

Tutorial de LaTeX

Introducción básica

Gastón Simone

Tutorial de LaTeX: Introducción básica
por Gastón Simone

Este documento es libre, Ud. puede redistribuirlo y/o modificarlo bajo los términos de la FDL (GNU Free Documentation Licence) publicada por la Fundación de Software Libre; ya sea la versión 1 de la licencia o (a su consideración) cualquier versión posterior.

Este documento es distribuído con el deseo de que sea de utilidad, pero SIN NINGUNA GARANTÍA (vea la FDL por más detalles).

Historial de revisiones

Revisión 0.0 19-06-2002 Revisado por: gs

Documento inicial

Tabla de contenidos

1. Introducción	1
2. Conocimientos básicos	2
2.1. Introducción	2
2.2. Caracteres especiales	2
2.3. Comentarios	2
2.4. Estructura de un archivo LaTeX	2
2.5. Sesión de comandos típica	3
2.6. La disposición del documento	3
2.6.1. Clases de documentos	4
2.6.2. Paquetes	4
2.7. Archivos que se pueden encontrar	5
2.8. Estilos de páginas	6
2.9. Proyectos y documentación de gran tamaño	6
3. Formato del texto	8
3.1. Idioma y estructura del texto	8
3.2. Saltos de línea y saltos de página	8
3.2.1. Párrafos justificados	9
3.2.2. Separación de palabras por sílaba	9
3.3. Texto prefabricado	10
3.4. Símbolos y caracteres especiales	10
3.4.1. Comillas y comillas simples	10
3.4.2. Guiones	10
3.4.3. El carácter ~	11
3.4.4. El símbolo de grado	11
3.4.5. Elipsis (...)	11
3.4.6. Acentos y caracteres especiales	11
3.5. Soporte para lenguaje internacional	12
3.6. El espacio entre palabras	12
3.7. Títulos, capítulos y secciones	13
3.8. Referencias cruzadas	14
3.9. Notas al pie	15
3.10. Palabras destacadas (subrayado, itálica y fuentes)	15
3.11. Ambientes	15
3.11.1. Itemize, Enumerate y description	16
3.11.2. Flushleft, Flushright y Center	16
3.11.3. Quote, Quotation y Verse	17
3.11.4. Verbatim (<i>al pie de la letra</i>)	17
3.11.5. Tabular	18
3.12. Cuerpos flotantes (imágenes y tablas)	19
3.13. Protegiendo comandos débiles	21
4. Especialidades	23
4.1. Incluyendo gráficos PostScript	23
4.2. Bibliografía	23
4.3. Indexado	24
4.4. Fuentes y tamaños	25
4.4.1. Comandos para cambiar de fuente	25
4.5. Espaciado	26
4.5.1. Espaciado entre líneas	27
4.5.2. Formato de párrafos	27

4.5.3. Espacio horizontal	27
4.5.4. Espacio vertical.....	28
4.6. Disposición de la página	28
4.7. Cajas	29
4.8. Reglas y contoneos.....	30

Lista de tablas

3-1. Algunos ejemplos de texto prefabricado	10
3-2. Tipos de guiones en LaTeX	11
4-1. Ejemplos del comando " <i>\index</i> "	24
4-2. Comandos para fuentes.....	25
4-3. Comandos para tamaños.....	25
4-4. Tamaños de las distintas fuentes para las distintas clases.....	26
4-5. Unidades de medida manejadas en LaTeX	27
4-6. Comandos para establecer los márgenes en LaTeX	29

Capítulo 1. Introducción

LaTeX es un procesador de documentos muy conveniente para la producción de documentos científicos y matemáticos de gran calidad tipográfica. El sistema LaTeX también es conveniente para producir una gran cantidad de tipos distintos de documentación, desde simples cartas hasta libros completos. LaTeX utiliza TeX como su motor a la hora de darle formato a los documentos.

Este documento busca impulsar al lector de la forma mas rápida posible a la utilización de LaTeX como su procesador de documentos. Si el lector desea profundizar sus conocimientos sobre LaTeX, le recomiendo visite la página oficial del proyecto LaTeX (www.latex-project.org (<http://www.latex-project.org>)), donde podrá encontrar una gran variedad de documentación sobre este sistema. Por cualquier otro tipo de material relacionado con LaTeX, diríjase a la CTAN (Comprehensive TeX Archive Network, www.ctan.org (<http://www.ctan.org>)). Es recomendable que visite también el proyecto *CervanTeX* ("LaTeX en español", <http://apolo.us.es/CervanTeX/CervanTeX.html>).

LaTeX está disponible para la mayoría de las plataformas actuales, desde PC's Intel, Mac hasta varios sistemas UNIX. El objetivo de este documento no es ilustrar al lector de cómo instalar LaTeX en su computador, sino el como escribir documentos para que puedan ser procesados con LaTeX.

Capítulo 2. Conocimientos básicos

2.1. Introducción

Los documentos en LaTeX se crean en un archivo de texto plano ASCII. LaTeX no hace diferencias entre espacios y tabulaciones en el texto. Varios caracteres-espacio consecutivos son tratados como un único espacio. Espacios al principio de una línea son generalmente ignorados. Un solo retorno de carro es interpretado como un espacio. Una línea vacía entre dos líneas define el final de un párrafo. Varias líneas vacías consecutivas son interpretadas como una única línea vacía.

2.2. Caracteres especiales

Los caracteres especiales son los siguientes:

```
# $ % ^ & _ { } ~ \
```

Con la excepción del propio backslash, estos pueden ser imprimibles agregándoles antes un carácter backslash `\` de la siguiente forma:

```
\# \$ \% \^ \& \_ \{ \} \~
```

Dos backslashes seguidos (`\\`) son interpretados como un retorno de carro. Para imprimir un backslash se debe ingresar:

```
\\backslash$
```

Los comandos de LaTeX son *Case sensitive* y comienzan con un backslash. LaTeX ignora espacios en blanco después de un comando. Para generar un espacio después de un comando se debe escribir `{ }` exactamente luego del comando y a continuación un espacio. La secuencia `{ }` le dice a LaTeX que pare de "comerse" los espacios luego del comando.

2.3. Comentarios

Todo lo que esté después de un `%` en una misma línea es un comentario (comentario en una línea). Para crear comentarios largos se debe ingresar:

```
\begin{comment}  
...comentario...  
\end{comment}
```

2.4. Estructura de un archivo LaTeX

Todo archivo debe comenzar con el comando:

```
\documentclass{...}
```

Este comando especifica que tipo de documento se va a escribir. Luego se pueden incluir comandos que influenciarán el estilo de todo el documento o cargar paquetes que agregarán nuevas utilidades al sistema LaTeX. El comando para cargar paquetes es:

```
\usepackage{...}
```

Cuando todo el trabajo de seteo está hecho, se comienza el cuerpo del documento con el comando:

```
\begin{document}
```

Luego se ingresa el texto del documento junto con comandos LaTeX y se termina el mismo con el comando:

```
\end{document}
```

Todo lo que esté después de este comando será ignorado. Ejemplo de documento LaTeX:

```
\documentclass[a4paper,11pt]{article}
% define el titulo
\author{H.~Partl}
\title{Minimalismo}
\begin{document}
% genera el titulo
\maketitle
% inserta la table de contenidos
\tableofcontents
\section{Start}
Bien, aqui comienza mi articulo.
\section{End}
\ldots{} y aqui termina.
\end{document}
```

2.5. Sesión de comandos típica

1. Se crea el documento en formato LaTeX con cualquier editor de texto plano.
2. Se ejecuta el comando:


```
$ latex archivo.tex
```

 Este genera el documento en formato DVI (archivo.dvi).
3. Se puede visualizar el documento en formato DVI:


```
$ xdvi archivo.dvi
```
4. o pasarlo a formato PostScript:


```
$ dvips -Pcmz archivo.dvi -o archivo.ps
```

2.6. La disposición del documento

2.6.1. Clases de documentos

Para indicar la clase de documento que se va a procesar se utiliza el comando "`\documentclass`" que tiene la siguiente sintaxis:

```
\documentclass[opciones]{clase}
```

El parámetro "*clase*" indica la clase del documento a procesar. Las clases de documentos posibles en LaTeX2e son los siguientes:

- "*article*" para artículos, presentaciones, reportes cortos, documentación de programas, invitaciones, etc.
- "*report*" para reportes más largos conteniendo varios capítulos, libros cortos, etc.
- "*book*" para libros reales.
- "*slides*" para volantes, avisos de una página, etc. Esta clase utiliza fuentes de gran tamaño.

El parámetro "*opciones*" personaliza el comportamiento de la clase del documento. Las opciones deben ser separadas por comas. Las opciones más comunes son las siguientes:

- "*10pt,11pt,12pt*" indica el tamaño de la fuente principal del documento. "*10pt*" es el valor por omisión.
- "*a4paper,letterpaper,...*" define el tamaño del papel. Otros tipos de papel son *a5paper,b5paper,legalpaper,executivepaper*. *letterpaper* es el valor por omisión.
- "*fleqn*" indica que las fórmulas matemáticas estén alineadas a la izquierda en vez de centradas.
- "*leqno*" posiciona la numeración de las fórmulas a la izquierda de cada una y no a la derecha.
- "*titlepage,notitlepage*" especifica si se debe comenzar una nueva página luego de imprimir el título del documento. Las clases *report* y *book* lo hacen automáticamente, mientras que la clase *article* no.
- "*onecolumn,twocolumn*" indica si el documento debe ser generado en una o dos columnas.
- "*twoside,oneside*" indica si el documento debe ser arreglado para una impresión de doble faz (de los dos lados de cada página). La clase *book* usa "*twoside*" por omisión, mientras que las clases *report* y *article* no. El usar la opción *twoside* NO le indica a la impresora que realice una impresión de doble faz, esto corre por cuenta del usuario, esto se utiliza por un tema de formato del documento.
- "*openright,openany*" indica si los capítulos del documento deben comenzar solo en páginas ubicadas a la derecha o en cualquiera. Esto no funciona con la clase *article* (esta no utiliza capítulos). La clase *report* comienza los capítulos en la siguiente página disponible (por omisión), mientras que la clase *book* lo hace en páginas a la derecha.

Ejemplo del comando "`\documentclass`":

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

2.6.2. Paquetes

Para resolver problemas que el LaTeX básico no resuelve, se agregan paquetes para aumentar las capacidades de LaTeX. Por ejemplo, para incluir gráficos, texto coloreado o código fuente proveniente de otro archivo. El comando para agregar paquetes tiene la siguiente sintaxis:

```
\usepackage[opciones]{paquete}
```

donde "*paquete*" es el nombre del paquete a incluir y "*opciones*" es una lista de argumentos que personalizan el comportamiento del paquete. Paquetes que vienen con la distribución básica de LaTeX2e:

- "*doc*" permite la documentación de programas LaTeX. Descrito en "*doc.dtx*".
- "*exscale*" provee versiones de la fuente extendida de matemáticas. Descrito en "*ltxscale.dtx*".
- "*fontenc*" especifica que codificación de fuente se debe utilizar. Descrito en "*loutenc.dtx*".
- "*ifthen*" provee comandos de la forma:

```
if...then...do...otherwise...do...
```

 Descrito en "*ifthen.dtx*".
- "*latexsym*" para acceder a la fuente de símbolos de LaTeX. Este paquete es muy útil. Descrito en "*latexsym.dtx*".
- "*makeidx*" provee comandos para generar índices.
- "*syntonly*" procesa un documento solo revisando su sintaxis.
- "*inputenc*" permite la especificación de la codificación de entrada del documento, como ser ASCII, ISO Latin-1, ISO Latin-2, páginas de código 437/850 de IBM, Apple Macintosh, Next, ANSI-Windows o una definida por el usuario. Descrito en "*inputenc.dtx*".

Para ver la descripción de cada uno de estos paquetes, ejecutar LaTeX sobre su archivo correspondiente.

2.7. Archivos que se pueden encontrar

Trabajando con LaTeX se utilizan varios formatos para los distintos archivos con los que se trabaja. ésta es una lista de las extensiones utilizadas para cada tipo de archivo:

- ".*tex*" Archivo de entrada LaTeX. Compilado con LaTeX.
- ".*sty*" *LaTeX Macro Package*. Este es un archivo que se puede cargar en el documento usando el comando "*\usepackage*".
- ".*dtx*" TeX documentado. Formato de distribución principal para archivos de estilo de LaTeX. Si se procesa un archivo de este tipo, se obtiene un código documentado del paquete contenido en este archivo.
- ".*ins*" Instalador de los archivos contenidos en el archivo ".*dtx*" correspondiente. Si se baja un paquete de la red, se obtendrán un archivo ".*ins*" y uno ".*idx*". Se debe ejecutar LaTeX sobre el archivo ".*ins*" para instalar el archivo ".*idx*" y con él, el paquete deseado.
- ".*cls*" Archivo de clase. Define como el documento debe lucir. Este archivo es cargado con el comando "*\documentclass*".

Los siguientes tipos de archivo son generados cuando se ejecuta LaTeX en el archivo de entrada

- *".dvi"* *Device Independent File*. El resultado principal de la compilación de un archivo de entrada LaTeX.
- *".log"* Brinda información detallada de lo que sucedió en la última compilación generada con LaTeX.
- *".toc"* Contiene todas las secciones de cabezales. Es leído para la siguiente compilación y se utiliza para generar la tabla de contenidos.
- *".lof"* Como *".toc"* pero para la lista de imágenes.
- *".lot"* Como *".toc"* pero para la lista de tablas.
- *".aux"* Otro archivo que transporta información desde una compilación hacia la siguiente. Entre otras cosas, contiene información sobre las referencias cruzadas.
- *".idx"* Si el documento contiene un índice, LaTeX guarda todas las palabras que van dentro del mismo en este archivo. Se debe procesar este archivo con el comando *"makeindex"*.
- *".ind"* Es el archivo *".idx"* luego de procesado, listo para ser incluido en el siguiente ciclo de compilación.
- *".ilg"* Como *".log"* pero para el comando *"makeindex"*.

2.8. Estilos de páginas

LaTeX soporta tres combinaciones de cabezales y pies de páginas predeterminadas llamadas *estilos*. El comando para especificar el estilo de página es:

```
\pagestyle{estilo}
```

donde el parámetro *"estilo"* define el estilo a utilizar. Los posibles estilos son los siguientes:

- *"plain"* imprime los números de páginas al final de cada página en el medio del pie de página. Estilo de página por omisión.
- *"headings"* imprime el nombre del capítulo actual y el número de página en el cabezal de cada página, mientras en pie de página queda vacío.
- *"empty"* deja tanto el cabezal como el pie de página vacíos.

Es posible cambiar el estilo de la página actual con el comando:

```
\thispagestyle{estilo}
```

Más adelante veremos como crear cabezales y pies de páginas personalizados.

2.9. Proyectos y documentación de gran tamaño

Cuando se trabaja con documentos grandes, se podría desear dividir el archivo de entrada en varias partes. LaTeX posee dos comandos para solucionar dicho problema. Con el comando:

```
\include{nombre_archivo}
```

ubicado en el cuerpo del documento podemos insertar el contenido de otro archivo llamado "*nombre_archivo.tex*". LaTeX comenzará una nueva página antes de procesar el contenido del archivo externo. Con el comando:

```
\includeonly{archivo1,archivo2,...}
```

ubicado al comienzo del documento se puede especificar cuales de los archivos incluidos en el correr del documento deben realmente ser incluidos y cuales no. El comando "*include*" comienza a procesar el texto incluido en una nueva página. Esto es útil cuando se utiliza "*includeonly*", ya que los saltos de página no se moverán, incluso si se omiten algunos archivos a incluir. Pero a veces este efecto puede no ser deseado; en este caso se utiliza el comando:

```
\input{archivo}
```

Este comando incluye el archivo especificado y nada más. Sin complicaciones.

Para decirle a latex que revise la sintaxis de nuestro documento se puede utilizar el paquete "*syntonly*". Este paquete hace que LaTeX recorra todo el documento revisando la sintaxis y la utilización de comandos, pero sin generar ninguna salida en formato DVI. LaTeX corre mas rápido en este modo y puede ahorrar tiempo para cuando se procesan grandes documentos. Su utilización es muy simple:

```
\usepackage{syntonly}  
\syntonly
```

Cuando se desee producir la salida en formato DVI, coméntese el segundo renglón agregando un signo de porcentaje al comienzo de la línea.

Capítulo 3. Formato del texto

3.1. Idioma y estructura del texto

El objetivo principal de escribir un texto es el transmitir ideas, información y conocimiento al lector. El mismo comprenderá mejor el texto si éste tiene sus ideas bien estructuradas, y verá mejor esta estructura si la forma tipográfica refleja la estructura lógica y semántica del contenido del documento. Con LaTeX debe ser especificada la estructura lógica y semántica del texto, así como también la forma tipográfica del texto (esto último a través de la reglas dadas en la clase del documento y en varios archivos de estilo).

La más importante unidad de texto en LaTeX (y en tipografía) es el párrafo. A éste se le llama justamente "*unidad de texto*", porque un párrafo es la forma tipográfica que debe expresar un pensamiento coherente, o una idea. Se mostrará a continuación como forzar saltos de línea (por ejemplo con "\") y saltos de párrafo (por ejemplo dejando una línea vacía en el documento fuente). Por lo tanto, si un nuevo pensamiento comienza, un nuevo párrafo debe comenzar, y si no, sólo saltos de línea deben ser utilizados. Errores con este tipo de sintaxis son muy comunes de cometerse cuando se utilizan ecuaciones en el texto. En los siguientes ejemplos se muestra como a veces líneas en blanco (saltos de párrafo) son utilizadas antes y después de una ecuación y a veces no. Si hay comandos en los ejemplos que no entiende todavía, no se preocupe, los mismos serán vistos más adelante en el documento.

```
% Ejemplo 1
\ldots cuando Einstein introdujo su formula
\begin{equation}
e = m \cdot c^2 \ ; \ ,
\end{equation}
la cual es al mismo tiempo la mas conocida y la
menos bien entendida de la formulas fisicas.

% Ejemplo 2
\ldots from which follows Kirchoff s current law:
\begin{equation} \sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
Kirchhoff s voltage law can be derived
\ldots

% Ejemplo 3
\ldots which has several advantages.
\begin{equation} I_D = I_F - I_R
\end{equation} is the core of a very different transistor model. \ldots
```

La siguiente unidad de texto mas pequeña es la oración. Finalmente los párrafos de un texto deben ser estructurados lógicamente en un nivel superior, agrupándolos en capítulos, secciones, subsecciones, etc. Sin embargo, el efecto tipográfico de escribir

```
\section{La estructura del idioma y texto}
```

es tan obvio que es casi auto-evidente cómo estas estructuras de alto nivel deben ser utilizadas.

3.2. Saltos de línea y saltos de página

3.2.1. Párrafos justificados

LaTeX realiza la justificación de los textos automáticamente (cuando hablamos de justificación de textos hablamos de que todas las líneas del texto tengan el mismo largo). El tipo de alineación de un párrafo depende de la clase del documento. En algún caso especial puede ser necesario indicarle a LaTeX que divida una línea. Para hacer esto se puede escribir una de las siguientes opciones:

- `"\|", "\newline"` Comienzan una nueva línea sin comenzar un nuevo párrafo.
- `"\|*"` Adicionalmente prohíbe un salto de página luego del salto de línea forzado.
- `"\newpage"` Comienza una nueva página.
- `"\linebreak[n]"` Salto de línea.
- `"\nolinebreak[n]"` No hacer salto de línea.
- `"\pagebreak[n]"` Salto de página.
- `"\nopagebreak[n]"` No hacer salto de página.

Los últimos cuatro comandos de esta lista habilitan al autor a influenciar sus acciones con el parámetro opcional `"n"`. Este parámetro puede ser un número entre cero y cuatro. Si `"n"` es un valor menor a cuatro, se le está permitiendo a LaTeX el ignorar el comando si el resultado de aplicar el mismo se verá muy mal. No confundir los comandos que utilizan la palabra `"break"` en vez de la palabra `"new"`, no son lo mismo. Si realmente se desea comenzar una nueva línea, utilícese el comando `"\newline"`.

3.2.2. Separación de palabras por sílaba

LaTeX permite separar palabras según sus sílabas para mejorar el justificado (por ejemplo). LaTeX no separa palabras que tengan caracteres con acentuación u otro tipo de carácter no tradicional para la lengua inglesa. Para definir alguna palabra fuera de las comunes para que sea dividida se utiliza el comando:

```
\hyphenation{lista}
```

donde `"lista"` es una lista de palabras separadas por un espacio que tienen un guión ("-") en cada lugar donde pueden ser separadas. Veamos un ejemplo. El comando:

```
\hyphenate{FORTRAN su-per-mer-ca-do}
```

permite que la palabra `"supermercado"` (o también la palabra `"Supermercado"`) sea separada en cualquiera de sus sílabas, mientras que le prohíbe a la palabra `"FORTRAN"` (o `"Fortran"`, o también `"fortran"`) ser dividida en cualquiera de sus sílabas. Este comando debe ser ubicado al comienzo del texto. Por otra parte, existe una definición en sílabas en el correr mismo del documento. El comando para esto es:

```
" \- "
```

Por ejemplo, se podría escribir en el medio de un párrafo el texto

```
"su\ -per\ -mer\ -ca\ -do"
```

produciendo el mismo efecto que en el ejemplo antes mencionado. Este comando es útil cuando se tienen palabras acentuadas.

Se puede mantener un grupo de palabras siempre junto con el comando:

```
\mbox{texto}
```

éste causa que toda palabra contenida en "texto" permanezca junto a las otras bajo cualquier circunstancia. El comando:

```
\fbox{texto}
```

es similar a "*mbox*" con la diferencia que se dibujará un recuadro alrededor del texto en cuestión.

3.3. Texto prefabricado

Tabla 3-1. Algunos ejemplos de texto prefabricado

COMANDO	EJEMPLO	DESCRIPCION
<code>\today</code>	9 Agosto 2001	Fecha actual en el idioma actual
<code>\TeX</code>	--	La palabra <i>TeX</i> escrita según su logo
<code>\LaTeX</code>	--	La palabra <i>LaTeX</i> escrita según su logo
<code>\LaTeX2e</code>	--	La palabra <i>LaTeX2e</i> escrita según su logo

3.4. Símbolos y caracteres especiales

3.4.1. Comillas y comillas simples

No se deberían utilizar las "comillas" como en un procesador de texto. Para representar comillas en el texto, enciérrele la porción de texto entre los caracteres " y " como en el ejemplo:

```
"El veloz murcielago volaba entre los arboles sin golpearse, ciego."
```

Si se desean expresar comillas simples, se utiliza el mismo criterio, pero encerrando el texto entre un solo ' y un solo ', de la siguiente forma:

```
'El veloz murcielago volaba entre los arboles sin golpearse, ciego.'
```

3.4.2. Guiones

LaTeX conoce cuatro tipos de guiones. Los mismos son los siguientes:

Tabla 3-2. Tipos de guiones en LaTeX

NOMBRE	REPRESENTACION	EJEMPLO
hyphen	"_"	"físico-mental"
en-dash	"--"	--
em-dash	"---"	--
Signo de menos	"-"	"\$-1\$" (matemática)

3.4.3. El carácter ~

Hay dos maneras de representar este carácter, y producen resultados distintos, aquí se brindan dos ejemplos que se sugiere se prueben para ver la diferencia:

```
http://www.rich.edu/~{ }bush
http://www.clever.edu/$\sim$demo
```

3.4.4. El símbolo de grado

Este es un ejemplo de como imprimir el símbolo de grado en LaTeX:

Hay $\$-30\text{,}^{\circ}\mathrm{C}\$,$ hace mucho frío.

3.4.5. Elipsis (...)

La forma correcta de representar una elipsis (tres puntos seguidos) es poniendo el comando:

```
\ldots
```

inmediatamente después de la palabra que debe preceder a la elipsis.

3.4.6. Acentos y caracteres especiales

LaTeX soporta el uso de acentos y caracteres especiales de varios idiomas. Se representaran todos los acentos posibles para la letra "o" (y se da por entendido que funciona para cualquier otra vocal), así como también un

gran número de caracteres especiales. Se sugiere probar los mismos para ver su resultado gráfico.

```
\'o \'o \^o \~o
\=o \.o \" "o \c c
\u o \v o \H o \c o
\d o \b o \t oo
\oe \OE \ae \AE
\aa \AA
\o \O \l \L
\i \j i ÿ
```

3.5. Soporte para lenguaje internacional

El idioma por omisión utilizado por LaTeX es el inglés, por lo tanto se deben tener algunas consideraciones antes de comenzar a escribir un documento en castellano utilizando LaTeX:

1. Todas las cadenas de texto automáticamente generadas deben ser adaptadas al nuevo idioma. Esto se logra utilizando el paquete "*babel*".
2. LaTeX necesita saber las reglas de separado de sílabas del nuevo lenguaje. Esto se logra cambiando el archivo de formato con nuevas reglas de separado de sílabas.

Para utilizar el paquete agréguese el comando:

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

después del comando "*documentclass*" y sustituyendo la palabra "*lenguaje*" por el idioma de su elección. Los idiomas soportados por su distribución deben estar listados en la Guía Local. El paquete *Babel* activará automáticamente las reglas apropiadas de separado de sílabas. Si el idioma de su elección no es soportado por LaTeX, *Babel* seguirá funcionando de todos modos, pero deshabilitará las reglas de separado de sílabas, lo que trae un efecto contraproducente. Para algunos lenguajes, *Babel* especifica nuevos comandos que facilitan la entrada de caracteres especiales. Algunos sistemas permiten ingresar caracteres especiales directamente desde el teclado, LaTeX puede manejar dichos caracteres.

3.6. El espacio entre palabras

Para conseguir un margen derecho recto, LaTeX inserta una cantidad variable de espacios entre cada palabra. También inserta una leve mayor cantidad de espacios al final de una oración para hacer el texto más legible. LaTeX asume que las oraciones terminan con puntos, signos de interrogación o de exclamación. Si un punto es precedido por una letra mayúscula, éste no es interpretado como el final de una oración, sino como el final de una abreviación. Cualquier excepción de estos criterios debe ser especificada por el autor. Una contrabarra ("\") frente a un espacio generará un espacio que no será alargado. Un carácter "~" generará un espacio que no podrá ser alargado y que también prohibirá un salto de línea. El comando:

```
\@
```

frente a un punto especifica que este punto termina una oración, aun si éste es precedido por una letra mayúscula. Un ejemplo podría ser el siguiente:

Me gusta BASIC\@. A ti?

El espacio adicional después de puntos puede ser deshabilitado con el comando:

```
\frenchspacing
```

que le dice a LaTeX que no ingrese más espacio después de un punto que el carácter ordinario. Si se utiliza el comando "*frenchspacing*", no es necesario utilizar el comando "\@".

3.7. Títulos, capítulos y secciones

Para ayudar al lector a encontrar su camino a través del texto lo aconsejable es dividir el mismo en capítulos, secciones y subsecciones. LaTeX soporta este tipo de divisiones con comandos que toman el título de la sección como su parámetro. Corre por cuenta del autor el uso de los mismos en el orden indicado. Los siguientes comandos de seccionado están disponibles para la clase "*article*".

```
\section{...}
\paragraph{...}
\subsection{...}
\subparagraph{...}
\subsubsection{...}
```

Pueden ser utilizados dos comandos de seccionado adicionales para las clases "*book*" y "*report*":

```
\part{...}
\chapter{...}
```

Como la clase "*article*" no conoce capítulos ("*chapters*"), es muy fácil agregar artículos en forma de capítulo a un libro. El espacio entre secciones, el numerado y el tamaño de fuente de los títulos será designado automáticamente por LaTeX. Dos de los comandos de seccionado son un poco especiales:

- El comando "*part*" no influye en el numerado de los capítulos.
- El comando "*appendix*" no toma parámetros, sólo cambia el numerado de los capítulos a letras.

LaTeX crea una tabla de contenidos tomando los cabezales de las secciones y los números de página desde el último ciclo de compilación del documento. El comando:

```
\tableofcontents
```

crea una tabla de contenidos en el lugar donde es escrito el mismo. Un nuevo documento debe ser compilado dos veces para conseguir una tabla de contenidos correcta. Existen oportunidades donde será necesario compilar el documento por una tercera vez. LaTeX avisará cuando esto sea necesario.

Todos los comandos de seccionado anteriormente nombrados también existen en su versión "*estrellada*". La versión estrellada de un comando es creada agregando un carácter "*" luego del nombre del comando. Estos generan cabezales de secciones que no aparecen en la tabla de contenidos y que no son numerados. Por ejemplo, el comando:

```
\section{Ayuda}
```

se podría escribir como:

```
\section*{Ayuda}
```

Normalmente los cabezales de sección aparecen en la tabla de contenidos exactamente en el lugar que fueron ingresados en el texto. A veces esto no es posible, porque el cabezal es muy largo como para caber en la tabla de contenidos. El texto para la tabla de contenidos puede ser especificado como un argumento opcional frente al cabezal actual. Por ejemplo:

```
\chapter[Titulo para la tabla de contenidos]{Un titulo largo y aburrido mostrado en el texto}
```

El título general del documento es generado ingresando el comando:

```
\maketitle
```

El contenido del título debe ser definido por los comandos:

```
\title{...}
\author{...}
\date{...} % ESTE ULTIMO ES OPCIONAL
```

En el parámetro del comando "*\author*" es posible ingresar varios nombres separados por comandos del tipo:

```
\and
```

Aparte de los comandos de seccionado nombrados anteriormente, LaTeX2e introduce tres comandos adicionales para ser utilizados con la clase "*book*". éstos son útiles para dividir una publicación. Estos comandos alteran los cabezales de los capítulos y el numerado de páginas para trabajar como es de esperarse en un libro:

- "*\frontmatter*" debería ser el primer comando a ingresar luego de "*\begin{document}*". éste generará el numerado de páginas con números romanos.
- "*\mainmatter*" debe situarse justo antes del primer capítulo del documento. éste cambia a numerado de página Arábico y reinicia el contador de páginas.
- "*\appendix*" indica el comienzo de material adicional en el documento. Luego de este comando los capítulos serán numerados con letras.
- "*\backmatter*" debería ser ingresado antes de los últimos ítems del documento, como ser la bibliografía y el índice. En las clases de documentos estándar, este comando no tiene efectos visuales.

3.8. Referencias cruzadas

En libros, reportes y artículos, existen referencias cruzadas a figuras, tablas y segmentos de texto especiales. LaTeX provee los siguientes comandos para crear referencias cruzadas:

```
\label{marcador}
\ref{marcador}
\pageref{marcador}
```

donde el parámetro "*marcador*" es un identificador elegido por el usuario. En el texto final, el comando "*\ref*" aparecerá como el número de sección, subsección, figura, tabla o teorema que representa su comando "*\label*" correspondiente; mientras que el comando "*\pageref*" aparecerá como el número de página en el que se encuentra su comando "*\label*" correspondiente. Un ejemplo podría ser el siguiente:

Una referencia a esta subseccion `\label{sec:this}` luce como:
"vea la seccion~\ref{sec:this} en la pagina~\pageref{sec:this}."

3.9. Notas al pie

Con el comando:

```
\footnote{texto}
```

una nota al pie es impresa al final de la página actual. Las notas al pie deberían ser ubicadas siempre luego de la palabra a la que se refieren. Las notas al pie que refieren a una oración o parte de ella deberían ser ubicadas luego de una coma o el punto. Un ejemplo podría ser el siguiente:

```
Las notas al pie\footnote{Esto es una nota al pie.} son muy utiles.
```

3.10. Palabras destacadas (subrayado, itálica y fuentes)

Si un texto es escrito con una máquina de escribir, las palabras que se desean destacar son subrayadas. Esto se logra con el comando:

```
\underline{texto}
```

En libros impresos, sin embargo, las palabras son destacadas imprimiéndolas con una fuente itálica. Esto se logra con el comando:

```
\emph{texto}
```

Lo que realmente hace este comando con el texto de su parametro depende del contexto. Por ejemplo:

```
\emph{Si se utiliza el comando de destaque dentro de un texto que ya  
esta destacado, \LaTeX{} utiliza la fuente \emph{normal} para  
destacar el texto en cuestion.}
```

También se le puede pedir a LaTeX que cambie la fuente del documento, los comandos son los siguientes:

- `"\textit{texto}"` produce el efecto itálica (al igual que `"\emph{"`).
- `"\textsf{texto}"` escribe el texto con la fuente `"Sans Serif"`.
- `"\texttt{texto}"` escribe el texto como si fuera una máquina de escribir.

3.11. Ambientes

Un ambiente se define de la siguiente forma:

```
\begin{ambiente} ...texto... \end{ambiente{ambiente}}
```

siendo el parámetro "*ambiente*" el nombre del ambiente. Los ambientes pueden ser declarados varias veces a lo largo del documento sin importar su orden, siempre y cuando se respete el orden de apertura y cerrado de los mismos. Por ejemplo:

```
\begin{aaa}
...
\begin{bbb}
...
\end{bbb}
...
\end{aaa}
```

A continuación se explicarán los ambientes más importantes.

3.11.1. Itemize, Enumerate y description

El ambiente "*itemize*" es útil para crear listas simples, el ambiente "*enumerate*" para listas numeradas, y el ambiente "*description*" para descripciones. A continuación se presenta un ejemplo para probar:

```
\flushleft
\begin{enumerate}
\item Puedes mezclar los ambientes de lista a tu gusto:
\begin{itemize}
\item Pero podrian llegar a verse mal.
\item[-] Con un guion.
\end{itemize}
\item Por lo cual debes recordar:
\begin{description}
\item[Stupid] Las no se veran bien por el hecho de que esten en una lista.
\item[Smart] Sin embargo, las cosas se pueden mostrar bellas en una lista.
\end{description}
\end{enumerate}
```

3.11.2. Flushleft, Flushright y Center

Los ambientes "*flushleft*" y "*flushright*" generan párrafos alineados a la izquierda y derecha respectivamente. El ambiente "*center*" genera texto centrado. Si no se especifican saltos de línea con el comando "\\\", LaTeX los determinará automáticamente. A continuación se presentan algunos ejemplos descriptivos:

```
\begin{flushleft}
Este texto esta\\ alineado a la izquierda.
\LaTeX{} no esta tratando de dejar cada linea del mismo largo.
```

```
\end{flushleft}

\begin{flushright}
Este texto esta \\alineado a la derecha.
\LaTeX{} no esta tratando de dejar cada linea del mismo largo.
\end{flushright}

\begin{center}
En el centro\\de la tierra
\end{center}
```

3.11.3. Quote, Quotation y Verse

El ambiente *"quote"* es útil para agregar citas, frases importantes y ejemplos en el texto. Por ejemplo:

```
Una regla tipografica para el largo de una linea es:
\begin{quote}
En promedio, ninguna linea deberia superar los 66 caracteres.
\end{quote}
Es por esta razon que las paginas de LaTeX{} tienen bordes tan largos
por defecto y tambien es por esta razon que se usa la impresion en
varias columnas en los periodicos.
```

Existen dos ambientes similares al *"quote"*: *"quotation"* y *"verse"*. El ambiente *"quotation"* es útil para citas largas que llevarán varios párrafos, porque realiza párrafos *"dentados"*. El ambiente *"verse"* es útil para poemas, donde los saltos de línea son importantes. Las líneas son separadas ingresando un *"\\"* al final de cada línea y una línea vacía al final de cada verso. Por ejemplo:

```
Solo se un poema en ingles de memoria. Habla de Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty hay a great fall.\\
All the King's horses and all the King's men\\
Couldn't put Humpty together again.
\end{verse}
\end{flushleft}
```

3.11.4. Verbatim (*al pie de la letra*)

Todo texto que se encuentre dentro del ambiente *"verbatim"* será directamente impreso, como si fuera escrito en una máquina de escribir, con todos los saltos de líneas y espacios, sin ser ejecutado ningún comando LaTeX. Para obtener el mismo efecto dentro de un párrafo se puede utilizar el comando:

```
\verb+texto+
```

sustituyendo *"texto"* con lo que se desea imprimir sin ser procesado. El carácter *"+"* es solo un ejemplo de carácter delimitador, este podría ser cualquier carácter con la excepción de letras, el asterisco (*"*"*) y el espacio.

El ambiente *"verbatim"* y el comando *"verb"* no deben ser utilizados dentro de parámetros de otros comandos.

3.11.5. Tabular

El ambiente *"tabular"* puede ser utilizado para crear tablas con líneas verticales y horizontales opcionales. LaTeX determina el ancho de las columnas automáticamente. El parámetro *"formato"* del comando:

```
\begin{tabular}{formato}
```

define el formato de la tabla. Se pueden utilizar los siguientes parámetros:

- *"l"* Columna con texto alineado a la izquierda.
- *"r"* Columna con texto alineado a la derecha.
- *"c"* Columna con texto centrado.
- *"p{ancho}"* Columna con texto justificado y saltos de línea.
- *"|"* Línea vertical.

En el ambiente *"tabular"*:

- *"&"* salta a la siguiente columna.
- *"\\"* comienza una nueva línea.
- *"\hline"* inserta una línea horizontal.
- *"\cline{j-i}"* agrega una línea parcial, donde *"j"* y *"i"* son los números de columna en los cuales la nueva línea se debe extender.

Por ejemplo:

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexagecimal \\
3700 & Octal \\ \cline{2-2}
11111000000 & binario \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

```
\begin{tabular}{|p{4.7cm}|}
\hline
Bienvenido al parrafo de Boxy.
Esperamos que todos disfruten del show. \\
\hline
\end{tabular}
```

El separador de columnas puede ser especificado con la construcción:

```
@{...}
```

Este comando inhibe el espacio entre columnas y los sustituye con lo que se haya puesto entre llaves. Un uso comun de este comando es explicado a continuación. Trata el problema del alineado de cifras decimales.

También se utiliza comunmente para eliminar espacios no deseados en la tabla. Pruébense los siguientes códigos y compárense:

```
\begin{tabular}{@{} l @{}}
\hline
sin espacios\\
\hline
\end{tabular}

\begin{tabular}{l}
\hline
con espacios a la derecha y a la izquierda\\
\hline
\end{tabular}
```

No existe manera manera de alinear columnas decimales a un punto decimal en una instalación estándar. Pero se puede realizar un pequeño truco: un entero alineado a la derecha y una fracción alineada a la izquierda. El comando:

```
@{.}
```

en la línea "`\begin{tabular}`" reemplaza el espacio entre columnas con un punto ("."), dando la apariencia de una sola columna justificada al nivel del punto decimal. No se debe olvidar de reemplazar los puntos decimales de los números por un separador de columnas ("&"). Una etiqueta de columna puede ser ubicada sobre nuestra "columna" numérica (que en realidad son dos columnas) utilizando el comando "`\multicolumn`". Véanse los siguientes ejemplos:

```
\begin{tabular}{c r @{.} l}
Expresion Pi &
\multicolumn{2}{c}{Valor} \\
\hline
 $\pi$  & 3.1416 & \\
 $\pi^{\pi}$  & 36.46 & \\
 $(\pi^{\pi})^{\pi}$  & 80662.7 & \\
\end{tabular}

\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Todo material contenido en un ambiente tabular siempre se mantiene dentro de una misma página. Si se desea crear tablas muy extensas, véanse los ambientes "`supertabular`" y "`longtabular`".

3.12. Cuerpos flotantes (imágenes y tablas)

La mayoría de las publicaciones contienen figuras y tablas. Las mismas necesitan un trato especial dentro del documento, porque no pueden quedar partidas a lo largo de las distintas páginas. Un método podría ser comenzar una nueva página cada vez que una figura o tabla es muy grande como para ubicarla en la página actual que se está procesando. Este método dejaría páginas parcialmente vacías, lo que luce muy mal. La solución a este

problema es hacer *"flotar"* la figura o tabla que no cabe en la página actual a la siguiente página, mientras que se sigue llenando la página actual con texto. LaTeX ofrece dos ambientes para cuerpos flotantes, uno para tablas y otro para figuras. Para obtener el mayor provecho de estos dos ambientes, es importante entender como LaTeX maneja el *"flote"* internamente. De otro modo, los flotes se podrían convertir en una fuente de frustración, ya que LaTeX nunca los pone en el lugar que uno desean que estén.

Primero veremos los comandos que LaTeX provee para los flotes: Todo material contenido dentro de un ambiente *"figure"* o *"table"* será tratado como un cuerpo flotante. Ambos ambientes mencionados soportan un parámetro opcional llamado *"especificador de lugar"*:

```
\begin{figure}[especificador de lugar]
```

```
\begin{table}[especificador de lugar]
```

Este parámetro es utilizado para decirle a LaTeX las ubicaciones a donde el cuerpo flotante puede ser desplazado. Un especificador de lugar es creado construyendo una cadena de permisos de lugares para flote. Las posibles opciones para esta cadena son las siguientes:

- *"h"*: "here", en el mismo lugar del texto en donde se realizó la declaración del objeto.
- *"t"*: "top", al comienzo de una página.
- *"b"*: "bottom", al final de una página.
- *"p"*: "page", en una página especial en donde se encuentran sólo objetos flotantes.
- *"!"*: Sin tener en cuenta parámetros internos que podrían detener la inserción del objeto (como ser el mayor número de objetos flotantes permitidos por página).

Por ejemplo, una tabla podría ser declarada de la siguiente forma:

```
\begin{table}[!hbp]
```

El especificador de lugar *"[!hbp]"* le permite a LaTeX ubicar la tabla en el mismo lugar de la declaración (h) o al final de alguna página (b) o en una página especial de objetos flotantes (p), todo esto aún si no luce bien en el documento (!). Si no se ingresa un especificador de lugar, los valores por omisión para el mismo son *"[tbp]"*. LaTeX ubicará cada objeto flotante que encuentre, de acuerdo con el especificador de lugar ingresado por el autor. Si un objeto flotante no puede ser ubicado en la página actual, este es diferido a la cola de figuras o de tablas. Estas colas son del estilo FIFO ("First In First Out"), lo que quiere decir que el primer objeto en entrar en la cola, es el primero en salir, manteniendo este orden a lo largo de toda la lista. Cuando una nueva página es iniciada, primero LaTeX revisa si es posible crear una página especial para cuerpos flotantes con cuerpos de las colas. Si esto no es posible, el primer objeto de cada cola es tratado como si recién hubiese aparecido en el texto: LaTeX trata de ubicarlo de acuerdo con su especificador de lugar (con la excepción de *'h'* que ya no es mas útil). Cualquier nuevo objeto flotante que aparezca en el texto es enviado a su respectiva cola. LaTeX mantiene estrictamente el orden en el que fueron apareciendo los objetos flotantes a lo largo del documento. Esta es la explicación de por qué una figura que no puede ser ubicada desplaza todas las figuras siguientes al final del documento. Por lo tanto, si LaTeX no está ubicando los objetos flotantes donde se espera, comunmente es solo un objeto el que da problemas obstruyendo una o las dos colas FIFO. El brindar una sola ubicación posible en los especificadores de lugar causa problemas. Si el objeto no puede ser ubicado en dicha posición, este se queda atascado, bloqueando los objetos flotantes subsecuentes. En particular no se debería utilizar nunca la opción *'h'*, es tan mala que en las ultimas versiones de LaTeX es sustituida por *'ht'*. Existen algunas cosas más a mencionar acerca de los ambientes *"table"* y *"figure"*. Con el comando:

```
\caption{texto}
```

se puede definir un texto explicativo para el objeto flotante. Un número secuencial y el texto "*Figure*" o "*Table*" (dependiendo del idioma) serán agregados por LaTeX. Los comandos:

```
\listoffigures
```

```
\listoftables
```

operan de forma análoga al comando "*tableofcontents*", imprimiendo una lista de figuras o tablas respectivamente. En estas listas el texto completo creado por el comando "*caption*" es repetido. Si se utilizan textos largos, se deben especificar versiones más cortas de los mismos para ser utilizados en las listas. Esto es logrado ingresando la versión corta entre paréntesis rectos dentro del comando "*caption*":

```
\caption[texto corto]{texto largo}
```

Con los comandos "*label*" y "*ref*" se pueden crear referencias a cuerpos flotantes a lo largo del texto. El siguiente ejemplo dibuja un cuadrado y lo inserta en el documento. Se puede utilizar esto si se desea reservar espacio para imágenes que serán pegadas en el documento finalizado.

```
La figura~\ref{white} es un ejemplo de Arte Pop.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Cinco por cinco centímetros.} \label{white}
\end{figure}
```

En este ejemplo LaTeX intentará con mucho esfuerzo (!) ubicar la figura justo en "*este lugar*" (h) (siempre y cuando la cola esté vacía). Si no es posible, intentará ubicarla al final de la página (b). Si falla al intentar ubicar la imagen en la página actual, determinará cuando es posible crear una página especial para objetos flotantes conteniendo esta imagen y posiblemente tablas contenidas en la cola de tablas. Si no hay suficiente material como para crear una página de esta índole, LaTeX comenzará una nueva página, y nuevamente tratará a esta imagen como recién ingresada en el texto. Bajo ciertas circunstancias puede ser necesario utilizar el comando

```
\clearpage
```

o aún el comando

```
\cleardoublepage
```

El comando "*clearpage*" obliga a LaTeX a ubicar inmediatamente todos los objetos flotantes que están en las colas y comenzar una nueva página. El comando "*cleardoublepage*" comienza una página de mano derecha (en el caso de que el texto se esté creando para imprimirse de los dos lados de las hojas). Se mencionará más adelante como incluir dibujos en formato *PostScript* en documentos LaTeX2e.

3.13. Protegiendo comandos débiles

Texto brindado como argumento de comandos del estilo de "*caption*" o "*section*" pueden aparecer más de una vez a lo largo del documento (Por ejemplo en la tabla de contenidos como en el cuerpo del documento). Algunos comandos suelen fallar cuando son utilizados como parte del parámetro de comandos como "*section*". Estos son llamados "*comandos débiles*". Comandos débiles son, por ejemplo,

```
\footnote
```

```
\phantom
```

Lo que necesitan estos comandos para trabajar es protección. Se pueden proteger introduciendo el comando:

`\protect`

antes de ellos. El comando "*\protect*" solo protege al comando que está en seguida a continuación de él, ni siquiera a sus argumentos. En la mayoría de los casos un "*\protect*" superfluo no hará daño. Por ejemplo:

```
\section{Soy considerado \protect\footnote{y protejo mis notas al pie}}
```

Capítulo 4. Especialidades

4.1. Incluyendo gráficos PostScript

LaTeX provee facilidades básicas para trabajar con cuerpos flotantes (como imágenes o gráficas) con los ambientes *"figure"* y *"table"*. Existen también varias posibilidades para generar gráficos con el LaTeX básico o con un paquete de extensión de LaTeX. Desafortunadamente la mayoría de los usuarios encuentran estos paquetes difíciles de usar. Por lo cual no se abordarán en este documento. Una manera mucho más fácil de incluir gráficos en un documento es generarlos con un software especial (como CorelDraw, Gimp, XFig, etc.) y luego incluir los gráficos finalizados en el documento. Aquí nuevamente, LaTeX ofrece muchas maneras para realizar este trabajo. En esta introducción se discutirá únicamente el uso de gráficos *PostScript* Encapsulados (EPS del inglés *Encapsulated PostScript*), ya que es fácil de hacer y muy utilizado. Para utilizar imágenes en formato EPS, se debe tener una impresora PostScript disponible como salida. Un buen conjunto de comandos para la inclusión de paquetes se encuentra en el paquete *"graphicx"* creado por D. P. Carlisle. Se asumirá que se posee una impresora PostScript y LaTeX instalado con el paquete *"graphicx"*. A continuación se ofrece una lista de pasos para incluir una imagen en un documento:

1. Exportar la imagen desde el software de gráficos en formato EPS.
2. Cargar el paquete *"graphicx"* en el preámbulo del archivo de entrada con el comando:

```
\usepackage[driver]{graphicx}
```

donde *"driver"* es el nombre del programa que convierte el formato DVI a *PostScript* que se posee (por ejemplo *"dvi2ps"*).

3. Usar el comando:

```
\includegraphics[key=valor,...]{archivo}
```

para incluir al archivo *"archivo.eps"* en el documento. El parámetro opcional acepta una lista de parámetros (*"keys"*) separados por comas con sus respectivos valores. Las *"keys"* más importantes son los siguientes:

- *"width"*: Ancho de la imagen.
- *"height"*: Altura de la imagen.
- *"angle"*: ángulo de rotación
- *"scale"*: Escala.

El siguiente ejemplo incluye el gráfico contenido en el archivo *"test.eps"*. Primero, el gráfico es rotado 90 grados y luego es escalado hasta lograr un ancho de 0.5 veces el ancho de un párrafo común:

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

4.2. Bibliografía

Se puede producir una bibliografía con el ambiente *"thebibliography"*. Cada ítem comienza con el comando:

```
\bibitem{marcador}
```

El *"marcador"* es utilizado luego para citar el libro, artículo o papel a lo largo del documento con el comando:

```
\cite{marcador}
```

El numerado de las entradas se genera automáticamente. El parámetro que sigue al comando:

```
\begin{thebibliography}
```

indica el máximo de estos números. En el siguiente ejemplo, *"{99}"* le indica a LaTeX que espere que ninguno de los números de los ítems en la bibliografía supere al 99:

```
H. Partl~\cite{pa} ha propuesto que \ldots
```

```
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl: \emph{German \TeX}, TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Se recomienda la exploración del programa BibTeX para realizar la bibliografía de proyectos grandes.

4.3. Indexado

Una de las mayores utilidades que tiene un libro es su índice. Con LaTeX y el programa de soporte *"makeindex"*, se puede generar un índice fácilmente. En este documento sólo se explicarán los comandos básicos para la creación de índices. Para habilitar la creación de un índice se debe cargar el paquete *"makeidx"* en el preámbulo del archivo de entrada con el comando:

```
\usepackage{makeidx}
```

y los comandos para realizar el indexado se deben habilitar con el comando:

```
\makeindex
```

también en el preámbulo del archivo de entrada. El contenido del índice es especificado con el comando:

```
\index{key}
```

donde *"key"* es la entrada del índice, el identificador. Se deben ingresar los comandos *"\index"* en los lugares a los cuales se desea que las entradas finales del índice apunten. A continuación se brindan una lista de ejemplos que explican la sintaxis del parámetro *"key"* del comando *"\index"*:

Tabla 4-1. Ejemplos del comando *"\index"*

EJEMPLO	ENTRADA DEL ÍNDICE	COMENTARIO
<code>\index{hello}</code>	hello, 1	Entrada simple
<code>\index{hello!Peter}</code>	Peter, 3	Subentrada bajo 'hello'
<code>\index{Sam@\textsl{Sam}}</code>	Sam, 2	Entrada formateada

<code>\index{Lin@\textbf{Lin}}</code>	Lin, 7	Entrada formateada
<code>\index{Jenny textbf}</code>	Jenny, 3	Número de página formateado
<code>\index{Joe textit}</code>	Joe, 5	Número de página formateado

El archivo es procesado nuevamente, este índice ordenado es incluido en el documento en el lugar donde LaTeX encuentra el comando:

```
\printindex
```

4.4. Fuentes y tamaños

4.4.1. Comandos para cambiar de fuente

LaTeX escoge la fuente apropiada y el tamaño de fuente apropiado basándose en la estructura lógica del documento (secciones, notas al pie, etc.). En algunos casos se puede desear cambiar la fuente y el tamaño de la misma a mano. Para hacer esto se pueden utilizar los siguientes comandos:

Tabla 4-2. Comandos para fuentes

COMANDO	FUENTE
<code>\textrm{...}</code>	Roman
<code>\texttt{...}</code>	Typewriter
<code>\textmd{...}</code>	Medium
<code>\textup{...}</code>	Upright
<code>\textsl{...}</code>	Slanted
<code>\emph{...}</code>	Emphasized
<code>\textsf{..}</code>	Sans Serif
<code>\textbf{...}</code>	Bold Face (Negrita)
<code>\textit{...}</code>	Italica
<code>\textsc{...}</code>	Small caps
<code>\textnormal{...}</code>	Fuente normal

Tabla 4-3. Comandos para tamaños

COMANDO	TAMAÑO
<code>\tiny</code>	Diminuta
<code>\scriptsize</code>	Muy diminuta
<code>\footnotesize</code>	No tan pequeña
<code>\small</code>	Pequeña
<code>\normalsize</code>	Normal
<code>\large</code>	Grande

<code>\Large</code>	Mas Grande
<code>\LARGE</code>	Muy Grande
<code>\huge</code>	Enorme
<code>\Huge</code>	Mas Enorme

El tamaño por omisión de cada fuente depende de la clase del documento y de sus opciones. Los tamaños por omisión para las clases más comunes son los siguientes:

Tabla 4-4. Tamaños de las distintas fuentes para las distintas clases

COMANDO	10pt option (default)	11pt option	12pt option
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

A continuación se muestra un ejemplo de como utilizar los comandos:

```
{\small The small and \textbf{bold} Romans ruled}
{\Large all of great big \textit{Italy}.}
```

En LaTeX2e los atributos de las fuentes son independientes. Esto quiere decir que, por ejemplo, se puede cambiar el tipo de fuente y seguir manteniendo el seteo de negrita indicado anteriormente. En la utilización de los comandos de tamaño de fuente, las llaves son muy importantes, ya que indican el área de texto a la que afecta el comando. Por ejemplo:

A el le gustan las palabras {\LARGE grandes y {\small small} pequeñas}.

Los comandos para cambiar el tamaño de la fuente también cambian el espaciado entre líneas, pero sólo si el párrafo termina dentro del comando de tamaño de fuente. Si se desea activar un comando de cambio de tamaño de fuente para un párrafo entero o incluso para algún área mayor de texto, conviene utilizar el comando como si fuese un ambiente. Por ejemplo:

```
\begin{Large}
Esto no es verdad. Pero, que lo es en estos dias \ldots
\end{Large}
```

Esto ahorrará muchas llaves "{}" incómodas.

4.5. Espaciado

4.5.1. Espaciado entre líneas

Si se desea utilizar un espaciado mayor entre líneas en un documento, se puede cambiar su valor introduciendo el comando:

```
\linespread{factor}
```

en el preámbulo del documento. Utilice "`\linespread{1.3}`" para un espaciado de "Uno y medio", y "`\linespread{1.6}`" para un espaciado "Doble". El valor por omisión es 1.

4.5.2. Formato de párrafos

En LaTeX existen dos parámetros que influyen en la disposición de un párrafo. Incluyendo una definición como la siguiente:

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

en el preámbulo del documento, se puede cambiar el formato de los párrafos. Estos dos comandos aumentan el espacio entre dos párrafos, mientras que especifican el "*dentado*" (*indent*) de los párrafos a cero. Las partes "*plus*" y "*minus*" de la segunda línea le dicen a TeX, que puede comprimir y expandir el salto entre párrafos por el monto especificado si es necesario para mantener los párrafos en una misma página (siempre que sea posible). Para "*dentar*" un párrafo que no está dentado se puede utilizar el comando:

```
\indent
```

al principio del párrafo. Para crear un párrafo no dentado, se puede utilizar el comando:

```
\noindent
```

como el primer comando del párrafo. Esto puede ser útil cuando se comienza un documento con texto y sin un comando de seccionado.

4.5.3. Espacio horizontal

LaTeX determina los espacios entre palabras y oraciones automáticamente. Para agregar un espacio horizontal, se debe utilizar el comando:

```
\hspace{largo}
```

Si el espacio deseado debe ser mantenido aún si este cae al final o al principio de una línea, se debe utilizar el comando "`\hspace*`" en lugar de "`\hspace`". El parámetro "*largo*", en el caso más simple, es solo un número seguido de una unidad. Las unidades más importantes son las siguientes:

Tabla 4-5. Unidades de medida manejadas en LaTeX

UNIDAD	MEDIDA	EQUIVALENCIA
--------	--------	--------------

mm	milímetro	1/25 pulgadas
cm	centímetro	10mm
in	pulgada	25.4mm
pt	punto	1/72in ~ 1/3mm
em	El ancho del carácter 'M' en la fuente actual	
ex	El ancho del carácter 'X' en la fuente actual	

Un ejemplo de la utilización del comando "`\hspace`" puede ser el siguiente:

Esto `\hspace{1.5cm}` es un espacio de 1.5 cm.

El comando:

```
\stretch{n}
```

genera un espacio "goma" especial. El comando se alarga hasta que todos los espacios faltantes en una línea son llenados. Si dos comandos del estilo "`\hspace{\stretch{n}}`" son implementados en la misma línea, los mismos crecerán de acuerdo al factor del comando "`\stretch`".

4.5.4. Espacio vertical

El espacio entre espacios, secciones, subsecciones, etc. es determinado automáticamente por LaTeX. Si es necesario, se puede agregar espacio vertical adicional entre dos párrafos con el comando:

```
\vspace{largo}
```

Este comando debería ser utilizado normalmente entre dos líneas vacías. Si el espacio debe ser preservado al principio o al final de una página, se debe utilizar el comando "`\vspace*`" en lugar de "`\vspace`". El comando "`\stretch`" en conexión con "`\pagebreak`" puede ser utilizado para escribir texto en la última línea de una página o para centrar un texto verticalmente en la página. Por ejemplo:

```
Algun texto \ldots
```

```
\vspace{\stretch{1}}
```

```
Esto va en la ultima linea de la pagina.\pagebreak
```

Con el comando:

```
\\[largo]
```

se puede especificar espacio adicional entre dos líneas de un mismo párrafo o dentro de una tabla. Con los comandos:

```
\bigskip
```

```
\smallskip
```

se pueden saltar una cantidad predefinida de espacio vertical sin necesidad de preocuparse por números exactos.

4.6. Disposición de la página

LaTeX2e permite especificar el tamaño de la página en el comando "`\documentclass`". Luego de este comando, automáticamente se eligen los márgenes correctos. Pero a veces se puede no estar conforme con los valores por omisión. Naturalmente, estos pueden ser cambiados. Las opciones para modificar los márgenes son las siguientes:

Tabla 4-6. Comandos para establecer los márgenes en LaTeX

COMANDO	DESCRIPCION
<code>\hoffset</code>	<i>Horizontal offset</i>
<code>\voffset</code>	<i>Vertical offset</i>
<code>\evensidemargin</code>	Margen del lado par
<code>\topmargin</code>	Margen superior
<code>\headheight</code>	Alto del cabezal
<code>\headsep</code>	Separación del cabezal
<code>\textheight</code>	Alto del campo para texto
<code>\textwidth</code>	Ancho del campo para texto
<code>\marginparsep</code>	
<code>\marginparwidth</code>	
<code>\footskip</code>	
<code>\marginparpush</code>	
<code>\paperwidth</code>	Ancho del papel
<code>\paperheight</code>	Alto del papel

4.7. Cajas

LaTeX crea sus páginas a través de cajas. Primero, cada letra es una caja pequeña, que luego es "*pegada*" a otras letras para formar las palabras. Estas son nuevamente pegadas a otras palabras, pero con un pegamento especial, que es elástico, así las series de palabras pueden ser estiradas o comprimidas para llenar exactamente una línea en la página.

Esta es una versión muy simplificada de lo que realmente ocurre, pero el punto es que TeX trabaja con pegamento y cajas. No solo una letra puede ser una caja. Se puede introducir virtualmente cualquier cosa en una caja, inclusive otras cajas. Cada caja luego será tratada por LaTeX como si fuese una letra. El ambiente "*tabular*" y el comando "*includegraphics*" por ejemplo, crean ambos una caja. Por lo tanto, se pueden alinear fácilmente tablas o imágenes una al lado de la otra. Sólo se debe tener la precaución de que el ancho combinado de estos objetos no supere el ancho de una línea ("*textwidth*"). Se puede también empaquetar un párrafo con el comando:

```
\parbox[posicion]{ancho}{texto}
```

o con el ambiente:

```
\begin{minipage}[posicion]{ancho}
```

```
...texto...
\end{minipage}
```

El parámetro "*posicion*" puede tomar uno de los siguientes valores:

- "*c*": center.
- "*t*": top.
- "*b*": bottom.

Todos estos valores son relativos a la línea base del texto que rodea la caja. El parámetro "*ancho*" especifica el ancho de la página. La mayor diferencia entre el comando "*\parbox*" y "*\minipage*" es que no se pueden utilizar todos los comandos dentro de "*\parbox*", mientras que casi cualquier cosa puede ser utilizada dentro de "*\minipage*". Mientras que "*\parbox*" empaqueta un párrafo completo realizando saltos de línea y todo lo relacionado a un párrafo, existe también una clase de comandos para cajas que trabajan sólo con material horizontalmente alineado. "*mbox*" es uno de ellos, simplemente empaqueta una serie de cajas dentro de otra, y puede ser utilizado para prohibirle a LaTeX que divida dos palabras. Ya que se pueden poner cajas dentro de otras cajas, estos empaquetadores de cajas horizontales brindan la última flexibilidad.

```
\makebox[ancho][posicion]{texto}
```

El parámetro "*ancho*" define el ancho de la caja resultante (mas allá de su contenido). Además de las expresiones para especificar medidas, se pueden utilizar también los comandos:

```
\width
\height
\depth
\totalheight
```

en el parámetro "*width*". Los valores de estos comandos son obtenidos según las medidas del parámetro "*texto*". El parámetro "*posicion*" puede obtener uno de los siguientes parámetros:

- "*C*": centrado.
- "*L*": alineado a la izquierda.
- "*R*": alineado a la derecha.
- "*S*": exparse el texto dentro de la caja hasta llenarla.

El comando "*\framebox*" trabaja exáctamente de la misma forma que "*\makebox*", con la diferencia que "*\framebox*" dibuja una caja alrededor del texto, y "*\makebox*" no. El comando:

```
\raisebox{elevacion}[ancho][altura]{texto}
```

permite definir propiedades verticales en una caja. Se pueden utilizar los comandos:

```
\width
\height
\depth
\totalheight
```

en sus primeros tres parámetros para trabajar sobre el tamaño de la caja dentro del parámetro "*texto*".

4.8. Reglas y contoneos

El comando:

```
\rule[elevacion]{ancho}{altura}
```

produce una caja negra. Aquí se muestran algunos ejemplos:

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```

Este comando es útil para dibujar líneas verticales y horizontales. Un caso especial es un comando "*rule*" sin ancho, pero con un cierto alto mínimo. Esto se puede utilizar en un ambiente `tabular` para asegurarse que un registro tiene un alto mínimo deseado. Por ejemplo:

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\
\hline
\rule{0pt}{4ex}Strut\
\hline
\end{tabular}
```