

Creación de documentos con L^AT_EX y Kile

Asael Fabian Martínez Martínez
fabian_023@yahoo.com.mx

Febrero del 2006

Copyright 2005 Redes-Área Linux

Asael Fabian Martínez Martínez fabian_023@yahoo.com.mx Se puede copiar, modificar o distribuir este manual bajo las condiciones de la Licencia de Documentación Libre de GNU (GNU FDL). Si se desea hacer una copia total o parcial del documento se deberá adjuntar debidamente la identidad del autor así como la dirección www.acatlan.unam.mx/linux.

El autor no se hace responsable de los daños producidos por la utilización de la información del documento.

Todas las marcas ilustradas son propiedad de sus respectivos dueños.

Índice general

1. Introducción	1
1.1. Objetivos del curso	1
1.2. Requisitos para trabajar con L ^A T _E X	1
1.3. Ayuda para Kile	2
2. Edición de documentos	4
2.1. Mi primer documento	4
2.1.1. Mensajes de L ^A T _E X	6
2.2. Escribiendo párrafos	6
2.2.1. División silábica	7
2.2.2. Párrafos centrados	8
2.2.3. Párrafos alineados a un solo lado	9
2.2.4. Citas textuales	11
2.2.5. Poemas	12
2.3. Instrucciones	12
2.3.1. Comandos	12
2.3.2. Declaraciones	13
2.3.3. Entornos	13
2.4. Caracteres reservados	14
2.4.1. Comentarios en el código fuente	14
2.4.2. Evitar la separación entre palabras	15
2.5. Tipos de letra	15
2.5.1. Familias	16
2.5.2. Perfil	16
2.5.3. Grosor	17
2.5.4. Tamaño	17
2.5.5. Colores	18
2.6. Clases de documentos	19
2.6.1. Paquetes	20
2.7. Seleccionando el idioma	21
2.7.1. Configuración del idioma en Linux	22
2.8. Portada	22
2.8.1. Resúmenes	23
2.9. Proyectos	23

3. Elementos de un documento	26
3.1. Títulos de secciones	26
3.2. Referencias cruzadas	28
3.3. Notas	29
3.3.1. Notas a pie de página	29
3.3.2. Notas al margen	30
3.4. Listas	30
3.4.1. Listas numeradas	31
3.4.2. Listas con viñetas	32
3.4.3. Descripciones	33
3.5. Código	34
3.6. Tablas	35
3.6.1. Consideraciones especiales	38
3.7. Gráficos	39
3.8. Elementos flotantes	40
3.9. Estilos de página	43
Más información	43
4. Matemáticas	45
4.1. Para iniciar	45
4.1.1. Sintaxis en modo matemático	46
4.1.2. Símbolos más frecuentes	50
4.2. Matrices y determinantes	52
4.3. Ecuaciones largas	52
4.4. Teoremas y demostraciones	53
Más información	55
5. Bibliografía	56
5.1. Sistemas de citación bibliográfica	56
5.2. Estilos bibliográficos	57
5.3. Composición de bibliografía y referencias bibliográficas con \LaTeX	58
5.4. Bases de datos bibliográficos con Bib \TeX	61
5.4.1. Uso de la base de datos bibliográficos en \LaTeX	62
Más información	64
6. Índices	66
6.1. Preparación de índices alfabéticos con MakeIndex	66
Más información	69
7. Personalizar tareas	70
7.1. Nuevas instrucciones	70
7.2. Nuevos entornos	71
7.3. Contadores	72
7.4. Longitudes	73
Más información	73

8. Presentaciones	74
8.1. Opciones de la clase prosper	74
8.2. Compilación	74
8.3. Estructura del archivo fuente	75
Más información	77

Índice de cuadros

2.1. Caracteres reservados dentro de \LaTeX	14
3.1. Tabla de verdad	42
4.1. Operadores	48
4.2. Funciones matemáticas	48
4.3. Delimitadores verticales	50
4.4. Acentos en modo matemático	51
4.5. Letras griegas	51
4.6. Relaciones binarias	51
4.7. Operadores binarios	51
4.8. Flechas	51
5.1. Entradas estándar de información	63
7.1. Unidades de medida disponibles	73
8.1. Opciones de la clase prosper	75
8.2. Transiciones entre diapositivas	76

Capítulo 1

Introducción

\LaTeX es un sistema orientado a la composición de textos académicos y científicos, principalmente textos que incluyen muchos elementos de notación matemática, por ejemplo: tesis, apuntes de asignaturas, trabajos de investigación, libros técnicos y científicos, entre otros.

\LaTeX funciona a través de *macros* (una macro es un nombre que define un conjunto de instrucciones que serán sustituidas cuando el nombre de esa macro aparezca en un programa, proceso denominado expansión de macros, en el momento de compilar o ensamblar el programa) que permiten componer e imprimir documentos con una calidad tipográfica excelente. \TeX , antecesor de \LaTeX , fue escrito por Donald E. Knuth, quien dice que \TeX fue pensado para la creación de *libros hermosos*, principalmente para libros que contienen muchas matemáticas. Tiempo después, Leslie Lamport creó \LaTeX , un conjunto de instrucciones (o macros, como se mencionó) que facilitan el uso de \TeX .

Si bien, tanto \TeX como \LaTeX fueron pensados para crear documentos que contienen fórmulas, también es bueno para crear algún otro tipo de documentos.

1.1 Objetivos del curso

Este curso pretende dar una introducción a \LaTeX para crear documentos. En cada sección daremos una breve explicación de las distintas instrucciones de \LaTeX , y se acompañarán con varios ejemplos para que puedan comprenderse mejor.

Además, si deseas ejercitar un poco lo expuesto en cada lección, se agregarán algunas actividades. Podrás descargar un archivo `dvi` y el objetivo es que reproduzcas tal cual ese archivo; también podrás descargar el archivo `tex` donde se encuentra una solución, pero no es la única.

1.2 Requisitos para trabajar con \LaTeX

Para trabajar con \LaTeX utilizaremos lo siguiente:

Compilador de \LaTeX . Todas las distribuciones de Unix/Linux incluyen el programa `tex`, que es un compilador de \TeX , \LaTeX y todas sus extensiones.

Visor de archivos. Necesitarás también algún visor de archivos DVI, PostScript y PDF, que tampoco tienen problema alguno para conseguirlos.

Editor. Por último, existen los llamados «entornos integrados» que facilitan mucho la edición de documentos. Uno de ellos es KILE, y es precisamente el que utilizaremos, la versión 1.8.1. Lo puedes descargar del sitio <http://kile.sourceforge.net>.

1.3 Ayuda para Kile

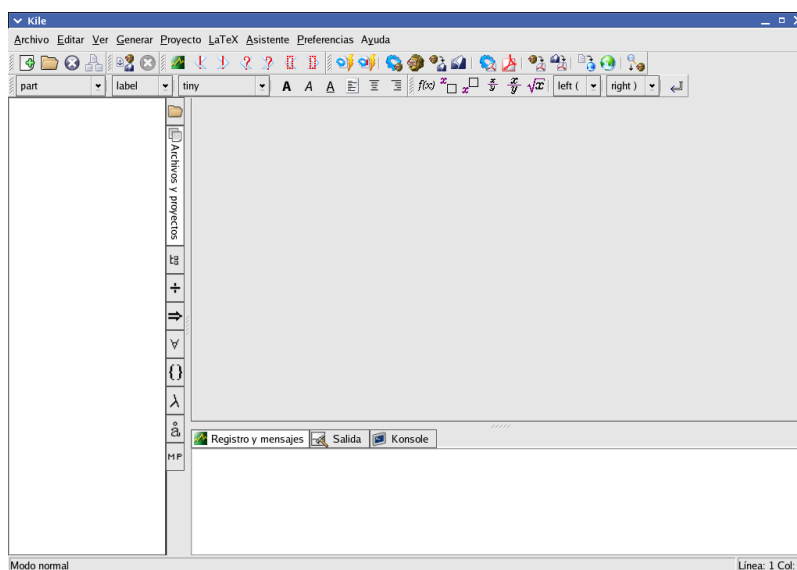
Para terminar con la introducción, contaremos con una ayuda para utilizar KILE; esta ayuda se verá como el siguiente recuadro:



Utilizaremos apartados como éste para explicar cómo obtener los mismos resultados con KILE del tema que estemos tratando. Principalmente explicaremos los botones de las barras de herramientas y sólo cuando no se encuentren allí las instrucciones, utilizaremos los menús; la forma de describir la instrucción por medio del menú será así: *[Menú/Submenú/.../Opción]*, por ejemplo, la opción guardar que se encuentra en el menú archivo, únicamente aparecerá como *[Archivo/Guardar]*.

No pretendemos dar una explicación completa de KILE, sólo daremos lo indispensable para utilizarlo.

La razón de ésto es, como ya mencionamos, KILE ofrece una gran cantidad de botones y menús que nos evitan escribir instrucciones para componer nuestro documento. El aspecto de KILE es el siguiente.



Viéndolo de arriba a abajo y de izquierda a derecha, tenemos lo siguiente:

Menús. Como en cualquier programa de edición de textos tenemos menús, que además de los tradicionales «Archivo», «Edición», «Ver» y «Ayuda», contamos con menús útiles para usar y configurar \LaTeX .

Barras de herramientas. De igual manera que los menús, están los clásicos botones, pero hay algunos que nos serán de utilidad (los veremos más adelante).

Ventana de varios. Esta ventana está a la izquierda de la pantalla y contiene varias pestañas que podrán ser de utilidad (también las veremos más adelante).

Editor. En el editor podemos escribir los documentos.

Mensajes. Abajo del editor está una ventana con tres pestañas:

- Registro y mensajes. Aquí aparecerán los errores, avisos y cajas malas generadas por la compilación del documento.
- Salida. Aquí se muestran todos los mensajes generados por la compilación, como si estuviéramos compilando en una terminal.
- Konsole. Una terminal.

Capítulo 2

Edición de documentos

Como se mencionó en la introducción, tanto $\text{T}_{\text{E}}\text{X}$ como $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ son lenguajes de programación pero orientados a la creación de documentos, por tal razón para utilizar $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ necesitamos conocer las cosas básicas de cualquier compilador.

Durante este primer capítulo conoceremos lo básico para crear nuestros documentos con $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

2.1 Mi primer documento

Para comenzar, vamos a escribir nuestro primer documento en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. El código de este documento será el siguiente:


```
\documentclass{article}
\begin{document}
¡Hola mundo!

Este es mi primer documento hecho con \LaTeX, es pequeño, pero
pronto le podré poner más cosas, por ejemplo, una fórmula como
esta:  $f(x)=\frac{x^2 - x - 2}{x-2}$ 

Mi documento fue escrito y compilado el día \today.
\end{document}
```

Copia todo tal cual está escrito, respetando los espacios, mayúsculas y minúsculas. Puedes utilizar cualquier editor de textos común, pero deberás guardarlo como texto sin formato y con extensión `tex`, por ejemplo, `hola.tex`.



Para crear un documento deberás dar clic en el botón . Aparecerá un diálogo pidiendo que elijas alguna plantilla, puedes elegir «Empty Document» para que escribas todo o elegir «Article»; si eliges la última opción tendrás que borrarle algunas instrucciones.


Para guardar, simplemente das clic en el botón .

Una vez capturado y guardado, compila el archivo. Para hacer esto escribe lo siguiente:

```
latex hola.tex
```



Para compilar, puedes hacer clic en el botón  para compilar con L^AT_EX y crear el archivo dvi.



Otra opción es crear un archivo pdf. Para hacerlo puedes dar clic en el botón ; si estás en la terminal simplemente sustituye latex por pdflatex.

Después de compilarlo tendrás un archivo llamado hola.dvi, si lo abres, aparecerá algo parecido a esto:

```
Hola mundo!
Este es mi primer documento hecho con LATEX, es pequeño, pero pronto le podr poner
mas cosas, por ejemplo, una frmula como esta:  $f(x) = \frac{x^2-x-2}{x-2}$ 
Mi documento fue escrito y compilado el da February 22, 2006.
```

Si observas bien, faltan algunos caracteres: un signo de admiración, la ñe en «pequeño», y todas las letras acentuadas. Hay otras cosas que se ven distintas del código fuente en este archivo dvi, por ejemplo, los espacios entre palabras y algunas instrucciones como \LaTeX, \frac y \today.



Para visualizar los archivos hay que dar clic en el botón  para los archivos dvi o en el botón  para los archivos pdf.

Para mejorar nuestro documento, vamos a agregar unas siguientes instrucciones en el archivo hola.tex entre la línea 1 y la 2. El archivo modificado quedará como sigue:

```
\documentclass{article}
\usepackage[spanish]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
¡Hola mundo!

Este es mi primer documento hecho con \LaTeX, es pequeño, pero
pronto le podré poner más cosas, por ejemplo, una fórmula como
esta:  $f(x)=\frac{x^2 - x - 2}{x-2}$ 

Mi documento fue escrito y compilado el día \today.
\end{document}
```

Si compilamos este último archivo, desaparecerán los errores que mencionamos anteriormente. La explicación de las líneas 2 y 3 se dará más adelante.

Para terminar esta primera sección, es importante mencionar lo siguiente:

- Todos los archivos deberán comenzar con la instrucción \documentclass{article}, la cual especifica el tipo de documento que se quiere componer (article se puede cambiar por otra opción, pero se verá en la sección 2.6).

- El texto del documento debe aparecer entre las instrucciones `\begin{document}` y `\end{document}`, si se introduce algún texto antes de `\begin{document}` la compilación generará un error, y si aparece después de `\end{document}` simplemente se ignorará.
- La parte comprendida entre `\documentclass` y `\begin{document}` se conoce como «preámbulo».
- Todos los «paquetes» y las nuevas instrucciones que se necesiten deberán aparecer en el preámbulo.
- El preámbulo que utilizaremos para este curso será el siguiente:

```
\documentclass{article}
\usepackage[spanish]{babel}
\usepackage[latin1]{inputenc}
```

2.1.1 Mensajes de L^AT_EX

Cuando escribimos un programa, por ejemplo en C++, y queremos generar el ejecutable, tenemos que compilarlo. En la compilación se revisan varias cosas, que las variables estén bien definidas, que las instrucciones se usen correctamente, etc. Después de todo esto, el compilador da una serie de mensajes para saber si hubo errores o hay algunas líneas que tenemos que revisar, los conocidos «errors» y «warnings».







De la misma manera funciona L^AT_EX; cuando compilamos nuestro archivo fuente, revisa que hayamos escrito correctamente las instrucciones que L^AT_EX reconoce y que los entornos estén cerrados correctamente, entre otras cosas. Los mensajes que L^AT_EX muestra son de tres tipos: «errors», «warnings» y «bad box(es)».

Los errores se presentan cuando estamos utilizando incorrectamente alguno de los caracteres reservados o escribimos mal una instrucción.

Los avisos (o «warnings») aparecen cuando, por ejemplo, hacemos una referencia con una etiqueta que no está definida. A lo largo de este manual explicaremos otros posibles «warnings».

Por último, las cajas malas (o «bad box(es)») se generan cuando alguna palabra no se puede dividir correctamente al final del renglón o porque el espaciado entre palabras está fuera de los límites que L^AT_EX tiene.



En la ventana de «Registro y mensajes» aparecerán los errores, avisos y cajas malas. Para ir directamente al lugar donde se originó alguno de ellos existen los botones  y  para ir al error anterior o siguiente, respectivamente; los botones  y  funcionan de manera similar para los avisos; por último, los botones  y  muestran las cajas malas.

2.2 Escribiendo párrafos

Ahora nos corresponde aprender cómo se escribe un párrafo de texto en L^AT_EX.

Para comenzar, utilizando el preámbulo que definimos en la sección anterior, es posible escribir un párrafo de texto tal cual se escribe en cualquier otro procesador de textos normal. Tenemos un ejemplo:

```
Hacer un punto y aparte es muy fácil.
Pero no basta con cambiar de línea.

Hay que dejar una en blanco.

Para iniciar \\ otra no es necesario cambiar de línea.\par
Dejar uno o más espacios es irrelevante. Si se desea
forzar un espacio como el que viene a
continuación \\ es necesario incluir una instrucción.\par
\noindent
Si no se quiere sangrar un párrafo se cuenta con una
instrucción especial.
```

El resultado de la compilación de este fragmento es el siguiente:

Hacer un punto y aparte es muy fácil. Pero no basta con cambiar de línea.
Hay que dejar una en blanco.
Para iniciar
otra no es necesario cambiar de línea.
Dejar uno o más espacios es irrelevante. Si se desea forzar un espacio como el que
viene a continuación es necesario incluir una instrucción.
Si no se quiere sangrar un párrafo se cuenta con una instrucción especial.

Como observas, aparecen algunos caracteres medio raros, los explicaremos en un momento, pero antes es importante saber que \LaTeX compondrá todos los párrafos justificados a los dos márgenes y cada uno será sangrado.

Las instrucciones que ocupamos en el ejemplo anterior, y algunas otras, se explican a continuación:

- Para iniciar un nuevo párrafo se puede utilizar `\par`. Otra forma de iniciar un párrafo es dejando cuando menos una línea en blanco.
- Con `\\` se inicia una nueva línea pero no un nuevo párrafo
- Si no se quiere sangrar algún párrafo se utiliza `\noindent` antes de comenzar el párrafo. Esta instrucción sólo afecta a un solo párrafo.
- Dado que no importa cuantos espacios en blanco haya entre palabras, ya que \LaTeX sólo lo tomará como uno, con la instrucción `_`¹ se puede agregar un espacio extra.

2.2.1 División silábica

Durante la compilación, \LaTeX compone línea por línea del texto buscando optimizar el espacio, si alguna palabra no entra completa en un renglón, \LaTeX la dividirá en sílabas para que los renglones no queden dispares. Sin embargo, no siempre \LaTeX sabe cómo dividir una

¹La marca que aparece después de la diagonal indica que debe dejarse un espacio en blanco

palabra. En este caso pondrá la palabra completa al final del renglón, y además mandará un mensaje indicando que una palabra sale de la «caja» del texto.

Existen dos formas de indicarle a \LaTeX cómo puede dividir una palabra:

1. Utilizando la instrucción

```
\hyphenation{palabra1, palabra2, ...}
```

en el preámbulo del documento. Las sílabas de las palabras se dividen con un guión, además de que no se distinguen mayúsculas de minúsculas. Sin embargo, no se permiten palabras con caracteres acentuados.

2. Otra opción es indicar directamente en el texto la división apropiada. Esto se logra con la instrucción

```
\-
```

Estas instrucciones raramente se utilizarán, ya que el paquete `babel` con la opción `spanish` incluye una larga lista de palabras con sus respectiva división silábica (véase la sección 2.7).

2.2.2 Párrafos centrados

Como mencionamos en la sección anterior, \LaTeX justifica todos los párrafos por ambos márgenes, pero también es posible alinear párrafos ya sea a la derecha, a la izquierda o centrados. Primeramente vamos a ver cómo crear párrafos centrados.

Para centrar algún texto o algún otro elemento, como un gráfico o una tabla, se tiene la siguiente instrucción

```
\begin{center}
texto centrado
\end{center}
```

A este tipo de instrucciones les llamamos, en el lenguaje de \LaTeX , «entornos». Los entornos se explican en la siguiente sección, pero veamos cómo funciona. Todo lo que se encuentre entre las instrucciones `\begin{center}` y `\end{center}` es lo que deseamos centrar.

Cuando se escribe algún texto dentro de este entorno se desactiva la división de palabras, entonces, para indicar el final de un renglón se puede utilizar la instrucción `\\`, de otra forma \LaTeX buscará la mejor forma de dividir cada renglón.

Para comprender mejor esto, veamos el siguiente ejemplo:

```
Como ejemplo de párrafos centrados tenemos el siguiente
fragmento:
\begin{center}
¡Oh, si él me besara con besos de su boca!\\ porque mejores
son tus amores que el vino.\\ A más del olor de tus suaves
ungüentos,\\ tu nombre es como ungüento derramado; \\
por eso las doncellas te aman. \\
Atráeme; en pos de ti correremos. \\
El rey me ha metido en sus cámaras;
\end{center}
es del Cantar de los Cantares.
```

Podemos observar que todo lo anterior es un solo párrafo.

ya compilado se tiene

Como ejemplo de párrafos centrados tenemos el siguiente fragmento:

¡Oh, si él me besara con besos de su boca!
 porque mejores son tus amores que el vino.
 A más del olor de tus suaves ungüentos,
 tu nombre es como ungüento derramado;
 por eso las doncellas te aman.
 Atráeme; en pos de ti correremos.
 El rey me ha metido en sus cámaras;

es del Cantar de los Cantares.

Podemos observar que todo lo anterior es un solo párrafo.

Como se ve, antes y después de utilizar este entorno \LaTeX añade un pequeño espacio. Ahora bien, si sólo queremos centrar una frase tenemos la instrucción

`\centerline{texto}`

Con esta instrucción \LaTeX no añadirá ningún espacio antes y después del *texto*. Es importante mencionar que con esta instrucción no se pueden centrar varios párrafos y la instrucción `\\` tampoco tiene efecto alguno.

Por último, tenemos otra instrucción que sirve para centrar texto o cualquier otro elemento, ésta es:

`\centering`

Este tipo de instrucciones también tiene un nombre especial, se conocen como «declaraciones». También se estudiarán en la siguiente sección.

2.2.3 Párrafos alineados a un solo lado

Otra forma de alinear los párrafos es cuando se necesitan párrafos justificados a un solo margen, ya sea por la derecha o por la izquierda. De manera general se tienen instrucciones parecidas a las que vimos en el apartado anterior. Estas instrucciones se resumen en la siguiente tabla

Alineación	Entorno	Instrucción	Declaración
Centrado	<i>center</i>	<code>\centerline</code>	<code>\centering</code>
Por la izquierda	<i>flushleft</i>	<code>\leftline</code>	<code>\raggedright</code>
Por la derecha	<i>flushright</i>	<code>\rightline</code>	<code>\raggedleft</code>

Veamos ahora un ejemplo de cómo usar estas nuevas instrucciones.

Continuamos con los párrafos alineados a la izquierda:

```
\begin{flushleft}
Morena soy, oh hijas de Jerusalén, pero codiciable\\
como las tiendas de Cedar,\\ como las cortinas de
Salomón.\\ No reparéis en que soy morena,\\
porque el sol me miró.\\ Los hijos de mi madre
se airaron contra mí;\\ me pusieron a guardar
las viñas;\\ y mi viña, que era mía, no guardé.\\
Hazme saber, oh tú a quien ama mi alma,\\
dónde apacientas, dónde sesteas al mediodía;\\
pues ¿por qué había de estar yo como errante
junto a los rebaños de tus compañeros?
\end{flushleft}
```

Si ahora dejamos una línea en blanco, comenzamos un párrafo.

Continuamos con los párrafos alineados a la izquierda:

Morena soy, oh hijas de Jerusalén, pero codiciable
como las tiendas de Cedar,
como las cortinas de Salomón.
No reparéis en que soy morena,
porque el sol me miró.
Los hijos de mi madre se airaron contra mí;
me pusieron a guardar las viñas;
y mi viña, que era mía, no guardé.
Hazme saber, oh tú a quien ama mi alma,
dónde apacientas, dónde sesteas al mediodía;
pues ¿por qué había de estar yo como errante junto a los rebaños de tus compañeros?

Si ahora dejamos una línea en blanco, comenzamos un párrafo.




Por último, un pequeño ejemplo de la alineación a la derecha:


```
\begin{flushright}
Si tú no lo sabes, oh hermosa entre las mujeres,\\
ve, sigue las huellas del rebaño,\\
y apacienta tus cabritas junto a las cabañas
de los pastores.
\end{flushright}
```

Por último, un pequeño ejemplo de la alineación a la derecha:

Si tú no lo sabes, oh hermosa entre las mujeres,
ve, sigue las huellas del rebaño,
y apacienta tus cabritas junto a las cabañas de los pastores.



El botón  crea un entorno *center* y coloca el cursor dentro del mismo. De manera análoga el botón  y  colocan un entorno *flushleft* o *flushright*, respectivamente.

Un botón útil para hacer el cambio de línea es , el cual insertará `\\`.

2.2.4 Citas textuales

Cuando queremos hacer una cita textual de alguna obra, el texto de la cita se compone en un párrafo aparte con sus márgenes más amplios. Dentro de \LaTeX existen dos entornos que nos sirven para hacer citas textuales.

El primer entorno es *quote*, su funcionamiento se muestra en el siguiente ejemplo.

```
Continuando con el Cantar de los Cantares, tenemos otro
fragmento:
\begin{quote}
A yegua de los carros de Faraón te he comparado, amiga
mía.\\ Hermosas son tus mejillas entre los pendientes,
tu cuello entre los collares.\par Zarcillos de oro te
haremos, tachonados de plata.
\end{quote}
Como vemos, los renglones son más pequeños dentro de la cita.
```

Continuando con el Cantar de los Cantares, tenemos otro fragmento:

A yegua de los carros de Faraón te he comparado, amiga mía.
Hermosas son tus mejillas entre los pendientes, tu cuello entre los collares.
Zarcillos de oro te haremos, tachonados de plata.

Como vemos, los renglones son más pequeños dentro de la cita.

El otro entorno que tenemos para hacer citas es *quotation*. La diferencia con *quote* es que éste no sangra los párrafos y *quotation* sí lo hace.

```
Ahora utilizando el entorno quotation, la cita queda así:
\begin{quotation}
Mientras el rey estaba en su reclinatorio,
mi nardo dio su olor.\\
Mi amado es para mí un manojito de mirra,
que reposa entre mis pechos.\\
Racimo de flores de alheña en las viñas de En-gadi
es para mí mi amado.

He aquí que tú eres hermosa, amiga mía;
he aquí eres bella; tus ojos son como palomas.
\end{quotation}
```

Ahora utilizando el entorno `quotation`, la cita queda así:

Mientras el rey estaba en su reclinatorio, mi nardo dio su olor.
 Mi amado es para mí un manojito de mirra, que reposa entre mis pechos.
 Racimo de flores de alheña en las viñas de En-gadi es para mí mi amado.
 He aquí que tú eres hermosa, amiga mía; he aquí eres bella; tus ojos son
 como palomas.

De igual forma que en los entornos anteriores, las instrucciones `\\` y `\par` dividen los renglones y comienzan un nuevo párrafo, respectivamente.

2.2.5 Poemas

Para terminar esta sección veremos un entorno que \LaTeX tiene para componer poemas: el entorno `verse`. Para dividir los versos se utiliza `\\` y una estrofa se divide con `\par`. Si un verso no cabe en un solo renglón, el segundo se sangrará.

```
Para terminar la sección de párrafos, terminamos también
con el primer capítulo del Cantar de los Cantares.
\begin{verse}
He aquí que tú eres hermoso, amado mío, y dulce;
nuestro lecho es de flores.\\
Las vigas de nuestra casa son de cedro,
y de ciprés los artesonados.
\end{verse}
```

Para terminar la sección de párrafos, terminamos también con el primer capítulo del Cantar de los Cantares.

He aquí que tú eres hermoso, amado mío, y dulce; nuestro lecho es de flores.
 Las vigas de nuestra casa son de cedro, y de ciprés los artesonados.



En el menú *[LaTeX/Entorno]* aparecen dos opciones Cita, una para agregar un entorno `quote` y el otro para `quotation`; de igual forma aparece la opción para insertar un entorno `verse`.

2.3 Instrucciones

Como hemos podido ver en las secciones anteriores, \LaTeX tiene una serie de instrucciones para darle formato a nuestro documento. En esta sección describiremos brevemente el tipo de instrucciones que podemos utilizar en \LaTeX .

2.3.1 Comandos

Instrucciones como `\LaTeX`, `\frac` y `\centerline` se conocen como comandos. Todos los comandos tienen la misma estructura:

1. una barra invertida: \
2. un nombre: sólo pueden utilizarse letras para definir una instrucción, además de que \LaTeX distingue entre mayúsculas y minúsculas
3. argumentos opcionales: estos se colocan entre corchetes ([])
4. argumentos obligatorios: que se colocan entre llaves ({ })

Algunos comandos no permiten agregar ningún espacio en blanco como comúnmente lo haríamos, para insertarlo se puede utilizar { } o \ después del comando y por último el espacio. Para entender mejor ésto, tenemos un ejemplo:

```
\chapter[Introducción a \LaTeX]{Una pequeña introducción a
\LaTeX}
\label{sec:Introduccion}
```

Si escribimos \LaTeX y dejamos un espacio, no se insertará, tenemos que escribir $\LaTeX\{$ o $\LaTeX\backslash$ para que se deje el espacio en blanco.

Estamos en el capítulo~\ref{sec:Introduccion} en la página%~\pageref{sec:Introduccion}.

Por último tenemos la misma ecuación que en el primer documento: $f(x)=\frac{x^2}{x-2}$

En este ejemplo vemos la instrucción $\backslash\text{chapter}$ que tiene tanto un argumento opcional como uno obligatorio, además utilizamos $\backslash\text{label}$, $\backslash\text{ref}$ y $\backslash\text{pageref}$ que necesitan de un argumento obligatorio. También usamos \LaTeX que no necesita argumentos y por último utilizamos la instrucción $\backslash\text{frac}$ que requiere de dos argumentos obligatorios. Los detalles de estas instrucciones se darán en las siguientes secciones.

2.3.2 Declaraciones

Las declaraciones son en sí comandos, pero funcionan un poco diferente. Una declaración tiene la siguiente estructura:

```
{\deklaración elementos afectados por la declaración}
```

Las diferencias con un comando son que una declaración afectará a todo lo que se encuentre después se escribirla, y para delimitar su función se utilizan llaves.

Ejemplos de declaraciones son: $\backslash\text{centering}$, $\backslash\text{raggeright}$ y $\backslash\text{raggedleft}$, que se vieron en las secciones 2.2.2 y 2.2.3.

2.3.3 Entornos

Por último, tenemos los «entornos». Un entorno tiene la siguiente estructura:

```
\begin{nombre del entorno}
  elementos afectados por el entorno
\end{nombre del entorno}
```

Los entornos pueden llevar argumentos tanto optativos como obligatorios y siempre se escribirán después de `\begin{nombre del entorno}`.

Se puede tener un entorno dentro de otro, pero siempre hay que tener cuidado de cerrarlos correctamente. En ocasiones las declaraciones pueden ser más útiles, ya que la función de esta declaración será delimitada por el entorno.

Ya hemos hablado de algunos entornos, entre ellos están *center*, *flushleft* y *quote*, además del entorno *document*.



El menú *[LaTeX]* contiene varios submenús para insertar una gran cantidad de comandos y entornos; los más utilizados se encuentran en las barras de herramientas.

En cuanto a los entornos, una ventaja de KILE es que basta con escribir el inicio del entorno y al dar «enter», el programa agrega el final del entorno.

Algunas versiones de KILE muestran una lista completa de las instrucciones al escribir `\`; esta lista es de entornos y comandos de \LaTeX , \TeX y \MetaPost .

2.4 Caracteres reservados

Así como en los lenguajes de programación existen palabras reservadas (palabras que tienen un significado especial para el compilador), \LaTeX tiene 10 «caracteres» reservados, caracteres que no se pueden escribir como cualquier otro. Estos caracteres se muestran en el cuadro 2.1, así como una breve descripción del significado para \LaTeX y cómo se pueden escribir si se necesitan.

Carácter	Uso en \LaTeX	Cómo se escribe
<code>\</code>	Símbolos especiales e instrucciones	<code>\textbackslash</code>
<code>{</code>	Comienzo de un grupo	<code>\{</code>
<code>}</code>	Final de un grupo	<code>\}</code>
<code>%</code>	Comentarios	<code>\%</code>
<code>~</code>	Evita separar dos palabras	<code>\~{ }</code>
<code>&</code>	Separación entre columnas dentro de una tabla	<code>\&</code>
<code>\$</code>	Inicio o fin del «modo matemático»	<code>\\$</code>
<code>^</code>	Exponentes	<code>\^{ }</code>
<code>_</code>	Subíndices	<code>_{ }</code>
<code>#</code>	Parámetros para instrucciones	<code>\#</code>

Cuadro 2.1: Caracteres reservados dentro de \LaTeX

Estos caracteres fueron escogidos debido a que no es muy frecuente que los utilicemos en un escrito.

Los primeros tres caracteres ya hemos explicado cómo \LaTeX los utiliza. A continuación explicamos cómo funcionan `%` y `~`, los demás se verán en las otras secciones.

2.4.1 Comentarios en el código fuente

Es común que dentro del código de algún programa incluyamos algunos comentarios, ya sea para dar a conocer quién hizo el programa, para explicar qué hace cada uno de los módulos

o para dar detalles del funcionamiento de algunas líneas que no estén claras. En \LaTeX podemos hacer lo mismo. Por ejemplo, en C++ un comentario debe estar entre los caracteres `/*` y `*/`, cuando se trata de un bloque de texto de varias líneas, o después de las dos diagonales `//`, si es un comentario de una sola línea.

En \LaTeX , el carácter `%` funciona de manera similar a `//` en C++, es decir, sólo se pueden hacer comentarios de una línea. Todo lo que escribamos después de `%` será ignorado, hasta que comencemos una nueva línea. Veamos un ejemplo.

```
Si estamos escribiendo y utilizamos % lo siguiente será ignorado
debemos escribir \verb+\%+ para que aparezca \%.
```

```
Si yo quiero bajar a la siguiente línea y utilizo%
no se agregará ningún espacio después de él.
% De igual manera si lo coloco entre párrafos
Si no existe una línea completamente en blanco, \LaTeX\
no comenzará el párrafo, sino que seguirá siendo uno solo.
```

<p>Si estamos escribiendo y utilizamos debemos escribir <code>\%</code> para que aparezca <code>%</code>.</p> <p>Si yo quiero bajar a la siguiente línea y utilizono se agregará ningún espacio después de él. Si no existe una línea completamente en blanco, \LaTeX no comenzará el párrafo, sino que seguirá siendo uno solo.</p>
--

2.4.2 Evitar la separación entre palabras

Cuando escribimos nuestro documento en \LaTeX , es difícil saber en dónde se iniciará cada línea del párrafo, ya que \LaTeX lo hace de forma automática. Sin embargo, en ocasiones es necesario que algunas palabras vayan en una misma línea. Es una norma tipográfica no separar términos entre los que exista una dependencia o que son complementarios, por ejemplo: Sr. Director, capítulo 1, A. Einstein, etc.

Para evitar la separación entre palabras, utilizamos el carácter `~` en lugar del espacio en blanco, para los ejemplos del párrafo anterior, el código sería así: `Sr.~Director, cap.~1` y `A.~Einstein`, respectivamente. Asimismo es posible unir varias palabras.

2.5 Tipos de letra

Al igual que en cualquier procesador de textos, \LaTeX permite cambiar el tipo de letra para nuestro documento. Si bien, las tipografías de las que \LaTeX dispone no son tan variadas, sí son suficientes para un documento de buena calidad. En esta sección conoceremos cómo poner el texto, por ejemplo, en **negritas**, *cursiva* o en **VERSALITAS**, además de otras tipografías disponibles.

De manera general, un tipo de letra posee tres características: familia, perfil y grosor. Los tipos que \LaTeX utiliza se llaman Computer Modern Fonts, fueron creados juntamente con \TeX por Donald Knuth. Pero es posible incluir otros tipos, por ejemplo fuentes PostScript o TrueType.

2.5.1 Familias

\LaTeX cuenta con tres familias de tipos de letra: roman, sans serif y monoespaciada. Para utilizar alguna de estas familias se tienen las siguientes instrucciones y declaraciones:

Instrucción	Declaración	Muestra
<code>\textrm</code>	<code>\rmfamily</code>	roman
<code>\textsf</code>	<code>\sffamily</code>	sans serif
<code>\texttt</code>	<code>\ttfamily</code>	monoespaciada

La tipografía predeterminada que \LaTeX utiliza es la roman. Las instrucciones son válidas para frases, los cambios de párrafo ocasionan errores con éstas y por esto es mejor utilizar las declaraciones.

La tipografía predeterminada es la roman, como ésta.

`\textsf{La sans serif no tiene adornos y es más útil para escribir títulos.}`

`\texttt{Por último, si necesitamos poner algún código, la monoespaciada es mejor que las anteriores.}`

La tipografía predeterminada es la roman, como ésta.
 La sans serif no tiene adornos y es más útil para escribir títulos.
 Por último, si necesitamos poner algún código,
 la monoespaciada es mejor que las anteriores.

Cuando utilizamos ya sea `\texttt` o `\ttfamily`, la división de palabras se anula, entonces, si deseamos que se dividan las palabras debemos hacerlo explícitamente con `\-`.

2.5.2 Perfil

Para cada familia de tipos tenemos cuatro perfiles disponibles. Las instrucciones y declaraciones para cada uno son las siguientes:

Instrucción	Declaración	Muestra
<code>\textup</code>	<code>\upshape</code>	redonda
<code>\textit</code>	<code>\itshape</code>	<i>cursiva</i>
<code>\textsl</code>	<code>\slshape</code>	<i>inclinada</i>
<code>\textsc</code>	<code>\scshape</code>	VERSALITA

Funcionan igual que las anteriores.

Texto recto

`{\itshape Texto cursivo}`

`{\slshape Texto inclinado}`

`{\scshape Texto en versalitas}`

Texto recto
Texto cursivo
Texto inclinado
 TEXTO EN VERSALITAS

2.5.3 Grosor

Por último, podemos elegir entre dos grosores para colocar el texto, ya sea normal o negrito. Para ésto tenemos lo siguiente:

Instrucción	Declaración	Muestra
<code>\textmd</code>	<code>\mdseries</code>	normal
<code>\textbf</code>	<code>\bfseries</code>	negrito

Además, con estas instrucciones podemos combinar las familias con perfiles y grosores. Por ejemplo:

```
Comenzamos con un texto normal. \textbf{Este está en negritas,
\itshape y además en cursiva.}

\textsl{\sffamily Ahora es sans serif inclinada.}

\ttfamily Por último, es importante mencionar que no todas las
combinaciones están disponibles, \textbf{como ésta.}
```

Comenzamos con un texto normal. **Este está en negritas, y *además en cursiva.***
Ahora es sans serif inclinada.
 Por último, es importante mencionar que no todas las
 combinaciones están disponibles, como ésta.



En el menú *[LaTeX/Estilos de tipo de letra]* se encuentran las instrucciones para modificar la familia, el perfil y el grosor de la tipografía.

2.5.4 Tamaño

Además de los distintos tipos de letra, es posible seleccionar el tamaño de la letra. Tenemos las siguientes declaraciones:

Declaración	Muestra
<code>\tiny</code>	diminuta
<code>\scriptsize</code>	muy pequeña
<code>\footnotesize</code>	bastante pequeña
<code>\small</code>	pequeña
<code>\normalsize</code>	normal
<code>\large</code>	grande
<code>\Large</code>	mayor
<code>\LARGE</code>	muy grande
<code>\huge</code>	enorme
<code>\Huge</code>	gigante

Existen los respectivos entornos para cada declaración mostrada arriba, por ejemplo el entorno *tiny* compondrá la tipografía dentro de éste con el tamaño más pequeño.



En la barra de herramientas hay una lista desplegable con los distintos tamaños, al seleccionar uno de ellos KILE insertará el entorno correspondiente.

2.5.5 Colores

Por último en esta sección hablaremos de cómo cambiar el color del texto. El texto se muestra en color negro, pero con el paquete `color` disponemos de varios más, además de algunas instrucciones para definir nuevos colores. Los colores que define este paquete son: rojo (rojo), verde (green), azul (blue), cian (cyan), magenta (magenta), amarillo (yellow) blanco (white) y negro (black).

Para cambiar el color tenemos la instrucción

```
\color{color}
```

la cual funciona como una declaración, es decir, cambiará el color del texto que se encuentre después de la instrucción. Tenemos también la instrucción

```
\textcolor{color}{texto}
```

En ambas instrucciones, *color* es el color que deseamos y *texto*, en el caso de `\textcolor`, es el texto que cambiará de color.

Por último, para definir un color tenemos la instrucción:

```
\definecolor{nombre}{modelo}{parámetros}
```

Con ésta, definimos el color *nombre* con el *modelo* seleccionado (gray, rgb o cmyk). Los *parámetros* son números entre 0 y 1, y varían según el modelo.

Modelo	Parámetros
gray	{X}
rgb	{R,G,B}
cmyk	{C,M,Y,K}

Por ejemplo:

```
\color{blue} Escribimos en color azul.
Definimos el color rosa:
\definecolor{rosa}{cmyk}{0,0.5,0.5,0}
\textcolor{rosa}{ahora lo utilizamos,} y volvemos al color
azul. {\color{red} Por último, un poco de rojo.}
```

Escribimos en color azul. Definimos el color rosa: ahora lo utilizamos, y volvemos al color azul. Por último, un poco de rojo.

Los colores se pueden definir en cualquier parte del documento, pero es recomendable definirlos en el preámbulo.

2.6 Clases de documentos

Como vimos en la primera sección, \LaTeX necesita que la primera línea de nuestro archivo sea `\documentclass` seguida de un nombre que va entre llaves, en ese caso `article`.

A `article` se le conoce como «clase de documento». Las clases que podemos utilizar en \LaTeX son las siguientes:

Clase	Ejemplos de uso
<code>article</code>	Artículos de revistas, trabajos de seminarios, informes pequeños y otros
<code>report</code>	Informes mayores como proyectos de fin de carrera, tesis doctorales, guiones, etc.
<code>book</code>	Libros
<code>letter</code>	Cartas
<code>slide</code>	Transparencias

Estas son estándar para todas las distribuciones de \LaTeX . Además existen otras que son elaboradas por escuelas o editoriales.

Una clase nos ayuda a dar el formato a nuestro documento, por ejemplo, define el tamaño de papel, los márgenes, el tamaño de los títulos de capítulos, secciones, etc. y algunas otras cosas específicas para la institución.

Como vimos en la sección 2.3, una instrucción (como `\documentclass`) puede llevar argumentos opcionales, que se colocan entre corchetes. En el caso de `\documentclass`, las opciones nos permiten modificar algunas definiciones para nuestro documento; a continuación mencionamos cuáles son estas opciones.

Tamaño de la letra. \LaTeX maneja tres tamaños para la tipografía del texto normal, estos son: 10, 11 y 12 puntos; y dentro de las opciones lo indicamos con *10pt*, *11pt* y *12pt*, respectivamente.

La opción predeterminada para todas las clases es *10pt*.

Tamaño de papel. Para especificar el tamaño del papel se tienen las siguientes opciones: *letterpaper*, *legalpaper*, *executivepaper*, *a4paper*, *a5paper* y *b5paper*.

La opción predeterminada es *letterpaper* en todas las clases.

Número de columnas. Podemos hacer que un documento se componga ya sea a una o a dos columnas, para ésto se tienen las opciones: *onecolumn* y *twocolumn*, respectivamente.

Todas las clases tienen como opción predeterminada a *onecolumn*.

Impresión. Es posible especificar si nuestro documento será impreso a una o a dos caras, es decir, una página por hoja o dos. Con la opción *oneside* el documento se compondrá a una cara, y con *twoside* se compondrá a dos caras. La diferencia entre estas dos se observa principalmente en los márgenes y los encabezados.

La opción predeterminada para las clases *article* y *report* es a una cara y para *book* a dos caras.

Inicio de capítulos. En algunas publicaciones, los capítulos inician en una hoja nueva, o viendo un libro abierto, en una página del lado derecho, dejando la página anterior en blanco si es necesario; con la opción *openright* se obtiene ésto mismo. Pero si se quiere que los capítulos inicien en cualquier página se tiene la opción *openany*.

Estas opciones son válidas únicamente para las clases *report* y *book*, y su opción predeterminada es *openany* y *openright*, respectivamente.

Portada. Cada clase tiene un formato para componer la portada. La opción *titlepage* pondrá la portada en una página y el texto comenzará en una página nueva, y con la opción *notitlepage* el título aparecerá al inicio de la página y el texto abajo de éste.

La opción predeterminada para la clase *article* es *notitlepage* y para *report* y *book* es *titlepage*.

Borrador. Existe la opción *draft* que, entre otras cosas, ayuda a identificar las líneas donde no se puede hacer una división de palabra con ayuda de una marca como esta: █. También, con esta opción es más rápida la compilación, ya que no incluye los gráficos que hayamos puesto, sino que sólo muestra un cuadro con el nombre del archivo. Al contrario de *draft* («borrador») se tiene la opción *final*, que compondrá un documento con todos los elementos y sin las marcas en los renglones.

Todas las clases tienen como opción predeterminada a *final*.

2.6.1 Paquetes

Un paquete en \LaTeX es un archivo que contiene la definición de instrucciones, declaraciones o entornos. Los paquetes son muy útiles, ya que dan más potencia a \LaTeX .

La manera de incluir un paquete es la siguiente:

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


Funciona de manera similar a `\documentclass`. En el argumento optativo colocamos las opciones que necesitamos del paquete; en ocasiones el paquete no tiene opciones, en este caso no se colocan los corchetes. El nombre del paquete va entre llaves. Por ejemplo, en la primera sección utilizamos el paquete *babel* con la opción *spanish*, y lo escribimos así:

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

Un ejemplo de paquete sin opciones es *mathptmx* que modifica la tipografía normal y matemática predeterminada de \LaTeX por la Times; para incluirlo escribimos:

```
\usepackage{mathptmx}
```



Como mencionamos en la primera sección, el botón  muestra un diálogo para seleccionar una plantilla para nuestro documento.

Existe también el menú *[Asistente/Inicio Rápido]* es útil también para crear nuevos documentos. Es un diálogo en donde podemos elegir la clase del documento con las opciones que queramos, además permite incluir algunos paquetes (los más comunes).

Además el menú *[LaTeX/Preámbulo]* tiene las opciones para incluir la instrucción `\documentclass` y `\usepackage`.

2.7 Seleccionando el idioma

\TeX y \LaTeX fueron pensados, en primera instancia, para escribir textos en inglés. Actualmente, es posible escribir en una gran variedad de idiomas de una forma sencilla con ayuda de algunos paquetes.

Uno de estos paquetes es `inputenc`. Con este se define la página de códigos del teclado. Esto simplifica, por ejemplo la escritura de caracteres acentuados, ya que como vimos en el primer ejemplo de este capítulo, caracteres como: á, ñ, ü, etc. no se imprimen en el documento final si no se incluye este paquete.

Una de las opciones de `inputenc` es *latin1*, que es útil para escribir documentos en castellano. Por ejemplo, si no incluimos `inputenc`, para obtener la palabra límite tendríamos que escribir: `l\'\{i\}mite`; en cambio, incluyendo `inputenc` con la opción *latin1* simplemente escribimos `límite`.

Otro paquete útil es `babel`. Con él \LaTeX colocará las etiquetas en el idioma que escojamos. Por ejemplo, sin incluir `babel`, cada capítulo tendrá la etiqueta «Chapter» seguido del número de capítulo y el nombre del mismo; de igual manera con la tabla de contenidos, las listas de figuras y cuadros y la bibliografía, entre otras. Además de estos cambios, utilizando `babel` se activa la división silábica de palabras de acuerdo al idioma que seleccionamos.

Son varios los idiomas que soporta `babel`. El que nos interesa por el momento es el español, que lo podemos utilizar con la opción *spanish*.

El paquete `babel` proporciona algunas instrucciones para cambiar el idioma de nuestro texto.

La instrucción:

```
\selectlanguage{Idioma}
```

cambiará el idioma del documento al *Idioma* escogido; el *Idioma* debe ser alguno de los que soporta `babel`. Con esta instrucción se verán afectadas las etiquetas de los títulos, la fecha y reglas de división silábica.

Para modificar el idioma de un texto corto se tiene la instrucción

```
\foreignlanguage{Idioma}{Texto}
```

la cual tiene el mismo efecto que `\selectlanguage` pero sólo sobre el *Texto* entre llaves.

2.7.1 Configuración del idioma en Linux

Si creamos un documento con el idioma español, seguramente \LaTeX mandará un aviso como el siguiente:

```
No hyphenation patterns were loaded for the language 'spanish'
I will use the patterns loaded for \language=0 instead.
```

Lo que nos dice es que no se cargaron las reglas de división silábica para el idioma español; entonces, nosotros tendremos que indicar la posible división de cada palabra cuando aparezcan las cajas malas, como vimos en la sección 2.2.1.

La forma de solucionar lo anterior es reconfigurando \LaTeX de la siguiente manera:

1. Buscar el archivo `language.dat`, comúnmente se encuentra en

`/usr/share/texmf/tex/generic/config/`.
2. Buscar la línea que dice `%!spanish` y borrar `%!`. Hacer lo mismo para los idiomas que necesitemos.
3. Guardar los cambios en el archivo.

Después de hacer las modificaciones es necesario «reiniciar» \LaTeX ejecutando lo siguiente:

- `initex /usr/share/texmf/tex/latex/base/latex.ltx, o`
- `pdfinitex /usr/share/texmf/pdftex/latex/config/pdflatex.ini`

La primera opción es necesaria si compilaremos con `latex`, mientras que la segunda la utilizaremos al compilar con `pdflatex`. Las rutas pueden variar de una versión a otra; en este caso se utilizó Fedora 2.

Para verificar que el idioma español (y los demás que hayamos configurado) se cargan correctamente podemos revisar el archivo de extensión `log` que \LaTeX genera automáticamente. Una de las primeras líneas es la siguiente:

```
Babel <v3.7h> and hyphenation patterns for american, french,
german, ngerman, spanish, nohyphenation, loaded.
```

Vemos que uno de los idiomas cargados es el español, la configuración del idioma está completa.

2.8 Portada

Ahora mencionaremos cómo ponerle portada a nuestros documentos. La primera manera de hacerlo es con las siguientes instrucciones:

```
\title{Título}
\author{Autores}
\date{Fecha}
```

Con éstas se define el título del trabajo, el autor o autores del mismo y la fecha en que se escribió, respectivamente. Y para imprimir estos datos en la portada se tiene que escribir la instrucción

`\maketitle`

donde se quiera colocar la portada. Es importante mencionar que `\maketitle` debe escribirse después de definir las tres instrucciones anteriores.

En `\author` cada nombre se separa con la instrucción `\and`. La fecha es opcional; si no se incluye `\date` se colocará en la portada la fecha del sistema cuando fue compilado el documento (ésto es diferente a dejar vacío el argumento, en cuyo caso no se incluirá la fecha).

La segunda forma de crear la portada es con el entorno *titlepage*. Con este entorno se dejará toda una página para colocar la información que queramos que lleve nuestra portada. Aquí las instrucciones `\title`, `\author` y `\date` no tienen efecto alguno, ya que nunca son llamadas al generar la portada, así que tendremos que escribir todos los datos de la portada y diseñarla.

Para obtener la fecha de «hoy» se tiene la instrucción `\today`. Esta fecha se verá afectada por el idioma que hayamos seleccionado con `babel`.



El diálogo *[Asistente/Inicio Rápido]* que mencionamos anteriormente, incluye una pestaña donde podemos escribir nuestros datos para crear un nuevo documento.

Otra opción es utilizar el menú *[LaTeX/Preámbulo]*, el cual contiene las opciones para incluir el título, el autor y generar la portada.

2.8.1 Resúmenes

En los trabajos de investigación es indispensable incluir después de la portada un «resumen» del trabajo, también conocido como «abstract». El propósito del resumen que los lectores del trabajo puedan saber si les será útil o no leer todo el trabajo.

Para incluir el resumen en nuestro documento, tenemos el entorno *abstract*. Todo lo que coloquemos dentro de este entorno será el resumen. Además, \LaTeX le agrega el título de «Resumen» (si incluimos `babel` con la opción `spanish`).



De la misma manera, el menú *[LaTeX/Preámbulo]* tiene la opción para insertar el resumen.

2.9 Proyectos

Para terminar este capítulo hablaremos de los «proyectos». Por ejemplo, si estamos escribiendo un libro, sería poco funcional tener todos los capítulos (con su contenido) en un solo archivo; \LaTeX no tendrá problema para compilar un archivo tan grande (sólo tardará más tiempo), pero para nosotros sí será difícil actualizar el documento o corregirlo, por decir algo.

Entonces, lo que podemos hacer para que la edición de nuestro libro sea más fácil es fragmentarlo, creando un archivo `tex` por cada capítulo y crear otro archivo «principal» el cual incluya a los demás. Los proyectos no son parte de \LaTeX , pero existen algunas instrucciones que nos permiten crearlos fácilmente.

La instrucción:

`\include{archivo}`

sirve, como su nombre lo indica, para incluir el *archivo* dentro de otro. Regresando al ejemplo del libro, tenemos un capítulo que se llama «Prefacio» y lo guardamos en el archivo `prefacio.tex` (el archivo `libro.tex` será nuestro archivo principal); para incluirlo en el libro, únicamente escribimos

```
\include{prefacio}
```

en el archivo `libro.tex`. Después de compilar `libro.tex`, veremos todo el contenido de `prefacio.tex` además de lo que hayamos colocado en el archivo principal.

Es importante mencionar que los archivos que vayan a ser incluidos no deben contener preámbulo (ni `\documentclass` ni `\usepackage`) ni el entorno *document*, de otra manera, al momento de compilar el archivo principal se generarán errores.

Otra instrucción útil para los proyectos es:

```
\includeonly{lista de archivos}
```

Esta instrucción debe aparecer únicamente en el preámbulo. Con esta instrucción únicamente se incluirán los archivos que aparezcan en la *lista de archivos* (los archivos deben separarse por comas y la extensión no debe escribirse) y estén incluidos en el documento. Por ejemplo, suponiendo que ya terminamos el libro y estamos escribiendo los apéndices; nuestro archivo principal se ve más o menos así:

```
\documentclass{book}
% Aquí va toda la lista de paquetes...

\begin{document}
% Los datos de la portada y otras cosas...
\include{Introduccion}
\include{Capitulo1}
% ... Los demás capítulos
\include{Apendices}
\end{document}
```

Si compilamos este archivo se tardará un buen rato, ya que compila todos los archivos incluidos; pero lo único que nos interesa es ver el último capítulo, los demás ya sabemos cómo quedaron. Para agilizar este asunto incluimos la línea `\includeonly{Apendices}` antes de `\begin{document}`. En este último caso fue más rápido compilar, ya que \LaTeX toma únicamente la información del archivo `Apendices.tex`.

Por último, la instrucción

```
\input{archivo}
```

funciona de manera similar a `\include`; las principales diferencias son las siguientes

- No es posible «anidar» instrucciones `\include`, es decir, no se puede incluir un archivo que a su vez incluya a otro. Con `\input` sí es posible.
- Cuando utilizamos `\include`, siempre se iniciará una hoja antes de colocar la información, de igual forma pasa después de terminar de incluir el archivo. Con `\input` no se agrega nada, ni antes ni después.
- No es posible utilizar `\includeonly` e `\input` juntas, como vimos anteriormente con `\include`.

Algo importante de mencionar es que los nombres de los archivos `tex` no deben contener caracteres especiales, ñes, acentos, símbolos, etc.



El menú *[Proyecto]* permite administrar los proyectos. Asimismo en la ventana de varios hay una pestaña que dice «Archivos y proyectos» el cual muestra un esquema de los archivos que integran el proyecto.

Con respecto a la compilación, si tenemos varios archivos abiertos y damos clic para compilar, KILE compilará el archivo activo; pero si estamos trabajando con un proyecto y el archivo activo es parte del proyecto, se compilará el archivo principal (no el archivo activo).

Capítulo 3

Elementos de un documento

Cuando elaboramos algún documento, es común que lleve elementos como notas a pie de página, tablas, gráficos, referencias a otras secciones del documento, etc. Para estos elementos \LaTeX provee de algunas instrucciones y entornos, además, las tareas que no puede realizar \LaTeX directamente, las podemos hacer con ayuda de algunos paquetes.

En este capítulo trataremos precisamente cómo hacer todo lo mencionado.

3.1 Títulos de secciones

Para que un documento sea fácil de leer, debe tener una estructura. Por ejemplo, un libro está dividido en capítulos y éstos a su vez se dividen en secciones. Con \LaTeX podemos fácilmente indicar cual es el título de un capítulo o una sección con una simple instrucción.

Tenemos las siguientes instrucciones para indicar qué tipo de título estamos escribiendo:

```
\part{Título}
\chapter{Título}
\section{Título}
\subsection{Título}
\subsubsection{Título}
\paragraph{Título}
\subparagraph{Título}
```

Están enlistadas de mayor a menor jerarquía. La instrucción `\part` es opcional y `\chapter` sólo está disponible en las clases `report` y `book`, ya que `article` no maneja capítulos.

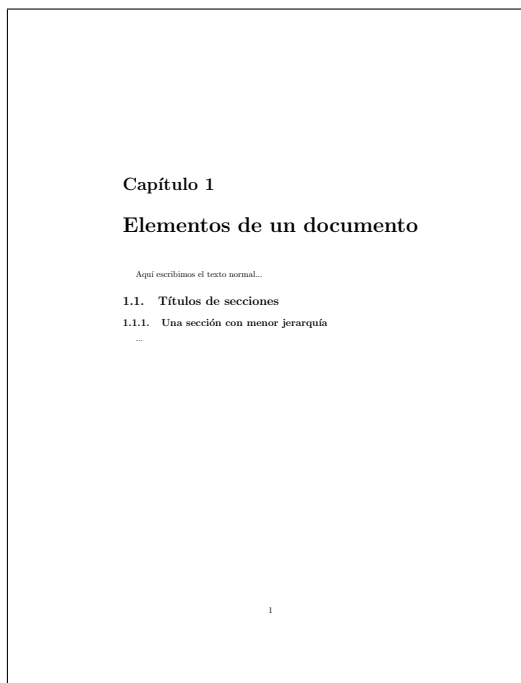
Con estas instrucciones solamente le decimos a \LaTeX cuál es el texto del título y él se encarga de darle formato. Además, cuando utilizamos alguna de las cuatro instrucciones con mayor jerarquía, el texto comienza con un número. Por ejemplo, el siguiente código (válido ya sea en `report` o `book`)

```
\chapter{Elementos de un documento}

Aquí escribimos el texto normal\...

\section{Títulos de secciones}
\subsection{Una sección con menor jerarquía}
\...
```

genera una página similar a esta:



La palabra «Capítulo» que aparece cuando utilizamos `\chapter`, cambia según el idioma hayamos escogido con `babel`.

Los números que aparecen junto al título van «ligados», es decir, cuando se utiliza `\chapter`, el número de capítulo aumenta en uno y todos los números de las instrucciones con jerarquía menor (`\section`, `\subsection`, ...) se reinician a uno; lo mismo pasa cuando se utiliza `\section` con las de menor jerarquía. La instrucción `\part`, sin embargo, no modifica los demás números.

Existe una variante para estas instrucciones se tiene añadiendo un asterisco al final de la instrucción, por ejemplo `\chapter*`. Estas instrucciones no se enumerarán ni aparecerán en la tabla de contenido.



Hay una lista desplegable en la barra de herramientas que inserta las instrucciones anteriores. Únicamente seleccionamos qué instrucción insertaremos y aparecerá un diálogo, escribimos el título en la caja que aparece y listo. Este diálogo permite incluir también las instrucciones con asterisco.

En la ventana de varios hay una pestaña que dice «Estructura», la cual muestra el esquema del documento activo en el editor.

También todas estas instrucciones tienen un argumento optativo. Cuando se compone la tabla de contenidos, \LaTeX tomará el texto entre llaves para colocarlo en la tabla, pero en ocasiones ese título es muy largo o no es tan descriptivo del tema. Entonces, el argumento optativo sirve para colocar ese texto como entrada de la tabla de contenidos y el texto entre llaves aparecerá en la página correspondiente.

Para crear apéndices tenemos la instrucción

`\appendix`

Si utilizamos `report` o `book` la palabra «Capítulo» es cambiada por «Apéndice», además el número de capítulo se cambia a letras mayúsculas y se comienza con la letra A. Si estamos utilizando `article` el cambio sucede con `\section`.

Por último, si queremos que nuestro documento tenga una tabla de contenidos, simplemente utilizamos la instrucción

`\tableofcontents`

en donde queremos que aparezca. Para que se presente correctamente es necesario compilar nuestro documento cuando menos dos veces.



El menú *[LaTeX/Tablas y listas]* contiene la opción para insertar la tabla de contenidos.

3.2 Referencias cruzadas

Cuando elaboramos algún documento, es frecuente que necesitemos hacer referencias a otras partes del mismo documento, por ejemplo, un capítulo, una sección, una ecuación, una figura, un cuadro, etc. La referencia aparece casi siempre como un número o serie de números.

Para hacer una referencia cruzada, \LaTeX tiene tres instrucciones:

```
\label{etiqueta}
\ref{etiqueta}
\pageref{etiqueta}
```

Con `\label` se define una *etiqueta* que servirá para poder hacer la referencia, comúnmente definimos las etiquetas enseguida del elemento al que haremos referencia. Con `\ref` hacemos la referencia a la *etiqueta* entre llaves. Por último, con `\pageref` hacemos también una referencia a la página donde se encuentre la etiqueta. Veamos un ejemplo:

```
\section{Referencias Cruzadas}
\label{sec:Referencias}
```

Acabamos de definir una etiqueta llamada `\verb+sec:Referencias+`. Si queremos hacer una referencia a la sección escribo: véase la sección~`\ref{sec:Referencias}`, con ésto aparece el número de sección que le corresponda.

Definimos ahora otra etiqueta: `\label{enTexto}`. Si ahora quiero una referencia a la página donde está la etiqueta `\verb+enTexto+` escribimos: véase la página~`\pageref{enTexto}`.

1.1 Referencias Cruzadas

Acabamos de definir una etiqueta llamada `sec:Referencias`. Si queremos hacer una referencia a la sección escribo: véase la sección 1.1, con ésto aparece el número de sección que le corresponda.

Definimos ahora otra etiqueta: . Si ahora quiero una referencia a la página donde está la etiqueta `enTexto` escribimos: véase la página 28.

Vemos que en el ejemplo antes de escribir, ya sea, `\ref` o `\pageref` está el carácter `~`, esto se debe a que como mencionamos anteriormente con él se evita la separación entre dos palabras. Véase la sección 2.4.2.

Es importante mencionar que las etiquetas no deben contener ninguno de los caracteres reservados ni letras acentuadas ni espacios en blanco, ya que ocasionan errores en la compilación. Además, si hacemos una referencia a una etiqueta no definida, además de que \LaTeX generará un «warning», en lugar de la referencia aparecerán dos signos de interrogación: `??`.

Cuando incluimos estas instrucciones, la primera compilación generará varios «warning», pero compilando dos o tres veces (según lo indique \LaTeX) se quitarán.



En la barra de herramientas está una lista desplegable que, entre otras cosas, permite insertar las etiquetas y las referencias. En cuanto a las referencias, aparece un diálogo el cual permite seleccionar únicamente las definidas en el documento, evitando así que se inserten etiquetas no válidas.

3.3 Notas

3.3.1 Notas a pie de página

Para colocar una nota a pie de página, \LaTeX tiene la instrucción

```
\footnote[Número]{Texto}
```

Con ésta, el *Texto* será la nota a pie de página y debe ponerse en donde se desea hacer esta nota. Todas las notas se enumerarán automáticamente.

El argumento optativo *Número* sirve para especificar el número que deseamos para la nota, pero no cambiará los números de las notas que sigan.

Veamos un ejemplo:

```
Es fácil hacer notas a pie de página\footnote{Como esta.}.
Todas las notas serán numeradas automáticamente, pero podemos
indicar otro número\footnote[4]{Esta es otra nota}, aunque
ésto no modificará la secuencia\footnote{La tercer nota}.
```

Es fácil hacer notas a pie de página¹. Todas las notas serán numeradas automáticamente, pero podemos indicar otro número⁴, aunque ésto no modificará la secuencia².

¹Como esta.

⁴Esta es otra nota

²La tercer nota



La misma lista desplegable que inserta las etiquetas y referencias, permite insertar las notas a pie de página.

Para casos especiales podemos utilizar las siguientes instrucciones

```
\footnotemark[Número]
\footnotetext[Número]{Texto}
```

Número y *Texto* funcionan igual que `\footnote`.

Para entender mejor como funcionan, veamos el siguiente ejemplo:

```
Aquí escribimos una nota\footnote[2]{Una nota con el número 2.}
\par...Más adelante hacemos referencia a la nota
anterior\footnotemark[2].
```

```
Ahora utilizamos la otra instrucción\footnotetext{Otra nota.},
como no hemos utilizado \verb+\footnote+ el contador está en cero.
```

Aquí escribimos una nota²
 ...Más adelante hacemos referencia a la nota anterior².
 Ahora utilizamos la otra instrucción, como no hemos utilizado `\footnote` el contador
 está en cero.

²Una nota con el número 2.
⁰Otra nota.

3.3.2 Notas al margen

Para colocar notas al margen tenemos la instrucción

```
\marginpar[TextoIzquierda]{TextoDerecha}
```

En este caso, el texto de la nota aparecerá al nivel del renglón en donde se haya incluido la nota.

Por ejemplo:

```
Una nota al margen\marginpar{Como esta} es fácil de colocar.
Cada clase de documento reserva un espacio para colocarlas.
```

Una nota al margen es fácil de colocar. Cada clase de docu- Como esta
 mento reserva un espacio para colocarlas.

3.4 Listas

Las listas son útiles para presentar información que deseamos sea localizada fácilmente, ya que su formato es distinto al texto normal. Con \LaTeX podemos componer tres tipos de listas: con viñetas, numeradas y descripciones.

3.4.1 Listas numeradas

Para componer listas numeradas tenemos el entorno *enumerate*. Dentro del entorno, cada ítem debe ir precedido por la instrucción `\item`. Por ejemplo

```
De manera general, el desarrollo de cualquier sistema computacional
se divide en las siguientes etapas:
\begin{enumerate}
  \item Análisis y definición de requerimientos
  \item Diseño
  \begin{enumerate}
    \item del sistema
    \item de programas
  \end{enumerate}
  \item Implementación de programas
  \item Pruebas
  \begin{enumerate}
    \item unitaria
    \item de integración
    \item del sistema
  \end{enumerate}
  \item Entrega del sistema
  \item Mantenimiento
\end{enumerate}
```

De manera general, el desarrollo de cualquier sistema computacional se divide en las siguientes etapas:

1. Análisis y definición de requerimientos
2. Diseño
 - a) del sistema
 - b) de programas
3. Implementación de programas
4. Pruebas
 - a) unitaria
 - b) de integración
 - c) del sistema
5. Entrega del sistema
6. Mantenimiento

Como se ve, es posible incluir un entorno *enumerate* dentro de otro (sólo es posible tener hasta 4 entornos «anidados») y la numeración cambiará automáticamente, además de que la sangría aumenta (las sangrías en el código no afectan para nada la composición de la lista, pero es más claro para nosotros ver dónde comienza un entorno y dónde termina). Estos números varían en su presentación dependiendo del idioma seleccionado con `babel`.

La instrucción `\item` tiene un argumento optativo, el cual sustituye al número que corresponda.

```

\begin{enumerate}
  \item Análisis y definición de requerimientos
  \item[*] Diseño
  \item Implementación de programas
  \item[] Pruebas
\end{enumerate}

```

- | |
|---|
| <ol style="list-style-type: none"> 1. Análisis y definición de requerimientos <ul style="list-style-type: none"> * Diseño 2. Implementación de programas <ul style="list-style-type: none"> Pruebas |
|---|

3.4.2 Listas con viñetas

Para crear listas con viñetas tenemos el entorno *itemize*. Este entorno funciona igual que *enumerate*, sólo que cambia el número por una viñeta.

De manera general, el desarrollo de cualquier sistema computacional se divide en las siguientes etapas:

```

\begin{itemize}
  \item Análisis y definición de requerimientos
  \item Diseño
  \begin{itemize}
    \item del sistema
    \item de programas
  \end{itemize}
  \item Implementación de programas
  \item Pruebas
  \begin{itemize}
    \item unitaria
    \item de integración
    \item del sistema
  \end{itemize}
  \item Entrega del sistema
  \item Mantenimiento
\end{itemize}

```

De manera general, el desarrollo de cualquier sistema computacional se divide en las siguientes etapas:

- Análisis y definición de requerimientos
- Diseño
 - del sistema
 - de programas
- Implementación de programas
- Pruebas
 - unitaria
 - de integración
 - del sistema
- Entrega del sistema
- Mantenimiento

La instrucción `\item` también funciona igual que el entorno anterior.

3.4.3 Descripciones

Por último, tenemos otro tipo de listas, llamadas descripciones. Las descripciones son útiles, por ejemplo, para dar definiciones.

Para crear descripciones tenemos en entorno *description*, el cual funciona igual que los dos anteriores, salvo que el argumento optativo de `\item` se vuelve más obligatorio. Veamos un ejemplo:

Las distintas etapas dentro de las pruebas son:

```
\begin{description}
  \item[Pruebas unitarias.] Se inicia probando cada módulo ...
  \item[Prueba de integración.] Una vez que todos los módulos
funcionan correctamente ...
  \item[Prueba de función.] Aquí se evalúa el sistema ...
  \item[Prueba de rendimiento.] En esta etapa se revisa ...
  \item[Prueba de aceptación.] Cuando el sistema ha pasado las
pruebas anteriores ...
  \item[Prueba de instalación.] Por último, el sistema se
instala ...
\end{description}
```

Las distintas etapas dentro de las pruebas son:

Pruebas unitarias. Se inicia probando cada módulo, verificando que realice lo especificado en el diseño.

Prueba de integración. Una vez que todos los módulos funcionan correctamente, se integran y se revisa que funcionen según el diseño.

Prueba de función. Aquí se evalúa el sistema comparándolo con los requerimientos, para ver si realiza lo que el cliente solicitó.


Prueba de rendimiento. En esta etapa se revisa que tan confiables son los resultados generados por el sistema, además del tiempo de respuesta. Si existen algunas restricciones de hardware o software, el sistema se prueba con dichas restricciones.

Prueba de aceptación. Cuando el sistema ha pasado las pruebas anteriores, éste se muestra al cliente para ver si cumple con sus expectativas.

Prueba de instalación. Por último, el sistema se instala en el lugar donde será utilizado y se revisa que todo funcione correctamente.

Si no utilizamos los corchetes, la primera letra o instrucción que se encuentre después de `\item` se pondrá en negritas.



El menú *[LaTeX/Entorno de lista]* tiene las opciones para incluir cualquiera de estos tres entornos; además el botón  inserta automáticamente la instrucción `\item` si estamos dentro de estos entornos.

3.5 Código

Cuando estamos escribiendo algún documento que contiene el código de algún programa, es recomendable utilizar una tipografía monoespaciada, ya que se puede seguir mejor el código.

En \LaTeX tenemos unas instrucciones que nos permiten hacerlo, una es útil para componer bloques de texto y la otra cuando sólo son algunas palabras.

La primera forma se obtiene con el entorno *verbatim*. Todo lo que \LaTeX encuentre dentro de este entorno no le dará ningún formato, es decir, todos los espacios en blanco que haya se colocarán, al igual que las líneas en blanco y los caracteres reservados no tienen efecto alguno, tampoco acomodará las líneas al mismo ancho. Veamos el siguiente ejemplo.

```
\begin{verbatim}
Si estamos en el entorno verbatim, \LaTeX\ no modifica nada,
así    todos los
      espacios se verán.
```

```
Así como las líneas en blanco. Tampoco los caracteres reservados
afectan en nada: \, {, }, &, %, #, ^, _ y ~; y por lo tanto
tampoco las instrucciones: \end{document}
\end{verbatim}
```



```
Si estamos en el entorno verbatim, \LaTeX\ no modifica nada,
así    todos los
      espacios se verán.
```

```
Así como las líneas en blanco. Tampoco los caracteres reservados
afectan en nada: \, {, }, &, %, #, ^, _ y ~; y por lo tanto
tampoco las instrucciones: \end{document}
```

La segunda forma de utilizarlo es con la instrucción

`\verb?Texto?`

Esta tiene el mismo efecto que *verbatim*, sólo que se utiliza dentro de un párrafo. Los signos de interrogación (?) funcionan como delimitadores de la instrucción y pueden cambiarse por cualquier otro carácter.

Tanto el entorno *verbatim* como la instrucción `\verb` tienen una versión con asterisco, la cual mostrará los espacios en blanco con este carácter: `_`. Por ejemplo:

```
Si utilizamos \verb+\verb+ debemos utilizar el mismo carácter
al principio y al final, sino, podemos tener resultados no deseados:
\verb{Iniciamos aquí}, pero necesitamos cerrar con {.
```

```
Ahora la versión con asterisco: \verb*+dejamos    varios    espacios+.
Lo mismo pasa con \verb+verbatim*+.
```

```
Si utilizamos \verb debemos utilizar el mismo carácter al prin-
cipio y al final, sino, podemos tener resultados no deseados:
Iniciamos aquí}, pero necesitamos cerrar con .
Ahora la versión con asterisco: dejamos\_\_\_varios\_\_espacios. Lo mismo pasa con
verbatim*.
```



En el menú *[LaTeX/Entorno de Código]* se pueden insertar cualquiera de las instrucciones y entornos vistos en esta sección.

3.6 Tablas

Otro elemento que es común que necesitemos en nuestro documento son las tablas. Con \LaTeX podemos crear las tablas de la siguiente manera:

```
\begin{tabular}{FormatoColumnas}
  Fila1-Columna1 & Fila1-Columna2 & \cdots & Fila1-ColumnaN & \\
  Fila2-Columna1 & Fila2-Columna2 & \cdots & Fila2-ColumnaN & \\
  \cdots & \cdots & \cdots & \cdots & \\
  FilaM-Columna1 & FilaM-Columna2 & \cdots & FilaM-ColumnaN & \\
\end{tabular}
```

Lo primero que tenemos que hacer es indicar en *FormatoColumnas* cuántas columnas va a tener nuestra tabla, esto se hace al mismo tiempo que se indica la alineación de cada columna.

La alineación se indica con alguna de las siguientes letras:

l	Izquierda
r	Derecha
c	Centrada

y según la cantidad de letras será la cantidad de columnas. Por ejemplo:

El código de la tabla anterior es:

```
\begin{tabular}{cl}\hline
\texttt{l} & Izquierda \\
\texttt{r} & Derecha \\
\texttt{c} & Centrada \\
\end{tabular}
```

Indicamos que la primer columna estará centrada y la segunda alineada a la izquierda.

El código de la tabla anterior es:

l	Izquierda
r	Derecha
c	Centrada

Indicamos que la primer columna estará centrada y la segunda alineada a la izquierda.

Con este ejemplo podemos ver algunas otras cosas de interés. Para indicar que cambiamos de columna utilizamos el carácter reservado & y el fin de renglón con la instrucción `\\`; se puede omitir `\\` en la última línea, excepto si queremos utilizar algunas otras instrucciones para darle formato a la tabla (como en el ejemplo).

Cada columna tendrá el ancho de la celda más larga en esa columna. Para indicar un ancho específico tenemos la instrucción:

`p{Ancho}`

que se utiliza igual que las letras anteriores. El *Ancho* es una medida, por ejemplo, si queremos que la columna sea de 3 cm de ancho, escribimos: `p{3cm}`. Ejemplo:

```
\begin{center}
\begin{tabular}{p{35mm}r}\hline
\multicolumn{2}{c}{Tercer semestre}\hline\hline
Asignatura & Créditos \\
Introducción a la Inteligencia Artificial & 4.5 \\
Ingeniería de Software & 6 \\
Sistemas de Transporte de Datos & 4.5 \\
\end{tabular}
\end{center}
```

Tercer semestre	
Asignatura	Créditos
Introducción a la Inteligencia Artificial	4.5
Ingeniería de Software	6
Sistemas de Transporte de Datos	4.5

En el ejemplo anterior la primera columna tiene un ancho de 35 mm. Además, con la instrucción `p` la celda se justifica a ambos márgenes.

Otra instrucción útil en una tabla es la siguiente:

`\multicolumn{n}{justificado}{texto}`

Con ella podemos unir varias celdas en el mismo renglón, es útil, como vemos en el ejemplo anterior, cuando queremos colocar títulos a varias columnas o para darle un formato distinto a una celda del especificado al inicio de la tabla.

Si nuestra tabla tendrá muchas columnas con el mismo formato, podemos utilizar una notación abreviada, ésta es:

`*{Cantidad}{Formato}`

Por ejemplo, para hacer una tabla con 12 renglones donde el formato se repite: una columna centrada, la siguiente a la derecha y después una línea vertical dividiendo las columnas, podemos hacerlo, en la parte del *FormatoColumns*, de dos formas:

`| cr | cr | cr | cr | cr | cr |`

o

`|*{6}{cr|}`

Para colocar una línea vertical que divida las columnas tenemos el carácter `|` y se coloca, como hemos visto en los ejemplos, en *FormatoColumns* entre las letras donde deseamos la línea.

Si queremos una línea que divida los renglones tenemos la instrucción

`\hline`

esta instrucción se debe colocar al final del renglón donde queremos la línea (después de `\\`); esta línea se colocará abajo del renglón correspondiente. Pero si queremos una línea horizontal en ciertas columnas, tenemos la instrucción

`\cline{x-y}`

que colocará una línea horizontal que comienza en la columna x y termina en la columna y . Veamos un ejemplo:

```

\begin{center}
\begin{tabular}{l*{4}{c}}
\multicolumn{5}{c}{\textbf{Ayuda oficial al desarrollo}}\\
\multicolumn{2}{c}{2000}&\multicolumn{2}{c}{2001}\\
&\$ & \% \text{ PNB}&\$ & \% \text{ PNB}
\end{tabular}
\end{center}

```

Ayuda oficial al desarrollo				
	2000		2001	
	Mill. \$	% PNB	Mill. \$	% PNB
Estados Unidos	9955	0.10	11429	0.11
Alemania	5030	0.27	4990	0.27
España	1195	0.22	1737	0.30
Noruega	1264	0.80	1346	0.83

3.6.1 Consideraciones especiales

Cuando utilizamos el entorno *tabular* debemos tener en cuenta lo siguiente:

- Toda la tabla será tratada como un solo objeto, es decir, no se puede dividir entre páginas.
- Cuando iniciamos una tabla no se inicia un nuevo párrafo ni se centra, tenemos que indicarlo nosotros. Por ejemplo:

El primer ejemplo de esta sección es:

```

\begin{tabular}{cl}\hline
\texttt{l} & Izquierda \\
\texttt{r} & Derecha \\
\texttt{c} & Centrada \\
\end{tabular}

```

como no dejamos una línea en blanco antes de iniciar la tabla, se considera parte del renglón anterior.

El primer ejemplo de esta sección es:	l	Izquierda
	r	Derecha
	c	Centrada
como no dejamos una línea en blanco antes de iniciar la tabla, se considera parte del renglón anterior.		

Para centrar la tabla podemos utilizar en entorno *center*. Esto sí colocará la tabla en un párrafo aparte.

- Si nuestra tabla tiene muchas columnas, \LaTeX mandará un mensaje de «caja mala».
- Podemos colocar una tabla dentro de otra.



En el menú [*LaTeX/Entorno de tabulación*] contiene las opciones vistas en esta sección tanto para insertar un entorno *tabular*, combinar celdas y colocar líneas.

El menú [*Asistente/Tabla*] abre un diálogo que permite crear tablas de manera visual.

3.7 Gráficos

Si queremos incluir un gráfico en nuestro documento, podemos hacerlo de dos formas:

1. «Importando» un archivo gráfico a \LaTeX .
2. Creando el gráfico con ayuda de paquetes e instrucciones de \LaTeX .

Si importamos un archivo, no podremos modificar su contenido, sólo algunas cosas que veremos más adelante. Es útil importar los archivos cuando se tratan de fotografías o gráficos complicados.

La segunda opción se logra con ayuda de algunos paquetes. Con esta opción podemos tener el control sobre todo el gráfico, podemos indicar los tipos de línea, la tipografía, etc., ya que es código que \LaTeX maneja.

Sin embargo, en esta sección sólo veremos cómo importar o incluir un archivo gráfico.

Para incluir un archivo, antes que nada, debemos utilizar el paquete `graphicx`. Con este paquete podemos incluir archivos con ayuda de la instrucción

```
\includegraphics[Opciones]{Archivo}
```

El *Archivo* debe incluir la ruta del archivo con su extensión, las subcarpetas se dividen con una diagonal /. Si el archivo está en la misma carpeta del archivo `tex`, no será necesario incluir la ruta, sólo el nombre del archivo. Las extensiones de los archivos que \LaTeX puede importar son:

- `pdf`, `png`, `jpg`, `jpeg`, `tif` y `tiff` cuando se compila con `pdflatex`
- `ps` y `eps` cuando se compila con `latex`

Dentro de las *Opciones* se pueden escribir las siguientes opciones (separadas con comas):

<code>width</code>	el gráfico se escala al ancho indicado
<code>height</code>	el gráfico se escala al alto indicado
<code>angle</code>	se rota el gráfico en el sentido de las manecillas del reloj
<code>scale</code>	el gráfico es escalado proporcionalmente

Se debe tener cuidado con el orden de las instrucciones, ya que se leerán de izquierda a derecha.

Al igual que las tablas, cuando incluimos un gráfico será tratado como un solo objeto, así que tendremos que indicar si queremos centrarlo o componerlo en un párrafo aparte.

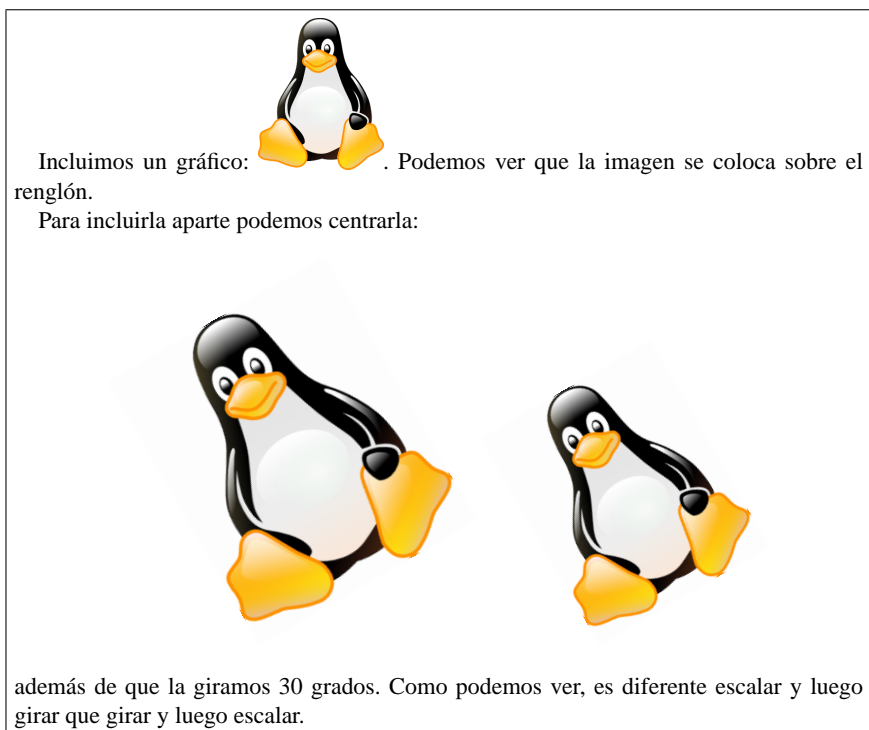
Veamos un ejemplo:

Incluimos una imagen: `\includegraphics[height=2cm]{images/tux.png}`. Podemos ver que la imagen se coloca sobre el renglón.

Para incluirla aparte podemos centrarla:

```
\begin{center}
\includegraphics[height=4cm, angle=30]{images/tux.png}
\includegraphics[angle=30, height=4cm]{images/tux.png}
\end{center}
```

además de que la giramos 30~grados. Como podemos ver, es diferente escalar y luego girar que girar y luego escalar.



En el menú *[LaTeX/Inserción de imagen]* abre un diálogo para incluir archivos gráficos. Este diálogo contiene varias opciones que evitan escribir todo el código.

3.8 Elementos flotantes

Como mencionamos en la sección de tablas y gráficos, estos elementos no pueden dividirse entre páginas, y dado que no sabemos con exactitud dónde quedarán, sería tardado buscar cómo colocarlos sin que nuestro documento se vea mal.

Para facilitar la composición de estos elementos, \LaTeX tiene unos entornos que se conocen como «flotantes». Estos elementos, como su nombre lo indica, flotarán en el texto hasta que \LaTeX encuentre un lugar donde puedan colocarse sin que aparezcan cortados.

Los elementos flotantes que \LaTeX tiene son: *figure* para gráficos o figuras y *table* para cuadros o tablas. Podemos incluir cualquier elemento dentro de estos elementos: texto, fórmulas o algunos otros entornos que hemos visto; pero lo más común es incluir un entorno *tabular* para los cuadros y utilizar la instrucción `\includegraphics` para las figuras.

Cuando componemos nuestro documento a dos columnas, los dos entornos anteriores se ajustarán al ancho de la columna. Sin embargo, con la versión con asterisco de cada entorno (*figure** y *table**) se ajustará al ancho de las dos columnas.

De manera general ambos entornos tienen la siguiente estructura:

```
\begin{Entorno}[Colocación]
Objetos dentro del entorno
\end{Entorno}
```

donde *Entorno* puede ser *figure* o *table*. El argumento opcional *Colocación* sirve para que nosotros le indiquemos a \LaTeX en dónde coloque este entorno, para hacerlo tenemos las siguientes letras:

Argumento	Lugar donde se colocará
h	Aquí, en donde fue definido.
t	Al inicio de la página.
b	Al final de la página .
p	En una página especial exclusiva para elementos flotantes.

éstas serán tomadas de izquierda a derecha, hasta que se pueda colocar en el documento. Las opciones predeterminadas son `tbp`.

Por ejemplo, el siguiente código crea la tabla 3.1.

```
\begin{table}
\centering
\begin{tabular}{*3{c|}c}
 $p \rightarrow q \vee r$  & ( $p \rightarrow q \vee r$ ) & \hline
V&V&V\\
V&V&F\\
V&F&V\\
V&F&F\\
F&V&V\\
F&V&F\\
F&F&V\\
F&F&F
\end{tabular}
\caption{Tabla de verdad}
\label{tab:TablaDeVerdad}
\end{table}
```

Si las opciones que nosotros escogimos no ocasionan que el entorno sea colocado, \LaTeX mandará un «warning» indicando que cambió las opciones para colocar el entorno.

También existe el carácter `!`; con él le indicamos a \LaTeX que no considere otra opción para colocar nuestro entorno y debe colocarse al principio de nuestras opciones. Sin embargo, si el entorno no puede ser colocado como indicamos, \LaTeX hará el cambio.

p	q	r	$(p \rightarrow (q \vee r))$
V	V	V	V
V	V	F	V
V	F	V	V
V	F	F	F
F	V	V	V
F	V	F	V
F	F	V	V
F	F	F	V

Cuadro 3.1: Tabla de verdad

Cuando se está compilando nuestro archivo, \LaTeX guarda en una «cola» todos los elementos flotantes y los va sacando conforme son colocados. Si queremos que se coloquen todos los elementos que tenemos guardados, tenemos las siguientes instrucciones:

```
\clearpage
\cleardoublepage
```

Con estas, se utilizará una o dos páginas, respectivamente, para colocar los elementos flotantes en la cola. Además de estas instrucciones, cuando utilizamos `\chapter`, `\chapter*` y `\end{document}` se colocan los elementos flotantes en la cola.

Dado que estos entornos no siempre aparecerán en donde fueron definidos en el código fuente, es común que aparezca una leyenda en el cuadro o figura la cual va enumerada. Esta leyenda puede colocarse con la instrucción

```
\caption{Leyenda del entorno}
```

Además, podemos utilizar la instrucción `\label` para poder hacer referencia al entorno. Por ejemplo:

Podemos hacer una referencia a la tabla de verdad del ejemplo anterior, para ésto utilizamos la instrucción `\verb+\ref+` así:
La tabla~\ref{tab:TablaDeVerdad}.

Podemos hacer una referencia a la tabla de verdad del ejemplo anterior, para ésto utilizamos la instrucción `\ref` así: La tabla 3.1.

Como vemos, la referencia sólo colocará el número y nosotros debemos colocar la palabra figura o tabla, según corresponda.

Por último, una vez que definimos todas nuestras figuras y cuadros, podemos incluir una «Lista de figuras» o «Lista de cuadros» con las instrucciones:

```
\listoffigures
\listoftables
```

respectivamente.

La instrucción `\caption` tiene un argumento opcional que funciona igual que en las instrucciones mencionadas en la sección 3.1.



El menú *[LaTeX/Entorno variable]* contiene las opciones para insertar ya sea un entorno *figure* o *table*. De igual manera el menú *[Asistente/Flotadores]* abre un diálogo que permite insertar alguno de estos dos entornos, pero incluye más opciones que evitan escribir todo el código.

Para insertar las instrucciones que generan las listas de figuras o cuadros, podemos irnos al menú *[LaTeX/Tablas y listas]* y seleccionar la lista que queramos.

3.9 Estilos de página

Para terminar este capítulo vamos a hablar de cómo colocar las cabeceras y el pie de la página.

En una cabecera se coloca la información del capítulo o sección que se está tratando y en el pie (que es distinto a los pies de página generados por las notas) se coloca en ocasiones el número de página.

En \LaTeX tenemos de tres estilos básicos, éstos son:

<code>plain</code>	Imprime los números de página en el centro del pie. Es la opción predeterminada.
<code>headings</code>	En la cabecera se imprime el capítulo y el número de página, y el pie queda vacío.
<code>empty</code>	Tanto la cabecera como el pie quedan vacíos.

Para elegir alguno, utilizamos la instrucción:

```
\pagestyle{Estilo}, o
\thispagestyle{Estilo}
```

donde el *Estilo* es uno de los anteriores. Utilizando `\pagestyle` todas las páginas utilizarán el *Estilo* seleccionado, comúnmente se coloca al inicio de todo el documento. Con `\thispagestyle` se cambia únicamente la página en donde se coloque esta instrucción.

Más información

- El paquete `minitoc` nos permite crear varias tablas de contenido.
- Podemos crear referencias cruzadas semejantes a los «links» de una página web. El paquete `hyperref` sirve para ésto.
- Con el paquete `endnotes` nos ayuda si queremos colocar notas a pie de página pero que se coloquen al final del documento.
- Si queremos crear nuestras propias listas existe el entorno `list`.
- El paquete `fancyvrb` proporciona varias opciones para componer la información de un entorno *verbatim*.

- Cuando necesitemos hacer tablas que ocupan más de una página, el paquete `longtable` es el adecuado.
- Para crear gráficos directamente en \LaTeX se puede utilizar el entorno `picture`. También existen varios paquetes adicionales, conocidos como PSTricks.
- Si queremos crear nuevos elementos flotantes podemos hacerlo con ayuda del paquete `float`.
- Por último, para crear nuevos estilos de página existe el paquete `fancyhdr`.

Capítulo 4

Matemáticas

Con lo que hemos visto hasta ahora, ya podemos crear un documento completo (excepto por la bibliografía, que veremos en el siguiente capítulo). Pero si nuestro documento necesita fórmulas, necesitamos conocer cómo componerlas. Como mencionamos al principio, \TeX (y por consiguiente \LaTeX) fue diseñado para documentos que contienen muchas matemáticas.

En este capítulo veremos cómo introducir diversas fórmulas en nuestro documento.

4.1 Para iniciar

Para escribir fórmulas matemáticas, \LaTeX tiene lo que se conoce como «modo matemático» y para acceder a él necesitamos de instrucciones o entornos especiales.

Si nuestra fórmula irá dentro de un párrafo de texto, podemos utilizar las instrucciones:

$\$fórmula \$$,
 $\backslash(fórmula \backslash)$ y
el entorno *math*

Pero si lo que queremos es que nuestra fórmula aparezca en un párrafo aparte, ya sea para destacarla o porque es muy larga, podemos utilizar la instrucción:

$\backslash[fórmula \backslash]$ y
el entorno *displaymath*.

Por ejemplo:

Siendo a y b los catetos y c la hipotenusa de un triángulo rectángulo, entonces
$$c^2 = a^2 + b^2$$

(Teorema de Pitágoras).

Siendo a y b los catetos y c la hipotenusa de un triángulo rectángulo, entonces $c^2 = a^2 + b^2$ (Teorema de Pitágoras).

La diferencia entre las fórmulas dentro de un mismo párrafo y las compuestas en párrafos independientes se puede ver en el siguiente ejemplo:

Tenemos que: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

Tenemos que: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

Tenemos que: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

Tenemos que:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Como vemos, los exponentes se colocan de manera distinta, al igual que el tamaño para las fracciones varía. Esto se debe a que cuando la fórmula está dentro de un párrafo de texto, \LaTeX busca que la fórmula tenga el mismo alto que el demás texto, y cuando se tiene la fórmula en un párrafo independiente se utilizan tipos más grandes.

Es común que las ecuaciones o fórmulas de nuestro documento vayan numeradas, para lograr ésto tenemos el entorno *equation*. Y si queremos hacer referencia a la ecuación, simplemente utilizamos `\label` y `\ref`, como ya lo hemos visto; además existe la instrucción

`\eqref{Etiqueta}`

que funciona igual que `\ref`, salvo que coloca el número de la ecuación entre paréntesis. Para utilizar `\eqref` se necesita incluir el paquete `amsmath`.

```
\begin{equation} \label{eq:eps} \epsilon > 0 \tag{4.1}
\epsilon > 0
\end{equation}
De \eqref{eq:eps} se deduce...
```

Estas ecuaciones se centran al ancho del renglón y su número de referencia se coloca del lado derecho. Sin embargo, es posible modificar su presentación con las siguientes opciones de clase:

<i>fleqn</i>	Recorre las ecuaciones a la izquierda.
<i>leqno</i>	Coloca el número de referencia a la izquierda.

4.1.1 Sintaxis en modo matemático

Cuando estamos en modo matemático existen algunas diferencias entre el texto normal y es importante considerarlas. Estas diferencias son:

1. Los espacios en blanco y los cambios de línea no tienen ningún significado.
2. Las letras son consideradas como variables y se escribirán en cursiva.
3. No se pueden incluir caracteres acentuados como en el texto normal, para hacerlo se necesitan instrucciones especiales (véase el cuadro 4.4).

4. Sólo está permitido un párrafo por entorno.

Asimismo, para modificar la tipografía dentro de modo matemático se tienen instrucciones especiales. Estas instrucciones son las siguientes:

Instrucción	Muestra
<code>\mathnormal{ABCdef123}</code>	$ABCdef_{123}$
<code>\mathrm{ABCdef123}</code>	$ABCdef_{123}$
<code>\mathbf{ABCdef123}</code>	$ABCdef_{123}$
<code>\mathit{ABCdef123}</code>	<i>$ABCdef_{123}$</i>
<code>\mathtt{ABCdef123}</code>	$ABCdef_{123}$
<code>\mathcal{ABC}</code>	\mathcal{ABC}
<code>\mathbb{ABC}</code>	\mathbb{ABC}
<code>\mathfrak{ABCabc}</code>	\mathfrak{ABCabc}

Para utilizar las dos últimas instrucciones necesitamos incluir el paquete `amssymb` o `amsfonts`. Algo muy importante es que todas éstas deben utilizarse en modo matemático, de otra forma \LaTeX mandará un error.

Además de lo que explicamos arriba, es necesario considerar lo siguiente.

Agrupaciones

En modo matemático casi todas las instrucciones afectan sólo al carácter que le sigue. Para que la instrucción afecte a más caracteres, éstos se deberán encerrar entre llaves (`{}`). Por ejemplo:

$$\backslash[a^x+y \neq a^{x+y}]\qquad a^x + y \neq a^{x+y}$$

Notación matemática

- Los exponentes se indican con el acento circunflejo (`^`) y los subíndices con el guión bajo (`_`).
- Existen varios tipos de puntos suspensivos: los alineados a la base del renglón (`\ldots`, `...`), centrados en el renglón (`\cdots`, `...`), verticales (`\vdots`, `...`) y diagonales (`\ddots`, `...`).

- La n -ésima raíz se introduce con la instrucción

$$\sqrt[n]{\textit{radicando}}$$

si se omite la n se compondrá como raíz cuadrada.

- Una fracción se compone con la instrucción

$$\frac{\textit{numerador}}{\textit{denominador}}$$

- Operadores como el signo de integral (`\int`), la suma (`\sum`) y otros se muestran en el cuadro 4.1. Los límites se indican como exponentes y subíndices.

\bigcap	<code>\bigcap</code>	\bigcup	<code>\bigcup</code>	\biguplus	<code>\biguplus</code>	\bigsqcup	<code>\bigsqcup</code>
\bigodot	<code>\bigodot</code>	\bigoplus	<code>\bigoplus</code>	\bigotimes	<code>\bigotimes</code>		
\prod	<code>\prod</code>	\sum	<code>\sum</code>	\int	<code>\int</code>	\oint	<code>\oint</code>
\bigwedge	<code>\bigwedge</code>	\bigvee	<code>\bigvee</code>				

Cuadro 4.1: Operadores

<code>arccos</code>	<code>\arccos</code>	<code>csc</code>	<code>\csc</code>	<code>ker</code>	<code>\ker</code>	mín	<code>\min</code>
<code>arcsin</code>	<code>\arcsin</code>	<code>deg</code>	<code>\deg</code>	<code>lg</code>	<code>\lg</code>	Pr	<code>\Pr</code>
<code>arctan</code>	<code>\arctan</code>	<code>det</code>	<code>\det</code>	lím	<code>\lim</code>	sec	<code>\sec</code>
<code>arg</code>	<code>\arg</code>	<code>dim</code>	<code>\dim</code>	líminf	<code>\liminf</code>	sin	<code>\sin</code>
<code>cos</code>	<code>\cos</code>	<code>exp</code>	<code>\exp</code>	límsup	<code>\limsup</code>	sinh	<code>\sinh</code>
<code>cosh</code>	<code>\cosh</code>	<code>gcd</code>	<code>\gcd</code>	ln	<code>\ln</code>	sup	<code>\sup</code>
<code>cot</code>	<code>\cot</code>	<code>hom</code>	<code>\hom</code>	log	<code>\log</code>	tan	<code>\tan</code>
<code>coth</code>	<code>\coth</code>	<code>inf</code>	<code>\inf</code>	máx	<code>\max</code>	tanh	<code>\tanh</code>

Cuadro 4.2: Funciones matemáticas

- Las funciones matemáticas siempre deben ir con tipografía normal, nunca en *cursiva*. Por esta razón se tienen las instrucciones del cuadro 4.2 para escribir los nombres de funciones.

Para comprender mejor esto, veamos algunos ejemplos.

```

Dado un proceso estocástico  $\{X_t\}_{t \in T}$ 
y dado  $F \subset T$  se define la función  $\varphi_{F,X}(\omega, \mathcal{F})$  como
\begin{equation}
\varphi_{F,X}(\omega) = \{X_t(\omega)\}_{t \in F}
\end{equation}
(si  $F=T$  ponemos simplemente  $\varphi_X$  en lugar de
 $\varphi_{F,X}$ ). Es claro que  $\varphi_{F,X}$  es medible
debido a que para cada  $t \in F$ ,  $\varphi_{F,X}^t$  ó
 $\varphi_{F,X}^t = X_t$  es medible.
\begin{enumerate}
\item A la función  $\varphi_{F,X}(\omega)$  se le llama
\textit{la trayectoria de  $\omega$ }.
\item La probabilidad  $P \circ \varphi_{F,X}^{-1} : \mathcal{E}^F \rightarrow [0,1]$ 
definida por
\begin{equation}
(P \circ \varphi_{F,X}^{-1})(A) = P(\varphi_{F,X}^{-1}(A)),
\quad \forall A \in \mathcal{E}^F
\end{equation}
se llama la \textit{ley} o la \textit{distribución}
del proceso  $X$ .
\end{enumerate}

```

Dado un proceso estocástico $\{X_t\}_{t \in T}$ y dado $F \subset T$ se define la función $\varphi_X^F : (\Omega, \mathcal{F}, P) \rightarrow (E^F; \mathcal{E}^F)$ como

$$\varphi_X^F(\omega) = \{X_t(\omega)\}_{t \in F} \quad (4.2)$$

(si $F = T$ ponemos simplemente φ_X en lugar de φ_X^T). Es claro que φ_X^T es medible debido a que para cada $t \in F$, π_t^F ó $\varphi_X^T = X_t$ es medible.

1. A la función $\varphi_X(\omega)$ se le llama *la trayectoria de ω* .
2. La probabilidad $P \circ \varphi_X^{-1} : \mathcal{E}^T \rightarrow [0, 1]$ definida por

$$(P \circ \varphi_X^{-1})(A) = P(\varphi_X^{-1}(A)), \quad \forall A \in \mathcal{E}^T \quad (4.3)$$

se llama la *ley* o la *distribución* del proceso X .

Sea $f(t) = e^{-ct}$, $t \geq 0$, en donde c es una constante real diferente de cero; entonces,

$$\int_0^\infty e^{-ct} dt = \lim_{A \rightarrow \infty} \int_0^A e^{-ct} dt = \lim_{A \rightarrow \infty} \left[-\frac{1}{c} e^{-ct} \right]_0^A = \lim_{A \rightarrow \infty} \left(-\frac{1}{c} e^{-cA} + \frac{1}{c} \right)$$

Se concluye que la integral impropia converge si $c < 0$ y diverge si $c > 0$. Si $c = 0$, entonces el integrando es la unidad y una vez más la integral diverge.

Sea $f(t) = e^{ct}$, $t \geq 0$, en donde c es una constante real diferente de cero; entonces,

$$\int_0^\infty e^{ct} dt = \lim_{A \rightarrow \infty} \int_0^A e^{ct} dt = \lim_{A \rightarrow \infty} \left[\frac{e^{ct}}{c} \right]_0^A$$

Se concluye que la integral impropia converge si $c < 0$ y diverge si $c > 0$. Si $c = 0$, entonces el integrando es la unidad y una vez más la integral diverge.

Delimitadores horizontales

Para poner líneas horizontales encima o debajo de una expresión se tienen las instrucciones:

`\overline{objetos}`
`\underline{objetos}`

y las instrucciones

`\overbrace{objetos}`
`\underbrace{objetos}`

componen llaves horizontales. Por ejemplo:

$$\underline{x} \quad \overline{m+n} \quad \underbrace{a+b+\cdots+z}_{26}$$

Delimitadores verticales

Para obtener los delimitadores de un tamaño adecuado para la ecuación, es necesario utilizar las instrucciones

```
\left
\right
```

seguidos del delimitador. Los distintos delimitadores que tiene \LaTeX se muestran en el cuadro 4.3. Por ejemplo:

```
Se tiene la desigualdad:
\left[\int_0^1 f(x)dx\right]\leq\int_0^1\left|f(x)\right|dx
Un intervalo semiabierto $\left(\frac{a}{b},\frac{c}{d}\right]$
```

Se tiene la desigualdad:

$$\left|\int_0^1 f(x)dx\right| \leq \int_0^1 |f(x)|dx$$

Un intervalo semiabierto $\left(\frac{a}{b}, \frac{c}{d}\right]$

(())	↑	\uparrow	⇧	\Uparrow
[[]]	↓	\downarrow	⇩	\Downarrow
{	\{	}	\}	↕	\updownarrow	⇕	\Updownarrow
<	\langle	>	\rangle		\vert		\Vert
⌊	\lfloor	⌋	\rfloor	⌈	\lceil	⌋	\rceil
/	/	\	\backslash				

Cuadro 4.3: Delimitadores verticales

Se puede sustituir por un punto (.) el delimitador que no se desee incluir, ya que \LaTeX busca un `\right` por cada `\left` encontrado.



Hay una barra de herramientas que contiene algunos botones para insertar código en modo matemático. Además hay dos listas que insertan delimitadores verticales.

También está los menús *[LaTeX/Ordenes Matemáticas]*, *[LaTeX/Entorno matemático estándar]* y *[LaTeX/Entornos matemáticos AMS]* que contienen las opciones para insertar instrucciones y entornos para modo matemático.

4.1.2 Símbolos más frecuentes

Como es sabido, las fórmulas matemáticas contienen gran variedad de símbolos. Dado que \LaTeX fue creado para ésto, tenemos muchísimas instrucciones para generar símbolos. Se presentan a continuación algunos de los más utilizados.

\hat{o}	<code>\hat{o}</code>	\check{o}	<code>\check{o}</code>	\tilde{o}	<code>\tilde{o}</code>	\acute{o}	<code>\acute{o}</code>
\grave{o}	<code>\grave{o}</code>	\dot{o}	<code>\dot{o}</code>	\ddot{o}	<code>\ddot{o}</code>	\breve{o}	<code>\breve{o}</code>
\bar{o}	<code>\bar{o}</code>	\vec{o}	<code>\vec{o}</code>				

Cuadro 4.4: Acentos en modo matemático

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	v	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ϵ	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

^aLas demás letras se omitieron debido a que su forma es igual en el alfabeto latino, por ejemplo, la alfa mayúscula es A y puede escribirse como `\mathrm{A}`.

Cuadro 4.5: Letras griegas

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>	\equiv	<code>\equiv</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\ll	<code>\ll</code>	\gg	<code>\gg</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>
\sim	<code>\sim</code>	\simeq	<code>\simeq</code>	\approx	<code>\approx</code>	\cong	<code>\cong</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code>

Cuadro 4.6: Relaciones binarias

$+$	<code>+</code>	$-$	<code>-</code>	\pm	<code>\pm</code>	\mp	<code>\mp</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\times	<code>\times</code>	\setminus	<code>\setminus</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\vee	<code>\vee</code>	\wedge	<code>\wedge</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\oslash	<code>\oslash</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\otimes	<code>\otimes</code>	\triangleleft	<code>\triangleleft</code>	\triangleright	<code>\triangleright</code>
\star	<code>\star</code>	\ast	<code>\ast</code>	\bullet	<code>\bullet</code>	\diamond	<code>\diamond</code>

Cuadro 4.7: Operadores binarios

\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>

Cuadro 4.8: Flechas



En la ventana de varios hay 5 pestañas que contienen una gran lista de símbolos matemáticos, para incluirlos sólo hay que hacer doble clic en el símbolo y se insertará el código correspondiente en el editor.

4.2 Matrices y determinantes

Si queremos crear matrices, tenemos el entorno *array*. Funciona igual que el entorno *tabular*, sólo que *array* debe utilizarse en modo matemático.

Por ejemplo:

<pre>\[\left \begin{array}{*{5}{r}} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{array} \right = 0</pre>	$\begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{vmatrix} = 0$
---	---

4.3 Ecuaciones largas

Es posible utilizar también el entorno *array* para poner, por ejemplo, funciones continuas por partes.

<pre>\begin{displaymath} f\left(x\right) =\left\{ \begin{array}{cc} \frac{x^2-x-2}{x-2} & \text{si } x \neq 2 \\ 1 & \text{si } x=2 \end{array} \right. \end{displaymath}</pre>	$f(x) = \begin{cases} \frac{x^2-x-2}{x-2} & \text{si } x \neq 2 \\ 1 & \text{si } x = 2 \end{cases}$
--	--

Para ecuaciones que ocupan más de un párrafo o para sistemas de ecuaciones podemos utilizar el entorno *eqnarray*. Funciona como una tabla con tres columnas. Por ejemplo:

<pre>\begin{eqnarray} f(x) & = & \cos x \\ f'(x) & = & -\sin x \\ \int_0^x f(y) dy & = & \sin x \end{eqnarray}</pre>	$\begin{aligned} f(x) &= \cos x & (4.4) \\ f'(x) &= -\sin x & (4.5) \\ \int_0^x f(y) dy &= \sin x & (4.6) \end{aligned}$
--	--

Existe el entorno *eqnarray** que no numera las ecuaciones; y dentro de un entorno *eqnarray* para indicar que no se numere la ecuación se escribe la instrucción `\nonumber`

antes del fin de renglón (`\\`).



Además de los menús *[LaTeX/Entorno matemático estándar]* y *[LaTeX/Entornos matemáticos AMS]* que contienen las opciones para insertar los entornos que hemos visto en estas dos secciones, en el menú *[Asistente/Matemática]* aparece un diálogo que inserta estos entornos y contiene algunas otras opciones que evitan tener que escribir todo el código.

4.4 Teoremas y demostraciones

En textos científicos es común que necesitemos incluir enunciados tales como teoremas, corolarios, lemas, definiciones, axiomas, casos, ejemplos, demostraciones, etc. Estos enunciados tienen un formato especial: se componen en un párrafo aparte y se enumeran para hacer referencias.

Para componer este tipo de enunciados \LaTeX nos permite definir entornos que tendrán un formato especial. La instrucción

```
\newtheorem{Nombre}[compartir-contador]{cabecera}[ligar-contador]
```

define un nuevo entorno para componer enunciados. El argumento *Cabecera* es el texto que se compone como título del enunciado, por ejemplo, Teorema. El argumento opcional *compartir-contador* sirve para que tanto el entorno *Nombre* como el definido como *compartir-contador* compartan la numeración. El argumento opcional *ligar-contador* define el contador compuesto por varios niveles, por ejemplo, si se escribe `chapter` la etiqueta mostrará el número del enunciado junto con el número del capítulo. Los enunciados se escriben dentro del entorno *Nombre*.

Con los siguientes ejemplos se podrá entender mejor esto.

- En el preámbulo definimos lo siguiente:

```
\newtheorem{propC}{Proposición}[chapter]
```

```
\newtheorem{propS}{Proposición}[section]
```

```
\newtheorem{prop}{Proposición}
```

Utilizando los entornos tenemos lo siguiente:

```
\begin{propC}
El conjunto de los números primos es infinito.
\end{propC}
\begin{propS}
El conjunto de los números primos es infinito.
\end{propS}
\begin{prop}
El conjunto de los números primos es infinito.
\end{prop}
```

Proposición 4.1 *El conjunto de los números primos es infinito.*

Proposición 4.4.1 *El conjunto de los números primos es infinito.*

Proposición 1 *El conjunto de los números primos es infinito.*

- Para ligar dos contadores definimos los entornos de la siguiente manera:

```
\newtheorem{definicion}{Definición}
```

```
\newtheorem{proposicion}[definicion]{Proposición}
```

Utilizando estos entornos tenemos lo siguiente:

```
\begin{definicion}
Sea  $\{X_t\}_{t \in T}$  un proceso estocástico con
espacio de estados  $(E, \mathcal{E})$ , ...
\end{definicion}
\begin{proposicion}
Sea  $T$  un espacio métrico separable,  $E$  espacio
métrico y  $\{X_t\}_{t \in T}$  un proceso estocástico...
\end{proposicion}
```

Definición 1 *Sea $\{X_t\}_{t \in T}$ un proceso estocástico con espacio de estados (E, \mathcal{E}) ,...*

Proposición 2 *Sea T un espacio métrico separable, E espacio métrico y $\{X_t\}_{t \in T}$ un proceso estocástico...*

Además, todos los enunciados tienen un argumento opcional *identificación* que permite citar el autor original o el nombre con el cual se conoce al enunciado.

```
\begin{nombre}[identificación]
```

Esta *identificación* se compone entre paréntesis y en negrita, en la cabecera del enunciado. Por ejemplo:

```
\begin{teorema}[Grassman]
Sean  $F$  y  $G$  dos subespacios vectoriales de  $E$  y
supongamos que la dimensión de  $E$  es finita.
Entonces  $F$ ,  $G$ ,  $F \cap G$  y  $F + G$  son todos de
dimensión finita y

$$\dim F + \dim G = \dim(F + G) + \dim(F \cap G).$$

\end{teorema}
```

Teorema 1 (Grassman) *Sean F y G dos subespacios vectoriales de E y supongamos que la dimensión de E es finita. Entonces F , G , $F \cap G$ y $F + G$ son todos de dimensión finita y*

$$\dim F + \dim G = \dim(F + G) + \dim(F \cap G).$$

Para hacer una referencia a los enunciados sólo tenemos que incluir la instrucción `\label` dentro del entorno.

Más información

- Los paquetes desarrollados por la AMS (*American Mathematical Society*) tienen una gran cantidad de instrucciones para componer matemáticas.
- Existe el paquete `chemsym`, que es muy útil para componer fórmulas químicas.

Capítulo 5

Bibliografía

La bibliografía es una parte muy importante cuando hacemos un trabajo de investigación, ya que, a quienes les interese el tema, podrán continuar con la investigación o profundizar en algún tema específico. De igual manera, cuando hacemos una cita textual de un fragmento de otro trabajo o incluimos datos extraídos de otra fuente, es preciso dar una referencia bibliográfica completa de la fuente citada, además de enlazar la cita con la referencia bibliográfica correspondiente mediante una «cita bibliográfica».

Normalmente, al final del trabajo se debe incluir una lista de todas las referencias bibliográficas correspondientes a las citas bibliográficas hechas dentro del escrito.

5.1 Sistemas de citación bibliográfica

Cada editorial y escuela tiene un estilo propio para componer las citas bibliográficas, pero tenemos básicamente tres estilos para componerlas: el sistema «autor-fecha», el sistema «numérico» y el sistema de «referencias insertadas», además de un sistema mixto, el sistema «autor-fecha abreviado», que combina las ventajas de los sistemas autor-fecha y numérico.

Brevemente veremos cómo se componen estos estilos.

Sistema autor-fecha

Las citas bibliográficas se forman con el apellido del autor seguido del año de la publicación, este último entre paréntesis. Cuando se trata de dos autores se incluyen ambos, por ejemplo: Pujol y Solà (1995), puede utilizarse también la ligadura latina & para separar los autores. Si son más de dos sólo se incluye el primero seguido de la abreviación latina et al., por ejemplo: Goossens et al. (1993). En caso de haber dos o más citas idénticas pero correspondientes a publicaciones diferentes, se distinguen mediante una letra al final del año de la publicación: Knuth (1986a), Knuth (1986b), etc.

Este sistema permite agregar, modificar y borrar citas fácilmente. Sin embargo, la desventaja principal es el espacio que ocupan, esto puede distraer al lector del contenido del trabajo.

Sistema numérico

Con este sistema, las citas se numeran entre corchetes, por ejemplo, la primera entrada en la lista de referencias bibliográficas será: [1]. En cuanto al orden en que se numeran las citas,

se tienen dos opciones: según el orden de aparición dentro del trabajo, o por orden alfabético de los autores de las referencias.

Si bien, este sistema es muy compacto, la desventaja principal es que es poco descriptivo: el lector tendrá que consultar frecuentemente las referencias bibliográficas durante la lectura del trabajo.

Sistema de referencias insertadas

Con el sistema de referencias insertadas las citas bibliográficas se reemplazan por referencias bibliográficas completas. Por ejemplo, en lugar de escribir Pujol y Solà (1995) ó [1], se incluye dentro del texto la referencia bibliográfica correspondiente entre paréntesis, por ejemplo cuando se comenta el libro de J. M. Pujol y J. Solà (Ortotipografia. Manual de l'autor, l'autoeditor y el dissenyador gràfic. Columna, Barcelona, 1995).

Este sistema resulta muy compacto y descriptivo, pero tiene la gran desventaja de distraer al lector del contenido, además de obligar a volver a escribir las referencias bibliográficas enteras cada vez que se hace una cita.

Sistema autor-fecha abreviado

Las ventajas del sistema autor-fecha y del sistema numérico se combinan en una variante del sistema autor-fecha, que consiste en abreviar el apellido del autor a las tres primeras letras, seguidas de los dos últimos dígitos del año de publicación y cerrar la cita entre corchetes: [Knu86]. En caso de tener dos o tres autores, se junta la primera letra del apellido de cada uno de los autores, y si son más de tres, se escriben las iniciales de los dos primeros seguidos de un signo más (+), por ejemplo: [PS95] y [GM+94].

A pesar de que este sistema no permite distinguir las citas bibliográficas idénticas pero de publicaciones de siglos diferentes, es un sistema de cita bibliográfica muy utilizado.

5.2 Estilos bibliográficos

Para la presentación de las referencias bibliográficas, se tienen cinco estilos básicos: el estilo de «notas», el «numérico», el «alfabético», el «alfabético abreviado» y el estilo «chicago».

Con el estilo bibliográfico de notas, junto con el sistema numérico de citas, las referencias bibliográficas se disponen como notas a pie de página.

El estilo bibliográfico numérico, usado juntamente con el sistema numérico de citas, compone las referencias, ya sea ordenadas alfabéticamente o por orden de citación, pero siempre al final del trabajo y llevan por etiqueta la cita bibliográfica correspondiente..

El estilo bibliográfico alfabético, junto con el sistema autor-fecha abreviado de citas, dispone las referencias bibliográficas por orden alfabético al final del trabajo y su etiqueta correspondiente, en formato autor-fecha abreviado.

El estilo bibliográfico alfabético abreviado es una variante del estilo bibliográfico alfabético. A diferencia del anterior, este estilo lleva el nombre de los autores abreviado con las iniciales.

Por último, el estilo bibliográfico *chicago* es un estilo muy difundido en el mundo académico anglosajón, pero ha sido adoptado por varias asociaciones. Este estilo, juntamente con el sistema autor-fecha, se pone al final del trabajo, y se compone el apellido antes del nombre del autor, y no se etiquetan.

5.3 Composición de bibliografía y referencias bibliográficas con L^AT_EX

Las citas bibliográficas son referencias cruzadas hacia una lista de referencias bibliográficas. De igual manera que funcionan las instrucciones `\label` y `\ref`, se tienen las instrucciones `\bibitem` y `\cite` para las citas bibliográficas.

La instrucción

```
\cite[nota]{clave}
```

introduce una cita bibliográfica hacia la referencia bibliográfica que tiene *clave* por identificador. El parámetro *nota* permite componer una nota como parte de la cita bibliográfica, por ejemplo, mencionar la página de donde se extrajo la información.

Las estructuras de grafo~\cite[pág.~188]{loewe} han sido interpretadas en~\cite{atupa} como álgebras parciales unarias.

Las estructuras de grafo (Löw93, pág. 188) han sido interpretadas en (BR+96) como álgebras parciales unarias.

Cuando tenemos que hacer dos o más citas bibliográficas consecutivas, debemos incluirlas todas dentro de una misma instrucción `\cite` separadas por comas.

Como complemento de las citas bibliográficas es necesaria una bibliografía o una lista de referencias bibliográficas. Existe el entorno *thebibliography* que compone la bibliografía o lista de referencias bibliográficas. Los elementos dentro de este entorno se especifican con la instrucción

```
\bibitem[cita]{clave} datos de la referencia
```

donde *cita* es la cita bibliográfica, *clave* es una etiqueta para identificar la referencia y *datos de la referencia* es la información de la referencia bibliográfica. Este entorno funciona como un entorno *itemize*, pero los ítems se separan con `\bibitem` en lugar de `\item`.

A continuación mostramos algunos ejemplos para componer la bibliografía.

- Sistema numérico de cita y estilo bibliográfico *numérico*:

Algunos autores, como por ejemplo~\cite{ortotipografia}, afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

```
\begin{thebibliography}{9}
\bibitem{ortotipografia} Josep M. Pujol y Joan Solà.
\textit{Ortotipografía. Manual de l'autor, l'autoeditor i el dissenyador gràfic}. Columna, Barcelona, 1995.
\end{thebibliography}
```

Algunos autores, como por ejemplo [1], afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

Referencias

- [1] Josep M. Pujol y Joan Solà. *Ortotipografia. Manual de l'autor, l'autoeditor i el dissenyador gràfic*. Columna, Barcelona, 1995.

- Sistema autor-fecha de cita y estilo bibliográfico alfabético:

Algunos autores, como por ejemplo~\cite{ortotipografia}, afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

```
\begin{thebibliography}{XXX99}
\bibitem[PS95]{ortotipografia} Josep M. Pujol y Joan Solà.
\textit{Ortotipografia. Manual de l'autor, l'autoeditor i el
dissenyador gràfic}. Columna, Barcelona, 1995.
\end{thebibliography}
```

Algunos autores, como por ejemplo [PS95], afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

Referencias

- [PS95] Josep M. Pujol y Joan Solà. *Ortotipografia. Manual de l'autor, l'autoeditor i el dissenyador gràfic*. Columna, Barcelona, 1995.

- Sistema autor-fecha abreviado de cita y estilo bibliográfico alfabético abreviado:

Algunos autores, como por ejemplo~\cite{ortotipografia}, afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

```
\begin{thebibliography}{XXX99}
\bibitem[PS95]{ortotipografia} J. M. Pujol y J. Solà.
\textit{Ortotipografia. Manual de l'autor, l'autoeditor i el
dissenyador gràfic}. Columna, Barcelona, 1995.
\end{thebibliography}
```

Algunos autores, como por ejemplo [PS95], afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

Referencias

[PS95] J. M. Pujol y J. Solà. *Ortotipografia. Manual de l'autor, l'autoeditor i el dissenyador gràfic*. Columna, Barcelona, 1995.

- Sistema autor-fecha de cita y estilo bibliográfico chicago:

Algunos autores, como por ejemplo~ \cite{ortotipografia}, afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

```
\begin{thebibliography}{}
\bibitem[Pujol y Solà, 1995]{ortotipografia} Pujol, J.~M. y Solà J. (1995)
\textit{Ortotipografia. Manual de l'autor, l'autoeditor i el
dissenyador gràfic}. Columna, Barcelona.
\end{thebibliography}
```

Algunos autores, como por ejemplo (Pujol y Solà, 1995), afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

Referencias

Pujol, J. M. y Solà, J. (1995). *Ortotipografia. Manual de l'autor, l'autoeditor i el dissenyador gràfic*. Columna, Barcelona.

- Sistema *numérico* de cita y estilo bibliográfico *notas*:

Algunos autores\footnote{Pujol, J.~M. y Solà J.
\textit{Ortotipografia. Manual de l'autor, l'autoeditor i el
dissenyador gràfic}. Columna, Barcelona, 1995.} afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

Algunos autores¹ afirman que la ortografía y la tipografía tienden, hoy en día, hacia la ortotipografía.

¹Pujol, J. M. y Solà J. *Ortotipografia. Manual de l'autor, l'autoeditor i el dissenyador gràfic*. Columna, Barcelona, 1995.



En el menú [LaTeX/Tablas y listas] está la opción que inserta un entorno *thebibliography*. En la misma lista desplegable que inserta las etiquetas, referencias y notas a pie de página, se puede insertar una cita (\cite).

5.4 Bases de datos bibliográficos con BibT_EX

Muchos de los datos bibliográficos que utilizamos en un trabajo podemos reutilizarlos, así que una cuidadosa recolección y un registro correcto de esos datos puede ser buena inversión. BibT_EX es un sistema de marcado que comprende un formato estándar para la compilación de referencias bibliográficas, un conjunto de estilos bibliográficos estándares, un lenguaje de programación para la definición de nuevos estilos bibliográficos y programas informáticos de soporte a la gestión de bases de datos bibliográficos.

Dentro de una base de datos bibliográficos en BibT_EX las referencias tienen la forma

```
@categoría { clave,
    entrada = "texto",
    entrada = "texto",
    :
    entrada = "texto" }
```

donde *clave* es el identificador usado para hacer citas bibliográficas mediante la instrucción `\cite`. El *texto* asociado a cada *entrada* tiene que ir delimitado por comillas dobles o por llaves de apertura y cierre, salvo que se trate de un número (en este caso, las llaves son optativas).

Dentro de BibT_EX pueden definirse abreviaciones con la instrucción:

```
@string{abreviación = "texto"}
```

Las categorías estándares de BibT_EX son las siguientes:

article Un artículo publicado en una revista. Entradas obligatorias: *author*, *title*, *journal* y *year*. Entradas optativas: *volume*, *number*, *pages*, *month* y *note*.

book Un libro. Entradas obligatorias: *author* o *editor*, *title*, *publisher* y *year*. Entradas optativas: *volume* o *number*, *series*, *address*, *edition*, *month* y *note*.

booklet Un documento impreso y encuadernado pero que no ha sido publicado. Entrada obligatoria: *title*. Entradas optativas: *author*, *howpublished*, *address*, *month*, *year* y *note*.

inbook Una parte de un libro, como por ejemplo un capítulo o una serie de páginas consecutivas. Entradas obligatorias: *author* o *editor*, *title*, *chapter* y/o *pages*, *publisher* y *year*. Entradas optativas: *volume* o *number*, *series*, *type*, *address*, *edition*, *month* y *note*.

incollection Una parte de un libro con un título propio. Entradas obligatorias: *author*, *title*, *booktitle*, *publisher* y *year*. Entradas optativas: *editor*, *volume* o *number*, *series*, *type*, *chapter*, *pages*, *address*, *edition*, *month* y *note*.

inproceedings Un artículo publicado en la miscelánea de un congreso. Entradas obligatorias: *author*, *title*, *booktitle* y *year*. Entradas optativas: *editor*, *volume* o *number*, *series*, *pages*, *address*, *month*, *organization*, *publisher* y *note*.

manual Un manual de documentación técnica. Entrada obligatoria: *title*. Entradas optativas: *author*, *organization*, *address*, *edition*, *month*, *year* y *note*.

masterthesis Una tesis de licenciatura. Entradas obligatorias: *author*, *title*, *school* y *year*. Entradas optativas: *type*, *address*, *month* y *note*.

misc Un trabajo que no se ajusta a ninguna de las categorías. No tiene entradas obligatorias. Entradas optativas: *author*, *title*, *howpublished*, *month*, *year* y *note*.

phdthesis Una tesis doctoral. Entradas obligatorias: *author*, *title*, *school* y *year*. Entradas optativas: *type*, *address*, *month* y *note*.

proceedings La miscelánea de un congreso. Entradas obligatorias: *title* y *year*. Entradas optativas: *editor*, *volume* o *number*, *series*, *month*, *organization*, *publisher* y *note*.

techreport Un reporte técnico o de investigación. Entradas obligatorias: *author*, *title*, *institution* y *year*. Entradas optativas: *type*, *number*, *address*, *month* y *note*.

unpublished Un trabajo inédito. Entradas obligatorias: *author*, *title* y *note*. Entradas optativas: *month* y *year*.

Ahora bien, las entradas estándar de información son las que se muestran en el cuadro 5.1.

Dentro de las entradas *author* y *editor* el nombre tiene que escribirse antes del apellido, o bien, el nombre después del apellido pero separados por una coma. Por ejemplo, ambas entradas son equivalentes:

```
author = "Kurt Gödel"
```

```
author = "Gödel, Kurt"
```

Para incluir los dos apellidos del autor o editor, es preciso utilizar siempre la segunda forma.

Por ejemplo, la definición de un libro es la siguiente:

```
@book{ LaTeX:Goossens,
  author   = {Goossens, Michel and Mittelbach, Frank
              and Samarin, Alexander},
  title    = {The \LaTeX\ Companion},
  publisher = {Addison-Wesley},
  year     = {1994}
}
```

Ahora bien, para hacer una cita dentro de L^AT_EX se utiliza la instrucción `\cite`; para este ejemplo sería:

```
\cite{LaTeX:Goossens}
```

5.4.1 Uso de la base de datos bibliográficos en L^AT_EX

Una vez elaborada la base de datos bibliográficos con BIBT_EX, puede ocuparse dentro de L^AT_EX. La instrucción:

```
\bibliography{lista de archivos}
```

sustituye al entorno *bibliography* y especifica el nombre o los nombres de los archivos con extensión *.bib* que contienen la base de datos bibliográficos, es decir, que contienen las referencias bibliográficas correspondientes a las citas bibliográficas hechas en el documento.

Durante la composición de la bibliografía solamente se incluirá la lista de las referencias bibliográficas citadas a lo largo del documento. Pero con la instrucción:

```
\nocite{lista de claves}
```

<i>address</i>	La dirección de la editorial o la institución que publica oficialmente el trabajo.
<i>annote</i>	Comentario. Se puede utilizar para componer una bibliografía comentada.
<i>author</i>	El nombre del autor o los autores.
<i>booktitle</i>	El título completo del libro o el volumen de miscelánea con el subtítulo, si lo hay.
<i>chapter</i>	El número del capítulo o apartado.
<i>crossref</i>	La clave de la referencia bibliográfica de la cual se hace una referencia cruzada.
<i>edition</i>	El ordinal de la edición concreta, si no es la primera, por ejemplo: Segunda.
<i>editor</i>	El nombre del editor o los editores.
<i>howpublished</i>	Indicación de cómo se ha hecho público un trabajo de tipo booklet o misc.
<i>institution</i>	El nombre de la institución que publica oficialmente el trabajo.
<i>journal</i>	El título de la revista.
<i>key</i>	Una clave alternativa que se puede usar para ordenar alfabéticamente la referencia bibliográfica, para hacer referencias cruzadas y para componer la cita bibliográfica correspondiente cuando faltan los datos <i>author</i> y <i>editor</i> .
<i>month</i>	El mes de la publicación o el mes en que fue escrito, en el caso de un trabajo inédito.
<i>note</i>	Información adicional que pueda resultar de interés para el lector.
<i>number</i>	El número de una revista, un reporte o un trabajo dentro de una serie.
<i>organization</i>	El nombre de la institución que organiza el congreso o que publica el manual de documentación técnica.
<i>pages</i>	La paginación del trabajo.
<i>publisher</i>	El nombre de la editorial que publica el libro o el volumen de miscelánea.
<i>school</i>	El nombre de la universidad, la facultad o el departamento donde se ha presentado la tesis doctoral o de licenciatura.
<i>series</i>	El nombre de la serie o de la colección de libros o volúmenes de miscelánea.
<i>title</i>	El título completo del trabajo científico con el subtítulo, si lo hay.
<i>type</i>	La clase de reporte, técnico o de investigación, la clase de parte de un libro en un trabajo de tipo inbook o incollection, o la clase de tesis, de licenciatura o doctoral.
<i>volume</i>	El número de volúmenes de una revista científica o de un libro de varios volúmenes.
<i>year</i>	El año de publicación o el año en que fue escrito, en el caso de un trabajo inédito.

Cuadro 5.1: Entradas estándar de información

se incluirán también las referencias bibliográficas que tienen los argumentos *clave* como clave de cita. Y con la instrucción

```
\nocite{*}
```

se incluirán todas las referencias bibliográficas de una base de datos bibliográficas.

Estilos bibliográficos

En cuanto a los estilos bibliográficos estándares, la instrucción:

```
\bibliographystyle{estilo}
```

especifica tanto el estilo de cita bibliográfica como el estilo bibliográfico, donde *estilo* es uno de los siguientes:

plain	Sistema numérico de citación y estilo bibliográfico numérico. Las referencias bibliográficas son ordenadas alfabéticamente.
unsrt	Sistema numérico de citación y estilo bibliográfico numérico. Las referencias bibliográficas no son ordenadas alfabéticamente sino que mantienen el orden que tienen dentro de la base de datos bibliográficos.
alpha	Sistema autor-fecha abreviado de citación y estilo bibliográfico alfabético.
abbrv	Sistema autor-fecha abreviado de citación y estilo bibliográfico alfabético abreviado.

Además de estos cuatro estilos estándares, el estilo *chicago* permite utilizar el sistema autor-fecha de citación y el estilo bibliográfico *chicago*.

Por último, para que la bibliografía aparezca en nuestro documento, debemos hacer lo siguiente:

1. Compilar una vez con \LaTeX nuestro archivo `tex`.
2. Compilar el mismo archivo `tex` pero ahora con el programa `bibtex`
3. Compilar nuevamente el archivo `tex` con \LaTeX .

obviamente, ninguna de estas compilaciones deberá arrojar errores.



En el menú *[LaTeX/Bibliografía]* están las opciones para crear un archivo `bib`.

Más información

- El paquete `cite` modifica la presentación de las citas, por ejemplo, `[1,2,3,4,8]` aparecerá como `[1–4,8]` en el documento.
- El paquete `citesort` además de hacer lo mismo que el paquete `cite`, ordena las citas antes de agruparlas; por ejemplo, `[1,2,3,4,7,6,5,8]` aparecerá como `[1–8]`.

- El paquete `footbib` permite componer las citas bibliográficas en el estilo de notas, pero la información se toma de la base de datos o del entorno *thebibliography*.
- Con el paquete `bibunits` permite incluir varias listas de referencias bibliográficas en distintas partes del documento. Por ejemplo, se puede incluir la bibliografía de cada capítulo al final del mismo.
- Existe el paquete `gloss` que permite crear glosarios para nuestro documento. Este paquete utiliza B_IB_T_EX.

Capítulo 6

Índices

Con todo que hemos visto en los capítulos anteriores es posible crear un documento completo. Lo que trataremos en este capítulo es cómo crear índices alfabéticos. Los índices son muy utilizados en trabajos grandes, como libros, ya que permiten localizar rápidamente la información que se necesita sin tener que recorrer todo el documento. Un buen índice debe dar mayor información que la tabla de contenidos.

Un índice alfabético es una colección de entradas ordenadas alfabéticamente, que corresponden a los conceptos principales que son tratados en el documento y facilitan su localización mediante referencias al número o a los números de página correspondientes, como también a otros conceptos que están relacionados directa o indirectamente.

Cuando las entradas del índice alfabético contienen muchas referencias, resulta conveniente desglosar las entradas en subentradas, que corresponden a subconceptos del concepto principal.

L^AT_EX cuenta con el entorno *theindex* que sirve para crear un índice alfabético. La estructura de este entorno es la siguiente:

```
\begin{theindex}
\item{Entrada, Página}
\subitem{SubEntrada, Página}
\subsubitem{SubSubEntrada, Página}
...
\end{theindex}
```

En cada `\item` se coloca la entrada correspondiente, pudiendo colocar hasta tres niveles de profundidad para la *Entrada*.

Este entorno se deberá colocar donde queremos que aparezca el índice. La gran desventaja de utilizar este entorno es que los números de página tendrán que actualizarse cada vez que se hagan cambios en el texto principal, además de que se deberán ordenar alfabéticamente «manualmente».

6.1 Preparación de índices alfabéticos con MakeIndex

Una forma de facilitar la tarea para crear un índice alfabético es utilizar el programa MAKEINDEX, junto con algunos paquetes; principalmente existe el paquete `makeidx`.

Una vez incluido el paquete `makeidx`, tenemos que incluir la instrucción

```
\makeindex
```


en el preámbulo de nuestro documento, e incluir la instrucción

```
\printindex
```

en donde queremos componer el índice alfabético. Por lo general el índice alfabético se coloca al final del documento.

Una vez hecho esto, la instrucción

```
\index{texto}
```

define la entrada *texto* que aparecerá en el índice alfabético. Además esta instrucción agregará el número de página donde se encuentra la entrada.

El índice alfabético se compone dentro de un entorno *theindex* en donde se ordenan las entradas alfabéticamente y se componen a dos columnas. Este entorno se incluye en el documento con la instrucción `\printindex`.

El texto de las subentradas y subsubentradas de una entrada del índice alfabético se separan del texto de la entrada correspondiente mediante signos de admiración (!). La instrucción

```
\index{texto!subtexto}
```

define la subentrada *subtexto* dentro de la entrada *texto*, mientras que la instrucción

```
\index{texto!subtexto!subsubtexto}
```

define la subsubentrada *subsubtexto* de la subentrada *subtexto* dentro de la entrada *texto*.

Veamos un ejemplo.

Página 63:	<code>\index{bibliografías!concepto}</code>
Página 64:	<code>\index{bibliografías!estilo!descriptivo}</code>
Página 64:	<code>\index{bibliografías!estilo!normalizado}</code>
Página 71:	<code>\index{bibliografías!estilo!descriptivo}</code>
Página 81:	<code>\index{bibliografías!ordenación}</code>
Página 85:	<code>\index{bibliografías!estilo!normalizado}</code>
Página 69:	<code>\index{bibliografías!estilo!descriptivo}</code>

Los números de página indican en dónde se definió la entrada, y el resultado de la compilación es el siguiente:

Índice alfabético

```

bibliografías
  concepto, 63
  estilo
    descriptivo, 64, 71, 89
    normalizado, 64, 85
  ordenación, 81

```

Algunas otras opciones se verán a continuación.

Para especificar el rango de páginas en donde se trata el concepto se puede utilizar la misma instrucción `\index` con la siguiente modificación:

```
\index{texto|{}
```

con `|{` indicamos que inicia la definición de la entrada *texto*, y con

```
\index{texto|})}
```

indicamos que la definición de la entrada *texto* ha terminado. Por ejemplo:

```
Página 63: \index{bibliografías!concepto|{}
```

```
Página 64: \index{bibliografías!concepto|})}
```

```
Página 64: \index{bibliografías!estilo!descriptivo|{}
```

```
Página 66: \index{bibliografías!estilo!descriptivo|})}
```

```
Página 71: \index{bibliografías!estilo!descriptivo}
```

```
bibliografías
concepto, 63–64
estilo
descriptivo, 64–66, 71
```

MAKEINDEX sustituirá todas las entradas que comienzan y terminan en la misma página por una entrada con referencia a una sola página. Además, elimina todas las entradas dentro de un rango de páginas definido con la misma entrada.

Ahora bien, en cuando a la ordenación alfabética, MAKEINDEX coloca las palabras con acento después de las palabras sin acento gráfico. No obstante, es posible especificar la ordenación alfabética de cada entrada con la instrucción

```
\index{orden@texto}
```

Con esta instrucción se define la entrada *texto* que será ordenada según el texto *orden*.

También es posible resaltar el texto de los números de página. Para lograr esto utilizamos la instrucción

```
\index{texto|estilo}
```

la cual define la entrada *texto* y el número de página correspondiente se compone según el *estilo* especificado, es decir, el número de página *n* se compone como `\estilo{n}`. Por ejemplo:

```
Página 281: \index{bibliografías|textit}
```

```
Página 296: \index{bibliografías|{}
```

```
Página 297: \index{bibliografías|})}
```

```
Página 308: \index{sistema autor-fecha|textbf}
```

```
Página 311: \index{sistema autor-fecha|{}
```

```
Página 314: \index{sistema autor-fecha|})}
```

```
bibliografías, 281, 296–297
sistema autor-fecha, 308, 311–314
```

Por último, para incluir una referencia cruzada dentro del índice alfabético se tiene la instrucción

```
\index{texto|see{referencia}}
```

Por ejemplo:

```
Página 121: \index{Valor presente neto}  
Página 122: \index{\textit{VPN}|see{Valor presente neto}}
```

Valor presente neto, 121
VPN, véase Valor presente neto



En el menú *[LaTeX/Tablas y listas]* se pueden insertar las instrucciones para generar e imprimir el índice.

En la lista desplegable de las referencias también se puede insertar la instrucción `\index`.

Más información

- El paquete `index`, además de redefinir algunas instrucciones para crear de manera más adecuada los índices, permite crear varios tipos de índices.

Capítulo 7

Personalizar tareas

Ahora veremos algunas formas de hacer tareas que \LaTeX no hace directamente. Como mencionamos al principio, \TeX y \LaTeX no son procesadores de texto comunes, sino que son compiladores, así que necesitamos de varias instrucciones para darle el formato a nuestro documento. Muchas tareas las podemos realizar fácilmente con las instrucciones y entornos que provee \LaTeX , sin embargo, habrá ciertos formatos que nosotros tendremos que definir.

7.1 Nuevas instrucciones

Es posible con \LaTeX crear nuevas instrucciones y entornos. Para las instrucciones tenemos las siguientes instrucciones:

```
\newcommand*{\Nombre}[NúmArg][ArgPred]{Definición}  
\renewcommand*{\Nombre}[NúmArg][ArgPred]{Definición}  
\providecommand*{\Nombre}[NúmArg][ArgPred]{Definición}
```

Con `\newcommand*` definimos una instrucción *Nombre*. Esta instrucción puede necesitar de 0 a 9 argumentos obligatorios (en el caso de ser cero, puede omitirse) y hasta un argumento optativo (que tomará el valor de 1); para utilizar estos argumentos se coloca el carácter # seguido del número del argumento. Lo que hará la instrucción es lo que definamos en *Definición*.

Veamos un ejemplo:

```
\newcommand*{\software}[2][\large]{\textit{#1#2}}
```

```
Existen varios programas que sirven para elaborar documentación  
técnica, entre ellos se tiene: \software[\huge]{FrameMaker},  
\software{Ventura Publisher}, \software{FreeHand},  
\software[\small]{Word},...
```

Existen varios programas que sirven para elaborar documentación técnica, entre ellos se tiene: *FrameMaker*, *Ventura Publisher*, *FreeHand*, *Word*,...

Como vemos en el ejemplo, definimos la instrucción `\software` que necesita de dos argumentos: uno optativo, el tamaño de la letra que está definido como `\large`, y el otro es el texto

que va entre llaves. Al utilizar la instrucción se compondrá el texto en cursiva y del tamaño especificado.

Si definimos una instrucción que ya existe, L^AT_EX mandará un error. Para modificar el comportamiento de una instrucción que ya existe, utilizamos `\renewcommand*` en lugar de `\newcommand*`. Por ejemplo

```
\renewcommand*{\textit}{\textbf{Ya no hace lo mismo que antes}}
```

La instrucción `\verb+\textit+` cambió: `\textit`.
Como vemos, ya no necesita de ningún argumento.

La instrucción `\textit` cambió: **Ya no hace lo mismo que antes**. Como vemos, ya no necesita de ningún argumento.

Sin embargo, no es recomendable redefinir las instrucciones de L^AT_EX, ya que podemos crear errores inesperados. Una forma de evitar esto es utilizando `\providecommand*`, la cual no realizará ningún cambio sobre la instrucción que definamos en caso de que dicha instrucción ya exista. Por ejemplo:

```
\providecommand*{\itshape}{\textbf{Ya no hace lo mismo que antes}}
```

La instrucción `\verb+\itshape+` no cambió: `\itshape`.
Como ya estaba definida, se ignora su modificación.

La instrucción `\itshape` no cambió: . Como ya estaba definida, se ignora su modificación.

Estas tres instrucciones no permiten argumentos que contengan varios párrafos, ya que mandarían errores al compilar. Pero si queremos que los acepte podemos utilizar las versiones sin asterisco de cada una: `\newcommand`, `\renewcommand` y `\providecommand`.

Sin embargo, pasar como argumentos varios párrafos ocupará más memoria en la compilación, ya que primero se guardan los argumentos y luego se aplican los cambios. En estos casos lo mejor es definir un entorno.

7.2 Nuevos entornos

Para definir entornos, tenemos las siguientes instrucciones:

```
\newenvironment*{Nombre}[NúmArg][ArgPred]{DefInicio}{DefFinal}  
\renewenvironment*{Nombre}[NúmArg][ArgPred]{DefInicio}{DefFinal}
```

las cuales definen un entorno *Nombre*. Los argumentos se manejan igual que en la definición de instrucciones. En la parte *DefInicio* se indica qué se hará cuando se escriba `\begin{Nombre}`, y en *DefFinal* se indica lo que se hará cuando se escriba `\end{Nombre}`.

Veamos un ejemplo:

```

\newenvironment*{cita}[1]%
{\newcommand\Autor{#1}\begin{quote}\itshape}%
{\end{quote}\centerline{\Autor}}

\begin{cita}{Javier}
Mi carrera ha sido lenta como la del caracol,
pero segura y sólida como su concha.
\end{cita}

```

Mi carrera ha sido lenta como la del caracol, pero segura y sólida como su concha.

Javier

En este ejemplo observamos que primero se inicia un entorno *quote* y el texto se colocará en cursiva y al final se colocará el nombre del autor de la cita. Como podemos ver, primero se definió la instrucción `\Autor` que guardará el argumento obligatorio; esto se debe a que no es posible utilizar directamente los argumentos en la parte *DefFinal*, sólo en *DefInicio*.

De igual manera que con las instrucciones, es posible utilizar las versiones sin asterisco de `\newenvironment` y `\renewenvironment`, para poder modificar más de un párrafo dentro del entorno, lo cual es más común.

7.3 Contadores

\LaTeX utiliza muchos contadores para componer un documento: el número de página, el número de capítulo, sección y subsección, el número de la nota a pie de página, etc. Estos contadores funcionan como una variable entera en los demás lenguajes, y podemos nosotros manipular los contadores según nuestras necesidades.

Para manipular un contador tenemos las instrucciones

```

\setcounter{NombreContador}{Valor}
\addtocounter{NombreContador}{Valor}

```

y para utilizar el valor del contador tenemos la siguiente sintaxis:

```
\theNombreContador
```

Veamos un ejemplo:

```

\setcounter{chapter}{7}
Este es el capítulo~\thechapter, en la página%
~\thepage.
\addtocounter{chapter}{-1}
Ahora se disminuye el capítulo en uno: \thechapter.

```

Este es el capítulo 7, en la página 72. Ahora se disminuye el capítulo en uno: 6.

7.4 Longitudes

Las longitudes también son muy utilizadas en \LaTeX para componer el documento. Las longitudes se ocupan, por ejemplo, para definir los márgenes, y vienen a ser como variables reales. Para manipular las longitudes tenemos las instrucciones

```
\setlength{\NombreLongitud}{Valor}
\addtolength{\NombreLongitud}{Valor}
```

el valor es un número real acompañado de su unidad de medida. Las unidades que \LaTeX acepta se muestran en el cuadro 7.1. Para mostrar el valor de la longitud utilizamos la siguiente sintaxis:

```
\the\NombreLongitud
```

la longitud se expresa en puntos. Por ejemplo:

```
\setlength{\parindent}{1cm}
```

En este párrafo se modificó el sangrado con la longitud `\verb+\parindent+`, y su valor es de `\the\parindent`.

En este párrafo se modificó el sangrado con la longitud `\parindent`, y su valor es de 28.45274pt.

Unidad	Significado	
mm	milímetro	□
cm	centímetro	□
in	pulgada	□
pt	punto	□
em	largo de una M	□
ex	alto de una x	□

Cuadro 7.1: Unidades de medida disponibles

Más información

- Existen más instrucciones para personalizar nuestras tareas. No se incluyen en este curso debido a que sería algo muy extenso y en ocasiones no tan necesario.
- Otras instrucciones que existen son `\def` y `\let`, propias de \TeX , que dan mayor control en la definición de instrucciones que las vistas aquí.
- Existen diversas instrucciones y entornos para manipular «cajas». Son muy útiles ya que todo en \LaTeX es una caja: una letra, una tabla, una página, etc.
- También es posible «programar» dentro de \LaTeX . Es posible utilizar condicionales y ciclos para manipular la composición de texto; algunas son propias de \TeX y otras de \LaTeX .

Capítulo 8

Presentaciones

También podemos crear presentaciones con \LaTeX , existen varios paquetes que generan presentaciones de alta calidad. Con algunos paquetes se podemos obtener, ya sea una versión impresa de la presentación, o electrónica, para proyectarla en pantalla (generalmente en formato pdf).

En este capítulo utilizaremos la clase `prosper` para crear presentaciones. Al final del capítulo colocamos un ejemplo completo de una presentación hecha con `prosper` para entender mejor cómo funciona.

8.1 Opciones de la clase `prosper`

La clase `prosper` se utiliza como las clases estándar de \LaTeX , pero con diferentes opciones, además de que incluye los paquetes `seminar`, `hyperref` y `graphicx`. Las opciones de esta clase se muestran en el cuadro 8.1; las opciones predeterminadas son: *ps*, *final*, *total*, *slideBW*, *nocolorBG*, *noaccumulate*.

Otra opción que debemos especificar es el «estilo». El estilo define los colores del texto, la tipografía y el fondo de las diapositivas. Algunos estilos disponibles son los siguientes:

alienglow	autumn	azure	contemporain
darkblue	frames	lignesbleues	nuancegris
troispoints	gyom	pascal	rico

8.2 Compilación

Para obtener nuestro archivo final de la presentación tenemos un proceso un poco distinto que cuando creamos un escrito. Este proceso es el siguiente:

1. Una vez que tenemos nuestro archivo `tex`, compilamos con `latex`. Esto crea el archivo `dvi`.
2. Utilizamos el programa `dvips` para convertir el archivo `dvi` a formato `ps`. En formato PostScript es posible imprimir la presentación.
3. Por último, necesitamos utilizar algún programa, como `GHOSTVIEW` o `PS2PDF` con `GHOSTSCRIPT` o `ACROBAT DISTILLER` para pasar el archivo `ps` a formato `pdf`. Este es el mejor formato para proyectar la presentación

Opción	Efecto
<i>ps, pdf</i>	Se genera un archivo PostScript o en formato pdf, respectivamente.
<i>final, draft</i>	Con <i>final</i> se incluye en el pie de la diapositiva la información de <code>\slideCaption</code> y con <i>draft</i> se incluye el nombre del archivo, el título, el autor y la fecha de compilación.
<i>total, nototal</i>	Con <i>total</i> aparece en el pie el número de diapositiva junto con el total de diapositivas y con <i>nototal</i> sólo se incluye el número de diapositiva correspondiente.
<i>slideBW, slideColor</i>	Componen la presentación a blanco y negro o a color, respectivamente.
<i>nocolorBG, colorBG</i>	Con <i>nocolorBG</i> el fondo de las diapositivas será blanco, mientras que con <i>colorBG</i> el fondo dependerá del «estilo» elegido.
<i>accumulate, noaccumulate</i>	Estas opciones son útiles para manipular las animaciones, <i>noaccumulate</i> muestra las animaciones y <i>accumulate</i> muestra toda la información de la diapositiva sin animaciones.

Cuadro 8.1: Opciones de la clase prosper

Si queremos compilar directamente a pdf con `pdflatex`, tendremos muchos errores.

8.3 Estructura del archivo fuente

Cuando utilizamos la clase `prosper`, la estructura del archivo de la presentación es más simple que la de un documento común. En el preámbulo, además de las instrucciones comunes (tipo de documento, inclusión de paquetes y definición de instrucciones) podemos incluir las siguientes:

```
\title{Título de la presentación}
\author{Autores}
\subtitle{Subtítulo}
\email{E-mails}
\institution{Organismo}
\Logo(x,y){Logotipo}
\slideCaption{Leyenda}
\displayVersion
\DefaultTransition{Transición}
```

Las instrucciones `\title` y `\author` son las únicas que necesitamos necesarias para generar la portada y funcionan igual que en las demás clases. Con `\Logo` se colocará el archivo *Logotipo* con extensión `eps` o `ps` en la posición (x,y) (la coordenada $(0,0)$ está en la esquina inferior izquierda de la diapositiva); si no se incluye la coordenada, el *Logotipo* se colocará en la posición que le asigne el estilo elegido.

La instrucción `\slideCaption` define la información que se presentará en el pie de la diapositiva, y con la declaración `\displayVersion` se colocará en el pie la información que se muestra al compilar con la opción *draft*.

La instrucción `\DefaultTransition` define la transición predeterminada de todas las diapositivas. *Transición* puede ser alguna de las mencionadas en el cuadro 8.2.

Opción	Efecto
<code>Split</code>	Dos líneas limpian la pantalla para mostrar la nueva diapositiva, como unas cortinas.
<code>Blinds</code>	Similar a <code>Split</code> , pero con varias líneas.
<code>Box</code>	Un cuadro crece desde el centro para mostrar la nueva diapositiva.
<code>Wipe</code>	Una línea limpia la pantalla para mostrar la nueva diapositiva.
<code>Dissolve</code>	La pantalla se disuelve para mostrar la nueva diapositiva.
<code>Glitter</code>	Similar a <code>Dissolve</code> , pero esta opción inicia en un lado de la pantalla.
<code>Replace</code>	La pantalla es remplazada con la nueva diapositiva. Esta es la opción predeterminada.

Cuadro 8.2: Transiciones entre diapositivas

Dentro del entorno *document*, la primera instrucción que debe aparecer es `\maketitle` si deseamos crear la portada, y la información de cada diapositiva está contenida dentro de un entorno *slide* de la siguiente manera:

```
\begin{slide}[Transición]{Título de la diapositiva}
  contenido de la diapositiva
\end{slide}
```

La *Transición* es opcional, si no la indicamos se toma la definida en `\DefaultTransition`. Con la instrucción:

```
\part[Transición]{Título de la sección}
```

se crea una diapositiva en blanco únicamente con el texto *Título de la sección*.

Para elaborar animaciones tenemos la instrucción:

```
\overlays{n}{...}
```

donde *n* es la cantidad de etapas que contendrá la diapositiva para presentar toda la información. El segundo argumento de esta instrucción es comúnmente un entorno *slide*.

Además de `\overlays` existe el entorno *itemstep* que ayuda a crear la animación. Este entorno funciona igual que *itemize*, sólo que va mostrando los ítems etapa por etapa.

Por último, tenemos las siguientes instrucciones:

```
\fromSlide{p}{Texto}
\onlySlide{p}{Texto}
\untilSlide{p}{Texto}
\FromSlide{p}
\OnlySlide{p}
\UntilSlide{p}
```

que también ayudan a construir las animaciones. Con `\fromSlide` el *Texto* aparecerá a partir de la etapa *p*, `\onlySlide` sólo muestra el *Texto* en la etapa *p* y `\untilSlide` mostrará el *Texto* hasta la etapa *p*. Las últimas tres declaraciones funcionan de igual manera que su respectiva instrucción.

Cabe mencionar que para obtener las animaciones en formato pdf, en `\documentclass` debemos utilizar la opción *pdf*; además para utilizar las últimas instrucciones que ayudan a construir la animación se deben especificar antes las etapas con `\overlays`, de otra forma sólo aparecerá una sola diapositiva.

Más información

- Otro paquete que funciona bien para crear diapositivas es *web*, sólo que debemos utilizar ADOBE ACROBAT para que funcionen todas sus opciones.
- También existen los paquetes *pdfslide* y *pdfscreen* para crear presentaciones.
- Los paquetes *pause* y *background* ayudan a crear animaciones dentro de la presentación.
- Existe el programa PPOWER4 que complementa la animación de la presentación cuando utilizamos los paquetes antes mencionados.
- Otra clase para crear presentaciones es *beamer*.

Bibliografía

- [1] T_EX User Group. <http://www.tug.org>.
- [2] CervanT_EX. <http://www.cervantex.org>.
- [3] Comprehensive T_EX Archive Network. <http://www.ctan.org>.
- [4] Bernardo Cascales Salinas, Pascual Lucas Saorín, José Manuel Mira Ros, Antonio José Pallarés Ruiz y Salvador Sánchez-Pedreño Guillén. *El libro de L^AT_EX*. Pearson Educación, 2003.
- [5] Michel Goossens, Frank Mittelbach y Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1994.
- [6] Donald Ervin Knuth. *The T_EXbook*. Addison-Wesley, 1986.
- [7] Leslie Lamport. *L^AT_EX — A document Preparation System. User's guide and manual references*. Addison-Wesley, segunda edición, 1994.
- [8] Kate L. Turabian. *A Manual for Writers of Term Papers, Theses, and Dissertations*. The University of Chicago Press, quinta edición, 1987.
- [9] Gabriel Valiente Feruglio. *Composición de textos científicos con L^AT_EX*. Alfaomega, 2001.