

`\hrulefill` Stretchable horizontal rule.
`\hspace` Fixed horizontal space. Vertical space
`\addvspace` Fixed vertical space.
`\bigskip` Fixed vertical space.
`\medskip` Fixed vertical space.
`\smallskip` Fixed vertical space.
`\vfill` Stretchable vertical space.
`\vspace` Fixed vertical space. Boxes
`\fbox` Framebox.
`\framebox` Framebox, adjustable position.
`lrbox` An environment like `\sbox`.
`\makebox` Box, adjustable position.
`\mbox` Box.

`\newsavebox` Declare a name for saving a box.
`\parbox` Box with text in paragraph mode.
`\raisebox` Raise or lower text.
`\rule` Lines and squares.
`\savebox` Like
`\makebox`, but save the text for later use.
`\sbox`
Like `\mbox`, but save the text for later use.
`\bezier{puntos}(x,y)(x,y)(x,y)`

aquí el control `\dotfill`y la línea termina aquí.

aquí el control `\hrulefill` _____y la línea termina aquí.

El control `\fbox` pone un rectángulo alrededor del escrito

`\parbox[position][height][inner-pos]{width}{text}`

este es el texto
 que debemos colo-
 car en el interior
 del `\parbox` para
 que lo encierre en
 una caja rectan-
 gular, que en es-
 te caso la hemos
 definido con seis
 centímetros de al-
 to y tres de ancho

Si queremos que la parbox se mueva algo a la derecha

este es ahora el
 texto que coloca-
 mos en el interior
 del parbox para
 que lo encierre en
 una caja rectan-
 gular, seis de alto
 y tres de ancho

A parbox is a box whose contents are created in paragraph mode. The `\parbox` has two mandatory arguments:

width specifies the width of the parbox, and

text the text that goes inside the parbox.

LaTeX will position a parbox so its centre lines up with the centre of the text line. The optional position argument allows you to line up either the top or bottom line in the parbox (default is top).

If the height argument is not given, the box will have the natural height of the text. The inner-pos argument controls the placement of the text inside the box. If it is not specified, position is used.

t text is placed at the top of the box.

c text is centred in the box.

b text is placed at the bottom of the box.

s stretch vertically. The text must contain vertically stretchable space for this to work.

A `\parbox` command is used for a parbox containing a small piece of text, with nothing fancy inside. In particular, you shouldn't use any of the paragraph-making environments inside a `\parbox` argument. For larger pieces of text, including ones containing a paragraph-making environment, you should use a minipage environment See minipage.

```
\begin{minipage}[position]{width}
  texto
\end{minipage}
```

The minipage environment is similar to a `\parbox` command. It takes the same optional position argument and mandatory width argument. You may use other paragraph-making environments inside a minipage.

Footnotes in a minipage environment are handled in a way that is particularly useful for putting footnotes in figures or tables. A `\footnote` or `\footnotetext`

command puts the footnote at the bottom of the minipage instead of at the bottom of the page, and it uses the mpfootnote counter instead of the ordinary footnote counter See Counters.

NOTE: Don't put one minipage inside another if you are using footnotes; they may wind up at the bottom of the wrong minipage.

aquí se escribe todo el texto que nos de la impresion que queremos establecer como una pagina, que tenga a su vez los clásicos párrafos.

por ejemplo ahora empezamos un nuevo párrafo^a y queremos ver el resultado, para ver si es cierto que la nota al pie se coloca solamente a pie de la minipágina.

^a esto es el pie de página de la minipágina

`\makebox[width] [position]{text}`

The makebox command creates a box just wide enough to contain the text specified. The width of the box is specified by the optional width argument. The position of the text within the box is determined by the optional position argument.

c centred (default)

l flushleft

r flushright

s stretch from left to right margin. The text must contain stretchable space for this to work.

See makebox (picture). Ahora a continuación un mandato makebox:

este es el texto que queremos insertar
Ahora los framebox, que son como los makebox con dibujo

`\framebox[width] [position]{text}`

The `framebox` command is exactly the same as the `makebox` command, except that it puts a frame around the outside of the box that it creates.

The `framebox` command produces a rule of thickness `\fboxrule`, and leaves a space `\fboxsep` between the rule and the contents of the box.

este es el texto que insertamos en el framebox

este es el texto que insertamos en el fbox

aquí se escribe todo el texto que nos de la impresion que queremos establecer como una pagina, que tenga a su vez los clásicos párrafos.

Ahora podríamos insertar una figura en la minipage a ver qué ocurre. Por ejemplo:

por ejemplo ahora empezamos un nuevo párrafo^a y queremos ver el resultado, para ver si es cierto que la nota al pie se coloca solamente a pie de la minipágina.

^a esto es el pie de página de la minipágina

6.2. trazar líneas

por ejemplo la sentencia que estan leyendo ahora se ha construido diciendo que lo que se quiere poner no tendría que colisionar con nada de lo que anteriormente hayamos podido establecer para la construcción de esta caja tan perfecta del ejercicio 21.3 que es algo increíble.

El control `\hrule` permite trazar líneas horizontales. Sin parámetros, la línea ocupará el ancho del texto; al escribir `\hrule` obtenemos la línea que se ve a continuación,

La línea trazada por el control `\hrule` se comporta como si de un nuevo párrafo se tratara: termina la línea de texto inmediatamente después de la palabra que lo precede, traza una línea gráfica establece un nuevo párrafo a partir de la palabra de texto que sigue al control. `\hrule` puede modularse en cuanto a su longitud y a su grueso mediante dos parámetros. Si escribimos

```
\hrule width6cm height4pt
```

obtendremos un segmento horizontal de 6 cm. de longitud, y de un grueso de 4 puntos (pt), tal como la siguiente

Para desplazar hacia la derecha vale el control

```
\moveright3cm \vbox{\hrule width4cm height5pt}
```

que produce el segmento siguiente:



tendremos lo que vemos como cuadradito, pero si es pues entonces esta acostado

6.3. Encuadrar texto

6.4. Recuadrar texto

6.5. Formato página

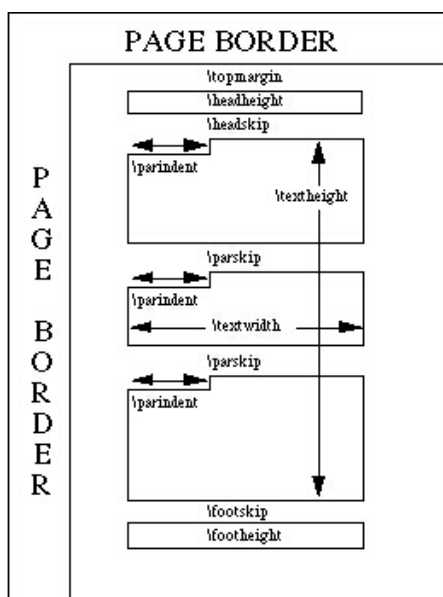


Figura 6.1: Formato de la página

Parameter Name	Purpose	Default Value	
		Points	Inches
mm \textheight	height of the text body	528	7-1/3
186 \textwidth	width of the text body	345	
4-19/24 122 \topmargin	blank space at page top		27
3/8 10 \headheight	page header height		12
1/6 4 \headsep	space between header and text		25

25/72	9	\oddsidemargin	odd-numbered page margin	44	
11/18	16	\evensidemargin	even-numbered page margin	82	
1-5/36	29				
\parskip		space between paragraphs	1	1/72	0.4
\parindent		paragraph indentation	15	5/24	5
\footskip		space between text and footer	30	5/12	
11 \footheight		height of the page footer	12	1/6	
4					

You can modify the page style by changing the parameter values either in the preamble of the .tex file or in the body of the text itself. The page border is one inch wide at the top and left side of the page. The values of \parindent and \parskip can be changed anywhere in your text. The other parameters in Figure 1 should be changed in the preamble only.

For example, to print pages with a text area 6 inches wide and 10 inches high, with no space for a page header, include the following lines in the preamble of your .tex file:

```
\headheight=0in \headsep=0in \oddsidemargin=0in
\evensidemargin=0in \textwidth=6in \textheight=10in
```

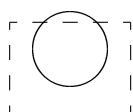

Apéndice A

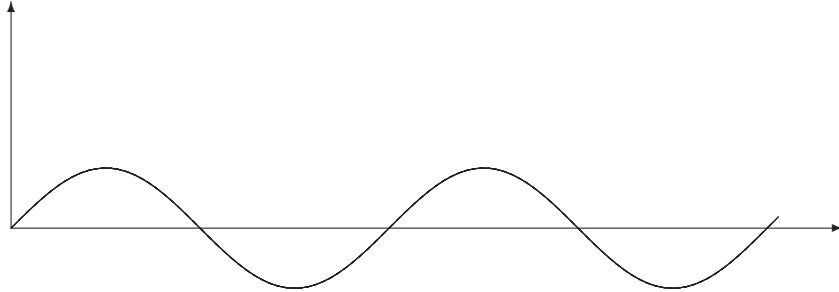
Controles de picture

`\circle[*]{diameter}`

The `\circle` command produces a circle with a diameter as close to the specified one as possible. If the `*`-form of the command is used, LaTeX draws a solid circle.

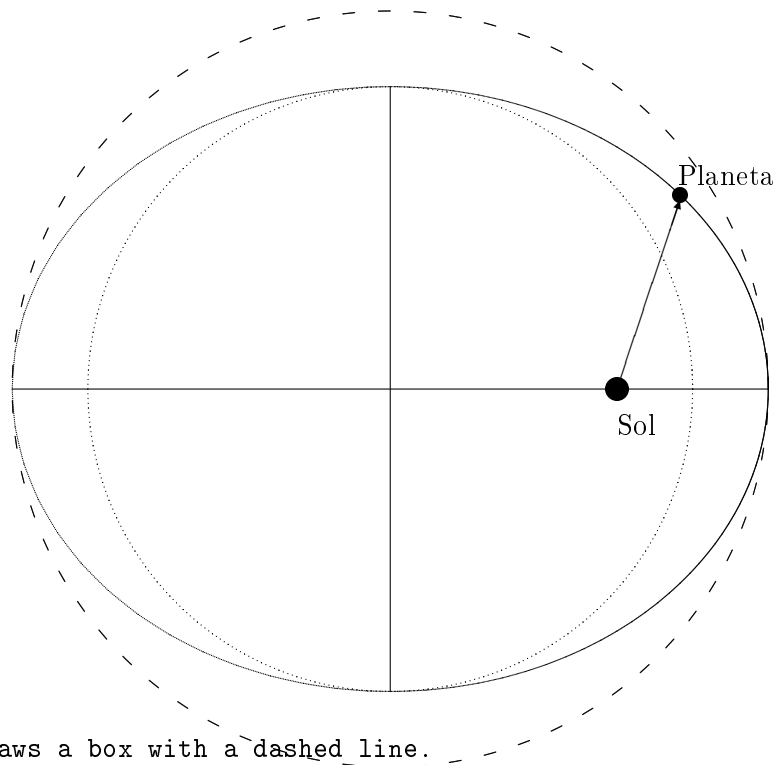
Note that only circles up to 40 pt can be drawn.





□
(40,10) Función $y = \sin x$

Figura A.1: Curva obtenida con el mandato `\bezier`.



Draws a box with a dashed line.

```
\dashbox{dash_length}(width,height){...}
```

The `\dashbox` has an extra argument which specifies the width of each dash. A dashed box looks best when the width and height are multiples of the `dash_length`.

```
\frame{...}
```

The `\frame` command puts a rectangular frame around the object specified in the argument. The reference point is the bottom left corner of the frame. No extra space is put between the frame and the object.

```
\framebox(width,height)[position]{...}
```

The `\framebox` command is exactly the same as the `\makebox` command, except that it puts a frame around the outside of the box that it creates.

The `\framebox` command produces a rule of thickness `\fboxrule`, and leaves a space `\fboxsep` between the rule and the contents of the box.

```
line(x slope,y slope){length}
```

The `\line` command draws a line of the specified length and slope.

Note that LaTeX can only draw lines with $\text{slope} = x/y$, where x and y have integer values from -6 through 6.

```
\linethickness{dimension}
```

Declares the thickness of horizontal and vertical lines in a `picture` environment to be `dimension`, which must be a positive length. It does not affect the thickness of slanted lines and circles, or the quarter circles drawn by `\oval` to form the corners of an oval.

```
\makebox(width,height)[position]{...}
```

The `\makebox` command for the `picture` environment is similar to the normal `\makebox` command except that you must specify a width and height in multiples of `\unitlength`.

The optional argument, [position], specifies the quadrant that your text appears in. You may select up to two of the following:

û t - Moves the item to the top of the rectangle

û b - Moves the item to the bottom

û l - Moves the item to the left

û r - Moves the item to the right

See \makebox.

`makebox[width][position]{text}`

The \makebox command creates a box just wide enough to contain the text specified. The width of the box is specified by the optional width argument. The position of the text within the box is determined by the optional position argument.

û c -- centred (default)

û l -- flushleft

û r -- flushright

û s -- stretch from left to right margin.

The text must contain stretchable space for this to work.

See \makebox (picture). \multput(x coord,y coord)
(delta x,delta y){number of copies}{object}

The \multput command can be used when you are putting the same object in a regular pattern across a picture.

`\oval(width,height)[portion]`

The `\oval` command produces a rectangle with rounded corners. The optional argument, `[portion]`, allows you to select part of the oval.

`û t` - Selects the top portion

`û b` - Selects the bottom portion

`û r` - Selects the right portion

`û l` - Selects the left portion

Note that the "corners" of the oval are made with quarter circles with a maximum radius of 20 pt, so large "ovals" will look more like boxes with rounded corners.

`put(x coord,y coord){ ... }`

The `\put` command places the item specified by the mandatory argument at the given coordinates.

`\shortstack[position]{... \\ ... \\ ...}`

The `\shortstack` command produces a stack of objects. The valid positions are:

`û r` - Moves the objects to the right of the stack

`û l` - Moves the objects to the left of the stack

`û c` - Moves the objects to the centre of the stack (default)

`\vector(x slope,y slope){length}`

The `\vector` command draws a line with an arrow of the specified length and slope. The `x` and `y` values must lie between -4 and +4, inclusive.

Apéndice B

Ejemplos de expresiones matemáticas

B.1. sec:instalar

como se instala un entorno de trabajo latex

B.2. ejemplos

$$\left(\begin{array}{ccc|ccc|c} x_2 & y_3 & z_3 & & & & \mathbf{0} \\ t_1 & u_2 & v_2 & & & & \mathbf{0} \\ 0 & w_1 & m_1 & & & & \\ \hline x_1 & y_2 & z_2 & x_2 & y_3 & z_3 & \\ 0 & u_1 & v_1 & t_1 & u_2 & v_2 & \mathbf{0} \\ 0 & 0 & 0 & 0 & w_1 & m_1 & \\ \hline 0 & y_1 & z_1 & x_1 & y_2 & z_2 & x_2 \\ 0 & 0 & 0 & 0 & u_1 & v_1 & t_1 \\ \hline 0 & 0 & 0 & 0 & y_1 & z_1 & x_1 \end{array} \right)$$

$$\begin{array}{ccc} \mathcal{M}(r, s) & \xrightarrow{\varphi} & \mathcal{M}(k, h) \\ \pi \downarrow & & \downarrow \pi \\ \mathcal{M}(r, s)/g(s) & \longrightarrow_{\tilde{\varphi}} & \mathcal{M}(k, h)/\mathcal{G}(h) \end{array}$$

Los símbolos siguientes provienen del amssymb. Necesitan del control `\usepackage{latexsym,amssymb}` en el preámbulo.

[illegible]

Apéndice C

Instalación

C.1. `sec:instalamos`

como se instala un entorno de trabajo latex

Índice alfabético

Coulomb

 biografía, 53

 ley de, 53

Referencias cruzadas, 51

Índice

 de palabras, 52