

Preparación de textos con L^AT_EX

1 Escritura básica de texto

El documento La estructura general de un documento L^AT_EX 2_ε es:

```
\documentclass[opciones]{clase}
```

Preámbulo

```
\begin{document}
```

Documento

```
\end{document}
```

Por ahora usaremos el preámbulo

```
\documentclass[a4paper,12pt]{article}
```

Espacios Uno o más espacios en blanco en el texto fuente producen un único espacio en el texto compilado. Un cambio de línea equivale a un espacio en blanco. Una línea en blanco produce un cambio de párrafo. Varias líneas en blanco consecutivas equivalen a una línea en blanco.

Para conseguir varios espacios seguidos hemos de usar `_`

Para prohibir el cambio de línea entre dos palabras se usa `~`. Por ejemplo, conviene escribir

```
...desde 1 hasta~10.
```

L^AT_EX deja un espacio adicional después de un punto y seguido. Este efecto ha de evitarse detrás de un punto correspondiente a una abreviatura, lo cual se consigue introduciendo un espacio normal `_`. Por ejemplo, hay que escribir

```
En la p\’agina 5 y ss.\ se habla de...
```

Sin embargo, cuando el punto está precedido de una mayúscula \LaTeX supone que se trata de una abreviatura y ya deja tras él un espacio normal, con lo que no es necesario hacer esto. Por ejemplo, podemos escribir

```
... el Dr.\ D. S. Ram\'on y Cajal
```

El punto tras la ‘r’ requiere la indicación de que no es final de frase, pero los puntos tras la ‘D’ y la ‘S’ no lo requieren. Estaría mejor aún si escribiéramos

```
... el Dr.~D.~S.~Ram\'on y Cajal
```

Recíprocamente, si una frase termina con mayúscula hemos de indicarlo para que \LaTeX inserte un espacio de fin de frase. Esto se consigue con $\@$. Por ejemplo, hemos de escribir

```
El m\'as votado ha sido el P.P\@. En segundo lugar...
```

Lo mismo sucede con el punto y coma, la interrogación, la exclamación e incluso con un punto seguido de paréntesis o comillas. Por ejemplo, hemos de escribir

```
Los c\'{\i}tricos (naranjas, etc.)\ tienen vitamina C\@.
```

Podemos obtener espacios horizontales o verticales de cualquier longitud mediante las instrucciones

```
\hspace{4mm} y \vspace{5.6cm}
```

El comando \vspace no funciona a principio de página (para evitar que quede un espacio en blanco indeseado al comienzo). Si pese a todo deseamos un espacio vertical a principio de página usamos $\vspace*$.

Si queremos una separación extra estándar entre algunos párrafos podemos usar

```
\smallskip, \medskip, \bigskip.
```

Signos ortográficos Los acentos y los signos ortográficos que no tienen un código ASCII estándar se consiguen como sigue:

á	$\'a$	à	$\`a$	â	$\^a$	ä	$\"a$
ñ	$\~n$	¿	$\?`$	¡	$\!`$	ç	$\c c$

El comando \i produce i , es decir, una i sin punto. Hemos de usarlo cuando queramos poner acentos o diéresis sobre la i , para evitar cosas como \acute{i} o \ddot{i} . Así pues, hemos de escribir $\text{Mar}\'{\i}a$ y no $\text{Mar}\'ia$.

\LaTeX tiene cuatro guiones distintos: $-$ se usa entre palabras, $--$ se usa entre números, $---$ es el guión ortográfico y $\-$ es el guión de partición de palabras que aparece sólo si es necesario. Notemos la diferencia:

Físico-químico, páginas 4-8, Hay—de hecho—varios tipos...

Las comillas se consiguen con el acento grave y el apóstrofo:

```
`comillas simples'           ‘comillas simples’,  
``comillas dobles''         “comillas dobles”,
```

Hay ciertos signos que sí tienen un código ASCII estándar pero que T_EX reserva para usos especiales. Para obtenerlos como meros signos hemos de usar los comandos siguientes:

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

Los puntos suspensivos se consiguen con `\ldots`.

Hay ciertos signos ortográficos para los que L^AT_EX no da una forma inmediata de conseguirlos, pero pueden ser definidos mediante instrucciones en el preámbulo del documento. Por ejemplo, si incluimos la línea

```
\def\lgem{\discretionary{1-}{1}{\hbox{1$\cdot$1}}}
```

podremos usar el comando `\lgem` para escribir una l geminada catalana. Por ejemplo, `pe\lgem \'{i}cula` produce *pel·lícula*. La definición es complicada porque prevé la posibilidad de que la palabra deba ser partida por las *eles*, en cuyo caso el punto se sustituye automáticamente por un guión.

Las comillas «francesas» pueden definirse mediante las instrucciones

```
\def\cfi{\leavevmode\raise.2ex\hbox{$\scriptscriptstyle\ll$}}  
\def\cfd{\leavevmode\raise.2ex\hbox{$\scriptscriptstyle\gg$}}
```

Abreviaturas como 1^{er} pueden obtenerse mediante `1\ser` previa definición en el preámbulo del comando `\ser` como

```
\def\ser{\leavevmode\raise.585ex\hbox{\small er}}
```

Tamaños, estilos y tipos de letra El aspecto de un carácter depende de la combinación de varias características que se pueden fijar independientemente:

Forma La forma de un carácter puede ser Recta (Upright), *Itálica (Italic)*, *Inclinada (Slanted)* o MAYÚSCULAS PEQUEÑAS (SMALL CAPS).

Se consiguen con las declaraciones `\upshape`, `\itshape`, `\slshape` y `\scshape` respectivamente.

Serie La serie puede ser media (medium) o **negrita (boldface)**. Se obtienen con las declaraciones `\mdseries` y `\bfseries`

Familia Las familias posibles son `Roman`, `Sans Serif` y `Typewriter`, que se obtienen con las declaraciones `\rmfamily`, `\sffamily` y `\ttfamily`.

Tamaño Los distintos tamaños disponibles son

`tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`,
LARGE, `huge`, `Huge`.

Observamos que con la opción `12pt` los tamaños `\huge` y `\Huge` son iguales. No sucede lo mismo a `10pt`.

Las declaración `\normalfont` reestablece las características por defecto excepto el tamaño.

Una declaración afecta al texto que sigue hasta donde se cierre una llave abierta previamente. Por ejemplo, para poner una palabra en negrita escribimos

`\bfseries` **negrita**

Las instrucciones T_EX estándar para los estilos son en realidad `\it` para itálica, `\sl` para slanted, `\sc` para small caps, `\bf` para boldface, `\rm` para roman, `\sf` para sans serif y `\tt` para typewriter. Normalmente podemos usar estas abreviaturas, pero las declaraciones L^AT_EX 2_ε tienen la ventaja de que son combinables, por ejemplo, `{\bf \sf hola}` produce **hola**, mientras que `{\bfseries \sffamily hola}` produce **hola**.

Si queremos cambiar el estilo de un fragmento largo de texto puede ser más claro y cómodo usar un entorno:

```
\begin{itshape}    (o simplemente it)
```

Texto que aparecerá en cursiva

```
\end{itshape}
```

Por último, podemos subrayar una porción de texto mediante el comando `\underline{texto}`.

Justificado Normalmente T_EX justifica el texto por ambos lados, partiendo las palabras de la forma más adecuada. Si queremos el texto centrado, alineado por la izquierda o alineado por la derecha usamos respectivamente las declaraciones `\centering`, `\raggedright` y `\raggedleft`. Alternativamente podemos usar los entornos `center`, `flushleft` y `flushright`, es decir, para alinear por la izquierda una porción de texto podemos escribir

```
\begin{flushleft}
```

Texto que aparecerá alineado por la izquierda

```
\end{flushleft}
```

En cualquier contexto podemos forzar el final de una línea (sin justificarla) mediante `\` Opcionalmente podemos indicar el espacio hasta la línea siguiente: `\\[3cm]`.

NOTA Una declaración como `\raggedleft` sólo afectará a los párrafos enteros que estén contenidos entre llaves que la rodeen. Las llaves han de contener a la línea en blanco que termina el párrafo.

2 Escritura básica de matemáticas

Existen dos modos matemáticos: *text* y *display*. El primero se usa para símbolos matemáticos insertados entre texto y el segundo para fórmulas centradas separadas del texto. Unas mismas instrucciones pueden dar resultados diferentes según el modo. Por ejemplo $\sum_{n=1}^{\infty} \frac{1}{2^n} = 1$ está en modo *text*, mientras que

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

está en modo *display*. El primer ejemplo se obtiene con

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

Los signos \$ marcan el inicio y el fin del modo matemático *text*. Si en lugar de \$ ponemos \$\$ obtenemos el segundo ejemplo.

Debemos escribir en modo matemático cualquier signo matemático, aunque sea una sola letra. Por ejemplo, para obtener

Diremos que un elemento p de un dominio íntegro D es *irreducible* si no es nulo ni unitario y no tiene más divisores que sus asociados y las unidades.

escribimos

Diremos que un elemento p de un dominio \mathcal{D} es *irreducible* si no es nulo ni unitario y no tiene más divisores que sus asociados y las unidades.

Algunas instrucciones cambian de significado en modo matemático. Por ejemplo a' da a' en modo horizontal y da a' en modo matemático (apóstrofo en el primer caso, prima en el segundo).

Espacios Los espacios en blanco carecen de valor en modo matemático, de modo que añadir o quitar un espacio en el texto fuente nunca cambia el resultado. En cada caso el T_EX determina la distribución más conveniente de los espacios. No obstante hay ocasiones en que conviene aumentar o reducir los espacios, para lo cual contamos con los comandos $\!$, que da un espacio negativo, $\,$, que da un pequeño espacio adicional y $_$, que da un espacio normal. Para añadir espacios mayores de forma estándar conviene usar \quad y \qquad . Todos sirven también en modo horizontal excepto el espacio negativo. Por ejemplo, para escribir

$$dy = 2x dx$$

conviene poner $dy=2x\,dx$, de modo que dx quede un poco más separado.

Subíndices y superíndices Para crear subíndices se usa `_` y para superíndices `^`. Por ejemplo `\$a_{ij} = 3^i - b_{i_j}\$` produce $a_{ij} = 3^i - b_{i_j}$. No hay problema en poner al mismo tiempo un subíndice y un superíndice: `\$a_i^{j+1}\$` da a_i^{j+1}

Fracciones y raíces Las fracciones se consiguen con el comando

`\frac{numerador}{denominador}`.

Por ejemplo, si escribimos `\$1+\frac{1}{1+\frac{1}{5}}\$` obtenemos

$$1 + \frac{1}{1 + \frac{1}{5}}$$

Para las raíces tenemos el comando `\sqrt[índice]{radicando}`. El índice va entre corchetes porque es opcional. Por ejemplo, `\sqrt[5]{-1}` produce $\sqrt[5]{-1}$. La instrucción `\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}` produce

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Subrayado y similares El comando `\underline` vale también en modo matemático, pero ahora tenemos además `\overline`, que produce una barra sobre el texto, como en $\overline{A + \overline{B}}$, que sale de `\overline{A+\overline{B}}`.

Dos comandos similares son `\underbrace` y `\overbrace`, que producen llaves bajo o sobre el texto, como en $\overbrace{A+B}$. Un subíndice tras un `\underbrace` o un superíndice tras un `\overbrace` aparece como una etiqueta en la llave, como en $(\underbrace{a, \dots, a}_{15})$, que se obtiene con

`\(\, \underbrace{a, \ \ldots, a}_{15} \, \)`.

Símbolos matemáticos Tenemos a nuestra disposición varias decenas de signos matemáticos estándar. `TEX` los clasifica en diversos tipos según el espacio que debe dejar entre ellos.

Normales Son las letras y números que pueden escribirse también en modo horizontal, como `\$a\$`.

Ordinarios Son como los normales, pero sólo existen en modo matemático. Los más importantes son las letras griegas:

α	<code>\alpha</code>	θ	<code>\theta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>						
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Destacamos también los siguientes:

\emptyset	<code>\emptyset</code>	∇	<code>\nabla</code>	\forall	<code>\forall</code>
\exists	<code>\exists</code>	∞	<code>\infty</code>	∂	<code>\partial</code>
				\neg	<code>\neg</code>

Si queremos poner un acento, tilde o lo que sea a una i o una j tendremos que usar `\imath` y `\jmath`, que producen i y j .

Desde el teclado podemos entrar directamente |.

Operadores binarios Son signos que han de aparecer cercanos a los símbolos anterior y posterior. Destacamos los siguientes:

\pm	<code>\pm</code>	\mp	<code>\mp</code>	\times	<code>\times</code>	\div	<code>\div</code>
\circ	<code>\circ</code>	\cdot	<code>\cdot</code>	\cap	<code>\cap</code>	\cup	<code>\cup</code>
\setminus	<code>\setminus</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>	$*$	<code>\ast</code>

Notemos que para escribir $2 \cdot 3 = 6$ no hemos de poner `$2.3=6$`, pues resultaría $2.3 = 6$, sino `$2\cdot 3=6$`.

Relaciones Son símbolos que han de quedar algo separados de los que les rodean. Los más importantes son $+$, $-$, $/$, $<$, $>$, $=$, que se pueden entrar directamente desde el teclado, más los que producen los comandos siguientes:

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\sim	<code>\sim</code>
\simeq	<code>\simeq</code>	$ $	<code> </code>	\parallel	<code>\parallel</code>	\subset	<code>\subset</code>
\subseteq	<code>\subseteq</code>	\supset	<code>\supset</code>	\supseteq	<code>\supseteq</code>	\approx	<code>\approx</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code>

Observamos que `\neq` y `\notin` producen las negaciones de $=$ y \in . Para las demás relaciones, podemos conseguir su negación anteponiendo `\not`. Por ejemplo, `$a\not\equiv b$` produce $a \neq b$.

No hay que confundir los símbolos ordinarios $|$ y $\backslash|$ con las relaciones \backslashmid y \backslashparallel . Producen el mismo símbolo, pero el espaciado es distinto. Por ejemplo, para tener $|a + b| \leq |a| + |b|$ hemos de escribir $\$|a+b|\backslashleq|a|+|b|$, y no

$$\$\mid a+b\mid\leq\mid a\mid+\mid b\mid$,$$

que daría lugar a $|a + b| \leq |a| + |b|$.

Entre las relaciones se encuentran también las flechas. Las más importantes son:

\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>
\searrow	<code>\searrow</code>	\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>

Operadores Son los símbolos que deben unirse al símbolo que sigue. Entre ellos se encuentran las funciones matemáticas como

`\arccos` `\cos` `\sec` `\ln` `\lim` `\max` `\sup`
`\arcsin` `\sin` `\csc` `\log` `\ker` `\min` `\inf`
`\arctan` `\tan` `\cot` `\exp` `\det` `\dim` `\arg`

También se incluyen aquí los llamados operadores grandes, que cambian de tamaño según el modo `text/display`. Entre ellos están:

Σ `\sum` Π `\prod` \int `\int` \oint `\oint`
 \bigcup `\bigcup` \bigcap `\bigcap` \bigoplus `\bigoplus` \bigotimes `\bigotimes`

Para especificar, por ejemplo, los límites de una integral definida se usan los comandos de subíndice y superíndice. Por ejemplo, si escribimos

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

obtenemos

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

Las sumas e integrales son casos típicos donde conviene usar espacios negativos. Lo mismo vale por ejemplo con los límites. Para obtener

$$\lim_{x \rightarrow x_0} f(x)$$

basta escribir $\lim_{x \rightarrow x_0} f(x)$. Sin embargo, en modo text el resultado es $\lim_{x \rightarrow x_0} f(x)$. En general T_EX trata de evitar que de una línea de texto sobresalgan cosas. Si pese a todo queremos forzar que la flecha quede debajo podemos escribir

$$\lim_{\limits_{x \rightarrow x_0}} f(x).$$

En general la instrucción `\limits` delante de un subíndice o un superíndice y después de un operador hace que éste aparezca debajo o arriba del signo anterior en lugar de a la derecha. Por ejemplo, podemos conseguir $\frac{c}{b}$ mediante `\mathop{a}\limits_b^c`. El primer comando convierte en operador a la a . Si no lo ponemos obtenemos un error. Recíprocamente, `\nolimits` hace que los subíndices y los superíndices se comporten del modo habitual.

Puntuación Además de los signos de puntuación del modo horizontal, los modos matemáticos admiten otras variantes de puntos suspensivos: los del modo horizontal eran `\ldots`, que produce puntos suspensivos a la altura de la línea, pero ahora tenemos también `\cdots`, que produce puntos suspensivos centrados. Comparar

$$a_1 + \dots + a_n \quad \text{con} \quad a_1 + \cdots + a_n.$$

También tenemos puntos suspensivos verticales \vdots y diagonales \ddots , que se obtienen con `\vdots` y `\ddots`.

Los signos de puntuación se tratan a modo de separadores, como en (x, y) . Si queremos escribir algo como 2,3 y escribimos `$2,3$` obtenemos 2,3, que no es lo mismo. Es mejor `$2\mathnormal,3$`, con lo que la coma se transforma en un signo normal y no produce separación extra.

Delimitadores Son los símbolos que actúan a modo de paréntesis. Todos tienen una versión izquierda y una versión derecha. Los más importantes son (en su forma izquierda) `(`, `[`, `{`, `|`, `||`, `<`. Todos se introducen directamente desde el teclado excepto `{`, que se obtiene con `\{`, y `||`, que lo produce `\|` y `<`, que lo produce `\langle`.

Las instrucciones `\left` y `\right` precediendo a dos delimitadores hacen que su tamaño se ajuste al texto que encierran. Un `\left` no balanceado con un `\right` o viceversa provoca un error. Por ejemplo, si escribimos

$$\left(\frac{e^x + e^{-x}}{2}\right)^2$$

obtenemos

$$\left(\frac{e^x + e^{-x}}{2}\right)^2.$$

También hemos de usar `\left` y `\right` cuando queramos usar un delimitador izquierdo a la derecha o viceversa. Por ejemplo, para escribir $a \in]0, +\infty[$ conviene poner `$a\in\left]0,+\infty\right[$`, para que el espaciado sea correcto. En otro caso saldría $a \in]0, +\infty[$.

Acentos Los modos matemáticos admiten más acentos que el modo horizontal, y los acentos comunes se obtienen con comandos distintos. Éstos son:

\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>				

Los comandos `\widehat` y `\widetilde` producen versiones “anchas” de estos dos acentos, como en $\widehat{a+b}$.

Texto entre matemáticas Para introducir pequeñas porciones de texto entre expresiones matemáticas podemos usar `\mbox{texto}`. Por ejemplo,

$$\{x \in A \mid x > y \text{ para todo } y \in B\},$$

se obtiene con

```
$$\{x\in A \mid x>y \mbox{ para todo } y\in B\}$$
```

Definición de operadores y relaciones Si hemos de usar con frecuencia una función o relación no estándar conviene definirla en el preámbulo indicando su categoría para garantizar el espaciado correcto. Por ejemplo, para definir una función `sen` (en lugar de `sin`, que da L^AT_EX por defecto) deberemos poner en el preámbulo

```
\def\sen{\mathop{\mbox{\normalfont sen}}\nolimits}
```

Hemos de poner `\nolimits` para evitar que al escribir `\sen^2 x` obtenemos $\text{sen}^2 x$. En cambio, si queremos definir un operador máx (con acento) deberemos poner

```
\def\max{\mathop{\mbox{\normalfont m'ax}}\limits}
```

para permitir construcciones como $\max_{i \in I} a_i$.

En general, los comandos para especificar la categoría de un símbolo son `\mathbin` (operador binario), `\mathop` (operador), `\mathrel` (relación), `\mathord` (ordinario), `\mathnormal` (normal). Es raro que necesitemos definir un delimitador, un acento o un signo de puntuación.

Tamaños, estilos y tipos de letra En modo matemático las letras aparecen por defecto en cursiva, pero podemos elegir cualquiera de los estilos de los modos no matemáticos con las mismas declaraciones `\rm`, `\it`, `\bf`, `\ss`, `\tt`, aunque si queremos combinarlas deberemos usar las instrucciones completas y no sus abreviaturas, que en modo matemático son `\mathrm`, `\mathit`, etc.

En la práctica estas declaraciones sólo se aplican cuando queremos modificar el estilo de un símbolo concreto, pues para escribir texto es preferible salir del modo matemático o usar `\mbox{texto}`.

El modo matemático tiene un estilo adicional, llamado *caligráfico*, que vale sólo para letras mayúsculas. Se obtiene con `\mathcal{mayúsculas}`, aunque puede abreviarse en `\cal`. Por ejemplo, `\mathcal{D}(X)` produce $\mathcal{D}(X)$.

Todas estas instrucciones afectan sólo a letras, números y letras griegas mayúsculas. Por ejemplo, si escribimos `\mathbf{2+2=4}` obtenemos $\mathbf{2 + 2 = 4}$, donde los signos no aparecen en negrita. Si queremos una fórmula entera en negrita hemos de usar `\boldmath $ texto $`. Por ejemplo, `\boldmath $2+2=4$` produce $\mathbf{2 + 2 = 4}$. Es importante que `\boldmath` ha de usarse fuera del modo matemático, y su efecto es que todo el texto en modo matemático que aparezca en los límites de la declaración se ponga en negrita. Si sólo queremos un símbolo en negrita dentro de una fórmula hemos de usar `\mbox`. Por ejemplo, `$x+\mbox{\boldmath ∇}f$` produce $x + \nabla f$.

Junto a los estilos matemáticos *display* y *text* existen otros dos, llamados *script* y *scriptscript*, en los que \LaTeX entra automáticamente cuando escribe subíndices y subsubíndices, respectivamente, aunque también se usan en otros contextos, como en fracciones dentro de fracciones. Así, si escribimos

```
$$\sqrt{5} = 1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{\ddots}}}}$$
```

obtenemos

$$\sqrt{5} = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{\ddots}}}}$$

donde \LaTeX ha ido reduciendo el estilo paulatinamente, pero el resultado no es satisfactorio. En este caso queda mejor si forzamos a que todos los términos de la fracción continua mantengan el estilo *display*, mediante

```
$$\sqrt{5} = 1+\frac{1}{\displaystyle 1+\frac{1}{\displaystyle 1+\frac{1}{\displaystyle 1+\frac{1}{\ddots}}}}$$
```

que produce

$$\sqrt{5} = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{\ddots}}}}$$

En general, podemos forzar cualquiera de los cuatro estilos mediante las cuatro declaraciones

`\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle`.

Ya hemos visto las instrucciones `\left` y `\right`, que ajustan el tamaño adecuado de un delimitador. A veces \LaTeX no sabe calcular el tamaño adecuado, y entonces hemos de hacerlo directamente mediante las instrucciones \TeX `\bigl`, `\Bigl`, `\biggl`, `\Biggl`, que producen delimitadores izquierdos de distintos tamaños, y las correspondientes `\bigr`, etc., que producen los correspondientes delimitadores derechos. Así podemos obtener, por ejemplo $\left| |a| + |b| \right|$. Con la opción *12pt* algunas de estas instrucciones producen el mismo efecto.

Fuentes AMS La American Mathematical Society ha diseñado unas fuentes adicionales que pueden ser usadas desde el \LaTeX . Para disponer de ellas hemos de poner en el preámbulo la instrucción `\usepackage{amsfonts}`. Con ello tenemos a nuestra disposición las fuentes góticas `\mathfrak`, como en **Goethe** y las fuentes de “pizarra” (sólo mayúsculas) como \mathbb{R} , \mathbb{D} , etc., que se obtienen con `\mathbb{R}`, etc.

3 El formato artículo

La clase “article” La clase “article” se selecciona con la instrucción
`\documentclass[opciones]{article}`.

Las opciones más importantes que podemos especificar son las siguientes:

10pt, 11pt, 12pt Determinan el tamaño de letra del documento. La opción por defecto es 10pt, por lo que nunca es necesario especificarla.

twoside Hace que los márgenes de las páginas pares e impares sean distintos, de modo que al imprimir a dos caras se superpongan correctamente.

twocolum Escribe a dos columnas

a4paper Selecciona el tamaño de papel DINA4. Otras opciones son letterpaper, a5paper, etc.

landscape Apaisado

draft Borrador, marca las líneas demasiado largas mediante cajas negras.

titlepage Hace que el título y el abstract queden en una página aparte.

leqno, fleqn La primera numera las ecuaciones por la izquierda, la segunda alinea por la izquierda las ecuaciones en modo display.

Si se indican varias opciones éstas deben ir separadas por comas.

El título El título, autor, etc. del artículo se indica en el preámbulo mediante las declaraciones

```
\title{título}
\author{autor}
\date{fecha}
```

Para que aparezcan impresos debemos usar la orden `\maketitle` después de `\begin{document}`. Si no queremos que aparezca ninguna fecha habremos de especificar `\date{}`, o de lo contrario aparecerá la fecha de hoy. Si el título es largo y queremos cortarlo por algún punto en concreto podemos usar `\`.

Si hay varios autores hemos de separarlos mediante `\and`. Si uno o varios de los autores deben llevar una nota al pie de agradecimientos, reconocimiento de becas o proyectos etc. ésta se incluye con la instrucción `\thanks{agradecimientos}`.

Por ejemplo una declaración de autores puede ser:

```
\author{J. L'opez\thanks{Financiado en parte por...} \and
J. Garc'\{i}a\thanks{El segundo autor desea agradecer...}}
```

El abstract El abstract se escribe entre los comandos `\begin{abstract}` y `\end{abstract}`. La palabra “abstract” la pone L^AT_EX automáticamente. Si queremos poner otra cosa, por ejemplo “resumen”, incluimos en el preámbulo la instrucción

```
\renewcommand{\abstractname}{resumen}
```

Secciones Las distintas secciones del artículo se especifican con los comandos siguientes:

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

Por ejemplo, el principio de esta sección se ha obtenido con la instrucción

```
\section{El formato art\’{\i}culo}
```

El número 3 lo pone L^AT_EX automáticamente. Ahora estamos dentro del párrafo “secciones”, que ha sido creado con `\paragraph{secciones}`. No es necesario haber definido una subsección o una subsubsección para definir un párrafo.

En general, L^AT_EX numera automáticamente hasta las subsubsecciones. Cuando queramos hacer referencia a uno de estos números no debemos hacerlo directamente, sino que al lado del comando que da inicio a la (sub-sub)sección deberemos poner la instrucción `\label{etiqueta}` y hacer la referencia en la forma siguiente:

```
... como hemos visto en \ref{etiqueta}, se cumple...
```

De este modo la referencia seguirá siendo correcta aunque intercalemos o suprimamos otras secciones.

Podemos poner una etiqueta `\label{lo que sea}` en cualquier punto del texto. Si nos referimos a ella mediante `\ref{lo que sea}` obtendremos el número de la sección de menor nivel que contenga la etiqueta. Si nos referimos a ella mediante `\pageref{lo que sea}` obtendremos el número de página donde aparece la etiqueta.

Podemos alterar la numeración automática de las secciones. Por ejemplo, si queremos que después de la subsección 3.5 venga la subsección 3.7 (por ejemplo, porque la sección 3.6 va a escribirla un amigo nuestro) antes de iniciar la subsección 3.7 escribiremos `\setcounter{subsection}{6}`, con lo que L^AT_EX “se creará” que ya está en la subsección 6 y al iniciar una nueva le asignará el número 7.

Bibliografía La bibliografía al final del artículo se especifica de la forma siguiente:

```
\begin{thebibliography}{XXX}
```

```
\bibitem{Cer} M. de Cervantes, el Ingenioso Hidalgo...
```

```

\bibitem{Gon} L. de G\'ongora, Soledades.
\bibitem{Que} F. de Quevedo, Historia del Busc3n llamado...
\end{thebibliography}

```

El argumento XXX es una porci3n de texto de longitud mayor o igual que la mayor de las etiquetas que se asigne a las referencias. El resultado es

References

- [1] M. de Cervantes, el Ingenioso Hidalgo...
- [2] L. de G3ngora, Soledades.
- [3] F. de Quevedo, Historia del Busc3n llamado...

Si queremos cambiar la palabra “References” por otra, por ejemplo “Bibliograf3a”, escribiremos

```
\renewcommand{\refname}{Bibliograf\'\{i}a}
```

Para referirse a una obra, por ejemplo a la de Cervantes, usaremos la instrucci3n `\cite{Cer}`, o incluso, `\cite[pp.\ 20--21]{Cer}`, lo cual produce [1, pp. 20–21]. Si no queremos que las referencias aparezcan numeradas, sino con etiquetas, s3lo hemos de indicarlas entre corchetes al lado de cada `\bibitem`, por ejemplo

```
\bibitem[G72]{Gon} L. de G\'ongora, Soledades.
```

De este modo, tanto en la bibliograf3a como en las referencias que hagamos, figurar3a la etiqueta G72.

Referencias a ecuaciones Para que una ecuaci3n aparezca numerada, en lugar de escribirla entre d3lares dobles usaremos el entorno

```

\begin{equation}\label{suma}
2+2=5
\end{equation}

```

El resultado es

$$2 + 2 = 5 \tag{1}$$

de modo que al escribir

la ecuaci3n (`\ref{suma}`) no es exacta.

obtenemos “la ecuaci3n (1) no es exacta.”

Numeración de páginas \LaTeX numera las páginas automáticamente. El número de página se guarda en la variable “page”, que podemos alterar cuando queramos. Por ejemplo, si iniciamos el documento con la instrucción `\setcounter{page}{27}`, la primera página tendrá el número 27.

Hay cuatro estilos de página predefinidos. Con `\pagestyle{empty}` no aparecen números de página. Con `\pagestyle{plain}` obtenemos los números de página al pie y nada en la cabecera (ésta es la opción por defecto, por lo que no hemos de especificarla salvo que la hayamos cambiado antes). Con `\pagestyle{headings}` obtenemos el número de página en el pie y una cabecera con el nombre de la sección (o con la sección en las páginas pares y la subsección en las impares si hemos indicado la opción `twoside` en el comando `\documentclass`). Finalmente, está la opción `\pagestyle{myheadings}`, que pone el número de página al pie y en la cabecera pone el texto especificado por los comandos

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

```
\markright{cabecera derecha}
```

Si no hemos especificado la opción `twoside`, todas las páginas se consideran derechas. Podemos cambiar el estilo de la página en curso mediante `\thispagestyle`. Por ejemplo, si queremos que la página en curso no tenga número hemos de escribir `\thispagestyle{empty}`.

Teoremas Para numerar automáticamente los teoremas hemos de definir tantos entornos como tipos de “teoremas” vayamos a usar, es decir, teoremas, proposiciones, corolarios, definiciones, etc. El formato es el siguiente:

```
\newtheorem{teo}{Teorema}[section]
```

Esta orden (que conviene situar en el preámbulo) define un entorno llamado “teo”, de modo que cuando escribamos

```
\begin{teo} Hay infinitos n\’umeros primos
\end{teo}
```

obtenemos

Teorema 3.1 *Hay infinitos números primos*

Así pues, “teo” es el nombre del entorno que hemos de poner tras el `\begin` y el `\end`, “Teorema” es la palabra que aparece en el texto compilado y el argumento opcional “section” hace que el número de cada teorema

aparezca precedido por el número de la sección actual (y vuelve a 1 al cambiar de sección). Si queremos definir un entorno “Corolario” de modo que la numeración de los corolarios sea correlativa a la de los teoremas escribiremos en el preámbulo

```
\newtheorem{cor}[teo]{Corolario}
```

De este modo, al escribir

```
\begin{cor} El conjunto de los primos no est\'a acotado.
\end{cor}
```

obtenemos

Corolario 3.2 *El conjunto de los primos no está acotado.*

Si hubiéramos puesto `\newtheorem{cor}{Corolario}` habríamos obtenido la numeración **Corolario 1**, independiente de la numeración de los teoremas.

Observar que el texto de un entorno “teorema” aparece en itálica. Si no lo queremos así usamos `\rm`. Si queremos poner nombre a un teorema lo hacemos entre corchetes. Por ejemplo,

```
\begin{teo}[Euclides] Hay infinitos n\'umeros primos
\end{teo}
```

produce

Teorema 3.3 (Euclides) *Hay infinitos números primos*

Una etiqueta `\label{lo que sea}` tras un `\begin{teo}` permite referirse al teorema mediante `\ref{lo que sea}`. Podemos alterar la numeración de los teoremas con `\setcounter`. Por ejemplo, si hacemos

```
\setcounter{teo}{12}
```

y volvemos a escribir el teorema anterior obtenemos

Teorema 3.13 (Euclides) *Hay infinitos números primos*

Citas y notas al pie Para citar un texto existen los entornos “quote” (para citas de un solo párrafo) y “quotation” (para varios párrafos). Por ejemplo,

```
\begin{quote}
```

```
El buen cristiano debe estar precavido frente a los ...
```

```
\end{quote}
```

produce

El buen cristiano debe estar precavido frente a los matemáticos y todos aquellos que hacen profecías vacías. Existe el peligro de que los matemáticos hayan hecho un pacto con el diablo para oscurecer el espíritu y confinar al hombre en el infierno.¹

SAN AGUSTÍN, De genesi ad Litteram, libro II, xviii, 37

La diferencia entre “quote” y “quotation” es que el primero no sangra los párrafos y los separa un poco más de lo habitual.

La nota al pie ha sido obtenida mediante:

```
... en el infierno.\footnote{Conviene aclarar que ...}
```

¹Conviene aclarar que San Agustín llama matemáticos a los astrólogos.

4 Matrices y tablas

Nos ocupamos ahora de las distintas formas de disponer texto en filas y columnas. Los entornos `array` y `tabular` son idénticos excepto por que el primero se usa en modo matemático y el segundo en modo normal. Comenzaremos con `array`.

Matrices Consideremos la matriz

$$A = \begin{pmatrix} 1.234 & -5 & x & 0.234 \\ 280 & 0 & x^2 + 2 & 1.22 \end{pmatrix}$$

Ha sido obtenida con el texto fuente siguiente:

```
$$
A = \left(
\begin{array}{rccl}
1.234 & -5 & x & 0.234 \\
280 & 0 & x^2+2 & 1.22
\end{array}
\right)
$$
```

Al poner `\begin{array}{rccl}` indicamos que vamos a escribir una matriz con cuatro columnas, de las cuales la primera estará justificada por la derecha, las otras dos por el centro y la última por la izquierda. Dentro de cada fila, las distintas entradas se separan mediante signos `&` y el final de cada fila se indica mediante `\\`. Notemos que `array` no pone los paréntesis, sino que éstos los ponemos nosotros antes y después de la matriz.

Las estructuras matriciales pueden usarse para conseguir fórmulas que no representan realmente matrices. Por ejemplo, si escribimos

```
$$
f(x) = \left\{
\begin{array}{cl}
x^2+y & \text{si } x > y \\
y^3 & \text{si } x \leq y
\end{array}
\right.
$$
```

obtenemos

$$f(x) = \begin{cases} x^2 + y & \text{si } x > y \\ y^3 & \text{si } x \leq y \end{cases}$$

Notemos que el delimitador izquierdo `\left\{` debe equilibrarse con un delimitador derecho. Cuando no queremos que aparezca ninguno escribimos `\right.` en el lugar donde debería aparecer. Otro ejemplo de estructura matricial es

$$f: \mathbb{R}^2 \longrightarrow \mathbb{R} \\ (x, y) \mapsto x^2 + y$$

obtenida mediante

```

$$
\begin{array}{rccl}
f:&\mathbb{R}^2&\longrightarrow&\mathbb{R} \\
&(x, y)&\mapsto&x^2 + y
\end{array}
$$

```

Si entre las “erres”, “ces” y “eles” que determinan las columnas escribimos `@{‘algo’}`, el texto que figure en ‘algo’ se escribirá entre las columnas correspondientes y si hay comandos, éstos se ejecutarán en cada fila. Por ejemplo, si consideramos que las columnas del ejemplo anterior están demasiado separadas podemos insertar espacios negativos entre ellas para obtener

$$f: \mathbb{R}^2 \longrightarrow \mathbb{R} \\ (x, y) \mapsto x^2 + y$$

El texto fuente es

```

$$
\begin{array}{r@{\hspace{-2pt}}c@{\hspace{-4pt}} \\
c@{\hspace{4pt}}l}
f:&\mathbb{R}^2&\longrightarrow&\mathbb{R} \\
&(x, y)\ \ \ &\mapsto&x^2 + y
\end{array}
$$

```

Hemos dejado un par de espacios tras (x, y) para desplazarlo hacia la derecha respecto al \mathbb{R}^2 que tiene encima. Puede parecer extraño que un espacio positivo de cuatro puntos acerque las dos últimas columnas. La razón es que `LATEX` añade un cierto espacio entre las columnas de una matriz, pero al poner una `@` este espacio se suprime. Por lo tanto `@{\hspace{4pt}}` resta el espacio extra y suma 4 puntos, y el resultado es negativo. Un simple `@{}` ya reduce el espacio entre columnas.

Una forma de alterar el espaciado de varias columnas a un tiempo es poner `@{\extracolsep{3mm}}`. El efecto es un espacio extra de 3mm entre TODAS las columnas que siguen (salvo que pongamos otra instrucción de este tipo más adelante). Posteriores `@` no anulan este espacio extra.

Recordemos que la distancia entre filas se puede alterar con `\\[2mm]`.

A menudo queremos poner líneas verticales entre las columnas de una matriz. En principio deberíamos poner `@{|}`, pero dada la frecuencia de uso puede abreviarse en `|`. Por ejemplo, si escribimos

```

$$
\left(
\begin{array}{c|ccc}
a&0&\cdots &0\\
\hline
0&1 & & \\
\vdots & &\ddots & \\
0 & & & 1
\end{array}
\right)
$$

```

obtenemos

$$\left(\begin{array}{c|ccc} a & 0 & \cdots & 0 \\ \hline 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{array} \right)$$

Notar que `\hline` produce una línea horizontal. Debe ponerse antes de la primera fila o después de `\\`. Si ponemos `||` entre dos columnas obtendremos una doble barra vertical, también podemos poner varios `\hline` seguidos.

Tablas El entorno `tabular` es idéntico a `array` salvo que se usa en modo normal (no matemático). Todo lo dicho anteriormente para `array` vale aquí y todo lo nuevo que diremos aquí vale también para `array`.

Podemos unir varias columnas en una mediante `\multicolumn`. Consideremos por ejemplo la tabla siguiente:

Producto	Precio
	Mín-Máx
A	100-300
B	1.230-2.000
C	3.000-5.000

Observamos que la palabra “Precio” se extiende sobre la segunda y la tercera columna. Para conseguirlo, tras dejar en blanco la primera posición de la primera fila escribimos `\multicolumn{2}{c|}{Precio}`, cuyo efecto es que la palabra “precio” ocupe las dos columnas siguientes en posición centrada y con una línea vertical a la derecha.

Entre la primera y la segunda fila hay una línea horizontal que se extiende sólo entre la segunda y la tercera columna. Esto se consigue con `\cline{2-3}`.

El texto fuente completo es:

```
\begin{center}
\begin{tabular}{|c|r@{--}l|}
\hline
&\multicolumn{2}{c|}{Precio}\\
\cline{2-3}
Producto &Mín &Máx & \\
\hline
A &&100 && 300 & \\
B &&1.230 && 2.000 & \\
C &&3.000 && 5.000 & \\
\hline
\end{tabular}
\end{center}
```

Podemos usar `\multicolumn{1}{r}{algo}` para escribir “algo” en una entrada pero con un justificado distinto del correspondiente a la columna.

A veces queremos introducir texto en una entrada de una tabla pero no queremos que la anchura de la columna sea la del texto, sino que éste se parta en líneas. Para ello, al declarar la columna correspondiente no ponemos `r`, `l` o `c`, sino `p{4cm}`, donde la longitud indica la anchura de la columna. Por ejemplo, la tabla:

Producto	Precio	Observaciones
	Mín–Máx	
A	100–300	Es el más barato de todos.
B	1.230–2.000	Presenta la mejor relación calidad-precio.
C	3.000–5.000	Producto de super-extra-mega-lujo.

se consigue con

```

\begin{center}
\begin{tabular}{|c|r@{--}l||p{4cm}||}
\hline
&\multicolumn{2}{c||}{Precio}& \\\
\cline{2-3}
Producto &Mín&Máx&\multicolumn{1}{c|}{Observaciones}\\\
\hline
A&100 & 300 & Es el más barato de todos.\\
B&1.230 & 2.000& Presenta la mejor relación calidad-precio.\\
C&3.000 & 5.000& Producto de super-extra-mega-lujo.\\
\hline
\end{tabular}
\end{center}

```

A veces queremos que una tabla tenga una anchura prefijada, por ejemplo la de la página. Esto se consigue con el entorno `\tabular*`, que es idéntico a `tabular` salvo por que admite como argumento opcional la anchura de la tabla. Si ponemos `\textwidth` obtenemos una tabla cuyo ancho es el de la página. En tal caso hemos de indicar entre qué columnas queremos que se inserte el espacio extra para cuadrar la tabla. Esto se consigue con `@{\extracolsep{\fill}}`. Por ejemplo, la tabla

Año	A	B	C	D	E	Mín-Máx
1996	1.000	250	400	2.130	300	250-2.130
1997	1.230	200	800	2.000	500	200-2.000
1998	1.600	220	700	2.100	1.500	220-2.100

tiene el ancho de la página y éste se consigue rellenando el espacio entre las columnas 2-3, 3-4, 4-5 y 5-6. El texto fuente es

```

\noindent
\begin{tabular*}{\textwidth}
{r|c@{\extracolsep{\fill}}cccc|@{\extracolsep{0mm}\ }r@{--}l||}
Año&A&B&C&D&E&Mín&Máx\\\
\hline
1996&1.000&250&400&2.130&300&250&2.130\\
1997&1.230&200&800&2.000&500&200&2.000\\
1998&1.600&220&700&2.100&1.500&220&2.100\\
\hline
\end{tabular*}

```

Alineación de ecuaciones Para alinear ecuaciones podemos usar una estructura matricial. No obstante, en el caso más simple conviene usar los

entornos `eqnarray` y `eqnarray*`. Ambos son equivalentes a un entorno `\begin{array}{rcl}`, con la peculiaridad de que las columnas primera y tercera aparecen en estilo display mientras que la segunda en estilo texto. No hay que poner dólares para usarlos. Por ejemplo, si escribimos

```
\begin{eqnarray*}
x&=&y\\
x^2&=&xy\\
x^2-y^2&=&xy-y^2\\
(x+y)(x-y)&=&y(x-y)\\
x+y&=&y\\
2y&=&y\quad \mbox{(por la primera ecuación)}\\
2&=&1
\end{eqnarray*}
```

obtenemos

$$\begin{aligned}
 x &= y \\
 x^2 &= xy \\
 x^2 - y^2 &= xy - y^2 \\
 (x + y)(x - y) &= y(x - y) \\
 x + y &= y \\
 2y &= y \quad (\text{por la primera ecuación}) \\
 2 &= 1
 \end{aligned}$$

Si suprimimos los asteriscos obtenemos las ecuaciones numeradas:

$$\begin{aligned}
 x &= y & (1) \\
 x^2 &= xy & (2) \\
 x^2 - y^2 &= xy - y^2 & (3) \\
 (x + y)(x - y) &= y(x - y) & (4) \\
 x + y &= y & (5) \\
 2y &= y \quad (\text{por la primera ecuación}) & (6) \\
 2 &= 1 & (7)
 \end{aligned}$$

Si no queremos numerar alguna ecuación usamos `\nonumber`. Por ejemplo, mediante

```

\begin{eqnarray}
x&=&y\label{prime}\\
x^2&=&xy\nonumber\\
x^2-y^2&=&xy-y^2\nonumber\\
(x+y)(x-y)&=&y(x-y)\nonumber\\
x+y&=&y\nonumber\\
2y&=&y\quad \mbox{por (\ref{prime})}\nonumber\\
2&=&1\nonumber
\end{eqnarray}

```

obtenemos

$$\begin{aligned}
 x &= y && (1) \\
 x^2 &= xy \\
 x^2 - y^2 &= xy - y^2 \\
 (x + y)(x - y) &= y(x - y) \\
 x + y &= y \\
 2y &= y \quad \text{por (1)} \\
 2 &= 1
 \end{aligned}$$

Otras estructuras matriciales A menudo es útil `\shortstack`, que produce una tabla (en modo no matemático) de una sola columna. Por ejemplo, podemos obtener

$$\lim_{\substack{(x,y) \rightarrow (0,0) \\ y=mx}} f(x, y)$$

mediante

```

$$
\lim_{\shortstack{\scriptstyle (x, y)\rightarrow (0, 0) \\ \scriptstyle y = mx}} f(x, y)
$$

```

Notemos que con `\shortstack` volvemos a modo no matemático, por lo que hemos de poner dólares en cada columna y, más aún, hemos de pasar a tamaño de subíndice. Por defecto el texto aparece centrado, pero podemos escribir `\shortstack[1]{texto}` si lo queremos alineado por la izquierda o con una `r` si lo queremos por la derecha.

El macro básico para trabajar con $\text{T}_\text{E}_\text{X}$ es el llamado Plain $\text{T}_\text{E}_\text{X}$. El $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ incorpora algunas de sus instrucciones. Entre ellas figuran varias sobre estructuras matriciales. La sintaxis es completamente distinta. Por ejemplo,

Otra forma de conseguir matrices es con `\matrix`. La sintaxis es completamente distinta. Por ejemplo, podemos obtener una matriz con columnas centradas como

$$\begin{matrix} 3 & 52 & 300 \\ 41 & 2 & x^2 \end{matrix}$$

mediante `$$\matrix{3&52&300\cr 41&2&x^2}$$`. Si en lugar de `\matrix` usamos `\pmatrix` obtenemos los paréntesis:

$$\begin{pmatrix} 3 & 52 & 300 \\ 41 & 2 & x^2 \end{pmatrix}$$

Una opción más interesante es `\bordermatrix`, que produce una fila y una columna fuera de los paréntesis, como en

$$\begin{matrix} & & j & & k & & \\ & & & & & & \\ j & \left(\begin{array}{cccccc} 1 & & & & & \\ & \ddots & & & & \\ & & 0 & & 1 & \\ & & & \ddots & & \\ k & & 1 & & 0 & \\ & & & & & \ddots \\ & & & & & & 1 \end{array} \right) & & & & & \end{matrix},$$

que ha sido obtenida con

```

$$
\bordermatrix{& & j& & k\cr
&1\cr
& &\ddots\cr
j& &0& &1\cr
& & & \ddots\cr
k& &1& &0\cr
& & & & & \ddots\cr
& & & & & & 1}
$$

```

También es de ayuda el comando `\cases`, que produce una matriz de dos columnas, la primera en modo matemático y la segunda en modo no matemático, con una llave abierta a la izquierda, como en

$$f(x) = \begin{cases} x^2 & \text{si } x > 0 \\ 3 & \text{si } x \leq 0 \end{cases}$$

obtenida mediante

```

$$f(x) = \cases{x^2&si $x > 0\cr 3&si $x\leq 0}$$

```

El entorno `tabbing` Hay otro entorno que produce texto alineado en columnas de forma similar a los tabuladores de una máquina de escribir. Se entra y sale de él mediante `\begin{tabbing} ... \end{tabbing}` en modo no matemático. Se cambia de línea mediante `\\`. Cada vez que se sitúa un `\=` se fija un stop y el comando `\>` salta al siguiente stop. Por ejemplo,

```
Nombre Apellido Teléfono
Juan   Gómez   3141592
Pedro  Saenz    2718281
```

se consigue con

```
\begin{tabbing}
Nombre \= Apellido \= Tel\a'efono\\
Juan \> G\a'omez\> 3141592\\
Pedro \> Saenz\> 2718281
\end{tabbing}
```

Notemos que en este entorno hay que poner los acentos con `\a'` en lugar de `\'` debido a que `\'` tiene aquí otro significado.

Una línea acabada con `\kill` en lugar de `\\` no se imprime, pero los stops que contiene se conservan. Por ejemplo, si escribimos

```
\begin{tabbing}
Nombrexxxxx\= Apellidoxxxxxxxxxxx \= Tel\a'efono\kill
Nombre\> Apellido \> Tel\a'efono\\
Juan \> G\a'omez\> 3141592\\
Pedro \> Saenz\> 2718281
\end{tabbing}
```

obtenemos

```
Nombre      Apellido      Teléfono
Juan        Gómez         3141592
Pedro       Saenz         2718281
```

El comando `\+` hace que las líneas siguientes empiecen en el stop siguiente al previsto. Por ejemplo, mediante

```
\begin{tabbing}
margen xxxxx\=Nombrexxxxx\= Apellidoxxxxxxxxxxx
\= Tel\a'efono\+\kill
Nombre\> Apellido \> Tel\a'efono\\
Juan \> G\a'omez\> 3141592\\
Pedro \> Saenz\> 2718281
\end{tabbing}
```

obtenemos

Nombre	Apellido	Teléfono
Juan	Gómez	3141592
Pedro	Saenz	2718281

Los comandos `\+` se pueden acumular y a su vez pueden ser contrarrestados con `\-`. El comando `\<` salta al stop anterior. Al principio de una línea contrarresta localmente el efecto de un `\+` (pero en las líneas siguientes sigue actuando el `\+`).

El comando `\'` hace que el texto precedente se justifique por la derecha respecto al stop anterior. Por ejemplo,

```
\begin{tabbing}
xxxxxxxxxxxxx\=\kill
20\> 30\'500\\
100\>300\'8000
\end{tabbing}
```

produce

20	30	500
100	300	8000

El comando `\`` lleva el texto que sigue contra el margen derecho. Por ejemplo,

```
\begin{tabbing}
Nombrexxxxx\=\kill
Nombre \> Apellido \\'Tel\'a\'efono\\
Juan \> G\'a\'omez\' 3141592\\
Pedro \> Saenz\' 2718281
\end{tabbing}
```

produce

Nombre	Apellido	Teléfono
Juan	Gómez	3141592
Pedro	Saenz	2718281

Listas Para enumerar o clasificar párrafos disponemos de los tres entornos `enumerate`, `description` e `itemize` (para entrar en ellos escribimos `\begin{enumerate}` etc.) Cada párrafo se inicia con el comando `\item`. Por ejemplo, si escribimos:

Orden del día:

```
\begin{enumerate}
\item Lectura y aprobación del acta anterior.
\item Tribunal de la plaza XXX.
\item Asuntos de tercer ciclo.
\item Ruegos y preguntas.
\end{enumerate}
```

El resultado es:

Orden del día:

1. Lectura y aprobación del acta anterior.
2. Tribunal de la plaza XXX.
3. Asuntos de tercer ciclo.
4. Ruegos y preguntas.

Podemos referirnos indirectamente a los puntos de la forma habitual, por ejemplo, si ponemos

```
\item\label{plaza} Tribunal de la plaza XXX.
```

después podremos escribir:

La documentación referente al punto `\ref{plaza}` está disponible en Secretaría,

con el efecto:

La documentación referente al punto 2 está disponible en Secretaría

Si cambiamos `enumerate` por `itemize` el resultado es

- Lectura y aprobación del acta anterior.
- Tribunal de la plaza XXX.
- Asuntos de tercer ciclo.

- Ruegos y preguntas.

Con `description` no aparece ninguna marca al comienzo de cada párrafo. En cualquiera de los tres entornos, `\item` puede llevar como argumento opcional la etiqueta que queramos que aparezca. Por ejemplo,

G. Rossini Il barbiere di Siviglia, La gazza ladra, La cenerentola, Semiramide, Il viaggio a Reims, Guillaume Tell.

G. Puccini Tosca, La Boheme, Madama Butterfly, Turandot, Manon Lescaut, La fanciulla dal west.

G. Verdi Nabucco, Un ballo in maschera, Rigoletto, Il trovatore, La traviatta, La forza del destino, Otello, Aida, Falstaff.

se obtiene con

```
\begin{description}
\item[G. Rossini] Il barbiere di Siviglia,...
\item[G. Puccini] Tosca,...
\item[G. Verdi] Nabucco,...
\end{description}
```

El resultado con `itemize` es similar:

G. Rossini Il barbiere di Siviglia, La gazza ladra, La cenerentola, Semiramide, Il viaggio a Reims, Guillaume Tell.

G. Puccini Tosca, La Boheme, Madama Butterfly, Turandot, Manon Lescaut, La fanciulla dal west.

G. Verdi Nabucco, Un ballo in maschera, Rigoletto, Il trovatore, La traviatta, La forza del destino, Otello, Aida, Falstaff.

Podemos modificar el estilo de `enumerate`. Por ejemplo, si ponemos en el preámbulo

```
\makeatletter\renewcommand\theenumi{\@alph\c@enumi}\makeatother
\renewcommand\labelenumi{\theenumi}}
```

Las etiquetas serán a), b), c) ... en lugar de 1., 2., 3. ...

Con más detalle: donde pone `alph` podemos poner:

arabic Produce números 1, 2, 3, ... (no es necesario especificarlo, es la opción por defecto).

alph Produce letras a, b, c, ...

Alph Produce letras mayúsculas A, B, C, ...

Roman Produce números romanos I, II, III, IV, ...

roman Produce romanos en minúsculas i, ii, iii, iv, ...

El argumento de `\labelenumi` contiene los símbolos adicionales que acompañan al número (representado por `\theenumi`). Por ejemplo, si queremos los números entre corchetes [1], [2], [3], ... especificaremos

```
\renewcommand\labelenumi{[\theenumi]}
```

5 Objetos flotantes

Objetos flotantes son porciones de texto que no pueden cortarse al terminar la página. \LaTeX reconoce dos tipos: tablas y figuras. Si queremos que una tabla quede al principio o al final de la página no podríamos hacerlo tecleándola sin más, pues no sabemos en qué punto de la página aparece cada cosa que escribimos en el texto fuente. Para conseguirlo usamos el entorno `table`. Por ejemplo, la tabla 1 que aparece al final de la página ha sido generada mediante:

```
\begin{table}[b]
\caption{Las 15 categorías de signos \TeX}\label{etiqueta}

\vspace{5mm}
\begin{tabular}{|c|l|c|l|c|}
\hline
Categ.&Significado&Ejemplo&Categ.&Significado&Ejemplo\\
\hline
.....
\end{tabular}
\end{table}
```

El argumento opcional `[b]` (bottom) indica que la tabla debe ir al final de la página. Otras alternativas son `t` (top, arriba), `h` (here, aquí) y `p` (page, en una página aparte, toda con tablas y figuras). Dentro de un entorno `table` podemos poner una o más tablas (que aparecerán todas en bloque). Cada una puede llevar su `\caption{título}`, lo que produce además la numeración automática de la tabla y la posibilidad de referirse a ella mediante etique-

Tabla 1: Las 15 categorías de signos \TeX

Categ.	Significado	Ejemplo	Categ.	Significado	Ejemplo
0	Carácter de escape	<code>\</code>	8	Subíndice	<code>-</code>
1	Inicio grupo	<code>{</code>	9	Ignorado	
2	Fin grupo	<code>}</code>	10	Espacio	<code>_</code>
3	Modo matemático	<code>\$</code>	11	Letra	<code>a, b, ...</code>
4	Tabulador	<code>&</code>	12	Otros	<code>@</code>
5	Fin de línea		13	Activo	
6	Parámetro	<code>#</code>	14	Comentario	<code>%</code>
7	Superíndice	<code>^</code>	15	Inválido	

tas. En principio \LaTeX no escribe “Tabla”, sino “Table”. Para traducirlo ponemos en el preámbulo `\renewcommand\tablename{Tabla}`.

Es importante comprender que lo único que hace el entorno `table` es buscar espacio para la tabla, pero no crea ninguna tabla. La tabla se crea normalmente con un entorno `tabular` o `tabbing` o de cualquier otro modo. También es cosa nuestra distribuir las tablas (si es que hay varias) una al lado de otra, una bajo la otra o como sea, así como regular la distancia entre el título y la tabla etc.

Si estamos escribiendo a dos columnas (con la opción `twocolumn`) las tablas ocupan la página entera. Si queremos una tabla que sólo ocupe una columna usamos el entorno `table*`.

El entorno `figure` se comporta exactamente igual que `\table`, sólo que en los títulos pone “figura” en lugar de “tabla”. Mejor dicho, pone “figure”. Si queremos traducirlo usamos `\renewcommand\figurename{Figura}`.

NOTA: Para insertar una figura en el texto podemos usar el comando `\special`. Los ejemplos de este comando que aparecen más abajo sólo valen para Macintosh. En un PC la sintaxis es distinta, pero todo lo que queda fuera del comando `\special` vale igual.

Para insertar una figura hemos de reservar un espacio. Hay varias formas de hacerlo. La más habitual es mediante un entorno `picture`. Por ejemplo, esta figura:

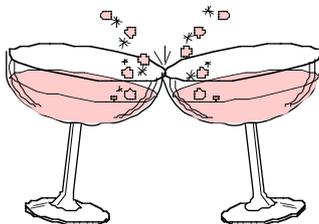


Figura 1: Feliz navidad

ha sido creada mediante

```
\begin{figure}[h]
\begin{center}
\begin{picture}(44,25)
\special{picture copas scaled 500}
\end{picture}
\caption{Feliz navidad}
\end{center}
\end{figure}
```

El entorno `picture` requiere como argumento las dimensiones del espacio que ha de reservar para la figura expresadas en la unidad de longitud en curso. Esta se fija (normalmente en el preámbulo) mediante

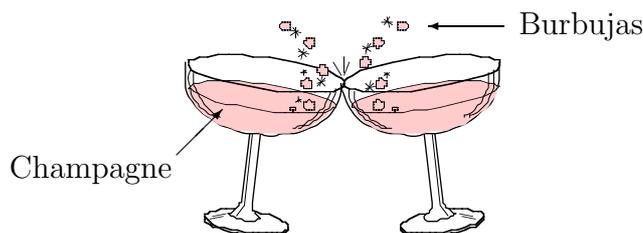
```
\setlength\unitlength{1mm}.
```

Podemos elegir cualquier otra unidad.

Si las figuras han de contener algún texto, es preferible no incorporarlo en el programa que la genera, sino desde el propio \LaTeX , de modo que no se produzcan cambios de fuentes. Dentro del entorno `picture` podemos usar el comando `\put(x, y){algo}` para poner “algo” en la posición (x, y) , donde el origen de coordenadas es el extremo inferior izquierdo de la figura. Por ejemplo, si tras la instrucción `\special` incluimos las líneas

```
\put(-22,9){Champagne}
\put(0,12){\vector(1,1){6}}
\put(45,28){Burbujas}
\put(43, 29){\vector(-1, 0){10}}
```

obtenemos



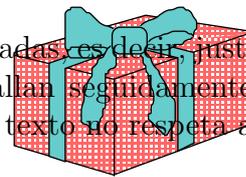
A veces no interesa reservar espacio para una figura, sino justo lo contrario, dibujarla sin reservar ningún espacio para poder escribir a su lado, como ocurre con ésta de aquí al lado. Para ello la hemos de encerrar en una caja y después reducir sus dimensiones a cero. Una caja es una especie de variable \TeX donde podemos guardar un texto temporalmente y luego imprimirlo. Existen 256 cajas que podemos llenar y vaciar cuando queramos. Por ejemplo, las instrucciones



```
\setbox0\vbox{
\begin{flushright}
\begin{picture}(33,18)
\special{picture regalo}
\end{picture}
\end{flushright}
}
```

meten en la caja número 0 (`\setbox0`) una caja (`\vbox`) que contiene el regalo justificado por la derecha. Si a continuación escribiéramos `\box0` vaciaríamos la caja e imprimiríamos su contenido. El efecto sería el mismo que si no hubiéramos hecho nada con las cajas. Sin embargo, si antes de vaciarla escribimos `\wd0=0pt\ht0=0pt`, estamos haciendo nulas la anchura (width) y altura (height) de la caja número 0, con lo que al vaciarla su contenido se imprime normalmente, pero a todos los efectos es como si no hubiéramos imprimido nada, es decir, el texto que escribamos a continuación irá a parar donde hubiera ido aunque no hubiéramos imprimido la caja.

Este párrafo ha sido escrito tras las instrucciones indicadas *es decir, justo después de haber tecleado las instrucciones que se detallan seguidamente.* Vemos que el resultado no es lo que queríamos porque el texto *no respeta* al dibujo.



```
\setbox0\vbox{\begin{flushright}
\begin{picture}(33,18)
\special{picture regalo}
\end{picture}
\end{flushright}}
\wd0=0pt\ht0=0pt\box0
Este párrafo ha sido escrito ...
```

Para que todo quede perfecto sólo nos falta diseñar párrafos en forma de L. Esto se consigue añadiendo la línea

```
\hangindent=-37mm\hangafter =-5\vspace{-3mm}
```

Esto hace que las 5 primeras líneas del párrafo se acorten en 37 milímetros por la derecha. Si quisiéramos que se acortaran por la izquierda especificaríamos `37mm` sin el signo negativo. Si quisiéramos que se redujeran las 5 últimas en lugar de las 5 primeras especificaríamos un 5 positivo.



Por ejemplo, este párrafo se ha obtenido con las instrucciones que se detallan abajo. Observamos que hemos quitado el signo negativo en los `37mm`. También hemos quitado el justificado por la derecha y hemos añadido un `\noindent` para evitar que `TEX` sangre el regalo dentro de la caja. Una vez pasadas las 5 primeras líneas el texto vuelve automáticamente a su anchura normal.

```

\setbox0\vbox{\noindent
\begin{picture}(33,24)
\special{picture regalo}
\end{picture}
}
\wd0=0pt\ht0=0pt\box0
\hangindent=37mm\hangafter =-5\vspace{-3mm}
Por ejemplo, este párrafo ...

```

Notar que las intrucciones `\hangindent` y `\hangafter` sólo afectan al párrafo que las contiene. Si queremos cambiar de párrafo antes de terminar el número de líneas que queremos sangrar deberemos volverlas a copiar (descontando de `\hangafter` las líneas que ya se han sangrado).

Veamos otra inserción interesante:

En el preámbulo del documento podemos definir el comando `\membrete` de la forma siguiente:

```

\def\membrete{
\vspace*{-35mm}
\hspace{-35mm}
\vbox{\hsize = 70mm
\begin{center}\footnotesize\sf
\begin{picture}(23,15)
\special{picture escudo scaled 200}
\end{picture}

```

```

{\bf UNIVERSITAT DE VAL\`ENCIA}

```

```

Departament\\ d'Economia Financera i Matemàtica\\
Edificio Departamental Oriental (5ª planta)\\
Avda.\ de los Naranjos s/n\\
46071 VALENCIA
\end{center}}

```

En la página siguiente tenemos un ejemplo de su efecto.



UNIVERSITAT DE VALÈNCIA
Departament
d'Economia Financera i Matemàtica
Edificio Departamental Oriental (5ª planta)
Avda. de los Naranjos s/n
46071 VALENCIA

REUNIÓN DE LA UNIDAD DOCENTE

Estimado compañero:

Te convoco a la reunión de la Unidad Docente de Matemáticas que tendrá lugar el próximo jueves día 24 de diciembre a las 21'30 horas en el aula del Departamento con el siguiente orden del día:

1. Informe del coordinador.
2. ¿Qué son las matemáticas?
3. Existencia del alma.
4. Debate sobre la influencia de los últimos progresos de la arqueología etrusca en la docencia de las matemáticas para economistas.
5. Ruegos y preguntas.

Valencia, a 7 de julio de 1999.

Fdo.: El Coordinador

El texto fuente es:

```
\pagestyle{empty}
\membrete

\vspace{1cm}

\centerline{\Large \bfseries \underline{\underline{
REUNION DE LA UNIDAD DOCENTE}}}}

\vspace{1.5cm}

Estimado compañero:

\medskip

Te convoco ...

\begin{enumerate}
\item Informe del coordinador.

\item ¿Qué son las matemáticas?

.....

\end{enumerate}

\vspace{2cm}

{\raggedleft
\begin{tabular}{l}
Valencia, a \today.\\[2.5cm]
Fdo.: El Coordinador
\end{tabular}
}
```

Observar que no hace falta poner la fecha, sino que `\today` la genera automáticamente.

6 Ajustes de estilo

Recogemos aquí algunas instrucciones que alteran el estilo predeterminado del documento. Comenzamos con las longitudes que regulan el aspecto de la página.

`\oddsidemargin` Distancia del texto al borde izquierdo del papel menos una pulgada. Si el estilo distingue entre páginas pares e impares, esta longitud sólo afecta a las impares. Las páginas pares las regula `\evensidemargin` (pero si el estilo no distingue las páginas pares y las impares entonces `\evensidemargin` no tiene efecto).

`\textwidth` Anchura del texto.

`\topmargin` Distancia de la cabecera al borde superior del papel menos una pulgada.

`\headheight` Altura de la cabecera.

`\headsep` Distancia de la cabecera al texto.

`\textheight` Altura del texto.

`\footskip` Distancia del texto al pie de página.

Ejemplo:

`\setlength\textheight{10cm}` ajusta la altura del texto a 10cm.

`\addtolength\textheight{-5cm}` resta 5cm. a la altura del texto.

Las instrucciones anteriores son GLOBALES: afectan a todo el documento. A continuación vemos instrucciones LOCALES que afectan al texto que sigue:

<code>\hoffset</code>	margen izquierdo menos una pulgada.
<code>\size</code>	ancho de línea.
<code>\leftskip</code>	espacio adicional a principio de línea (0 por defecto)
<code>\rightskip</code>	espacio adicional a final de línea (0 por defecto)
<code>\parindent</code>	longitud del sangrado
<code>\parfillskip</code>	espacio de relleno en la última línea de un párrafo.
<code>\baselineskip</code>	distancia entre las líneas base de líneas consecutivas.
<code>\lineskiplimit</code>	mínima distancia admisible entre las líneas base de líneas consecutivas. Si no se respeta se añade espacio adicional usando <code>\lineskip</code>
<code>\lineskip</code>	espacio entre la parte inferior de una línea y la superior de la siguiente si no se respeta <code>\baselineskip</code> .
<code>\parskip</code>	espacio vertical entre dos párrafos.
<code>\abovedisplayskip</code>	espacio por encima de una fórmula centrada.
<code>\belowdisplayskip</code>	espacio por encima de una fórmula centrada.

La sintaxis para modificar estas magnitudes es `\baselineskip=.5cm`. Además se dispone de las instrucciones siguientes para hacerlo:

`\nointerlineskip` suprime el espacio adicional entre las línea anterior y posterior.

`\offinterlineskip` suprime el espacio adicional entre líneas en lo sucesivo.

`\noindent` suprime el sangrado en el párrafo siguiente.

`\hangafter = n` añade una sangría adicional (independiente de la que produce `\parindent`) a partir de la línea n del párrafo y cuya longitud se especifica mediante `\hangindent = 5pt`. Si n es negativo la sangría se aplica a las primeras líneas del párrafo. Si `\hangindent` es negativo la sangría se aplica a la derecha.

`\parshape = n $i_1 l_1 i_2 l_2 \dots i_n l_n$` produce un párrafo donde las n primeras líneas tienen sangría i_k y longitud l_k .

El primer párrafo tras un título de sección no se sangra. Para sangrarlo ponemos al principio del documento `\usepackage{indentfirst}`