

Escribir con \LaTeX

Jesús García de Jalón de la Fuente
IES Ramiro de Maeztu
Madrid

2016-2017

1. L^AT_EX

En general, un texto está compuesto por el contenido y por el formato. Cuando hablamos de formato nos referimos al tipo de letra, a la división en párrafos, a la división en capítulos y secciones, tipo de papel, márgenes, etc.

L^AT_EX es un lenguaje de marcas. Un archivo L^AT_EX es un archivo de texto puro en el que alternando con el contenido se insertan marcas que indican el formato. Posteriormente, este archivo se compila y se obtiene el texto formateado de acuerdo con las instrucciones de las marcas. Para distinguir el contenido de las marcas, éstas empiezan siempre por el carácter `\`. Veamos un ejemplo, para obtener

$$f(x) = \begin{cases} \operatorname{sen} x & \frac{3\pi}{2} < x < 3\pi \\ x - 3\pi & x \geq 3\pi \end{cases}$$

en L^AT_EX se escribe:

```
\[
f(x)=\begin{cases}\operatorname{sen} x&\frac{3\pi}{2}<x<3\pi\\x-3\pi&x\geq3\pi\end{cases}
\]
```

Aunque es más complicado aprender a escribir de este modo que con los procesadores de texto tradicionales, después de un pequeño período de aprendizaje se empiezan a apreciar las ventajas de este sistema: los ficheros de L^AT_EX son ficheros de texto puro y, por consiguiente, mucho más ligeros, la presentación de un texto científico con L^AT_EX es muy superior que la que se obtiene con procesadores del tipo Office y, además, tras un poco de práctica, incluso resulta más fácil.

Decía el Quijote que empezar las tareas es tenerlas medio terminadas. Efectivamente, lo más difícil con L^AT_EX es empezar. Para que empezar con L^AT_EX resulte más fácil hemos escrito estas páginas.

2. Marcas L^AT_EX

En L^AT_EX hay tres tipos de marcas:

◇ Comandos.

Los comandos se aplican al contenido que aparece seguidamente entre los signos `{` y `}`. Un comando puede tener uno o varios argumentos. Por ejemplo, para escribir una fracción se utiliza el comando `\frac{}{}` que tiene dos argumentos, uno para el numerador y otro para el denominador.

Los comandos pueden tener también parámetros que se colocan entre el comando y sus argumentos entre corchetes. Por ejemplo, para escribir $\sqrt[3]{x}$, el índice de la raíz hay que escribirlo como parámetro. La raíz se escribiría `\sqrt[3]{x}`.

Algunos comandos no tienen argumentos y se limitan a escribir determinado signo. Esto es muy común con los símbolos matemáticos. Por ejemplo, para obtener π escribiremos `\pi`, para obtener \in escribiremos `\in`, etc. Estos símbolos solo se pueden escribir dentro de un entorno matemático. Más adelante se verá qué significa entorno.

◇ Declaraciones.

Se trata de marcas que producen un determinado formato dentro de un bloque de contenido. En L^AT_EX los bloques se establecen mediante los símbolos `{` y `}` o mediante los entornos que veremos en el apartado siguiente. Por ejemplo, supongamos que queremos escribir un determinado texto en small caps: ESTO ESTÁ ESCRITO EN MAYÚSCULAS PEQUEÑAS. Escribiremos `{\scshape esto está escrito en mayúsculas pequeñas}`. La declaración `\scshape` hace que todo el texto que sigue hasta el fin del bloque se escriba en small caps. Muchas veces el mismo efecto puede conseguirse mediante un comando o una declaración. El texto anterior podría escribirse también mediante `\textsc{esto está escrito en mayúsculas pequeñas}`.

◇ **Entornos.**

Los entornos se utilizan para tareas complejas como escribir tablas, matrices, listas, etc. Escribir una matriz 3×3 mediante un comando requeriría 9 argumentos y, quizá, varios parámetros. Por todo ello es preferible el procedimiento de los entornos.

Un entorno L^AT_EX tiene la forma:

```
\begin{nombre del entorno}
    contenido del entorno
\end{nombre del entorno}
```

Por ejemplo la matriz:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

se escribe:

```
\[
  \begin{bmatrix}
    1&2&3\\4&5&6\\7&8&9
  \end{bmatrix}
\]
```

Aquí los corchetes indican que se escribe un texto matemático, equivalen a otro entorno como se verá más adelante.

En ocasiones, un determinado efecto puede obtenerse mediante una declaración o mediante un entorno. Por ejemplo, centrar el texto se puede conseguir mediante la declaración `\centering` o mediante el entorno `\begin{center}\end{center}`. También el aumento de tamaño de letra que se consigue con `\large` puede conseguirse con `\begin{large}\end{large}`. Obsérvese que el nombre del entorno no lleva la barra `\`.

Los efectos no son exactamente iguales puesto que el entorno `\begin{center}\end{center}` produce un cambio de párrafo y la declaración `\centering` no lo hace.

Un entorno puede tener argumentos entre llaves o parámetros entre corchetes. Por ejemplo `\begin{array}{lll}` o `\begin{enumerate}[topsep=0]`.

3. El preámbulo

El contenido del documento debe situarse dentro del entorno `\begin{document}\end{document}`. Antes se sitúan unos comandos de formato que forman el preámbulo del documento.

El comando `\documentclass[parámetros]{argumento}` permite definir el tipo de documento que se va a escribir y establecer como parámetros algunas variables de formato separadas por comas:

- ◇ Las clases básicas que se toman como argumentos son `article`, `book`, `report` y `letter`. La clase `book` supone un documento a doble cara con los márgenes exteriores aproximadamente dobles de los márgenes interiores. En cambio, en la clase `article` se supone que la impresión se hará a una cara y los márgenes son iguales.
- ◇ Algunos de los parámetros del comando son los siguientes
 1. El tamaño de papel que puede tomar los valores `a4paper`, `a5paper`, `legalpaper`, `letterpaper` o `executivepaper`.
 2. El tamaño de letra `10pt`, `11pt` o `12pt`.

3. `landscape` para cambiar de orientación.
4. `oneside` o `twoside` para escribir a una o dos caras.
5. Para escribir en columnas `twocolumn`.
6. Para indicar que los capítulos empiecen en página impar o cualquiera se utiliza `openright` o `openany`.
7. Para poner el título en una página separada o no se usa `titlepage` o `notitlepage`. En la clase `article` se establece por defecto `notitlepage`.
8. Para alinear las ecuaciones a la izquierda `fleqn`.

Los comandos \LaTeX se encuentran en archivos que se llaman paquetes o `packages`. Solamente algunos de ellos se cargan al iniciar el programa por lo que, si queremos usar alguno de estos comandos debemos decirlo al compilador. Para ello se usa el comando `\usepackage{nombre}`. Por ejemplo, para incluir, cosa muy conveniente, los comandos y entornos de la American Mathematical Society deberemos poner en el preámbulo `\usepackage{amsfonts,amsmath,amssymb}`.

Otro `package` que es conveniente cargar es `geometry` que permite configurar la página de modo más eficiente que `\documentclass`. Las variables pueden establecerse como parámetros de `\usepackage` o como argumentos de `\geometry`. Por ejemplo, puede fijarse la anchura del texto como

```
\usepackage[textwidth=16cm]{geometry}
```

o, una vez cargado el `package` mediante

```
geometry{textwidth=16cm}.
```

Algunas de las propiedades que pueden especificarse con `geometry` son las siguientes:

- ◊ Tamaño de papel:
 - ▷ `paper=a4paper, legalpaper, ...`
 - ▷ `paperwidth`
 - ▷ `paperheight`
 - ▷ `landscape` o `portrait`
- ◊ Área de texto:
 - ▷ `textwidth`
 - ▷ `textheight`
- ◊ Márgenes:
 - ▷ `left`
 - ▷ `right`
 - ▷ `inner`
 - ▷ `outer`
 - ▷ `top`
 - ▷ `bottom`
 - ▷ `twoside`

Se pueden definir nuevos comandos con `\newcommand{nombre}{comando}`. Conviene utilizar esta posibilidad para definir comandos para escribir las funciones circulares en español:

```
\newcommand{\sen}{\rm \;sen\,}
\newcommand{\tg}{\rm \;tg\,}
\newcommand{\arsen}{\rm \;arsen\,}
\newcommand{\arcos}{\rm \;arcos\,}
\newcommand{\artg}{\rm \;artg\,}
\newcommand{\arsec}{\rm \;arsec\,}
\newcommand{\arcosec}{\rm \;arcosec\,}
```

En resumen un preámbulo para poder empezar a trabajar podría ser el siguiente:

```

\documentclass[fleqn]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[spanish]{babel}
\usepackage{amsfonts, amsmath, amssymb}
\usepackage[a4paper, textheight=24cm, textwidth=16cm,]{geometry}
\usepackage{graphicx}
\setlength{\parskip}{8pt}
\setlength{\parindent}{0cm}
\setlength{\mathindent}{1cm}
\newcommand{\sen}{\rm \;sen\,}
\newcommand{\tg}{\rm \;tg\,}
\newcommand{\arsen}{\rm \;arsen\,}
\newcommand{\arcos}{\rm \;arcos\,}
\newcommand{\artg}{\rm \;artg\,}
\newcommand{\arsec}{\rm \;arsec\,}
\newcommand{\arcosec}{\rm \;arcosec\,}

```

en donde se fija que las ecuaciones aparecerán alineadas a la izquierda (`fleqn`) con un sangrado de 1 cm (`\setlength{\mathindent}{1cm}`), que no habrá sangrado de párrafos (`\setlength{\parindent}{0cm}`) y la distancia entre párrafos será de 8 puntos (`\setlength{\parskip}{8pt}`). Se incluye el paquete `graphicx` para insertar imágenes.

4. Estructura

Si el documento ha sido especificado como `\documentclass{book}` puede estructurarse en partes, capítulos, secciones, subsecciones y subsubsecciones. Para ello se dispone de los comandos `\part{título}`, `\chapter{título}`, `\section{título}`, etc. Los demás tipos de documento no disponen de la posibilidad de estructurarse en partes y capítulos.

El programa numera automáticamente los distintos apartados en que se estructura el documento. No obstante, si se prefieren los títulos sin numerar puede optarse por la versión con asterisco `\chapter*{título}`, `\section*{título}`, etc.

La separación del texto en párrafos se marca con una línea en blanco. Es preciso tener claro que un cambio de línea en el archivo fuente de L^AT_EX (que tiene la extensión `.tex`) no significa que aparezca un cambio de línea en la versión compilada (normalmente con la extensión `.dvi` o `.pdf`). Como se ha dicho, para indicar un cambio de párrafo debe dejarse una línea en blanco. También hay que tener en cuenta que la separación no cambia por poner varias líneas en blanco. Para el compilador es lo mismo una línea en blanco que varias.

La separación entre párrafos se fija (normalmente en el preámbulo) con el comando

```
\setlength{\parskip}{separación}.
```

Por ejemplo para establecer la separación en 10 puntos escribiremos

```
setlength{\parskip}{10pt}.
```

También pueden usarse milímetros, centímetros, etc.

El cambio de línea lo efectúa el compilador automáticamente pero puede forzarse con el comando `\.`. También, si es necesario, divide las palabras entre dos líneas de acuerdo con el idioma establecido en el preámbulo. Si se efectúa mal la división de en alguna palabra, pueden indicarse los puntos de división posibles con el comando `\-`. Por ejemplo `espe\-\c\i\-\fi\-\ca\-\mente` deja al compilador que elija entre cuatro puntos para efectuar la división.

La separación entre palabras se le indica al compilador mediante el espacio en blanco. En este caso para el compilador es igual varios espacios o un cambio de línea que un solo espacio en blanco. Si se quiere cambiar la separación existen los comandos `\,`, `\;`, `\:`, `\quad`, `\qqquad` que aumentan la separación y el

comando `\hspace{longitud}` que aumenta o disminuye según que pongamos como longitud un número positivo o negativo.

5. Tamaño y tipo de letra

Cambiar el tamaño o el tipo de letra puede conseguirse mediante comandos, declaraciones o entornos. Los más usados se relacionan en la siguiente tabla:

Tipo de letra			
	COMANDO	DECLARACIÓN	ENTORNO
serif	<code>\textrm{...}</code>	<code>\rmfamily</code>	<code>\begin{rmfamily}</code>
sanserif	<code>\textsf{...}</code>	<code>\sffamily</code>	<code>\begin{sffamily}</code>
monospace	<code>\texttt{...}</code>	<code>\ttfamily</code>	<code>\begin{ttfamily}</code>
Peso			
	COMANDO	DECLARACIÓN	ENTORNO
negrita	<code>\textbf{...}</code>	<code>\bfseries</code>	<code>\begin{bfseries}</code>
medio	<code>\textmd{...}</code>	<code>\mdseries</code>	<code>\begin{mdseries}</code>
Forma			
	COMANDO	DECLARACIÓN	ENTORNO
itálica	<code>\textit{...}</code>	<code>\itshape</code>	<code>\begin{itshape}</code>
slanted	<code>\textsl{...}</code>	<code>\slshape</code>	<code>\begin{slshape}</code>
small caps	<code>\textsc{...}</code>	<code>\scshape</code>	<code>\begin{scshape}</code>

Para volver a la fuente por defecto se usa el comando `\textnormal{...}` o la declaración `\normalfont`.

Para resaltar un texto (normalmente mediante cursiva) se usa el comando `\emph{...}`, la declaración `\em` o el entorno `\begin{em}`.

En LaTeX hay 10 tamaños de letra que se especifican mediante declaraciones o entornos. Son los siguientes en orden de tamaño creciente:

```

\tiny    \scriptsize  \footnotesize  \small  \normalsize
\large   \Large       \LARGE         \huge   \Huge

```

Los correspondientes entornos son `\begin{tiny}`, `\begin{scriptsize}`, etc.

6. Entornos matemáticos

Todas las expresiones matemáticas deben estar incluidas en alguno de estos entornos. Podemos clasificar las fórmulas matemáticas en tres tipos:

- ◊ Insertadas en el texto como, por ejemplo, cuando escribimos: sea $f(x)$ una función continua.
- ◊ En una única línea separada del texto, por ejemplo:

$$f(b) = f(a) + f'(\xi)(b - a)$$

◇ En varias líneas:

$$\begin{aligned} f'(x) &= \frac{(6x+5)(x+5) - (3x^2+5x-20)}{(x+5)^2} \\ &= \frac{3x^2+30x+45}{(x+5)^2} \end{aligned}$$

Para escribir en la línea de texto se usa el entorno `\begin{math}\end{math}` que, por ser tan usual, se abrevia rodeando la fórmula con signos de dólar `$`. Por ejemplo $f(x)$ se escribe `$f(x)$`. Cuando se escriben insertadas en una línea algunas expresiones como fracciones, integrales y otras, el compilador reduce su tamaño para encajarlas en la línea. Para que no suceda esto puede utilizarse la declaración `\displaystyle`.

Si queremos escribir una fórmula matemática en una sola línea usaremos los signos `\[` y `\]` rodeando la fórmula. Estos signos son la abreviatura del entorno `\begin{displaymath}\end{displaymath}`. Por ejemplo, la fórmula:

$$f(b) = f(a) + f'(\xi)(b - a)$$

se escribe:

```
\[
  f(b)=f(a)+f'(\xi)(b-a)
\]
```

Si queremos que las ecuaciones aparezcan numeradas sustituiremos el entorno `\[...]` por:

```
\begin{equation}
\label{etiqueta}
...
\end{equation}
```

Así, la ecuación aparecerá numerada y, opcionalmente, mediante el comando `\label{...}` se nos ofrece la posibilidad de ponerle un nombre para poder después referirnos a esta ecuación con el comando `\ref{etiqueta}`.

La expresión anterior escrita con este entorno se mostrará como

$$f(b) = f(a) + f'(\xi)(b - a) \tag{1}$$

Para escribir una expresión matemática en varias líneas existen varias posibilidades. Desde nuestro punto de vista lo mejor es el entorno `\begin{align*}\end{align*}` si no hace falta que las ecuaciones aparezcan numeradas y el mismo entorno sin asterisco para numerar las ecuaciones. En estos entornos se pone el signo `&` donde se quiera que las expresiones aparezcan alineadas (por ejemplo en el signo `=`) y `\\` para cambiar de línea. Por ejemplo:

$$\begin{aligned} f'(x) &= \frac{(6x+5)(x+5) - (3x^2+5x-20)}{(x+5)^2} \\ &= \frac{3x^2+30x+45}{(x+5)^2} \end{aligned}$$

se escribe:

```
\begin{align*}
f'(x)&=\frac{(6x+5)(x+5)-(3x^2+5x-20)}{(x+5)^2}\\
&=\frac{3x^2+30x+45}{(x+5)^2}
\end{align*}
```

Obsérvese que se ha puesto el ampersand (`&`) delante de los signos igual para alinear las expresiones y que después del signo de separación de líneas (`\\`) se ha puesto `[1mm]` para incrementar la separación en

un milímetro. Existe también la posibilidad de alineación en varios puntos, por ejemplo para escribir un sistema de ecuaciones alineando las incógnitas, pero es un poco más complicado.

Existen comandos como `\mbox{...}` o `\text{...}` que permiten escribir texto no matemático dentro de un entorno matemático.

A continuación se dan algunas fórmulas con el código necesario para producirlas. El código debe estar siempre dentro de alguno de los entornos matemáticos que hemos visto.

CÓDIGO	RESULTADO
<code>\frac{2}{3}</code>	$\frac{2}{3}$
<code>\left(\frac{3}{5}\right)^2</code>	$\left(\frac{3}{5}\right)^2$
<code>\log_a N=\frac{\ln N}{\ln a}</code>	$\log_a N = \frac{\ln N}{\ln a}$
<code>a^x=e^{x \ln a}</code>	$a^x = e^{x \ln a}$
<code>a^n=\underbrace{a \cdot a \cdot a \cdot \dots \cdot a}</code>	$a^n = \underbrace{a \cdot a \cdot a \cdot \dots \cdot a}$
<code>\cos x=\sqrt{1-\sen^2x}</code>	$\cos x = \sqrt{1 - \sen^2 x}$
<code>x=\frac{-b \pm \sqrt{b^2-4ac}}{2a}</code>	$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
<code>\lim_{x \to 0} f(x)=\infty</code>	$\lim_{x \rightarrow 0} f(x) = \infty$
<code>u \to 1, v \to \infty \Longrightarrow \lim u^v = e^{\lim(u-1)v}</code>	$u \rightarrow 1, v \rightarrow \infty \implies \lim u^v = e^{\lim(u-1)v}$
<code>\int_a^b f(x) dx = F(b) - F(a)</code>	$\int_a^b f(x) dx = F(b) - F(a)$
<code>\exists \xi \in (a, b) \mid f'(\xi) = 0</code>	$\exists \xi \in (a, b) \mid f'(\xi) = 0$
<code>\mid x \mid = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}</code>	$ x = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$
<code>(a+b)^n = \sum_{i=0}^n \binom{n}{i} a^{n-i} b^i</code>	$(a+b)^n = \sum_{i=0}^n \binom{n}{i} a^{n-i} b^i$
<code>\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}</code>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$

7. Listas

Existen dos entornos de lista principales en \LaTeX . El entorno `\begin{enumerate}\end{enumerate}` para construir listas numeradas y el entorno `\begin{itemize}\end{itemize}` para listas no numeradas.

Ambas son ampliamente configurables por medio de parámetros.

Para escribir la lista:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

el código es:

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

Como puede verse, cada elemento de la lista se marca con el comando `\item` y se numera automáticamente.

Lo mismo ocurre con las listas no numeradas, para conseguir:

- Primer elemento
- Segundo elemento
- Tercer elemento

se escribe:

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

Se pueden anidar sin problemas varias listas y el programa cambia sin necesidad de especificarlo el tipo de numeración. Por ejemplo la lista:

1. Batracios:
 - a) Ranas
 - b) Salamandras
2. Reptiles:
 - a) Lagartos
 - b) Serpientes
 - c) Tortugas

se consigue con el siguiente código (el sangrado no es necesario):

```
\begin{enumerate}
\item Batracios:
  \begin{enumerate}[nolistsep]
  \item Ranas
  \item Salamandras
  \end{enumerate}
\item Reptiles:
  \begin{enumerate}[nolistsep]
```

```

\item Lagartos
\item Serpientes
\item Tortugas
\end{enumerate}
\end{enumerate}

```

Como hemos dicho anteriormente, pueden configurarse muchos parámetros de la lista, entre ellos muchos aspectos geométricos como distancia entre items, distancia al párrafo precedente, distancia entre el número de item y el texto y muchos más. Por ejemplo, en la lista anterior se ha puesto el parámetro `nolistsep` para reducir la separación entre los items de las listas secundarias.

De todos estos parámetros, los más importantes nos parecen los siguientes:

- ◊ Si queremos que la numeración no empiece en uno sino en ocho escribiremos `start=8`.
- ◊ Si la lista ha terminado y queremos que una nueva lista comience siguiendo la numeración de la lista anterior pondremos el parámetro `resume`.
- ◊ Para cambiar el sistema de numeración, por ejemplo para utilizar letras o números romanos usaremos el parámetro `label`. El valor de este parámetro puede ser `\arabic*` (números), `\alph*` (letras minúsculas), `\Alph*` (letras mayúsculas), `\roman*` (números romanos en minúsculas) y `\Roman*` (números romanos en mayúsculas). Pueden añadirse también paréntesis, signos de puntuación o configurar los signos por medio de comandos por ejemplo escribiendo `label=(\emph{\alph*})` con lo que se obtendría la letra en cursiva y entre paréntesis.

Los parámetros se escriben entre corchetes separados por comas. Por ejemplo:

```

\begin{enumerate}[resume, label=\emph{\roman*}]
\item
...
\end{enumerate}

```

Finalmente debemos indicar que el parámetro `label` puede también como valor cualquier signo como `♠`, `♣`, etc. De este modo puede crearse una lista no numerada con el entorno `\begin{enumerate}`.

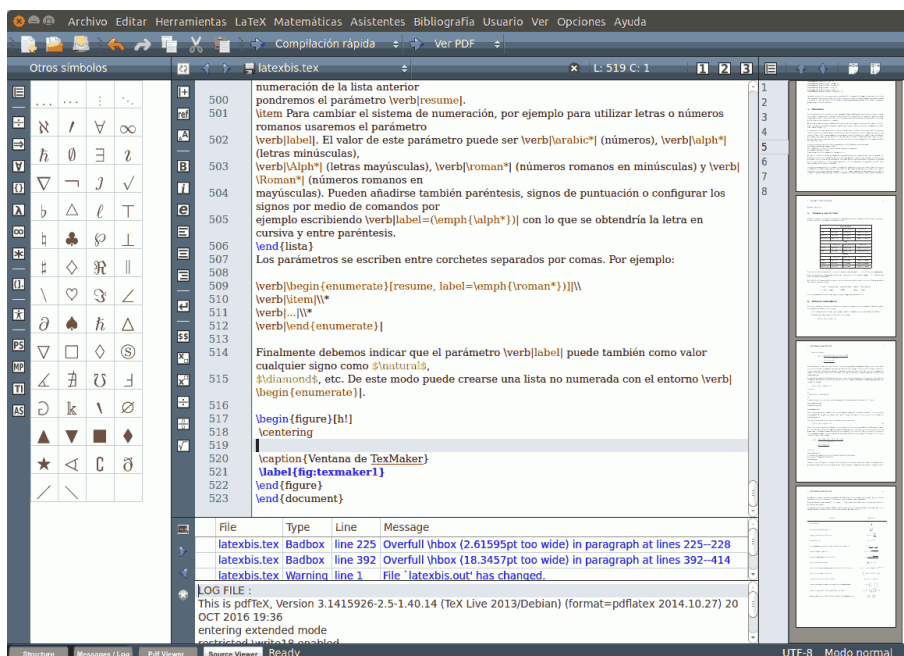


Figura 1: Ventana de TexMaker

8. El editor TexMaker

El archivo fuente de \LaTeX es un archivo de texto puro que lleva la extensión `.tex`. Este archivo debe ser procesado por un programa para obtener el documento formateado para ser leído e impreso, normalmente con la extensión `.pdf`.

Existen numerosos programas gratuitos para procesar el archivo fuente de \LaTeX . Además, suelen incorporar un editor que facilita la escritura del fichero fuente. Uno de los mejores es *Kile* que solamente funciona con el sistema operativo Linux. Por ello recomendamos *TexMaker* que funciona en todas las plataformas.

Se puede descargar TexMaker para Windows en <http://www.descargas.five-fingers.es/latex.zip>. Se debe instalar en primer lugar el archivo de MikTeX y a continuación TexMaker.

La ventana de TexMaker tiene el aspecto que se muestra en la figura 1.

La ventana está dividida en dos paneles. En el de la izquierda aparece o la estructura del documento o una lista de comandos usuales. En el de la derecha es donde se escribe el archivo fuente.

En Opciones >> Configurar TexMaker se presentan varias alternativas sobre el funcionamiento y la apariencia del programa. Nosotros recomendamos la siguiente configuración. En la primera ventana

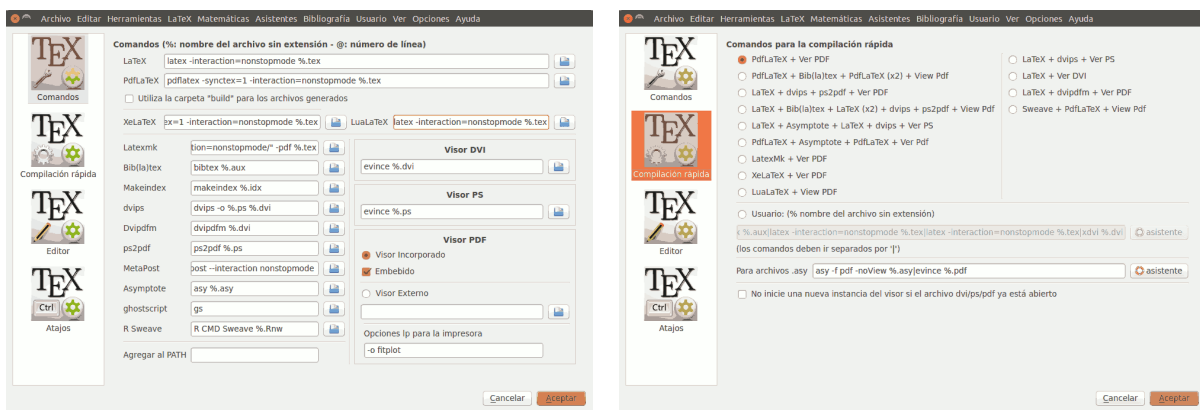


Figura 2: Configuración de TexMaker (1)

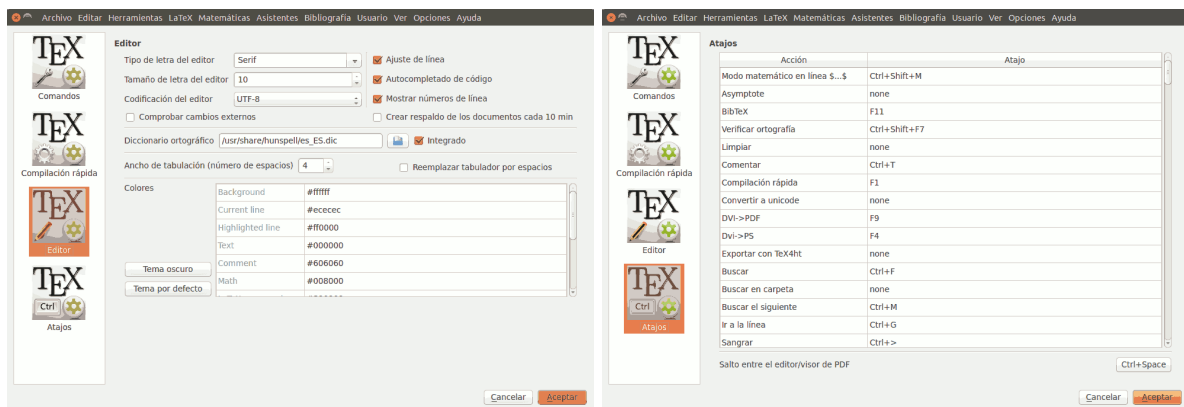


Figura 3: Configuración de TexMaker (2)

casi no es necesario cambiar nada. Únicamente si en nuestro ordenador tenemos instalado el Adobe Reader lo podemos elegir como visor externo de archivos pdf. Aunque el visor de archivos pdf que viene con el programa funciona perfectamente y tiene la posibilidad de integrarse en la ventana de TexMaker, si se va a imprimir el documento, Adobe Reader ofrece más posibilidades (ver figura 2).

La siguiente ventana ofrece varias posibilidades para la compilación del documento. Nosotros recomendamos la opción PdfLaTeX + Ver PDF. El archivo se compila y se muestra en el formato pdf.

En la ventana de configuración del editor elegimos la codificación de caracteres UTF-8. También se nos ofrece la posibilidad de elegir un diccionario para la corrección ortográfica. Si en la lista no aparece el diccionario español se puede poner la ruta del diccionario de LibreOffice en caso de que tengamos instalado este programa. También puede descargarse el diccionario desde la página de LibreOffice.

Existe la posibilidad de configurar los atajos de teclado para los comandos más usuales. Recomendamos aprender los atajos para escribir los entornos matemáticos así como comandos muy frecuentes como fracciones, raíces, subíndices, superíndices y todos los que utilicemos frecuentemente (ver figura 3).

Finalmente, para empezar a escribir el documento \LaTeX , abrimos un nuevo documento vacío, copiamos el preámbulo, abrimos el entorno $\text{\begin{document}}\text{\end{document}}$ y tenemos todo listo (ver figura 4).

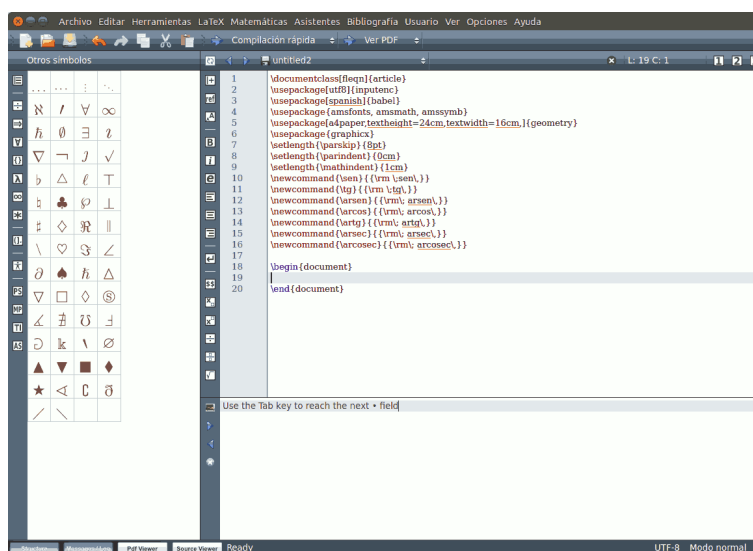


Figura 4: TexMaker configurado y listo para escribir

Para compilar el archivo pulsamos en la flecha donde pone Compilación rápida y veremos en el panel derecho el documento pdf.