

Tutorial de Latex

Juan Carlos Inostroza
jci@tux.cl

18 de octubre de 2004

Índice

I	Introducción a L^AT_EX	3
1.	Introducción	3
1.1.	La Potencia de L ^A T _E X	4
1.2.	Instalación	4
1.3.	Primeros Pasos	5
II	Formatos de Fuentes y Párrafos en L^AT_EX	6
2.	Formatos	6
2.1.	Fuentes	6
2.2.	Tipos de Letras	7
2.3.	Tamaños de Fuente	8
2.4.	Caracteres especiales	8
2.5.	Párrafos	9
2.6.	Medidas	9
2.7.	Un ejemplo	12
2.8.	Opciones	12
2.9.	Estilos de Página	13
2.10.	Paquetes Externos	13
2.10.1.	Babel	14

III Documentos	14
3. La clase Book	14
3.1. Título y Autor	15
3.2. Separaciones y Secciones	15
3.3. Tabla de Contenidos	17
3.4. Figuras	17
3.5. Referencias	19
3.6. Imágenes	21
3.7. Tabla de figuras	24
3.8. Notas al pie de página	24
3.9. Notas al margen	25
4. Clase Report	25
5. Más elementos	26
5.1. Enumeraciones	26
5.2. Listas No numeradas (o Listas)	28
5.3. Ambientes Verbatim	28
5.4. Tablas	28
5.5. Ejemplo	31
6. Modo matemático	31
6.1. LatexSym	31
6.2. Alfabeto Griego	31
6.3. Simbolos relacionales	32
6.4. Operadores Binarios	32
6.5. Superíndices y Subíndices	32
6.6. Ecuaciones	34
6.7. Matrices	34
6.8. Condicionales	35
6.9. Teoremas	35
6.10. Ejemplo	37

Índice de figuras

1. Resultado del ejemplo dado	12
2. Ejemplo de una figura	17
3. Figura alineada al tope de la página	18
4. Figura con formato “Here, if possible”	18
5. Ejemplo de “Here, definitely”	19
6. Ejemplo de una imagen inserta en una figura	23
7. Ejemplo de enumeraciones anidadas	27

Resumen

Latex es uno de los lenguajes para documentación (junto con Docbook) más versátiles conocidos. Extensible, con uso de plantillas, entre otras cosas. Este tutorial tratará de cubrir las partes más simples para generar documentos.

Parte I

Introducción a L^AT_EX

1. Introducción

L^AT_EX (<http://www.latex-project.org>) es la extensión del antiguo T_EX, escrito originalmente por Donald Knuth. A diferencia de otro tipo de editores de textos (como Openoffice o AbiWord), el documento es escrito en primer lugar en un documento de texto plano, luego interpretado y exportado en algún otro formato manejable para impresión, como DVI, PS o PDF, entre otros.

T_EX es además un lenguaje de programación (poderoso, por cierto), en las cuales se pueden agregar nuevas y mejoradas opciones. L^AT_EX (haciendo honor al nombre) es una vasta cantidad de adiciones (de ahí lo de La del nombre, de Large) al T_EX original. T_EX y METAFONT (el sistema de manejos de fuentes de T_EX) fueron presentados en la American Mathematical Society como forma de escribir documentos matemáticos usando sistemas de Typesetting. Los componentes de esa sociedad se interesaron de inmediato de la propuesta de Knuth.

Con el objetivo de hacer que las funciones primitivas de L^AT_EX fueran un poco más amigables, Leslie Lamport comenzó a trabajar en una modificación a T_EX llamada L^AT_EX, permitiendo que el autor se concentrara en el contenido y la estructura del documento, en vez de los detalles del formato. Además, L^AT_EX posee programas auxiliares externos que ayudan a la generación de índices, bibliografías, referencias cruzadas, entre otros.

Por el momento, existe LaTeX version 2e, el cual sera reemplazado en el futuro por L^AT_EX 3. Mas información acerca de la diferencia entre estos dos, se encuentra en <http://www.latex-project.org/latex3.html> .

1.1. La Potencia de L^AT_EX

L^AT_EX es un lenguaje algo complicado al principio, debido a la sintaxis (los programadores se podran sentir algo mas comodis trabajando en él). Pero posee ventajas significativas sobre otros procesadores de textos:

- L^AT_EX no es un procesador de palabras (ahorrando tiempo de proceso en otras tareas)
- L^AT_EX es un lenguaje de Type-Setting (similar a una imprenta, donde los tipos de letras se van colocando, y luego bajo un proceso físico se imprimen en papel). Es decir, los tipos y estilos de letras (entre otras cosas) de forma muy fina y depurada.
- Para la redacción de documentos científicos, L^AT_EX es muy superior a otros procesadores de textos.
- La redacción de formulas matemáticas es más simple en L^AT_EX que en otros programas.
- El formato del document puede cambiarse entre páginas con uno o mas comandos. Es posible cambiar incluso la orientación de la página.
- Como funciona usando archivos de textos, puede usarse con sistemas de control de versiones (CVS, por ejemplo) de forma más simple, para llevar un control preciso de los cambios.
- Un documento L^AT_EX puede contener mas de un documento externo (incluirlo).
- No contiene formatos propietarios
- No está limitado al hardware en el que corre

1.2. Instalación

La gran mayoría de las distribuciones poseen L^AT_EX como paquetes opcionales.

Para tener corriendo L^AT_EX, es necesario instalar los paquetes basicos de teTex (tetex-bin, tetex-base y tetex-extra). Los archivos base (tanto de teTex como de L^AT_EX) se encuentran en /usr/share/texmf (generico y distro-neutral). Además, hay paquetes opcionales como latex2html.

1.3. Primeros Pasos

\LaTeX no es similar a otros lenguajes para documentos (como SGML). A diferencia de SGML (estructurado con tags), \LaTeX está estructurado por comandos. Estos son ingresados usando "\" (backslash) y el nombre del comando. Opcionalmente, los argumentos (como el texto a afectar por el comando) se encierran con las llaves {} y []. Mas adelante, se verá la diferencia entre estas dos llaves.

El comando mas basico de todos (en \LaTeX), al igual que otros lenguajes de programación (Pascal) es usar las clausulas "begin" y "end".

Las clausulas "Begin" y "End" definen un "ambiente", es decir, lo que haya dentro de él se pueden contener varios elementos distintos y definen, al mismo tiempo, el inicio y fin de alguna propiedad global o específica del formato. La primera y mas basica forma de ambientes es "document".

```
\begin{document}
...
\end{document}
```

Este es un documento muy básico en \LaTeX , pero no pasara por el interprete, ya que es necesario definir antes el tipo de documento que se escribirá. Esto se hace con el comando

```
\documentclass{clase_de_documento}
```

Algunos de las clases de documentos existentes son "article", "book", "letter", entre otros. Estas clases de \LaTeX son archivos .cls que se encuentran en /usr/share/texmf/tex/latex/. En una instalación comun de \LaTeX , se encuentran cerca de 40 clases de documentos.

Un documento basico en \LaTeX seria:

```
% mi primer documento en latex
\documentclass{article}
\begin{document}
  Hola, mundo!
\end{document}
```

No está demás decir que los comentarios en \LaTeX se denotan con %.

Grabar este archivo como 1-holamundo.tex e interpretarlo como:

```
$ latex 1-holamundo.tex
```

Se generan 3 archivos:

- `primer.aux` : un archivo auxiliar usado generalmente para referencias
- `primer.log` : el registro de interpretación el archivo \LaTeX , contiene los errores y advertencias
- `primer.dvi` : el archivo de \LaTeX listo para ser visto por algun visor de DVI, como `xdvi` (este archivo NO es Postscript).

Es opcional convertir el archivo DVI a PostScript (o PDF). La utilidad `texi2pdf` convierte automaticamente un archivo \TeX a PDF (pero veremos mas adelante que esto es en algunos casos mas complicado de lo que parece).

Bien! Como primer documento de \LaTeX , es extremadamente simple, asi que en la siguiente seccion explicaremos algunas herramientas de formato.

Parte II

Formatos de Fuentes y Párrafos en \LaTeX

2. Formatos

Algunos comandos de formato no requieren las clausulas `\begin...\end`, ya que afectan directamente el texto escrito (TypeSetting puro). La mayoría de los comandos son del tipo `\comando{texto}`.

Ademas, los comandos pueden ser completamente anidados, como por ejemplo

```
\comando1{ejemplo1 \comando2{texto} y sigue con ejemplo 1}
```

2.1. Fuentes

Para negrillas, se usa el comando `\textbf{texto}` :

```
Este ejemplo contiene \textbf{texto en negrilla}
```

Este ejemplo contiene **texto en negrilla**

Para itálicos, se usa el comando `\textit{texto}`

```
Este ejemplo contiene \textit{texto en italico}.
```

Este ejemplo contiene *texto en italico*.

Para mayúsculas destacadas (Small Caps), se usa el comando `\textsc{texto}`. Generalmente usado para destacar algunos términos.

```
Este ejemplo tiene \textsc{mayusculas destacadas}.
```

Este ejemplo tiene MAYUSCULAS DESTACADAS

Para dar "énfasis" (o realce) a un trozo de texto, se usa `\emph{texto}`. Por ejemplo, si el texto es normal, `\emph{texto}` queda en itálicos. Si el texto está en itálicos, `\emph{texto}` queda en modo normal.

```
Este ejemplo usa \emph{enfasis}. \textit{Ahora, como esta en italicos, \emph{aquí va el enfasis} de lo que se dice}.
```

Este ejemplo usa *enfasis*. *Ahora, como esta en italicos*, aquí va el énfasis de lo que se dice.

2.2. Tipos de Letras

Estos tipos de letras, también llamados "familias", son tres definidos por defecto, en las plantillas básicas. Estas familias son Roman (definida por `\textrm{texto}`), Sans Serif (`\textsf{texto}`) y Typewriter (`\texttt{texto}`). El tipo de letra por defecto es Roman.

```
Este texto esta en Roman, \textsf{mientras que este texto esta en Sans Serif}, \texttt{y este texto como Typewriter}.
```

Este texto está en Roman, mientras que este texto está en Sans Serif,
y este texto como Typewriter.

2.3. Tamaños de Fuente

Es bastante simple dar distintos tamaños a fuentes.

La fuente de tamaño normal se define con `\normalsize{texto}`, la fuente mas pequeña `\tiny{texto}`, la siguiente mas pequeña `\small{texto}`, la fuente grande con `\large{texto}` y la fuente gigante con `\Huge` o `\huge`.

La diferencia de `\huge` con `\Huge` es que en la última, la fuente es un poco mas grande que `\huge`, de ahí a diferenciarlos con la letra mayúscula.

La ventaja es que estos tamaños pueden ser combinados en la misma frase.

```
Este ejemplo tiene letras \tiny{muy peque~nas},
\small{menos peque~na},
\normalsize{normal},
\large{mas grande},
\huge{y gigante}
\Huge{(no olvidar la mas gigante!)}
```

Este ejemplo tiene letras muy peque as, menos peque a,

normal, mas grande, **y gigante**

(no olvidar la mas gigante!)

2.4. Caracteres especiales

Una de las características especiales de \LaTeX es el soporte para caracteres especiales, pero los documentos deben tener el soporte de idioma (en el siguiente punto se indica el idioma del documento).

Para la acentuación se usa `\y'`.

Este pequeño texto de ejemplo indica la acentuación:

```
LaTeX no s'olo permite insertar caracteres dif'iciles en
otros idiomas, como el espa~nol.
```

LaTeX no sólo permite insertar caracteres difíciles en otros idiomas, como el español.

2.5. Párrafos

Para alinear párrafos, se usa `\begin{formato}...\end{formato}`.

Para alinear texto al centro, se usa `\begin{center}...\end{center}`

Para alinear texto a la derecha, se usa `\begin{flushright}...\end{flushright}`

Para alinear texto a la izquierda, se usa `\begin{flushleft}...\end{flushleft}`.
Este es el formato por defecto.

Además, los que se hayan dado cuenta, agregando líneas en el ejemplo, que las primeras líneas de texto están indentadas. Para evitar la indentación (una tabulación al principio de la línea), agregar `\noindent` al principio de la línea.

Para agregar una línea en blanco entre líneas, solo agregar una línea en blanco. Los retornos de línea por lo general son ignorados, a excepción cuando existen dos consecutivos.

2.6. Medidas

\LaTeX permite generar documentos con grados de precisión bastante efectivos. Esto es debido a que es capaz de entender medidas de distancia arbitrarias.

Entre las unidades de medidas tenemos

- pt : Points
- mm : Millimeters
- cm : Centimeters
- in : Inches
- pc : Percentage

Espacios

Se pueden agregar espacios en blanco de tamaño arbitrario, usando `\vspace{}` y `\vfill`.

`\vspace` permite hacer una separación de tamaño arbitrario.

<pre> Texto 1 \vspace{10mm} Texto 2 </pre>
--

Texto 1

Texto 2

`\vfill` permite rellenar el espacio vertical entre los textos de manera ecuánime.

Este texto est\ 'a ecu\ 'animemente separado

Este texto está ecuánimamente separado

Este texto est\ 'a ecu\ 'animemente separado

Tutorial de LaTeX
<http://www.tux.cl>

DIPLOMA

Certifico con este diploma, que

(colocar nombre aqui)

ha podido crear su **segundo** documento en LaTeX,
 usando un simple tutorial.

Resultó complicado el tutorial, o no?

Figura 1: Resultado del ejemplo dado

2.7. Un ejemplo

Ahora un ejemplo más completo, con el uso de las herramientas de formato ya vistas.

```
\documentclass{article}
\begin{document}
\begin{flushright}
Tutorial de LaTeX
\end{flushright}
\begin{center}
\Huge{\textbf{DIPLOMA}}
\end{center}
Certifico con este diploma, que
\begin{center}
\huge{\textit{(colocar nombre aqui)}}
\end{center}
\noindent ha podido crear su \textbf{segundo} documento en LaTeX,
usando un simple tutorial.
\begin{flushright}
\tiny(Result\'o complicado el tutorial, o no?)
\end{flushright}
\end{document}
```

El resultado es lo que se puede ver en la figura 1.

2.8. Opciones

Habia dicho anteriormente que los argumentos para los comandos `\comando` se delimitaban con `{}` y `[]`. Todos los comandos anteriores usan `{}`. Para definir algunos argumentos adicionales, algunas veces mas un argumento, se usan los `[]`.

Por ejemplo, para indicar que la clase del documento será "article" y que el tamaño de la fuente por defecto será 11 puntos,

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

Por defecto, L^AT_EX asume que el tamaño de página es “Letter”.

Entre algunas de las opciones poco comunes, estan

- `titlepage/notitlepage` : indica si las paginas deben contener el titulo del documento.
- `twoside/oneside` : indica si el documento se genera a dos caras o a una cara.
- `openright/openany` : indica si los capítulos empiecen en una pagina en la derecha o en la proxima disponible

Estas opciones no estan presente en todas las clases.

2.9. Estilos de Página

Existen tres estilos basicos:

- `plain` : imprime los numeros de pagina al centro del pie de pagina
- `headings` : imprime una cabecera con el nombre del capitulo y el pie de pagina esta vacio
- `empty` : no imprime cabeceras ni numeros de pagina

Para modificar el estilo de la pagina del documento, se usa

```
\pagestyle{estilo}
```

Y para cambiar el estilo de pagina en la pagina actual

```
\thispagestyle{estilo}
```

2.10. Paquetes Externos

Hacíamos referencia en el capítulo anterior que se necesitaba el paquete "babel" y el soporte para español. La inserción de paquetes externos hace que nuestro L^AT_EX posea funcionalidades adicionales.

Para insertar un paquete se usa la clausula `\usepackage{nombre_paquete}`. Esta clausula siempre se inserta antes del inicio del documento (`\begin{document}`).

Para modificar las opciones por defecto en las cuales viene el paquete, se pasan parametros (al igual que en `\documentclass`) con los corchetes `[]`.

Por ejemplo, para hacer referencia al idioma español en el paquete Babel,

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

2.10.1. Babel

Hay que dedicarle una seccion al documento a babel :-)

Debido al lenguaje "estándar" (no el codigo, sino al lenguaje hablado) del inglés, al hacer documentos básicos, algunas cadenas quedan en ingles (como las secciones, no los nombres de ellas). Por ejemplo, insertando capítulos en un libro, quedan como "Chapter <numero>".

Aqui es donde entra Babel, permitiendo traducir las cadenas mas básicas e internas de L^AT_EX a otro idioma. Por ejemplo, "Chapter" quedaría como "Capítulo" en español, "Rozdzia" en Polaco y "Capitro" en Esperanto. :)

Babel no es un paquete de traducción desde y hacia el inglés. Lo que hace babel es traducir las cadenas internas de L^AT_EX.

Parte III

Documentos

Entre algunas de las clases existentes de documentos, esta "book" (libros), "letter" (para escribir cartas), "report" (para reportes) y "slides" (para hacer presentaciones).

Cada una de estas clases posee un formato diferente, por ejemplo las cabeceras y las secciones, el tamaño de hoja por defecto, entre otros. Pero entraremos en detalles mas tarde, ya que estos son vistos una vez generado el documento.

Los archivos de clases de documentos tienen extension "cls". Generalmente están ubicados en `/usr/share/texmf/tex/latex`

3. La clase Book

Para este ejemplo completo, usaremos la clase Book y haremos un pequeño libro, con índice de contenidos y algunos capítulos.

Comenzamos entonces con la clase del documento y las opciones respectivas.

```
\documentclass[11pt]{book}
```

Agregamos el uso de Babel

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

Luego, iniciamos el documento con

```
\begin{document}
```

Podemos insertar elementos que son dependientes de la clase de documento, como Autor, Título y las respectivas separaciones.

3.1. Título y Autor

Para definir el título del libro, se define con `\title{Titulo de la obra}`.

Para el autor, `\author{Autor del libro}`.

Una vez escritos estos dos elementos, se debe agregar a continuación de `\author{}` el comando `\maketitle`.

3.2. Separaciones y Secciones

Anteriormente, no habíamos definido el cómo separar los documentos en secciones (ya podríamos escribir eternamente “Hola Mundo” :-)) y así tener nuestro documento mas ordenado. El uso de secciones es importante al momento de escribir un reporte o un libro, ya que no es la idea separar y escribir las secciones a mano.

La plantilla "book" posee las siguientes separaciones

- Chapter
- Section
- Subsection
- Subsubsection

Dependiendo de cada elemento, se abra una pagina nueva con el nombre (Chapter) o se incluyan secciones numericamente (Section, Subsection, Subsubsection). Esta numeración es automática.

Estos delimitadores no son de la forma `\begin{comando}...\end{comando}`, (de hecho no son delimitadores, sino definiciones de elementos) y son excluyentes entre si. Es decir, al definir un capítulo, éste no terminará hasta que se defina un nuevo capítulo. Y al insertar una nueva sección, ésta heredará la numeración base del capítulo.

Estos delimitadores son de la siguiente forma:

- `\chapter{Nombre del Capitulo}`

- `\section{Nombre de la Seccion}`
- `\subsection{Nombre de la Subseccion}`
- `\subsubsection{Nombre de la Subsubseccion}`

Además, existe la manera para “sobrepasar” la numeración (ya que los capítulos y las secciones se autonumeran). Es decir, se obvia la numeración (por ejemplo, para agregar secciones especiales, como los agradecimientos). Esto se logra con

- `\chapter*{Nombre del Capitulo}`
- `\section*{Nombre de la Seccion}`
- `\subsection*{Nombre de la Subseccion}`
- `\subsubsection*{Nombre de la Subsubseccion}`

Ahora, teniendo todos los materiales a mano, la representación jerárquica de como podría componerse nuestro libro sería la siguiente:

```

\documentclass{book}
\begin{document}
\title{Mi Primer libro de LaTeX}
\author{J. C. Inostroza}
\maketitle
\chapter{Capitulo}
  Texto...
  \section{Seccion}
    Texto...
    \subsection{Subseccion}
      Texto...
      \subsubsection{Subsubseccion}
\chapter{Capitulo}
  Texto...
  .....
\end{document}

```

El ejemplo siguiente es demasiado largo de incluir dentro de este documento, pero puede encontrarse en los archivos como `1-libro.tex`.

Como corresponde, conviene interpretar y convertir el archivo \LaTeX a DVI con

```
$ latex 1-libro.tex
```

Esta es una figura. En este momento no tiene ningún formato en particular. Podríamos considerar que esta figura solamente tiene texto, pero podemos además insertar alguna figurilla en especial que queramos utilizar. Pues no desesperéis, ya que luego insertaremos una imagen.

Figura 2: Ejemplo de una figura

3.3. Tabla de Contenidos

En la clase anterior, indicamos que los capítulos, secciones y subsecciones se autonumeraban. Pero en el ejemplo anterior, no indicamos el índice del libro.

Para eso, usamos una tabla de contenidos. Insertamos entonces

```
\tableofcontents{}
```

No es necesario llenar con argumentos en {}.

Es opcional además insertar una tabla de contenidos en un documento en \LaTeX .

Además, la flexibilidad es tal que se puede insertar la tabla de contenidos al principio o al final del documento.

3.4. Figuras

Algunas veces necesitamos figuras para indicar una fórmula, una imagen o cualquier otro elemento externo. Son bastante usados en reportes científicos, libros de computación, etc.

Una figura en \LaTeX es un elemento flotante (comúnmente llamado Figure Float) independiente del formato del párrafo en el que se encuentra. Permite insertar figuras, textos, etc.

Para crear una figura en cualquier parte del documento, usamos:

```
\begin{figure}  
...  
\end{figure}
```

Lo que va dentro de la figura puede ser texto, imágenes, etc.

El formato de la figura es automático, no requiriendo formato adicional. Este formato (al igual que otros en \LaTeX) puede pasarse por alto.

Dependiendo de la página y del texto, la figura puede quedar en la página siguiente o en la anterior en la que se inserta. Esto permite una mayor libertad para la composición

```
\begin{figure}[t]
```

Figura 3: Figura alineada al tope de la página

de los documentos, pero de cierta forma, algunas veces resulta molesto. Entonces se puede indicar a \LaTeX que este formato se pase por alto.

Ahora, si insertamos una figura, esta solo contendrá una leyenda y un número (Figure #). Necesitamos indicar además una leyenda de lo que queremos mostrar.

Esto se llama "caption". Un ejemplo puede ser:

```
\begin{figure}
...
\caption{Descripcion de la figura}
\end{figure}
```

Nuevamente, al igual que las secciones y los capítulos, las figuras se numeran automáticamente.

Puede realizarse algunos formatos a la figura, como la posición. Estos formatos se especifican entre [] después de definir la figura.

Por ejemplo, para formatear la figura en el tope (Top, o simplemente "t") de la página:

```
\begin{figure}[t]
```

Existe otro llamado "Here, if possible". Indica que la figura iría a continuación del texto, si es posible.

```
\begin{figure}[h]
```

```
\begin{figure}[h]
```

Figura 4: Figura con formato "Here, if possible"

Y otro, más extremo, "Here, definitely". Indica que la figura obligatoriamente debe insertarse en la posición indicada.

```
\begin{figure}[H]
```

Este texto está antes de la figura 5.

Este es el ejemplo de “Here definitely”

```
\begin{figure}[H]
```

Nota para el lector : observar los textos antes y después de esta figura.

Figura 5: Ejmplo de “Here, definitely”

Y este texto está después de la figura5.

Incluso, se pueden agrupar todas las figuras en paginas separadas del texto (Page of Floats) (que queda a ejercicio del lector) :

```
\begin{figure}[p]
```

3.5. Referencias

Una de las cosas frustrantes de los procesadores de texto es la limitación dentro de la inserción de figuras y capitulos (ya que la autonumeración funciona bastante deficiente, no hablar de las figuras).

Otra cosa frustrante es la generación de referencias cruzadas (ya que no es trivial).

Una referencia cruzada es un elemento que se define en cualquier parte del documento y que puede ser usada en otra parte del documento. Estas referencias funcionan de forma inteligentísima: es posible usarlas para Capítulos, Figuras, Secciones, etc.

Primero, se inserta una etiqueta con un nombre arbitrario. Puede insertarse en casi cualquier parte (excepto fuera del documento, por supuesto).

Por ejemplo, para una etiqueta LABEL1 (arbitraria):

```
\label{LABEL1}
```

Para llamar a las referencia, se utiliza solamente la etiqueta recién definida.

```
\ref{LABEL1}
```

Para indicar el actual número dentro del documento (Capítulo, Sección, Página, etc) de la referencia, se utiliza para indicar el numero de pagina de la etiqueta

```
\pageref{LABEL1}
```

y para indicar la referencia "virtual" de la etiqueta (por ejemplo, "la pagina siguiente" o "la pagina anterior"), lo que se escribira en palabras,

```
\vpageref{LABEL1}
```

Por ejemplo, insertando una etiqueta a un capitulo:

```
\chapter{\label{TEST1}La televisi'on es cada vez mas fome}
```

Y para referenciarla, por ejemplo

```
Podemos encontrar en \ref{TEST1} la referencia  
que buscamos.
```

Podemos encontrar en [3.5¹](#) la referencia que buscamos.

O también usar `\pageref{}`

```
Y puede encontrarse ademas en la pagina \pageref{TEST1}.
```

Y puede encontrarse ademas en la pagina [19](#).

Podemos usarla incluso en las figuras,

```
\begin{figure}  
\caption{\label{ET1}Leyenda de la figura, como se pudo  
ver en la figura \ref{LABEL2}}  
\end{figure}
```

Hay si que indicar una salvedad. La definicion de las etiquetas y las referencias se hacen en ese orden. Es decir, primero las etiquetas y luego las referencias. De esta forma, \LaTeX interpreta correctamente el archivo.

En caso de se haga al azar (primero la referencia y despues la etiqueta, lo cual es lo más típico), \LaTeX arrojará mas de una advertencia, como esta:

```
LaTeX Warning: Reference 'SI1' on page 4 undefined  
on input line 65.
```

¹Queda a ejercicio del lector verificar la razon del numero [3.5](#)

Para arreglar esto, debe ejecutarse nuevamente \LaTeX , y éste usará el archivo `.aux` que genera en la primera pasada como ayuda para las referencias. En el caso de más advertencias (que no sean de referencias y etiquetas mal escritas), revisar el archivo de \LaTeX (puede ser otra cosa).

Es decir, si aparece esta advertencia,

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

Se debe correr \LaTeX nuevamente.

3.6. Imágenes

Se puede insertar diversos formatos de imágenes (con mayores prestaciones en las figuras) en \LaTeX , entre los que soporta nativamente EPS y PS.

Nota : anteriormente había indicado que se puede convertir un documento desde \LaTeX a PDF usando `latex2pdf`. Ahora, \LaTeX nativamente soporta archivos `.eps` y `.ps` (Encapsulated PostScript y PostScript). `latex2pdf` soporta archivos de imágenes comunes, como `.jpg`, `.png`, etc, los que nativamente \LaTeX no soporta (en versiones anteriores). Esto es debido a que \LaTeX es incapaz de determinar el tamaño en que está encuadrada la imagen, arrojando errores. Para resolver esto, usar el programa "convert", que es parte de ImageMagick, para convertir las imágenes desde el formato original (JPG, PNG, etc) a EPS o PS.

Para insertar imágenes, primero se debe utilizar el paquete `graphicx`

```
\usepackage{graphicx}
```

Luego, para insertar cualquier imagen `.eps`:

```
\includegraphics{archivo.eps}
```

Y eso es todo.



Este es una imagen originalmente en JPG y convertida usando Convert.

Para insertar en figuras,

```
\begin{figure}
\begin{center}
\includegraphics{1.eps}
\end{center}
\caption{Ejemplo de un imagen inserta en una figura}
\end{figure}
```

La figura 6 es un ejemplo de la forma de insertar una imagen dentro de una figura.

Si las imagenes estan en formato en .jpg,

```
\includegraphics{test2.jpg}
```

se puede directamente convertir a PDF usando



Figura 6: Ejemplo de una imagen inserta en una figura

```
$ texi2pdf archivo.tex
```

O bien, convertir la imagen a .eps

```
$ convert test2.jpg test2.eps
```

e incluirla en el documento:

```
\includegraphics{test2.eps}
```

Algunas de las cosas que pueden hacerse con las figuras, dependiendo de las opciones (en [] nuevamente) son:



- Rotarla en 45 grados:

```
\includegraphics[angle=45]{1.eps}
```

- Cambiarle el tamaño:



```
\includegraphics[scale=.25]{1.eps}
```

- O ambas



```
\includegraphics[scale=1.5,angle=270]{1.eps}
```

3.7. Tabla de figuras

Antes de definir este punto, habia que definir las figuras. Una vez terminada la explicacion, la tabla de figuras se inserta de este modo:

```
\listoffigures
```

Esto, por conveniencia, se inserta despues de la Tabla de Contenidos.

3.8. Notas al pie de página

Suponiendo que la explicacion de un concepto o una idea no pueda caber dentro de la idea principal del parrafo, se usan los pies de página. Es visualmente difícil de entender la sintaxis, ya que se insertan donde el concepto quiera recalcar².

Un ejemplo puede ser

```
Al tipear un documento, es imposible seguir el
ritmo\footnote{Se entiende como flujo, no como movimiento}
cuando hay distracciones visuales.
```

Al tipear un documento, es imposible seguir el ritmo³ cuando hay distracciones visuales.

²Como dije, es visualmente complejo. A veces, tiende a generar confusión dentro del fuente del documento.

³Se entiende como flujo, no como movimiento

Por supuesto, la numeracion es automática.

Esta definicion a veces es incomoda, sobre todo ante pies de pagina muy extensos. La idea es usar brevedad en el concepto (este es un tip gratis :-)

```
Por ejemplo, al usar el comando apt-get\footnote{Para mas
informacion, \texttt{man apt-get} o bien consultar en pagina
\pageref{aptget}} es necesario conocer un poco del sistema
de paquetes de Debian.
```

Por ejemplo, al usar el comando apt-get⁴ es necesario conocer un poco del sistema de paquetes de Debian.

3.9. Notas al margen

Al igual que los pie de página, se pueden usar para algun concepto asociado. Se vera formateado al lado derecho de la pagina.

```
Al navegar en Internet \marginpar{Mozilla y Firefox son
excelentes ejemplos},ideal es usar algun navegador con mayor
cantidad de prestaciones.
```

Al navegar en Internet ,
ideal es usar algun navegador con mayor cantidad de prestaciones.

Mozilla y Firefox son exce-
lentes ejemplos

4. Clase Report

Por fin! Era necesario hablar de algunas herramientas adicionales antes de lanzarse a escribir un reporte científico.

Sera un poco desilusionante saber que el tipo de documento es similar al de Book, con las mismas divisiones de secciones.

Pero existen diferencias importantes entre estos dos tipos:

- Book posee automaticamente una cabecera, con el nombre del Capitulo actual, la que Report carece
- La numeracion de paginas es Book es al lado del nombre del capitulo (a la izquierda o la derecha). Cambia de acuerdo a la paginacion. Report, en cambio, numera la numeracion en la parte inferior, centrada.

⁴Para mas información, `man apt-get` o bien consultar en página 19

- Book siempre agrega una página adicional entre capítulos.

Como se indico al principio, para indicar que el documento usa la clase report,

```
\documentclass{report}
```

El archivo de ejemplo se llama `2-report.tex`.

5. Más elementos

Ahora vamos a algo un poco mas abstracto, como Listas, Enumeraciones, Tablas, entre otros.

5.1. Enumeraciones

Generalmente las enumeraciones se denotan como

1. Item
2. Otro item
3. Otro item mas
4.

Para \LaTeX , se definen como

```
\begin{enumerate}  
\item Item  
\item Otro item  
\item Otro item mas  
\item ...  
\end{enumerate}
```

Mejor aun, estas pueden anidarse

1. Ingresar el nombre de usuario
2. Ingresar la contraseña.
3. Si no funciona ingresar al sistema
 - a) Verificar el nombre de usuario
 - 1) Quizás el nombre de usuario es incorrecto
 - 2) Quizás fue mal escrito
 - 3) Quizás el usuario no existe
 - b) Verificar que Caps Lock no este activado
 - c) Verificar que la contraseña sea la correcta

Figura 7: Ejemplo de enumeraciones anidadas

```

\begin{enumerate}
  \item Ingresar el nombre de usuario
  \item Ingresar la contrase\~na.
  \item Si no funciona ingresar al sistema
    \begin{enumerate}
      \item Verificar el nombre de usuario
        \begin{enumerate}
          \item Quiz\'as el nombre de
            usuario es incorrecto
          \item Quiz\'as fue mal escrito
          \item Quiz\'as el usuario no
            existe
        \end{enumerate}
      \item Verificar que Caps Lock no este
        activado
      \item Verificar que la contrase\~na
        sea la correcta
    \end{enumerate}
\end{enumerate}

```

La indentación es a propósito, para mostrar mejor los elementos de la enumeración.

Fijarse bien en estos detalles:

- El primer nivel será numérico (1..2..3..)
- El segundo nivel será con letras (a..b..c...)
- El tercer nivel será con numeros romanos (i...ii...iii...)

Ojalá no usar más de cuatro niveles (\LaTeX reclamará con un "Too Deeply Nested")

5.2. Listas No numeradas (o Listas)

Estas se definen como

```
\begin{itemize}
\item Primer Item
\item Segundo Item
\end{itemize}
```

Queda a ejercicio del lector hacer una prueba anidando niveles entre Listas y Enumeraciones.

5.3. Ambientes Verbatim

Para imprimir, por ejemplo, código fuente, es complicado usando

```
\texttt{#include <stdio.h>
\\
void main (void);}
```

Para eso se usan Verbatims, que son interpretaciones textuales de lo escrito dentro de este ambiente.

Instrucciones de \LaTeX , ciertos formatos y tipos de letras son simplemente ignorados y escritos tal cual.

```
\begin{verbatim}
#include <stdio.h>
int main(void)
{
return 0;
}
\end{verbatim}
```

```
#include <stdio.h>
int main(void)
{
return 0;
}
```

5.4. Tablas

Insertar tablas en \LaTeX es mucho más complicado de lo que aparenta ser. Incluso llega a ser algo incómodo, ya que muchos errores de formato siempre se encuentran aquí.

Se pueden crear las tablas mas simples de todas (1 fila, 1 columna), hasta llegar a complejas, con combinacion de filas o columnas.

Las tablas se definen como un ambiente de tipo tabular

```
\begin{tabular}{opciones}
...
\end{tabular}
```

Las opciones se denotan como sigue:

- | (pipe) indica una linea vertical
- l indica formato del texto a la izquierda
- r indica formato del texto a la derecha
- c indica formato del texto al centro
- p{tamaño} indica el tamaño de la columna

Lo que va dentro de la tabla es aun mas complejo.

- \hline escribe una linea horizontal
- & indica salto a la siguiente columna
- \\ separa las filas (como saltos de linea)

Un ejemplo para apreciar la complejidad :-)

```
\begin{tabular}{|c|c|}
\hline
Nombre & Correo Electronico \\
\hline
jci & jci@nospam.cl \\
fcatrin & fcatrin@nospam.com \\
csv & csv@nospam.org \\
\hline
\end{tabular}
```

Nombre	Correo Electronico
jci	jci@nospam.cl
fcatrin	fcatrin@nospam.com
csv	csv@nospam.org

Nice, uh? :)

Ahora, un poco más complejo.

Por ejemplo, si se desea crear dos columnas, y ademas la segunda columna tiene mas de una linea. Ya que \\ separa las filas, seria imposible lograrlo.

La opcion es usar dos comandos en conjunto

- `\raggedright` : indica que el `\` no inicia una nueva fila de la tabla
- `\tabularnewline` : indica el fin de la fila

Si no hay más líneas que dividir, se omite `\raggedright` (generalmente la última fila)

Un ejemplo

```
\begin{tabular}{lp{6cm}}
Instrumento & tipo \tabularnewline
Guitarra & \raggedright Instrumento de Cuerdas \\
Usa 6 Cuerdas, Nylon o Metalicas \\
Acustica, Electroacustica o Electrica
\tabularnewline
Piano & \raggedright Instrumento de digitacion \\
Electronico o de Percusion/Cuerda \\
Notas Blancas/Negras
\tabularnewline
Violin & \raggedright Instrumento de Cuerda \\
Asociado a la musica clasica \\
Tama\~no reducido
\tabularnewline
Laud & Ni idea. No lo conozco. :) \tabularnewline
\end{tabular}
```

Instrumento	tipo
Guitarra	Instrumento de Cuerdas Usa 6 Cuerdas, Nylon o Metalicas Acustica, Electroacustica o Electrica
Piano	Instrumento de digitacion Electronico o de Percusion/Cuerda Notas Blancas/Negras
Violin	Instrumento de Cuerda Asociado a la musica clasica Tamaño reducido
Laud	Ni idea. No lo conozco. :)

Ahora usaremos multicolumna, que permitirá unir varias columnas en una (Column Span)

```
\begin{tabular}{|l|r|}
\hline
\multicolumn{2}{|c|}{Instrumentos de cuerda}\hline
Material & Cuerdas \\
Guitarra Acustica & Nylon\\
Guitarra Electrica & Metalicas \\
Violin & Tripa de Gato (!) \\
\hline
\end{tabular}
```

Instrumentos de cuerda	
Material	Cuerdas
Guitarra Acustica	Nylon
Guitarra Electrica	Metalicas
Violin	Tripa de Gato (!)

5.5. Ejemplo

Un ejemplo de lo visto hasta el momento está en el archivo 5-complex.tex

6. Modo matemático

La fama de \LaTeX es ir de la mano con la escritura de formulas matemáticas y complejas formulas, lo cual no deja de ser cierto.

\LaTeX posee un modo especial, llamado “modo matemático”, el cual se inicia y se termina con el caracter “\$”. Tambien puede ser invocado usando el ambiente “equation”.

El formato del modo matemático es hacia la izquierda. Es posible forzar el formato usando “\$\$”, más simple.

Por ejemplo, nótese la diferencia de formato entre

$$a=a+1$$

y

$$a = a + 1$$

la cual es significativa.

6.1. LatexSym

Para algunos de los simbolos a continuación es necesario utilizar el paquete Latexsym.

6.2. Alfabeto Griego

Es relativamente simple agregar una letra del alfabeto griego. Solamente escribir \backslash y el nombre en ingles de la letra.

```

 $\alpha, \backslash beta, \backslash gamma, \backslash delta, \backslash epsilon, \backslash zeta, \backslash eta, \backslash
\theta, \backslash iota, \backslash kappa, \backslash lambda, \backslash mu, \backslash nu, \backslash xi, \backslash pi, \backslash rho, \backslash
\sigma, \backslash tau, \backslash upsilon, \backslash phi, \backslash chi, \backslash psi, \backslash omega$$ 
```

$\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta,$
 $\theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \pi, \rho,$
 $\sigma, \tau, \upsilon, \phi, \chi, \psi, \omega$

6.3. Símbolos relacionales

```
$\leq, \prec, \preceq, \ll, \subset, \subseteq, \sqsubset, \llcorner, \sqsubseteq, \in, \vdash, \geq, \succ, \succeq, \gg, \supset, \supseteq, \sqsupset, \sqsupseteq, \ni, \dashv, \equiv, \sim, \simeq, \asymp, \approx, \cong, \neq, \doteq, \notin, \models, \perp, \mid, \parallel, \bowtie, \Join, \smile, \frown, \propto$
```

$\leq, \prec, \preceq, \ll, \subset, \subseteq, \sqsubset,$
 $\llcorner, \sqsubseteq, \in, \vdash, \geq, \succ, \succeq, \gg, \supset,$
 $\supseteq, \sqsupset, \sqsupseteq, \ni, \dashv, \equiv, \sim,$
 $\simeq, \asymp, \approx, \cong, \neq, \doteq, \notin, \models,$
 $\perp, \mid, \parallel, \bowtie, \Join, \smile, \frown, \propto$

6.4. Operadores Binarios

Algunos operadores binarios útiles

```
$\pm, \mp, \times, \div, \ast, \star, \circ, \bullet, \cdot, \cap, \cup, \uplus, \sqcap, \sqcup, \vee, \wedge, \setminus, \wr, \diamond, \bigtriangleup, \bigtriangledown, \triangleleft, \triangleright, \trianglelefteq, \trianglerighteq, \lhd, \rhd, \unlhd, \unrhd, \oplus, \ominus, \otimes, \oslash, \odot, \bigcirc, \dagger, \ddagger, \amalg$
```

$\pm, \mp, \times, \div, \ast, \star, \circ, \bullet,$
 $\cdot, \cap, \cup, \uplus, \sqcap, \sqcup, \vee, \wedge,$
 $\setminus, \wr, \diamond, \triangle, \nabla,$
 $\triangleleft, \triangleright, \trianglelefteq, \trianglerighteq,$
 $\oplus, \ominus, \otimes, \oslash, \odot, \bigcirc, \dagger,$
 \ddagger, \blacksquare

6.5. Superíndices y Subíndices

El modo matemático permite el uso de otros operadores, como “^” para superíndices y “_” para subíndices.

Por ejemplo:

```
$a^n = a^{n-1} + a^{n-2} + a^{n-3} + \dots$
```

$$a^n = a^{n-1} + a^{n-2} + a^{n-3} + \dots$$

```


$$a_n = a_{n-1} + a_{n-2} + a_{n-3} + \dots$$


```

$$a_n = a_{n-1} + a_{n-2} + a_{n-3} + \dots$$

Como detalle, las llaves de corchete {} funcionan como agrupadores de elementos.

Algunos ejemplos (ya que sería excesivamente largo explicar cada uno de los elementos de este modo):

```


$$\sum_0^n 1 = n$$


```

$$\sum_0^n 1 = n$$

```


$$z - \gamma_{10}x - \sum_{m+n \geq 2} \gamma_{mn} x^m z^n$$


```

$$z - \gamma_{10}x - \sum_{m+n \geq 2} \gamma_{mn} x^m z^n$$

```


$$\frac{a}{b} = n + \frac{1}{n + \frac{a}{b+8}}$$


```

$$\frac{a}{b} = n + \frac{1}{n + \frac{a}{b+8}}$$

```


$$\sum_0^n \int_{-\infty}^{\infty} e^{-x^2} + \int_{-\infty}^{\infty} e^{x^2} + \int_{-\infty}^{\infty} e^{x^{-2}} = 0$$


```

$$\sum_0^n \int_{-\infty}^{\infty} e^{-x^2} + \int_{-\infty}^{\infty} e^{x^2} + \int_{-\infty}^{\infty} e^{x^{-2}} = 0$$

```


$$\sqrt[n]{\log \Bigl[ \int_0^{\infty} \exp(-\beta \varepsilon) \, d\varepsilon \Bigl] \approx \int_0^{\infty} f(x) \, dx + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} \, dx \, dy}$$


```

$$\sqrt[n]{\log \left[\int_0^\infty \exp(-\beta \epsilon) d\epsilon \right]} \approx \int_0^\infty f(x) dx + \int_{-\infty}^\infty \int_{-\infty}^\infty e^{-(x^2+y^2)} dx dy$$

```

$$
{123 \choose 456} + {x^n - y^n \choose 1 + u^{2n}} =
{x^{n+1} - y^{n+1} \choose 1 - u^{4n-1}}
$$

```

$$\binom{123}{456} + \binom{x^n - y^n}{1 + u^{2n}} = \binom{x^{n+1} - y^{n+1}}{1 - u^{4n-1}}$$

6.6. Ecuaciones

Para definir ecuaciones, no es necesario usar “\$”, sino definirla dentro de un ambiente *equation*.

El formato al centro y la numeracion es automatica. Cabe destacar que ademas puede agregarse etiquetas.

```

\begin{equation}
a_n = a_{n-1} + a_{n-2} + \dots
\end{equation}

```

$$a_n = a_{n-1} + a_{n-2} + \dots \quad (1)$$

6.7. Matrices

Para definir matrices se usa `\matrix{}`. Los elementos se separan con `&`, al igual que los tabulares. Para crear una nueva fila se usa `\cr`.

```

\matrix{1 & 2 & 3\cr 2 & 3 & 4\cr 3 & 4 & 5}

```

```

1 2 3
2 3 4
3 4 5

```

Para este ejemplo, además hemos agregado paréntesis que abarcan el alto completo de la matriz. Para esto se usan `\left` y `\right` y el tipo de cerradura.

Por ejemplo, con `\left(y \right)`.

```
$\left(\matrix{1 & 2 & 3 & 4\cr 2 & 3 & 4 & 5}\right)$
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix}$$

Ahora con `\left[y \right]`

```
$\left[\matrix{1 & 2 & 3 & 4\cr 2 & 3 & 4 & 5}\right]$
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{bmatrix}$$

6.8. Condicionalidades

Para crear condicionalidades para resolución de problemas, por ejemplo

```
$f(x) = \cases{x+2, & para x<2 \cr x+4, & para x>=2}$
```

$$f(x) = \begin{cases} x+2, & \text{para } x < 2 \\ x+4, & \text{para } x \geq 2 \end{cases}$$

6.9. Teoremas

Antes de entrar de lleno a definir los teoremas, lemas, corolarios y ejemplos (todo un set de herramientas matematicos), es necesario indicar algunas salvedades.

- Primero, cuando se utilizan paquetes externos (dicho anteriormente) se insertan antes del `\begin{document}`. Esta zona se le llama “preambulo” (y de aqui en adelante, se especificara algunos comandos en el preambulo del documento, sin indicar donde ubicarla).
- Segundo, se puede agregar comandos adicionales o definir algunas propiedades que pueden ser heredadas por los ambientes posteriores. Por ejemplo, definir que los ambientes Verbatim tendran una caja alrededor (como en este documento), sin tener que pasar las propiedades a cada uno de los Verbatims.

Después de esta aclaración, podemos continuar con los teoremas.

Para utilizar teoremas, es necesario insertar en el preambulo del documento

```
\newtheorem{thm}{Teorema}
```

y para crear un teorema,

```
\begin{thm}
Aquí irá el postulado del teorema.
\end{thm}
```

Teorema 1 *Aquí irá el postulado del teorema.*

Igualmente para el corolario, se define en el preambulo

```
\newtheorem{col}{Corolario}
```

Y se crea un Corolario

```
\begin{col}[Autor del corolario o detalle corto del corolario]
Este es el texto del corolario
\end{col}
```

```
\begin{col}[Inostroza, 2004]
Este es el texto del corolario
\end{col}
```

Corolario 1 (Inostroza, 2004) *Este es el texto del corolario*

Analogamente para el Lema, se define en el preambulo, con la diferencia que debe estar además definido el Teorema previamente

```
\newtheorem{lem}{Lema}[thm]
```

Y para crear el Lema

```
\begin{lem}
Texto del lema
\end{lem}
```

Lema 1.1 *Texto del lema*

Y los ejemplos del Lema,

```
\newtheorem{exa}{Ejemplo}[lem]
```

```
\begin{exa}
Este es el texto del ejemplo
\end{exa}
```

Ejemplo 1.1.1 *Este es el texto del ejemplo*

6.10. Ejemplo

Un ejemplo de lo visto hasta el momento esta en el archivo 6-math.tex