

**TÍTULO:** *Escritura de documentos en LaTeX*

**OBJETIVOS:**

- Conocer el procesador de textos LaTeX, gratuito y con calidad profesional
- Aprender a instalar y usar LaTeX en mi ordenador
- Usar LaTeX para generar documentos de formatos habituales (libros y artículos)
- Usar LaTeX para hacer presentaciones (pósters y presentaciones con *beamer*)
- introducción al BibTeX para gestión de la bibliografía

**DESARROLLO CONCEPTUAL**

Durante los estudios universitarios tendrá que realizar diversos documentos para su posterior presentación y en muchos casos evaluación. Por otra parte también es cierto que se da por hecho que cualquier persona que haya superado unos estudios universitarios (o incluso que esté realizando los últimos cursos de dichos estudios) debe ser capaz de realizar documentos impresos (o electrónicos) con una calidad de presentación profesional. En un pasado no demasiado lejano, y por tanto familiar a muchos estudiantes de esta universidad, se consideraba como algo normal realizar estos documentos “a máquina” o incluso “a mano”. Sin embargo, en la actualidad el estándar comúnmente aceptado es que *(casi) cualquier documento que vayamos a presentar debe haber sido realizado con un ordenador por medio de un **procesador de textos***. Aparece entonces la cuestión fundamental:

***¿Qué procesador de textos debo usar?***

La respuesta lógica es que debemos usar **el mejor** procesador que tengamos a nuestra disposición, siempre y cuando su precio caiga dentro de lo que nos podemos permitir ¡claro! Pero, *¿y si resulta que el mejor procesador del mercado es además gratuito?* Con los *procesadores de texto* sucede lo mismo que viene sucediendo en los últimos años con los *sistemas operativos* (ya saben, ese programa que nos permite usar el ordenador): existe abundante **software libre** (gratuito) en la red con una calidad **incomparablemente superior** al software comercial, cuyo uso a pesar de todo sigue estando (incomprensiblemente) bastante extendido.

Para centrarnos un poco mencionaremos que el mejor procesador de textos que existe es el **LaTeX**, objeto de esta nota. Paradójicamente, por motivos que sería largo explicar esta nota ha sido elaborada con otro procesador de textos mucho peor. Les pido por tanto disculpas por la mala presentación del documento.

El LaTeX está muy extendido en el mundo de las publicaciones científicas, especialmente (pero no exclusivamente) en los campos de *matemáticas, física, computación e ingeniería*. En la actualidad la inmensa mayoría de las publicaciones científicas de esos campos se escriben en LaTeX, tanto en lo referente a libros de texto como a publicaciones científicas periódicas. La superioridad de la calidad de presentación de los documentos escritos en LaTeX en comparación con los realizados con los

procesadores de texto comerciales habituales está fuera de toda duda, basta con una breve búsqueda en internet para obtener abundante documentación en este sentido.

### ***Entonces ¿tengo que aprender LaTeX forzosamente?***

Por supuesto que no. Lo que *sí* tiene que aprender forzosamente para estar dentro de lo que se considera estándar es a manejar *algún* procesador de textos, el que Vd. elija, y debe saber usarlo lo suficientemente bien como para realizar documentos con una presentación buena. De lo contrario está fuera de lo que se considera estándar. Si por ejemplo realiza sus documentos “a mano” y cuando se pide un archivo PDF los escanea y guarda en PDF (parece mentira, pero hay gente que hace eso), en ese caso estaría haciendo algo por debajo del estándar actual, es decir, estaría dando una *mala impresión*, independientemente de la calidad del contenido de dicho documento. Un estudiante universitario debe ser capaz de usar un procesador de textos y realizar documentos con una calidad de presentación buena. En el caso de estudiantes de ciencias esos documentos incluirán con mucha frecuencia fórmulas matemáticas, gráficas, tablas, referencias bibliográficas, enlaces a páginas web, etc., de modo que es necesario aprender a hacer todo eso. *Qué procesador de texto usar* es algo que cada uno debemos elegir de manera personal, de la misma forma como elegimos qué tipo de ordenador usar y con qué sistema operativo.

El objeto de esta nota es presentar el **LaTeX** como una posible elección. No como una posible elección cualquiera, sino como una elección que recomendamos fuertemente.

### ***No es posible que todo sean ventajas ¿qué problema tiene el LaTeX?***

Efectivamente, algún problema debe tener. Los principales inconvenientes del LaTeX son dos: (primero) ***Hay que aprender a usarlo***. No es que sea muy complicado, pero hay que dedicar un poco de tiempo a aprender cómo funciona, al menos a nivel básico. (segundo) ***Es un poco más rígido que los procesadores de texto tradicionales***. En LaTeX no se puede empezar a escribir un documento (una hoja en blanco) sin más, en LaTeX tenemos que saber de antemano *qué tipo de documento* queremos escribir. Los tipos de documento más habituales son *artículo* o *libro* (hay más, pero esos son los que más se usan). En cada uno estos casos comenzaríamos el documento LaTeX con una *instrucción* del estilo de:

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

o

```
\documentclass[11pt,twoside,a4paper]{book}.
```

o algo similar. Al hacer esto estamos diciendo al procesador LaTeX que vamos a escribir un documento con una determinada *estructura* (posiblemente con un *título* y *autor*, *capítulos*, *secciones*, etc.) y en LaTeX sucede que introducir variaciones que se salgan de la norma en esa estructura seleccionada es complicado, por eso comentábamos antes que el LaTeX “es más rígido” en su funcionamiento que los procesadores de texto tradicionales.

Si el documento que quiere realizar no tiene ninguna estructura, es decir, si lo que está buscando es la versión electrónica de una verdadera *hoja en blanco*, entonces LaTeX no es la mejor elección (a menos que ya sea un experto en LaTeX, ¡claro!). En ese caso la mejor elección sería un procesador de textos tradicional, como el **LibreOffice**, que es gratuito y funciona extraordinariamente bien en cualquier sistema operativo.

Si por el contrario lo que quiere escribir es un *libro* o un *artículo*, lo más probable (lo más lógico y lo más sensato) es que Vd. quiera ceñirse al estilo y formato que le diga su editorial (el cual especifica cosas como márgenes, tipos de letra, espaciados, etc., etc., etc.) de modo que Vd. pueda ***centrarse en el contenido*** del documento, no en su *forma*. En ese caso LaTeX es la mejor elección. Precisamente esa es la filosofía del LaTeX, que puede resumirse en la conocida frase: “deje el ***diseño de estilos*** de documentos a los profesionales de diseño de estilos de documentos, y céntrese en el ***contenido*** de su documento”.

Las instrucciones que acabamos de escribir más arriba nos permiten presentar el tema de las *instrucciones en LaTeX*. En la palabra “\documentclass” la barra “\” indica al procesador LaTeX que lo que viene a continuación no es texto normal del documento, sino una *instrucción* en LaTeX, es decir, una orden o *comando* que especifica algo sobre cómo debe formatearse el documento. En este caso la instrucción es “\documentclass” que, como su propio nombre indica, es el comando que define el tipo, o clase, de documento que vamos a escribir, que queda especificado por el argumento “book” o “article” que va entre llaves al final de la instrucción. Los otros argumentos, los que van entre corchetes, son argumentos opcionales (es decir, se pueden omitir), e indican algunas opciones (como tamaño de la fuente, tamaño de la página, etc.). Todos los documentos en LaTeX comienzan con la instrucción “\documentclass{...}”, seguida de otras instrucciones entre las que figura \begin{document} y terminan con la instrucción \end{document} con un significado bastante obvio. En la práctica no es necesario conocer estas instrucciones, ya que siempre podemos disponer de “plantillas de documentos” (o *templates*) que ya las incluyen y que podemos emplear como punto de partida para nuestros documentos.

### ¿Cómo se trabaja en LaTeX?

LaTeX es parecido a un *lenguaje de programación*. Para elaborar documentos en LaTeX lo que se hace es escribir en un archivo de *texto plano* el *código* LaTeX que posteriormente, al ser interpretado por el programa LaTeX, genera finalmente el documento formateado. Aquí es donde está la complicación de este lenguaje, aunque realmente es menos de lo que parece. Es cierto que para poder comenzar hay una serie de instrucciones de código LaTeX que es necesario conocer. Esto puede parecer complicado, pero de hecho es bastante sencillo. Principalmente porque siempre escribimos el mismo tipo de documentos (libros, artículos, ...) con el mismo tipo de ingredientes (texto, ecuaciones, figuras, tablas), de modo que al final solo necesitamos aprender unas pocas instrucciones, siempre las mismas. Aparte de eso está la web, una fuente de información virtualmente infinita, donde podremos localizar instantáneamente cualquier instrucción LaTeX que necesitemos usar y que no conozcamos, incluyendo ejemplos de uso, plantillas e incluso documentos completos que podemos emplear como puntos de partida para generar los nuestros (cambiando el texto, conservando las instrucciones).

La situación es distinta si queremos ser usuarios de LaTeX a nivel profesional, como por ejemplo el responsable de la hoja de estilos de una publicación periódica (como *Physical Review Letters*, formateada en LaTeX con la hoja de estilos REVTeX) o de una editorial (como Springer-Verlag o Cambridge University Press). Para un uso de LaTeX a ese nivel hay que ser un verdadero experto, para lo cual es necesario invertir cierto tiempo de estudio. Para generar documentos a nivel de usuario normal el tiempo de aprendizaje es mínimo, además se amortiza en seguida si tiene en cuenta todo el tiempo que se va a ahorrar en “mover cajitas y/o márgenes etc.” que se descontrolan fácilmente en los procesadores de textos habituales (...).

### ¿Cómo se trabaja en LaTeX? (ahora sí)

Existen otras alternativas más directas, pero el esquema *clásico* de trabajo en LaTeX es el siguiente:

1. Tener LaTeX instalado en el ordenador
2. Escribir el código en un archivo de *texto plano*. Suele ponerse la extensión “tex” a estos archivos de texto plano, p. ej.:

mi-archivo.tex

3. Enviar ese código al intérprete LaTeX mediante el comando “latex” en *línea de comandos*

```
>> latex mi-archivo.tex
```

Eso genera el archivo

mi-archivo.dvi

(de tipo DVI), que contiene el documento formateado para que podamos examinarlo. También genera una serie de archivos (.log, etc.) con diversa información sobre el proceso de generar el DVI. De momento nos podemos olvidar de ellos.

4. A medida que vamos generando el documento, escribiendo el texto y las instrucciones LaTeX necesarias en el archivo de texto plano `mi-archivo.tex`, lo vamos “compilando” por medio del comando “`latex`”, y vamos examinando el documento formateado (`mi-archivo.dvi`) para ver si todo va saliendo como se espera, y si el contenido está bien o hay que modificarlo.

5. Una vez finalizado el documento el último paso es generar el archivo final en PDF, el cual posteriormente podemos enviar al destinatario correspondiente (p. ej. a la “entrega de trabajos” del curso virtual de alguna asignatura). Para ello empleamos el comando “`dvipdf`” en línea de comandos

```
>> dvipdf mi-archivo.dvi
```

esto nos genera finalmente el archivo

`mi-archivo.pdf`

con el documento final formateado y listo para ser enviado a sus potenciales lectores.

Como comentábamos antes esa es la *manera clásica* de usar LaTeX, exagerando un poco hay quien diría que es *como se hacía antiguamente*. Actualmente es posible que muchos usuarios preferirían emplear el comando “`pdflatex`”, el lugar de “`latex`”, lo cual genera directamente el documento en PDF y nos evita pasar por el archivo DVI intermedio. De todas formas la anterior sigue siendo una forma de trabajo bastante habitual y práctica, y a día de hoy muchos usuarios de LaTeX siguen generando sus documentos de esta manera.

Según decíamos en el punto segundo del párrafo anterior, en LaTeX realizamos los documentos en archivos de *texto plano*, en los que escribimos el texto del documento junto con las instrucciones LaTeX necesarias.

### *¿Qué es exactamente y cómo se escribe un archivo de texto plano?*

Un archivo de *texto plano* es un archivo que contiene exclusivamente *caracteres estándar ASCII*, es decir, letras, números, signos estándar de puntuación, espacios y saltos de línea, etc., pero no caracteres no estándar (como p. ej. letras griegas o símbolos matemáticos no estándar), ni tampoco instrucciones de formato, es decir, instrucciones sobre cómo debe mostrarse ese texto (aparte de instrucciones totalmente básicas, como *espacios* o *saltos de línea*). Los archivos de texto plano contienen por tanto una serie de *caracteres*, escritos uno tras otro y separados por signos de puntuación estándar, espacios y saltos de línea, pero no contienen elementos complejos como textos resaltados o escritos con distintos tipos de fuente, distintos colores, tablas, figuras, etc. Los archivos de texto plano son los que generalmente se emplean para escribir **código** en cualquier lenguaje de programación, y también en LaTeX, y contienen exclusivamente el conjunto de instrucciones (es decir, el código) del programa que sea, o en el caso de archivos tipo TEX contienen el conjunto de instrucciones LaTeX que genera por medio del comando “`latex`” el documento formateado final.

Para escribir archivos de texto plano, ya sea para hacer un documento en LaTeX o para escribir un programa en el lenguaje que sea, se emplean unos programas denominados **editores de texto**. Es muy importante saber diferenciar entre un **editor de texto** y un **procesador de texto**. Como su propio nombre indica, lo único que permiten hacer los *editores de texto* es *crear, mostrar y modificar* archivos de texto plano, mientras que los *procesadores de texto* tradicionales son programas bastante más complejos que permiten escribir documentos en los que, aparte del texto, existen instrucciones sobre cómo debe mostrarse dicho texto. Existen multitud de editores de texto, algunos de ellos son los siguientes: *gedit*, *nedit*, *kwrite*, *kate*, *emacs*, *xemacs*, *vi*, *vim*, etc. Para sistemas con Windows, aparte de los anteriores, puede usarse el *notepad2* (por favor, no usen el *notepad*).

Por tanto, para trabajar en LaTeX (según el *esquema clásico* que decíamos antes) tendremos abiertas simultáneamente tres ventanas:

1. una ventana con el *archivo de texto* plano donde escribimos el código de nuestro documento
2. una consola de *línea de comandos* donde por medio de la orden  
`latex mi-archivo.tex`  
generamos el DVI
3. una ventana con el archivo DVI donde vamos viendo el aspecto del documento formateado

Como decíamos antes, actualmente existen otras formas de trabajo más directas, que permiten hacer todo eso en una única ventana, de una manera bastante parecida al modo de funcionamiento de los procesadores de texto habituales. Una posibilidad es usar TeXmacs, o LyX, que permiten escribir LaTeX casi sin darse cuenta de que estamos escribiendo LaTeX. De todas formas el método tradicional que exponemos aquí nos parece recomendable por ser más potente, y también por que nos permite tener mayor control sobre el documento que estamos realizando.

En la web existen multitud de páginas con información útil sobre LaTeX a cualquier nivel, desde simplemente saber un poco qué es todo eso hasta el nivel de experto más avanzado y todos los niveles intermedios. En la sección de REFERENCIAS se incluyen algunos enlaces que completan esta presentación. La primera referencia que citamos es la página de LaTeX en wikipedia, muy recomendable para tener una visión global sobre LaTeX y su funcionamiento. La segunda referencia es el clásico “*The Not So Short Introduction to LaTeX*” de T. Oetiker y cols., que incluye un curso práctico acelerado sobre el lenguaje LaTeX. Por medio de este curso en poco más de dos horas de trabajo (en 157 minutos para ser exactos) **sabrás todo lo necesario** para escribir documentos en LaTeX incluyendo figuras, tablas, ecuaciones, bibliografía, etc. Mi impresión personal es que la inmensa mayoría de los usuarios de este lenguaje hemos aprendido con “*The Not So Short Introduction to LaTeX*”.

## EJEMPLOS

### Instalación de LaTeX

Ejemplos de instalación del programa LaTeX en sistemas con Linux (para instalar LaTeX en ordenadores con Windows vea el programa MiKTeX citado en las REFERENCIAS)

#### 1. Instalación en Linux debian o ubuntu

Ejecute en línea de comandos el siguiente código:

```
sudo apt-get install texlive
```

#### 2. Instalación en Linux fedora

Ejecute en línea de comandos el siguiente código:

```
sudo yum install texlive
```

### Ejemplos de documentos básicos

Todos estos ejemplos se han tomado de la web (wikipedia el primero y <http://www.rpi.edu/dept/arc/training/latex/Examples/exbook.tex> el segundo). En la web pueden localizarse fácilmente muchos más.

### 1. Ejemplo de artículo (creamos un archivo de texto plano con el siguiente código):

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system for the \TeX{} typesetting program. It offers
  programmable desktop publishing features and extensive facilities for automating most
  aspects of typesetting and desktop publishing, including numbering and cross-referencing,
  tables and figures, page layout, bibliographies, and much more. \LaTeX{} was originally
  written in 1984 by Leslie Lamport and has become the dominant method for using \TeX; few
  people write in plain \TeX{} anymore. The current version is \LaTeXe.
  % This is a comment, not shown in final output.
  % The following shows typesetting power of LaTeX:
  \begin{align}
    E_0 &= mc^2 && \backslash
    E &= \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}}
  \end{align}
\end{document}
```

### 2. Ejemplo de libro (creamos un archivo de texto plano con el siguiente código):

```
\documentclass[11pt]{book}           % Book class in 11 points
\parindent0pt \parskip10pt          % make block paragraphs
\raggedright                          % do not right justify

\title{\bf An Example of Book Class} % Supply information
\author{for \LaTeX\ Class}           % for the title page.
\date{\today}                        % Use current date.

% Note that book class by default is formatted to be printed back-to-back.
\begin{document}                    % End of preamble, start of text.
\frontmatter                        % only in book class (roman page #s)
\maketitle                          % Print title page.
\tableofcontents                    % Print table of contents
\mainmatter                         % only in book class (arabic page #s)
\part{A Part Heading}              % Print a "part" heading
\chapter{A Main Heading}           % Print a "chapter" heading
Most of this example applies to \texttt{article} and \texttt{book} classes
as well as to \texttt{report} class. In \texttt{article} class, however,
the default position for the title information is at the top of
the first text page rather than on a separate page. Also, it is
```



not usual to request a table of contents with `\texttt{article}` class.

```
\section{A Subheading}      % Print a "section" heading
```

The following sectioning commands are available:

```
\begin{quote}              % The following text will be
```

```
part \\\                   % set off and indented.
```

```
chapter \\\                % \\\ forces a new line
```

```
section \\\
```

```
subsection \\\
```

```
subsubsection \\\
```

```
paragraph \\\
```

```
subparagraph
```

```
\end{quote}                % End of indented text
```

But note that---unlike the `\texttt{book}` and `\texttt{report}` classes---the `\texttt{article}` class does not have a ```chapter"` command.

```
\end{document}            % The required last line
```

### Ejemplos de otros tipos de documentos (con *beamer*)

Para compilar estos ejemplos es necesario tener cargado el paquete ***beamer***, que es un ingrediente adicional de LaTeX incluido por defecto en las versiones modernas. Beamer permite hacer presentaciones y pósters con LaTeX con un aspecto realmente bueno. Para emplear beamer no se compila el documento con el comando "latex" (como comentábamos antes) sino con el comando "pdflatex". Al hacer esto pdflatex genera directamente el documento final (póster o presentación) en PDF, sin necesidad de pasar por el archivo intermedio DVI.

#### 1. Ejemplo de póster (creamos un archivo de texto plano con el siguiente código, tomado de <http://tug.ctan.org/macros/latex/contrib/beamerposter/example.tex>):

```
\documentclass[final]{beamer} % beamer 3.10: do NOT use option
```

```
% hyperref={pdfpagelabels=false} !
```

```
%\documentclass[final,hyperref={pdfpagelabels=false}]{beamer}% beamer 3.07: get rid of  
beamer warnings
```

```
\mode<presentation> { %% check http://www-i6.informatik.rwth-  
aachen.de/~dreuw/latexbeamerposter.php for examples
```

```
\usetheme{Berlin}  %% you should define your own theme e.g. for big headlines using  
your own logos
```

```
}
```

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

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

```
\usepackage{amsmath,amsthm,amssymb,latexsym}
```

```
%\usepackage{times}\usefonttheme{professionalfonts}
```

```
% times is obsolete
```

```
\usefonttheme[onlymath]{serif}
```

```
\boldmath
```

```

\usepackage[orientation=portrait,size=a0,scale=1.4,debug]{beamerposter} % e.g.
for DIN-A0 poster
%\usepackage[orientation=portrait,size=a1,scale=1.4,grid,debug]{beamerposter}
% e.g. for DIN-A1 poster, with optional grid and debug output
%\usepackage[size=custom,width=200,height=120,scale=2,debug]{beamerposter}
% e.g. for custom size poster
%\usepackage[orientation=portrait,size=a0,scale=1.0,printer=rwth-glossy-
uv.df]{beamerposter} % e.g. for DIN-A0 poster with rwth-glossy-uv printer check
% ...
%
\title[Fancy Posters]{Making Really Fancy Posters with \LaTeX}
\author[Dreuw \& Deselaers]{Philippe Dreuw and Thomas Deselaers}
\institute[RWTH Aachen University]{Human Language Technology and Pattern
Recognition,RWTH Aachen University}
\date{Jul. 31th, 2007}
\begin{document}
\begin{frame}{}
\vfill
\begin{block}{\large Fontsizes}
\centering
{\tiny tiny}\par
{\scriptsize scriptsize}\par
{\footnotesize footnotesize}\par
{\normalsize normalsize}\par
{\large large}\par
{\Large Large}\par
{\LARGE LARGE}\par
{\veryHuge veryHuge}\par
{\VeryHuge VeryHuge}\par
{\VERYHuge VERYHuge}\par
\end{block}
\vfill
\end{frame}
\end{document}

```

**2. Ejemplo de presentación (creamos un archivo de texto plano con el siguiente código, tomado de <http://en.wikibooks.org/wiki/LaTeX/Presentations>):**

```

\documentclass[xetex,mathserif,serif]{beamer}
\begin{document}
\begin{frame}
\frametitle{This is the first slide}
%Content goes here
\end{frame}
\begin{frame}
\frametitle{This is the second slide}
\framesubtitle{A bit more information about this}
%More content goes here
\end{frame}
% etc

```



```
\end{document}
```

## EJERCICIO DE AUTOCOMPROBACIÓN

Modifique el contenido de los ejemplos proporcionados para generar a partir de ellos sus propios documentos con formatos artículo, libro, póster o presentación.

Si le surgen dificultades empleando beamer deje este tema para más adelante, espere a estar razonablemente familiarizado con el uso de LaTeX en los tipos de documentos sencillos como “book” y sobre todo el básico “article”, antes de hacer presentaciones o pósters con beamer.

## REFERENCIAS:

- <http://en.wikipedia.org/wiki/LaTeX>
- <https://tobi.oetiker.ch/lshort/lshort.pdf>
- <http://en.wikibooks.org/wiki/LaTeX>
- [http://en.wikibooks.org/wiki/LaTeX/Sample\\_LaTeX\\_documents](http://en.wikibooks.org/wiki/LaTeX/Sample_LaTeX_documents)
- <http://miktex.org/> (para usuarios de Windows)
- <http://www.latex-project.org/>
- <http://www.bibtex.org/> (**BibTeX**, para la gestión de las referencias)
- <http://en.wikipedia.org/wiki/BibTeX>
- [http://en.wikipedia.org/wiki/Beamer\\_\(LaTeX\)](http://en.wikipedia.org/wiki/Beamer_(LaTeX)) (**Beamer**, para presentaciones y pósters)

## AUTORES:

- Manuel Arias Zugasti
- Daniel Rodríguez Pérez