

Una Introducción a L^AT_EX

Manuel Maia*

Junio 12, 2004

Contenido

| | | |
|----------|--|-----------|
| 1 | Introducción a L^AT_EX | 3 |
| 1.1 | Qué es L ^A T _E X? | 3 |
| 1.2 | Un archivo fuente L ^A T _E X típico | 3 |
| 1.3 | Sucesiones y símbolos de control (comandos) | 5 |
| 2 | Produciendo documentos simples con L^AT_EX | 7 |
| 2.1 | Creando un documento fuente L ^A T _E X | 7 |
| 2.2 | Produciendo texto ordinario con L ^A T _E X | 8 |
| 2.3 | Cambiando el estilo y tamaño de las letras | 9 |
| 2.4 | Acentos | 10 |
| 2.5 | Título, autor, etc. | 11 |
| 2.6 | Comandos seccionales de L ^A T _E X | 12 |
| 2.7 | Ambientes L ^A T _E X | 12 |
| 2.7.1 | Ambiente <code>em</code> | 13 |
| 2.7.2 | Ambiente <code>center</code> | 13 |
| 2.7.3 | Ambientes <code>itemize</code> , <code>enumerate</code> , <code>description</code> | 14 |
| 2.7.4 | Ambiente <code>thebibliography</code> | 16 |
| 2.8 | Mensajes de error | 16 |
| 3 | Expresiones Matemáticas usando L^AT_EX | 17 |
| 3.1 | Modo Matemático | 18 |
| 3.2 | Desplegando una expresión | 19 |

*UCV,FACES, Escuela de Administración y Contaduría.

| | | |
|----------|--|-----------|
| 3.3 | Caracteres en modo Matemático | 20 |
| 3.4 | Símbolos Matemáticos | 20 |
| 3.4.1 | Letras Griegas | 21 |
| 3.4.2 | Letras caligráficas mayúsculas | 22 |
| 3.4.3 | Operadores binarios | 22 |
| 3.4.4 | Relaciones binarias | 23 |
| 3.4.5 | Símbolos misceláneos | 23 |
| 3.4.6 | Flechas | 24 |
| 3.4.7 | Operadores como \int y \sum | 24 |
| 3.4.8 | Acentos | 24 |
| 3.5 | Algunas estructuras matemáticas comunes | 25 |
| 3.5.1 | Subíndices y superíndices | 25 |
| 3.5.2 | Fracciones | 26 |
| 3.5.3 | Raíces | 26 |
| 3.5.4 | Puntos suspensivos | 27 |
| 3.5.5 | Texto dentro de una expresión matemática | 27 |
| 3.5.6 | Funciones predefinidas | 28 |
| 3.5.7 | Delimitadores | 29 |
| 3.5.8 | Símbolos superpuestos | 30 |
| 3.5.9 | Subrayado | 31 |
| 3.5.10 | Operadores; Sumas, Integrales, etc. | 31 |
| 3.5.11 | Matrices y otros arreglos en L ^A T _E X | 32 |
| 3.5.12 | Espaciado | 34 |
| 3.5.13 | Despliegues multilineales | 35 |
| 3.5.14 | Teoremas, Proposiciones, | 36 |
| 3.6 | A dónde debemos dirigirnos ahora? | 38 |
| 4 | Referencias Cruzadas | 38 |
| 5 | Objetos flotantes | 40 |
| 5.1 | Tablas y Figuras | 40 |
| 5.2 | Objetos Flotantes | 42 |
| 6 | Ajustes finales | 43 |

1 Introducción a L^AT_EX

1.1 Qué es L^AT_EX?

T_EX es un programa para componer documentos. Toma un archivo, preparado de acuerdo a las reglas de T_EX y lo convierte a una forma que puede ser enviada a una impresora, como una laser writer, para producir un documento impreso de alta calidad tipográfica. De hecho, las publicaciones periódicas especializadas en ciencias básicas utilizan T_EX para producir sus artículos, reportes o libros. Se pueden producir muy fácilmente documentos simples, que no contienen expresiones matemáticas o tablas: en efecto, todo lo que uno hace es escribir el texto. Componer matemáticas es algo más complicado. Sin embargo, aquí T_EX es comparativamente sencillo de usar cuando se considera la complejidad de la expresión y la gran cantidad de símbolos que hay que reproducir.

Hay varios ‘dialectos’ de T_EX todos ellos basados en la versión de T_EX creada por D. E. Knuth el cual se conoce como Plain T_EX. L^AT_EX (creado por L. B. Lamport) es uno de esos ‘dialectos’. Es particularmente adecuado para producir artículos, reportes y libros, puesto que tiene facilidades para la numeración automática de capítulos, secciones, teoremas, ecuaciones, etc., además tiene facilidades para hacer referencias cruzadas. Ciertamente es la versión más adecuada de T_EX para principiantes, y el manual “L^AT_EX—User’s Guide and Reference Manual” [2] es ciertamente más fácil de leer que el correspondiente manual (i.e., “The T_EXbook” [1]) para Plain T_EX.

Este artículo pretende ser una introducción a L^AT_EX y de ningún modo constituye un manual, pues sólo se describen los aspectos básicos de éste, dejando por fuera varios *ambientes* propios de L^AT_EX, así como su capacidad para hacer referencias cruzadas, índices, crear figuras sencillas e importar gráficos. El manual por excelencia de L^AT_EX es [2].

1.2 Un archivo fuente L^AT_EX típico

Para producir un documento con L^AT_EX, primeramente debemos crear un adecuado *archivo fuente* en el computador. Procesamos este archivo con el programa T_EX y entonces usamos un programa adicional para visualizar el archivo llamado ‘DVI’ producido por el programa T_EX o, también podemos imprimirlo. Aquí hay un ejemplo de un archivo fuente L^AT_EX típico:

```

\documentclass[12pt]{article}
\begin{document}
Los fundamentos del {\it an\'alisis} fueron establecidos
en el siglo diecinueve, principalmente por los
matem\'aticos Cauchy y Weierstrass. El objeto central
de estudio eran las definiciones formales de
{\it l\'imite} y {\it continuidad}.

Sea  $D$  un subconjunto de  $\mathbf{R}$  y sea
 $f: D \rightarrow \mathbf{R}$  una funci\'on a valores reales
en  $D$ . La funci\'on  $f$  se dice {\it continua} en  $D$  si,
para cada  $\epsilon > 0$  y para cada  $x \in D$ , existe un
 $\delta > 0$  (el cual depende de  $x$ ) tal que si
 $y \in D$  verifica,
 $|y - x| < \delta$ 
entonces
 $|f(y) - f(x)| < \epsilon$ .
No es dif\'icil verificar que si  $f$  y  $g$  son
funciones continuas en  $D$ , entonces las funciones
 $f+g$ ,  $f-g$  y  $f \cdot g$  son continuas. Si,
adem\'as  $g$  nunca vale cero, entonces  $f/g$  es
continua.
\end{document}

```

Cuando aplicamos T_EX a esos párrafos producimos el texto

Los fundamentos del *análisis* fueron establecidos en el siglo diecinueve, principalmente por los matemáticos Cauchy y Weierstrass. El objeto central de estudio eran las definiciones formales de *límite* y *continuidad*.

Sea D un subconjunto de \mathbf{R} y sea $f: D \rightarrow \mathbf{R}$ una función a valores reales en D . La función f se dice *continua* en D si, para cada $\epsilon > 0$ y para cada $x \in D$, existe un $\delta > 0$ (el cual depende de x) tal que si $y \in D$ verifica

$$|y - x| < \delta,$$

entonces

$$|f(y) - f(x)| < \epsilon.$$

No es difícil verificar que si f y g son funciones continuas en D , entonces las funciones $f + g$, $f - g$ y $f \cdot g$ son continuas. Si, además g nunca vale cero, entonces f/g es continua.

Este ejemplo ilustra varios aspectos de L^AT_EX. Note que las líneas

```

\documentclass[12pt]{article}
\begin{document}

```

están colocadas al comienzo del archivo fuente. Están seguidas por la parte principal del texto y finalmente, por:

```
\end{document}
```

Note además que, aunque la mayoría de los caracteres que aparecen tienen el significado usual, sin embargo hay caracteres tales como `\`, `$`, `{` y `}` los cuales tienen significados especiales en $\text{T}_{\text{E}}\text{X}$. Note en particular que hay sucesiones de caracteres que comienzan con un ‘backslash’ `\` las cuales son utilizadas para producir símbolos matemáticos y letras griegas, y para otros fines como cambiar las fuentes. Esas sucesiones de caracteres son conocidas como sucesiones de control, que llamaremos brevemente *comandos*.

1.3 Sucesiones y símbolos de control (comandos)

Ahora explicaremos con más detalle algunos de los aspectos de $\text{T}_{\text{E}}\text{X}$ descritos en el ejemplo anterior.

La mayoría de los caracteres del teclado, tales como letras y números, tienen el significado usual. Sin embargo los caracteres

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

son utilizados para propósitos especiales en $\text{T}_{\text{E}}\text{X}$. Escribiendo uno de esos caracteres no se producirá el carácter correspondiente en el documento final, de hecho esos caracteres son usados rara vez en el texto ordinario, y hay métodos para producirlos cuando se requieran.

Cuando escribimos en el documento fuente lo haremos de dos modos distintos. Escribiremos *texto literal* (el cual constituirá los párrafos del documento final) o escribiremos texto que será *interpretado* por $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ como una instrucción para insertar un símbolo especial o para realizar alguna acción. Es decir $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ tiene dos modos, a saber, modo *texto* y modo *matemático*.

Una sucesión de control o comando, es algo de la forma `\comando`, donde comando es una palabra compuesta únicamente de letras `a...z` y `A...Z`. Un símbolo de control consiste de un `\` seguido de un único símbolo, no una letra.

Aquí hay algunos ejemplos:

- los comandos para escribir símbolos como % y \$ son símbolos de control, estos son \% y \\$
- \LaTeX es un comando que le dice a L^AT_EX que inserte su nombre,
- \clubsuit le dice a L^AT_EX que inserte un ♣,
- \pounds inserta un símbolo £,
- \S inserta un símbolo ¶,
- \em hace que el texto sea *enfaticado*.

Otra clase de comandos la constituyen aquellos que aceptan *argumentos*. Algunos comandos necesitan un parámetro o argumento (obligatorio) que debe estar entre llaves { } después del nombre del comando. Otros, soportan argumentos opcionales que serán añadidos después del nombre del comando, en corchetes [] y antes del argumento obligatorio. Aquí hay algunos ejemplos:

- \chapter{El principio} hace que L^AT_EX comience un nuevo capítulo con el título “El principio”, numerándolo en secuencia con el resto de los capítulos, escribiendo el encabezado del capítulo en la fuente adecuada, y añadiendo una entrada en la tabla de contenido,
- \underline{palabras} produce palabras,
- \frac{a+b}{c+d} produce: $\frac{a+b}{c+d}$,
- \sqrt[5]{a+b} escribe la raíz quinta de $a + b$, : $\sqrt[5]{a + b}$. El 5 está en corchetes porque es un argumento opcional y puede ser omitido del todo (produciendo la raíz cuadrada).

Para producir un documento matemático es necesario escribir una cantidad considerable de símbolos matemáticos. Podemos además necesitar cambiar las fuentes. Además el documento quizá contenga arreglos de números o símbolos (matrices) y otras expresiones complicadas. Todas estas cosas son producidas en T_EX utilizando *sucesiones de control (comandos)*.

En el ejemplo de arriba hemos usado los comandos \it y \bf para cambiar la fuente a *itálica* y **boldface** respectivamente. Además usamos \to, \in,

`\delta` y `\epsilon` para producir los símbolos matemáticos \rightarrow y \in y las letras griegas δ y ϵ .

Los caracteres especiales `{` y `}` son utilizados para *agrupar*. Cualquier cosa encerrada en llaves es tratado como una unidad. Hemos aplicado esas llaves en el ejemplo de arriba siempre que cambiamos las fuentes. Veremos otras instancias donde necesitaremos usar `{` y `}` para agrupar palabras y símbolos juntos.

2 Produciendo documentos simples con \LaTeX

2.1 Creando un documento fuente \LaTeX

Describiremos la estructura de un archivo fuente \LaTeX típico.

La primera línea consiste del comando `\documentclass`. Los estilos básicos en \LaTeX son `letter`, `article`, `report`, y `book`. El estilo `article` se utiliza para documentos que tienen la apariencia de un artículo. El estilo `report` usualmente se utiliza para documentos más grandes que un artículo. Esos estilos realmente difieren en el tamaño de la página, fuentes, ubicación de título y autor, unidades seccionales, etc.

Hay ciertas opciones con cada estilo de documento. La sintaxis es como sigue (recuerde que un argumento en corchetes es opcional y puede ser omitido del todo, incluyendo los corchetes)

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

donde `estilo` es, por ejemplo, `report` y el argumento opcional se puede elegir de la siguiente lista:

- 11 `pt` elige 11 puntos como el tamaño de fuente, en lugar de 10 puntos (por defecto).
- 12 `pt` elige 12 puntos como el tamaño de fuente.
- `twocolumn` produce un documento con dos columnas en cada página.

De hecho hay muchas más opciones, pero no las mencionaré aquí.

El comando `\documentclass` puede ser seguido por ciertos comandos opcionales, tales como `\pagestyle` (que permite modificar el formato de numeración de las páginas) u otros, como definiciones de comandos, fuentes o ambientes nuevos. Su descripción escapa al objetivo de este artículo.

Después del comando `\documentclass` y esos otros comandos opcionales colocamos el comando

```
\begin{document}
```

Este comando será seguido por el cuerpo principal del documento en el formato prescrito por \LaTeX .

Finalmente, el documento fuente debe terminar con una línea que contenga el comando

```
\end{document}
```

2.2 Produciendo texto ordinario con \LaTeX

Para producir un documento con \LaTeX debemos crear un archivo o documento fuente (con cualquier editor de texto y guardándolo con la extensión `.tex`) como describimos en la sección anterior.

Ahora veremos un ejemplo de un documento que contiene dos párrafos (preste atención a su contenido):

```
\documentclass[12pt]{article}
\begin{document}
Si queremos simplemente producir texto ordinario,
sin expresiones matemáticas o efectos especiales
como cambios de fuentes, entonces simplemente lo
escribimos. Un espacio entre palabras
es equivalente a cualquier
número de espacios. De esta forma presionando
Return al final de una línea y comenzando una
nueva palabra en la siguiente línea solo sirve
para separar palabras.

El final de un párrafo es indicado dejando una
línea en blanco. No tenemos que preocuparnos
de la justificación de los párrafos ni de las
sangrías. Todos los párrafos serán
correctamente justificados. Se les proporcionará
sangría, a excepción del primer párrafo de
cada sección.
\end{document}
```

Producirá (compare con los párrafos del ejemplo):

Si queremos simplemente producir texto ordinario, sin expresiones matemáticas o efectos especiales como cambios de fuentes, entonces simplemente lo escribimos. Un espacio entre palabras es equivalente a cualquier número de espacios. De esta forma presionando Return al final de una línea y comenzando una nueva palabra en la siguiente línea solo sirve para separar palabras.

El final de un párrafo es indicado dejando una línea en blanco. No tenemos que preocuparnos de la justificación de los párrafos ni de las sangrías. Todos los párrafos serán correctamente justificados. Se les proporcionará sangría, a excepción del primer párrafo de cada sección.

Una vez creado el documento fuente, lo procesamos en L^AT_EX y luego visualizamos o imprimimos el archivo resultante (conocido como ‘DVI’). Se trata de un archivo con el mismo nombre del documento fuente, pero con la extensión .dvi.

2.3 Cambiando el estilo y tamaño de las letras

Hemos mencionado que existen comandos para cambiar las fuentes como `\em`, pero no hemos dicho como utilizarlos.

Para delimitar el área del texto sobre la cual queremos que estos comandos tengan efecto debemos colocar el texto en lo que se llama un *grupo*. Los grupos son de uso extensivo en L^AT_EX para lograr efectos en un área restringida, no en todo el documento. Para ello se utilizan los símbolos especiales `{` y `}`. Los comandos L^AT_EX para cambiar de estilo están en la Tabla 1 y los comandos para cambiar el tamaño de las fuentes están en la Tabla 2. El estilo por defecto es Roman y el tamaño por defecto es normalsize.

| | | | | | |
|------------------|-------------------|------------------|----------------|------------------|------------|
| <code>\rm</code> | Roman | <code>\it</code> | <i>italic</i> | <code>\sc</code> | CAPITALS |
| <code>\em</code> | <i>Emphasised</i> | <code>\sl</code> | <i>slanted</i> | <code>\tt</code> | typewriter |
| <code>\bf</code> | boldface | <code>\sf</code> | sans serif | | |

Tabla 1: Comandos para seleccionar estilos

Mencionamos que para restringir el alcance de un comando que cambia el tipo de fuente debemos colocar el texto afectado en un grupo. Veamos un ejemplo:

Cuando queremos `{\em enfatizar}` algo de texto usamos el

| | | | |
|----------------------------|---------------------------|---------------------|-------|
| <code>\tiny</code> | <code>tiny</code> | <code>\large</code> | large |
| <code>\scriptsize</code> | <code>scriptsize</code> | <code>\Large</code> | Large |
| <code>\footnotesize</code> | <code>footnotesize</code> | <code>\LARGE</code> | LARGE |
| <code>\small</code> | <code>small</code> | <code>\huge</code> | huge |
| <code>\normalsize</code> | <code>normalsize</code> | <code>\Huge</code> | Huge |

Tabla 2: Comandos para seleccionar tamaños

comando `{\tt em}`, y agrupamos para restringir su alcance. Podemos cambiar el `{\large tama~no de la fuente}` de la misma manera. Podemos obtener `{\it it\’alica}`, `{\bf negrita}`, `{\sc may\’usculas peque~nas}`, etc.

Produce

Cuando queremos *enfatizar* algo de texto usamos el comando `em`, y agrupamos para restringir su alcance. Podemos cambiar el tamaño de la fuente de la misma manera. Podemos obtener *itálica*, **negrita**, MAYÚSCULAS PEQUEÑAS, etc.

2.4 Acentos

Hay una variedad de comandos para producir acentos, si ha leído con atención habrá observado algunos de ellos en los ejemplos. Por ejemplo `\’o` produce una letra *o* acentuada.

De esta forma, escribiendo

```
Enonc\’e du th\’eor\’eme
```

producimos

Enoncé du théorème

Similarmente usamos el comando `\c{c}` para producir la cedilla en ‘Curaçao’ y `\"a` para producir la diéresis en ‘Universität’. La lista de acentos está en la Tabla 3.

Se requiere especial cuidado cuando acentuamos una *i* o una *j*, ellas deben perder sus puntos. Para esto usamos los comandos `\i` y `\j`, que producen *i* y *j* respectivamente. Por ejemplo, `Geometr\’\i` a produce Geometría. El

| | | |
|---------------------|----|--|
| <code>\'e</code> | é | e.g., <code>matem\'atica</code> produce ‘matemática’ |
| <code>\'e</code> | è | e.g., <code>alg\'ebre</code> produce ‘algèbre’ |
| <code>\^e</code> | ê | e.g., <code>h\^ote</code> produce ‘hôte’ |
| <code>\"o</code> | ö | e.g., <code>H\"older</code> produce ‘Hölder’ |
| <code>\~n</code> | ñ | e.g., <code>ma\~nana</code> produce ‘mañana’ |
| <code>\=o</code> | ō | |
| <code>\.o</code> | ò | |
| <code>\u{o}</code> | ů | |
| <code>\v{c}</code> | č | e.g., <code>\v{C}ech</code> produce ‘Čech’ |
| <code>\H{o}</code> | ő | |
| <code>\t{oo}</code> | őo | |
| <code>\c{c}</code> | ç | e.g., <code>gar\c{c}on</code> produce ‘garçon’ |
| <code>\d{o}</code> | ð | |
| <code>\b{o}</code> | ö | |

Tabla 3: Comandos para acentos

espacio después de la *i* es indispensable, pues de otra manera L^AT_EX interpretaría `\ia` como una sucesión de control, que al no estar definida, produciría un error.

2.5 Título, autor, etc.

Declaramos la información de título y autor usando los comandos `\title` y `\author`, cada uno de los cuales acepta un argumento, al igual que `\abstract`. Varios autores pueden ser listados en el argumento de `\title`, separados por `\and`'s. El comando `\date` puede ser utilizado para fechar un documento. *Después de haber declarado cada uno de ellos, escribimos el comando `\maketitle`*. El estilo determinará la ubicación y las fuentes.

```
\title{La Topolog\'i a de Zariski}
\author{Manuel Maia \and Michael Atiyah}
\date{Noviembre 1992}
\maketitle
```

producirá

La Topología de Zariski
Manuel Maia Michael Atiyah

Noviembre 1992

Estas líneas de comandos deben aparecer *después* de `\begin{document}` y *antes* de cualquier otro material impreso.

Si queremos que \LaTeX cree una tabla de contenido, debemos escribir el comando `\tableofcontents` en el punto donde queremos que ésta aparezca. Observe la tabla de contenido de este artículo, no la hice yo, la hizo \LaTeX . *Para que la tabla aparezca es necesario procesar tres (3) veces con \LaTeX el archivo fuente.*

2.6 Comandos seccionales de \LaTeX

Como parte de nuestra tarea de descripción de la estructura lógica del documento, debemos indicarle a \LaTeX dónde comienzan las unidades seccionales. Para hacer esto usamos los comandos mostrados en la Tabla 4. Cada co-

| | | |
|----------------------------|-------------------------------|----------------------------------|
| <code>\part{...}</code> | <code>\section{...}</code> | <code>\subsubsection{...}</code> |
| <code>\chapter{...}</code> | <code>\subsection{...}</code> | <code>\appendix</code> |

Tabla 4: Comandos seccionales de \LaTeX

mando seccional acepta un argumento simple—el título de la sección. La excepción la constituye `\appendix`, que no lleva argumento, pues lo que hace es cambiar números por letras. Los comandos `\part` y `\chapter` son de uso exclusivo con los estilos `report` y `book`. \LaTeX proporciona la numeración, de manera que no tenemos que incluir ningún número en el argumento. \LaTeX tendrá cuidado además, de espaciar correctamente los párrafos y títulos de secciones, y colocar una fuente adecuada. Además comenzará en una nueva página, si comienza un nuevo capítulo.

El título de esta sección fue producido con

```
\subsection{Comandos Seccionales de \LaTeX }
```

2.7 Ambientes \LaTeX

Quizá el concepto más poderoso en la sintaxis de \LaTeX sea el de *ambiente*.

Algunos ambientes son utilizados para desplegar una porción de texto o matemática y destacarlos.

Todos los ambientes comienzan con `\begin{nombre}` y deben finalizar con `\end{nombre}`, donde `nombre` es el nombre del ambiente. Estos son los delimitadores del grupo. Mencionaremos sólo algunos de estos ambientes.

2.7.1 Ambiente `em`

Comenzaremos con un ambiente simple, que proporciona una alternativa al comando `\em`.

```
\begin{em}
  Debemos tener cuidado de emparejar
  correctamente los delimitadores de un
  grupo. Si no est\'an bien emparejados,
  entonces \LaTeX\ tendr\'a una confusi\'on,
  en efecto, no sabr\'a d\'onde comienza o
  termina un grupo
\end{em}
```

Produce

Debemos tener cuidado de emparejar correctamente los delimitadores de grupo. Si no están bien emparejados, entonces L^AT_EX tendrá una confusión, en efecto, no sabrá dónde comienza o termina un grupo.

No importa cómo esté formateado el documento fuente. Sin embargo es una buena idea construirlo de una manera lógica y legible. Esto ayuda a minimizar errores y a encontrarlos fácilmente. Por esta razón coloqué los delimitadores `\begin` y `\end` alineados.

2.7.2 Ambiente `center`

Este ambiente permite centrar líneas consecutivas de texto. Las líneas nuevas serán indicadas con un `\`

Veamos un ejemplo:

```
El ambiente {\tt center} toma cuidado del espacio vertical antes
y despu'es de \'el, de manera que no nos preocuparemos de eso,
\begin{center}
  si no dejamos una l\'i nea en blanco despu'es del\\
  ambiente {\tt center},\\
  entonces \LaTeX\ asumir\'a que este no\\
  es parte del texto de un nuevo p\'arrafo\\
  y por tanto no le pondr\'a sangr\'i a.
```

```
\end{center}
```

En este caso dejamos una línea en blanco, para empezar un párrafo nuevo.

produce el siguiente texto

El ambiente `center` toma cuidado del espacio vertical antes y después de él, de manera que no nos preocuparemos de eso,

```
si no dejamos una línea en blanco después del
ambiente center,
entonces LATEX asumirá que este no
es parte del texto de un nuevo párrafo
y por tanto no le pondrá sangría.
```

En este caso dejamos una línea en blanco, para empezar un párrafo nuevo.

2.7.3 Ambientes `itemize`, `enumerate`, `description`

L^AT_EX proporciona tres ambientes para producir listas, a saber: `itemize`, `enumerate`, `description`.

El único comando nuevo es `\item`, que indica el comienzo de un nuevo ítem y el final del anterior. Este comando acepta un argumento opcional (lo que significa que debemos encerrarlo en corchetes) que puede ser usado para ponerle una etiqueta al ítem. Si no se le pone el argumento opcional, L^AT_EX se lo pondrá.

```
\begin{itemize}
\item Un ítem comienza con \verb@\item@.
\item Si no especificamos la etiqueta, LATEX pondrá una.
\item Colocamos los ítems en una columna sólo
para hacer legible el documento fuente.
\item Una línea en blanco entre ítems será
ignorada por LATEX.
\end{itemize}
```

produce

- Un ítem comienza con `\item`.
- Si no especificamos la etiqueta, L^AT_EX pondrá una.
- Colocamos los ítems en una columna sólo para hacer legible el documento fuente.

- Una línea en blanco entre ítems será ignorada por \LaTeX .

Podemos poner listas dentro de listas (o grupos dentro de grupos). Veamos esto con la siguiente lista, que muestra además el uso del ambiente `enumerate`.

Para crear un artículo con \LaTeX hacemos lo siguiente:

```
\begin{enumerate}
\item Crear un documento fuente con el siguiente formato:
\begin{itemize}
\item Comienza con la línea
\verb@\documentstyle{article}@,
\item Sigue con \verb@\begin{document}@,
\item Escribimos el cuerpo del documento con
las reglas prescritas por  $\LaTeX$ .
\item Finaliza con \verb@\end{document}@.
\end{itemize}
\item Guardamos el documento fuente con la extensión {\tt .tex}.
\item Lo procesamos con  $\LaTeX$ .
\item El archivo resultante lo visualizamos
y/o imprimimos con el software apropiado para ello.
\end{enumerate}
```

producirá la siguiente lista

Para crear un artículo con \LaTeX hacemos lo siguiente:

1. Crear un documento fuente con el siguiente formato:
 - Comienza con la línea `\documentstyle{article}`,
 - Sigue con `\begin{document}`,
 - Escribimos el cuerpo del documento con las reglas prescritas por \LaTeX .
 - Finaliza con `\end{document}`.
2. Guardamos el documento fuente con la extensión `.tex`.
3. Lo procesamos con \LaTeX ,
4. El archivo resultante lo visualizamos y/o imprimimos con el software apropiado para ello.

Observe que dejé el documento fuente con un aspecto legible (alineando los ítems). Esto me ayuda a mí, no a \LaTeX . Por otra parte, trate de interpretar el uso del comando `\verb`.

El ambiente `description` es para hacer listas de descripciones. Entre los corchetes ponemos lo que deseamos.

```

\begin{description}
  \item[\tt itemize] un ambiente para hacer listas itemizadas,
  \item[\tt enumerate] un ambiente para hacer listas numeradas,
  \item[\tt description] un ambiente para hacer listas con
                        descripciones.
\end{description}

```

producirá la siguiente lista

```

itemize un ambiente para hacer listas itemizadas,
enumerate un ambiente para hacer listas numeradas,
description un ambiente para hacer listas con descripciones.

```

2.7.4 Ambiente thebibliography

Con el ambiente `thebibliography` podemos crear una bibliografía. Será titulada como ‘Bibliography’, si usamos el estilo `report` o `book`, o ‘References’, si usamos el estilo `article`. Más adelante veremos qué hacer para que sea titulada ‘Bibliografía’ o ‘Referencias’.

```

\begin{thebibliography}{99}
  \item Donald Ervin Knuth. {\em The \TeX book},
      Addison-Wesley Pub Co, 1986.
  \item Leslie Lamport.
      {\em \LaTeX\ : User's Guide and Reference Manual},
      Addison-Wesley Pub Co, 1994.
  \item Gavin Maltby. {\em An introduction to \TeX\ and friends},\@
      \verb@ftp:\@ftp.dante.de\tex-archive\documentation\maltby.tex@
  \item D. R. Wilkins. {\em Getting Started with \LaTeX},\@
      \verb@ftp:\@ftp.dante.de\tex-archive\documentation\tcd-manual\@
\end{thebibliography}

```

produjo las referencias de este artículo.

El resto de los ambientes predefinidos en \LaTeX (en modo `texto`) son: `quote`, `quotation`, `verse`, `flushright`, `flushleft`, `verbatim`, `tabbing`, `tabular`, `figure`, `picture` y `table`.

2.8 Mensajes de error

Hasta ahora hemos supuesto que trabajamos cuidadosamente preparando un documento. Sin embargo, si escribimos incorrectamente un comando u olvidamos cerrar un ambiente correctamente, \TeX se detendrá dándonos un

mensaje de error cuando procesemos el documento fuente. Que no aparezca `\end{ambiente}` puede causar gran confusión debido a que \LaTeX tratará de colocar material en un ambiente que no está diseñado para ser colocado en ese lugar. Además, omitir un argumento obligatorio causa invariablemente un error. Supongamos que escribimos `\bold` en lugar de `\bf` en la siguiente línea:

```
esto va a producir un {\bold gran} error.
```

Producirá el siguiente mensaje de error:

```
! Undefined control sequence.
1.506 esto va a producir un {\bold
                                gran} error.^^M
?
```

La línea que comienza con `!` nos dice que hemos intentado usar un comando (control sequence) desconocido para \LaTeX , el `1.506` nos dice que el error ocurrió en la línea 506 del documento fuente, y el mensaje de error se parte en dos líneas, con el salto donde \LaTeX detectó el problema. El signo `?` significa que \LaTeX espera una respuesta. Una de las respuestas posibles es `x` [Enter]; nos sacará del problema, volvemos al documento fuente, lo corregimos e intentamos de nuevo.

La mayoría de los errores pueden ser evitados preparando *cuidadosamente* el documento fuente. Escribir los comandos con la correcta sintaxis es un buen comienzo, pero también es conveniente:

1. Escribir el documento fuente de una manera ordenada. Vea los ejemplos de los ambientes.
2. Emparejar correctamente los delimitadores de grupos, esto es, llaves `{}` y `\begin{...}\end{...}`. Es un buen hábito escribir el par y luego escribir entre ellos.
3. No olvidar los argumentos de los comandos cuando son obligatorios. Por ejemplo `\section`, obviamente espera que le digan algo más.

3 Expresiones Matemáticas usando \LaTeX

En los documentos que contienen sólo texto vimos que nuestra tarea era describir los componentes lógicos de cada oración, párrafo, sección, etc. Cuando

le digamos a \LaTeX que trabaje en *modo matemático*, tenemos que describir las partes lógicas de una fórmula, matriz, operador, símbolo especial, etc. \TeX está diseñado para reconocer estos aspectos, nosotros se los describimos y él hace el resto, colocar el tipo de letra y espaciado correcto.

3.1 Modo Matemático

El símbolo $\$$ está especialmente reservado como el símbolo de cambio a modo matemático. Cuando \TeX encuentra el símbolo $\$$ cambia a modo matemático y regresa al modo normal (texto) cuando encuentra el siguiente símbolo $\$$.

Cuando una expresión matemática aparece dentro del texto de un párrafo, debemos colocar un símbolo $\$$ antes y después de la expresión. De esta forma, para obtener una expresión como

Sea f la función definida por $f(x) = 3x + a$, donde a es un número real positivo.

debemos escribir

Sea f la función definida por $f(x)=3x+a$, donde a es un número real positivo.

En particular, note que aún cuando una expresión matemática consista de un único carácter, como f y a en el ejemplo anterior, debe ser colocado entre símbolos $\$$. Esto es para asegurar que aparezcan en tipo itálico como se acostumbra en la tipografía de documentos matemáticos. Hay además dos aspectos importantes que debemos tener en cuenta cuando escribimos expresiones matemáticas:

1. Todas las letras que no sean parte de un argumento de algún comando serán colocadas en itálica. Escribiendo $f(x)>0$ para $x>1$ se producirá

$$f(x) > 0 \text{ para } x > 1$$

en lugar de la expresión correcta

$$f(x) > 0 \text{ para } x > 1$$

producida con $f(x)>0$ para $x>1$.

2. \LaTeX ignora, sin excepción, todos los espacios que coloquemos en modo matemático. Por lo tanto, escribiendo ‘la constante@’ se producirá ‘la constante’. Lo correcto sería escribir ‘la constante a ’.

3.2 Desplegando una expresión

L^AT_EX considera una expresión $\$. . \$$ igual que una palabra, en el sentido que la considera elegible para separarla en dos líneas (pero sin guión, de hecho). L^AT_EX trata de evitar esto. Cuando no es posible, busca el lugar adecuado para separar la expresión. Sin embargo hay expresiones que son muy largas para caber dentro del texto de un párrafo. Estas se ven mejor ‘desplegadas’. Además, algunas expresiones son tan importantes que deben ser separadas del texto común.

Para obtener una expresión matemática desplegada o una ecuación en una línea, colocamos $\[$ antes y $\]$ después de la expresión.

Para obtener

El producto de dos polinomios de primer grado es un polinomio cuadrático. Por ejemplo, si $f(x) = 3x + 7$ y $g(x) = x + 4$, entonces

$$f(x)g(x) = 3x^2 + 19x + 28.$$

El recíproco no ocurre para polinomios sobre el cuerpo de los números reales.

debemos escribir

```
El producto de dos polinomios de primer grado es un polinomio
cuadr\'atico. Por ejemplo, si $f(x)=3x+7$ y $g(x)=x+4$, entonces
\[ f(x)g(x)=3x^2+19x+28. \]
El rec\'i proco no ocurre para polinomios sobre el cuerpo
de los n\'umeros reales.
```

También es posible usar $$$$ en lugar de $\[$ y $\]$ para marcar el principio y el fin de una expresión desplegada. Así

$$f(x)g(x) = 3x^2 + 19x + 28.$$

puede ser producida escribiendo $$$f(x)g(x)=3x^2+19x+28.$$$

De hecho este es el método de producir expresiones matemáticas desplegadas en otros dialectos de T_EX como Plain T_EX y $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX.

L^AT_EX provee facilidades para la numeración automática de expresiones desplegadas. Si queremos una expresión matemática numerada debemos usar $\begin{equation}$ y $\end{equation}$ en lugar de $\[$ y $\]$. De esta forma

```
El producto de dos polinomios de primer grado es un polinomio
cuadr\'atico. Por ejemplo, si $f(x)=3x+7$ y $g(x)=x+4$, entonces
\begin{equation}
f(x)g(x)=3x^2+19x+28.
\end{equation}
```

produce

El producto de dos polinomios de primer grado es un polinomio cuadrático. Por ejemplo, si $f(x) = 3x + 7$ y $g(x) = x + 4$, entonces

$$f(x)g(x) = 3x^2 + 19x + 28. \quad (1)$$

Por otra parte, \LaTeX tiene la habilidad de manejar referencias cruzadas. Puedo referirme a la ecuación (1) como estoy haciendo ahora, independientemente del número que esta posea en una etapa cualquiera de la creación del documento. Si desea saber cómo hacer esto, consulte [2].

3.3 Caracteres en modo Matemático

Todos los caracteres del teclado tienen su significado usual en modo matemático, con la excepción de los caracteres

`# $ % & ~ _ ^ \ { } ’`

Las letras serán colocadas en itálica. En modo matemático el caracter `’` tiene un significado especial: escribiendo `$f’+g’$`, se producirá $f' + g'$.

Para obtener los caracteres

`# $ % & _ { }`

en modo matemático, debemos escribir

`\# \$ \% \& _ \{ \}` .

Para obtener `\` en modo matemático, escribimos `\backslashash`.

3.4 Símbolos Matemáticos

\LaTeX pone todos los símbolos esotéricos de la matemática a nuestra disposición. A ellos nos referiremos por nombre, siendo la nominación perfectamente lógica y sistemática. Ninguno de los comandos para acceder esos símbolos acepta un argumento.

Veremos además que algunos de los símbolos tienen un aspecto diferente dependiendo de dónde están siendo colocados. Por ejemplo, cuando yo escribo $\sum_{i=1}^n a_i$ dentro de una línea de un párrafo los límites están colocados de manera diferente a cuando escribo esa expresión desplegada:

$$\sum_{i=1}^n a_i.$$

Es importante notar que estos símbolos están disponibles sólo en modo matemático.

3.4.1 Letras Griegas

Las letras griegas se producen en modo matemático precediendo el nombre de la letra por un backslash `\`. Así, las letras griegas alpha (α), pi (π) y chi (χ) se obtienen escribiendo `\alpha`, `\pi` y `\chi` respectivamente.

| | | | | | | | |
|------------|-----------------------|---------------|--------------------------|----------|---------------------|------------|-----------------------|
| α | <code>\alpha</code> | β | <code>\beta</code> | γ | <code>\gamma</code> | δ | <code>\delta</code> |
| ϵ | <code>\epsilon</code> | ε | <code>\varepsilon</code> | ζ | <code>\zeta</code> | η | <code>\eta</code> |
| θ | <code>\theta</code> | ϑ | <code>\vartheta</code> | ι | <code>\iota</code> | κ | <code>\kappa</code> |
| λ | <code>\lambda</code> | μ | <code>\mu</code> | ν | <code>\nu</code> | ξ | <code>\xi</code> |
| π | <code>\pi</code> | ϖ | <code>\varpi</code> | ρ | <code>\rho</code> | ϱ | <code>\varrho</code> |
| σ | <code>\sigma</code> | ς | <code>\varsigma</code> | τ | <code>\tau</code> | υ | <code>\upsilon</code> |
| ϕ | <code>\phi</code> | φ | <code>\varphi</code> | χ | <code>\chi</code> | ψ | <code>\psi</code> |
| ω | <code>\omega</code> | | | | | | |

Tabla 5: Letras griegas minúsculas

| | | | | | | | |
|----------|---------------------|----------|---------------------|----------|---------------------|------------|-----------------------|
| Γ | <code>\Gamma</code> | Δ | <code>\Delta</code> | Θ | <code>\Theta</code> | Λ | <code>\Lambda</code> |
| Ξ | <code>\Xi</code> | Π | <code>\Pi</code> | Σ | <code>\Sigma</code> | Υ | <code>\Upsilon</code> |
| Φ | <code>\Phi</code> | Ψ | <code>\Psi</code> | Ω | <code>\Omega</code> | | |

Tabla 6: Letras griegas mayúsculas

El area A de un círculo de radio r está dado por $A = \pi r^2$.

se obtiene con

El area A de un círculo de radio r está dado por $A = \pi r^2$.

Las mayúsculas se obtienen colocando el primer caracter del nombre en mayúscula. No hay comando especial para omicron, sólo use `o`. Los comandos para obtener letras griegas están en las tablas 5 y 6. Note que hay *variantes* de algunas letras. Para hacer las mayúsculas que no están en la tabla 6 vea la sección 3.5.6.

3.4.2 Letras caligráficas mayúsculas

Las letras $\mathcal{A} \dots \mathcal{Z}$ están disponibles por medio del comando de cambio de estilo `\cal`. Este comando funciona igual que los comandos para cambiar estilos como `\em`, `\it`, etc., su alcance debe ser delimitado junto con ellos. Podemos escribir

... para el filtro `\cal F` tenemos que `\varphi(\cal F) = \cal G`.

para obtener

... para el filtro \mathcal{F} tenemos que $\varphi(\mathcal{F}) = \mathcal{G}$.

La lista completa es:

ABCDEFGHIJKLMNÓPQRSTUVWXYZ

3.4.3 Operadores binarios

\LaTeX está diseñado para reconocer operadores binarios y colocar el espacio apropiado a ambos lados de un operador. La tabla 7 muestra los operadores binarios disponibles vía comandos \LaTeX (los operadores $+$, $-$ y $*$ se pueden escribir desde el teclado en modo matemático).

| | | | | | | | |
|------------|-----------------------|-------------|------------------------|------------------|-------------------------------|------------|-----------------------|
| \pm | <code>\pm</code> | \cap | <code>\cap</code> | \diamond | <code>\diamond</code> | \oplus | <code>\oplus</code> |
| \mp | <code>\mp</code> | \cup | <code>\cup</code> | \triangleup | <code>\bigtriangleup</code> | \ominus | <code>\ominus</code> |
| \times | <code>\times</code> | \uplus | <code>\uplus</code> | \triangledown | <code>\bigtriangledown</code> | \otimes | <code>\otimes</code> |
| \div | <code>\div</code> | \sqcap | <code>\sqcap</code> | \triangleleft | <code>\triangleleft</code> | \oslash | <code>\oslash</code> |
| $*$ | <code>\ast</code> | \sqcup | <code>\sqcup</code> | \triangleright | <code>\triangleright</code> | \odot | <code>\odot</code> |
| \star | <code>\star</code> | \vee | <code>\vee</code> | \wedge | <code>\wedge</code> | \bigcirc | <code>\bigcirc</code> |
| \dagger | <code>\dagger</code> | \setminus | <code>\setminus</code> | \amalg | <code>\amalg</code> | \circ | <code>\circ</code> |
| \ddagger | <code>\ddagger</code> | \cdot | <code>\cdot</code> | \wr | <code>\wr</code> | \bullet | <code>\bullet</code> |

Tabla 7: Símbolos de operaciones binarias

| <i>Escriba</i> | <i>Para producir</i> |
|--|---|
| <code>\\$a+b\\$</code> | $a + b$ |
| <code>\\$(a+b) \ otimes c\\$</code> | $(a + b) \otimes c$ |
| <code>\\$(a \vee b) \wedge c\\$</code> | $(a \vee b) \wedge c$ |
| <code>\\$X - (A \cap B) = (X-A) \cup (X-B)\\$</code> | $X - (A \cap B) = (X - A) \cup (X - B)$ |

3.4.4 Relaciones binarias

L^AT_EX reconoce también el uso de relaciones binarias. La tabla 8 muestra las que son accesibles con comandos. El resto se pueden obtener desde el teclado (en modo matemático): $<$, $>$, $=$ y $|$. Para negar un símbolo, precedemos el comando correspondiente al símbolo con el comando `\not`. Por ejemplo, `\not\subset` produce $\not\subset$.

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

Tabla 8: Relaciones binarias

3.4.5 Símbolos misceláneos

La tabla 9 muestra varios símbolos de propósito general. Recuerde que están disponibles sólo en modo matemático. Para obtener un símbolo ‘prima’ podemos usar `\prime` o solamente escribir `'` en modo matemático, como en `\f''(x)=x`, que produce $f''(x) = x$.

| | | | | | | | |
|----------|---------------------|-------------|------------------------|--------------|-------------------------|----------------|---------------------------|
| \aleph | <code>\aleph</code> | \prime | <code>\prime</code> | \forall | <code>\forall</code> | ∞ | <code>\infty</code> |
| \hbar | <code>\hbar</code> | \emptyset | <code>\emptyset</code> | \exists | <code>\exists</code> | \Box | <code>\Box</code> |
| \imath | <code>\imath</code> | ∇ | <code>\nabla</code> | \neg | <code>\neg</code> | \triangle | <code>\triangle</code> |
| \jmath | <code>\jmath</code> | \surd | <code>\surd</code> | \flat | <code>\flat</code> | \triangle | <code>\triangle</code> |
| ℓ | <code>\ell</code> | \top | <code>\top</code> | \natural | <code>\natural</code> | \clubsuit | <code>\clubsuit</code> |
| \wp | <code>\wp</code> | \perp | <code>\perp</code> | \sharp | <code>\sharp</code> | \diamondsuit | <code>\diamondsuit</code> |
| \Re | <code>\Re</code> | \parallel | <code>\parallel</code> | \backslash | <code>\backslash</code> | \heartsuit | <code>\heartsuit</code> |
| \Im | <code>\Im</code> | \angle | <code>\angle</code> | ∂ | <code>\partial</code> | \spadesuit | <code>\spadesuit</code> |
| \mho | <code>\mho</code> | | | | | | |

Tabla 9: Símbolos misceláneos

3.4.6 Flechas

La tabla 10 muestra la multitud de flechas que L^AT_EX tiene disponible. Por ejemplo $\$f:X \rightarrow Y\$$ produce $f : X \longrightarrow Y$.

| | | | | | |
|----------------------|---------------------------------|------------------------|-----------------------------------|----------------|---------------------------|
| \leftarrow | <code>\leftarrow</code> | \longleftarrow | <code>\longleftarrow</code> | \uparrow | <code>\uparrow</code> |
| \Leftarrow | <code>\Leftarrow</code> | \Lleftarrow | <code>\Lleftarrow</code> | \Uparrow | <code>\Uparrow</code> |
| \rightarrow | <code>\rightarrow</code> | \longrightarrow | <code>\longrightarrow</code> | \downarrow | <code>\downarrow</code> |
| \Rightarrow | <code>\Rightarrow</code> | \Longrightarrow | <code>\Longrightarrow</code> | \Downarrow | <code>\Downarrow</code> |
| \leftrightarrow | <code>\leftrightarrow</code> | \longleftrightarrow | <code>\longleftrightarrow</code> | \updownarrow | <code>\updownarrow</code> |
| \Leftrightarrow | <code>\Leftrightarrow</code> | \Llongleftrightarrow | <code>\Llongleftrightarrow</code> | \Updownarrow | <code>\Updownarrow</code> |
| \mapsto | <code>\mapsto</code> | \longmapsto | <code>\longmapsto</code> | \nearrow | <code>\nearrow</code> |
| \hookrightarrow | <code>\hookrightarrow</code> | \hookleftarrow | <code>\hookleftarrow</code> | \searrow | <code>\searrow</code> |
| \lleftarrow | <code>\lleftarrow</code> | \rightharpoonup | <code>\rightharpoonup</code> | \swarrow | <code>\swarrow</code> |
| \lleftarrow | <code>\lleftarrow</code> | \rightharpoonup | <code>\rightharpoonup</code> | \nwarrow | <code>\nwarrow</code> |
| \rightleftharpoons | <code>\rightleftharpoons</code> | \leadsto | <code>\leadsto</code> | | |

Tabla 10: Flechas

3.4.7 Operadores como \int y \sum

Esta clase de operadores tienen una apariencia diferente cuando son usados en una línea de texto y cuando son ‘desplegados’. En el texto aparecerán en su forma pequeña y seguidos de los límites, como en $\sum_{i=1}^n a_i$. ‘Desplegados’ aparecerán en su forma grande y con los límites arriba y abajo, como en

$$\sum_{i=1}^n a_i.$$

La tabla 11 muestra esta clase de operadores, en su forma pequeña (texto) y en su forma grande (desplegada). En la sección 3.5.10 aprenderemos a colocarle los límites a esos operadores.

3.4.8 Acentos

Los comandos que aprendimos para acentuar en modo texto no se aplican en modo matemático. Consulte la tabla 12 para ver cómo acentuar un símbolo (en este caso el símbolo u) en modo matemático. Recuerde que i y j deben perder sus puntos cuando se acentúan, en este caso debemos usar `\imath` y `\jmath` vea la tabla 9.

| | | | | | |
|-------------------|----------------------|--------------------|------------------------|----------------------|-------------------------|
| $\Sigma \sum$ | <code>\sum</code> | $\cap \bigcap$ | <code>\bigcap</code> | $\odot \bigodot$ | <code>\bigodot</code> |
| $\prod \prod$ | <code>\prod</code> | $\cup \bigcup$ | <code>\bigcup</code> | $\otimes \bigotimes$ | <code>\bigotimes</code> |
| $\coprod \coprod$ | <code>\coprod</code> | $\sqcup \bigsqcup$ | <code>\bigsqcup</code> | $\oplus \bigoplus$ | <code>\bigoplus</code> |
| $\int \int$ | <code>\int</code> | $\vee \bigvee$ | <code>\bigvee</code> | $\uplus \biguplus$ | <code>\biguplus</code> |
| $\oint \oint$ | <code>\oint</code> | $\wedge \bigwedge$ | <code>\bigwedge</code> | | |

Tabla 11: Símbolos de tamaño variable

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

Tabla 12: Acentos matemáticos

\LaTeX proporciona además, acentos de tamaño variable; por ejemplo, `\widehat`, `\widetilde` y `\overrightarrow`:

| <i>Escriba</i> | <i>Para producir</i> |
|--|--|
| <code>\widehat x</code> , <code>\widetilde x</code> | \hat{x} , \tilde{x} |
| <code>\widehat{xy}</code> , <code>\widetilde{xy}</code> | \widehat{xy} , \widetilde{xy} |
| <code>\widehat{xyz}</code> , <code>\widetilde{xyz}</code> | \widehat{xyz} , \widetilde{xyz} |
| <code>\overrightarrow{AB}</code> , <code>\overrightarrow{ABC}</code> | \overrightarrow{AB} , \overrightarrow{ABC} |

Los comandos tales como `\'` y `\"`, utilizados para producir acentos en el texto ordinario, no pueden ser usados en modo matemático.

3.5 Algunas estructuras matemáticas comunes

En esta sección aprenderemos cómo manipular todos los símbolos listados en la sección 3.4. De hecho, al final de la sección seremos capaces de escribir expresiones verdaderamente complicadas.

3.5.1 Subíndices y superíndices

Subíndices y superíndices son obtenidos utilizando los caracteres especiales `_` y `^` respectivamente. Así, la expresión $t^3 + x_1^2 - x_2$ se obtiene escribiendo `\$t^3+x_1^2-x_2\$`. Cuando el índice consiste de más de un carácter, entonces los caracteres envueltos deben ser encerrados en llaves. Así, para obtener $u_{i,j}^{12}$, debemos escribir `\$u_{i,j}^{12}\$`.

| <i>Escriba</i> | <i>Para producir</i> |
|---------------------------------------|------------------------|
| <code>\$x^{y^z}\$</code> | x^{y^z} |
| <code>\$2^{2^{2^{\aleph_0}}}\$</code> | $2^{2^{2^{\aleph_0}}}$ |
| <code>\$\$\Pi_1(X, x_0)\$</code> | $\Pi_1(X, x_0)$ |
| <code>\$\$A_{m+n}\neq A_m+n\$</code> | $A_{m+n} \neq A_m + n$ |
| <code>\$\$\Gamma^{z_c^d}\$</code> | $\Gamma^{z_c^d}$ |

3.5.2 Fracciones

L^AT_EX proporciona el comando `\frac` que acepta dos argumentos:

`\frac{numerador}{denominador}`

Igual que los operadores como \int y \sum las fracciones serán pequeñas cuando estén en una línea de texto y grandes cuando estén desplegadas.

| <i>Escriba</i> | <i>Para producir</i> |
|--|---------------------------------|
| <code>\$\$\frac{x+1}{x+2}\$</code> | $\frac{x+1}{x+2}$ |
| <code>\$\$\frac{1}{x^2+1}\$</code> | $\frac{1}{x^2+1}$ |
| <code>\$\$\frac{1+x^2}{x^2+y^2} + x^2 y\$</code> | $\frac{1+x^2}{x^2+y^2} + x^2 y$ |
| <code>\$\$\frac{1}{1 + \frac{x}{2}}\$</code> | $\frac{1}{1 + \frac{x}{2}}$ |
| <code>\$\$\frac{1}{1+x/2}\$</code> | $\frac{1}{1+x/2}$ |

3.5.3 Raíces

El comando `\sqrt` acepta dos argumentos. El primero, y opcional, especifica el orden de la raíz si es otra distinta de la raíz cuadrada. El segundo, y obligatorio, especifica la expresión que encierra el signo de la raíz.

| <i>Escriba</i> | <i>Para producir</i> |
|--|---|
| <code>\$\$\sqrt{a+b}\$</code> | $\sqrt{a+b}$ |
| <code>\$\$\sqrt[5]{a+b}\$</code> | $\sqrt[5]{a+b}$ |
| <code>\$\$\sqrt[n]{\frac{1+x}{1+x^2}}\$</code> | $\sqrt[n]{\frac{1+x}{1+x^2}}$ |
| <code>\$\$\frac{\sqrt{x+1}}{\sqrt[3]{x^3+1}}\$</code> | $\frac{\sqrt{x+1}}{\sqrt[3]{x^3+1}}$ |
| <code>\$\$\frac{\sqrt[\alpha]{x^{\beta+1}}}{\sqrt{x^{\beta-1}+1}}\$</code> | $\frac{\sqrt[\alpha]{x^{\beta+1}}}{\sqrt{x^{\beta-1}+1}}$ |

3.5.4 Puntos suspensivos

Escribiendo tres puntos no se obtiene el correcto espaciado, de manera que L^AT_EX provee los comandos `\ldots` y `\cdots`. Los puntos centrados (`\cdots`) deben usarse entre símbolos como $+$, $-$, $*$, $=$, \subset y otras relaciones similares. Los puntos bajos (`\ldots`) se utilizan entre comas, y cuando las cosas están yuxtapuestas sin símbolos entre ellas. Aquí hay algunos ejemplos.

Escriba

`$a_1+ \cdots + a_n$`

`$x_1 \times x_2 \times \cdots \times x_n$`

`$v_1 = v_2 = \cdots = v_n = 0$`

`$f(x_1,\ldots,x_n) = 0$`

`$x_1 x_2 \ldots x_n$`

`$(1-x)(1-x^2)\ldots(1-x^n)$`

`$n(n-1)\ldots(1) = n!$`

Para producir

$a_1 + \cdots + a_n$

$x_1 \times x_2 \times \cdots \times x_n$

$v_1 = v_2 = \cdots = v_n = 0$

$f(x_1, \dots, x_n) = 0$

$x_1 x_2 \dots x_n$

$(1-x)(1-x^2) \dots (1-x^n)$

$n(n-1) \dots (1) = n!$

3.5.5 Texto dentro de una expresión matemática

Podemos usar el comando `\mbox` para insertar texto dentro de una expresión matemática. Este comando obliga a L^AT_EX a salir temporalmente del modo matemático, de manera tal que el argumento es tratado como texto normal.

Obtenemos

$$M^\perp = \{f \in V' : f(m) = 0 \text{ para todo } m \in M\}$$

Escribiendo

$$M^\perp = \{f \in V' : f(m)=0 \text{ para todo } m \in M\}$$

Note los espacios en blanco antes y después de las palabras ‘para todo’. Si hubiera escrito

$$M^\perp = \{f \in V' : f(m)=0 \text{ para todo } m \in M\}$$

Obtendría

$$M^\perp = \{f \in V' : f(m) = 0 \text{ para todo } m \in M\}$$

3.5.6 Funciones predefinidas

Los nombres de funciones deben ser escritos con el estilo **roman**. Por ejemplo

$$f(\theta) = \cos(\theta) + \log(\theta + 1) - \cosh(\theta^2 + 1).$$

\LaTeX provee estas funciones definidas como comandos. La tabla 13 muestra las que están disponibles.

| | | | | | | | |
|---------------------|-------------------|------------------|------------------|---------------------|---------------------|------------------|-------------------|
| $\backslash\arccos$ | $\backslash\cos$ | $\backslash\csc$ | $\backslash\exp$ | $\backslash\ker$ | $\backslash\limsup$ | $\backslash\min$ | $\backslash\sinh$ |
| $\backslash\arcsin$ | $\backslash\cosh$ | $\backslash\deg$ | $\backslash\gcd$ | $\backslash\lg$ | $\backslash\ln$ | $\backslash\Pr$ | $\backslash\sup$ |
| $\backslash\arctan$ | $\backslash\cot$ | $\backslash\det$ | $\backslash\hom$ | $\backslash\lim$ | $\backslash\log$ | $\backslash\sec$ | $\backslash\tan$ |
| $\backslash\arg$ | $\backslash\coth$ | $\backslash\dim$ | $\backslash\inf$ | $\backslash\liminf$ | $\backslash\max$ | $\backslash\sin$ | $\backslash\tanh$ |

Tabla 13: Funciones predefinidas

Observe que las funciones \sin , \arcsin , \sinh están predefinidas en inglés. Por otra parte, podríamos querer escribir una función que no esté predefinida en \LaTeX . Los nombres de otras funciones o abreviaciones que no están en la lista se pueden obtener cambiando el tipo de letra a **roman**. De esta forma obtenemos $\text{Aut}(V)$ escribiendo $\{\text{\rm Aut}\}(V)$. Note que si escribimos $\text{Aut}(A)$, obtendríamos $\text{Aut}(V)$, debido a que \TeX trató Aut como el producto de las tres cantidades A , u y t . De la misma manera obtenemos las letras griegas mayúsculas que no están listadas en la tabla 6.

3.5.7 Delimitadores

Los delimitadores izquierdos más frecuentes incluyen (, [y {, los cuales se obtienen escribiendo (, [y \{ respectivamente. Los correspondientes delimitadores derechos son),] y } que se obtienen escribiendo),] y \}. Además | y || se obtienen escribiendo | y \| respectivamente.

Sea X un espacio de Banach y sea $f : B \rightarrow \mathbf{R}$ un funcional lineal acotado en X . La *norma* de f , denotada por $\|f\|$, está definida por

$$\|f\| = \inf\{K \in [0, +\infty) : |f(x)| \leq K\|x\| \text{ para todo } x \in X\}.$$

podemos obtenerlo, escribiendo

Sea X un espacio de Banach y sea $f : B \rightarrow \mathbf{R}$ un funcional lineal acotado en X . La *norma* de f , denotada por $\|f\|$, está definida por

$$\|f\| = \inf\{K \in [0, +\infty) : |f(x)| \leq K\|x\| \text{ para todo } x \in X\}.$$

| | | | | | |
|---|---------|---|------------|---|--------------|
| (| (|) |) | ↑ | \uparrow |
| [| [|] |] | ↓ | \downarrow |
| { | \{ | } | \} | ↕ | \updownarrow |
| ⌊ | \lfloor | ⌋ | \rfloor | ↑ | \Uparrow |
| ⌈ | \lceil | ⌋ | \rceil | ↓ | \Downarrow |
| ⟨ | \langle | ⟩ | \rangle | ↕ | \Updownarrow |
| / | / | \ | \backslash | | |
| | | | \ | | |

Tabla 14: Delimitadores

Algunas veces se requieren delimitadores del tamaño apropiado. Considere por ejemplo, el problema de escribir

$$f(x, y, z) = 3y^2z \left(3 + \frac{7x + 5}{1 + y^2} \right).$$

La manera de obtener los paréntesis grandes consiste en escribir `\left(` para el paréntesis izquierdo y `\right)` para el paréntesis derecho (vea ud. mismo lo que ocurre si coloca (y)). Así, la expresión de arriba se obtiene con

$$f(x,y,z)=3y^2z\left(3+\frac{7x+5}{1+y^2}\right).$$

La tabla 14 (página anterior) muestra los símbolos que L^AT_EX reconoce como delimitadores, esto es, símbolos que pueden ser precedidos por `\left` y `\right`.

Podemos escribir delimitadores dentro de otros delimitadores.

```
\[ \left|4x^3+\left(x+\frac{42}{1+x^4}\right)\right|. \]
```

produce

$$\left|4x^3 + \left(x + \frac{42}{1+x^4}\right)\right|.$$

Escribiendo `\left.` y `\right.` obtenemos *delimitadores nulos* los cuales son completamente invisibles. Considere, el problema de escribir

$$\left. \frac{du}{dx} \right|_{x=0} = -1.$$

Quisieramos hacer la barra vertical grande, de manera que abarque la derivada que la precede. para hacer esto, supongamos que la derivada está encerrada por delimitadores, donde el delimitador izquierdo es invisible y el derecho es la barra vertical. El delimitador invisible se produce usando `\left.` (usaríamos `\right.` para un delimitador derecho invisible) y la expresión total con

```
\[ \left. \frac{du}{dx} \right|_{x=0} = -1. \]
```

3.5.8 Símbolos superpuestos

L^AT_EX permite colocar un símbolo sobre otro por medio del comando `\stackrel` este comando acepta dos argumentos y coloca el primero centrado sobre el segundo.

Escriba

```
$X \stackrel{f^*}{\rightarrow} Y$
```

```
$f(x) \stackrel{\triangle}{=} x^2 + 1$
```

```
$f_k \stackrel{\mu}{\longrightarrow} f$
```

```
$\psi(x) \stackrel{\rm def}{=} \Delta(x-x_0)$
```

Para producir

$$X \xrightarrow{f^*} Y$$

$$f(x) \triangleq x^2 + 1$$

$$f_k \xrightarrow{\mu} f$$

$$\psi(x) \stackrel{\text{def}}{=} \Delta(x - x_0)$$

3.5.9 Subrayado

El comando `\underline` colocará una línea continua bajo su argumento, y el comando `\overline` colocará una línea continua sobre su argumento.

| <i>Escriba</i> | <i>Para producir</i> |
|--|---|
| <code>\overline{a+bi} = a- bi</code> | $\overline{a + bi} = a - bi$ |
| <code>\overline{\overline{a+bi}} = a+bi</code> | $\overline{\overline{a + bi}} = a + bi$ |

Podemos colocar llaves horizontales sobre o debajo de una expresión haciéndola argumento de `\overbrace` y `\underbrace`. A su vez podemos colocar una etiqueta sobre o debajo de las llaves colocándola como superíndice o subíndice respectivamente.

`$A^n=\overbrace{A \times A \times\cdots \times A}^{\mbox{\$n\$ t' erminos}}$`

`$\forall x \underbrace{\exists y (y \succ x)}_{\mbox{alcance de }}\forall x$`

producirá

$$A^n = \overbrace{A \times A \times \cdots \times A}^{n \text{ términos}} \quad y \quad \forall x \underbrace{\exists y (y \succ x)}_{\text{alcance de } \forall}$$

3.5.10 Operadores; Sumas, Integrales, etc.

Cada uno de los símbolos de operación en la tabla 11 generalmente aparecen con límites. Ellos son especificados como subíndices y superíndices del operador, y \LaTeX los colocará en la posición apropiada. En una expresión ‘en el texto’ los índices aparecerán en la posición usual; pero en una expresión ‘desplegada’ serán colocados sobre y debajo del símbolo (el cual será algo más grande). Lo siguiente dará una idea de su uso.

| <i>Escriba</i> | <i>Para producir</i> |
|---|--|
| <code>\sum_{i=1}^N a_i</code> | $\sum_{i=1}^N a_i$ |
| <code>\int_a^b f</code> | $\int_a^b f$ |
| <code>\oint_{\cal C} f(x)\,dx</code> | $\oint_{\mathcal{C}} f(x) dx$ |
| <code>\prod_{\alpha \in A} X_\alpha</code> | $\prod_{\alpha \in A} X_\alpha$ |
| <code>\lim_{N \rightarrow \infty} \sum_{i=1}^N f(x_i) \Delta x_i</code> | $\lim_{N \rightarrow \infty} \sum_{i=1}^N f(x_i) \Delta x_i$ |

Veremos el uso de `\`, en la sección 3.5.12. Veamos esas mismas expresiones cuando están desplegadas.

$$\sum_{i=1}^N a_i \quad , \quad \int_a^b f \quad , \quad \oint_C f(x) dx \quad , \quad \prod_{\alpha \in A} X_\alpha \quad , \quad \lim_{N \rightarrow \infty} \sum_{i=1}^N f(x_i) \Delta x_i$$

3.5.11 Matrices y otros arreglos en \LaTeX

Las matrices y otros arreglos son producidos en \LaTeX usando el ambiente `array`. Por ejemplo, supongamos que queremos escribir el siguiente pasaje:

El *polinomio característico* $\chi(\lambda)$ de la matriz

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

está dado por

$$\chi(\lambda) = \begin{vmatrix} \lambda - a & -b & -c \\ -d & \lambda - e & -f \\ -g & -h & \lambda - i \end{vmatrix}.$$

Este pasaje fue producido con:

```
El {\em polinomio caracter}\'\i stico} $\chi(\lambda)$ de la matriz
\[ \left( \begin{array}{ccc}
a & b & c \\
d & e & f \\
g & h & i
\end{array} \right) \]
est\'a dado por
\[ \chi(\lambda) =
\left|
\begin{array}{ccc}
\lambda - a & -b & -c \\
-d & \lambda - e & -f \\
-g & -h & \lambda - i
\end{array}
\right|. \]
```

Primero que nada, note el uso de `\left` y `\right` para producir los delimitadores grandes alrededor de los arreglos. Seguidamente observe el uso del caracter tabulador `&` para separar las entradas de la matriz, y el uso de `\\` para separar las filas de la matriz. El arreglo comienza con `\begin{array}`

y finaliza con `\end{array}`. Lo único que resta por explicar es el misterioso `{ccc}` el cual aparece inmediatamente después de `\begin{array}`. Cada una de las `c` en `{ccc}` representa una columna de la matriz e indica que las entradas de las columnas deben ser *centradas* (`center`). Si la `c` es reemplazada por `l` entonces la columna correspondiente será escrita con todas las entradas alineadas a la *izquierda* (`left`) y `r` producirá una columna con sus entradas alineadas a la *derecha* (`right`). Es decir este argumento permite declarar el número y justificación de las columnas.

Algunas veces necesitaremos escribir matrices con un patrón genérico, donde debemos usar puntos suspensivos verticales y diagonales. \LaTeX provee `\vdots` (puntos verticales) y `\ddots` (puntos diagonales). Por ejemplo,

```
\det A = \left| \begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{array} \right|
```

produce

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{vmatrix}$$

Podemos usar un arreglo para producir una expresión como

$$|x| = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$$

Note que las dos columnas del arreglo están alineadas a la izquierda. Por lo tanto usé `{ll}` inmediatamente después de `\begin{array}`. La llave fue producida usando `\left{`. Sin embargo se requiere un correspondiente delimitador derecho. Entonces usé el *delimitador invisible* `\right.` discutido anteriormente. La expresión fue obtenida escribiendo

```
\[ |x| = \left\{ \begin{array}{ll}
x & \mbox{si } x \geq 0 \\
-x & \mbox{si } x < 0
\end{array} \right. \]
```

3.5.12 Espaciado

Algunas veces \LaTeX necesita algo de ayuda al colocar el espaciado en una expresión, o quizá usted piense que el espaciado necesita algún ajuste. Para ese propósito tenemos los siguientes comandos:

| | | | |
|------------------|--------------------------|--------------------|----------------------|
| $\backslash,$ | espacio pequeño | $\backslash:$ | espacio mediano |
| $\backslash!$ | espacio pequeño negativo | $\backslash;$ | espacio grande |
| \backslashquad | un quad de espacio | \backslashqqquad | dos quads de espacio |

Los comandos $\backslash,$, \backslashquad y \backslashqqquad también pueden ser usados en modo texto.

Una integral típica es la siguiente

$$\int_a^b f(x) dx.$$

La cual se escribe con

```
\[ \int_a^b f(x) dx. \]
```

El propósito de $\backslash,$ antes de dx es colocar un espacio extra antes de la d . Esto es necesario para producir la apariencia correcta.

En algunas integrales múltiples encontraremos que \LaTeX coloca mucho espacio entre los signos de integral. Debemos usar el comando $\backslash!$ para remover el espacio extra que no queremos. Así, por ejemplo, la integral múltiple

$$\int_0^1 \int_0^1 x^2 y^2 dx dy.$$

se obtiene escribiendo

```
\[ \int_0^1 \! \int_0^1 x^2 y^2 dx dy. \]
```

Si hubiera escrito

```
\[ \int_0^1 \int_0^1 x^2 y^2 dx dy. \]
```

obtendría

$$\int_0^1 \int_0^1 x^2 y^2 dx dy.$$

Un ejemplo notable ocurre cuando escribimos una integral múltiple como

$$\iint_D f(x, y) dx dy.$$

Aquí utilicé $\backslash!$ tres veces para obtener el espacio adecuado entre los signos de integral. Esta integral fue obtenida con

`\[\int \!\!\! \int_D f(x,y)\,dx\,dy. \]`

Si hubiera escrito

`\[\int \int_D f(x,y)\,dx\,dy. \]`

obtendría

$$\int \int_D f(x,y) dx dy.$$

Aquí hay algunos ejemplos de utilización de los comandos de espaciado para hacer algunas modificaciones a esas expresiones

| <i>Escriba</i> | <i>Para producir</i> |
|--|----------------------------------|
| <code>\sqrt{2} \, x</code> | $\sqrt{2} x$ |
| <code>\int_a^b f(x)\,dx</code> | $\int_a^b f(x) dx$ |
| <code>\Gamma_{\!2}</code> | Γ_2 |
| <code>\int_a^b \!\! \int_c^d f(x,y)\,dx\,dy</code> | $\int_a^b \int_c^d f(x,y) dx dy$ |
| <code>x / \! \cos x</code> | $x/\cos x$ |
| <code>\sqrt{\, \cos x}</code> | $\sqrt{\cos x}$ |

3.5.13 Despliegues multilineales

Recordemos que el ambiente `equation` puede ser usado para desplegar y numerar automáticamente una ecuación en una sola línea. El ambiente `eqnarray` se utiliza para desplegar y numerar automáticamente, sea una sola expresión que se desglosa en varias líneas o bien, expresiones múltiples, teniendo cuidado de la alineación, en particular de los operadores binarios. La sintaxis es similar a la del ambiente `array`, excepto que no es necesario ningún argumento para declarar el número y justificación de las columnas. El ambiente `\eqnarray*` hace lo mismo, pero sin numerar ninguna de las ecuaciones.

```
\begin{eqnarray}
(a+b)(a-b) & = & a^2-b^2 \\
(a+b)(a+b) & = & a^2+2ab+b^2
\end{eqnarray}
```

Produce

$$(a + b)(a - b) = a^2 - b^2 \tag{2}$$

$$(a + b)(a + b) = a^2 + 2ab + b^2 \tag{3}$$

Vemos cómo identificar las columnas de manera que los signos = sean alineados, esto vale para cualquier símbolo. Podemos dejar entradas vacías para obtener un efecto como el siguiente

$$\begin{aligned}\cos 2\theta &= \cos^2 \theta - \operatorname{sen}^2 \theta \\ &= 2 \cos^2 \theta - 1.\end{aligned}$$

El cual fue producido con

```
\begin{eqnarray*}
\cos 2\theta &= & \cos^2 \theta - \{\rm sen\!}\^2 \theta \\
&= & 2 \cos^2 \theta - 1.
\end{eqnarray*}
```

Por otra parte podríamos querer numerar sólo algunas líneas. Para esto usamos el comando `\nonumber` en las líneas que no queremos numerar

```
\begin{eqnarray}
(a+b)(a-b) &= & a^2-ab+ab+b^2 \nonumber \\
&= & a^2-b^2; \\
(a+b)^2 &= & a^2+2ab+b^2
\end{eqnarray}
```

Produce

$$\begin{aligned}(a+b)(a-b) &= a^2 - ab + ab + b^2 \\ &= a^2 - b^2;\end{aligned}\tag{4}$$

$$(a+b)^2 = a^2 + 2ab + b^2\tag{5}$$

3.5.14 Teoremas, Proposiciones, ...

Cuando escribimos un documento matemático probablemente necesitemos escribir “Lemas”, “Definiciones”, “Axiomas” y estructuras similares. Siendo estructuras tan importantes, estas deben distinguirse del resto del texto común, esto es, los enunciados deben ser escritos con un tipo de letra distinguida. Además deben poseer una numeración. \LaTeX maneja estas estructuras “como-teorema” con el comando

```
\newtheorem{nombre}[contador]{texto}[section]
```

El argumento `nombre` es una palabra de nuestra propia elección para identificar el “teorema”. Con el argumento `texto` definimos el nombre real del

“teorema” el cual será impreso en el documento final. Estos comandos (uno para cada “teorema”) deben ir en el preámbulo del documento fuente, esto es, antes de `\begin{document}`.

Los argumentos en corchetes son opcionales. Ambos son utilizados para especificar la numeración utilizada para el “teorema”. Con el argumento `contador` especificamos el nombre de un “teorema” previamente declarado. El nuevo “teorema” será numerado en la misma secuencia. El argumento `section` nos permite especificar la unidad seccional dentro de la cual queremos que nuestro “teorema” sea numerado.

Por ejemplo, declarando

```
\newtheorem{Teo}{Teorema}
\newtheorem{Corolario}{Corolario}
```

Podemos escribir después

```
\begin{Teo}
  Para cada conjunto abierto  $\Omega$  en el plano, cada  $f \in H(\Omega)$ 
  es representable por una serie de potencias en  $\Omega$ .
\end{Teo}

\begin{Corolario}
  Si  $f \in H(\Omega)$ , entonces  $f' \in H(\Omega)$ .
\end{Corolario}

\begin{Teo}[Liouville]
  Si  $f$  es entera y acotada para todos los valores de  $z$  en el
  plano complejo, entonces  $f(z)$  es constante.
\end{Teo}
```

Para producir

Teorema 1 *Para cada conjunto abierto Ω en el plano, cada $f \in H(\Omega)$ es representable por una serie de potencias en Ω .*

Corolario 1 *Si $f \in H(\Omega)$, entonces $f' \in H(\Omega)$.*

Teorema 2 (Liouville) *Si f es entera y acotada para todos los valores de z en el plano complejo, entonces $f(z)$ es constante.*

3.6 A dónde debemos dirigirnos ahora?

Hemos cubierto gran parte de las habilidades matemáticas de \LaTeX que nos permiten escribir un documento matemático más o menos complejo. Podríamos, sin embargo necesitar escribir alguna expresión algo más complicada de las que hemos visto disponiendo solamente de las herramientas descritas en este artículo—no es difícil pensar en una expresión que no sabríamos cómo escribirla con \LaTeX . Trate de escribir *exactamente* la siguiente expresión

$$\text{Si } [x-s[x]] \approx [y-s[y]], \text{ entonces } \sum'_{x \in A} f(x) \stackrel{\text{def}}{=} \sum_{\substack{x \in A \\ x \neq 0}} f(x*y), \text{ con } A \subseteq \mathbf{R} \text{ y } f \in \mathfrak{p}.$$

No puede verdad? Pues tampoco podrá escribir

$$\underbrace{\overbrace{\{a, \dots, a\}}^{k \text{ a's}} \overbrace{\{b, \dots, b\}}^{l \text{ b's}}}_{k+l \text{ elementos}} = {}_k \mathfrak{P}_l^a \stackrel{\text{def}}{=} \left. k^{k^{k^{\dots^k}}} \right\}^l.$$

De hecho, no podemos criticar a \LaTeX sin saber cual es su verdadera capacidad. Por lo tanto, el primer lugar a dónde debemos dirigirnos es el Manual *\LaTeX : User's Guide and Reference Manual* [2]. Un buen conocimiento de los ambientes \LaTeX y sus opciones (y no he descrito varios de ellos) son una poderosa herramienta para enfrentarse a la mayoría de los problemas de tipografía. Un poco de imaginación (digamos, por ejemplo, dándole un uso ligeramente no standard a un ambiente) puede quizá resolver muchos problemas difíciles. Por último, la mayor parte de \TeX “vive” en \LaTeX y así es cierto que podemos hacer *cualquier cosa* con \LaTeX (usted no tiene la menor idea de lo que \TeX es capaz de hacer con cerca de 300 comandos primitivos)—usted debe consultar el *\TeX Book* [1] o pedirle a un experto en \TeX que le ayude.

4 Referencias Cruzadas

Secciones En general, \LaTeX numera automáticamente hasta las subsecciones. Cuando queramos hacer referencia a uno de estos números no debemos hacerlo directamente, sino que al lado del comando que da inicio a la (sub-sub)sección deberemos poner la instrucción `\label{etiqueta}`,

(donde `etiqueta` es una secuencia corta de letras y/o números exclusiva de esa (sub-sub)sección) y hacer la referencia en la forma siguiente

... como hemos visto en `\ref{etiqueta}`, se cumple ...

De este modo la referencia seguirá siendo correcta aunque intercalemos o suprimamos otras secciones.

Importante: es preciso compilar el documento dos (2) veces para que las referencias cruzadas tomen efecto.

Bibliografía Supongamos que la bibliografía está especificada como:

```
\begin{thebibliography}{99}
\item{Knu} Donald Ervin Knuth. {\em The \TeX book},
      Addison-Wesley Pub Co, 1986.
\item{Lam} Leslie Lamport.
      {\em \LaTeX\ : User's Guide and Reference Manual},
      Addison-Wesley Pub Co, 1994.
\item{Mal} Gavin Maltby. {\em An introduction to \TeX\ and friends},\@
      \verb@ftp:\@ftp.dante.de\tex-archive\documentation\maltby.tex@
\item{Wil} D. R. Wilkins. {\em Getting Started with \LaTeX},\@
      \verb@ftp:\@ftp.dante.de\tex-archive\documentation\tcd-manual\@
\end{thebibliography}
```

Para referirse a una obra, por ejemplo a la de Lamport, usamos la instrucción `\cite{Lam}`.

Ecuaciones Para que una ecuación aparezca numerada, en lugar de escribirla entre dólares dobles usamos:

```
\begin{equation}\label{suma}
2+2=5
\end{equation}
```

que produce

$$2 + 2 = 5 \tag{6}$$

de manera que al escribir

la ecuación `\ref{suma}` no es correcta

obtenemos “la ecuación (6) no es correcta”

Teoremas Una etiqueta `\label{etiqueta}` tras un `\begin{theorem}` permite referirse al teorema mediante `\ref{etiqueta}`.

Tablas y Figuras Ver la sección de objetos flotantes.

5 Objetos flotantes

5.1 Tablas y Figuras

Tablas Las tablas (sencillas) se editan en forma similar a las matrices pero en las tablas se pueden poner líneas verticales y horizontales. El modo matemático debe especificarse en una tabla.

- Para poner líneas verticales se ponen marcas como `|` o `||` en la parte que corresponde al alineamiento de columnas.
- Para poner líneas horizontales, al final de cada fila se especifica
 - `\hline`: línea tan larga como la tabla
 - `\cline{i-j}`: línea de columna i a columna j .

Por ejemplo

```
\begin{center}
\begin{tabular}{|c|c|c|} \hline


$p$  &  $q$  &  $p \rightarrow q$  \\
\hline
F & F & V \\
F & V & V \\
V & F & F \\
V & V & V \\
\hline
\end{tabular}
\end{center}


```

produce

| p | q | $p \rightarrow q$ |
|-----|-----|-------------------|
| F | F | V |
| F | V | V |
| V | F | F |
| V | V | V |

Figuras Con MiKTeX se puede incluir gráficos en diferentes formatos: *.emf, *.wmf, *.gif, *.png, *.tiff y *.bmp. Se puede el formato *.eps (es el más recomendable) si está instalado el programa Ghostscript. En el preámbulo del documento hay que poner:

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

y la sintaxis es

```
\includegraphics[opciones]{archivo.eps}
```

Importante: el archivo a incluir debe estar en la misma ubicación del archivo .tex que estamos trabajando.

Por ejemplo

```
\begin{center}  
  \includegraphics[height=2cm]{tipografia.eps}  
\end{center}
```

produce



Para más opciones consulte C:\texmf\doc\latex\graphics\epslatex.ps, si tiene instalado MiKTeX.

5.2 Objetos Flotantes

Un objeto (gráfico o una tabla) debe aparecer en el lugar más cercano al texto que hace referencia a él. Al ir haciendo cambios en el texto, los objetos pueden desplazarse de manera no apropiada. \LaTeX resuelve este problema manipulando las figuras como objetos flotantes en el documento. \LaTeX nos ofrece dos comandos (ambientes) para indicarle nuestras preferencias sobre el desplazamiento del objeto:

```
\begin{figure}[h]
...
\label{fig:nombre}
\caption{}
\end{figure}
```

y

```
\begin{table}[h]
...
\label{nombre}
\caption{}
\end{table}
```

- `[h]` le indica a \LaTeX que queremos la figura o la tabla, exactamente en ese lugar (`h=here`). Otras opciones son `[t]=top`, `[b]=bottom`, `[htb]=here o top o bottom,...`. Si no se pone algo, el default es `[htbp]`. Por otra parte, `[h!]` obliga a que \LaTeX coloque el objeto exactamente en el lugar señalado.
- `\caption{texto}` es la etiqueta de cada objeto (numerándolo automáticamente). Se puede omitir.
- `\label` es la identificación del objeto. En el texto podemos hacer referencia a la tabla o a la figura, poniendo en la figura `\ref{fig:nombre}...` o en la tabla `\ref{nombre}...`. Si no vamos a hacer referencia, podemos omitir este comando.

El texto

```
\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline


$p$  &  $q$  &  $p \rightarrow q$  \\


\hline
F & F & V \\
F & V & V \\
V & F & F \\
V & V & V \\
\hline
\end{tabular}
\caption{ {\small Tabla de verdad para  $p \rightarrow q$  } }
\end{center}
\end{table}
```

produce

| p | q | $p \rightarrow q$ |
|-----|-----|-------------------|
| F | F | V |
| F | V | V |
| V | F | F |
| V | V | V |

Tabla 15: Tabla de verdad para $p \rightarrow q$

6 Ajustes finales

Cualquier documento escrito en español necesita ajustes obvios, pues \LaTeX escribe por defecto todos esos elementos en inglés.

Para que \LaTeX escriba en lugar de esos elementos lo que deseemos podemos usar el comando básico \TeX `\def`, en el preámbulo del documento. Por ejemplo, si queremos que cada capítulo de un reporte o libro comience con **Capítulo** en lugar de **Chapter** (por defecto), colocamos la siguiente línea en el preámbulo del documento.

```
\def\chaptername{Cap\'}\i tulo}
```

También podemos usar el comando `\newcommand` en lugar de `\def`. La lista completa de comandos es, aunque algunos de ellos no se describen en este artículo, es:

```
\def\chaptername{Cap\'}\i tulo}
\def\contentsname{Contenido}
\def\abstractname{Res\'}umen}
\def\refname{Referencias}
\def\bibname{Bibliograf\'}\i a}
\def\tablename{Tabla}
\def\listfigurename{Lista de Figuras}
\def\indexname{Indice}
\def\figurename{Figura}
\def\partname{Parte}
\def\appendixname{Ap\'}endice}
```

\TeX opera con ciertos algoritmos para separar las palabras en sílabas. Estos se necesitan cuando parte de una palabra “cae” al final de una línea y la parte restante comienza la siguiente línea. Estos algoritmos son muy eficaces, particularmente en inglés. Si nuestro documento está en español debemos revisar el final de cada línea del documento, para detectar guiones que hayan sido colocados en el lugar incorrecto.

Una manera de corregir esto es obligar a \LaTeX a que coloque el guión en el lugar más conveniente. Esto se hace con el comando `\-`. Por ejemplo, en la página 31 la palabra operador está separada incorrectamente. Esto pude haberlo corregido escribiendo

```
... ope\-rador
```

obligando a \LaTeX a colocar un guión entre la `e` y la `r`.

Referencias

- [1] Donald Ervin Knuth. *The T_EXbook*, Addison-Wesley Pub Co, 1986.
- [2] Leslie Lamport. *L^AT_EX : User's Guide and Reference Manual*, Addison-Wesley Pub Co, 1994.
- [3] Gavin Maltby. *An introduction to T_EX and friends*,
`ftp:\\ftp.dante.de\tex-archive\documentation\maltby.tex`
- [4] D. R. Wilkins. *Getting Started with L^AT_EX*,
`ftp:\\ftp.dante.de\tex-archive\documentation\tcd-manual\`

Manuel P. Maia F.
Universidad Central de Venezuela
Facultad de Ciencias Económicas y Sociales
Escuela de Administración y Contaduría
`mmaia@euler.ciens.ucv.ve`