

L^AT_EX: Una herramienta para procesar textos utilizando Software Libre

Pablo L. De Nápoli

`pdenapo@dm.uba.ar`

¿Qué son $\text{T}_{\text{E}}\text{X}$ / $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$?

¿Qué es T_EX?

- TeX es un sistema de preparación de documentos creado por **Donald E. Knuth**.
- TeX no es un procesador de textos tradicional, sino que mas bien se parece a un lenguaje de programación.
- En lugar de ver nuestro documento tal como se verá impreso, debemos crear un archivo con instrucciones sobre como queremos que nuestro documento se vea.

Donald Knuth dixit



“TeX is intended for the creation of beautiful books, and especially for books that contain a lot of mathematics.”

(Donald Knuth, preface to the TeXBook)

TEXes software libre

Un programa es software libre si el usuario tiene (de acuerdo a la **licencia** del programa) las siguientes libertades o derechos:

- 0) La libertad de **ejecutar el programa**, con cualquier propósito.
- 1) La libertad de **estudiar cómo funciona el programa**, y **adaptarlo** a sus necesidades.
- 2) La libertad de **distribuir copias**, con lo que puede ayudar a su vecino.
- 3) La libertad de mejorar el programa y **hacer públicas las mejoras** a los demás, de modo que toda la comunidad se beneficie.

Ventajas de T_EX

- TeX es **software libre**.
- Excelente calidad final.
- Facilidad para incluir **fórmulas matemáticas**.
- TeX utiliza un **formato abierto**.
De hecho, los documentos en TeX son archivos de texto, que se pueden editar con nuestro editor favorito, y procesar con herramientas estándar.
- TeX está muy bien documentado.
Existe una amplia documentación que se suele incluir en todas las distribuciones de TeX.
- TeX es multiplataforma: se puede utilizar en GNU/Linux, MS-Windows (MikTeX) o Mac OS.

Dialectos de T_EX

Existen varios dialectos de T_EX (o mejor dicho distintos paquetes de macros) tales como:

- plain T_EX (TeX sin agregados).
- L^AT_EX, paquete de macros creado por **Leslie Lamport**.
- AmsTeX, dialecto de TeX creado por la A.M.S. (American Mathematical Society).
- AmsLaTeX (extensión de LaTeX de la AMS).

En esta charla veremos a modo de ejemplo como crear un documento en LaTeX (a mi criterio, el dialecto más sencillo de utilizar).

Ventajas de L^AT_EX

- LaTeX es mucho más fácil de aprender y de utilizar que Plain T_EX.

LaTeX es como un lenguaje de alto nivel para T_EX, mientras que Plain es el equivalente a un lenguaje de bajo nivel.

- El autor tiene que preocuparse más por el contenido que por el formato.
- Los documentos LaTeX tienen estructura.
- Existen conversores de LaTeX a diversos formatos (PDF, HTML, RTF, etc.)

¿Quiénes utilizan T_EX?

- La comunidad científica (matemática, física, computación científica).
- La industria editorial (principalmente para libros científicos o técnicos).
- Los proyectos de documentación de software libre:
 - GNU Texinfo.
 - The Linux Documentation Project (TLDP) utiliza Docbook, que a su vez utiliza TeX / LaTeX para generar versiones imprimibles de los documentos (por medio de jadetex).

Desventajas de TeX

- Curva de aprendizaje.
- En TeX/LaTeX no podemos ver el documento tal como va a verse cuando esté impreso mientras lo estamos escribiendo (antes de compilarlo).

Un primer documento en L^AT_EX

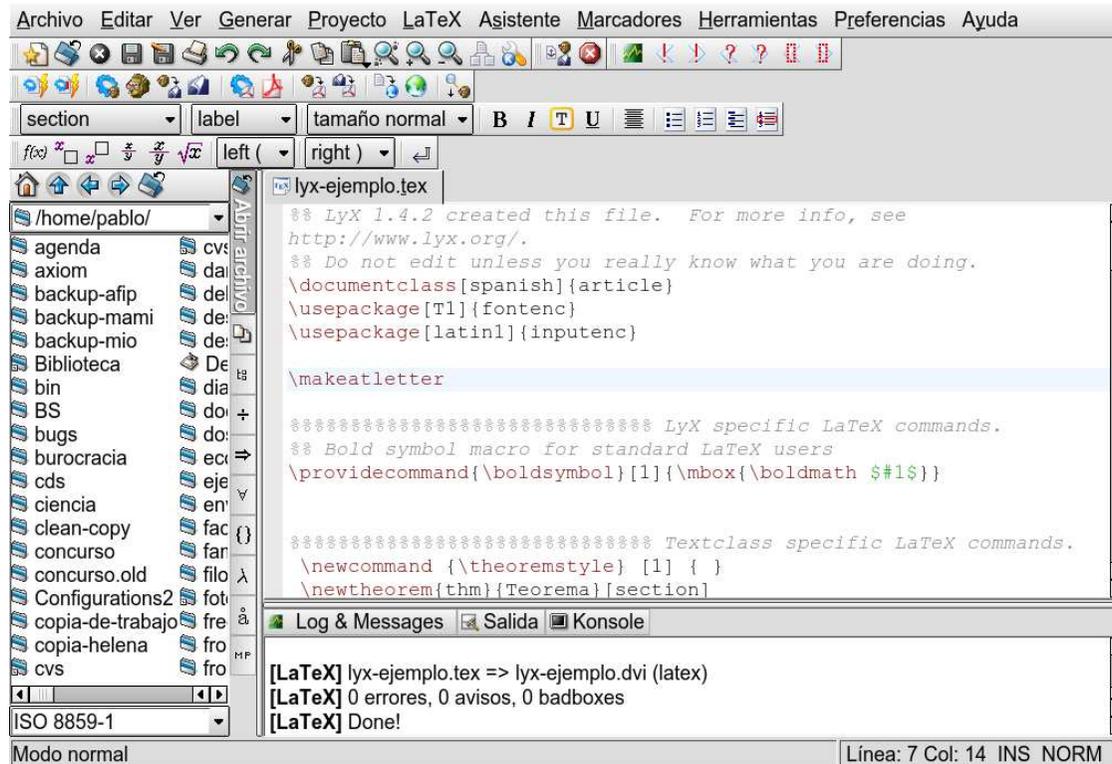
HolaMundo.tex

A modo de ejemplo, veamos como crear un documento sencillo en LaTeX.

Para ello creamos con nuestro editor de texto favorito un archivo **HolaMundo.tex**, que contenga las siguientes instrucciones (en rojo).

Podemos utilizar cualquier editor de texto, aunque existen editores especializados como **Kile** (un IDE para KDE en GNU/Linux), **TeXnicCenter** (un IDE para MS-Windows) o **AUCTeX** (un modo especial para EMACS).

Kile



En este taller utilizaremos **Kile** que es un entorno integrado de desarrollo (IDE) para escribir en LaTeX, para el entorno **KDE** (sobre GNU/Linux).

1: ¿Qué clase de documento queremos crear?

La primera línea de nuestro archivo `HolaMundo.tex` será:

```
\documentclass {article}
```

Esta primera línea le dice a LaTeX que clase de documento queremos crear. En este caso un artículo (`article`). Otras posibilidades son un libro (`book`), una carta (`letter`), transparencias (`slides`), un reporte (`report`), etc.

El efecto de esta orden es incluir un archivo `article.cls`, con instrucciones de como debe verse un artículo. Podemos crear nuestras propias clases de documentos.

2: Incluyendo paquetes extra

A continuación, incluimos las siguientes líneas:

```
\usepackage [spanish] {babel}  
\usepackage [T1]{fontenc}  
\usepackage [latin1]{inputenc}
```

Estas líneas tienen el propósito de incluir **paquetes**, que proporcionan facilidades adicionales. En este caso, los paquetes **fontenc** y **inputenc**, permiten que LaTeX reconozca los caracteres utilizados en castellano (acentos y ñe), y el paquete **babel** hace que LaTeX separe correctamente las sílabas en castellano.

Estas líneas se aplican si usamos la codificación latin1 (iso-8859-1 - alfabeto europeo occidental).

Si queremos usar unicode...

Si quisiéramos utilizar la codificación utf8 (unicode) deberíamos poner:

```
\usepackage{ucs}
```

```
\usepackage[utf8x]{inputenc}
```

3:Encabezado del documento

Con la línea:

```
\begin{document}
```

damos comienzo al texto del documento en sí mismo.

Las siguientes líneas crearán el encabezado del documento:

```
\title{Un documento de Prueba}
```

```
\author{Pablo Luis De Nápoli}
```

```
\maketitle
```

4: El cuerpo del documento

```
\section{Introducción}
¡Hola mundo \TeX !, para no romper
la tradición.
```

```
\section{Una fórmula}
Y ahora para que vean porque \TeX es mejor, una
matemática:
```

```
$$ \frac{\pi}{4} =
\int_0^1 \frac{1}{1+x^2} dx $$
```

```
\end{document}
```

Secciones

Los documentos de LaTeX tienen una estructura clara.

Por ejemplo la orden

```
\section{ título }
```

indica el comienzo de una nueva sección. LaTeX se ocupará de elegir el tipo de letra para el título , numerar las secciones, etc.

Más comandos para separar en secciones

Otros comandos para separar en secciones son:

- `\part`
- `\chapter`
(en las clases book y report)
- `\subsection`
- `\subsubsection`
- `\paragraph`
- `\subparagraph`
- `\appendix`

Inclusión de fórmulas matemáticas

Es muy fácil insertar fórmulas matemáticas tales como:

```
$$ \frac{\pi}{4} = \int_0^1 \frac{1}{1+x^2} dx $$
```

Esto producirá la fórmula

$$\frac{\pi}{4} = \int_0^1 \frac{1}{1+x^2} dx$$

Los signos **\$\$... \$\$** indican a LaTeX que es una fórmula, y que queremos que la centre.

\$... \$ produciría una fórmula no centrada)

Explicación de la Fórmula:

- La orden `\pi`, es simplemente la letra griega π .
- La secuencia

`\int^1_0`

producirá una integral con extremos 0 y 1.

- La orden

`\frac{1}{1+x^2}`

producirá una fracción con numerador 1 y denominador $1 + x^2$.

Compilando nuestro documento

Después de haber creado nuestro documento, debemos compilarlo usando el comando

```
pdflatex HolaMundo.tex
```

Esto creará un archivo **HolaMundo.pdf**

También es posible generar archivos en otros formatos

- dvi (“device independent”)

```
latex HolaMundo.tex
```

- postscript (.ps) [a partir del dvi]

```
dvips -o HolaMundos.ps HolaMundo.dvi
```

**Otras cosas que se
pueden hacer en LaTeX**

Etiquetas

Podemos incluir referencias a secciones o fórmulas que aparecen en otra parte de nuestro documento. Por ejemplo la secuencia

```
\begin{equation}
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\label{raices}
\end{equation}
```

creará la fórmula (numerada):

$$(1) \quad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

La orden `\label` le da un nombre a la fórmula (o sección).

Referencias Cruzadas

Más adelante podemos incluir una referencia a nuestra fórmula:

La fórmula (`\ref{raices}`) proporciona las raíces de la ecuación cuadrática $ax^2 + bx + c = 0$

Esto se verá de la siguiente forma:

La fórmula (1) proporciona las raíces de la ecuación cuadrática

$$ax^2 + bx + c = 0$$

Más sobre referencias cruzadas

Si nuestro documento contiene referencias cruzadas puede que tengamos que recompilarlo dos veces. LaTeX nos lo avisará con el mensaje:

```
LaTeX Warning: Label(s) may have changed.  
Rerun to get cross-references right.
```

Latex usa un archivo con extensión **.aux** para recordar las referencias cruzadas.

Creación de índices

- La orden

```
\tableofcontents
```

crea una tabla de contenidos (índice por secciones)

- Las órdenes (en el preámbulo)

```
\usepackage {makeidx}
```

```
\makeindex
```

crean un índice analítico. La orden

```
\printindex
```

lo imprime (en algún lugar del documento). La orden

```
\index{palabra}
```

agrega una palabra al índice.

Otras cosas que podemos hacer en LaTeX

- Notas al pié:

```
\footnote{Pon esto como nota al pié.}
```

- Listas de ítems:

```
\begin{itemize}  
\item ítem 1.  
\item ítem 2.  
\item ítem 3.  
\end{itemize}
```

Incluyendo gráficos:

Para incluir gráficos, podemos utilizar el paquete **graphics**:

```
\usepackage {graphics} %en el preámbulo
```

Podemos incluir un gráfico contenido en el archivo logo_jornadas.eps mediante la orden:

```
\begin{figure}[h]  
\includegraphics[width=0.15 \textwidth]  
{logo_jornadas.eps}  
\end{figure}
```



Tablas

```
\begin{tabular}{|c|c|}  
\hline  
$x$ & $f(x)=x^2$ \\  
\hline  
$1$ & $1$ \\  
$2$ & $4$ \\  
\hline  
\end{tabular}
```

x	$f(x) = x^2$
1	1
2	4

Aún más cosas:

- **Cambiar el *tipo de letra*.**

```
\texbf{Pon esto en negrita.}
```

```
\texsl{Pon esto en cursiva.}
```

- **Incluir bibliografías.**
- **Definir nuevos comandos.**

```
% Una abreviatura para Software Libre  
\newcommand {\SL} {software libre}
```

Para matemáticos... ¡Teoremas!

Ponemos en el preámbulo

```
\newtheorem{teorema}{Teorema}
```

y en el cuerpo del documento

```
\begin{teorema}[Pitágoras]
```

En un triángulo rectángulo con hipotenusa A y catetos B y C

```
$$ A^2 = B^2 + C^2 $$
```

```
\end{teorema}
```

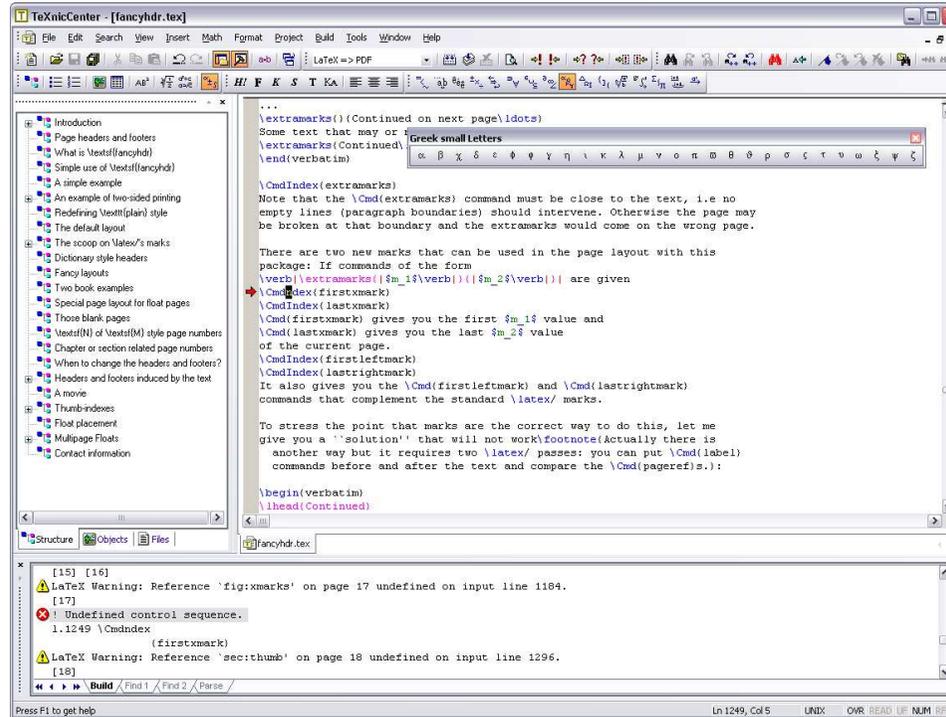
Esto se verá así:

Teorema 1 (Pitágoras) *En un triángulo rectángulo con hipotenusa A y catetos B y C se cumple que*

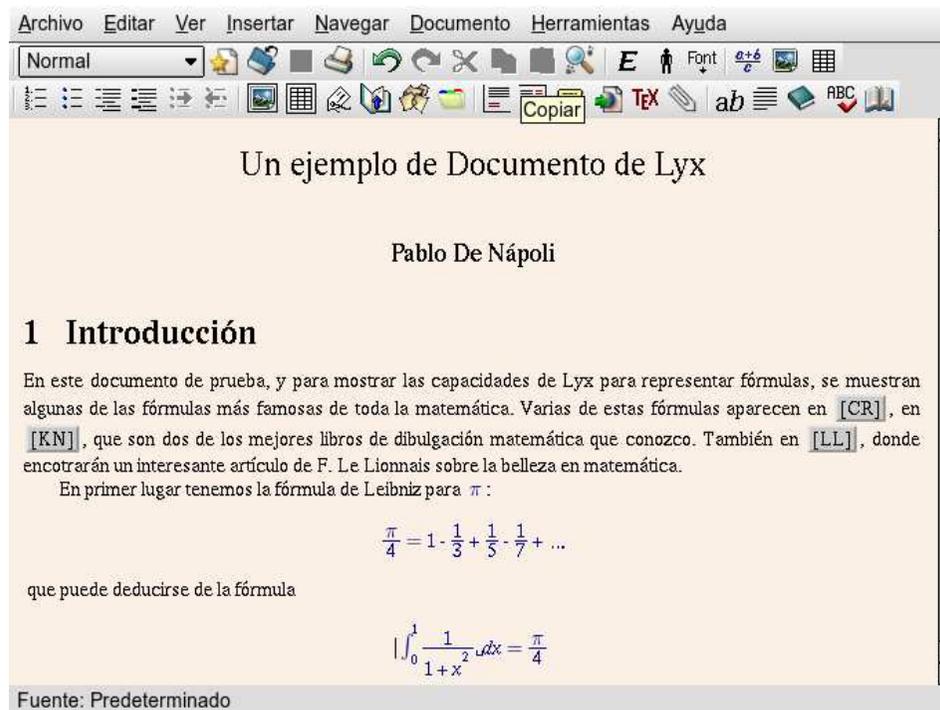
$$A^2 = B^2 + C^2$$

Herramientas Útiles

TeXnicCenter



Es un entorno integrado de desarrollo (IDE) para escribir en LaTeX, para MS-Windows.



Es un procesador de textos que permite en LaTeX, sin saber LaTeX y viendo como se verá lo que estamos escribiendo al mismo tiempo.

Verificando la ortografía

Es posible verificar la ortografía de un documento en LaTeX utilizando **ispell**

Por ejemplo, para verificar la ortografía de este documento, utilicé la orden

```
ispell -t -T latin1 -d espa~nol  
charla-latex.tex
```

El parámetro **-t** indica a ispell que queremos procesar un documento en Tex.

Los demás parámetros especifican el diccionario a utilizar, y la codificación utilizada para los caracteres en español.

Otras herramientas útiles

- **Conversores a otros formatos:** Latex2html, Latex2rtf
- **Prosper:** Un paquete para generar presentaciones en LaTeX. (Lo utilicé para crear esta presentación)
- **Sistemas de control de Versiones:** (CVS, Subversion, Mercurial, Git,...) Al ser los documentos de TeX archivos de texto, es posible usar cualquier sistema de control de versiones para trabajar con ellos.

Comunidad: Grupos de Usuarios

- TeX Users Group: Grupo de usuarios de TeX.

<http://www.tug.org/>

- CervanTeX: grupo de usuarios de TeX hispanohablantes.

<http://www.cervantex.org>

- CTAN: Comprehensive TeX Archive

<http://www.ctan.org>

Licencia de este documento

Copyright 2005-2009 Pablo Luis De Nápoli

Se garantiza el permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia Creative Commons Atribución-Compartir Obras Derivadas Igual 3.0 Unported

<http://creativecommons.org/licenses/by-sa/3.0/d>