

```

\documentclass[11pt]{article}
\usepackage{amsmath}
\usepackage{bm}
\usepackage{tikz}

%Write an article
\title{An Article}
\author{Me}
\date{Today}

\begin{document}
\maketitle

\section{First Topic}
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla facilisis cursus justo, quis sodales orci tempus vitae. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam vitae ex nec ligula blandit pharetra nec pretium arcu. Nunc metus quam, iaculis vel mi nec, placerat ultricies nunc. Quisque magna sem, sodales eget tempus posuere, finibus ut quam. Integer feugiat nibh lectus, eget vestibulum est pellentesque vel. Vivamus commodo lorem metus, ac faucibus magna commodo sit amet. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam varius tempor odio, id finibus lectus. Suspendisse eget diam diam. Duis sed ex in ex blandit varius vitae id lorem. Duis suscipit leo non accumsan sagittis. Etiam quis auctor ligula.

\[\iint_{\Sigma}\nabla\!\times\!\mathbf{F}\cdot d\mathbf{bm}\{\Sigma\}=\oint_{\partial\Sigma}\mathbf{F}\cdot d\mathbf{bm}\{r\}.\]

\section{Second Topic}

```

# LaTeX y Git aplicado a la investigación científica

## Módulo 3 – LaTeX Básico

**del 15 de marzo al 17 de mayo de 2022**

**AULA VIRTUAL**



Manual: *"LaTeX y Git aplicado a la investigación científica"*.

Curso Virtual

Módulo 3 – LaTeX Básico

20 páginas.

Marzo 2022. Elaborado por Pablo Hinojosa.

Asociación Darwin Eventur

# Introducción a LaTeX

Procesadores de textos y su historia .....	1
Descargar LaTeX .....	2
Instalando LaTeX en Windows.....	4
Trabajando con LaTeX .....	5
El equivalente del "Hola Mundo" en LaTeX.....	5
Generación de fichero independiente del dispositivo.....	6
Generación directa de un PDF .....	7
Explicación del texto .....	8
Trabajando con TexMaker en Windows .....	9
El estado mental correcto .....	10
Estructura básica de un documento .....	10
Creando contenido.....	12
Escribiendo .....	14
Capítulos y secciones .....	15
Entornos.....	17
Fórmulas matemáticas .....	19

## Procesadores de textos y su historia

LaTeX no es exactamente un procesador de textos, sino un sistema de preparación de textos enfocado principalmente al área científica, aunque se puede extender a cualquier otra área.

Los primeros sistemas usados para crear textos fueron los editores; en éstos se trabaja simplemente con caracteres, sin más formato que el que se pueda crear a base de espacios y separación entre los mismos; los [procesadores de texto](#) añaden la posibilidad de formatear el texto de múltiples maneras añadiendo resaltes a las letras, usando diferentes tipos de las mismas y organizando el texto automáticamente en líneas, columnas y otras disposiciones.

El primer procesador de texto del tipo que conocemos hoy en día (tras otros usados en microordenadores y sistemas dedicados) fue el [WordStar](#), creado durante la primera mitad de los 80. Sin embargo, no tenía la característica [WYSIWYG](#) que hoy nos resulta

habitual; en el ambiente científico, esta característica no surgió hasta finales de los 80 en el procesador de textos [ChiWriter](#), con el que se han escrito más de una tesis y que tenía la particularidad avanzada de poder trabajar con ecuaciones y fórmulas matemáticas.

LaTeX, sin embargo, comenzó en 1976 en la forma de TeX, un sistema para poder imprimir fórmulas y expresiones matemáticas. Era un sistema de macros que se añadían al texto, pero en algunos casos era excesivamente complicado y no fue hasta que, en 1984, Leslie Lamport creó una serie de extensiones, que se denominaron LaTeX, cuando se empezó a popularizar.

LaTeX era mucho más fácil de usar que TeX, y se concentraba en la estructura del documento, más que en la apariencia. La forma de usarla era la misma: editando un texto al que se le añadían una serie de órdenes; posteriormente el texto se compilaba y se convertía en un formato imprimible (que en 1984 eran muy diferentes a los actuales) o se enviaba directamente a la impresora.

Desde entonces se han hecho diferentes versiones, pero la más popular hoy en día es la 2, aunque se lleva trabajando [en la 3](#) desde hace bastantes años. Dado que LaTeX es software libre, generalmente viene en una serie de distribuciones que incluyen el LaTeX (simplemente un sistema de proceso de textos) junto con otros programas necesarios para hacer el ciclo completo, incluyendo la inclusión de bibliografía y su conversión a diferentes formatos.

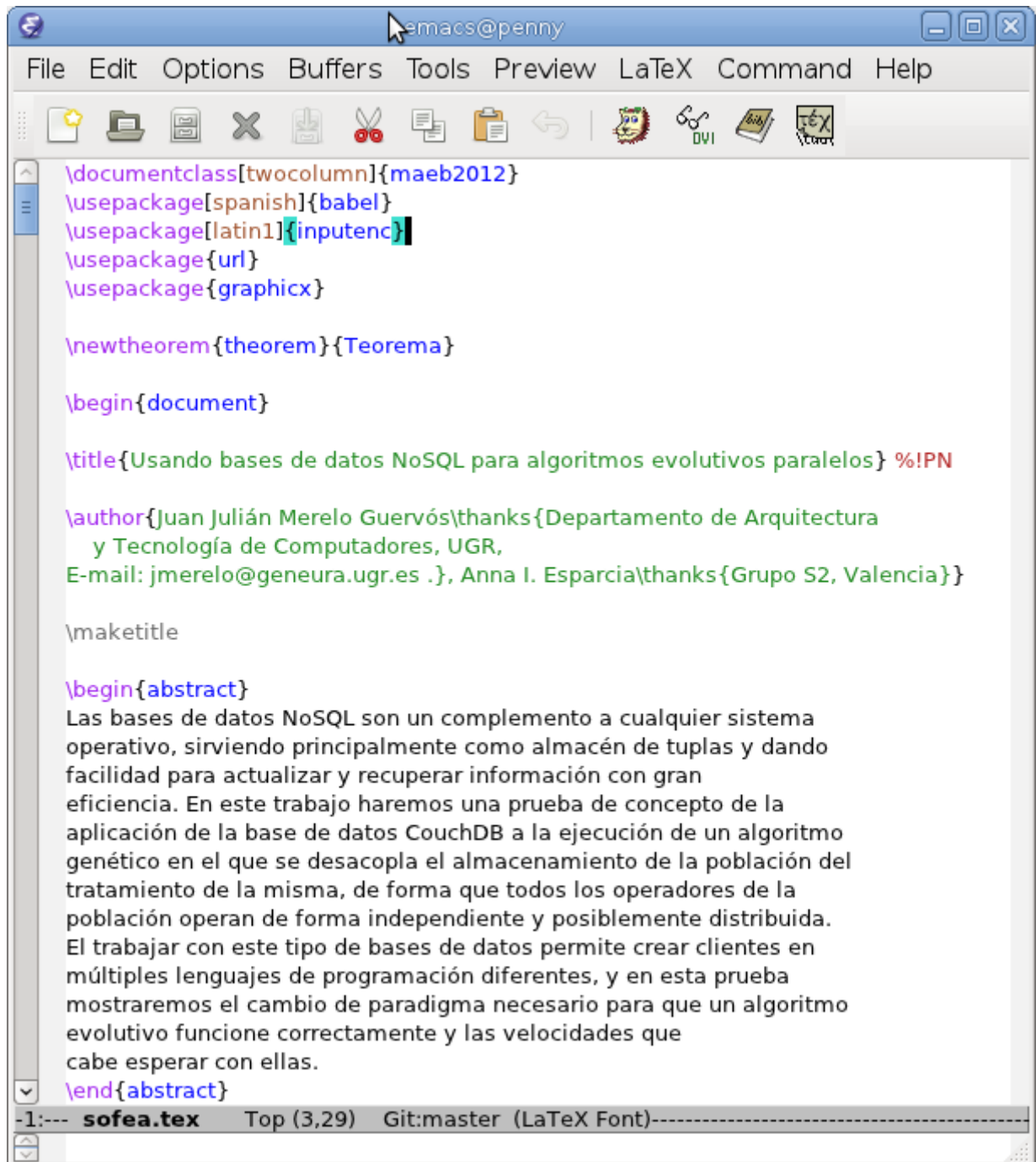
## Descargar LaTeX

En [esta página](#) se ofrecen varios enlaces a versiones de LaTeX para todas las distribuciones. Recordad que LaTeX no es un programa como el LibreOffice Writer, que se lanza e incluye todo lo necesario; una instalación descargará una serie de programas *exclusivamente para generar a partir del texto el fichero imprimible; no se puede lanzar un programa desde el menú y empezar a editar en él.*

Algunos editores de textos permiten trabajar fácilmente desde ellos con LaTeX, con *modos* que te ahorran teclear cosas y funciones que permiten compilar directamente desde él. Personalmente prefiero el [Emacs](#), disponible para todos los sistemas operativos, con el modo AuCTex. Para instalar todo junto, en mi caso y como usuario de Linux, simplemente se ejecuta la siguiente orden en un terminal:

```
sudo apt-get install auctex texlive emacs.
```

Emacs con este modo tendrá la apariencia que se muestra a continuación



En otros sistemas operativos se pueden usar los editores de programador que más se hayan usado, buscando siempre uno que incluya detección de sintaxis y alguna que otra función para facilitar la edición.

Como se puede ver, hay tres botones a la izquierda que sirven para *compilar* el fichero; las tres penúltimas opciones del menú también están relacionadas con LaTeX.

Pero Emacs es un editor complejo con una curva de aprendizaje muy pronunciada.

Para comenzar y al menos tener prácticamente toda la funcionalidad de LaTeX se puede usar [LyX](#), un entorno WYSIWYG que usa internamente LaTeX para representar la información y puede generar el mismo.

En cualquier caso, para este curso se recomienda el uso del editor [TexMaker](#), que es una herramienta multiplataforma fácil de usar y especialmente diseñada para trabajar con LaTeX.

## **Resumiendo**

Para usar LaTeX hacen falta dos cosas: una distribución que incluya todos los programas y un editor de textos; si este editor de textos es de programador facilitará la labor de edición de textos.

Los alumnos son libres de escoger el editor que prefieran, pero se recomienda [TexMaker](#).

## **Instalando LaTeX en Windows**

El problema principal de Windows es que con LaTeX hay que proveer todo lo necesario para trabajar con él, porque no suele incluir las utilidades habituales en estos casos. Afortunadamente, tratándose de software libre, hay una distribución llamada [ProTeXt](#) que tiene todo lo necesario. Sin embargo, hay que descargarse más de un giga para instalarlo; conviene que tengáis más de 3 gigas libres para la instalación completa.

En todo caso, la instalación se divide en tres partes (después de la hora o así que tarda en descargarse)

1. Descomprimir el paquete: se descomprime en un directorio de nuestra elección (también tarda un buen rato)
2. Buscar el directorio y ejecutar "setup.exe".
3. Te da la opción de instalar MiKTeX (que sería, en sí, el programa que incluye LaTeX) y un editor, ConTeXt, muy bien adaptado para trabajar con LaTeX. Es conveniente que instales los dos; el primero tardará también un buen rato (media hora), y el segundo sólo un minuto.

Como has visto, todo el proceso puede durar una buena parte de una mañana o una tarde. Aprovecha para leerte el resto del material del curso, o para reflexionar sobre las ventajas de usar sistemas operativos libres.

## Trabajando con LaTeX

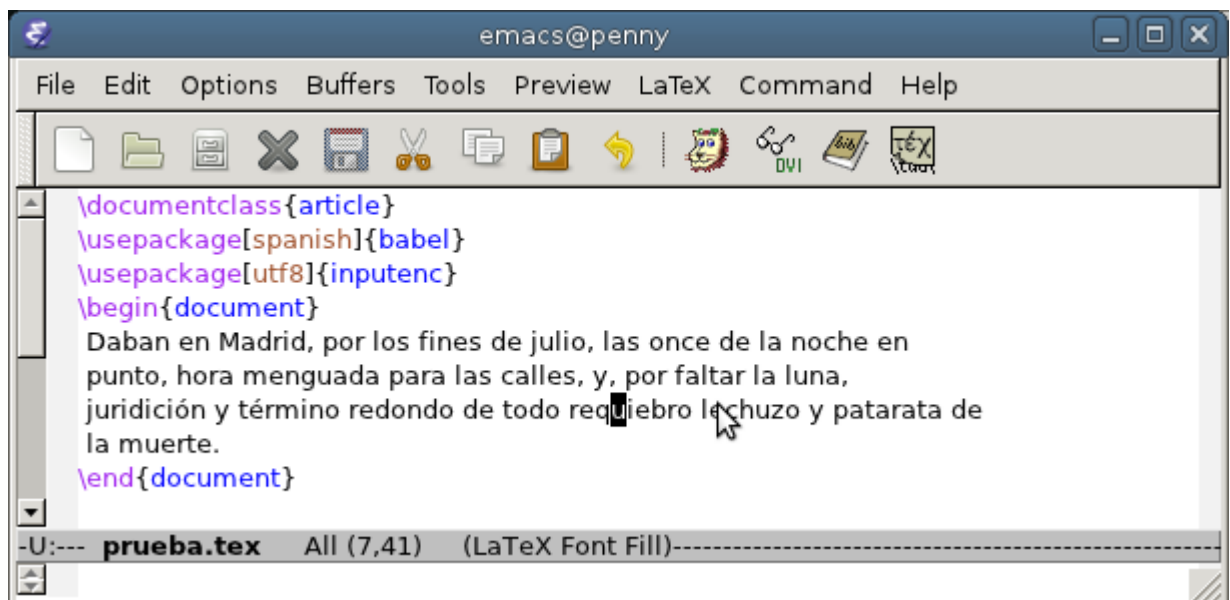
### El equivalente del "Hola Mundo" en LaTeX

LaTeX es un sistema de preparación de documentos y como tal funciona de forma diferente a lo que estamos acostumbrados. Mientras que en los procesadores de textos se trabaja sobre el documento en el formato aproximado en el que saldrá por impresora de forma WYSIWYG, en LaTeX están separadas las fases de edición, visualización y generación del documento en su formato final.

Para comenzar, por tanto, será necesario usar un editor y teclear o copiar/pegar en él un texto similar al siguiente, obtenido del [Diablo Cojuelo publicado en el proyecto Gutenberg](#)

```
\documentclass{article}
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\begin{document}
Daban en Madrid, por los fines de julio, las once de la noche en punto,
hora menguada para las calles, y, por faltar la luna, jurisdicción y término
redondo de todo requiebro lechuzo y patarata de la muerte.
\end{document}
```

Editado desde, por ejemplo, Emacs, tendría esta apariencia



Siempre que se use, como se ha indicado, la librería AuCTeX u otra similar; en este caso se trata, simplemente, del denominado modo LaTeX.

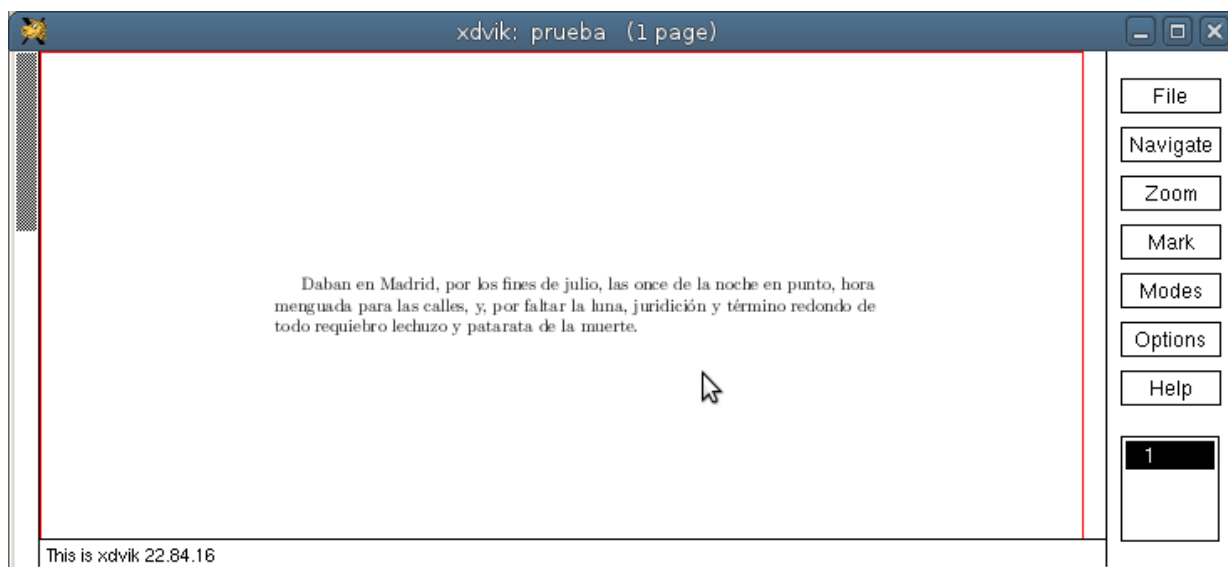


Desde este entorno, la creación del documento se hace en varios pasos y de dos formas diferentes, tras guardar el fichero poniéndole la extensión ".tex".

## Generación de fichero independiente del dispositivo

La configuración por defecto consiste en generar un fichero independiente del dispositivo; es un fichero con la extensión ".dvi" que, posteriormente, se puede imprimir o convertir a cualquier otro formato de fichero. En este caso, habrá que llevar a cabo dos pasos

1. Pulsar el botón con el leoncito, que compilará el fichero. Si no hay ningún error, dará un mensaje en la barra de estado que indicará que se ha podido compilar: "LaTeX: successfully formatted {1} page"; en caso contrario, aparecerá un símbolo que indicará que hay errores.
2. Pulsando sobre las gafas que indican "dvi" se abrirá un programa de visualización del fichero independiente del dispositivo que presentará la apariencia más o menos final del mismo. Desde este programa, aparte de poder hacerse zoom, se pueden también exportar o imprimir, pero generalmente se usa como una "previsualización". La apariencia de este programa es la que se muestra a continuación.



Estos dos pasos se pueden llevar a cabo también desde la línea de órdenes; es la única forma en la que estarán disponibles en todos los entornos de trabajo. Tras guardar el fichero y situarnos en el directorio donde lo hemos guardado, escribimos

```
latex prueba.tex
```

Este fichero (igual que el comando anterior, sólo que de forma invisible) nos habrá generado uno denominado prueba.dvi. Para visualizarlo, se escribe



```
xdvi prueba.dvi
```

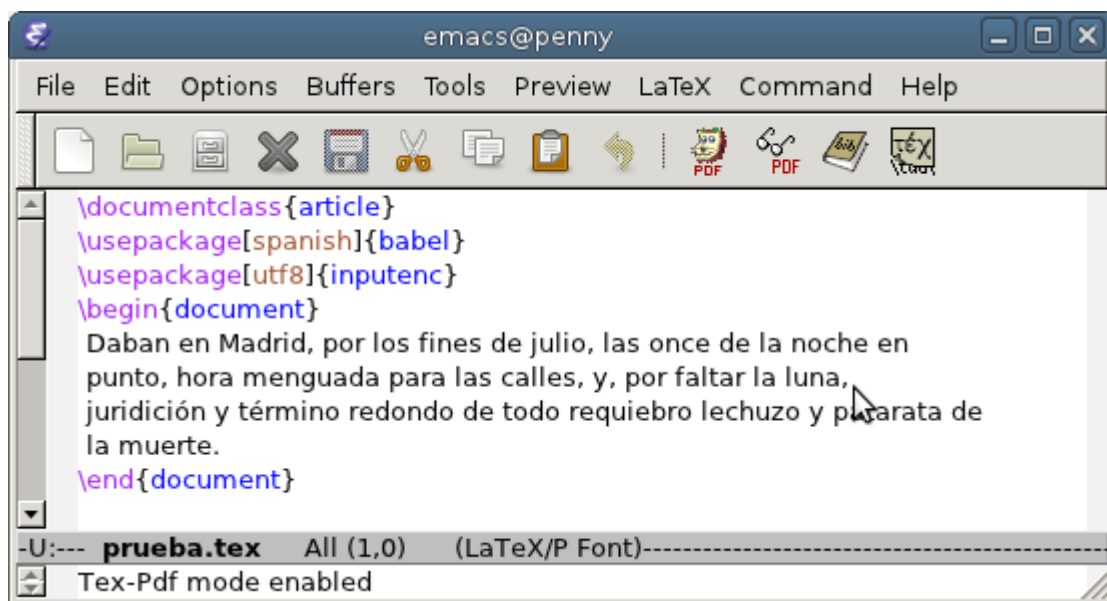
o algún otro programa en otro sistema operativo.

En cualquiera de los dos casos, para generar un PDF a partir del DVI hay que recurrir a la línea de órdenes:

```
dvipdf prueba.dvi
```

## Generación directa de un PDF

En el caso de tener un fichero que no incluya imágenes como este, también existe la opción de generar directamente un PDF. Se puede hacer desde Emacs: se marca Command-> Texing Options -> Generate Pdf (hay una combinación de teclas para esto, como lo hay para todo en Emacs). Como se ve, los iconos de LaTeX y las gafitas de visualización tienen ahora un PDF en rojo:



Pulsando sobre el icono de LaTeX se compilará y se generará directamente un PDF; ahora lanzando el visualizador se abrirá el visualizador de PDFs, posiblemente Evince, con el resultado.

Como hemos explicado anteriormente, también se puede obtener el mismo resultado desde la línea de órdenes, ejecutando

```
pdflatex prueba.tex
```

y, posteriormente, para visualizarlo,

```
evince prueba.pdf
```

## Explicación del texto

En LaTeX, las órdenes comienzan con `\` y consisten generalmente en el nombre de la orden y una serie de argumentos. El argumento va entre llaves y el modificador al mismo entre corchetes y antes de las llaves.

Las tres primeras líneas son órdenes: la primera establece el tipo de documento, y es imprescindible porque modifica el resultado de las otras órdenes. Vamos a usar el tipo más simple, artículo, aunque hay muchos otros tipos; los congresos y revistas, generalmente, tienen el suyo propio.

Las órdenes siguientes:

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

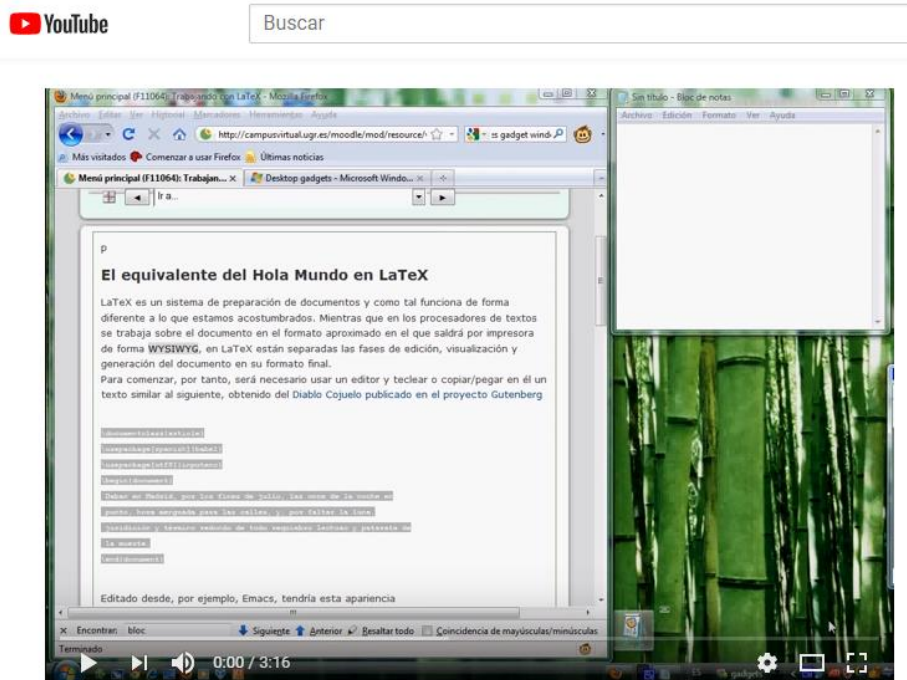
```
\usepackage[utf8]{inputenc}
```

Son imprescindibles para trabajar en español. En cualquier caso, se trata de *paquetes* que añaden funcionalidad al documento; el primero es el paquete Babel para trabajar con diferentes idiomas, y le damos la opción de que se trate de español. Esta orden es tomada en cuenta a la hora de dividir las palabras y usar convenciones como la forma de expresar las fechas. El segundo se refiere al alfabeto que vamos a usar en el fichero; LaTeX usa sólo los caracteres "normales" que no incluyen a las letras castellanas: ñ y ú, por ejemplo; con esta orden le decimos que la codificación del fichero de entrada es utf8, el formato de almacenamiento del mismo más usado por los editores hoy en día. Si tenéis algún problema con esto al compilar; si suprimen los caracteres comentados, probad a cambiar utf8 a latin1 (la codificación que incluye a todos los caracteres del alfabeto español).

Finalmente, tras estas órdenes que se suelen poner al principio del documento pero que no generan ningún tipo de contenido viene un *entorno* en LaTeX. Los entornos están entre *begin* y *end*, y se representan de una forma determinada; en este caso se representarán con el formato que tenga asignado el tipo de documento al principio. Los entornos se cierran en orden contrario al que se abren si están anidados. En este caso, `\end{document}` será siempre la última orden del documento. Todo esto se ampliará más en el tema siguiente; el objetivo de esta sección es simplemente que se comprenda qué es lo que se ha hecho y la estructura muy mínima para crear un documento en castellano.

## Trabajando con TexMaker en Windows

**TexMaker** es un editor libre que funciona en todos los sistemas operativos, pero que forma parte de la distribución ProTeXt que previamente hemos explicado como instalar en Windows. Es un entorno de edición, pero desde el que se pueden llevar a cabo todas las tareas. En el tema siguiente se explica cómo se trabaja con él en más profundidad, pero para que comencéis a usarlos hemos realizado.



Trabajando con TexMakerX

***Pincha en la imagen para ver un video sobre TexMaker***

## Recursos adicionales

### Introducción "oficial" a LaTeX

La (no tan) corta introducción a LaTeX es un documento que ha ido evolucionando desde el principio, y ya ocupa 153 páginas. Imprescindible como guía o como tutorial.

### CervanTeX - Grupo de Usuarios de TeX Hispanohablantes

Veterano grupo de usuarios con multitud de recursos útiles. Como, por ejemplo, [manuales](#).

## El estado mental correcto

Como ya hemos apuntado, los editores para LaTeX no se comportan igual que otros editores de texto de tipo WYSIWYG, sino que más bien funciona como un lenguaje de programación, en el que se crea un archivo fuente que hay que compilar.

LaTeX se mueve como pez en el agua en el entorno de las publicaciones científicas. La mayor parte de las primeras revistas del mundo exigen que los trabajos sean enviados en este formato. Esto es muy conveniente dada la separación natural que LaTeX impone entre información y presentación: El autor pone lo primero, y la revista se encargará de dar lo segundo. Existe una gran diferencia entre lo que es contenido y presentación (actualmente esto ocurre con los gestores de contenido web "cms").

Por ello, al aproximarse a LaTeX desde otros editores hay que tener en mente que su idea fundamental es que el autor se olvide completamente de cosas como la presentación o el formato, y pueda centrarse exclusivamente en su parte, el contenido. Al principio, esto puede resultar muy extraño al usuario novel ("*Yo quiero tal tipo de letra con tal formato en tal posición*"), pero es algo a lo que se acostumbra pronto y se aprende a valorar.

Por ejemplo: En cualquier editor WYSIWYG, el título de una sección no suele ser más que una línea de texto con una tipografía mayor y en negrita.

En LaTeX, sin embargo, el título de una sección no tiene una presentación definida, sino que es una estructura lógica que el compilador puede usar para automatizar tareas como la creación de referencias y notas a pie de página o la construcción de índices.

Como ya has visto en el capítulo anterior de este curso, cualquier editor de texto plano sirve para escribir LaTeX, pero existen herramientas como Texmaker que facilitan la tarea simplificando y automatizando parte del trabajo.

## Estructura básica de un documento

Todo archivo de LaTeX comienza con una declaración del tipo de documento:

```
\documentclass{tipo}
```

Existen multitud de tipos, como "*book*" para libros, "*letter*" para cartas, "*slides*" para transparencias o "*article*" para artículos. Los tipos se pueden definir (si bien de manera bastante compleja) por medio de módulos externos. El tipo que se elija influirá en la posterior estructura y presentación del documento.

`\documentclass`, como otros comandos, permite además que se le apliquen algunos modificadores opcionales, como el tamaño base del texto o el tamaño del papel que se usará. Esto se hace colocándolos entre corchetes `[]` (antes de las llaves`{}`) y separados por comas del siguiente modo:

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

Con esto estamos creando un artículo para una revista y definiendo el tamaño de letra a 11pt (los tamaños válidos son 10, 11 y 12 puntos) y el papel a formato a4

Una vez definido el tipo, ya podemos comenzar el documento en sí.

Todo el contenido del documento (párrafos, imágenes, tablas...) irá colocado entre los comandos `\begin{document}` y `\end{document}`, que son los que definen el principio y el final del documento,

De este modo, nuestro artículo inicial tomaría el siguiente aspecto:

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

```
\begin{document}
```

```
\end{document}
```

Que es un documento de LaTeX en principio perfectamente válido, pero en blanco.

El "Preámbulo" es todo aquello que hay antes de `\begin{document}` (pero siempre después de `\documentclass`), y es donde vamos a ubicar una serie de informaciones y comandos que describen o modifican el documento en conjunto.

Es importante saberlo ya que cuando, a lo largo del curso, hablemos de ubicar algo en la cabecera o "Preámbulo", nos referiremos a este sitio.

## Creando contenido

Por lo que hemos visto hasta el momento, ya podemos deducir que los comando en LaTeX comienzan siempre con una barra inclinada (`\`), que los argumentos que se aplican a esos comandos van entre llaves (`{}`) y que los modificadores se colocan entre corchetes (`[]`)

Por ahora, vamos a dedicarnos al cuerpo del documento (el espacio entre `\begin{document}` y `\end{document}`) viendo tres elementos básicos que nunca deben faltar:

`\title{Titulo del documento}`

`\author{Autor}`

`\date{Fecha}`

Probablemente no haga falta, pero diremos que son los elementos donde se definen, respectivamente, el Título, Autor y Fecha de creación del documento.

Nuestro artículo va tomando forma, y podemos ir ubicando los datos de un modo parecido a este:

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

`\begin{document}`

`\title{Articulo de prueba}`

`\author{Autor}`

`\date{Fecha}`

`\end{document}`

Si compilas el ejemplo anterior verás que, pese a que le hemos colocado un título (y un autor, etc.), este no aparece por ningún lado.

Esto es porque hemos definido los datos, pero no los estamos presentando. Probemos con el siguiente caso:

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

\begin{document}

\title{Articulo de prueba}

\author{Autor}

\date{Fecha}

\maketitle

\end{document}
```

Se diferencia del anterior en la instrucción `\maketitle`, encargada de construir un título para nuestro documento.

Aquí empezamos a ver la potencia de LaTeX. El mismo programa ha creado un título completo, con fecha y autor y lo ha ubicado en la página. Al ser un artículo, lo ha centrado y colocado en la parte superior, pero esto no tiene porqué ser así siempre, si no que depende del tipo de documento (De hecho, el formato *"letter"* ni siquiera soporta `/maketitle`).

Nota interesante: Si usamos `\maketitle` sin haber especificado una fecha, LaTeX usará por defecto la fecha actual.

Prueba a cambiar el tipo de documento por *"book"* y verás como el resultado es distinto.

Otra cosa que habrás notado es que la compilación da errores si usas acentos, eñes o ese tipo de caracteres. Un poco de paciencia: Más adelante veremos cómo arreglar eso.



## Escribiendo

Vamos a poner un poco de texto, para ir viendo algunas otras peculiaridades de LaTeX:

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

```
\begin{document}
```

```
\title{Artículo de prueba}
```

```
\author{Autor}
```

```
\date{Fecha}
```

```
\maketitle
```

Algo de contenido que no se muestra como era de esperar

Sin duda pasa algo muy muy raro

% Y esto es tremendamente misterioso

```
\end{document}
```

De nuevo nos topamos con la idiosincrasia de LaTeX: Todos esos espacios que hemos colocado en las separaciones entre palabra han colapsado a un solo hueco. Recordemos que la filosofía detrás de LaTeX es dedicarnos al contenido y dejarle la presentación y el formato al compilador.

Si se quiere forzar un espacio, este debe de ser precedido de una barra de este modo "\ ". Para acumular varios espacios se debería poner algo así: "\ \ \ \ \ ".

Los retornos de carro son interpretados también como espacios en blanco (como en [Markdown](#)), para influir sobre ello hay un comando que puede resultarnos útil y es "\", que introduce un retorno de carro.

Para separar párrafos LaTeX usa una línea en blanco (o, lo que es lo mismo, dos retornos de carro seguidos).

Otra cosa que descubrimos es que el signo % se usa para marcar los comentarios. Todo lo que le siga será ignorado por el compilador.

LaTeX introducirá automáticamente retornos de carro donde sea necesario para ajustar la línea al ancho de página (más adelante veremos que se puede ajustar este



comportamiento a las reglas de cada idioma), si quiere forzarse a que respete un espacio y no separe dos palabras en líneas distintas, se debe usar la tilde (~) entre ellas.

## Capítulos y secciones

El elemento fundamental para separar y organizar un texto es el párrafo (que, como vimos, se crea separándolo del resto del texto por una línea en blanco).

Pero como no sólo de párrafos se organiza un texto, existen una serie de comandos para indicar títulos de diferentes niveles de importancia y separar secciones, capítulos, etc.

Por orden de importancia, son:

<code>\part{Texto</code>	<i>del</i>	<i>Título</i>	<i>de</i>	<i>este</i>	<i>nivel}</i>
<code>\chapter{Texto</code>	<i>del</i>	<i>Título</i>	<i>de</i>	<i>este</i>	<i>nivel}</i>
<code>\section{Texto</code>	<i>del</i>	<i>Título</i>	<i>de</i>	<i>este</i>	<i>nivel}</i>
<code>\subsection{Texto</code>	<i>del</i>	<i>Título</i>	<i>de</i>	<i>este</i>	<i>nivel}</i>
<code>\subsubsection{Texto</code>	<i>del</i>	<i>Título</i>	<i>de</i>	<i>este</i>	<i>nivel}</i>

Algunos tipos de documento no usan alguno de estos niveles. Por ejemplo, el tipo "article" no usa el nivel *chapter*.

Para empezar a ver esto, hagamos un ejemplo:

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

```
\usepackage[utf8]{inputenc}
```

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

```
\begin{document}
```

```
\part{Apartado Principal}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

```
\section{Una sección}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

```
\subsection{Una subsección dentro de la sección}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla



```
\section{Otra sección}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

```
\section{Y otra sección}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

```
\end{document}
```

Como puedes ver al compilar este ejemplo, LaTeX se ha ocupado de colocar la numeración ante los títulos. Esto es mucho más cómodo de lo que parece a primera vista porque, si posteriormente hacemos modificaciones y agregamos secciones, LaTeX recalculará toda la numeración sin problemas (haz la prueba agregando alguna).

El comportamiento exacto de esta herramienta se ve afectado por el uso del paquete babel. Prueba a compilar este mismo ejemplo, pero sin usar la línea `\usepackage[spanish]{babel}` (borrándola o comentándola con %)

Por supuesto, como hemos visto varias veces, el tipo de documento que estamos usando (en este caso *article*) también afecta a la presentación.

Pero esto no es todo. Otra utilidad de esta forma de estructurar es que LaTeX puede crear automáticamente el índice de contenidos de nuestro documento.

Para ello tenemos el comando `\tableofcontents` que se usa, simplemente, insertándolo en el lugar donde queramos que aparezca nuestro índice:

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

```
\usepackage[utf8]{inputenc}
```

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

```
\begin{document}
```

```
\tableofcontents
```

```
\part{Apartado Principal}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

```
\section{Una sección}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla



```
\subsection{Una subsección dentro de la sección} Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

```
\section{Otra sección} Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

```
\section{Y otra sección}
```

Texto de relleno bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

```
\end{document}
```

**Nota Importante:** A menudo no se muestra ningún cambio en la primera compilación, porque LaTeX debe reconstruir el índice del documento cada vez. Simplemente con volver a compilar de nuevo se resuelve esto.

De nuevo, este índice cambiará de modo automático si cambiamos la estructura del documento.

Prueba a compilar este mismo ejemplo varias veces con y sin

`\usepackage[spanish]{babel}` y con distintos tipos de documento, verás que cambia considerablemente el modo de mostrar el índice.

## Entornos

Anteriormente vimos que el cuerpo de un documento LaTeX está contenido entre los comandos `\begin{document}` y `\end{document}`.

En realidad, lo que hacen `\begin` y `\end` es crear lo que se conoce como un "entorno".

Un entorno es una parte de un documento en la que se definen una serie de propiedades o que se comporta o muestra de un modo determinado.

Por ejemplo, para las citas se suele usar el entorno *quote*, y para alinear el texto a la derecha se usa *flushright*, del siguiente modo:

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

```
\begin{document}
```

```
\title{Artículo de prueba}
```

`\author{Autor}`

`\date{Fecha}`

`\maketitle`

Esto es un texto normal en el propio body, seguido de algo de relleno sin contenido intelectual real (salvo el meramente textual) para hacer un poco de bulto y que no se vea tan soso ni tan cortito.

`\begin{quote}`

Y esto es un texto en un entorno de quote, seguido de algo de relleno sin contenido intelectual real (salvo el meramente textual) para hacer un poco de bulto y que no se vea tan soso ni tan cortito.

`\end{quote}`

`\begin{flushright}`

Y esto otro es un texto en un entorno flushleft, seguido de algo de relleno sin contenido intelectual real (salvo el meramente textual) para hacer un poco de bulto y que no se vea tan soso ni tan cortito.

`\end{flushright}`

`\end{document}`

Según el tipo de documento, *quote* puede cambiar el tipo de letra, la indentación, etc.

De mismo modo que *flushright* obliga a alinear a la derecha, existe otro entorno *flushleft* para hacerlo a la izquierda. Igualmente, un poco más adelante en este curso veremos cómo las ecuaciones se representan con los entornos *displaymath* y *equation*, o las tablas se crean mediante los entornos *table* y *tabular*.

Existen multitud de entornos (más adelante en este curso veremos algunos más) y, además, LaTeX provee de un modo, para que puedan crearse nuevos en un documento cada vez que se necesiten.



## Fórmulas matemáticas

Probablemente la razón por la que LaTeX es famoso es su increíble capacidad para representar ecuaciones y fórmulas matemáticas. Supera en este aspecto a la totalidad de sus competidores y, los que se le acercan en potencia lo hacen porque usan intérpretes basados en LaTeX.

Los comandos, símbolos y lenguaje necesarios para ello son demasiados y demasiado complejos, por lo que se escapan de las posibilidades de este curso, pero aquí veremos una pequeña introducción a su uso.

La forma más simple de escribir una fórmula matemática es "en línea", colocando la fórmula en cuestión entre los símbolos \$ y \$, de este modo:

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

```
\begin{document}
```

La ecuación más famosa de la historia de la física probablemente sea la de  $E=mc^2$ , de donde se deduce que  $c=\sqrt{E/m}$ .

```
\end{document}
```

Nota que LaTeX ha interpretado la fórmula, representándola correctamente.

También pueden escribirse mediante los entornos *displaymath*, que coloca la ecuación en un párrafo aparte y centrada, o *equation*, que le asigna un número de orden y por tanto es más adecuado para identificar una ecuación y poder referenciarla luego:

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

```
\begin{document}
```

```
\begin{displaymath}
```

```
\sum_{0 \leq i \leq m} P(i, j)
```

```
\end{displaymath}
```

Hay muchos símbolos y toda una compleja sintaxis para escribir matemáticas en LaTeX

```
\begin{equation}
```

```
\label{mi ecuación}
```

$f(x)=\sqrt{g'(x)}dx+Z$

`\end{equation}`

La ecuación que se puede ver en `\ref{miecuacion}` es completamente inventada y no tiene sentido físico

`\end{document}`

Nota el uso de `\label` para crear una referencia que luego puede usarse en `\ref`, de modo que coloque automáticamente el número correspondiente a la ecuación.

Existe una ingente cantidad de comandos y símbolos para escribir fórmulas matemáticas en LaTeX y es, de hecho, prácticamente imposible conocerlos todos. Cualquier programa de edición de LaTeX (como, por ejemplo, Texmaker) dispone de atajos y ayudas para esto.

Más adelante en este mismo curso se verán algunos aspectos más avanzados de las fórmulas matemáticas. Mientras tanto, si se necesitan detalles más profundos para la redacción de fórmulas, se puede empezar por consultar [esta página de ayuda](#).

En cualquier caso, el listado de comandos para escribir fórmulas es ingente. Resulta muy útil tener a mano una [hoja de referencia](#).