

# FAQ in L<sup>A</sup>T<sub>E</sub>X

A. Sztrajman

15 de noviembre de 2011



# Índice general

<b>1. Primeros Pasos</b>	<b>5</b>
Hello world! . . . . .	5
Compilación . . . . .	6
Diseción del documento . . . . .	7
Espacios e interlineado . . . . .	8
<b>2. Edición y Formato</b>	<b>11</b>
Tipografías . . . . .	11
Cómo hago para cambiar el tamaño de letra? . . . . .	12
Cómo hago para poner una nota al pie? . . . . .	13
Cómo hago para generar un índice? . . . . .	14
Cómo hago para generar una lista de items? . . . . .	15
Cómo hago para forzar el espaciado? . . . . .	17
<b>3. Ecuaciones</b>	<b>19</b>
Cómo hago para incluir fórmulas matemáticas? . . . . .	19
Cómo hago para hacer referencia en el texto a una ecuación? . . . . .	20
Comandos frecuentes . . . . .	21
Símbolos frecuentes . . . . .	23
<b>4. Imágenes</b>	<b>25</b>
Cómo hago para incluir imágenes en mi documento? . . . . .	25
<b>5. Otros tutoriales</b>	<b>27</b>



# Capítulo 1

## Primeros Pasos

### Hello world!

Las instrucciones y paradigmas de un lenguaje de programación son relativamente arbitrarios. Responden sí a cierto desarrollo histórico de los lenguajes en general –sus características son derivadas de lenguajes anteriores–, pero su elección de palabras reservadas<sup>1</sup> no es predecible. Por eso, la mejor forma de empezar con un lenguaje es hacerlo con el programa más sencillo que compila correctamente. El ejemplo ya famoso es `hello world`, un programa que imprime la frase “hello world!” en la pantalla (en este caso en una hoja de un documento PDF).

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

\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}

\begin{document}
Hello world!
\end{document}
```

`\documentclass` determina algunos parametros generales del documento. En primer lugar se configura el tamaño de letra, en este caso `11pt` (`10pt` por default). Además se determina el tipo de hoja en el cual queremos acomodar lo que escribimos (en este caso `A4`). Podemos agregar otras opciones como por ejemplo `twocolumn` si queremos un documento en 2 columnas. Finalmente determinamos el tipo de documento que queremos generar (`article` para papers, `report` para tesis, `book` para libros, etc.).

`\usepackage` nos permite agregar funcionalidades externas que no están incluidas en el compilador de  $\text{\LaTeX}$  (equivalente al `#include` de `C++`).

`babel` Este paquete setea el idioma del documento. En este caso le pasamos como opción `spanish`. Entre otras cosas es importante para que

---

<sup>1</sup>reserved words

separe bien las palabras en sílabas (L<sup>A</sup>T<sub>E</sub>X no justifica los textos).

**inputenc** Este paquete setea el encoding del documento. Generalmente conviene usar utf8. Con este encoding podemos usar caracteres con tilde, sin necesidad de usar \ ' como en la prehistoria.

**\begin{algo}** declara un entorno o environment de tipo "algo". En este caso, el entorno declarado es el documento mismo. Cada vez que abrimos un entorno, debemos cerrarlo (**\end{algo}**).

## Compilación

Sí, los archivos de L<sup>A</sup>T<sub>E</sub>X se compilan, en un proceso similar al de cualquier lenguaje de programación. La diferencia es que el resultado de la compilación no es un programa sino un documento, que puede generarse en distintos formatos (en general, PDF, DVI o PS).

Para compilar el ejemplo de la sección anterior, guardamos su contenido en un archivo `helloworld.tex` y ejecutamos `pdflatex helloworld.tex`. El resultado es una serie de archivos, entre ellos `helloworld.pdf`.

También podemos compilar con `latex helloworld.tex` que nos devuelve un archivo `helloworld.dvi`. Con `dvipdf helloworld.dvi` lo convertimos a formato `pdf`, aunque podemos visualizarlo en cualquiera de los dos formatos:

```
okular helloworld.dvi u okular helloworld.pdf  
(también podemos usar evince o acroread).
```

Lo más cómodo en general es trabajar con `pdflatex`, pero hay que tener en cuenta que `latex` y `pdflatex` son compiladores distintos de código en L<sup>A</sup>T<sub>E</sub>X (existen otros compiladores, como XeLaTeX), cada uno con sus características. En particular, algunos paquetes (ver `\usepackage` en la sección anterior) funcionan exclusivamente en uno u otro compilador.

## Diseción del documento

A medida que nuestro documento crece, se vuelve inevitable separarlo en secciones, capítulos, etc. La mayoría de las clases conocidas de L<sup>A</sup>T<sub>E</sub>X (`article`, `book`, `report`, etc.) cuentan con comandos específicos de seccionamiento. Estos son:

```
\chapter{nombre de capítulo}
\section{nombre de sección}
\subsection{nombre de subsección}
\subsubsection{nombre de subsubsección}
```

La numeración de todas estas categorías es automática, una ventaja impagable cuando se trabaja con documento grandes: si queremos agregar un nueva sección en el medio de nuestro documento, no tenemos que reenumerar todas las secciones subsiguientes.

Como ejemplo de la sintaxis de `\subsection`, mostramos el código del comienzo de esta sección. En este caso estamos usando un asterisco en el comando, para indicar que no queremos numerar esta subsección:

```
\subsection*{Disecion del documento}
A medida que nuestro documento crece, se vuelve inevitable separarlo
en secciones, capitulos, etc. La mayoría de las clases conocidas de
\LaTeX{} (\texttt{article}, \texttt{book}, \texttt{report}, etc.)
cuentan con comandos especificos de seccionamiento. Estos son...
```

Una vez tenemos definidos todos los capítulos y secciones, es muy sencillo generar un índice de nuestro documento (hay una descripción de cómo hacerlo en el capítulo 2).

## Espacios e interlineado

El compilador de  $\text{\LaTeX}$  ignora todos los espacios que podamos dejar entre dos palabras, exceptuando el primero. Es decir que las palabras en el documento siempre están separadas por un único espacio. Algo similar ocurre con el interlineado: recién si dejamos una línea en blanco entre dos bloques de texto,  $\text{\LaTeX}$  los interpretará como dos párrafos distintos. Por eso

Todos los espacios y los retornos  
de carro (enter) adicionales son  
ignorados por  
el compilador.

No así si dejamos una línea en blanco entre dos párrafos.

Todos los espacios y los retornos de carro (enter) adicionales son ignorados  
por el compilador.

No así si dejamos una línea en blanco entre dos párrafos.



## Capítulo 2

# Edición y Formato

### Tipografías

En  $\LaTeX$  hay tres familias distintas de tipografías: roman, sans serif y monospace. Las tipografías romanas (e.g. Times) son las más antiguas y más fáciles de leer en papel. El aliasing de los LCDs hace que en general no sean las más cómodas de leer en un monitor, y en este caso quizá preferamos las tipografías sans serif (e.g. Arial). La cedilla (serif) es una marca al final del trazo de un grafema, que le da un aspecto más decorado. Las tipografías sans serif, como su nombre lo indica, no tienen esta decoración que es la que se ve mal en los monitores. Las tipografías monospace (e.g. Courier) son muy posteriores a las de las otras dos familias, y se caracterizan por el hecho de que todos los grafemas tienen el mismo ancho.

Además de la elección de familias, que por default nos permite elegir entre tres tipos de letra, podemos elegir varios estilos distintos:

Comando	Comando abreviado	Descripción
<code>\textnormal{...}</code>	<code>{\normalfont ...}</code>	Tipografía normal
<code>\textbf{...}</code>	<code>{\bfseries ...}</code>	<b>Texto en negrita</b>
<code>\textit{...}</code>	<code>{\it }</code>	<i>Texto en itálica</i>
<code>\underline{...}</code>		<u>Texto subrayado</u>
<code>\emph{...}</code>	<code>{\em ...}</code>	<i>Texto con énfasis</i>
<code>\textrm{...}</code>	<code>{\rm ...}</code>	Tipografía romana
<code>\textsf{...}</code>	<code>{\sf ...}</code>	Tipografía sans serif
<code>\texttt{...}</code>	<code>{\tt ...}</code>	Tipografía teletype (monospace)
<code>\textsl{...}</code>	<code>{\sl }</code>	<i>Tipografía inclinada. Similar a la itálica</i>
<code>\textsc{...}</code>	<code>{\sc }</code>	TEXTO EN MAYÚSCULAS PEQUEÑAS (SMALL CAPS)
<code>\uppercase{...}</code>		TEXTO EN MAYÚSCULA

En general hay dos formas de invocar los comandos que cambian el estilo de letra, como se muestra en la tabla. Por ejemplo, si queremos escribir

To **boldly** go...

podemos hacerlo de las siguientes dos formas:

To `\textbf{boldly}` go...  
To `{\bf boldly}` go...

## Cómo hago para cambiar el tamaño de letra?

Las tipografías en L<sup>A</sup>T<sub>E</sub>X vienen en una serie de tamaños predeterminados

<code>\tiny</code>	Texto de ejemplo
<code>\scriptsize</code>	Texto de ejemplo
<code>\footnotesize</code>	Texto de ejemplo
<code>\small</code>	Texto de ejemplo
<code>\normalsize</code>	Texto de ejemplo
<code>\large</code>	Texto de ejemplo
<code>\Large</code>	Texto de ejemplo
<code>\LARGE</code>	Texto de ejemplo
<code>\huge</code>	Texto de ejemplo
<code>\Huge</code>	Texto de ejemplo

Para cambiar el tamaño de texto de una cadena específica, podemos encerrarla entre llaves y llamar al comando de la tabla, como en el ejemplo

```
{\footnotesize 0}{\normalsize 1}{\large 2}{\Large 3}{\LARGE 4}{\huge 5}
```

012345

También podemos encerrar un bloque de texto usando como entorno el tamaño de letra

```
\begin{large}
Texto en entorno large
\end{large}
```

Texto en entorno large

## Cómo hago para poner una nota al pie?

Las notas al pie son muy fáciles de usar en L<sup>A</sup>T<sub>E</sub>X. La sintaxis se puede ver en el ejemplo a continuación

```
Pongamos una nota al pie\footnote{listo!}.
```

El comando `\footnote` aparece pegado al texto, para que la numeración de la nota la pie, que obviamente es automática, aparezca sin espacios. El resultado es

Pongamos una nota al pie<sup>1</sup>.

---

<sup>1</sup>listo!

## Cómo hago para generar un índice?

Una vez que tenemos definida la separación en capítulos y secciones de nuestro documento (como se describió en el capítulo 1), podemos generar un índice de manera automática, utilizando dos instrucciones:

- `\makeindex` se ejecuta antes del comienzo del entorno `document`. Se ocupa de recolectar la información del documento sobre secciones.
- `\tableofcontents` inserta el índice en el punto del documento donde se ejecuta. Se debe insertar adentro del entorno `document`.

Adicionalmente se pueden generar otros índices, por ejemplo de figuras y tablas insertados en el documento. Los comandos correspondientes son `\listoffigures` y `\listoftables`.

## Cómo hago para generar una lista de items?

El entorno básico para generar listas es `itemize`. Su uso se ve en el siguiente ejemplo

```
Tres formas basicas de listar items:
\begin{itemize}
  \item itemize
  \item enumerate
  \item description
\end{itemize}
Tambien podemos listar con nuestros propios simbolos:
\begin{itemize}
  \item[a] itemize
  \item[b] enumerate
  \item[c] description
\end{itemize}
```

Tres formas básicas de listar items:

- itemize
- enumerate
- description

También podemos listar con nuestros propios símbolos:

- a) itemize
- b) enumerate
- c) description

Con el entorno `enumerate` los items se numeran automáticamente:

```
\begin{enumerate}
  \item itemize
  \item enumerate
  \item description
\end{enumerate}
```

1. itemize
2. enumerate
3. description

Finalmente con `description` hacemos una lista en formato diccionario:

```
\begin{description}
  \item[Copyright] Figura legal que permite defender del futuro
  aquello que hemos robado del pasado.
```

```
\end{description}
```

**Copyright** Figura legal que permite defender del futuro aquello que hemos robado del pasado.

## Cómo hago para forzar el espaciado?

Si bien decirle a  $\text{\LaTeX}$  dónde debe ubicar el texto suele ser una mala idea, y a veces una señal de que estamos pensando “a lo Word”, en ocasiones es simplemente necesario. Si queremos forzar una nueva línea, podemos hacerlo con  $\backslash$ .

Para desplazarnos horizontal y verticalmente contamos con los comando  $\backslash$ hspace y  $\backslash$ vspace. El parámetro que llevan ambos es una distancia, que se puede especificar en distintas unidades:

$\backslash$ hspace{1cm} un centímetro hacia la derecha.

$\backslash$ hspace{-1.5cm} un centímetro y medio hacia la izquierda.

$\backslash$ vspace{1em} el tamaño de una **m** en la tipografía actual, hacia abajo.

$\backslash$ vspace{-20mm} veinte milímetros hacia arriba.

## Capítulo 3

# Ecuaciones

Para usar la mayor parte de los entornos y símbolos matemáticos disponibles, es necesario haber incluido al principio del código los paquetes `amsmath` y `amssymb`.

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

### Cómo hago para incluir fórmulas matemáticas?

La integración de ecuaciones en el texto es el punto más fuerte de  $\text{\LaTeX}$ , debido a que  $\text{\TeX}$  fue concebido específicamente para la edición de libros de matemática. La notación de símbolos matemáticos es muy intuitiva (se puede ver en la sección **Comandos y símbolos más frecuentes**) y hoy en día se usa en casi todos lados: sin ir muy lejos, las fórmulas de la Wikipedia están escritas en este lenguaje.

Para escribir ecuaciones contamos con varios entornos. El más sencillo es `equation`, que pone una única fórmula y la numera, como en el ejemplo:

```
\begin{equation}
\nabla^2 \varphi = 0
\end{equation}
```

$$\nabla^2 \varphi = 0 \tag{3.1}$$

Si queremos evitar la numeración de la ecuación, podemos cambiar el entorno por `equation*` (esto vale para la mayoría de los entornos), o bien incluir el comando `\nonumber` en la ecuación específica que no queramos numerar. En el siguiente ejemplo usamos el entorno `align` que nos permite escribir varias ecuaciones juntas y alinearlas con el carácter `&`:

```

\begin{align}
\nabla \cdot \vec{E} &= \rho \\
\nabla \cdot \vec{B} &= 0 \nonumber
\end{align}

```

de paso solamente numeramos una de las ecuaciones

$$\begin{aligned} \nabla \cdot \vec{E} &= \rho \\ \nabla \cdot \vec{B} &= 0 \end{aligned} \tag{3.2}$$

Para usar el entorno `align` nos vamos a tener que acordar de incluir el paquete `amsmath` al principio de nuestro documento:

```
\usepackage{amsmath}
```

Todos los entornos disponibles para insertar ecuaciones (hay varios más: `eqnarray`, `flalign`, etc., pero los dos vistos son los más útiles) generan un espacio separado para las fórmulas, pero a veces vamos a querer incluir ecuaciones adentro de un párrafo, como parte continuada de un texto. Para esto simplemente tenemos que encerrar nuestras fórmulas entre `$`:

```
Simplificando  $\frac{\sin{x}}{x} = 0$  obtenemos  $\sin{} = 0$ .
```

Simplificando  $\frac{\sin x}{x} = 0$  obtenemos  $\sin = 0$ .

## Cómo hago para hacer referencia en el texto a una ecuación?

Adentro de los entornos matemáticos (`equation`, `align`, etc.) es posible definir un parámetro llamado `\label`. Una vez definido, este parámetro se convierte en el “nombre” de nuestra ecuación. Cada vez que queramos referirnos a nuestra ecuación (cualquiera sea su número), lo vamos a hacer a través de la etiqueta (*label*) que le definimos, con el comando `\ref{}`:

```
...por lo tanto obtenemos la expresion
\begin{equation}
\partial_u \partial^u \varphi = 0
\label{demudemu}
\end{equation}
La ecuacion \ref{demudemu} es lo que conocemos como...
```

```
...por lo tanto obtenemos la expresion


$$\partial_u \partial^u \varphi = 0 \tag{3.3}$$


La ecuacion 3.3 es lo que conocemos como...
```

## Comandos frecuentes

<code>\frac{a}{b}</code>	$\frac{a}{b}$
<code>\frac{\partial \psi}{\partial x}</code>	$\frac{\partial \psi}{\partial x}$
<code>\partial_u \partial^u</code>	$\partial_u \partial^u$
<code>\sum_{i=0}^n</code>	$\sum_{i=0}^n$
<code>\prod_{i=0}^n</code>	$\prod_{i=0}^n$
<code>\int_a^b</code>	$\int_a^b$
<code>\oint_a^b</code>	$\oint_a^b$
<code>\lim_{n \rightarrow \infty}</code>	$\lim_{n \rightarrow \infty}$
<code>\vec{w} \times \vec{v}</code>	$\vec{w} \times \vec{v}$
<code>\vec{w} \cdot \vec{v}</code>	$\vec{w} \cdot \vec{v}$
<code>\vec{\nabla} \psi</code>	$\vec{\nabla} \psi$
<code>\varphi^{\prime}</code>	$\varphi'$
<code>\langle \psi   \varphi_n \rangle</code>	$\langle \psi   \varphi_n \rangle$
<code>\underbrace{w \cdot v}_{=0}</code>	$\underbrace{w \cdot v}_{=0}$
<code>\overbrace{w \cdot v}^{=0}</code>	$\overbrace{w \cdot v}^{=0}$

<code>\stackrel{?}{=}</code>	$\stackrel{?}{=}$
<code>\sqrt{x}</code>	$\sqrt{x}$
<code>\cos{x}</code>	$\cos x$
<code>\sin{x}</code>	$\sin x$
<code>\tan{x}</code>	$\tan x$
<code>\ln{x}</code>	$\ln x$
<code>\frac{\text{m}}{\text{seg}}</code>	$\frac{\text{m}}{\text{seg}}$

## Símbolos frecuentes

Comando	Grafema
<code>\partial</code>	$\partial$
<code>\times</code>	$\times$
<code>\vec{w}</code>	$\vec{w}$
<code>\nabla</code>	$\nabla$
<code>\dot{x}</code>	$\dot{x}$
<code>\ddot{x}</code>	$\ddot{x}$
<code>\hat{x}</code>	$\hat{x}$
<code>\tilde{x}</code>	$\tilde{x}$
<code>\rightarrow</code>	$\rightarrow$
<code>\longrightarrow</code>	$\longrightarrow$
<code>\Rightarrow</code>	$\Rightarrow$
<code>\rightharpoonup</code>	$\rightharpoonup$
<code>\infty</code>	$\infty$
<code>\hbar</code>	$\hbar$
<code>\forall</code>	$\forall$
<code>\in</code>	$\in$
<code>\notin</code>	$\notin$
<code>\exists</code>	$\exists$
<code>\bot</code>	$\perp$
<code>\pm</code>	$\pm$
<code>\approx</code>	$\approx$
<code>\neq</code>	$\neq$
<code>\equiv</code>	$\equiv$
<code>\sim</code>	$\sim$
<code>\simeq</code>	$\simeq$
<code>\ll \gg</code>	$\ll \gg$
<code>\bigoplus</code>	$\bigoplus$
<code>\bigotimes</code>	$\bigotimes$
<code>\ast</code>	$*$
<code>\dots</code>	$\dots$

Comando	Grafema
<code>\alpha</code>	$\alpha$
<code>\beta</code>	$\beta$
<code>\delta</code>	$\delta$
<code>\gamma</code>	$\gamma$
<code>\epsilon</code>	$\epsilon$
<code>\varepsilon</code>	$\varepsilon$
<code>\eta</code>	$\eta$
<code>\theta</code>	$\theta$
<code>\kappa</code>	$\kappa$
<code>\lambda</code>	$\lambda$
<code>\mu</code>	$\mu$
<code>\nu</code>	$\nu$
<code>\pi</code>	$\pi$
<code>\rho</code>	$\rho$
<code>\sigma</code>	$\sigma$
<code>\tau</code>	$\tau$
<code>\phi</code>	$\phi$
<code>\varphi</code>	$\varphi$
<code>\chi</code>	$\chi$
<code>\psi</code>	$\psi$
<code>\omega</code>	$\omega$
<code>\Delta</code>	$\Delta$
<code>\Gamma</code>	$\Gamma$
<code>\Theta</code>	$\Theta$
<code>\Lambda</code>	$\Lambda$
<code>\Pi</code>	$\Pi$
<code>\Sigma</code>	$\Sigma$
<code>\Phi</code>	$\Phi$
<code>\Psi</code>	$\Psi$
<code>\Omega</code>	$\Omega$

## Capítulo 4

# Imágenes

### Cómo hago para incluir imágenes en mi documento?

La inclusión de imágenes en los documentos es probablemente el aspecto más flojo de  $\LaTeX$ . Para empezar, la lista de formatos de imagen que podemos insertar es acotada, y depende del compilador de  $\LaTeX$  que usemos. Como vimos en la sección **Primeros Pasos**, existen dos compiladores principales:

- Si compilamos con el comando `latex`, sólo podemos insertar imágenes en formato EPS (encapsulated postscript). Es un formato vectorial, que quiere decir que no pierde calidad al escalearlo. Pero no es común que tengamos nuestras imágenes en este formato, y no es posible en general convertir a EPS desde otros formatos.
- Si compilamos con el comando `pdflatex`, podemos introducir fácilmente imágenes en formato JPG, PNG y PDF, pero perdemos la capacidad de insertar imágenes en EPS. La solución es convertir nuestras imágenes EPS a PDF, que se puede hacer manualmente, o automáticamente con el paquete `epstopdf` (ver **Cómo hago para insertar imágenes EPS usando pdflatex?**).

Para insertar imágenes tenemos que incluir el paquete `graphicx`:

```
\usepackage{graphicx}
```

Cada vez que queramos incluir una imagen usamos el comando

```
\includegraphics{<archivo de imagen>}
```

al que le podemos agregar algunas opciones sobre la imagen:

<code>width</code>	ancho
<code>height</code>	alto
<code>scale</code>	factor de escala
<code>angle</code>	ángulo de rotación antihorario

Por ejemplo:

```
\includegraphics[width=16cm, height=9cm]{imagen.png}
```

Además podemos incluir la imagen en un entorno `figure` para asociarle un epígrafe:

```
\begin{figure}  
\includegraphics{<archivo de imagen>}  
\caption{<epigrafe>}  
\end{figure}
```

El método que estamos viendo es la forma standard de insertar imágenes en  $\text{\LaTeX}$ , pero está lejos de ser óptimo. La política del compilador con las figuras es colocarlas donde le parece, que generalmente corresponde al final del capítulo. Las imágenes pueden aparecer incluso en una sección distinta! En el caso general esto ayuda muy poco a darle continuidad a la lectura. Por suerte existe una forma de obligar al compilador a insertar las figuras exactamente donde queremos, hablando directamente con  $\text{\TeX}$ , el cerebro límbico de  $\text{\LaTeX}$ . La explicación completa de cómo hacerlo se encuentra en **Cómo hago para poner las imágenes y tablas exactamente donde yo quiero?**.

## Capítulo 5

# Otros tutoriales

Existen muchas fuentes de donde obtener ayuda en internet:

**The not so short introduction to  $\LaTeX$**  El Griffiths de  $\LaTeX$ ! Se entiende todo y está lleno de chistes internos. Probablemente el mejor texto para comenzar con este lenguaje. También sirve para el que ya sabe un poco y quiere profundizar en los temas básicos.  
<http://tobi.oetiker.ch/lshort/lshort.pdf>

**$\LaTeX$  Wikibook** Un reunte de trucos muy muy bueno, especialmente para el que ya sabe escribir en  $\LaTeX$  y necesita hacer algo específico, escrito por la comunidad de *geeks* de internet. Es muy recomendable la introducción “getting started”.  
<http://en.wikibooks.org/wiki/LaTeX>

**$\LaTeX$  Visual FAQ** Un documento pdf con muchos trucos de  $\LaTeX$  implementados. Si vemos algo que nos gusta en el documento podemos clickearlo y nos manda a la página donde se explica cómo hacerlo.  
<http://www.tex.ac.uk/tex-archive/info/visualFAQ/visualFAQ.pdf>