

# Introducción a la edición de textos científicos con L<sup>A</sup>T<sub>E</sub>X

Daniel López Avellaneda ([iesmarsarena.org](http://iesmarsarena.org))

José Manuel Calahorra García ([iesmarsarena.org](http://iesmarsarena.org))

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

Manual para el curso online organizado por:

CEP Indalo

<http://www.cepindalo.es>

Octubre 2009 - Enero 2010

## Índice

1. ¿Qué es $\LaTeX$ ?	2
2. Distribuciones de $\LaTeX$	3
3. ¿Cómo funciona $\LaTeX$ ?	4
4. Formatos de salida	5

# Introducción a L<sup>A</sup>T<sub>E</sub>X

## 1 ¿Qué es L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas. L<sup>A</sup>T<sub>E</sub>X está formado por un gran conjunto de macros de T<sub>E</sub>X, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, T<sub>E</sub>X, creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con L<sup>A</sup>T<sub>E</sub>X es comparable a la de una editorial científica de primera línea. L<sup>A</sup>T<sub>E</sub>X, como no podría ser de otra forma, es software libre.

Particularmente, como profesor de Matemáticas, uso L<sup>A</sup>T<sub>E</sub>X prácticamente a diario, para confeccionar apuntes, relaciones de problemas o cualquier documento que tenga que imprimir o tenga que portar (en PDF) para que los destinatarios (normalmente alumnos/as y compañeros/as) puedan leerlo y/o imprimirlo. L<sup>A</sup>T<sub>E</sub>X me es imprescindible para expresar fórmulas matemáticas en la web (foros, blogs o webs que lo admitan).

Mis recursos, mi capacidad de expresión-comunicación, y mi tiempo disponible, se verían mermados en gran parte de no existir L<sup>A</sup>T<sub>E</sub>X.

Características como la capacidad de crear nuestros propios macros (comandos o procedimientos), como si de un lenguaje de programación se tratase, hicieron que L<sup>A</sup>T<sub>E</sub>X se extendiera rápidamente entre la comunidad científica hasta convertirse en obligatorio en importantes reuniones y congresos, e incluso sea un formato obligatorio a la hora de enviar artículos a revistas científicas (Nature, Science, Elsevier, ..).

Al ser software libre, su código abierto ayudó a que muchas personas colaboraran creando nuevas funcionalidades. Eso dio lugar a una reestandarización completa de L<sup>A</sup>T<sub>E</sub>X, mediante la versión L<sup>A</sup>T<sub>E</sub>X 2 $\epsilon$  (versión 2 y algo más ..) que incluye algunas de las extensiones aportadas por los usuarios y algunos estándares como la simbología de la AMS (American Mathematical Society). Ello fue el origen de su actual arquitectura modular: un núcleo central (el compilador), que mantiene las funcionalidades de versiones anteriores, acompañado de una serie de paquetes que aportan funcionalidades específicas.

Los paquetes más importantes y usados vienen con la distribución oficial, mientras que el resto debemos descargarlos e instalarlos sólo si los vamos a usar. Por ejemplo, si vamos a necesitar escribir símbolos especiales de Química, podríamos descargar e instalar algún paquete como "mhchem" que nos ayudará con esas fórmulas químicas (a lo largo del curso veremos cómo hacerlo).

L<sup>A</sup>T<sub>E</sub>X, como hemos dicho, es un conjunto de macros (miniprogramas o miniprocedimientos), pero un conjunto amplio: miles y miles de ellos. Piense en algunos de los más sencillos: representar una fracción, poner un texto en cursiva, poner un símbolo de integral definida, etc. Si el número de procedimientos fuese pequeño (digamos 50 a 100), bastaría con tener un programa con los 50-100 botones, uno para cada procedimiento (como suele suceder en los procesadores de textos: Writer, Word, etc.).

El elevado número de macros, hace imprescindible usar un fichero de texto plano (texto sin

formato) para escribir en código  $\LaTeX$  (por ejemplo una fracción:  $\frac{2x^2}{7}$ ) y compilarlo para obtener una salida (por ejemplo en formato PDF).

La filosofía de  $\LaTeX$  es distinta a los procesadores de textos WYSIWYG : What You See Is What You Get (lo que ves es lo que obtienes).  $\LaTeX$  se basa en comandos, del tipo  $\sqrt[3]{x^4-2x}$ , y esa desventaja, quizás la única, es el precio que hay que pagar para conseguir sus capacidades gráficas para representar ecuaciones, fórmulas complicadas, notación científica e incluso musical, estructurar fácilmente el documento (con capítulos, secciones, notas, bibliografía, índices analíticos, etc.), lo cual brinda comodidad y lo hace útil para artículos académicos y libros técnicos.

## 2 Distribuciones de $\LaTeX$

La manera de obtener  $\LaTeX$  es mediante alguna de las diferentes distribuciones que existen:

- **TeXLive**: Es la distribución que se usa en sistemas Unix/Linux. También tiene versiones para otros Sistemas Operativos.
- **MaCTeX**: Distribución usada en sistemas Mac. Básicamente es la TeXLive con instalador nativo para Mac.
- **proTeXt**: Nueva distribución de TeX para Windows basada en la clásica MiKTeX

Cualquiera de las distribuciones anteriores se puede descargar de Internet. Sin embargo es aconsejable algún software específico (Kile, LyX, etc.) que te instalan automáticamente todos los paquetes  $\LaTeX$  necesarios.

Posteriormente veremos como instalar estos programas: Kile y LyX

Algunas direcciones:

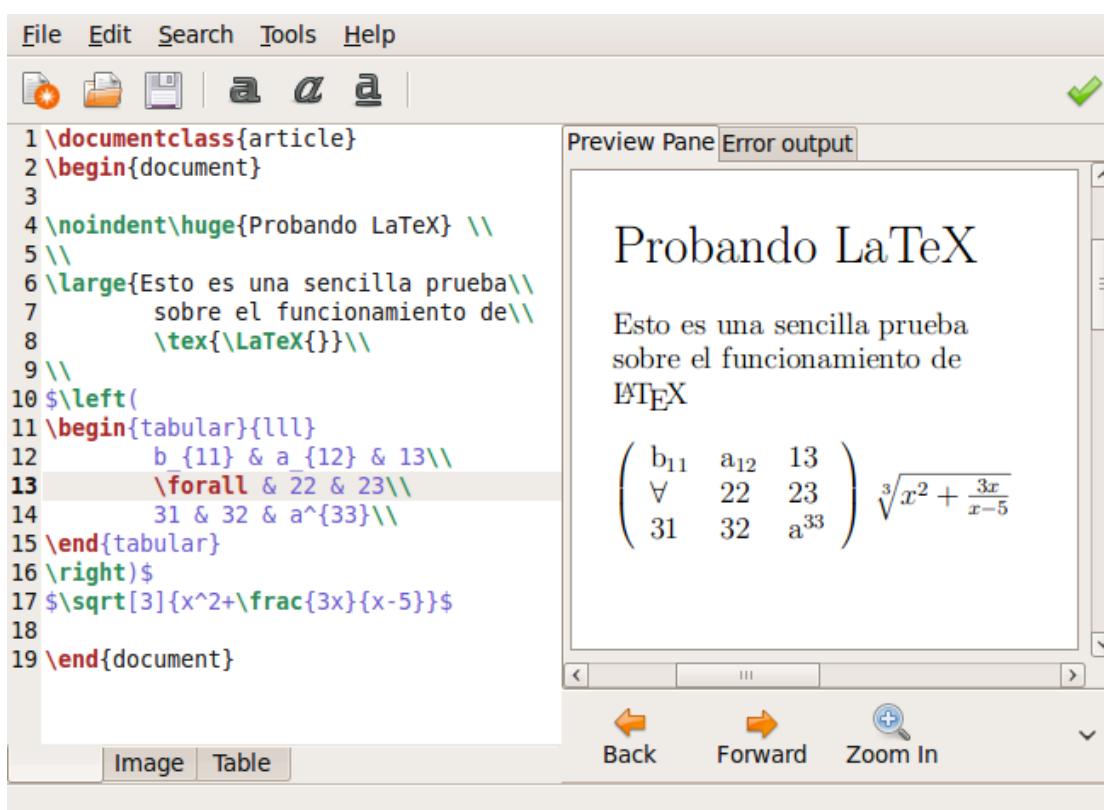
- TeXLive: <http://www.tug.org/texlive/>
- MaCTeX: <http://www.tug.org/mactex/>
- proTeXt: <http://www.tug.org/protext/>
- MiKTeX: <http://www.tug.org/miktex/>
- $\LaTeX$  project site: <http://www.latex-project.org/>
- TeX Users Group: <http://www.tug.org/>
- CervanTeX: <http://goliat.mecanica.upm.es/cervantex/> (Grupo de usuarios de TeX hispanohablantes)

### 3 ¿Cómo funciona L<sup>A</sup>T<sub>E</sub>X?

La creación de un documento requiere normalmente de dos etapas: en la primera hay que crear mediante cualquier editor de texto plano un fichero fuente que, con las órdenes y comandos adecuados, contenga el texto que queramos imprimir. La segunda consiste en procesar este fichero; el procesador de textos interpreta las órdenes escritas en él y compila el documento, dejándolo preparado para que pueda ser enviado a la salida correspondiente, ya sea la pantalla, la impresora, un fichero PDF, etc.

Por tanto, cuando descargamos e instalamos L<sup>A</sup>T<sub>E</sub>X en nuestro ordenador, tenemos sólo el compilador y un montón de macros. Para crear un documento necesitaremos un programa externo: al menos un editor de texto plano, como gEdit que usamos en LiNuX, NotePad o Bloc de Notas de Windows, etc. (no debemos confundir editor de texto con procesador de textos).

Afortunadamente existen algunos programas que nos ayudarán en la ardua labor de editar texto plano. Kile y L<sup>A</sup>X son los más importantes, y serán los que usemos durante el curso.



La imagen anterior resume los pasos en la creación de un documento:

1. En la izquierda de la imagen anterior tenemos el fichero fuente (prueba.tex) que podemos crear con cualquier editor de textos, preferiblemente un editor especializado en L<sup>A</sup>T<sub>E</sub>X, o al menos un editor que resalte el código en colores. El archivo se guarda con la extensión **.tex**

2. El segundo paso es compilar el fichero fuente. Se hace pulsando un simple botón (Kile, LyX) o tecleando un par de palabras en consola o terminal. La compilación de prueba.tex genera el fichero prueba.pdf
3. El tercer y último paso consiste en visualizar el resultado de la compilación (parte derecha de la imagen anterior), por ejemplo en un fichero PDF (prueba.pdf). En algunos programas (LyX por ejemplo), mediante un sólo clic obtenemos los pasos 2 y 3 (compila el documento y nos muestra el pdf en pantalla)

También existe software con el que se pueden hacer los tres pasos simultáneamente. La imagen anterior es una captura correspondiente a un sencillo editor que va generando el PDF al mismo tiempo que tecleamos el código fuente. Escribimos en la izquierda el código fuente y vemos al mismo tiempo el resultado (en pdf) en la parte derecha.

Es importante observar que el fichero fuente, en nuestro caso: prueba.tex tan sólo ocupa 0,3 kb. El pequeño tamaño de los ficheros fuente de  $\text{\LaTeX}$  hace que sea fácil y rápido trasportar, enviar por Internet, etc.

## 4 Formatos de salida

Podemos compilar un documento fuente (fichero.tex) para diferentes formatos de salida:

**DVI** Un archivo DVI es el archivo resultante de procesar el texto fuente a través de TEX. Su forma no depende del dispositivo que se vaya a utilizar para verlo/imprimirlo. Un archivo DVI está pensado para ser leído por un conversor que produzca una salida adecuada para una determinada impresora, o para un determinado visualizador en la pantalla del ordenador. Los archivos DVI utilizan una codificación interna de TEX, de forma que un documento fuente debe producir exactamente el mismo archivo DVI, con independencia de la implementación de TEX que se utiliza para producirlo.

**PostScript(PS)** es un lenguaje de descripción de páginas (en inglés PDL, page description language), utilizado en muchas impresoras y, de manera usual, como formato de transporte de archivos gráficos en talleres de impresión profesional (PostScript Encapsulado o EPS).

**PDF** (Portable Document Format) es, como su nombre indica, un formato de archivos transportable entre distintas plataformas, creado por Adobe y especialmente diseñado para visualizar documentos tal y como se han diseñado. Los documentos PDF se ven igual en cualquier Sistema Operativo. Es muy similar a PostScript, pero sin capacidades de programación. Actualmente es el sustituto de DVI y es el formato usado para ver la salida de  $\text{\LaTeX}$ .