



Tutorial de L^AT_EX

version 1.4

por
Freddy Pérez

Octubre 2005

Índice general

1. Introducción	3
2. Estructuración del documento	5
2.1. Clases y Paquetes	6
2.2. Comandos	8
2.2.1. Comandos de configuración de páginas	250
2.2.2. Comandos de configuración de párrafos	10
2.2.3. Comandos de configuración de palabras y letras	10
2.3. Ambientes	13
2.3.1. document	14
2.3.2. center	14
2.3.3. verbatim	14
2.3.4. verse	15
2.3.5. quotation	16
2.3.6. minipage	17
2.3.7. raggedright y raggedleft	18

2.3.8. tabbing	19
2.3.9. Ambientes de enumeración	19
2.3.10. Ambiente matemático	21
2.3.11. tabular	27
2.3.12. figure	28
2.3.13. Teoremas	30
2.4. Referenciando páginas, figuras, tablas y ecuaciones	31
2.5. Comandos personalizados	32
3. Partes de un Documento	33
3.1. Referencias Bibliográficas	34
4. Caracteres reservados de LaTeX	36
5. Compilación de archivos \LaTeX	37
5.1. Compilando archivos \LaTeX en Linux	37
5.2. Compilando archivos \LaTeX en Windows	38
5.3. Manejo de Errores	38

Capítulo 1

Introducción

\LaTeX fue escrito por Leslie B. Lamport, a mediados de los 80's, es una de las variantes más populares de \TeX ¹. La primera version que se tuvo fue la 2.09 (1986), actualmente la versión oficial es la 2 ϵ (junio de 1994). Entre las diferecias que hay entre estas versiones tenemos el comando `\documentclass` (versión 2 ϵ) y `\documentstyle` (versión 2.09), para los comandos y ejemplos de este tutorial usaremos la versión oficial.

Su gran potencialidad en el manejo de fórmulas matemáticas, cuadros y tablas lo hace adecuado para la producción de artículos científicos y libros de alta calidad. \LaTeX permite la enumeración automática tanto de páginas como de capítulos, secciones, teoremas, ecuaciones, figuras, etc., también facilita la referencia a figuras, tablas, y demás objetos.

\LaTeX no sólo se destaca por lo anteriormente escrito sino también por el eficiente manejo de figuras, las cuales pueden ser creadas en el propio lenguaje o pueden ser importadas, teniéndose un control total y eficiente de las mismas. Es pertinente hacer conocer que \LaTeX no sólo sirve para hacer documentos, sino tambien presentaciones, similares a "Power Point" existen diferentes proyectos orientados a esto, uno de ellos es Prosper², lo mismo para diseñar posters y páginas web, a través del uso de un convertidor

¹Tex es un programa para procesar texto creado por E. Knuth

²Puede bajarse de <http://prosper.sourceforge.net/>

($\text{\LaTeX}2_{\text{HTML}}$) escrito en Perl³. A diferencia de otros procesadores de texto \LaTeX es compilado, por tanto la forma en que se escribe no es la apariencia final del documento, este hecho es visto por sus detractores como una gran desventaja.

\LaTeX permite crear estilos, los cuales pueden usarse como formatos de diversos documentos, como: cartas, oficios, resumes, informes, papers, tesis, etc. Tal es así que muchas universidades tienen un estilo personalizado que los estudiantes usan como plantilla al escribir sus tesis. En el estilo se encuentran definidos los tipos de letra, numeración, márgenes y demás requisitos que exige la escuela graduada de la universidad.

Para editar un documento en \LaTeX se puede utilizar cualquier editor de texto, si es para Windows se puede usar desde el bloque de notas (“Notepad”) hasta WinEdt⁴, si es Linux se puede utilizar cualquier editor, como por ejemplo vi, xemacs, etc. Una vez terminada la edición del documento éste debe ser compilado. Este proceso será explicado en el último capítulo de este manual.

El objetivo de este manual es enseñar a escribir un documento de mejor calidad mostrando algunas de las herramientas que posee \LaTeX y como usarlas adecuadamente en la redacción de los mismos.

³Practical Extraction and Report Language, es un lenguaje de programación

⁴Disponible en <http://www.winedt.com>

Capítulo 2

Estructuración del documento

A diferencia de Word y otros procesadores de texto en los que se utiliza el enfoque denominado WYSIWYG (“What You See Is What You Get”), \LaTeX es un procesador de texto que primero necesita ser compilado, lo cual le dá una buena portabilidad obteniéndose siempre el mismo documento, pues es independiente de la máquina, el sistema operativo, el editor y la version que se utilice.

Luego de escoger el editor de su preferencia y escribir el documento \LaTeX necesita guardar el archivo con la extension “.tex”, puesto que el compilador solo reconoce los archivos que tienen esa terminación, por ejemplo: informe.tex. Si el documento va a tener una bibliografía, necesita crear un archivo con la extensión “.bib”, en el cual debe escribir sus referencias, usando los formatos establecidos para las diferentes clases de referencia.

Una vez que tenemos el archivo con la extensión correcta, la primera línea que necesitamos escribir, es la que especificará al compilador de \LaTeX el tipo de documento que vamos a redactar, este comando establece el tipo de letra, el tamaño y demás detalles del documento.

Por tanto, la primera línea en nuestro documento será la siguiente:

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

2.1. Clases y Paquetes

Las clases y las opciones que tienen estas, así como los diversos paquetes¹, son los que dan la potencia a \LaTeX , pues en ellas se definen las diferentes herramientas para la edición personalizada de los documentos. Existen diversos lugares en internet donde se puede conseguir diferentes estilos, fuentes y otras herramientas, la página oficial de \LaTeX (<http://www.ctan.org>) tiene un listado casi completo de esas ayudas, así como tutoriales.

Entre las clases más populares tenemos:

- **article** Este tipo es conveniente en la mayoría de los casos, por su flexibilidad y por las pocas opciones que tiene, las tablas y ecuaciones son enumeradas consecutivamente.
- **book** Este tipo de clase como su mismo nombre lo indica es usado para escribir libros, tiene estilos de encuadernación entre otros.
- **report** Este estilo es sugerido para documentos largos. Aquí cuentas con un nivel más de encabezados (Chapter) que en el estilo article. Además las tablas, figuras y ecuaciones son numeradas según los capítulos y no como en el estilo article.
- **letter** Al igual que la clase book es especial para escribir cartas y los sobres de las mismas.
- **slides** Esta clase nos permite hacer presentaciones similares a power point, hay diversas variedades de paquetes predefinidos para hacer presentaciones, el más popular es Prosper.
- **proc** Este tipo es basado en el tipo article y es especial para escribir la recopilación de artículos de una conferencia.
- **ltxdoc** Esta clase sirve para crear una documentación de \LaTeX y está basado también en la clase article.

¹se conocen porque tiene la extensión .sty

Cada una de estas clases tiene muchas opciones y el formato del documento cambia de acuerdo a la clase que se use.

Definida nuestra primera línea, necesitamos en seguida definir que tipo de paquetes se usarán, los cuales dependerán del documento a crear, por ejemplo;

```
\documentclass[twoside,dvips]{article}
\usepackage{graphics,color}
```

La segunda línea nos indica que se usarán los paquetes “color” y “graphics”, estos paquetes son usados para manejar figuras, cambiar el color de las letras, el fondo de la página, etc. Al igual que las clases existen muchos paquetes definidos, entre los más comunes tenemos:

- **amslatex** Este paquete fue creado por la “American Mathematical Society” es uno de los más avanzados para escribir fórmulas matemáticas y demás cosas relacionadas con esta ciencia. En este paquete se incluye el paquete “amsmath” el cual provee muchos comandos para escribir fórmulas matemáticas de alta complejidad.
- **babel** Este paquete es otro de los más usados pues nos da el soporte necesario para poder usar diferentes idiomas.
- **cyrillic** Para poder usar todas las ventajas que nos provee “Cyrillic Fonts” necesitamos incluir este paquete en nuestro documento, este paquete tiene diversos tipos de letra.
- **graphics** Este paquete provee el soporte necesario para la inclusión y manejo de gráficos además de poder incluir archivos producidos por otros programas tales como “Paint”, “Corel Draw”, etc. Dentro de este paquete está incluido el paquete “color” el cual nos permite manejar los colores tanto de las letras como de las figuras.
- **psnfs** Este paquete está diseñado para facilitarnos el uso de un amplio rango de tipos de letra llamado “Type 1 (PostScript)”.

Al igual que las clases se pueden crear diversos tipos de paquetes para documentos que tienen ciertas normas de creación.

2.2. Comandos

Seguido a estas dos líneas podemos incluir los paquetes que necesitamos usar para poder hacer uso de los diversos comandos que contienen. Reconocemos a un comando porque está precedido de un `\` y luego de este va el nombre del comando. Siguiendo con la estructuración del documento tenemos:

```
\documentclass[twoside,dvips]{article}
\usepackage{graphics,color}
\newcommand{\Real}{\mathbb R}
\textwidth 6.75in
\textheight 8.5in
\headheight 6.0in
\oddsidemargin 0in
\topmargin 0in
\leftmargin 1in
\parindent 0.5em
\parskip 2ex
\baselineskip 1.5pt
...
...
```

En este encabezado podemos observar varios comandos, tales como: “textwidth” donde como su nombre lo indica se refiere al ancho de la página, “textheight” se refiere a la altura de la página, “headheight” define el largo del texto excluyendo el encabezado y el pie de página, “oddsidemargin” dice al compilador que el margen izquierdo será de 1+0 pulgadas desde el borde izquierdo de la hoja, “topmargin” es el margen superior que en este caso será 1+0 pulgadas de la parte superior de la hoja, “leftmargin” es el margen izquierdo de la página, “parindent” se refiere a la indentación normal en un párrafo que en este caso nos dice que será a 3 espacios, “parskip” es para agregar un espacio adicional entre párrafos y “baselineskip” es la separación entre líneas.

\LaTeX nos permite definir nuevos comandos, como observamos en la tercera línea, donde definimos el comando `\Real` para escribir el símbolo que representa los números reales en matemáticas, “mathb” es un paquete para escribir letras usadas frecuentemente en la edición de textos matemáticos.

2.2.1. Comandos de configuración de páginas

Es posible configurar de acuerdo a las necesidades el contador de páginas en cualquier parte del documento, esto se hace con el comando `\setcounter{página}`, además puedes especificar el tipo de números que deseas a través del comando `\pagenumber`. Así, si quiero que el número de esta página sea 250 escribo `\setcounter{page}{250}` (si observan, el número de esta página es efectivamente 250). El comando `\pagenumbering{estilo}` coloca el contador de páginas a 1 y comienza a numerar de acuerdo con lo estilo especificado, pudiendo ser una de las siguientes opciones:

arabic : 1,2,3... (este es el estilo por defecto).

roman : i, ii, iii, iv, ...

Roman : I, II, III, IV, ...

alph : a, b, c, ...

Alph : A, B, C, ...

Si se desea que la numeración una página tenga un estilo predefinido o no tenga número se usa el siguiente comando `\pagestyle{estilo}`.

Los estilos disponibles para este comando son: **plain** el cual esta por defecto y coloca el número en el centro del pie de la página; **headings** en la cabecera de la página aparecerá el título del capítulo, el número de la página y el pie vacío; y **empty** es para que el pie y la cabecera de la página aparezcan vacíos. Estos estilos aplican a todo el documento por lo que si se quiere cambiar de estilo en la siguiente página se vuelve a inicializar el contador en el número y estilo deseado. Cabe destacar que es posible crear un estilo personalizado de página y/o usar los predefinidos en \LaTeX para ello se usa el comando `\thispagestyle{estilo}` el cual permite crear cabeceras y pie de página, es necesario conocer los parámetros que afectan estos formatos.

El comando `\newpage` fuerza un cambio de página, y los comandos `\nolinebreak` y `\nopagebreak` previenen un cambio de línea o página respectivamente.

2.2.2. Comandos de configuración de párrafos

Existen diversos comandos para la configuración de párrafos, para establecer la separación entre las líneas de un párrafo usamos el siguiente comando: `\setlength {\baselineskip} {1.5\baselineskip}` en este caso particular el espacio entre líneas se 1.5.

Si se desea una separación especial entre las líneas o párrafos en alguna parte del documento podemos insertar espacios verticales; para lo cual \LaTeX provee de tres comandos: `\smallskip` `\medskip` `\bigskip`. Los cuales insertan espacios predefinidos entre líneas; este espacio depende del estilo usando en el documento. Si se desea insertar una separación exacto se usa el comando `\vspace*{tamaño}`, el argumento tamaño consiste de un número seguido de una unidad de medida. Existe un comando especial y es: `\vfill` seguido de un texto y genera el espacio justo para que éste quede al final de la página.

Asi como es posible insertar espacios verticales también se puede insertar espacios horizontales, para ello usamo el comando `\hspace*{tamaño}`. De igual modo existe `\hfill` que coloca los espacios horizontales suficientes para que el texto que va seguido del comando quede al final de la línea, ejemplo:

Este espacio se hizo con `\hfill`

2.2.3. Comandos de configuración de palabras y letras

Para cambiar el estilo de letra podemos hacer uso de los siguientes comandos:

`\rm` cambia al estilo “roman”.

`\em` cambia al estilo “italic”.

`\bf` cambia al estilo “boldface”.

`\sl` *cambia al estilo “slanted”.*

`\sf` cambia al estilo “sans serif”.

`\sc` CAMBIA AL ESTILO “SMALL CAPS”.

`\tt` cambia al estilo “typewriter”.

El estilo por defecto es “roman” y “typewriter” es el estilo por defecto del ambiente verbatim. Para cambiar el estilo de letra necesitamos encerrar entre llaves las palabras o letras que se desea cambiar y luego de la primera llave escribir el comando respectivo, ejemplo: `{\bf cambio de estilo de letra}` que dá como resultado **cambio de estilo de letra**. Si no se encierra entre llaves cambia de estilo a todo el texto que va despues del comando.

Si queremos subrayar alguna letra palabra o alguna frase usamos el comando `\underline`, recordemos que se debe delimitar lo que deseamos subrayar por `{ }`. Otras variantes de este comando estan definidas en el estilo *ulem.sty* y son:

`\uline` Texto subrayado con un sola línea.

`\uuline` Texto subrayado con dos líneas.

`\uwave` Texto subrayado con una onda.

`\sout` ~~Texto subrayado con una línea que lo corta.~~

`\xout` ~~Texto cubierto con líneas oblicuas.~~

Podemos usar las comillas (”) y apóstrofes (’ ‘) libremente pues no son caracteres especiales. A continuacion una pequeña tabla con el uso de estos símbolos.

comando	resultado	comando	resultado
<code>\'a</code>	á	<code>\"á</code>	ä
<code>\'a</code>	à	<code>\= a</code>	ā
<code>\b a</code>	<u>a</u>	<code>\u a</code>	ǎ
<code>\â</code>	â	<code>\H a</code>	Ǻ
<code>\d a</code>	ḏ	<code>\t aa</code>	aâ
<code>\v a</code>	ă	<code>\c c</code>	ç
<code>\~ n</code>	ñ	<code>\'i</code>	í
<code>\'{\i}</code>	í	<code>\'I</code>	Í

Es posible generar tontos y acentos para las palabras y/o letras en el ambiente matemático, para tal propósito podemos utilizar los comandos de la siguiente tabla

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

De igual modo existen diferentes tamaños de letra como se muestra en la siguiente tabla:

Comando \tamaño	10pt regular	11pt regular	12pt regular
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Recuerde colocar entre llaves el texto que desea cambiar de tamaño, de lo contrario cambiará el tamaño especificado hasta el final del documento.

Medidas en L^AT_EX

Como es de esperarse, existen diferentes tipos de medida en L^AT_EX y los más comunes son:

- in - pulgadas.
- mm - milímetros.
- cm - centímetros.
- pt - puntos (aproximadamente 1/72 de pulgada).
- em - aproximadamente el ancho de la letra “M” en el tipo de letra usado.
- ex - aproximadamente el alto de la letra “x” en el tipo de letra usado.

Nota: es importante destacar que L^AT_EX es **sensitivo** a las letras mayúsculas y minúsculas, por lo que se debe tener cuidado en escribir los nombres correctamente, pues se puede u obtener un error u otro resultado.

2.3. Ambientes

Un ambiente esta definido de la siguiente forma:

```
\begin{nombre del ambiente}  
...  
\end{nombre del ambiente}
```

Existen muchísimos ambientes la gran mayoría definidos por los usuarios de L^AT_EX y otros que forman parte del sistema, a continuación mostramos algunos de ellos:

2.3.1. document

Es el ambiente principal de un documento en \LaTeX es:

```
\begin{document}
```

texto del documento y comandos

```
\end{document}
```

y es en este ambiente donde escribiremos todo el cuerpo del documento. Lo que va antes de este ambiente se le llama encabezado y en él se le indica al compilador el formato del documento, el estilo y algunos otros detalles que se le deben dar al compilador para obtener lo que deseamos. Es muy usual en que en el encabezado también se escriban las nuevas definiciones tanto de comandos como de ambientes así como es recomendable escribir los detalles de la caratula en este lugar del documento.

2.3.2. center

Usando este ambiente podemos centrar párrafos, figuras, tablas y cualquier otro objeto que se defina dentro de él.

```
\begin{center}
```

texto

```
\end{center}
```

Centrando este texto

2.3.3. verbatim

Este ambiente es muy usado cuando se necesita escribir códigos en algún lenguaje de programación o escribir texto con ubicación específica de líneas y

párrafos. Este ambiente presenta un problema pues como en nuestro idioma se tildan algunas palabras, esas tildes no podrán ser representadas y no es posible usarlo con texto en el que se usen tildes.

```
\begin{verbatim}
```

texto

```
\end{verbatim}
```

a continuación un ejemplo usando este ambiente para escribir un programa en C.

```
#include<stdio.h>
int a,b;
main() {
    a=1;
    b=2;
    c=a+b;
    printf("el resultado de la suma es % d",c);
}
```

2.3.4. verse

L^AT_EX posee un ambiente especial para escribir poesía y es el siguiente:

```
\begin{verse}
```

texto

```
\end{verse}
```

A continuación un fragmento de una de las poesías de un famoso poeta peruano César Vallejo.

Dios mío, estoy llorando el sér que vivo;
me pesa haber tomádote tu pan;
pero este pobre barro pensativo

no es costra fermentada en tu costado:
¡tú no tienes Marías que se van!
...

César Vallejo

L^AT_EX automáticamente justifica el texto, más en algunos casos no se desea que el texto este justificado para ello colocamos `\` en el lugar donde deseamos cortar la línea.

2.3.5. quotation

Cuando necesitamos hacer citas usamos este ambiente, el cual centra el texto y lo coloca a la misma distancia tanto del margen izquierdo como del derecho (la distancia es establecida por defecto).

```
\begin{quotation}
```

texto

```
\end{quotation}
```

El pensamiento es la principal facultad del hombre, y el arte de expresar los pensamientos es la primera de las artes.

usando el comando `\noindent` al comienzo de la primera línea la cita no quedará indentada; sino usamos este comando queda en la forma siguiente:

El vino es una cosa maravillosamente apropiada para el hombre si, en tanto en la salud como en la enfermedad, se administra con tino y justa medida.

2.3.6. minipage

Algunas veces necesitamos resaltar un texto, pero que no sea en forma de cita, que tiene las distancias establecidas por defecto, sino que lo queremos a un ancho personalizado, para ello usamos el siguiente ambiente:

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

texto

```
\end{minipage}
```

Con sus ojos muy grandemente llorando tornaba la cabeza y estábanlos mirando: vio las puertas abiertas, los postigos sin candado, las perchas vacías sin pieles y sin mantos y sin halcones y sin azores mudados. Suspiró mío Cid triste y apesadumbrado. Habló mío Cid y dijo resignado: «¡Llor a ti, señor Padre, que estás en lo alto! Esto me han urdido mis enemigos malos».

en el ejemplo usamos con un ancho de 4cm y una alineación a la izquierda “l”, pero podemos usar también “c” para centrala, “r” para alinearlo a la derecha, “t” para colocarlo en la parte superior o “b” para colocarlo en la parte inferior.

2.3.7. raggedright y raggedleft

Si deseamos que el texto quede alineado a la izquierda o derecha usamos los ambientes:

```
\begin{raggedright}
```

texto

```
\end{raggedright}
```

o

```
\begin{raggedleft}
```

texto

```
\end{raggedleft}
```

a continuación un ejemplo de alineación a la derecha

Tanta fue la congoja que me infundió
el espanto que de sus ojos salía,
que perdí la esperanza de la altura.
Y como aquel que goza en atesorar,
y llegado el tiempo en que perder le toca,
su pensamiento entero llora y se contrista;
así obró en mi la bestia sin paz,
que, viniéndome de frente, poco a poco,
me repelía a donde calla el Sol.

Para lograr el mismo efecto en los párrafos o textos, podemos usar los ambientes *flushright* y *flushleft* respectivamente. Estos ambientes sufren ligeras modificaciones dependiendo de la clase que se use.

2.3.8. tabbing

Este ambiente es usado para escribir texto con tabulaciones específicas

```
\begin{tabbing}
----- \= --- \= \kill
Texto \> Texto \> Texto \\
\> Texto \> Texto \> Texto

\end{tabbing}
```

nuevamente tomamos un programa pero esta vez escrito en Fortran para mostrar un ejemplo de como usar el ambiente.

```
C *** Este es un programa en fortran
      DO 200J=1,15
          Y(J) = A*X(J)
          SUM = SUM + Y(J)
200 CONTINUE
```

2.3.9. Ambientes de enumeración

L^AT_EX nos brinda tres opciones para crear listas la primera de ella es “enumerate”:

```
\begin{enumerate}
  \item primer item de una lista de ítemes
    \begin{enumerate}
      \item primer item de una lista de ítemes anidada
      \item segundo item de una lista de ítemes anidada
    \end{enumerate}
  \item tercer item de una lista de ítemes
\end{enumerate}
```

Las dos listas siguientes son: “itemize” y “description”. Para hacer uso

de ellas solo cambiamos la palabra enumerate por itemize o description, a continuación un ejemplo con cada una de ellas:

1. Primer item de la lista enumerate
2. Segundo item de la lista enumerate
 - a) Primer item en la lista anidada de enumerate
 - b) Segundo item en la lista anidada de enumerate
3. Tercer item de la lista enumerate
 - Primer item de la lista itemize
 - Segundo item de la lista itemize
 - * Primer item en la lista anidada de itemize
 - ▣ Segundo item en la lista anidada de itemize
 - Tercer item de la lista itemize

El ambiente “itemize” tiene hasta cuatro niveles de anidamiento, cada nivel se distingue por una marca predeterminada, por ejemplo para el primer nivel se marca con un punto grande, pero sí el usuario lo desea puede cambiarlo para tal propósito usa el siguiente comando, `\renewcommand {\labelitemi} {character}`, es importante recordar que si deseamos cambiar el marcador de la anidación del segundo nivel debemos usar `\labelitemii`, para la tercera `\labelitemiii` y para la cuarta `\labelitemiv`. Una forma más simple es escribir entre corchetes el nuevo marcador: `\[nuevo marcador]`, como se usó para la lista anidada del ejemplo.

Como su nombre lo indica el ambiente “description” es bueno para hacer descripciones de elementos o palabras

Mercurio es el planeta más cercano al Sol y recibe unas diez veces más energía solar que la Tierra.

Venus es el astro más brillante del cielo, después de la Luna.

Tierra tiene una forma aproximada esférica, con un diámetro medio de 12.750 km, con un leve achatamiento en la dirección de sus polos.

Luna el aspecto que presenta en el cielo cambia continuamente en un intervalo de 29,5 días.

Marte resulta tan notable por su color y brillo como por sus movimientos respecto de las estrellas.

Fobos la más grande de las lunas de marte.

Deimos la más distante de las lunas de marte.

2.3.10. Ambiente matemático

Si hay algo que L^AT_EX hace con la mejor calidad son las ecuaciones y fórmulas y como es de esperarse existe un ambiente matemático, este ambiente se puede utilizar de diversas formas, tales como:

```
\begin{math}
```

fórmulas

```
\end{math}
```

“math” se usa para escribir fórmulas en el texto, lo cual se puede sustituir por \$ *fórmulas* \$. Otra opción es “displaymath” que se usa para escribir una línea de ecuación no numerada, esto también se puede sustituir por \$\$ *fórmulas* \$\$. Se usa “equation” si se quiere enumerar las ecuaciones, “eqnarray” es usado para escribir varias líneas de ecuaciones.

A continuación algunos ejemplos usando los ambientes descritos.

Usando el ambiente “math” podemos escribir:

El teorema de Pitágoras. $x^2 + y^2 = h^2$.

usando el ambiente “displaymath” podemos escribir el mismo ejemplo

El teorema de Pitágoras.

$$x^2 + y^2 = h^2.$$

nuevamente el mismo ejemplo usando “equation”

El teorema de Pitágoras.

$$x^2 + y^2 = h^2. \tag{2.1}$$

ahora usaremos el ambiente “eqnarray” para escribir tres ecuaciones:

$$\Phi(x) \geq 5 - 2x \quad (x \leq 2,5). \tag{2.2}$$

$$\mathcal{F}(x) + \mathcal{G}(x) \leq a + b + c + d + e + f + g + h + i + j + k + l + m. \tag{2.3}$$

$$\sum_{n=0}^{\infty} \frac{1}{2^n} = 2.$$

La última ecuación no se enumeró porque al final de ésta se incluyó el comando `\nonumber`. Si no se desea enumerar ninguna de las ecuaciones se usa “eqnarray*”. existe un ambiente bastante similar a este y es “align” el cual permite mayor flexibilidad en el manejo de las ecuaciones.

Como se puede observar el estilo por defecto en el ambiente matemático es el itálico, mas no se recomienda usar el ambiente matemático para escribir en itálico, puesto que los espacios son eliminados y se producen algunas otras anomalías; así, si escribimos `$ Universidad de Puerto Rico$` tendremos como resultado:

UniversidaddePuertoRico

donde vemos que los espacios se han eliminado y hay una separación entre las letras r y s.

Los subíndices y superíndices solo se pueden escribir en el ambiente matemático, en la forma siguiente:

`\base_{subíndice}`
`\base^{superíndice}`

Se pueden hacer combinaciones como la siguiente: x_a^y

Un aspecto importante en el ambiente matemático son las letras griegas, para imprimir una letra griega necesitamos usar lo siguiente `\cal texto` para obtener una versión caligráfica del texto, ejemplo:

$\mathcal{F}(x)$ es una función continua.

La mayor parte de los símbolos especiales forman parte del ambiente matemático. Así para hacer una flecha desde A hasta B, debes escribir:

A `\longrightarrow` B y obtendras $A \longrightarrow B$

Los símbolos `$` marcan la entrada y la salida del ambiente matemático.

A continuación una lista con algunos de los símbolos matemáticos.

Operaciones binarias

<code>\pm</code>	±	<code>\mp</code>	∓
<code>\setminusminus</code>	\	<code>\cdot</code>	·
<code>\times</code>	×	<code>\ast</code>	*
<code>\star</code>	★	<code>\diamond</code>	◇
<code>\circ</code>	○	<code>\bullet</code>	●
<code>\div</code>	÷	<code>\cap</code>	∩
<code>\cup</code>	∪	<code>\uplus</code>	⊕
<code>\sqcap</code>	⊓	<code>\sqcup</code>	⊔
<code>\triangleleft</code>	◁	<code>\triangleright</code>	▷
<code>\wr</code>	ℳ	<code>\bigcirc</code>	◯
<code>\bigtriangleup</code>	△	<code>\bigtriangledown</code>	▽
<code>\vee</code>	∨	<code>\wedge</code>	∧
<code>\oplus</code>	⊕	<code>\ominus</code>	⊖
<code>\otimes</code>	⊗	<code>\oslash</code>	⊘
<code>\odot</code>	⊙	<code>\dagger</code>	†
<code>\ddagger</code>	‡	<code>\amalg</code>	⊔

Relaciones

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

Operadores grandes

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

Combinando estos comandos podemos construir ecuaciones como:

$$\sqrt{b^2 - 4ac} \sqrt[3]{x + y}$$

$$\sum_{i=1}^n x_i \prod_{i=1}^n x_i$$

Los comandos `\overline{x}` produce \bar{x} y `\underline{x}` produce \underline{x} si en lugar de líneas deseamos colocar llaves se hace con los comandos `\overbrace{x}` y `\underbrace{x}` a continuación un ejemplo:

$$\underbrace{a + \overbrace{b + \cdots + y}^{24} + z}_{35}$$

Para escribir fracciones se usa el comando `\frac{ numerador }{denominador}` que imprime $\frac{\text{numerador}}{\text{denominador}}$, por ejemplo

$$x = \frac{y + z/2}{y^2 + 1}$$

$$\frac{e^{xy}}{1 + \frac{y}{x+1}}$$

Los delimitadores se construyen usando los comandos `\left` y `\right`, seguido de cada uno de ellos va `{`, `(`, `[` y `,`, `(`, `[` respectivamente, ejemplo:

$$Y = 1 + \left[\frac{x + 3}{2x^2 + 4x - 5} \right]$$

También es posible poner un subíndice y un superíndice al delimitador:

$$\left[\frac{x - 3}{2x + 7} \right]_0^1$$

Por último, es posible darle un tamaño preciso al delimitador, para ello usamos el comando `\rule{0mm}{12mm}`, ejemplo:

$$\left\| x = \frac{1 - y}{1 + y} \right\|$$

El comando `\stackrel{arriba}{abaja}` se usa para colocar un símbolos arriba de otro ejemplo:

$$A \xrightarrow{k_1} B$$

Cabe recordar que este comando se puede usar recursivamente.

Como habíamos visto, el estilo matemático es por defecto itálico pero se puede cambiar a otros estilos usando los comandos `\textstyle` y `\scriptstyle`, ejemplo:

$$y = x^2 \text{ y } = x^2$$

En el modo matemático es posible generar espacios (de medida positiva o negativa) y se pueden utilizar varios comandos ejemplos:

<code>\$ a b \$</code>	genera ab
<code>\$ a \! b \$</code>	genera \boldsymbol{ab}
<code>\$ a \, b \$</code>	genera $a b$
<code>\$ a \: b \$</code>	genera $a b$
<code>\$ a \quad b \$</code>	genera $a \quad b$
<code>\$ a \qquad b \$</code>	genera $a \qquad b$

Muchas veces se genera una ecuación en varias líneas, las líneas extras aparecen después del signo igual (=) o de la desigualdad o de cualquier otro símbolo, algunas veces se desea que estas líneas fueran impresas un poco más a la izquierda, esto se logra usando el comando `\lefteqn{(Lado izquierdo de la ecuación y el símbolo)}` en el ambiente `eqnarray` por ejemplo:

$$\begin{aligned} \mathcal{R}(y) + \int_0^y \Psi(x) dx = \\ a + b + c + d + e + f + g + \\ h + i + j + k + l + m + o + p + q \end{aligned}$$

Nota: No poner líneas en blanco dentro de este ambiente, pues esto genera errores muy difíciles de decifrar.

El ambiente matemático nos permite también imprimir matrices de la siguiente forma:

```

\begin{center}
\[
\left[ \begin{array}{lcr}
a & p+q & \alpha + \beta + \gamma \\
b+c & r+s+t & \delta + \epsilon \\
d+e+f & u & \zeta
\end{array} \right]
\end{array}
\]
\end{center}

```

$$\left[\begin{array}{lcr} a & p+q & \alpha + \beta + \gamma \\ b+c & r+s+t & \delta + \epsilon \\ d+e+f & u & \zeta \end{array} \right]$$

2.3.11. tabular

El ambiente de tabuladores (`tabular`) de \LaTeX nos da la posibilidad de organizar texto o símbolos en filas y columnas. El formato es el siguiente:

```

\begin{center}
\begin{tabular}{cols} \hline
(texto) & (texto) & (texto) & ... \\
(texto) & (texto) & (texto) & ... \\
\end{tabular}
\end{center}

```

El argumento `cols` especifica el formato de las columnas siguiendo la siguiente norma:

- l coloca el texto en el borde izquierdo.
- c coloca texto centrado.
- r coloca el texto en el borde derecho.
- l imprime una línea vertical del largo de la tabla.

Tabla de ejemplo		
Nombre	Cantidad de (acciones)	Descripción
CTC	10.000.000	Mantenimiento, modernización de las líneas telefónicas y de telefonía celular del País
EMOS	5.000.000	Mantener un aprovisionamiento permanente de agua a la nación

Cuadro 2.1: Tabla de ejemplo

`||` imprime una doble línea vertical.

`&` es el separador de columnas.

`\\` es el separador de filas.

el comando `\arg` es opcional, el comando `{\hspace{largo}}` inserta un espacio horizontal en la tabla en el lugar donde se coloque, cuando se desea colocar título a más de una columna usamos `\multicolumn{num}{col}{titulo}` y finalmente los comandos `\hline` y `\vline` son para incluir una línea horizontal y vertical respectivamente, ejemplos:

Adams, John	I17	S28	87
Adams, Cynthia	E38	N17	68
Williams, George	I25	N7	123
Williams, Joan	I13	N38	72

\LaTeX también nos facilita la enumeración de tablas, permitiéndonos colocar una breve descripción de las mismas usando el comando `\caption{descripción }`.

2.3.12. figure

Al igual que el ambiente matemático este ambiente es bastante versátil pues nos permite hacer cosas increíbles con las figuras, ya sea que se importen

o que se dibujen, su formato es el siguiente:

```

\begin{figure}
\begin{center}
\includegraphics[medidas]{figura.*}
\caption{ descripción de la figura } \label{figura-ref}
\end{center}
\end{figure}

```

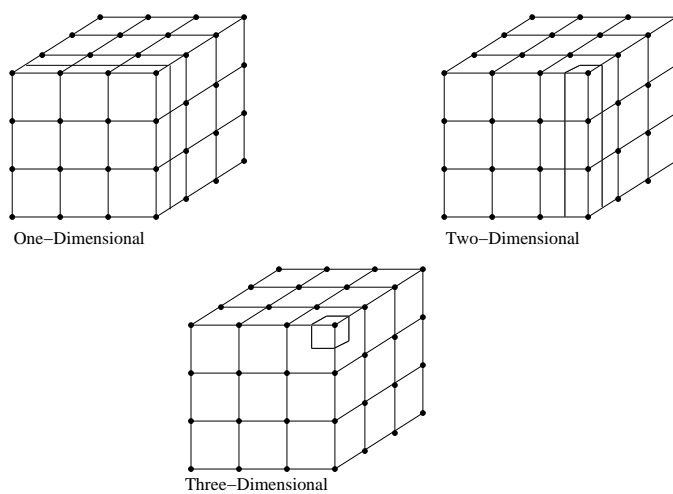
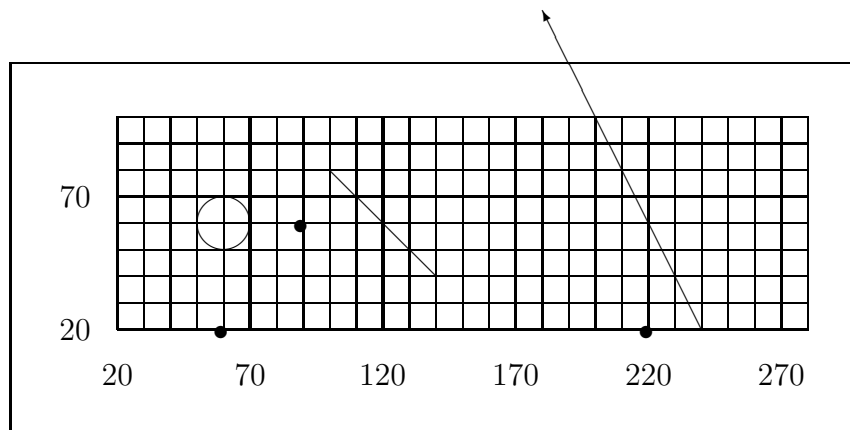


Figura 2.1: Particiones de un cubo

y aquí una muestra de como dibujar usando \LaTeX



2.3.13. Teoremas

L^AT_EX nos permite no sólo hacer uso de los ambientes predefinidos para un estilo o paquete específico, sino que nos brinda la opción de crear nuestros propios ambientes, en particular para definir un teorema, corolario, proposiciones, etc. para ello debemos usar el siguiente comando:

```
\newtheorem{nombre del ambiente}{título}[criterio de enumeración]
```

por ejemplo:

```
\newtheorem{teoremita}{Teorema}[chapter]
```

esto creará un nuevo ambiente llamado teoremita, cuyo título será Teorema y estará seguido de un número, en este caso, que hará referencia al capítulo en el que se está trabajando, seguido del número correspondiente, el uso de este nuevo ambiente se hace como cualquier otro, es decir:

```
\begin{teoremita}
```

texto y/o fórmulas

```
\end{teoremita}
```

A continuación un ejemplo usando el ambiente descrito arriba.

```
\begin{teoremita}
```

Sea α una parametrización natural de una curva cuya torsión τ no se anula y que está contenida en una esfera. Demostrar que dicha curva no tiene puntos de inflexión y que la función $\frac{1}{k^2} + \frac{\partial \tau}{\partial k}$ es constante.

```
\end{teoremita}
```

vemos que el resultado de escribir esto, nos muestra el título en negrita,

seguido de 2.1 donde 2 se refiere al capítulo al que pertenece y 1 a que es el primer teorema que escribimos, recordemos que L^AT_EX enumera los objetos.

Teorema 2.1 *Sea α una parametrización natural de una curva cuya torsión τ , no se anula y que está contenida en una esfera. Demostrar que dicha curva no tiene puntos de inflexión y que la función*

$$\frac{1}{k^2} + \frac{\partial_\tau k}{\tau k^2}$$

es constante.

Del mismo modo podemos definir las proposiciones, corolarios, etc

2.4. Referenciando páginas, figuras, tablas y ecuaciones

Para referenciar una página desde cualquier lugar del texto debemos usar los siguientes comandos:

`\label{ key }` y `\pageref{ key }`

El primer comando sirve para crear una la clave a la página, sección, ecuación, teorema, etc. que desees referenciar y el segundo sirve para hacer, en este caso, referencia a la página, en el resultado, este comando es sustituido por el número de la página.

Del mismo modo para referenciar figuras escribimos la clave correspondiente (por ejemplo: `\label{figure-key}`) en la figura y para hacer referencia a la figura escribimos el comando `\ref{ figure-key }`, para las secciones, tablas, ecuaciones, etc. usamos los mismos comandos.

Es bien importante llevar un control de las referencias (no repetir las claves) para evitar referencias cruzadas. Se recomienda colocar una clave que tenga que ver con el nombre del objeto.

2.5. Comandos personalizados

Al igual que \LaTeX nos permite crear ambientes personalizados, tambien nos permite crear nuestros propios comandos, para lograrlo usamos el comando:

```
\newcommand{\nombre}{definición}
```

Es bastante útil cuando en un documento repetimos mucho una palabra difícil de escribir o una frase o una ecuación, ejemplo:

```
\newcommand{\pitagoras}{a^2=b^2+c^2}
```

siempre que escribamos `\pitagoras` nos escribirá:

$$a^2 = b^2 + c^2$$

Una variante mas compleja es:

```
\newcommand{\nombre}[número]{definición}
```

donde el parámetro número indica el numero de argumentos que hay que incluir, ejemplo:

```
\newcommand{\cuadratica}[3]{\frac{-#2+\sqrt{#2^2-4*#1*#3}}{2*#1}}
```

y el resultado de colocar:

```
\newcommand{\cuadratica}{a}{b}{c}
```

será:

$$\frac{-b + \sqrt{b^2 - 4 * a * c}}{2 * a}$$

Capítulo 3

Partes de un Documento

\LaTeX provee varias formas de particionar un documento, tales como:

`\part{ }` (Opcional. Utilizado frecuentemente en el estilo book.)

`\chapter{ }` (Sólo en los estilos book y report)

`\section{ }`

`\subsection{ }`

`\subsubsection{ }`

`\paragraph{ }`

`\subparagraph{ }`

El encabezado de las secciones va entre llaves y es recomendable dejar una línea en blanco antes y después del comando. Es preciso recordar que \LaTeX enumera automáticamente, los capítulos, secciones y subsecciones, si no se desea que enumere alguna sección hay que colocar un asterisco después del nombre de la sección, ejemplo: `\chapter*{ }`. Recordemos también que \LaTeX coloca las páginas y nombres de las secciones correspondientes en el índice.

Como explicamos anteriormente para referenciar una sección o subsección, los comandos que se usan son los mismos que para una página, la diferencia es que el comando `\label {key}` hay que colocarlo justo después del comienzo de la sección o subsección.

3.1. Referencias Bibliográficas

Para crear las referencias bibliográficas, usamos el ambiente:

```
\begin{thebibliography}{numero}
\bibitem{name-key} autor, título de la referencia, editorial, año
\end{thebibliography}
```

el “name-key” nos permitirá hacer la referencia, en algún lugar del documento, unicamente a este item, un ejemplo:

```
....
Harla escuch\’o atentamente, absorto ante la visi\’on de un
poderoso c\’{\i}rculo en el Tiempo...
Para leer la historia completa vea \cite{asimov}....
...
\begin{thebibliography}
\bibitem{asimov} Asimov,I.,{\it El fin de la eternidad},
Ediciones Orbis, S.A., 1977.
\bibitem{marquez} M\’arquez,P.,{\it Social enterprise},
Ediciones IESA, 2004.
\bibitem{otra} ... ..
\end{thebibliography}
```

el resultado de esta porción de código \LaTeX se muestra en la siguiente página.

....
Harla escuchó atentamente, absorto ante la visión de un
poderoso círculo en el Tiempo...
Para leer la historia completa vea [1]....
...

Bibliografía

- [1] Asimov,I., *El fin de la eternidad*, Ediciones Orbis,
S.A., 1977.
- [2] Márquez,P.,*Social enterprise*, Ediciones IESA,
2004.
- [3]

Capítulo 4

Caracteres reservados de LaTeX

Es importante que como usuario de cualquier tipo language se conozca los caracteres reservados o palabras reservadas para evitar errores. Así en LaTeX, es necesario conocer cuales caracteres son usados como comandos o delimitadores por el compilador de LaTeX, estos caracteres son:

\$ % & - { } ~ ^ \

Para poder incluir estos caracteres dentro de tu documento, como parte del texto se debe anteponer un \ al caracter que se desea imprimir, por ejemplo:

Para obtener #, escribir \#.

Para obtener \$, escribir \\$.

Para obtener %, escribir \%.

Para obtener &, escribir \&.

Para obtener -, escribir \-.

Para obtener { o }, escribe \{ o \}.

Para obtener ~, escribir \~.

Para obtener ^, escribir \^.

Para obtener \, escribir \textbackslash.

Capítulo 5

Compilación de archivos \LaTeX

5.1. Compilando archivos \LaTeX en Linux

Para compilar archivos \LaTeX en linux, necesitas cargar un terminal o consola, ir al directorio donde tienes tus archivos y ejecutar la siguiente instrucción

```
/home/freddy> latex tutorial.tex
```

Esta instrucción genera cuatro archivos adicionales:

- tutorial.log en el cual se almacena un informe de la compilación realizada.
- tutorial.aux este archivo auxiliar permite recuperar objetos o parte del documento que se pierde por mala edición.
- tutorial.dvi este es uno de los más importantes pues se usa tanto para convertir a postscript, pdf y otros así como para imprimir el documento de \LaTeX . En linux se puede visualizar usando el comando `xdvi`, recuerde que solo se puede usar en ambiente gráfico y su sintaxis es la siguiente:

```
/home/freddy> xdvi tutorial.dvi
```

- tutorial.toc este archivo contiene la información del índice general, se produce al incluir el comando `\tableofcontents`.
- tutorial.bib este archivo contiene la información de la bibliografía referenciada en el documento.

Después de ejecutar la instrucción “`xdvi tutorial.dvi`” se abrirá una nueva ventana en la que podremos ver el documento tal como se obtendrá de la impresora, en la parte derecha de esta ventana, se encuentra una columna con los comandos respectivos para el manejo del documento. Para convertir el archivo dvi a un archivo ps (Postscript) necesitamos ejecutar el siguiente comando:

```
/home/freddy> dvips -o tutorial.ps tutorial.dvi
```

Luego para pasarlo a pdf ejecutamos la siguiente instrucción

```
/home/freddy> ps2pdf tutorial.ps tutorial.pdf
```

Quedando listo para imprimirse.

5.2. Compilando archivos \LaTeX en Windows

Para la compilación en windows, si no usamos el WinEdt, necesitamos de abrir una ventana de DOS, y luego de ubicarnos en el directorio donde están nuestros archivos de \LaTeX ejecutamos las orden de compilación que vimos inicialmente. Luego de generado el archivo dvi, necesitamos convertirlo a pdf, para ello podemos abrirlo usando GSview y hacer la conversión correspondiente. Si tiene instalado WinEdt todo el proceso se lleva a cabo haciendo “click” en los íconos respectivos.

5.3. Manejo de Errores

La depuración de errores en la compilación es un paso muchas veces difícil de hacer, por tanto en ésta sección proveémos de algunos “tips” para su

manejo. Los errores que se presentan al momento de la compilación de un documento tienen la forma siguiente por ejemplo:

```
LaTeX error. See LaTeX manual for explanation.  
Type H ;return;for immediate help.  
!Environment displaymat undefined. \Latexerr ...for immediate help.}\errmessage  
{#1} \endgroup  
1.21 \begin{displaymat}  
?
```

El signo de exclamación en la tercera línea es el indicador del error, y es seguido por el mensaje de error. Luego la cuarta y quinta línea indican el mismo error en lenguaje de bajo nivel. La línea seis indica la posición en que ocurrió el error, en este caso, el error ocurrió en la línea 21. Luego el signo ? es el prompt, ya sea de la consola de Linux o la ventana de DOS de Windows, es recomendable terminar el proceso y hacer la corrección respectiva, para llevar a cabo esta acción debemos escribir x para finalizar la compilación.

Existen otras opciones, si se escribe “H” se obtiene ayuda, si se escribe “r” se ejecuta la compilación sin parar, si escribimos “q ” ejecuta la compilación lentamente, si escribimos “i” tenemos la posibilidad de insertar texto.

Las advertencias de overfull y underfull (warnings), generalmente no son de cuidado, pero muchos expertos no les gusta y mejoran el texto para evitarlos. Pues a veces revisando el archivo dvi se podrían ver algunos errores que ocasionaron los warnings.

Cuando el error es:

* significa que el documento le falta el comando `\end{document}`.

! Missing inserted Este error puede tener varios orígenes puede ser que se ingreso al ambiente matemático de una forma y se salió dando otro comando distinto al que correspondía para cerrarlo o puede ser que trataras de generar ambientes como el de arreglos fuera del ambiente matemático o que trataras de obtener caracteres (que sólo se pueden obtener en el ambiente matemático) fuera de él o tratar de imprimir el signo \$ de forma inapropiada sin el signo \ antes o por poner línea en blanco en el ambiente matemático.

! Missing { (or)} inserted posiblemente se omitió una llave al abrir o cerrar un ambiente.

! Missing number, treated as zero es posible que omitieras un número donde Latex esperaba uno.

! Undefined control sequence es probable que un comando este mal escrito. Pues es muy común no dejar un espacio en blanco después de los comandos como `\bf` o `\alpha`.

! Extra alignment tab has been changed colocaste más `&` o menos de los que se suponía que debías colocar, u omitiste el `\\` en la línea anterior.