

Segunda Edición

Alexánder Borbón A.
Walter Mora F.

Instituto Tecnológico de Costa Rica

Edición de textos científicos

L^AT_EX 2025

$$a \xrightarrow{f} b$$

$$\lim_{x \rightarrow 0} f(x)$$

$$\begin{pmatrix} a \\ b \end{pmatrix}$$

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} a_i b_j$$

$$\prod_{\substack{i=0 \\ i \neq k}}^n \frac{w_i}{(w_i - w_k)}$$



Revista digital Matemática, Educación e Internet

www.tec-digital.itcr.ac.cr/revistamatematica/

Alexánder Borbón A., Walter Mora F.

Edición de Textos Científicos

LaTeX

*Composición, Diseño Editorial, Gráficos,
Inkscape, Tikz y Presentaciones Beamer*

Compartir



https://tecdigital.tec.ac.cr/servicios/revistamatematica/material_didactico/libros/



Revista digital

Matemática, Educación e Internet. (<https://tecdigital.tec.ac.cr/servicios/revistamatematica/>).

Copyright©Revista digital Matemática Educación e Internet (<https://tecdigital.tec.ac.cr/servicios/revistamatematica/>).
Correo Electrónico: wmora2@gmail.com aborbon@itcr.ac.cr
Escuela de Matemática
Instituto Tecnológico de Costa Rica
Apdo. 159-7050, Cartago
Teléfono (506)25502225
Fax (506)25502493

Mora Flores, Walter.

Edición de Textos Científicos con $\text{\LaTeX}$. Composición, Gráficos, Inkscape y Presentaciones Beamer/Walter Mora F.

Alexánder Borbón A. – 2da ed.

– Escuela de Matemática, Instituto Tecnológico de Costa Rica. 2016.

311 pp.

ISBN 978-9977-66-227-5

1. TeX. 2. Composición tipográfica-automatizada 3. Tipos - símbolos matemáticos.

Licencia.

Revista digital

Matemática, Educación e Internet.

<https://tecdigital.tec.ac.cr/servicios/revistamatematica/>



Este libro se distribuye bajo la licencia Creative Commons: Atribución-NoComercial-SinDerivadas CC BY-NC-ND (la “Licencia”). Usted puede utilizar este archivo de conformidad con la Licencia. Usted puede obtener una copia de la Licencia en <http://creativecommons.org/licenses/by-nc-nd/3.0/>. En particular, esta licencia permite copiado y distribución gratuita, pero no permite venta ni modificaciones de este material.

Límite de responsabilidad y exención de garantía: El autor o los autores han hecho su mejor esfuerzo en la preparación de este material. Esta edición se proporciona “tal cual”. Se distribuye gratuitamente con la esperanza de que sea útil, pero sin ninguna garantía expresa o implícita respecto a la exactitud o completitud del contenido.

La Revista digital Matemáticas, Educación e Internet es una publicación electrónica. El material publicado en ella expresa la opinión de sus autores y no necesariamente la opinión de la revista ni la del Instituto Tecnológico de Costa Rica.

ÍNDICE GENERAL

	PRÓLOGO	1
1	L^AT_EX: PRIMEROS PASOS	2
1.1	¿Qué es LaTeX?	2
1.2	Comenzar: Distribuciones T_EX y editores	3
2	ACCIONES EN UNA SESIÓN CON L^AT_EX	8
2.1	Editar, compilar y ver el resultado.	8
2.2	Compilar	9
2.3	Usar paquetes adicionales	10
2.4	Ajustes con el paquete Geometry	11
2.5	Idioma	12
	Si hay conflictos con el paquete babel...	13
	Plantilla para este capítulo.	14
2.6	Otros ajustes manuales del documento	15
3	ESCRITURA DE TEXTO NORMAL	18
3.1	Tipos y tamaños de fuentes.	18
	Caracteres especiales.	18
	Algunos tipos de fuentes (fonts).	18
	Tamaños de letras.	19
3.2	Párrafos y efectos especiales.	20
	Centrar	21
	Espacio horizontal y vertical	21
	Cajas	22
	Texto en columnas: multicol, minipage y parbox	22
	Líneas y otros efectos de texto	28
	Notas al pie de página.	29
	Texto como en la pantalla	29
3.3	Color y cajas con color	30
	Notas en el margen	33
3.4	Enumerado automático.	33
	Enumeración usando el paquete TikZ.	40
3.5	Título, contenido, secciones y bibliografía	42
	Índice alfabético	45
3.6	Modulación	48
3.7	(*) Más cosas sobre las fuentes.	48
	Las fuentes y sus atributos	49
	Usando las fuentes del sistema con XeLaTeX .	52

4	TEXTO EN MODO MATEMÁTICO	56
4.1	Potencias, subíndices y superíndices	57
4.2	Tamaño natural	57
4.3	Fracciones y expresiones de dos niveles	58
4.4	Raíces	59
4.5	Tres puntos consecutivos	60
4.6	Delimitadores	60
4.7	LLaves y barras horizontales	62
4.8	Acentos y “sombreros” en modo matemático	63
4.9	Vectores	63
4.10	Negritas en modo matemático	63
4.11	Espacio en modo matemático	64
4.12	Centrado	64
4.13	Entorno equation. Contadores	65
4.14	Arreglos	66
4.15	Matrices	69
4.16	Alineamiento	70
4.17	Tablas de símbolos matemáticos frecuentes	74
4.18	Cómo hacer nuevos Comandos.	76

5	TABLAS Y OBJETOS FLOTANTES	83
5.1	Tablas de más de una página	85
5.2	Objetos flotantes: Entornos figure y table	88
5.3	Fuentes en tabular.	90
5.4	Color en tablas.	90
5.5	Rotación de texto en celdas.	92
5.6	Unir celdas.	94
5.7	Espacio vertical en las filas.	95
5.8	Ancho de las columnas	98
5.9	Modo matemático con tabularx	102
5.10	Escalar una tabla	103
5.11	Personalizar las leyendas (“caption”)	104
5.12	Tablas con TikZ y tcolorbox	105
5.13	Generador automático de tablas	107

6	INSERTAR GRÁFICOS Y FIGURAS	110
6.1	Introducción	110
6.2	Compilando con PDFLATEX	111
6.3	Compilando con LaTeX. Figuras .eps	116
6.4	De nuevo: Paquete float	118
6.5	Paquete subfig	118
6.6	Los ambientes wrapfigure y floatflt	119
6.7	Alinear texto y figuras	121

7	EDICIÓN DE GRÁFICOS Y FIGURAS	127
7.1	Gráficos en formato vectorial	127
7.2	Inkscape	128
7.3	Extensión TexText para Inkscape	128
	Opciones	131
7.4	Edición adicional de figuras con Inkscape	132
8	CITAS BIBLIOGRÁFICAS CON BIBTEX	139
8.1	Entorno thebibliography	139
8.2	BibTeX	140
8.3	Gestor de referencias bibliográficas	141
	JabRef	141
	Zotero	145
9	DISEÑO EDITORIAL	147
9.1	Bajar la carga cognitiva	147
9.2	Amenidad: Los Cuatro Principios Básicos	149
	Proximidad.	149
	Alineamiento.	149
	Repetición.	150
	Contraste.	150
9.3	Legibilidad: Cómo escoger las fuentes.	151
9.4	Color	152
9.5	Editar un PDF con Inkscape	153
10	ENTORNOS Y CAJAS	156
10.1	Entornos simples	156
10.2	Entornos con xparse	159
10.3	Crear figuras nativas con TikZ	162
	Comando draw	163
	Nodos	165
	Nodos y flechas.	167
10.4	Cajas con el paquete tcolorbox	169
10.5	Capas (overlays)	175
10.6	Cajas con “newenvironment”	186
11	PERSONALIZAR EL DOCUMENTO	191
11.1	Entornos	191
11.2	Personalizar secciones	191
11.3	Personalizar capítulos	197
11.4	Personalizar páginas de título	201
11.5	Tablas de contenido	202
11.6	Encabezados de página	204

11.7	Insertar una portada	208
11.8	Listas de ejercicios con solución	209
	Paquete answers	209
	Paquete “ans”	212
11.9	Plantillas LaTeX para libros	213
	Otras Plantillas (templates) $\text{\LaTeX}$	216
11.10	Libro con algoritmos y programas	216
	Paquete algorithm2e (versión 2020.)	216
11.11	Color para el código	219
	Paquete Listings	219
	Paquete minted	221
	Paquete verbmments	225
11.12	LaTeX, R y Knitr	226

12 PRESENTACIONES CON BEAMER 232

12.1	Introducción	232
12.2	Un documento Beamer	232
12.3	Marcos (frames)	235
12.4	Velos (overlays)	235
12.5	Comando pause	238
12.6	Entornos para teoremas, definición, etc.	239
12.7	Blocks.	240
12.8	Opción fragile	241
12.9	Entornos para código de programas	241
	Entorno semiverbatim	241
	Iluminar código de lenguajes de programación.	243
12.10	Beamer y el paquete algorithm2e	244
12.11	Gráficos	245
12.12	Ligas y botones.	246
12.13	Efectos de Transición. Color	247
12.14	Ligas a Documentos Externos	248
12.15	Animaciones	249
12.16	Multicolumnas.	249
12.17	Color y otros ajustes	251

13 POSTERS Y TRÍPTICOS (BROCHURES) 253

13.1	Introducción	253
13.2	Posters	253
13.3	Trípticos (Panfleots o Brochures)	258
	Trípticos con beamer	258
	Trípticos con leaflet	261

14 TRADUCIR LATEX A HTML Y GENERAR UNA WEB INTERACTIVA 268

14.1	Make4ht	268
	Una plantilla.tex configurada	269

Compilar esta plantilla	271
Un sitio web de ejemplo	272

15 ALGUNOS PRINCIPIOS BÁSICOS PARA EXPLORAR CÓDIGO (L^A)T_EX **275**

15.1 Variables (registros)	275
15.2 Condicionales	284
15.3 Ciclos	289
15.4 Macros	292

BIBLIOGRAFÍA 298

Prólogo

Este texto cubre aspectos básicos e intermedios sobre composición tipográfica LaTeX, diseño editorial, presentaciones Beamer, edición adicional de gráficos y figuras con Inkscape y Tikz y entornos con los paquetes xparse y con tcolorbox. También se desarrollan tópicos que tienen que ver con paquetes especiales. Algunas veces la descripción se hace “por ejemplos” y/o usando plantillas de código, dada la vastedad del tema (la documentación de algunos paquetes sobrepasa las mil páginas). Los capítulos que se han incluido son los típicos más frecuentes en la edición de libros y artículos sobre matemáticas, educación, software y programación, según nuestra experiencia. Incluye nuevos paquetes y nuevos comandos que resuelven problemas cotidianos de edición de textos matemáticos de una manera más sencilla. Ahora se consideran nuevas cosas en diseño editorial e infografía y una nueva presentación de los entornos (basada en la versión más reciente de varios paquetes: TikZ, xparse y tcolorbox).

En el capítulo 10 se incluye una colección de plantillas LaTeX que “encapsulan” las grandes cantidades de código que se van necesitando para hacer libros atractivos y con aspecto profesional, incluida la plantilla LaTeX de este libro. El capítulo 15 es una exposición de principios básicos de programación **(La)TeX**, para explorar y/o modificar el código que se encuentra en los sitios de preguntas y respuestas para resolver problemas de edición, tal como [TeX - LaTeX Stack Exchange](https://tex.stackexchange.com/)

El código de todos los ejemplos y de cada trozo de código de este libro se puede descargar en el sitio web <https://tecdigital.tec.ac.cr/servicios/revistamatematica/Libros/LaTeX/index.htm#EjemplosLibro>

Este texto se ha usado en algunos cursos en el Instituto Tecnológico de Costa Rica y se usa frecuentemente en la revista digital de Matemática, Educación e Internet en la edición de artículos y libros. Agradecemos a todas las personas que nos han ayudado con la lectura detenida del libro, señalando errores en el texto y el código, inconsistencias, sugiriendo nuevas secciones y por todos sus comentarios.

Cartago, 2025.

W. MORA, A. BORBÓN.

1.1 ¿Qué es LaTeX?

“TeX is intended for the creation of beautiful books - and especially for books that contain a lot of mathematics”.

Donald Knuth



Donald Knuth, 1938—

El sistema **TeX** (se pronuncia [tej]) fue diseñado y desarrollado por Donald Knuth en la década del 70. Es un sofisticado programa para la composición tipográfica de textos *científicos* y es la mejor opción disponible para edición de textos con contenido matemático tales como artículos, reportes, libros, etc. **TeX** es en la práctica un estándar para publicaciones científicas en áreas como matemática, física, computación, etc. **LaTeX** es un conjunto de macros **TeX** preparado por Leslie Lamport. **LaTeX** es software libre y se distribuye bajo la [Licencia Pública del Proyecto LaTeX](#).

En general, solo necesitamos editar texto y algunos *comandos* y **LaTeX** se encarga de componer automáticamente el documento de acuerdo a la clase de documento.

A diferencia de un procesador de textos, con **LaTeX** tenemos un control más fino sobre cualquier aspecto tipográfico del documento. .

Comandos. Un documento **LaTeX** puede tener texto ordinario junto con texto en *modo matemático*. Los comandos vienen precedidos por el símbolo “\” (barra invertida).

LaTeX formatea las páginas de acuerdo a la *clase* de documento especificado por el *comando* `\documentclass{}`; por ejemplo, `\documentclass{book}` formatea el documento de tal manera que el producto sea un documento con formato de libro.

Modo “texto” y modo “matemático”. Hay comandos que funcionan en *modo texto* y hay comandos que solo funcionan en *modo matemático*. Todo lo que es lenguaje matemático se edita en modo matemático. Hay varios entornos para este modo, el más común es el entorno delimitado por dos signos de dólar (`$...$`). Un ejemplo de código **LaTeX** es el siguiente:

Ejemplo 1.1

```
\documentclass{article}
\usepackage{amsmath}
```

```

\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
$0^0$ es una expresi'on indefinida.
Si $a>0$ entonces $a^0=1$ pero $0^a=0$.
Sin embargo, convenir en que $0^0=1$ es adecuado para que
algunas f'ormulas se puedan expresar de manera sencilla,
sin recurrir a casos especiales, por ejemplo
$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}$$
$$(x+a)^n=\sum_{k=0}^n\binom{n}{k}x^ka^{n-k}$$
\end{document}

```

Este código se digita en un editor (en la figura de abajo se usó **Texmaker**) y luego se *compila*. La ventana a la derecha en la figura que sigue, muestra la salida:

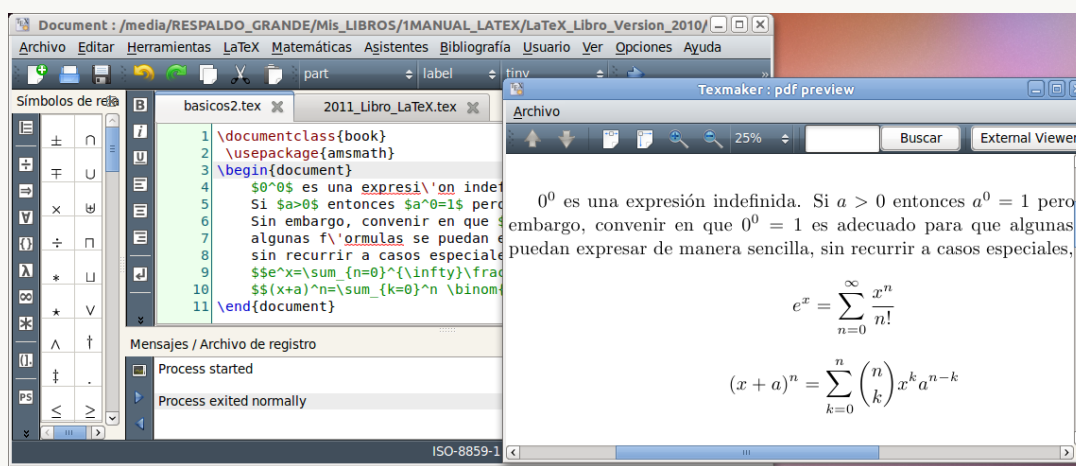


Figura 1.1. Sesión LaTeX en Texmaker

1.2 Comenzar: Distribuciones TeX y editores

Para comenzar necesitamos instalar una “distribución **TeX**” y it después instalar un editor. Una “distribución **TeX**” contiene binarios (los “motores”), paquetes y extensiones adicionales: Integra todo lo que hace falta para poner a funcionar todo.

Formalmente hay varios binarios (“motores”) con los que podemos trabajar: **TeX** y sus extensiones **pdfTeX**, **XeLaTeX**, **LuaTeX** (**TeX** con el lenguaje de programación **Lua**), etc. **LaTeX** y **PDFLaTeX** usan el “motor” **pdfTeX**, etc. La distribución completa viene con todas estos “motores”. Hay varias distribuciones **TeX**, por ejemplo: **TeXLive** (Windows, Linux, Mac), **MacTeX** (Mac OS X) y **MikTeX** (Windows).

En este manual vamos a usar los formatos (dialectos) **LaTeX** o **PDFLaTeX**. Una vez que instalamos una distribución **TeX**, es conveniente tener *un editor* no solo para editar de manera cómoda el texto, también para acceder de manera sencilla a las tareas usuales de una sesión **LaTeX**: editar, compilar y visualizar (en DVI o PDF).

Después de la instalación de la distribución TeX, instalamos un editor. Los editores en general ya vienen con un visor de documentos incorporado. Hay varios editores: **Texmaker**, **Texstudio**, **Winshell**, **Kile**, **Overleaf (ShareLaTeX + Overleaf)**, **LyX**, etc. Los editores buscan la instalación TeX de manera automática.

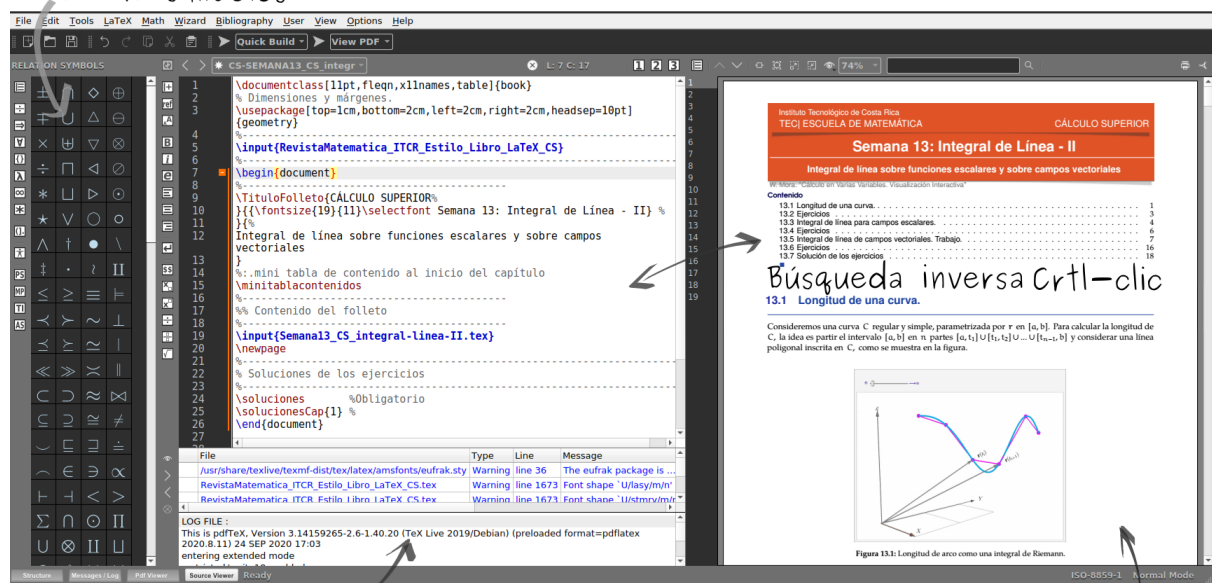
Luego se pueden configurar algunas cosas adicionales en el editor y también instalar software de apoyo para edición de figuras (“Recortes”, Inkscape, Shutter, etc.). ??.

Para empezar, podemos usar el siguiente ejemplo de instalación (en los distros Linux se puede usar la terminal para instalar, en Windows y macOS, se descargan las aplicaciones y se instalan)

Linux	Windows	macOS
- Distribución TeX TeXLive en Ubuntu <code>sudo apt-get install texlive-full</code> - Editor TexMaker <code>sudo apt-get install texmaker</code> - Un visor: Incorporado u Okular <code>sudo apt-get install okular</code>	- Distribución TeX MikTeX - Editor: TeXMaker - Visor: Incorporado	- Distribución TeX MacTeX - Editor: TeXMaker - Visor: Incorporado

Figura 1.2. Ejemplos de instalación de una distribución TeX y un editor

barra de símbolos



Resultado de la compilación Visor pdf incorporado

Figura 1.3. Editor TexMaker

Online

Se puede editar y compilar con un editor LaTeX “en línea”: Se podría usar por ejemplo **Overleaf**, **CoCalc**, etc. También se puede usar **VerTeX** en el celular (Android, iOS)

Aunque todavía es un poco laborioso (2020), se puede instalar y usar (offline) la distribución **TeXLive** en un dispositivo con Android, usando la aplicación **Termux**

Las pruebas de este libro se hicieron con **MiKTeX2.9** sobre Windows 10 y con **TeX Live 2019** sobre **Ubuntu**.

Editores de ecuaciones en Internet

También en Internet hay editores de ecuaciones “en línea”, por ejemplo **LaTeX4Thecnics**

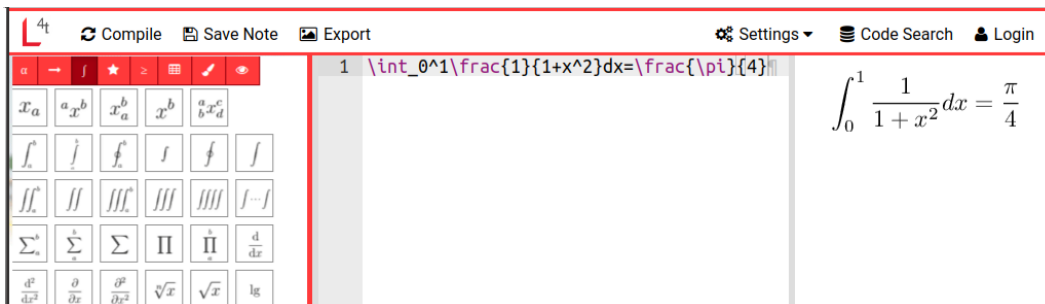


Figura 1.4. Editor de ecuaciones en línea LaTeX4Thecnics.

Por supuesto, en Internet se pueden obtener más opciones de editores de ecuaciones.

Símbolos

Hay una aplicación web llamada **Detexify** con la que podemos obtener el código necesario para un símbolo dibujándolo

Detexify² - LaTeX symbol classifier

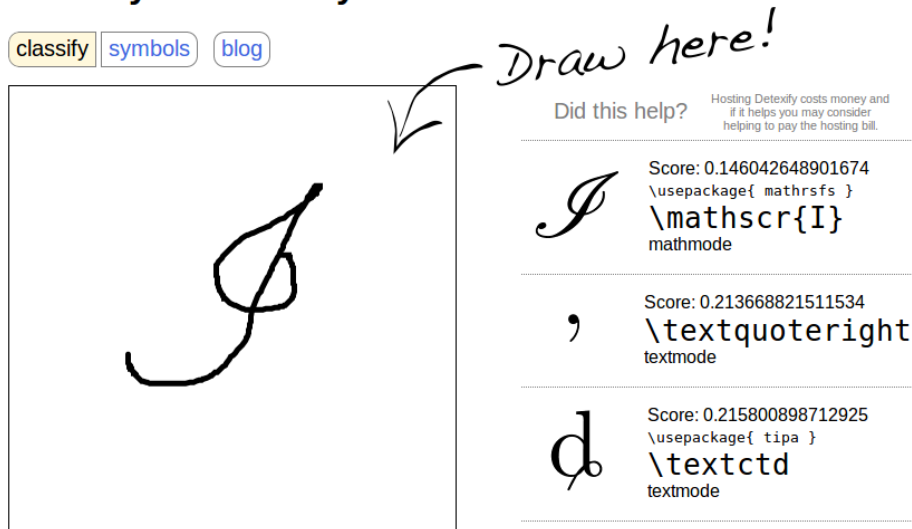


Figura 1.5. Detexify

También se puede usar el sitio “**Overleaf**” para editar un documento **LaTeX** de manera colaborativa.

Convertir documentos Word a LaTeX

Posiblemente haya interés en convertir documentos (con o sin ecuaciones) de Microsoft **Word** o de **Writer** (LibreOffice), a documentos LaTeX.

Una opción es **Writer2LateX2.1** que es una extensión de **LibreOffice**. La versión **Write2LaTeX2.1** tiene un buen desempeño. Para hacer la conversión se carga el documento con **Writer** de **LibreOffice** y se usa la opción **Archivo-Exportar-Tipo de Archivo-LaTeX 2e**.

Hay varios programas privativos que también hacen la conversión de manera aceptable (para documentos no muy complejos), por ejemplo **Word2TeX** y **Word-to-LaTeX**.

Convertir documentos PDF a LaTeX

Hay varias opciones para convertir documentos PDF a LaTeX. Una de ellas es el programa privativo **InftyReader**. Este programa permite hacer conversiones de cinco páginas diarias de manera gratuita. La conversión es aceptable para documentos no muy complejos. El software procede haciendo una imagen en formato .tif de cada página y luego aplicando OCR (Optical Character Recognition) para reconstruir el documento.

¿Preguntas LaTeX?

La documentación LaTeX es muy densa. Si tiene una duda, una de las mejores opciones es preguntar o buscar una respuesta en **TeX - LaTeX Stack Exchange**. Este es un sitio de preguntas y respuestas para los usuarios de TeX, LaTeX, ConTeXt y otros sistemas de composición.



En una sesión **L^AT_EX** realizamos varias acciones: Ponemos *un preámbulo* con la clase de documento, paquetes que se van a usar, cosas de maquetación, etc. y editamos el cuerpo del documento, luego compilamos (**L^AT_EX** o **PDFL^AT_EX**) y vemos el resultado en un visor (DVI o PDF).

2.1 Editar, compilar y ver el resultado.

- 1 Ponemos un **preámbulo**: La clase de documento, indicaciones sobre márgenes, largo y ancho de página, numeración, etc., y cargamos los paquetes adicionales (fuentes, símbolos, gráficos, etc.).



Descargar archivo

```
\documentclass{article} % art\ 'iculo
% Dimensiones 18cmx21cm. M\ 'argenes: Superior 2cm, izquierdo 2cm
\usepackage[total=18cm,21cm,top=2cm, left=2cm]{geometry}
  \parindent = 0mm % sangr\ '{\i}a = 0mm
% Paquetes con símbolos matemáticos
\usepackage{amsmath,amssymb,amsfonts,latexsym}
\usepackage[T1]{fontenc} % fuentes adecuadas para salida
\usepackage[utf8]{inputenc} % acentos desde el teclado
```

Preámbulo

```
\begin{document}
\section{Problema abierto}
La dificultad en la investigaci\ 'on de la convergencia de la
serie

$$\sum_{n=1}^{\infty} \frac{1}{n^3 \sin^2 n}$$

se debe a la falta de conocimiento sobre el comportamiento
de  $n \sin n$  conforme  $n \rightarrow \infty$ ,
y esto esta relacionado con las aproximaciones racionales
de  $\pi$ .
\end{document}
```

- 2 **Editamos**: Escribimos texto corriente y texto en *modo matemático* (posiblemente combinando ambos). Mucho del texto en modo matemático se edita en los entornos $\dots$ o $\dots$ (si va centrado). Esto le indica al programa que interprete el texto y lo convierta en símbolos matemáticos.
- 3 **Compilamos**: En el menú del editor está la opción **L^AT_EX** o la opción **PDFL^AT_EX** para compilar. Esto nos permite detectar, por ejemplo, errores en los comandos o en la sintaxis de

una fórmula.

4 Ver el archivo DVI o el PDF

- Una vez que hemos compilado con la opción **LaTeX**, usamos la opción **DVI** para ver el documento (esto hace que un visualizador ejecute el programa 'dvips' para ver el documento en pantalla). Si queremos una versión PDF, usamos la opción **DVI->PDF**.
- Una vez que hemos compilado con la opción **PDFLaTeX**, usamos la opción **Ver PDF** para ver el documento. Estas dos últimas opciones vienen usualmente juntas por defecto, haciendo clic en **QuickBuild**

2.2 Compilar

Después de compilar se producen varios archivos: *.tex, *.dvi (o .pdf), *.aux, *.log, *.toc. El archivo de edición tiene extensión *.tex mientras que el archivo .log contiene un informe del proceso de compilación. Para imprimir un documento **LaTeX** (generado con la opción **LaTeX**) solo necesitamos el archivo *.dvi y los archivos de los gráficos incluidos en el documento (si hubiera).

¿Compilar con PDFLaTeX o LaTeX?

Cuando compilamos con **LaTeX** obtenemos un archivo DVI y podemos usar algunos paquetes que nos permiten, por ejemplo, manipular gráficas **.eps**. Este formato a veces se solicita de manera obligatoria en cierto tipo de publicaciones.

PDFLaTeX es una extensión de **TeX** que puede crear archivos PDF directamente desde un archivo de origen .tex. Cuando compilamos con **PDFLaTeX**, generamos un PDF de igual apariencia que el DVI.

El Formato PDF se ha convertido en uno de los formatos de documentos electrónicos más utilizados para la publicación de documentos. Hay muchas ventajas que lo hicieron muy popular: Es adecuado para la visualización y para imprimir, permiten búsquedas, etc. Pero también, compilando con **PDFLaTeX**, podemos usar comandos adicionales en el archivo .tex para agregar propiedades en el archivo PDF: Agregar ligas, personalizar el menú de navegación, agregar video, etc., es decir, cosas que son importantes para leer documentos en pantalla y para la distribución en Internet.

Este libro fue compilado con **PDFLaTeX** porque usa ligas a otros documentos y tiene muchas figuras en distintos formatos.

Compilar desde un editor

Cada editor tiene una manera para compilar, por ejemplo con **TeXmaker** la compilación se hace desde la pestaña **Herramientas-Tools**. En **TeXMaKer** hay *teclas rápidas* para compilar y otras para otras tareas. Por ejemplo, el botón **Quick Build** viene configurado para compilar con **PDFLaTeX** e inmediatamente ver el pdf en el visor.

Compilación rápida: PDFLaTeX + Ver Pdf



10

Resultado de la compilación

Si no hay errores, el mensaje en la ventana inferior sería,

Process exited normally

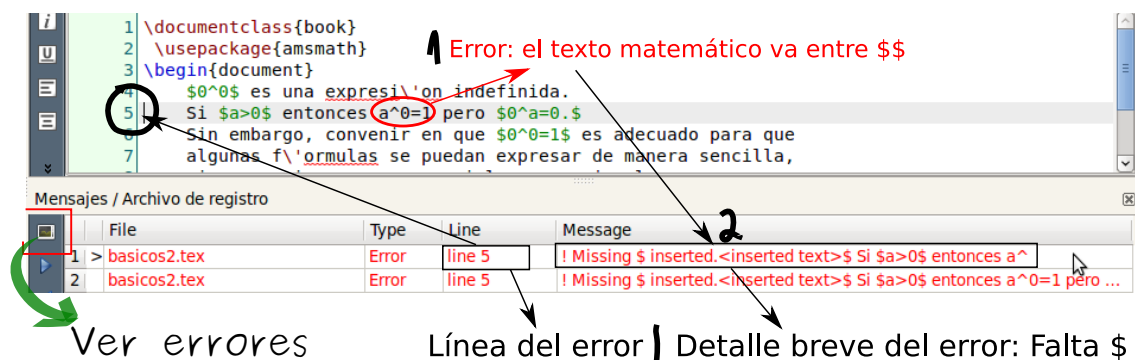
Si hay errores, el mensaje en la ventana inferior sería,

Process exited with error(s)

En este caso, la compilación genera una información de salida en la “ventana de mensajes” (se habilita o deshabilita en el menú **Ver**). Si la compilación encuentra algún error se indicará con un mensaje corto (en rojo), por ejemplo

Error line 323 ! Missing \\$ <inserted text>...

En este ejemplo, esto nos indica que falta '\$' en la línea 323. Como se ve, $\frac{x}{x+1}$ es texto matemático (inicia con un comando de fracción) y por tanto no se puede interpretar como texto corriente, necesita estar entre \$ \$. Al hacer clic en 'line 323' nos llevará a la línea del error.



2.3 Usar paquetes adicionales

Un paquete **LaTeX** es un archivo **.sty** (o varios archivos) con comandos y código de programación **TeX** que tiene como propósito agregar nuevas facetas (o modificar otras facetas) al documento. Todos los paquetes vienen con su propia documentación.

Los paquetes se invocan con el comando `\usepackage{nombre-del-paquete}`. **LaTeX** viene con una cantidad importante de comandos que se pueden usar de forma inmediata sin invocar ningún paquete adicional. Aunque siempre existe la posibilidad de instalar la *versión completa* de la distribución con todos los paquetes que hay disponibles en el momento (**TeXLive-full**,

con **MiKTeX** podría seleccionar todos los paquetes con la consola de MiKTeX), la mayoría de paquetes *deben ser invocados en el preámbulo* del documento para poder usarlos.

Si un paquete no está disponible en su instalación, se produce un error de compilación. En la distribución **MiKTeX** se puede habilitar la búsqueda e instalación del paquete “en el momento de la compilación” (“on the fly”). Esto lo puede habilitar al inicio de la instalación o usando después la consola de MiKTeX (“MiKTeX Console”)

La instalación de un nuevo paquete puede ser tan sencilla como agregar un solo archivo **.sty** en nuestra carpeta de trabajo o puede involucrar varios pasos.

Lo mejor es tener la distribución **TeX** siempre actualizada, porque siempre hay nuevos paquetes y también porque los paquetes viejos son modificados y a veces son incompatibles con la versión de nuestro sistema.

Para actualizar la distribución **TeX** y agregar nuevos paquetes puede ver el apéndice ?? .

2.4 Ajustes con el paquete Geometry

El diseño de documento se puede simplificar con el paquete **geometry**. Por ejemplo, si queremos un documento con región impresa con dimensiones 18cm×21cm y con margen superior de 2cm y margen izquierdo de 2cm, cambiamos nuestras instrucciones agregando en el *preámbulo*

```
\usepackage[total={18cm,21cm},top=2cm, left=2cm]{geometry}
```

   [Descargar archivo](#)

```
\documentclass{article}
% Dimensiones 18cmx21cm. M\argenes: Superior 2cm, izquierdo 2cm
\usepackage[total={18cm,21cm},top=2cm, left=2cm]{geometry}

\begin{document}
    Dise~no con el paquete "geometry"
\end{document}
```

Otras opciones con el paquete “geometry”

- 1 Si quisieramos un texto con dimensiones 18cm×21cm y centrado, usaríamos

```
\usepackage[total={18cm,21cm},centering]{geometry}
```

- 2 Si quisieramos un texto con dimensiones es 6.5 pulgadas de ancho por 8.75 pulgadas de alto y el margen superior en cada página a 1.2 pulgadas del borde superior de la página y el margen izquierdo a 0.9 pulgadas desde el borde izquierdo y además el pie de página con el número de página en la parte inferior del área de texto, entonces usaríamos

```
\usepackage[a5paper,total={6.5in,8.75in},top=1.2in,left=0.9in,
includefoot]{geometry}
```

- 3 Este libro usa la instrucción

```
\usepackage[text=15cm,25cm,centering,headsep=20pt,top=0.8in,bottom =
0.8in,letterpaper,showframe=false]{geometry}
```

- 4 Otras opciones del paquete son `landscape`, `twocolumn`, `twoside`, `foot=xcm`, `bottom=xcm`, etc.

Un manual (actualizado) de referencia para este paquete se puede encontrar en <https://ctan.org/pkg/geometry>.

2.5 Idioma

El idioma por defecto que utiliza **LaTeX** es el *inglés*, sin embargo, utilizando algunas instrucciones se puede lograr que soporte otros idiomas, en particular, veremos cómo hacer para que soporte el español.

Acentos y otros caracteres

LaTeX normalmente no acepta tildes, ni la “ñ”, tampoco el signo de pregunta ‘¿’, ni la apertura o el cierre de comillas. Para que acepte estos caracteres se deben utilizar las instrucciones que aparecen en la tabla 2.1 o usar un paquete que nos habilite para usar los acentos y otros símbolos desde el teclado (ver más abajo).

Comando	Símbolo	Comando	Símbolo
<code>\'a</code>	á	<code>?`</code>	¿
<code>\'e</code>	é	<code>!`</code>	¡
<code>\'i</code>	í	<code>\"</code>	“ ”
<code>\'o</code>	ó	<code>\'o</code>	º
<code>\'u</code>	ú	<code>\~n</code>	ñ

Tabla 2.1. Acentos en modo texto y otros símbolos

Para que **LaTeX** reconozca los acentos que usamos en español directamente del teclado (como `ú` en vez de `\'u`) y para que genere una salida adecuada para un PDF, colocamos en el *preámbulo*

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

Con estos paquetes se tendrá un soporte para los acentos en español. La codificación **utf-8** es la codificación estándar actual.

Actualmente, los editores están configurados para usar las comillas directamente del teclado. Las versiones actuales del editor **Kile** ya vienen configurados para hacer esta sustitución y en **TeXStudio** se puede habilitar esta faceta en su configuración. Para este libro usamos la versión más reciente del editor **Kile** así que podemos escribir “Ho la” para obtener “Hola”.

Cambiar las plantillas a español

Otro problema que tiene **LaTeX** con el idioma es que los textos para “capítulo”, “sección”, etc., están en inglés, por lo tanto, en un libro no saldría **Capítulo 1** sino **Chapter 1**.

Existen dos formas sencillas de solucionar este problema. La forma más simple y recomendada es usar el paquete **babel**, ponemos en el *preámbulo* del documento la instrucción

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

que carga la opción en español de la librería **babel**. Además quedará habilitada la división correcta de las palabras. A veces este paquete tiene conflictos con otros paquetes que podríamos querer usar.

División de palabras: Paquete **babel** y otras opciones.

Agregando ‘`\usepackage[spanish]{babel}`’ en el preámbulo (junto con `inputenc` y `fontenc`), se logra que **LaTeX** divida correctamente la mayoría de palabras en español, sin embargo hay algunos casos en los que no será así; si al componer el texto observamos que hay una palabra que se ha dividido mal, vamos a esa palabra en el archivo *.tex, y le indicamos exactamente donde la puede dividir, usando `\-`. Por ejemplo, `e\ -xa-men`, `ac\ -ci\ 'on`, `am\ -nis\ -t\ '\{i}\ -a`

Este sistema tiene el inconveniente de que **LaTeX** sólo divide bien la palabra en ese punto del documento y si dicha palabra aparece otra vez habrá que volver a decirle cómo se divide, y tiene la ventaja de que funciona con palabras que tienen acento.

También se puede usar `\hyphenation{e-xa-men, ...otras divisiones...}` (no se permiten palabras con tildes) al inicio del documento para que **LaTeX** divida las palabras *tal y como se especifica* en la lista.

2.5.1 Si hay conflictos con el paquete **babel**...

A veces el paquete **babel** tiene conflictos con algún otro paquete que queremos usar. Por eso tenemos que tener en cuenta un par de opciones más.

Si el documento es de tipo **article**, podemos poner en el *preámbulo*

```
\renewcommand{\contentsname}{Contenido}
\renewcommand{\partname}{Parte}
\renewcommand{\appendixname}{Ap\ 'endice}
\renewcommand{\figurename}{Figura}
\renewcommand{\tablename}{Tabla}
\AtBeginDocument{\renewcommand{\tablename}{Tabla}}
\renewcommand{\abstractname}{Resumen}
\renewcommand{\refname}{Bibliograf\ '\{i}a}
```

Si el documento es **book** se puede agregar en el *preámbulo*

```
\renewcommand{\contentsname}{Contenido}
\renewcommand{\partname}{Parte}
\renewcommand{\appendixname}{Ap\ 'endice}
```

```
\renewcommand{\figurename}{Figura}
\renewcommand{\tablename}{Tabla}
\AtBeginDocument{\renewcommand\tablename{Tabla}}
\renewcommand{\chaptername}{Cap\'\{\i\}tulo} % para 'book'
\renewcommand{\bibname}{Bibliograf\'\{\i\}a} % para 'book'
```

Una tercera opción

Una tercera opción es hacer este cambio permanente: Se puede editar los archivos `article.cls`, `report.cls` y/o `book.cls`. En **Ubuntu** estos archivos están en `/usr/share/.../tex/latex/base` (debe tener permisos de escritura para modificarlos). En la distribución **MiKTeX**, están en `C:/Archivosdeprograma/MiKTeX2.9/tex/latex/base`. En ambos casos, se busca y se abre el archivo de texto `article.cls` (o `report.cls` o `book.cls` y se buscan las líneas

```
\newcommand\contentsname{Contents}
\newcommand\listfigurename{List of Figures}
...
```

y se cambian por

```
\newcommand\contentsname{Contenido}
\newcommand\listfigurename{Lista de Figuras}
...
```

Luego, simplemente se guarda el archivo.

2.5.2 Plantilla para este capítulo.

A continuación se muestra una plantilla general para este capítulo,

   [Descargar archivo](#)

```
\documentclass{article}
\usepackage[total={18cm,21cm},top=2cm, left=2cm]{geometry}
\parindent = 0mm % Sin sangr\'\{\i\}a
\usepackage{latexsym,amsmath,amssymb,amsfonts}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[spanish]{babel} % Idioma espa\~nol
\renewcommand{\baselinestretch}{1.1} % espaciado 1.1
\usepackage{graphicx,xcolor}
\pagestyle{myheadings}
\markright{..... texto .....} % Encabezados simples
% -----
\begin{document}
\section{Problema abierto}
La dificultad en la investigaci\~on de la convergencia
de la serie  $\sum_{n=1}^{\infty} \frac{1}{n^3 \sin^2 n}$ 
se debe a la falta de conocimiento sobre el comportamiento de
 $\sin n$  conforme  $n \rightarrow \infty$ , y esto esta
relacionado con la 'medida de irracionalidad' de  $\pi$ 
\end{document}
```

2.6 Otros ajustes manuales del documento

Un documento básico en **LaTeX** se compone de dos partes: el *preámbulo* del documento y el *cuerpo*. Al inicio del documento se debe especificar la clase de documento y lo relativo al ajuste de las páginas, nada de lo que pongamos en el preámbulo aparecerá en el documento que se imprime al final. En el cuerpo se escribe el texto (normal y matemático). Es la parte que aparecerá impresa como producto final.

Preámbulo	<pre> \documentclass{article} % M'argenes \textheight = 21cm \textwidth = 18cm \topmargin = -2cm \oddsidemargin= -2cm % Paquetes \usepackage{amsmath,amssymb,amsfonts,latexsym} \usepackage[utf8]{inputenc} \usepackage{graphicx} </pre>
Cuerpo del documento	<pre> \begin{document} Sea $a>0$, definimos a^x por la f'ormula... \end{document} </pre>

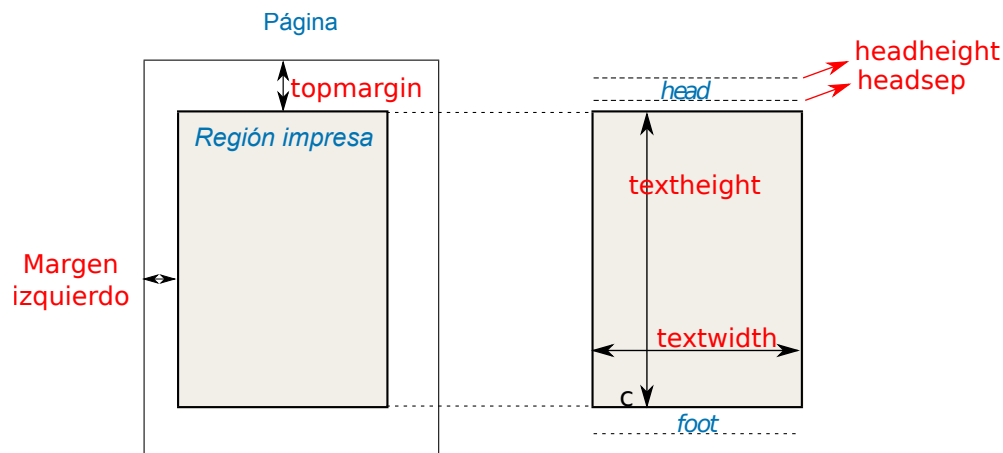


Figura 2.1. Diseño del documento.

- `\documentclass{article}`: Es la clase de documento; `article` se utiliza para editar documentos con formato de artículo. `article` se refiere al archivo `article.cls`. Estos archivos `.cls` implementan la estructura específica de un documento. También se usa “report” o “book” para un reporte o un libro.
- `\textheight=21cm`: Establece el largo del texto en cada página (en este caso, de 21 cm). El default es 19 cm.
- `\textwidth=18cm`: Establece el ancho del texto en cada página (en este caso, de 18 cm). El default es 14 cm.

- `\topmargin=-2cm`: Establece el margen superior. El default es de 3 cm, en este caso la instrucción `-2cm` sube el margen 2 cm hacia arriba.
- `\oddsidemargin=-2cm`: Establece el margen izquierdo de las páginas impares. El default es de 4.5 cm; sin embargo, con sólo poner esta instrucción el margen queda en 2.5 cm. Si el parámetro es positivo se aumenta este margen y si es negativo disminuye. Note que esto combinado con el ancho del texto, determina el ancho del otro margen!
- `\usepackage{amsmath,amssymb,amsfonts,latexsym}`: Esta instrucción indica que en este documento se usarán paquetes de símbolos adicionales (símbolos de la AMS).
- `\usepackage[utf8]{inputenc}`: Esta instrucción se usa para incluir un paquete que nos permite usar los acentos y otros símbolos, directamente del teclado. La codificación que usamos es el estándar utf-8.
- `\usepackage{graphicx}`: Esta instrucción se usa para incluir un paquete para el manejo de gráficos y figuras en el documento.

Otros comandos para el preámbulo

- `\renewcommand{\baselinestretch}{1.5}`: Genera un texto a espacio y medio. Si se pone 2 en vez de 1.5, lo hace a doble espacio.
- `\pagestyle{empty}`: Elimina la numeración de las páginas.
- `\parskip=Xmm`: Genera un espacio de X mm entre los párrafos.
- `\headheight`: Altura de la cabecera (page header) de la página.
- `\headsep`: Distancia desde la parte inferior de la cabecera al cuerpo de texto en una página.
- `\parindent=0mm`: Elimina la sangría.
- `\pagestyle{myheadings}`: Coloca la numeración de página en la parte superior.
- `\markright{'texto'}`: Coloca 'texto' en la parte superior de la página. Se pueden poner varios `\markright` en el texto (en cada sección, por ejemplo).

Ejemplo: `\markright{\LaTeX \hrulefill W . Mora, A. Borbón \; \;}`

- `\newpage`: Le indica a **L^AT_EX** que siga imprimiendo en la página siguiente.



En un archivo de **LaTeX** se combina el texto normal con el texto matemático, en este capítulo se iniciará trabajando con el texto normal (tipos de letra, párrafos, enumeración, secciones, etc.) y en el siguiente capítulo se mostrará el texto matemático.

3.1 Tipos y tamaños de fuentes.

3.1.1 Caracteres especiales.

Algunos caracteres están **reservados** para que cumplan alguna función, por eso no se pueden obtener digitándolos (tecleándolos) directamente como cualquier letra. El hacerlo puede producir algún error de compilación, o puede pasar que el carácter sea ignorado. En las siguientes dos tablas se especifica el uso de algunos caracteres y el comando que se debe digitar (teclear) para imprimirlos.

Caracter	Reservado para:
\	carácter inicial de comando
{ }	abre y cierra bloque de código
\$	abre y cierra el modo matemático
&	tabulador (en tablas y matrices)
#	señala parámetro en las macros
_, ^	para subíndices y exponentes
~	para evitar cortes de renglón
%	para comentarios

Caracter	Se imprime con:
\	\tt\char`\
{, }	\{, \}
\$	\\$
&	\&
_, ^	_ , \^{} }
#	\#
~	\~{ }
%	\%

Por ejemplo, para obtener llaves debemos usar los comandos \{ y \}. Para conjuntos escribimos $A = \{a, b, c\}$ y obtenemos: $A = \{a, b, c\}$. Podría ser mejor $A = \big\{a, b, c\big\}$ que produce: $A = \{a, b, c\}$.

El comando verb

El comando `\verb` permite imprimir los caracteres tal y como aparecen en pantalla. Por ejemplo, si se digita `\verb@{\$x^2+1\$}@` se imprimirá $\{ \$ x^2+1 \$ \}$. El símbolo '@' se usa como delimitador de lo que se quiere imprimir. Se pueden usar otros delimitadores no presentes en el texto a imprimir.

3.1.2 Algunos tipos de fuentes (fonts).

Para cambiar el tipo de letra se pone `\{tipo texto\}`. Por ejemplo, para escribir en negrita se pone `\{bf texto\}` o mejor `\textbf{texto}`

Comando	produce	Comando	produce
<code>{\rm Roman }</code>	Roman	<code>{\sl Slanted }</code>	<i>Slanted</i>
<code>{\em Enfático}</code>	<i>Enfático</i>	<code>{\sf Sans Serif }</code>	Sans Serif
<code>{\bf Negrita }</code>	Negrita	<code>{\sc Small Caps }</code>	SMALL CAPS
<code>{\it Itálica }</code>	<i>Itálica</i>	<code>{\tt Typewriter }</code>	Typewriter
		<code>\underline{Subrayado}</code>	<u>Subrayado</u>

Observe el uso de las llaves para delimitar el alcance del comando, es decir, el comando no tiene efecto más allá de lo que está entre llaves.

Combinaciones

Los comandos de la tabla anterior en realidad quedaron en desuso con la nueva versión de **LaTeX** (**LaTeX2e**, 1994), pero estaban tan arraigados que se decidió mantenerlos aún con sus limitaciones. Por ejemplo, ‘`{\it\bf Negrita en itálica}`’ no produce negrita en itálica, **produce**: ‘Negrita en itálica’.

En el nuevo **LaTeX2e** se usa `\textit{...}`, `\textbf{...}`, `\texttt{...}` etc. Ahora podemos escribir “`\textit{\textbf{Negrita en itálica}}`” que produce *Negrita en itálica*.

3.1.3 Tamaños de letras.

Una manera de especificar el tamaño de la fuente es la siguiente,

Comando	produce
<code>{\tiny Tiny}</code>	Tiny
<code>{\scriptsize Script}</code>	Script
<code>{\footnotesize Foot}</code>	Foot
<code>{\small Small}</code>	Small
<code>{\normalsize Normal}</code>	Normal
<code>{\large large}</code>	large
<code>{\Large Large}</code>	Large
<code>{\huge huge}</code>	huge
<code>{\Huge Huge}</code>	Huge

Se pueden hacer combinaciones de tipos de letras con diferentes tamaños.

Ejemplo 3.1

- `{\large \bf Nota:}` **produce**: Nota:
- `{\large \bf \tt Nota:}` **produce**: Nota: (no hay efecto!)
- `{\Large\textit{\textbf{Nota}}}` **produce**: *Nota*

El paquete “anyfontsize”

Los comandos anteriores a menudo son suficientes para el uso cotidiano, pero es posible que de vez en cuando sea adecuado tener control sobre el tamaño de la fuente. Esto se puede hacer

con el paquete “anyfontsize”, la manera de usarlo es como

```
{\fontsize{<size>}{<line space>}\selectfont ... texto...}
```

line space es el espacio entre renglones adyacentes. Si no tiene idea de qué espacio dejar entre líneas adyacentes, en general puede usar “line space=0.2*size”.

Por ejemplo, el **El código**:

```
\documentclass{article}
\usepackage{anyfontsize}
\begin{document}
Ejemplo de cómo usar el paquete \verb+anyfontsize+\par
{\fontsize{0.1cm}{0.1cm}\selectfont Variar el tama~no de la fuente\par}

{\fontsize{0.3cm}{0.06cm}\selectfont Variar el tama~no de la fuente\par}

{\fontsize{0.6cm}{0.12cm}\selectfont Variar el tama~no de la fuente\par}

\bigskip
Ejemplo de cómo usar el paquete \verb+anyfontsize+\par
{\fontsize{0.3cm}{0.5cm}\selectfont Variar el tama~no de la fuente\par}

{\fontsize{0.3cm}{0.5cm}\selectfont Variar el tama~no de la fuente\par}

{\fontsize{0.6cm}{0.5cm}\selectfont Variar el tama~no de la fuente\par}
\end{document}
```

produce:

Ejemplo de cómo usar el paquete anyfontsize

Varia el tamaño de la fuente

Varia el tamaño de la fuente

Ejemplo de cómo usar el paquete anyfontsize

Varia el tamaño de la fuente

Varia el tamaño de la fuente

Varia el tamaño de la fuente

3.2 Párrafos y efectos especiales.

En **LaTeX** se puede escribir de manera ordenada o desordenada, el programa acomoda el texto e interpreta los comandos que se digitaron. Pero, por tratarse de un código, mejor es indentar correctamente el texto. Para indicarle a **LaTeX** que un párrafo ha terminado hay que *dejar un renglón en blanco*.

Si entre dos palabras se deja más de dos espacios en blanco solo se imprimirá uno. También se tiene que dejar doble paso de línea (doble ‘enter’) para separar párrafos o usar “\\” para cambiar de renglón. Si usamos “\\” avanzamos dos renglones. En general se puede usar \\, \par o \newline para cambios de línea. Estos comandos pueden tener un comportamiento algo diferente en ciertos entornos (tabular, texto ordinario, texto centrado, etc.) y se deberá usar uno en vez de otro.

Ejemplo 3.2

Cambio de renglón:

El código:

```
{\bf introducción.} \\
Se parte de un conjunto  $\Omega$ 
de  $n$  patrones, objetos
o 'individuos',
descritos por un vector de  $p$ 
atributos.\\ \\ % doble renglón

{\bf Nota:}....
```

produce:**Introducción.**

Se parte de un conjunto Ω de n patrones, objetos o 'individuos', descritos por un vector de p atributos.

Nota:....**3.2.1 Centrar**

Para centrar un texto se pone éste en el entorno `\begin{center} ... \end{center}`

Ejemplo 3.3 (Centrado).**El código:**

```
\begin{center}
Manual de\\
LaTeX
\end{center}
```

produce:

Manual de
LaTeX

3.2.2 Espacio horizontal y vertical

Para dejar espacio horizontal se usa el comando `\hspace{Xcm}`. El efecto es abrir espacio o correr horizontalmente texto, tablas o gráficos. X cm es el corrimiento a la derecha o a la izquierda en centímetros, según sea X positivo o negativo. También se puede usar, por supuesto, X mm, X in, etc.

Si usamos `\hspace`, **LaTeX** quita el espacio horizontal (en blanco) que hay al final de una línea. Si queremos que **LaTeX** no elimine este espacio, se incluye el argumento opcional `*`, es decir, escribimos `\hspace*`

Ejemplo 3.4

● `$\subset$  \hspace{-3mm} / 30` **produce:** $\not\subset 30$

A la Izquierda!

● `\hspace{-6.5cm}A la Izquierda!` **produce:** (← ver en el margen!)

A la Izquierda!

● `\hspace*{-6.5cm}A la Izquierda!` **produce:** (← ver en el margen!)

- Podemos distribuir uniformemente piezas de texto, en una línea, a través del ancho de la página con `\hspace*{\fill}`. El comando `\fill` es lo mismo que `\stretch{1}`. Se usa para “estirar” el espacio en blanco hasta que se llena todo el espacio restante de una línea (que usualmente contiene texto).

```
\hspace*{\fill}{Texto A} \hfill {Otro texto B}\hspace*{\fill}
```

produce:

Texto A

Otro texto B

Para abrir espacio verticalmente se usa el comando `\vspace{Xcm}` que funciona de manera análoga a `\hspace`. En el ejemplo que sigue se debe dejar un renglón en blanco para lograr el efecto.

Ejemplo 3.5

Vamos a subir la palabra ‘VERDE’ 1.2 cm y a desplazarla 2.4 cm a la derecha

El código:

```
Línea superior - - - - - \\
\vspace{-1.2cm}
\hspace{2.4cm} VERDE
```

produce:

Línea superior VERDE

3.2.3 Cajas

Para encerrar palabras o texto en cajas se usan los comandos `\fbox{ texto }` o `\framebox{...}`.

Ejemplo 3.6 (Cajas)

El código:

```
\begin{center}
Manual de\\
\LaTeX }
\end{center}
```

produce:

Manual de
L^AT_EX

3.2.4 Texto en columnas: multicol, minipage y parbox

En **LaTeX** existen varias maneras de escribir doble columna, ahora veremos algunas de ellas.

El comando multicol.

La forma más sencilla es utilizar el paquete `multicol`, para ello se pone en el preámbulo,
`\usepackage{multicol}`

Antes del texto que se quiere escribir a doble columna se escribe `\begin{multicols}{#}` en donde # representa el número de columnas que se quieren utilizar, cuando se finaliza se escribe `\end{multicols}`.

Ejemplo 3.7 (Dos columnas)

El código:

```
\begin{multicols}{2}
Hace que el texto que está entre los delimitadores salga a
doble columna, LaTeX se encarga de manera automática de
distribuir el texto entre las columnas de la mejor manera,...
\end{multicols}
```

produce:

Hace que el texto que está entre los delimitadores salga a doble columna, LaTeX se encarga de manera automática de distribuir el texto entre las columnas de la mejor manera, además de acomodar el texto cuando hay un cambio de línea.	El campo de separación que se deja entre las columnas se define en el <i>preámbulo</i> con la instrucción <code>\setlength{\columnsep}{7mm}</code> la cual dejaría un espacio de 7 milímetros entre ellas.
--	---

El ambiente minipage

Otra manera de insertar texto a doble columna (o más columnas), es utilizar el entorno `minipage` (también se puede utilizar `\twocolumn`), la sintaxis es como sigue,

```
\begin{minipage}[b]{Xcm}
% primera columna
\end{minipage} \hfill \begin{minipage}[b]{Ycm}
% segunda columna
\end{minipage}
```

Donde 'Xcm' y 'Ycm' especifica el ancho de cada columna. En vez de 'Xcm' y 'Ycm' se puede usar un porcentaje del ancho de página: `\textwidth`. Por ejemplo, para usar la mitad del ancho de página ponemos:

```
\begin{minipage}[b]{0.5\textwidth}
```

El entorno **minipage** se usa frecuentemente para acomodar el contenido (texto, texto y figuras, texto y tablas, etc.) en dos columnas. **minipage** usa un punto de referencia para alinear el contenido lo que puede ser confuso. En general, la opción 'b' (bottom) se usa para alinear las dos columnas en el 'fondo' del ambiente 'minipage'. También se puede usar la opción 't' (top) o 'c' (center). La elección depende del contenido de cada ambiente 'minipage'. Como a veces

es un poco complicado alinear el contenido, se puede usar el comando `\raisebox` o también el paquete `adjustbox`. En el ejemplo 3.8 se incluye texto y una figura.

Ejemplo 3.8 (Figuras en un ambiente `minipage`)

El código:

```
\begin{figure}[!h]
% 30% de la página
\begin{minipage}[b]{0.3\textwidth}
La imagen de la derecha muestra un icosaedro junto con un
dodecaedro (figura central), los satélites son un icosaedro,
un dodecaedro y un tetraedro. Las figuras fueron generadas con
{\sc Mathematica} y maquilladas con {\it Inkscape}.
% 60% de la pág
\end{minipage} \hfill \begin{minipage}[b]{0.6\textwidth}
\begin{center}% Figuras: ver capítulo 5
\includegraphics{images/ML_fig3.pdf}
\caption{ Poliedros }
\end{center}
\end{minipage}
\end{figure}
```

produce:

La imagen de la derecha muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

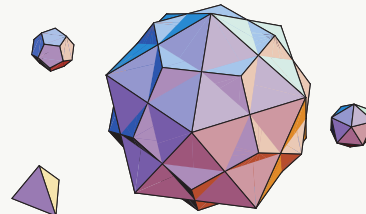


Figura 3.1. Poliedros

A veces, para acomodar las cosas (y no solo en el entorno “`minipage`”), debemos usar el comando `\raisebox{distancia}{text}`. La “distancia” del primer argumento especifica la distancia que elevará el texto respecto del segundo argumento. Si es negativa, entonces baja el texto. El texto no puede contener saltos de línea. Podemos usar figuras en vez de texto.

Ejemplo 3.9 (Usando el comando `\raisebox`)

En este ejemplo alineamos el gráfico con el texto, bajando un poco el gráfico. Por simplicidad se usa el paquete “`caption`”.

El código:

```
\documentclass{article}
\usepackage[margin=1cm, font=small]{caption}
```

```

\usepackage{graphicx}
%
\begin{document}
\begin{minipage}[t]{0.5\textwidth}
  \begin{center}
    \raisebox{-2.5cm}{\includegraphics[scale=0.8]{images/ML_fig3.pdf}}
    \captionof{figure}{Poliedros}
  \end{center}
\end{minipage}
\hfill \begin{minipage}[t]{0.5\textwidth}
  La imagen de la izquierda muestra un icosaedro junto con un
  dodecaedro (figura central), los satélites son un icosaedro,
  un dodecaedro y un tetraedro. Las figuras fueron generadas con
  {\sc Mathematica} y maquilladas con {\it Inkscape}.
\end{minipage}
%
\end{document}

```

Produce:

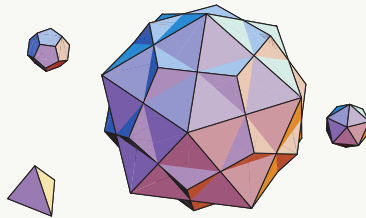


Figura 3.2. Poliedros

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

Ejemplo 3.10 (Usando el paquete adjustbox)

En este ejemplo alineamos el gráfico con el texto con el paquete **adjustbox**.

Podemos usar las opciones (entre otras) `valign=t` o `valign=b`. En este ejemplo usamos

```
\adjustbox{valign=t}{\begin{minipage}...\end{minipage}}
```

Por simplicidad se usa el paquete “caption”.

El código:

```

\documentclass{article}
\usepackage[margin=1cm, font=small]{caption}
\usepackage{graphicx}
%
\usepackage{adjustbox}
\begin{document}
\adjustbox{valign=t}{%
  \begin{minipage}{0.5\textwidth}
    \begin{center}
      \includegraphics[scale=0.8]{images/ML_fig3.pdf}
      \captionof{figure}{Poliedros}
    \end{center}
  \end{minipage}
}

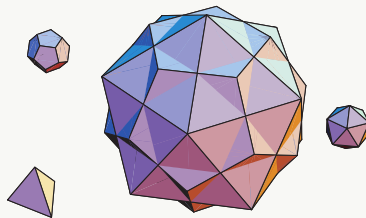
```



```

\end{center}
\end{minipage}
}
\hfill \adjustbox{valign=t}{%
\begin{minipage}{0.5\textwidth}
La imagen de la izquierda muestra un icosaedro junto con un
dodecaedro (figura central), los satélites son un icosaedro,
un dodecaedro y un tetraedro. Las figuras fueron generadas con
{\sc Mathematica} y maquilladas con {\it Inkscape}.
\end{minipage}
}
%
\end{document}

```

Produce:**Figura 3.3.** Poliedros

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

Hay una opción que no usa los paquetes anteriores. Si queremos alinear figuras en la parte superior de un ambiente minipage, podemos usar el comando `\vspace{0pt}` para cambiar el punto de referencia de la figura y así minipage alinea en la parte superior

Ejemplo 3.11

El código:

```

\begin{minipage}{4cm}
\vspace{0pt}
\includegraphics[scale=0.5]{images/ML_fig3.pdf}
\captionof{figure}{Poliedros}
\end{minipage}
\hfill \begin{minipage}{8cm}
\vspace{0pt}
La imagen de la izquierda muestra un icosaedro junto con un
dodecaedro (figura central), los satélites son un icosaedro,
un dodecaedro y un tetraedro. Las figuras fueron generadas con
{\sc Mathematica} y maquilladas con {\it Inkscape}.
\end{minipage}

```

produce:

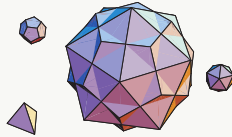


Figura 3.4. Poliedros

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

El comando `parbox`

Un comando similar a `minipage` es `parbox`. Se usa así:

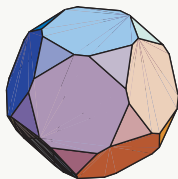
```
\parbox{xcmm}{texto}\parbox{ycmm}{texto}
```

Ejemplo 3.12 (Incluir una figura usando `parbox`).

El código:

```
\parbox{3cm}{\includegraphics{fig4.pdf}} \parbox{11.5cm}{En
{\sc Mathematica}, podemos eliminar una o varias caras de un dodecaedro,
seleccionar el color y el grosor de las aristas y poner color a las caras.
Para esto debemos utilizar los comandos ... } % Sale del 2do parbox!
```

produce:

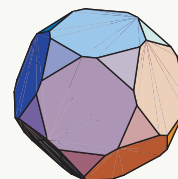


En MATHEMATICA, podemos eliminar una o varias caras de un dodecahedro, seleccionar el color y el grosor de las aristas y poner color a las caras. Para esto debemos utilizar los comandos ...

También puede usar el comando `\raisebox` o el paquete `adjustbox`, para alinear la figura con el texto. Por ejemplo, si cambiamos por

```
\parbox{2.5cm}{\raisebox{-3cm}{\includegraphics{images/ML_fig4}}}\dots
```

nos produce



En MATHEMATICA, podemos eliminar una o varias caras de un dodecahedro, seleccionar el color y el grosor de las aristas y poner color a las caras. Para esto debemos utilizar los comandos ...

`\parbox` también se puede usar en combinación con `\framebox` para hacer un recuadro:

```
\framebox{\parbox{xcmm}{texto}}.
```

Este comando no esta pensado para grandes bloques de texto.

- **N** Cuando se pone un gráfico en una columna debe tenerse el cuidado de ajustar el ancho del gráfico al ancho de la columna.

Cuando se utilice `\parbox` se debe tener el cuidado de terminar con `\\` para cambiar de renglón.

3.2.5 Líneas y otros efectos de texto

- `\hfill foo` : Se usa para alinear **foo** a la derecha.

Ejemplo 3.13

El código:

```
{\sc Instituto Tecnológico de Costa Rica} \hfill Tiempo: 2:45 hrs
```

produce:

INSTITUTO TECNOLÓGICO DE COSTA RICA

Tiempo: 2:45 hrs

- `\hrulefill` y `\dotfill`. Veamos ejemplos de su uso

Ejemplo 3.14

El código:

```
{\sc Instituto Tecnológico de Costa Rica} \hrulefill Tiempo: 2:45 hrs
```

produce:

INSTITUTO TECNOLÓGICO DE COSTA RICA _____Tiempo: 2:45 hrs

Ejemplo 3.15

El código:

```
{\sc Instituto Tecnológico de Costa Rica \dotfill Tiempo: 2:45 hrs}
```

produce:

INSTITUTO TECNOLÓGICO DE COSTA RICATIEMPO: 2:45 HRS

- `\rule[xcm]{y cm}{z cm}` . Este comando se usa para dibujar una línea horizontal o vertical de y cm y grosor z cm. La distancia de la línea a la base del texto se controla con el primer parámetro $[xcm]$.

Ejemplo 3.16**El código:**

```
{\sc Instituto Tecnológico de Costa Rica \hfill Tiempo: 2:45 hrs}\\
\rule[0.5cm]{14cm}{0.01cm}
```

produce:

 INSTITUTO TECNOLÓGICO DE COSTA RICA

TIEMPO: 2:45 HRS

3.2.6 Notas al pie de página.

Las notas al pie de página se producen con `\footnote{ ...texto... }`

El comando se escribe exactamente donde se quiere que quede la etiqueta que hará referencia al pie de página. La nota en el pie de esta página se generó con el código¹.

La nota...código `\footnote{Ejemplo de {\tt footnote}}`.

3.2.7 Texto como en la pantalla

Para reproducir exactamente lo que está en la pantalla (incluyendo espacios) se pone el texto en el entorno `\begin{verbatim}... \end{verbatim}`.

Ejemplo 3.17 (Usando verbatim).**El código:**

```
\begin{verbatim}
Sub Trapecio(a,b,n,delta)
Dim N As Integer
Dim F As New clsMathParser
    suma = 0
    h = (b - a) / N
For i = 1 To N - 1
    xi = a + i * h
    suma = suma + F.Eval1(xi)
Next i
End Sub
\end{verbatim}
```

produce:

```
Sub Trapecio(a,b,n,delta)
Dim N As Integer
Dim F As New clsMathParser
    suma = 0
    h = (b - a) / N
For i = 1 To N - 1
    xi = a + i * h
    suma = suma + F.Eval1(xi)
Next i
End Sub
```

¹Ejemplo de footnote

N verbatim es un comando *frágil*, en ciertos ambientes no funciona.

En la sección (11.11) se muestran varias opciones para obtener *código de programación en color* de acuerdo al lenguaje de programación.

3.3 Color y cajas con color

Paquete xcolor

Para usar colores se podría llamar al paquete **xcolor**: `\usepackage{xcolor}`. Aunque se pueden usar los colores predefinidos (`\color{black}`, `\color{red}`, `\color{blue}`, ...), también podemos personalizar los colores.

Para definir un color personalizado podemos usar alguna aplicación que tenga un selector de colores. Si, por ejemplo, el selector de colores codifica en el formato RGB², el color se puede definir así:

`\definecolor{micolor1}{RGB}{x, y, z}`, donde $x, y, z \in [0, 255]$.

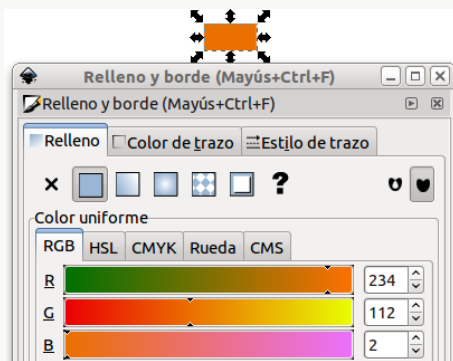
La definición de color se pone en el *preámbulo*. El formato **RGB** es adecuado para generar archivos PDF para ver en pantalla. Para imprimir se usa el formato **CMYK**.

Si el selector de colores devuelve valores entre 0 y 1 entonces sería (**rgb** en minúscula!) entonces el color se puede definir así:

`\definecolor{micolor1}{rgb}{x, y, z}`, donde $x, y, z \in [0, 1]$.

Ejemplo 3.18

Usando un selector de colores (en este caso, usamos **Inkscape**), podemos definir un anaranjado personalizado.



En **Inkscape**, el anaranjado que seleccionamos tiene parámetros RGB: 234 112 2, así que ponemos en el *preámbulo*

`\definecolor{miorange}{RGB}{234, 112, 2}`

En formato **rgb** sería

`\definecolor{miorange}{rgb}{0.9, 0.43, 0}`

La conversión de **RGB** a **rgb** se obtiene dividiendo por 255. Por ejemplo, 234 en **RGB** pasa a ser $234/255=0.91$ en **rgb**.

²RGB ("Red", "Green", "Blue") es la composición del color en términos de la intensidad de los colores primarios de la luz.

Mezclar colores.

Siempre es posible, utilizando el paquete `xcolor`, mezclar un color existente con otros colores, por ejemplo, con el color blanco para hacerlo “ligeramente” más claro.

`\color{color1<porcentaje>color2}` es una la mezcla con *un porcentaje del color original color1* y el resto del contenido de *color2*

Por ejemplo `{\color{blue!40!white} ... }` es una la mezcla con *un 40 % del color original* y un 60 % de contenido de blanco (el blanco en realidad se puede omitir y solo escribir `\color{blue!40!}`).

También `blue!80!black!30` sería $(blue*0.8 + black*0.2)*0.3 + white*0.7$

Veamos algunas mezclas:

- `{\color{blue!40!white} mezclando colores }` **produce:** mezclando colores
- `{\color{blue!40!green} mezclando colores }` **produce:** mezclando colores
- `\colorbox{blue!20!green!30!}{\parbox{10cm}{La imagen de la ...}}` **produce:**

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*

El uso de `\colorlet` podría ser más práctico, especialmente si el color se va a utilizar más de una vez, de hecho, esto define un nuevo color, pero no se aplica directamente, el uso de `\color{ }` sigue siendo necesario. Por ejemplo, el **código:**

```
\documentclass{article}
\usepackage{xcolor}
\definecolor{mibblue}{rgb}{.63,.79,.95}
\colorlet{miverdeoscuro}{blue!40!black}

\begin{document}

\colorbox{mibblue}{\parbox{10cm}{La imagen de la izquierda muestra ....}}\

{\color{miverdeoscuro} Este es un texto de prueba }

\end{document}
```

produce:

La imagen de la izquierda muestra

Este es un texto de prueba

Color en cajas

Para poner texto en una caja usando un color de fondo determinado, se usa

```
\fcolorbox{color fondo}{color borde}{ texto}
```

Ejemplo 3.19

El código:

produce:

```
\fcolorbox{orange}{orange}{ \color{white} LaTeX}
```

LaTeX

Para poner párrafos dentro de una caja se puede usar “minipage” para que el texto se acomode de manera adecuada dentro de la caja. También podemos usar “\fboxsep” para controlar el espacio entre el contenido de la caja y el rectángulo. El valor por defecto es 0pt.

Ejemplo 3.20

El código:

```
\begin{center}
{ \fboxsep 12pt
\fcolorbox {orange}{white}{
\begin{minipage}[t]{10cm}
$0^0$ es una expresión indefinida. Si $a>0$, $a^0=1$ pero $0^a=0$.
Sin embargo, convenir en que $0^0=1$ es adecuado para que
algunas fórmulas se puedan expresar de manera sencilla,
sin recurrir a casos especiales, por ejemplo
$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}$$
$$ (x+a)^n=\sum_{k=0}^n \binom{n}{k}x^k a^{n-k} $$
\end{minipage}
} }
\end{center}
```

produce:

0^0 es una expresión indefinida. Si $a > 0$, $a^0 = 1$ pero $0^a = 0$. Sin embargo, convenir en que $0^0 = 1$ es adecuado para que algunas fórmulas se puedan expresar de manera sencilla, sin recurrir a casos especiales, por ejemplo

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$(x+a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k}$$

Paquete pstricks

El paquete `pstricks` carga el paquete `xcolor` y además declara, entre muchas cosas, varios comandos de color. Para utilizarlos simplemente se carga el paquete en el preámbulo con el comando `\usepackage{pstricks}` y ahora, por ejemplo, el comando `{\blue AZUL}` produce **AZUL**. Los colores predefinidos en este paquete son:

<code>\black</code>	<code>\white</code>	<code>\cyan</code>
<code>\darkgray</code>	<code>\red</code>	<code>\magenta</code>
<code>\gray</code>	<code>\green</code>	<code>\yellow</code>
<code>\lightgray</code>	<code>\blue</code>	

3.3.1 Notas en el margen

Podemos poner figuras o texto en el margen con “`\hspace*`”. Si usamos `\hspace`, **LaTeX** quita el espacio horizontal (en blanco) que hay al final de una línea. Si queremos que **LaTeX** no elimine este espacio, se incluye el argumento opcional `*`.

Ejemplo 3.21

Podemos poner la palabra “Texto” en el margen, como se indica en el código

Texto →

```
\hspace*{-2.8cm} {\cyan \small Texto} $\longrightarrow$
```

Paquete marginnote

Hay varios paquetes para poner notas en el margen, por ejemplo el paquete “**marginnote**”. Una vez que hemos puesto `\usepackage{marginnote}` en el preámbulo, podemos poner una nota en el margen como se muestra en el margen actual; el código que produce esta nota es

podemos poner una nota en el margen como se muestra en el margen derecho; el código que produce esta nota es

```
\marginnote{\cyan \small Regla de la cadena: $z_t=\nabla z \cdot \vec{v}$}
```

Regla de la cadena:
 $z_t = \nabla z \cdot \vec{v}$

Paquete todonotes

También podemos usar el paquete **todonotes**. Formalmente es para hacer notas al margen con comentarios de “cosas por hacer”. De hecho hay una opción para hacer una listas de estas “cosas por hacer”.

3.4 Enumerado automático.

Uno de los comandos más usados para hacer listas numeradas es `enumerate`. Cada nuevo ítem se indica con `\item`, con esto se obtiene una enumeración automática. También uno puede controlar la enumeración con la etiqueta deseada. `enumerate` admite anidamiento hasta el cuarto nivel.

Ejemplo 3.22

El código:



Descargar archivo

```
...% paquetes y comandos...ver código de este ejemplo.
{\sc Instituto Tecnológico de Costa Rica} \hfill Tiempo 2:30 horas\\
{\sc Escuela de Matemática} \hfill Puntaje: 21 puntos\\
{\sc MA-0441. Primer Parcial}\\\\
{\bf Instrucciones.} Este es un examen de desarrollo, por lo tanto deben
aparecer todos los pasos que lo llevan a su respuesta. Trabaje de manera
clara y ordenada.\\
\begin{enumerate}
\item {\bf [3 Puntos]} Sea  $A=\{1,b,c,d,7\}$  y  $B=\{1,2,c,d\}$ .
    Calcule  $\mathcal{P}(A\Delta B)$ .
\item {\bf [5 Puntos]} Muestre que  $A-(B\cap C)=(A-B)\cup(A-C)$ 
\item {\bf [5 Puntos]} Mostrar que  $[A\cup C\subseteq B\cup C \wedge A\cap C=\emptyset] \implies A\subseteq B$ 
\item {\bf [2 Puntos]} Sea  $\Re=(\mathbb{R}^*,\mathbb{R}^*,R)$ 
    definida por  $x\Re y \iff xy>0$ .
    \begin{enumerate}
\item {\bf [3 Puntos]} Muestre que  $\Re$  es una relación de equivalencia.
\item {\bf [2 Puntos]} Determine las clases de equivalencia  $\overline{1}$  y  $\overline{-1}$ .
\item {\bf [1 Punto]} Determine  $\mathbb{R}^*/\Re$  (el conjunto cociente).
\end{enumerate}
\end{enumerate}
```

produce:

INSTITUTO TECNOLÓGICO DE COSTA RICA
 ESCUELA DE MATEMÁTICA
 MA-0441. PRIMER PARCIAL

Tiempo 2:30 horas
 Puntaje: 21 puntos

Instrucciones: Este es un examen de desarrollo, por lo tanto deben aparecer todos los pasos que lo llevan a su respuesta. Trabaje de manera clara y ordenada.

1. [3 Puntos] Sea $A = \{1, b, c, d, 7\}$ y $B = \{1, 2, c, d\}$. Calcule $\mathcal{P}(A \Delta B)$.
2. [5 Puntos] Muestre que $A - (B \cap C) = (A - B) \cup (A - C)$
3. [5 Puntos] Mostrar que $[A \cup C \subseteq B \cup C \wedge A \cap C = \emptyset] \implies A \subseteq B$
4. [2 Puntos] Sea $\Re = (\mathbb{R}^*, \mathbb{R}^*, R)$ definida por $x \Re y \iff xy > 0$.
 - a) [3 Puntos] Muestre que $\Re$ es una relación de equivalencia.
 - b) [2 Puntos] Determine las clases de equivalencia $\overline{1}$ y $\overline{-1}$.
 - c) [1 Punto] Determine $\mathbb{R}^*/\Re$ (el conjunto cociente).

El entorno `itemize` usa puntos u otros símbolos para los items mientras que `description` permite descriptores con texto. Todo esto se puede hacer con `enumerate`. En los ejemplos que siguen se muestran varias posibilidades.

Ejemplo 3.23 (Lista con `description`)

```
\begin{description}
  \item[Media muestral:] $\frac{1}{n-1}\sum_{i=1}^n (X_i-\bar{X}_n)^2$
  \item[Varianza muestral:] $\frac{1}{n-1} \sum_{i=1}^n (X_i-\bar{X}_n)^2$
  \item[Momentos muestrales:] $\frac{1}{n} \sum_{i=1}^n X_i^k$
\end{description}
```

> En el capítulo que sigue veremos cómo variar el tamaño de los símbolos

produce:

Media muestral: $\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2$

Varianza muestral: $\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2$

Momentos muestrales: $\frac{1}{n} \sum_{i=1}^n X_i^k$

Ejemplo 3.24 (Lista con `enumerate` e `itemize`)

Ejemplo con 4 niveles (máximo permitido). El código:

```
\begin{enumerate}
  \item[\fbox{1.}] {\bf Procedimiento}{\em Aprendizaje}
  \item[\fbox{2.}] {\bf comienzo} % Descriptor personalizado
  \begin{enumerate}
    \item Paso a.
    \item Paso b.
    \begin{enumerate}
      \item Paso c.1
      \item Paso c.2
      \begin{enumerate}
        \item Paso c.2.1
        \item Paso c.2.2
      \end{enumerate}
    \end{enumerate}
    \item Paso c.3
  \end{enumerate}
  \item Paso d.
\end{enumerate}
\item[\fbox{3.}] {\bf fin}
\end{enumerate}

% ITEMIZE
\begin{itemize}
```

```
\item {\red Sugerencia 1}
\item {\red Sugerencia 2}
\end{itemize}
```

produce:

1. Procedimiento *Aprendizaje*
2. comienzo
 - a) Paso a.
 - b) Paso b.
 - 1) Paso c.1
 - 2) Paso c.2
 - a' Paso c.2.1
 - b' Paso c.2.2
 - 3) Paso c.3
 - c) Paso d.
3. fin
 - Sugerencia 1
 - Sugerencia 2

Los símbolos que **enumerate** pone por defecto para enumerar una lista se pueden cambiar redefiniendo los comandos **labelenumi**, **labelenumii**, **labelenumiii** y **labelenumiv**.

Ejemplo 3.25

El código:

```
\renewcommand{\labelenumi}{\Roman{enumi}.}
\renewcommand{\labelenumii}{\arabic{enumii}$}$ }
\renewcommand{\labelenumiii}{\alph{enumiii}$}$ }
\renewcommand{\labelenumiv}{\bullet$ }
\begin{enumerate}
  \item Primer nivel (en Romanos)

  \begin{enumerate}
    \item Segundo nivel (en numeración arábica)

    \begin{enumerate}
      \item Tercer nivel (numeración alfabética)

      \begin{enumerate}
        \item Cuarto nivel (usamos {\tt bullet})
      \end{enumerate}
    \end{enumerate}
  \end{enumerate}
\end{enumerate}
```

```

\end{enumerate}

\end{enumerate}

\end{enumerate}
\end{enumerate}

```

produce:

- I. Primer nivel (en Romanos)
 - 1) Segundo nivel (en numeración arábica)
 - a) Tercer nivel (numeración alfabética)
 - Cuarto nivel (usamos bullet)

(N) Si se va a utilizar el mismo sistema de numeración durante todo el documento, estas instrucciones se pueden poner en el preámbulo.

Personalizar el entorno “enumerate”.

Se puede cambiar muchas cosas en el entorno “**enumerate**”, por ejemplo márgenes, espacio entre items, color, etc. Todo esto se puede hacer con el paquete **enumitem**. La documentación del paquete la puede encontrar en <http://www.ctan.org/pkg/enumitem>.

Para usar este paquete, debemos poner en el *preámbulo*

```
\usepackage[shortlabels]{enumitem} % Versión 3.0 o +
```

Con este paquete, entre otras cosas, nos permite opciones del tipo

```
\begin{enumerate}[opciones]
```

Ejemplo 3.26 (Algunas opciones con el paquete “enumitem”)

El **código**:

```

\begin{enumerate}[label=\emph{\alph*}]
\item Uno
\item Dos
\end{enumerate}

% resume= continuación de la numeración anterior
"resume" continúa una lista pero solo localmente,\
\begin{enumerate}[label=\emph{\alph*}], resume]

```

```
\item Tres
\item Cuatro
\end{enumerate}
```

Agregar un texto en negrita al item y correr margen izquierdo\\

```
\begin{enumerate}[label=\textbf{Idea (\emph{\alph*})}, leftmargin=2cm]
\item De nuevo Uno
\item Dos
\end{enumerate}
```

produce:

- a) Uno
- b) Dos

“resume” continúa una lista pero solo localmente,

- c) Tres
- d) Cuatro

Agregar un texto en negrita al item y correr margen izquierdo:

- Idea (a)** De nuevo Uno
- Idea (b)** Dos

La opción **shortlabels** que cargamos con el paquete **enumitem** permite poner etiquetas de manera simple.

Ejemplo 3.27

El **código:**

produce:

<code>% \usepackage[shortlabels]{enumitem}</code>	Etiquetas (a) , (b) ,...
<code>% Etiqueta (a.) -> (a), (b),...</code>	
<code>\begin{enumerate}[(a.)]</code>	
<code>\item \$11^2=121\$</code>	(a.) $11^2 = 121$
<code>\item \$(1+1)^2=1+2+1\$</code>	(b.) $(1+1)^2 = 1+2+1$
<code>\item \$12^2=144\$</code>	(c.) $12^2 = 144$
<code>\item \$(1+2)^2=1+4+4\$</code>	(d.) $(1+2)^2 = 1+4+4$
<code>\end{enumerate}</code>	
Otra lista con A.1), A.2),...\\	Otra lista con A.1) , A.2) ,...
<code>\begin{enumerate}[font=\sffamily\bfseries,</code>	
<code>label=A.\arabic*)]</code>	
<code>\item \$11^2=121\$</code>	A.1) $11^2 = 121$
<code>\item \$(1+1)^2=1+2+1\$</code>	A.2) $(1+1)^2 = 1+2+1$
<code>\item \$12^2=144\$</code>	A.3) $12^2 = 144$
<code>\item \$(1+2)^2=1+4+4\$</code>	A.4) $(1+2)^2 = 1+4+4$
<code>\end{enumerate}</code>	

La opción **resume** (para continuar una lista) se puede usar “globalmente” con la opción **series**

Ejemplo 3.28 (Opción “resume” global)

Continuación de una lista en un ambiente **minipage**: El tipo de etiquetas se declara en las opciones como `series = nombre_etiquetas`, luego en la lista que sigue se invoca en las opciones con `resume*=nombre_etiquetas`. En este ejemplo usamos **series = Aes** porque vamos a etiquetar con **A.1**, **A.2**,...

El **código**:

```
\begin{minipage}{7cm}
\begin{enumerate}[font=\sffamily\bfseries,label=A.\arabic*), series=Aes]
\item $11^2=121$
\item $(1+1)^2=1+2+1$
\item $12^2=144$
\item $(1+2)^2=1+4+4$
\end{enumerate}
\end{minipage} \hfill \begin{minipage}{7cm}
\begin{enumerate}[label=I.\arabic*), resume*=Aes]
\item $3^3 + 4^4 + 3^3 + 5^5 = 3435$
\item $2^4 = 4^2$
\item $2592=2^59^2$
\item $10^2+11^2+12^2=13^2+14^2$
\end{enumerate}
```

`\end{minipage}`

produce:

A.1) $11^2 = 121$

A.5) $3^3 + 4^4 + 3^3 + 5^5 = 3435$

A.2) $(1 + 1)^2 = 1 + 2 + 1$

A.6) $2^4 = 4^2$

A.3) $12^2 = 144$

A.7) $2592 = 2^5 9^2$

A.4) $(1 + 2)^2 = 1 + 4 + 4$

A.8) $10^2 + 11^2 + 12^2 = 13^2 + 14^2$

Enumeración por filas

Hay varias opciones para hacer listas enumeradas en varias columnas, de tal manera que la enumeración *transcurra por filas*. Una opción es usar el paquete `tabularx` con un entorno especial “`enumfilas`” [27].

a) $\lim_{x \rightarrow 0} h(x)$

b) $\lim_{x \rightarrow 0} h(x)$

c) $\lim_{x \rightarrow 0} h(x)$

d) $\lim_{x \rightarrow 0} h(x)$

e) $\lim_{x \rightarrow 0} h(x)$

f) $\lim_{x \rightarrow 0} h(x)$

El código completo es,

```
\documentclass{article}
\usepackage{tabularx}
% contador
\newcounter{lasfilas}
\renewcommand\thelasfilas{\alph{lasfilas}}
\newenvironment{enumfilas}[1]
{
  \setcounter{lasfilas}{0}
  \par\noindent\tablex{\the}[t]
  {\*{#1}{>\stepcounter{lasfilas}\makebox[1.8em][l]{\thelasfilas}\hfill}}X}}%
}{\endtablex}

\begin{document}

\begin{enumfilas}{3}
  $\lim\limits_{x \rightarrow 0} h(x)$ & $\lim\limits_{x \rightarrow 0} h(x)$ & &
  $\lim\limits_{x \rightarrow 0} h(x)$ & $\lim\limits_{x \rightarrow 0} h(x)$ & \\
  $\lim\limits_{x \rightarrow 0} h(x)$ & $\lim\limits_{x \rightarrow 0} h(x)$ & &
  $\lim\limits_{x \rightarrow 0} h(x)$ & $\lim\limits_{x \rightarrow 0} h(x)$ & \\
  $\lim\limits_{x \rightarrow 0} h(x)$ & $\lim\limits_{x \rightarrow 0} h(x)$ & &
  $\lim\limits_{x \rightarrow 0} h(x)$ & $\lim\limits_{x \rightarrow 0} h(x)$ & \\
\end{enumfilas}

\end{document}
```

3.4.1 Enumeración usando el paquete TikZ.

Podemos crear discos con números para usar en un entorno `enumerate`. Algo como

produce:

- ➊ Paso 1
- ➋ Paso 2
- ➌ Paso 3

42

3.5 Título, contenido, secciones y bibliografía

El código que sigue es el de una *plantilla básica* para un documento clase book. Para usar otras facetas, se debe invocar los paquetes respectivos. Si tiene una distribución completa y *actualizada* de **TeX** no tendrá problemas.




[Descargar archivo](#)

```
\documentclass{book}
% Dimensiones y márgenes-----
\usepackage[total={18cm,21cm},top=2cm, left=2cm]{geometry}
\parindent=0mm
% Otros paquetes -----
\usepackage[spanish]{babel} % Idioma español
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc} %
\usepackage{mathpazo} % fuente palatino
\usepackage{graphicx,pstricks}
\usepackage{latexsym,amsmath,amssymb,amsfonts,cancel}
\usepackage[shortlabels]{enumitem}
% Referencias - ligas
\usepackage[hyphens]{url}
\usepackage[breaklinks,colorlinks=true,linkcolor=red,
citecolor=red, urlcolor=blue]{hyperref}
% Comandos -----
\newcommand{\sen}{\mathop{\rm sen}\nolimits} % seno
\newcommand{\arcsen}{\mathop{\rm arcsen}\nolimits}
\newcommand{\arcsec}{\mathop{\rm arcsec}\nolimits}
\setcounter{chapter}{0}
\newtheorem{teo}{Teorema}[chapter] % entorno para teoremas
\newtheorem{ejemplo}{Ejemplo}[chapter] % entorno para ejemplos
\newtheorem{defi}{Definición}[chapter] % entorno para definiciones
% -----
\begin{document}
\title{\Huge Manual de LaTeX \\\
\small \gray {\fontfamily{phv}\selectfont % gris y Helvetica
Instituto Tecnológico de Costa Rica\\
Escuela de Matemática\\
```

```

Enseñanza de la Matemática\\
}

}}
\author{Preparado por Prof. Walter Mora F. y Alexander Borbón A.}
\date{2013}
\maketitle % despliega el título
\tableofcontents

\chapter{LaTeX }
\section{¿Qué es LaTeX ?}
...
\subsection{Preámbulo}
...
\subsubsection{Acerca del Título}
...
\section{Definiciones, teoremas y ejemplos}
% Definición
\begin{defi}  $f$  es de clase  $C^1[a,b]$  si ....
\end{defi}
...
% Teorema
\begin{teo} {\rm Si  $f \in C^1[a,b]$  entonces....} % fuente roman normal
\end{teo}
...
% Ejemplo
\begin{ejemplo} Si  $f(x)=\frac{1}{x-2}$  entonces  $f \in C^1[-1,1]$ .
\end{ejemplo}
...
% Bibliografía -----
\addcontentsline{toc}{chapter}{Bibliografía} % agregar al Índice
\begin{thebibliography}{99}
\bibitem{Hahn} Hahn, J. {\tt \fnt{LaTeX}}  $\backslash$ , for everyone''. Prentice Hall,
New Jersey, 1993.

\end{thebibliography}

\end{document}

```

Título

Como se observa, el título se define con el comando `\title`, además se pueden definir el o los autores con el comando `\author` y la fecha se puede definir con el comando `\date`, para que no salga fecha se deja este comando en blanco. Por último para que aparezca el título en el documento se debe poner el comando `\maketitle`.

Tabla de contenidos

LaTeX realiza automáticamente la tabla de contenidos de un documento, tan solo se debe poner el comando `\tableofcontents` en donde se quiera que aparezca.

```
\begin{document}
```

```
\title{...}
\maketitle
\tableofcontents %
```

Capítulos y Secciones

Para crear capítulos en un documento se utiliza el comando `\chapter`, las secciones del capítulo se definen con el comando `\section` y estas secciones se pueden dividir en subsecciones y subsubsecciones con los comandos `\subsection` y `\subsubsection`.

Las secciones dependen del tipo de documento que se esté realizando; por ejemplo, un artículo no posee capítulos, sólo secciones y subsecciones pero sí tiene un resumen (se usa el comando `\abstract`), un libro no tiene resumen pero sí tiene prefacio, éste se pone con el comando `\begin{preface}...\end{preface}`.

La Bibliografía

En la plantilla que sigue se muestra de nuevo el ambiente para una bibliografía sencilla, es un ambiente parecido a `enumerate`.

```
\addcontentsline{toc}{chapter}{Bibliografía}
\begin{thebibliography}{99}
  \bibitem{Hahn} Hahn, J. {\it \fnt{LaTeX} \, for everyone}. Prentice Hall,
    New Jersey, 1993.
\end{thebibliography}
```

Referencias a la bibliografía: cite

En el texto uno puede hacer referencia a algún ítem de la bibliografía. Para hacer esto, le ponemos una etiqueta al ítem: `\bibitem{Hahn}` hace que **Hahn** sea la referencia a este libro.

```
\begin{thebibliography}{99}
...
\bibitem{Hahn} Hahn, J. {\it \fnt{LaTeX} \, for everyone'}.
  Prentice Hall, New Jersey, 1993.
\end{thebibliography}
```

Entonces podemos hacer referencia a este libro en el texto con `\cite[nota]{clave}` o solamente con `\cite{clave}`.

Ejemplo 3.30

El texto "En `\cite[pág. 80]{Hahn}` se pueden ver los aspectos..."

produce: "En [20, pág. 80] se pueden ver los aspectos relativos a ...".

El uso del ambiente `thebibliography` se profundiza en la sección 8.1 donde también se muestra el uso de **BibTeX** para la creación de bibliografías en distintos formatos.

Referencia a definiciones, teoremas, etc.

Algo similar podemos hacer en los ejemplos, teoremas, definiciones, etc. Para estos usamos una etiqueta (label) para identificarlos. Por ejemplo, si ponemos

```
% El entorno "teo" fue definido en el preámbulo
\begin{teo}[Teorema del Valor Medio]\label{tvm1}
Sea  $f(x)$  continua en  $[a,b]$  y derivable en  $]a,b[$ ,
entonces  $\exists \xi \in ]a,b[$  tal que

$$f(a) - f(b) = f'(\xi)(b-a)$$

\end{teo}
```

podemos hacer referencia a este teorema así:

Teorema 3.5.1 (Teorema del valor Medio). Sea $f(x)$ continua en $[a,b]$ y derivable en $]a,b[$, entonces $\exists \xi \in]a,b[$ tal que

$$f(a) - f(b) = f'(\xi)(b-a)$$

“En el teorema \ref{tvm1} se indica que ...” **produce:**

En el teorema 3.5.1 se indica que

Lo mismo podemos hacer en los ambientes **table**, **figure**, **eqnarray**, **equation**, etc.

N El color de la liga se puede cambiar en las opciones del paquete **hyperref**,

```
\usepackage[colorlinks=true, linkcolor=red,
             citecolor=red, urlcolor=blue]{hyperref}
```

3.5.1 Índice alfabético

Para generar un índice alfabético se debe cargar el paquete `makeidx`, para esto se escribe en el preámbulo: `\usepackage{makeidx}` y se debe compilar tres veces en el orden: **PDFLaTeX** - **makeindex** - **PDFLaTeX**;

Una plantilla básica para crear un índice alfabético sería

```
\documentclass{book}
...
\usepackage{makeidx}
\makeindex
...
\begin{document}
...
\index{Entrada para el Índice}
...
\printindex
\end{document}
```

La instrucción `\makeindex` se utiliza para inicializar el índice. Cada vez que se quiera agregar una palabra al índice alfabético se utiliza el comando `index` (se coloca cerca de la palabra). En

el siguiente ejemplo se muestra el número de página en la que se puso el texto correspondiente.

Ejemplo 3.31 (Índice alfabético).

Si se escribe:

Página 1: `\index{Manzana}`
 Página 6: `\index{Naranja}`
 Página 10: `\index{Banano}`
 Página 12: `\index{Naranja}`

se produce:

Banano, 10
 Manzana, 1
 Naranja, 6, 12

Observe cómo **LaTeX** acomoda automáticamente de manera alfabética las palabras e indica en las páginas en las que aparecen. Dentro del documento se escribe el comando `\printindex` donde se quiera que aparezca el índice alfabético.

Compilar con índice alfabético

Para compilar un documento que tenga un índice alfabético se debe compilar con **PDFLaTeX** - **makeindex** - **PDFLaTeX**; es decir, primero compilar con **PDFLaTeX**, esto genera un archivo con todas las entradas del índice, luego opción **makeindex** que genera un nuevo archivo con el índice formateado correctamente y en orden alfabético y, por último nuevamente **PDFLaTeX** para que genere el documento con el índice correctamente. El comando `\printindex` al final del documento ordena la salida final.

El comando `\index` tiene algunas opciones adicionales.

Ejemplo 3.32 (Índice alfabético).

El código:

Página 1: `\index{Manzana|textbf}`
 Página 6: `\index{Naranja|}`
 Página 8: `\index{Manzana!De agua}`
 Página 10: `\index{Banano|see{Manzana}}`
 Página 12: `\index{Naranja|)}`
 Página 13: `\index{Manzana!Nacional}`
 Página 14: `\index{Beta@$\beta$}`
 Página 14: `\index{{Manzana!Americana}}`

produce:

Banano, *see* Manzana
 β , 14
 Manzana, 1
 Americana, 14
 De agua, 8
 Nacional, 13
 Naranja, 6–12

De estos casos se puede observar que si se agrega el comando `\textbf` se logra que el número de la página en el índice salga en negrita, también se puede lograr que salga en itálica con el comando `\textit`.

Si se coloca `| (` se inicia un rango de páginas que termina cuando se ponga la misma palabra terminada por `| )`, como 'la Naranja' del ejemplo

Para una palabra se puede poner un segundo nivel de palabras clave con el signo de admiración como el que se hizo con las manzanas del ejemplo.

Se puede hacer una referencia cruzada desde una palabra a cualquier otra agregando el comando `see{Llave}`, también existe el comando `|seealso{Llave}` (ver también). La palabra “see” y “see also” saldrán en español si se ha cargado el paquete `babel` (ver sección 2.5)

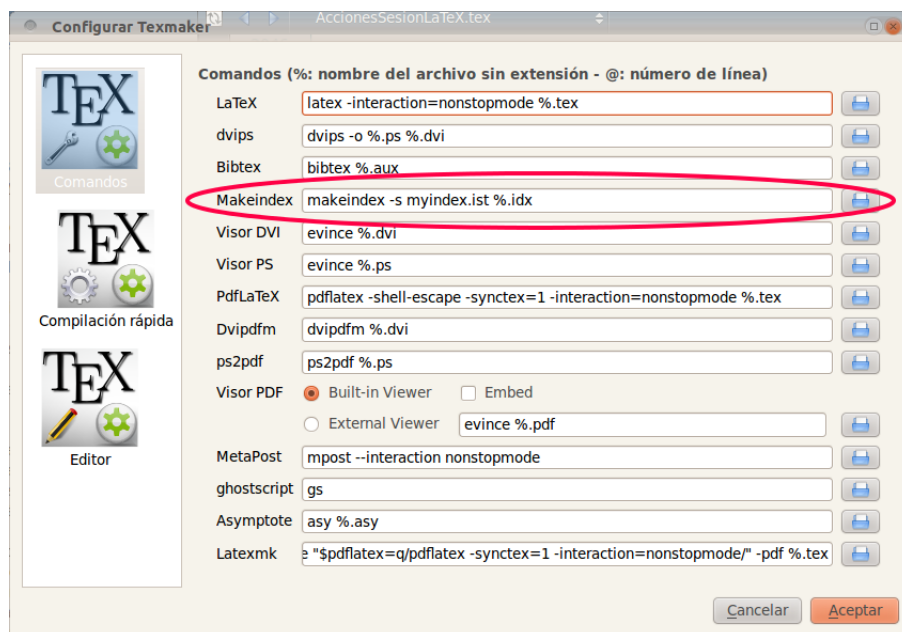
También se puede utilizar el “ para ordenar alfabéticamente un signo o alguna palabra diferente, por ejemplo, anteriormente se utilizó `\index{Beta@$\beta$}` que logra que el símbolo matemático β aparezca en el índice alfabético pero que se ordene alfabéticamente bajo el nombre de Beta.

Por último, es muy común que se quiera hacer que en el índice alfabético aparezca una letra antes de las palabras (tal y como se hizo el índice alfabético de este libro), para lograrlo se debe hacer un archivo aparte (que en nuestro caso lo llamamos **myindex.ist**), este archivo se puede hacer con el Bloc de Notas de Windows o el Editor de Textos en Linux y se debe poner en la misma carpeta donde está el archivo `.tex` que estamos editando. El archivo **myindex.ist** debe tener las siguientes líneas:

```
% MakeIndex style file myindex.ist
heading_prefix "{\bfseries " % Insert in front of letter
heading_suffix "\\hfil}\\nopagebreak\n" % Append after letter
headings_flag 1 % Turn on headings (uppercase)
```

Ahora se debe ir a las **Opciones** del editor que se esté utilizando y en el comando para compilar **makeindex** se debe agregar la opción **-s myindex.ist**.

Por ejemplo, en el caso de **TeXMaker** quedaría como se observa en la figura que sigue,



Utilizando el último ejemplo ahora se obtiene

B
 Banano, *see* Manzana
 β , 14

M
 Manzana, 1
 Americana, 14
 De agua, 8
 Nacional, 13

N
 Naranja, 6–12

3.6 Modulación

Para evitar la incomodidad de mantener archivos muy grandes, es conveniente modular el texto separándolo en varios archivos *sin preámbulo ni* `\begin{document}...` `\end{document}`. Por ejemplo, este texto tenía la siguiente plantilla

```
\documentclass{report}
\textheight=20cm
\textwidth=18cm
\topmargin=-2cm
...
\begin{document}
\input cap1.tex
\pagebreak
\input cap2.tex
\pagebreak
...
\end{document}
```

Cada archivo *.tex fue editado con el preámbulo hasta que estuviera afinado. Luego se recortó el archivo.

3.7 (*) Más cosas sobre las fuentes.

Paquete textcomp

Hay algunos caracteres especiales que requieren el paquete textcomp. Para usar este paquete agregamos en el *preámbulo*, `\usepackage{textcomp}`

Por ejemplo, el acento: ` se obtiene con el comando `\textasciigrave`, el apóstrofo: ' se obtiene con el comando `\textquotesingle` y el símbolo de copyright: © se obtiene con `\textcopyright`. La lista de símbolos se puede obtener en

<http://home.online.no/~pjacklam/latex/textcomp.pdf>

Tamaño global de fuentes.

La fuente default que usa **LaTeX** es de tamaño 10pt (72pt = 1 pulgada). Este tamaño lo pode-

mos cambiar a 11pt o 12pt agregando esta opción, por ejemplo `\documentclass[12pt]{article}`.

Si por alguna razón queremos variar este tamaño, por ejemplo para adaptar un documento para leer en un lector *Kindle*, para un libro de cuentos, poesía, tesis, etc., podemos usar la clase `memoir`, así tendremos soporte para fuentes de tamaño 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt, 25pt, 30pt, 36pt, 48pt y 60pt, además de muchas cosas adicionales de estilo. Esta clase se usan de la manera usual,

```
\documentclass[letterpaper,12pt,extrafontsizes]{memoir}
\usepackage{latexsym,amsmath,amssymb,amsfonts}
\begin{document}
  Después de haber definido  $e^x$  para  $x$  real cualquiera,
  es preferible definir  $a^x$  por la fórmula  $a^x=e^{x\log a}$ ...
\end{document}
```

8pt	10pt	14pt
Después de haber definido e^x para x real cualquiera, es preferible definir a^x por la fórmula $a^x = e^{x\log a}$	Después de haber definido e^x para x real cualquiera, es preferible definir a^x por la fórmula $a^x = e^{x\log a}$	Después de haber definido e^x para x real cualquiera, es preferible definir a^x por la fórmula $a^x = e^{x\log a}$

3.7.1 Las fuentes y sus atributos

LaTeX no usa las fuentes del sistema operativo, más bien usa las fuentes instaladas por default en la distribución $\text{\TeX}$. Otras fuentes especiales se pueden agregar de manera automática usando paquetes. Si una fuente está disponible, puede aplicar esta fuente a parte de un texto o de manera global.

Una lista de fuentes disponibles en **LaTeX** se puede encontrar en

“The LaTeX Font Catalogue” en <http://www.tug.dk/FontCatalogue/>

Cualquier fuente en **LaTeX** tiene cinco atributos: `encoding`, `family`, `series`, `shape`, `size`.

Ya hemos usado estos atributos para la fuente estándar. En un texto normal uno usa varios tipos de fuentes por eso es conveniente conocer como se hace en **LaTeX** estos cambios.

\fontencoding{}: Es la manera de identificar los caracteres usando números. Por ejemplo, el carácter $\sim$ lo podemos obtener directamente del teclado (pues estamos usando el paquete `inputenc` en la codificación `latin1`), pero lo podemos obtener también usando su codificación `latin1`: Este carácter se puede obtener con el comando `\char126`. El mismo código nos da otro carácter en el caso de que cambiemos a la codificación `OML`, por ejemplo.

\fontfamily{}: Nombre de la colección de fuentes. Familias comunes son

<code>\fontfamily{pag}\selectfont</code>	Avant Garde	Avant Garde
<code>\fontfamily{fvs}\selectfont</code>	Bitstream Vera Sans	Bitstream Vera Sans
<code>\fontfamily{pbk}\selectfont</code>	Bookman	Bookman
<code>\fontfamily{bch}\selectfont</code>	Charter	Charter
<code>\fontfamily{ccr}\selectfont</code>	Computer Concrete	Computer Concrete
<code>\fontfamily{cmr}\selectfont</code>	Computer Modern	Computer Modern
<code>\fontfamily{pcr}\selectfont</code>	Courier	Courier
<code>\fontfamily{phv}\selectfont</code>	Helvetica	Helvetica
<code>\fontfamily{fi4}\selectfont</code>	Inconsolata	Inconsolata
<code>\fontfamily{lmr}\selectfont</code>	Latin Modern	Latin Modern
<code>\fontfamily{lmss}\selectfont</code>	Latin Modern Sans	Latin Modern Sans
<code>\fontfamily{lmitt}\selectfont</code>	Latin Modern Typewriter	Latin Modern Typewriter
<code>\fontfamily{pnc}\selectfont</code>	New Century Schoolbook	N C Schoolbook
<code>\fontfamily{ppl}\selectfont</code>	Palatino	Palatino
<code>\fontfamily{ptm}\selectfont</code>	Times	Times
<code>\fontfamily{uncl}\selectfont</code>	Uncial	Uncial
<code>\fontfamily{put}\selectfont</code>	Utopia	Utopia
<code>\fontfamily{pzc}\selectfont</code>	Zapf Chancery	Zapf Chancery

`\fontseries{}`: “Peso” de la fuente.

<code>m</code>	Medium
<code>b</code>	Bold
<code>bx</code>	Bold extended
<code>sb</code>	Semi-bold
<code>c</code>	Condensed

`\fontshape{}`: Forma de la fuente.

<code>n</code>	Normal
<code>it</code>	Italic
<code>sl</code>	Slanted (``oblicua'')
<code>sc</code>	Caps and small caps

`\fontsize{tamaño}{baselineskip}`: Tamaño de la fuente y separación vertical entre líneas en un mismo párrafo (baselineskip).

Usualmente usamos instrucciones tales como

```
{\fontfamily{...}\selectfont{ texto}}
{\fontencoding{...}\fontfamily{...}\selectfont{ texto }}
{\fontencoding{...}\fontfamily{...} \fontseries{b}\selectfont{ texto }}
```

Ejemplo 3.33

Para usar la fuente Calligra, debemos poner en el *preámbulo*

```
\usepackage[T1]{fontenc}
\usepackage{calligra}
```

Luego podemos cambiar la fuente de una parte del texto. Por ejemplo, en la palabra ‘Había’ podemos cambiar la fuente y el tamaño para la ‘H’ solamente:

```
{\fontfamily{calligra}\fontsize{30}{1}
\selectfont{H}}\normalfont abía una vez ...
```

Esta instrucción **produce**: Había una vez ...

Podemos aplicar el tipo de fuente a partes determinadas del documento y regresar después a la fuente normal. Esto se hace usando el comando `\normalfont`. Por ejemplo,

```
Gauss publicó
{\fontfamily{hv}\selectfont{\blue Disquisitiones Arithmeticae}}
\normalfont tres años más tarde...
```

produce: Gauss publicó *Disquisitiones Arithmeticae* tres años más tarde...

Cambio global de fuente.

Para hacer un cambio global solo debe declarar el paquete correspondiente en el *preámbulo* del documento. Por ejemplo

<code>\usepackage{pslatex}</code>	<code>\usepackage{concrete}</code>
<code>\usepackage{bookman}</code>	<code>\usepackage{cmbright}</code>
<code>\usepackage{helvet}</code>	<code>\usepackage{fourier}</code>
<code>\usepackage{palatino}</code>	<code>\usepackage{mathptmx}</code>
<code>\usepackage{newcent}</code>	<code>\usepackage{mathpazo}</code>
<code>\usepackage{pxfonts}</code>	<code>\usepackage{concrete,eulervm}</code>
<code>\usepackage{txfonts}</code>	<code>\usepackage{pslatex,concrete}</code>

Este libro usa el paquete `mathpazo`.

Ejemplo 3.34

Fuentes `concrete` y `mathptmx`

La expresión en paréntesis cuadrados es un *promedio* de los valores de f'' en $[a, b]$, por lo tanto este promedio está entre el máximo y el mínimo absoluto de f'' en $[a, b]$ (asumimos f'' continua). Finalmente, por el teorema del valor intermedio, existe $\xi \in]a, b[$ tal que $f''(\xi)$ es igual a este valor promedio, es decir

$$-\frac{h^3}{12} \sum_{k=0}^{n-1} f''(\eta_k) = -\frac{(b-a)h^2}{12} \cdot f''(\xi), \quad \xi \in]a, b[$$

`mathptmx`

La expresión en paréntesis cuadrados es un *promedio* de los valores de f'' en $[a, b]$, por lo tanto este promedio está entre el máximo y el mínimo absoluto de f'' en $[a, b]$ (asumimos f'' continua). Finalmente, por el teorema del valor intermedio, existe $\xi \in]a, b[$ tal que $f''(\xi)$ es igual a este valor promedio, es decir

$$-\frac{h^3}{12} \sum_{k=0}^{n-1} f''(\eta_k) = -\frac{(b-a)h^2}{12} \cdot f''(\xi), \quad \xi \in]a, b[$$

Información adicional se puede obtener en [4] y [19].

3.7.2 Usando las fuentes del sistema con XeLaTeX .

LaTeX no usa las fuentes del sistema. **XeLaTeX** es una variante de **TeX** que puede usar las fuentes instaladas en el sistema operativo (si requiere un trabajo más profesional). Viene incluido en **MiKTeX** (2.8 en adelante) y en **TeXLive** 2010 en adelante.

Un documento **XeLaTeX** lo editamos de la manera usual (la codificación debe ser UTF8) por ejemplo

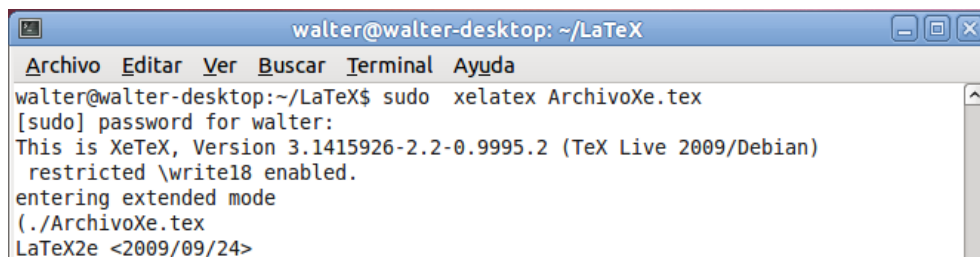
```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{xltextra}
\setmainfont[Mapping=tex-text]{Lucida Bright} % fuente en el sistema

\begin{document}
Después de haber definido  $e^x$  para  $x$  real cualquiera, no hay ninguna
dificultad para dar una definición de  $a^x$  para cada  $a > 0$ .
Un método es definir  $a^x$  como el número  $y$  tal que  $\log y = x$ ;
claro que este método no sirve para  $a = 1$  puesto que el logaritmo de base
1 no está definido. Otro modo es definir
 $a^x$  por la fórmula:


$$a^x = e^{x \log a}$$

\end{document}
```

Para compilar se usa el comando `xelatex` (presente en **Kile**; en **TeXMaker** hay que configurarlo). Si no se tiene el comando en el editor, abrimos una *terminal* y ejecutamos (en la carpeta correcta): `xelatex ArchivoTal.tex`. Por ejemplo en **Ubuntu** sería,



```
walter@walter-desktop: ~/LaTeX
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
walter@walter-desktop:~/LaTeX$ sudo xelatex ArchivoXe.tex
[sudo] password for walter:
This is XeTeX, Version 3.1415926-2.2-0.9995.2 (TeX Live 2009/Debian)
restricted \write18 enabled.
entering extended mode
(. /ArchivoXe.tex
LaTeX2e <2009/09/24>
```

El resultado es un texto con fuente **Lucida Bright** (presente en mi sistema).

Después de haber definido e^x para x real cualquiera, no hay ninguna dificultad para dar una definición de a^x para cada $a > 0$. Un método es definir a^x como el número y tal que $\log y = x$; claro que este método no sirve para $a = 1$ puesto que el logaritmo de base 1 no está definido. Otro modo es definir a^x por la fórmula:

$$a^x = e^{x \log a}$$

Un manual de referencia se encuentra en <http://tug.ctan.org/tex-archive/info/xetexref/XeTeX-reference.pdf>.

(*) El Paquete fancyvrb

Hay cosas que no podemos hacer con el entorno verbatim pero que en algunos contextos son necesarios. Por ejemplo, usar verbatim para una nota al pie de página o usar símbolos matemáticos, color, etc.; dentro del ambiente verbatim. Estas cosas se pueden hacer con el paquete fancyvrb: Ponemos en el preámbulo `\usepackage{fancyvrb}`

Con este paquete ahora ya podríamos usar verbatim en notas al pie de página. Se usa footnote de la manera usual pero, en este caso, precedido por `\VerbatimFootnotes`. Por ejemplo,

El peor ejemplo de programación recursiva es la de la función factorial

```
\VerbatimFootnotes \footnote{ Se refiere al código
\begin{verbatim}
  int factorial(int n){
    if (n == 0) return 1;
    return n * factorial(n-1);}
\end{verbatim}
} aunque es un ejemplo muy claro.
```

(*) Entorno Verbatim del paquete fancyvrb

A veces es adecuado introducir texto en modo matemático en un ambiente verbatim así como otros efectos. Para hacer esto, usamos el entorno Verbatim del paquete fancyvrb. Notar la mayúscula: `Verbatim`. El entorno sería

```
\begin{Verbatim}[opciones]
...
\end{Verbatim}
```

En el ejemplo que sigue, se usan varias opciones: `'frame=lines'` para poner segmentos de línea al inicio y al final, `'xleftmargin'` y `'xrightmargin'` para ajustar estos segmentos. `'commandchars'` se usa para indicar que `\` se va usar para aplicar comandos **LaTeX** en el entorno y `catcode` para especificar los caracteres de código matemático que se van a permitir en este ambiente, en este ejemplo caso: `$`, `^`, `\` y `_`.

Ejemplo 3.35

El código:

```
\begin{Verbatim}[xleftmargin=3.1cm,xrightmargin=4.5cm,resetmargins=true,
  frame=lines,formatcom=\color{blue},fontfamily=ptm, commandchars=\\{\},
  codes={\catcode`$=3\catcode`^=7\catcode`\_ =8}] % fin de opts

$x$          $y=x^2$
0.000005     2.5$\times 10^{-11}$
\end{Verbatim}
```

produce:

x	$y = x^2$
0.000005	2.5×10^{-11}

Si usamos el ambiente **Verbatim** para escribir código Basic (como VBA, OOoBasic, etc.), los comentarios usan el apóstrofe: ' . Este apóstrofe se introduce en el ambiente Verbatim con el comando `\textquotesingle` del paquete **textcomp**.



Recordemos que frecuentemente el texto matemático va en el entorno $\$ \dots \$$. También hay otros entornos que no requieren estos símbolos de dólar. En este capítulo vamos a usar símbolos especiales (los símbolos son fuentes) que no están presentes en el conjunto de símbolos que por defecto carga **LaTeX**.

Por ejemplo, para escribir “ $\text{sen}(x) \in \mathbb{R}$ ” se requiere definir un comando `\sen` y tener acceso al paquete que permite definir el comando `\R` para obtener $\mathbb{R}$.

Para obtener el texto matemático de este capítulo se necesitan los símbolos que por defecto carga **LaTeX** y adicionalmente varios paquetes: `amsmath`, `amssymb`, `amsfonts`, `latexsym`, `cancel`. También necesitamos cinco comandos especiales para `sen`, `arcsen`, etc. La plantilla que sigue viene con todo lo que necesitamos para los ejemplos que siguen en todo este capítulo:

Plantilla para este capítulo

   [Descargar archivo](#)

```
\documentclass{article} %o report o book
\usepackage[total={18cm,21cm},centering]{geometry}
%Paquetes adicionales de símbolos matemáticos
\usepackage{amsmath,amssymb,amsfonts,latexsym,cancel}
\usepackage[utf8]{inputenc} %Acentos desde el teclado
\usepackage[T1]{fontenc}
\usepackage[spanish]{babel}
\usepackage{palatino,eulervm} % fuentes
% Comandos personales - especiales
\newcommand{\sen}{\mathop{\rm sen}\nolimits} %seno
\newcommand{\arcsen}{\mathop{\rm arcsen}\nolimits}
\newcommand{\arcsec}{\mathop{\rm arcsec}\nolimits}
\newcommand{\R}{\mathbb{R}}
\newcommand{\N}{\mathbb{N}}
\newcommand{\Z}{\mathbb{Z}}
\def\max{\mathop{\mbox{\rm máx}}\nolimits} % máximo
\def\min{\mathop{\mbox{\rm mín}}\nolimits} % mínimo
\begin{document}
  La fórmula  $\$ \text{sen}^2(x)=1-\cos^2(x) \$ \dots$ 
\end{document}
```

Aquí suponemos que se tiene una distribución **TeX completa y actualizada** (digamos **MiKTeX** o **TeX Live**).

4.1 Potencias, subíndices y superíndices

Expresión	Código	Expresión	Código
x^p	<code>x^p</code>	x^{n+1}	<code>x^{n+1}</code>
$(2^2)^n$	<code>(2^2)^n</code>	2^{2^n}	<code>2^{2^n}</code>
$\sin^2(x)$	<code>\sin^2(x)</code>	$x^{\sin(x)+\cos(x)}$	<code>x^{\sin(x)+\cos(x)}</code>
a_n	<code>a_n</code>	a_{n+1}	<code>a_{n+1}</code>
u_{N+1}	<code>u_{N+1}</code>	u_{N+1}	<code>u_{N+1}</code>
a_i^j	<code>a_i^j</code>	$\int_a^b f(x) dx$	<code>\int_a^b f(x) dx</code>
$\sum_{n=1}^N u_n$	<code>\sum_{n=1}^N u_n</code>	u_{ij}	<code>u_{ij}</code>

Se pueden ajustar los subíndice y los superíndices de la siguiente manera

Normal: `$S_{N_j}$` **produce:** S_{N_j}

Mejor: `$S_{\{N_j\}}$` **produce:** S_{N_j}

4.2 Tamaño natural

Como se ve en la tabla anterior, el texto matemático se ajusta al ancho del renglón. Para desplegarlo en tamaño natural se usa el comando `\displaystyle`. Si sólo se quiere que una parte del texto matemático salga en tamaño natural se escribe `\displaystyle{}` y entre las llaves se pone el texto.

Ejemplo 4.1

El código:

La suma parcial N -ésima S_N se define con la igualdad

$$S_N = \sum_{k=1}^N a_k$$

produce:

La suma parcial N -ésima S_N se define con la igualdad $S_N = \sum_{k=1}^N a_k$

4.3 Fracciones y expresiones de dos niveles

Para hacer fracciones se pueden utilizar los comandos: `\over`, `\frac{ }{ }` o `{ \atop }`, etc. Veamos también otras “fracciones” útiles.

Expresión	Código
$\frac{x+1}{x-1}$	<code>{x+1 \over x-1}</code>
$\frac{x+1}{x-1}$	<code>\frac{x+1}{x-1}</code>
$\frac{x+1}{x-1}$	<code>\dfrac{x+1}{x-1}</code>
$\frac{x+1}{x-1}$	<code>\tfrac{x+1}{x-1}</code>
$\frac{\frac{x+1}{3}}{x-1}$	<code>{{x+1 \over 3} \over x-1}</code>
$\left(1 + \frac{1}{x}\right)^{\frac{n+1}{n}}$	<code>\displaystyle{\left(1+ {1 \over x} \right)^{n+1 \over n}}</code>
$\left(1 + \frac{1}{x}\right)^{\frac{n+1}{n}}$	<code>\displaystyle \left(1+ \frac{1}{x} \right)^{\frac{n+1}{n}}</code>
$\left(1 + \frac{1}{x}\right)^{\frac{n+1}{n}}$	<code>\displaystyle{\left(1+ {1 \over x} \right)^{\displaystyle{n+1 \over n}}}</code>
$\frac{x+1}{x-1}$	<code>{x+1 \atop x-1}</code>
$\frac{x+1}{x-1}$	<code>{x+1 \above 2pt x-1}</code> (2pt es el grosor)
$\{ \frac{x+1}{x-1} \}$	<code>{x+1 \brace x-1}</code>
$\left[\frac{x+1}{x-1} \right]$	<code>{x+1 \brack x-1}</code>
$a \xrightarrow{f} b$	<code>\displaystyle{a \stackrel{f}{\rightarrow} b}</code>
$\lim_{x \rightarrow 0} f(x)$	<code>\displaystyle{\lim_{x \rightarrow 0} f(x)}</code>
$\binom{a}{b}$	<code>\displaystyle{a \choose b}</code>
$\sum_{\substack{0 \leq i \leq m \\ 0 \leq j \leq n}} a_i b_j$	<code>\displaystyle{\sum_{\substack{0 \leq i \leq m \\ 0 \leq j \leq n}} a_i b_j}</code>

$$\prod_{\substack{i=0 \\ i \neq k}}^n \frac{w_i}{(w_i - w_k)} \quad \backslash \text{prod_}\{\overset{i=0}{i \neq k}\}^n \backslash \text{frac}\{w_i\}\{(w_i - w_k)\}$$

Ejemplo 4.2

El código:

```


$$L_{n,k}(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$


```

produce:

$$L_{n,k}(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

Note el uso de `\overset{i=0}{i \neq k}` para producir: $i \neq k$

Integrales

Expresión	Código
$\int_C \mathbf{F} \cdot d\mathbf{r}$	<code>\displaystyle{\int_C \boldsymbol{F} \cdot d\mathbf{r}}</code>
$\oint_C \mathbf{F} \cdot d\mathbf{r}$	<code>\displaystyle{\oint_C \pmb{F} \cdot d\mathbf{r}}</code>
$\iint_D f(x,y) dA$	<code>\displaystyle{\iint_D f(x,y) \, dA}</code>
$\iiint_Q f(x,y,z) dA$	<code>\displaystyle{\iiint_Q f(x,y,z) \, dA}</code>

4.4 Raíces

Raíces cuadradas y raíces n -ésimas.

Expresión	Código
$\sqrt{x+1}$	<code>\sqrt{x+1}</code>
$\sqrt[n]{x+\sqrt{x}}$	<code>\displaystyle{ \sqrt[n]{x+\sqrt{x}} }</code>
$\sqrt[n]{x+\sqrt{x}}$	<code>\sqrt[n]{x+\sqrt{x}}</code>

4.5 Tres puntos consecutivos

Se usa un grupo de tres puntos para indicar la continuación de un patrón, se obtienen con los siguientes comandos.

... `\ldots` ... `\cdots`

⋮ `\vdots` ⋯ `\ddots`

4.6 Delimitadores

Para ajustar delimitadores al tamaño de una fórmula se usan los comandos `\left` ... `\right`. Se puede usar un punto para evitar abrir o cerrar con uno de los delimitadores.

Ejemplo 4.3

El código: `\displaystyle \left[ \frac{x+1}{(x-1)^2} \right]^n`

produce: $\left[\frac{x+1}{(x-1)^2} \right]^n$

El código:

`\int_{a}^{b} \left( x^2 + \sin(x) \right) dx = \left. \frac{x^3}{3} - \cos(x) \right|_{a}^{b}`

produce: $\int_a^b (x^2 + \sin(x)) dx = \frac{x^3}{3} - \cos(x) \Big|_a^b$

El código:

`\displaystyle \left\{ n \in \mathbb{N} \text{ at } r \neq 1 \right\}`

produce: $\left\{ n \in \mathbb{N} \text{ at } r \neq 1 \right\}$

Ejemplo 4.4**El código:**

```
\[f(x)=\left\{ \begin{array}{rcl}
& x^2+1 & \text{si } x\geq 0\\
& & \\
& \ln|x| & \text{si } x< 0\\
\end{array} \right. \]
```

% Observe el punto que cierra: `\left\{ ... \right.`

produce:

$$f(x) = \begin{cases} x^2 + 1 & \text{si } x \geq 0 \\ \ln|x| & \text{si } x < 0 \end{cases}$$

Delimitadores del paquete `amsmath`.

También se puede usar los delimitadores del paquete **`amsmath`**:

`\Biggl, \Biggr, \biggl, \biggr, \Bigl, \Bigr, \bigr, \bigr.`

En algunos casos estos delimitadores son más eficientes.

Ejemplo 4.5 (Comparación de delimitadores)**El código:**

```
$$\biggl[ \sum_j \Bigl| \sum_i x_{ij} \Bigr|^2 \biggr]^{1/2}$$
```

produce:

$$\left[\sum_j \left| \sum_i x_{ij} \right|^2 \right]^{1/2}$$

Comparar con

```
$$\left[ \sum_j \left| \sum_i x_{ij} \right|^2 \right]^{1/2}$$
```

$$\left[\sum_j \left| \sum_i x_{ij} \right|^2 \right]^{1/2}$$

4.7 LLaves y barras horizontales

Barras horizontales.

Las barras horizontales sobre el texto se pueden obtener con el comando `\overline{}`

62

Ejemplo 4.6

El código:

Leyes de DeMorgan:

```
\displaystyle{ \left\{
\overline{A \cup B} = \overline{A} \cap \overline{B}
\overline{A \cap B} = \overline{A} \cup \overline{B}
\right.}
```

produce:

Leyes de DeMorgan:
$$\begin{cases} \overline{A \cup B} = \overline{A} \cap \overline{B} \\ \overline{A \cap B} = \overline{A} \cup \overline{B} \end{cases}$$

LLaves

Las llaves se ponen con `\{` y `\}`. Esto se usa tanto en texto corriente como en modo matemático.

Ejemplo 4.7

El código:

```
$$\max_{x \in A} \{ f(x) \} > \min_{x \in A} \{ g(x) \} $$
```

produce:

$$\max_{x \in A} \{ f(x) \} > \min_{x \in A} \{ g(x) \}$$

Los comandos `\max` y `\min` aparecen con acento pues así los definimos en el preámbulo propuesto al inicio del capítulo.

LLaves horizontales

Se puede poner tanto una llave horizontal superior como una llave horizontal inferior y un texto arriba o abajo de la llave, se usa `\overbrace{ }^{ }`, `\underbrace{ }_{ }`, etc.

Ejemplo 4.8

El código:

```
\[ \overbrace{(x_i-1)}^{K_i}f(x)+\underbrace{(x_i-1)}_{K_i}g(x)
= K_i(f(x)+g(x)) \]
```

produce:

$$\overbrace{(x_i-1)}^{K_i}f(x) + \underbrace{(x_i-1)}_{K_i}g(x) = K_i(f(x) + g(x))$$

4.8 Acentos y “sombreros” en modo matemático

$\hat{i}$ `\hat{\imath}` $\acute{a}$ `\acute{a}`

$\bar{p}$ `\bar{p}` $\vec{p}$ `\vec{p}`

4.9 Vectores

Probablemente sea mejor usar un paquete para producir vectores: `\usepackage{esvect}`. Una vez cargado el paquete podemos poner

$\vec{v}$ `\vv{v}` $\vec{A}$ `\vv{A}` $\overrightarrow{v \times w}$ `\vv{v \times w}`

4.10 Negritas en modo matemático

En una fórmula matemática, el comando `\boldmath` solo aplica negrita a las *fuentes* de texto. Para poner en negrita los símbolos se debe usar `\boldsymbol` o `\pmb`.

También se podría usar el paquete `bm`. Con este paquete las negritas se ponen con el comando `\bm{...}`

Ejemplo 4.9 (Negrita en modo matemático)

$\cos(x + 2\pi) = \cos x$ `\pmb{\cos(x+2\pi)=\cos x}`

$\cos(x + 2\pi) = \cos x$ `\cos(x+\pmb{2\pi})=\cos x`

$\cos(x + 2\pi) = \cos x$ `\boldsymbol{\cos(x+2\pi)=\cos x}`

4.11 Espacio en modo matemático

LaTeX no deja espacio horizontal en modo matemático. Para dejar espacio horizontal en modo matemático se usan los comandos `\,`, `\;`, `\!`, `\:` tanto como `\hspace{}`

Ejemplo 4.10 (Espacio en modo matemático)

Normal: $n \in \mathbb{N}, x \in \mathbb{R}$ `n \in \mathbb{N}, x \in \mathbb{R}`

Mejor: $n \in \mathbb{N}, x \in \mathbb{R}$ `n \in \mathbb{N}, \; ; \; x \in \mathbb{R}`

Normal: $\int f(x)dx$ `\int f(x)dx`

Mejor: $\int f(x)dx$ `\displaystyle{\int} f(x)\,, dx`

4.12 Centrado

Para centrar una fórmula se usa `$$...$$` o todavía mejor: `\[ ... \]`.

En las fórmulas centradas no es en general necesario utilizar `\displaystyle` para producir texto en tamaño natural.

Ejemplo 4.11

El código:

`\[ ab \leq \left( \frac{a+b}{2} \right)^2 \]`

produce:

$$ab \leq \left(\frac{a+b}{2} \right)^2$$

4.13 Entorno equation. Contadores

LaTeX puede llevar un conteo automático de capítulos, secciones, etc. Podemos llevar también un conteo automático de teoremas, ecuaciones, etc. Por ejemplo con el entorno `equation` (un entorno para el modo matemático que no requiere `$`),

```
\begin{equation}...\end{equation}
```

podemos poner un número de ecuación a la ecuación actual. Podemos también cambiar ese número usando

```
\setcounter{equation}{k}.
```

El efecto de este comando es sumar ($-k$ resta) k unidades al número de ecuación actual

Ejemplo 4.12 (Entorno equation)

```
%estamos en ecuación 4.1 (cap 4)
\begin{equation}
\log_{2}(xy)=\log_{2}x + \log_{2}y
\end{equation}
```

$$\log_2(xy) = \log_2 x + \log_2 y \quad (4.1)$$

```

%sumamos 2 y pasamos a ecuación 4.3
\setcounter{equation}{2}
\begin{equation}
\log_{2}(a^b)=b\log_{2}a
\end{equation}
```

$$\log_2(a^b) = b \log_2 a \quad (4.3)$$

Una vez establecido el contador, se puede usar el ambiente `subequations` para lograr una ‘subnumeración’:

Ejemplo 4.13 (subequations)

```
\begin{subequations}

\begin{equation}
\log_{2}(xy)=\log_{2}x + \log_{2}y
\end{equation}

\begin{equation}
\log_{2}(a^b)=b\log_{2}a
\end{equation}

\end{subequations}
```

$$\log_2(xy) = \log_2 x + \log_2 y \quad (4.4a)$$

$$\log_2(a^b) = b \log_2 a \quad (4.4b)$$

4.14 Arreglos

Para editar una matriz se debe indicar:

- Los delimitadores, digamos: `\left[ \dots \right]`
- Inicio del “array” y el número y alineación de las columnas (centrado (c), alineado a la izquierda (l) o a la derecha (r)), digamos 3 columnas: `\begin{array}{lcr}`
- Los delimitadores de columnas, para 3 columnas: `& & &`
- “\\” indica el cambio de fila. Se puede usar `\\[xcm]` para crear espacio entre las filas.
- Final del “array”: `\end{array}`

Nota: En la sección 4.15 se muestran otras formas de realizar matrices.

Ejemplo 4.14 (Entorno array)

El código:

```
\[
A = \left( \begin{array}{lcr}
a      & & a+b      & & k-a      & \\
b      & & b        & & k-a-b    & \\
\vdots & & \vdots   & & \vdots   & \\
z      & & z + z   & & k-z      & \\
\end{array} \right)
\]
```

produce:

$$A = \begin{pmatrix} a & a+b & k-a \\ b & b & k-a-b \\ \vdots & \vdots & \vdots \\ z & z+z & k-z \end{pmatrix}$$

Ejemplo 4.15

Hagamos algunos cambios: Agreguemos dos columnas vacías, cambiemos el alineamiento y ajustamos el espacio entre la primera y la segunda fila:

El código:

```
\[
A = \left( \begin{array}{lcccl}
\frac{1}{a} & & a+b & & k-a & \\
b & & b & & k-a-b & \\
\vdots & & \vdots & & \vdots & \\
z & & z + z & & k-z & \\
\end{array} \right)
\]
```

produce:

$$A = \begin{pmatrix} \frac{1}{a} & a+b & k-a \\ b & b & k-a-b \\ \vdots & \vdots & \vdots \\ z & z+z & k-z \end{pmatrix}$$

Ejemplo 4.16 (Función a trozos).

El código:

```
\[
f(x)= \left\{ \begin{array}{lcl}
x^2 & \text{ si } & x < 0 \\
x-1 & \text{ si } & x > 0
\end{array} \right.
\]
```

produce:

$$f(x) = \begin{cases} x^2 & \text{ si } x < 0 \\ x-1 & \text{ si } x > 0 \end{cases}$$

N `\mbox` se utiliza para escribir texto corriente dentro del modo matemático.

Ejemplo 4.17

El código:

```
$$\nabla L(x^*)=0 \Rightarrow x^* \text{ es punto de silla } L$$
```

produce:

$$\nabla L(x^*) = 0 \Rightarrow x^* \text{ es punto de silla de la función lagrangiana } L$$

Ejemplo 4.18**El código:**

```


$$\frac{d}{dx}(\ln|x| + K_i) = \frac{\text{signo}(x)}{|x|} = \begin{cases} \frac{1}{x} & \text{si } x > 0 \\ \frac{1}{x} & \text{si } x < 0 \end{cases}$$


```

Es decir, las constantes K_1 y K_2 no tienen por qué ser iguales. Entonces lo correcto sería escribir

```


$$\int \frac{1}{x} dx = \ln|x| + \begin{cases} K_1 & \text{si } x > 0 \\ K_2 & \text{si } x < 0 \end{cases}$$


```

produce:

$$\frac{d}{dx}(\ln|x| + K_i) = \frac{\text{signo}(x)}{|x|} = \begin{cases} \frac{1}{x} & \text{si } x > 0 \\ \frac{1}{x} & \text{si } x < 0 \end{cases}$$

Es decir, las constantes K_1 y K_2 no tienen por qué ser iguales. Entonces lo correcto sería escribir

$$\int \frac{1}{x} dx = \ln|x| + \begin{cases} K_1 & \text{si } x > 0 \\ K_2 & \text{si } x < 0 \end{cases}$$

Ejemplo 4.19

A veces es conveniente anidar un array dentro de otro array,

El código:

```


$$\cos x = 0 \Leftrightarrow x = (2k+1)\pi; k \in \mathbb{Z}$$


$$\sin x = -1 \Leftrightarrow x = (4k+3)\pi; k \in \mathbb{Z}$$


$$\cos(2x) = \frac{1}{2} \Leftrightarrow \begin{cases} x = \frac{\pi}{6} + k\pi; z \in \mathbb{Z} \\ x = -\frac{\pi}{6} + k\pi; z \in \mathbb{Z} \end{cases}$$


```


El entorno `array` es útil y versátil. Si solo queremos trabajar con matrices podemos usar los entornos `smallmatrix`, `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` y `Vmatrix`. Estos entornos producen, respectivamente, $()$, $[]$, $\begin{pmatrix} \end{pmatrix}$ y $\| \|$.

El entorno `smallmatrix` produce arreglos ajustados (sin delimitadores), para ser usadas en el texto normal, e.g. $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

El código del párrafo anterior es,

```
El entorno ... e.g.  $\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}$ 
```

Ejemplo 4.20 (Entorno para matrices)

El código:

```
\begin{pmatrix}
1 & 0 & 0 & & \cdots & 0 \\
h_0 & 2(h_0+h_1) & h_1 & & \cdots & 0 \\
0 & h_1 & 2(h_1+h_2) & h_2 & \cdots & 0 \\
& \ddots & \ddots & \ddots & & \\
0 & 0 & \cdots & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\
0 & 0 & & & & \cdots & 1 \\
\end{pmatrix} \cdot \begin{pmatrix}
c_0 \\
c_1 \\
\vdots \\
c_{n-1} \\
c_n
\end{pmatrix}
```

produce:

$$\begin{pmatrix} 1 & 0 & 0 & & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & & \cdots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ 0 & \cdots & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} & \\ 0 & 0 & & & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}$$

4.16 Alineamiento

Ambiente “`eqnarray`”

Se puede alinear una serie de pasos (o ecuaciones), en un razonamiento usando

```
\begin{eqnarray*} . . . \end{eqnarray*}
```

este comando construye una matriz de 3 *columnas*.

Si se quiere que cada uno de los pasos aparezca numerado se utiliza

`\begin{eqnarray}. . . \end{eqnarray}`

Si se usa `\begin{eqnarray}. . . \end{eqnarray}`, se puede evitar numerar una ecuación poniendo `\nonumber` al final de la fila (antes de `\\`).

Ejemplo 4.21 (Arreglos sin numeración. Espacio)

El código:

```
De acuerdo al lema de Euclides tenemos que
\begin{eqnarray*} % Espacio entre filas aumentado \\[0.2cm]
\mbox{mcd}(a,b) &= & \mbox{mcd}(a-r_0q,r_0) \\[0.2cm]
&= & \mbox{mcd}(r_1,r_0) \\[0.2cm]
&= & \mbox{mcd}(r_1,r_0-r_1q_2) \\[0.2cm]
&= & \mbox{mcd}(r_1,r_2) \\[0.2cm]
&= & \mbox{mcd}(r_1-r_2q_2,r_2) \\[0.2cm]
\end{eqnarray*}
```

produce:

De acuerdo al lema de Euclides tenemos que

$$\begin{aligned}
 \text{mcd}(a,b) &= \text{mcd}(a-r_0q,r_0) \\
 &= \text{mcd}(r_1,r_0) \\
 &= \text{mcd}(r_1,r_0-r_1q_2) \\
 &= \text{mcd}(r_1,r_2) \\
 &= \text{mcd}(r_1-r_2q_2,r_2)
 \end{aligned}$$

Ejemplo 4.22 (Sin numeración)

El código:

```
% Sin numeración >>
\begin{eqnarray*}
y=\sqrt[n]{x} &\Longrightarrow & y^n=x \\
&\Longrightarrow & n\log y=\log x;\mbox{si};x,y>0 \\
&\Longrightarrow & \log \sqrt[n]{x}=\frac{1}{n}\log x \\
\end{eqnarray*}
```

produce:

$$\begin{aligned} y = \sqrt[n]{x} &\implies y^n = x \\ &\implies n \log y = \log x; \text{ si } x, y > 0 \\ &\implies \log \sqrt[n]{x} = \frac{1}{n} \log x \end{aligned}$$

Ejemplo 4.23 (Con numeración)

El código:

```
% Con numeración >>
\begin{eqnarray}
y=\sqrt[n]{x} & \&\Longrightarrow & \&y^n=x \\
& \&\Longrightarrow & \&n\log y=\log x;\;\mbox{si}\; x,y>0 \\
& \&\Longrightarrow & \&\log \sqrt[n]{x}=\{1 \over n\}\log x \\
\end{eqnarray}
```

produce:

$$\begin{aligned} y = \sqrt[n]{x} &\implies y^n = x & (4.5) \\ &\implies n \log y = \log x, \text{ si } x, y > 0 & (4.6) \\ &\implies \log \sqrt[n]{x} = \frac{1}{n} \log x & (4.7) \end{aligned}$$

Ejemplo 4.24 (Numeración selectiva)

El código:

```
% Numeración selectiva >>
\begin{eqnarray}
y=\sqrt[n]{x} & \&\Longrightarrow & \&y^n = x & \nonumber \\
& \&\Longrightarrow & \&n\log y=\log x, & \\
& & & & \;\mbox{si}\; x,y>0 \\
& \&\Longrightarrow & \&\log \sqrt[n]{x}=\{1 \over n\}\log x & \\
\end{eqnarray}
```

produce:

$$y = \sqrt[n]{x} \implies y^n = x$$

$$\implies n \log y = \log x, \text{ si } x, y > 0 \quad (4.8)$$

$$\implies \log \sqrt[n]{x} = \frac{1}{n} \log x \quad (4.9)$$

Entornos align y multline

El entorno ‘**eqnarray**’ no se recomienda porque tiene algunos inconvenientes: Produce un espaciado irregular en los signos de igualdad y no hacen ningún intento para evitar la sobreimpresión del cuerpo de la ecuación y número de la ecuación. Si usamos el paquete **amsmath** (como lo asumimos aquí) podemos acceder a los entornos “**align**” y “**multline**”

El entorno ‘align’ es similar a ‘eqnarray’ pero sin los problemas indicados, y también permite ecuaciones numeradas o sin numerar (usando **align***).

En el entorno **align** :

- **&=** establece una igualdad en una misma columna mientras que **&** establece un cambio de columna.
- El comando `\intertext{texto}` intercala texto entre filas mientras se mantiene las columnas alineadas.

Ejemplo 4.25

El código:

```
\begin{align*}
\intertext{Agrupamos,}
\frac{a+ay+ax+y}{x+y} &= \frac{ax+ay+x+y}{x+y} && \text{\mbox{Agrupar}} \\
\intertext{sacamos el factor común,}
&= \frac{a(x+y)+x+y}{x+y} && \text{\mbox{Factor común}} \\
&= \frac{(x+y)(a+1)}{x+y} && \text{\mbox{Simplificar}} \\
&= a+1
\end{align*}
```

produce:

Agrupamos,

$$\frac{a + ay + ax + y}{x + y} = \frac{ax + ay + x + y}{x + y} \quad \text{Agrupar}$$

sacamos el factor común,

$$\begin{aligned} &= \frac{a(x+y)+x+y}{x+y} && \text{Factor común} \\ &= \frac{(x+y)(a+1)}{x+y} && \text{Simplificar} \\ &= a+1 \end{aligned}$$

Ejemplo 4.26 (Partir filas)

El código:

```
\begin{align*}
a &= b + c - d \\
&\quad + e - f \\
&= m
\end{align*}
```

produce:

$$\begin{aligned} a &= b + c - d \\ &\quad + e - f \\ &= m \end{aligned}$$

El código:

```
\begin{multline*}
a+b+c+d+e+f+t+x+y \\
+m+n+r+t+y
\end{multline*}
```

produce:

$$\begin{aligned} a+b+c+d+e+f+t+x+y \\ +m+n+r+t+y \end{aligned}$$

4.17 Tablas de símbolos matemáticos frecuentes

Letras griegas

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

Operadores binarios

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

Relaciones

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

Negación de relaciones

En general, cualquier negación se puede hacer anteponiendo la instrucción `\not` a cualquier relación anterior, algunos ejemplos se muestran en la tabla siguiente.

$\nless$ <code>\not<</code>	$\nsucceq$ <code>\not\succeq</code>
$\nleq$ <code>\not\leq</code>	$\nsupset$ <code>\not\supset</code>
$\nprec$ <code>\not\prec</code>	$\nsupseteq$ <code>\not\supseteq</code>
$\npreceq$ <code>\not\preceq</code>	$\nsqsupseteq$ <code>\not\sqsupseteq</code>
$\nsubset$ <code>\not\subset</code>	$\neq$ <code>\not=</code>
$\nsubseteq$ <code>\not\subseteq</code>	$\nequiv$ <code>\not\equiv</code>
$\nsqsubseteq$ <code>\not\sqsubseteq</code>	$\nsim$ <code>\not\sim</code>
$\n>$ <code>\not></code>	$\nsimeq$ <code>\not\simeq</code>
$\ngeq$ <code>\not\geq</code>	$\napprox$ <code>\not\approx</code>
$\nsucc$ <code>\not\succ</code>	$\ncong$ <code>\not\cong</code>

Otros símbolos

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

Especiales

$\tilde{A}$ `\widetilde{A}`
 $\hat{A}$ `\widehat{A}`

$\overrightarrow{v}$ `\overrightarrow{v}`

Símbolos del paquete amssymb

El paquete amssymb se carga si usamos el preámbulo propuesto al inicio del capítulo.

$\mathbb{R}$	<code>\mathbb{R}</code>	$\gtrdot$	<code>\gtrdot</code>	$\lesseqgtr$	<code>\lesseqgtr</code>	$\supset$	<code>\Supset</code>
$\mathbb{Q}$	<code>\mathbb{Q}</code>	$\gtrless$	<code>\gtrless</code>	$\doteqdot$	<code>\doteqdot</code>	$\sqsubset$	<code>\sqsubset</code>
$\mathbb{Z}$	<code>\mathbb{Z}</code>	$\eqslantless$	<code>\eqslantless</code>	$\risingdotseq$	<code>\risingdotseq</code>	$\sqsupset$	<code>\sqsupset</code>
$\mathbb{I}$	<code>\mathbb{I}</code>	$\lesssim$	<code>\lesssim</code>	$\fallingdotseq$	<code>\fallingdotseq</code>	$\succcurlyeq$	<code>\succcurlyeq</code>
$\therefore$	<code>\therefore</code>	$\lessapprox$	<code>\lessapprox</code>	$\circeq$	<code>\circeq</code>	$\preccurlyeq$	<code>\preccurlyeq</code>
$\because$	<code>\because</code>	$\approxeq$	<code>\approxeq</code>	$\triangleq$	<code>\triangleq</code>	$\curlyeqprec$	<code>\curlyeqprec</code>
$\leqq$	<code>\leqq</code>	$\lessdot$	<code>\lessdot</code>	$\thicksim$	<code>\thicksim</code>	$\curlyeqsucc$	<code>\curlyeqsucc</code>
$\geqq$	<code>\geqq</code>	$\gtreqless$	<code>\gtreqless</code>	$\thickapprox$	<code>\thickapprox</code>	$\prec$	<code>\prec</code>
$\leqslant$	<code>\leqslant</code>	$\gtreqgless$	<code>\gtreqgless</code>	$\backsim$	<code>\backsim</code>	$\succsim$	<code>\succsim</code>
$\geqslant$	<code>\geqslant</code>	$\lll$	<code>\lll</code>	$\backsimeq$	<code>\backsimeq</code>	$\precapprox$	<code>\precapprox</code>
$\eqslantgtr$	<code>\eqslantgtr</code>	$\ggg$	<code>\ggg</code>	$\subseteq$	<code>\subseteq</code>	$\succapprox$	<code>\succapprox</code>
$\gtrsim$	<code>\gtrsim</code>	$\lessgtr$	<code>\lessgtr</code>	$\supseteq$	<code>\supseteq</code>	$\vartriangleright$	<code>\vartriangleright</code>
$\gtrapprox$	<code>\gtrapprox</code>	$\lesseqgtr$	<code>\lesseqgtr</code>	$\Subset$	<code>\Subset</code>	$\trianglerighteq$	<code>\trianglerighteq</code>

Símbolos adicionales

Se puede encontrar una gran cantidad de símbolos adicionales (cerca de 164 páginas A4 indicando sus correspondientes paquetes) en la dirección:

www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf

4.18 Cómo hacer nuevos Comandos.

Podemos abreviar el código de los comandos creando comandos propios. Para esto usamos

- `\newcommand{\nuevo_nombre}{\comando_original}`
- `\newcommand{\nombre}[n]{\f{#1}...\h{#n}}`. n es el número de parámetros. Cada parámetro es recibido por un comando existente.

Las definiciones de los nuevos comandos se ponen en el *preámbulo* (para comodidad de otros usuarios).

Una práctica muy recomendada es hacerse un archivo aparte con estas definiciones, este archivo debe ir **sin** preámbulo **ni** `\begin{document}...\end{document}`. Si el archivo es "cmds.tex", éste se invoca en el preámbulo con `\input{cmds.tex}`. Este archivo puede estar en el directorio de trabajo preferiblemente.

Vamos a ver algunos ejemplos.

- Abreviaciones para `\begin{center}...\end{center}`

```
\newcommand{\bc}{\begin{center}}
\newcommand{\ec}{\end{center}}
```

- Abreviación para `\displaystyle`

```
\newcommand{\ds}[1] {\displaystyle{#1}} %un parámetro
```

- Abreviación para `\sii`: $\Longleftrightarrow$

```
\newcommand{\sii}{\Longlefttrightarrow}
```

- Comando para fracciones 'grandes' $\frac{a}{b}$

Este comando usa un comando definido anteriormente, `\ds`

```
\newcommand{\gfrac}[2] \ds{\frac{#1}{#2}}
```

Comentario: En realidad ya tenemos el comando `\dfrac{ }{ }` que hace los mismo que nuestro `\gfrac`.

- Comando personalizado para la proyección ortogonal de $\vec{v}$ sobre $\vec{w}$ usando el paquete **esvect**; $\text{proy}_{\vec{w}} \vec{v} = \text{proy}_{\vec{w}} \vec{v}$.

```
\newcommand{\proy}[2] \newcommand{\proy}[2]{\displaystyle\mbox{proy}_{\vec{\blue{#2}}}\vec{\blue{#1}}}}
```

- Un comando para hacer una arreglo 3×3 :

```
\newcommand{\mllave}[3]{\left\{\begin{array}{lcl}
#1 \\\
#2 \\\
#3 \\\
\end{array}\right.}
```

Este comando se podría usar así:

```
$f(x) = \mllave{x^2 & \mbox{si} & x>0 \\\
{2x+1 & \mbox{si} & x<0 \\\
{0 & \mbox{si} & x=0 }$
```

para producir

$$f(x) = \begin{cases} x^2 & \text{si } x > 0 \\ 2x + 1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \end{cases}$$

- Un comando para hacer referencias (usa el paquete **hyperref**) dentro del documento. Por ejemplo, hay una sección con etiqueta `\label{comandoconopciones}`, podemos hacer referencia a esa sección así: La sección 4.18 ..., . Esta liga se produce con:

La `\wref{sección}{comandoconopciones}` ...

El comando sería,

```
\newcommand{\wref}[2]{\hyperref[#2]{#1 \ref*{#2}}}
```

- Colores:

```
\newcommand{\colr}[1]{\red #1}
\newcommand{\colb}[1]{\blue #1}
```

Ejemplo 4.27 (Usando comandos personalizados).

Con los comandos definidos más arriba podemos abreviar el código.

El código:

```
%\usepackage{esvect,pstricks} %pstricks para color \blue, \red, ...
$\proy{v}{\blue w}\;=\;\dfrac{\vv{v}}{\|\vv{w}\|^2}\cdot\blue{\vv{w}}\;{\$}
|\blue{\vv{w}}|^2\;,\;\blue{w}$
```

produce:

$$\text{proy } \frac{\vec{v}}{\vec{w}} = \frac{\vec{v} \cdot \vec{w}}{\|\vec{w}\|^2} w$$

- N** Si se usa color, se debe usar llaves para delimitar el entorno donde este color tiene efecto, por ejemplo `{\blue ...txt...}` sólo afectaría '`...txt...`'

Ejemplo 4.28 (Comando para “minipage” y para “límite”).

- ```
\newcommand{\mpage}[2] {
 \begin{minipage}[b]{\textwidth}
 \begin{minipage}[b]{0.5 \textwidth}
 #1
 \end{minipage} \hfill \begin{minipage}[b]{0.5 \textwidth}
 #2
 \end{minipage}
\end{minipage} }
```

- `\newcommand{\limite}[2] { \lim_{#1} \rightarrow #2}`

Así, el código::

```
\mpage{
 \centering
 \includegraphics{images/ML_fig13.pdf}
}{
 \[\limite{n}{\infty} {\arctan (n) } \]
}
```

**produce:** (compilando con PDFLaTeX):



$$\lim_{n \rightarrow \infty} \arctan(n)$$



En los entornos minipage, parbox, etc. se puede usar el comando `\raisebox` o el paquete `adjustbox` para alinear, tal y como hicimos en los ejemplos 3.9, 3.10 y 3.12

Otros ejemplos son

```
%Texto con fuente helvetica, uso: \fhv{texto}
\newcommand{\fhv}[1]{\fontfamily{hv}\fontsize{9}{1}\selectfont{#1}}

%Entorno enumerate, uso \be \item... \item... \ee
\newcommand{\be}{\begin{enumerate}}
\newcommand{\ee}{\end{enumerate}}
```

## Comandos con opciones

Podemos agregar opciones a nuestros comandos dejando algunos valores por default. Esto lo podemos hacer con el paquete `xargs`: Ponemos `\usepackage{xargs}` en el *preámbulo*.

El código para un comando con  $n$  parámetros sería algo como,

```
newcommandx\nombre[n][opcion1 = o1, opcion2 = o2,...]{código latex}
```

Las opciones, si hubiera, se refieren al valor default de algunos o todos los parámetros.

Un ejemplo clásico es el de crear un comando para abreviar una sucesión:  $x_0, x_1, \dots, x_n$ . En este caso, es deseable que podamos tener un comando flexible que nos permita iniciar con subíndice 0 o con subíndice 1 y terminar con subíndice  $n$  o con subíndice  $k$  y cambiar  $x_i$  por  $u_i$ , etc.

La sucesión por defecto será  $x_0, x_1, \dots, x_n$ .

**Ejemplo 4.29** (Comando para sucesiones con opciones).

El nuevo comando `\coord{}{}{}` se define así,

```
\newcommand\coord[3][1=0, 3=n]{\left(#2_{#1},\ldots,#2_{#3}\right)}
```

`\coord` recibe tres argumentos, el primero y el tercero son opcionales y tienen valor default 0 y  $n$  respectivamente, por eso aparece `[1=0, 3=n]`. El parámetro #2 permite elegir  $x$ ,  $u$ , etc.

**El código:**

```
\coord{x}$
\coord[0]{y}$
\coord{z}[m]$
\coord[0]{t}[m]$
```

**produce:**

```
(x1,...,xn)
(y0,...,yn)
(z1,...,zm)
(t0,...,tm)
```

**Ejemplo 4.30**

Podemos hacer más flexible nuestro comando ‘`\mpage`’ (en el ejemplo 4.28) de la siguiente manera

```
%\usepackage{xargs}
\newcommand\mpage[4][1=0.45,2=0.45]{%Default=45% ancho página
\begin{minipage}{\textwidth}
 \begin{minipage}[b]{#1\textwidth}
 #3
 \end{minipage} \hfill \begin{minipage}[b]{#2\textwidth}
 #4
 \end{minipage}
\end{minipage}
}%fin
```

de tal manera que lo podemos usar como `\mpage[0.7][0.2]{...}{...}` para columnas con tamaños  $0.7\text{width}$  la primera y  $0.2\text{width}$  la segunda. También se puede usar como `\mpage{...}{...}` usando los valores por defecto (ambas columnas de igual tamaño).

**“def” y argumentos separados por comas**

Se pueden implementar comandos con argumentos separados por comas con `\def`. `\def` es una primitiva **TeX** mientras que `\newcommand` es una extensión **LaTeX** de `\def`.

A veces es mejor usar `\newcommand` en vez de `\def`. Los beneficios más evidentes de `\newcommand` sobre `\def` son:

- `\newcommand` verifica si el comando ya existe
- `\newcommand` permite definir un argumento opcional

`\def` no verifica de manera automática si el comando ya existe, si el comando ha sido definido, lo sobrescribe... esto podría ser bueno en algunos casos.

Veamos un ejemplo de cómo podríamos usar `\def`. Digamos que queremos escribir un comando que imprima la norma de un vector de tal manera que los argumentos se escriban *separados por comas*. Esto es fácil con `\def`. Podríamos escribir en el *preámbulo*,

```
\def\minorma(#1,#2,#3){\sqrt{#1^2+#2^2+#3^2}}
```

Así, podríamos escribir `\minorma(v_1,v_2,v_3)` para obtener  $\sqrt{v_1^2 + v_2^2 + v_3^2}$

Se puede redefinir el comando, de tal manera que usemos *llaves* en vez de paréntesis:

```
\def\lanorma#1{\minormaprevia(#1)}
\def\minormaprevia(#1,#2,#3){\sqrt{#1^2+#2^2+#3^2}}
```

Así, podríamos escribir (ahora sí, con llaves) `\lanorma{v_1,v_2,v_3}` para obtener  $\sqrt{v_1^2 + v_2^2 + v_3^2}$





Las tablas 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 agregar líneas verticales se ponen marcas como `|` o `||` en la parte que corresponde al alineamiento de columnas.
- Para agregar líneas horizontales, al final de cada fila se especifica
  - `\hline`: línea horizontal tan larga como la tabla
  - `\cline{i-j}`: línea de columna  $i$  a columna  $j$
  - El grosor y el color de las líneas horizontales se puede controlar con el paquete `booktabs`, por ejemplo.

#### Ejemplo 5.1 (Usando “tabular”)

Ambiente **tabular** con tres columnas.

El código:

produce:

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

p & q & $p \rightarrow q$ \\ \hline
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 1 & 1 \\ \hline
\end{tabular}


```

| $p$ | $q$ | $p \rightarrow q$ |
|-----|-----|-------------------|
| 0   | 0   | 1                 |
| 0   | 1   | 1                 |
| 1   | 0   | 0                 |
| 1   | 1   | 1                 |

#### Usando el paquete “booktabs”

Para tener opciones adicionales en el entorno **tabular** se puede utilizar el paquete **booktabs**. Por ejemplo, podemos controlar grosor, color y espacio adicional en las filas. Se escribe en el *preámbulo*:

```
\usepackage{booktabs}
\usepackage[table]{xcolor}
```

De esta forma siguen funcionando los comandos usuales de **tabular**, es decir, el ejemplo anterior se puede escribir de la misma forma y se verá exactamente igual, pero se agregan algunos comandos extras: `\toprule`, `\midrule`, `\bottomrule`, `\specialrule`, etc.

### Ejemplo 5.2 (Usando “booktabs”)

Usando los comandos adicionales de booktabs.

El código:

```
\begin{tabular}{llr} \toprule
Nombre & Apellido & Edad \\ \midrule
Julio & Cortés & 11 \\
Mario & Ramírez & 12 \\ \bottomrule
\end{tabular}
```

produce:

| Nombre | Apellido | Edad |
|--------|----------|------|
| Julio  | Cortés   | 11   |
| Mario  | Ramírez  | 12   |

### Ejemplo 5.3 (Color y grosor)

Usamos en el preámbulo

```
\usepackage{booktabs}
\usepackage[table]{xcolor}
```

Con estos dos paquetes, podemos controlar el color, grosor y espacio arriba y abajo de una fila, usando los comandos

```
\specialrule{grosor}{espacio arriba}{espacio abajo}
```

```
\arrayrulecolor{color}
```

En la tabla que sigue, le hacemos espacio arriba y abajo a la fracción  $\frac{1}{x^2}$ . El grosor default es 0.4pt

El código:

```
\begin{tabular}{rrcc} \toprule
A & Va & Vb & R \\ \midrule
A & 1 & 2 & 3 \\ \arrayrulecolor{blue!50}
\specialrule{2.5pt}{2pt}{5pt}
D & x & $\frac{1}{x^2}$ & 3 \\ \specialrule{0.4pt}{5pt}{2pt}
E & 1 & 2 & 3 \\ \arrayrulecolor{black} \bottomrule
\end{tabular}
```

produce:

| A | Va  | Vb              | R |
|---|-----|-----------------|---|
| A | 1   | 2               | 3 |
| D | $x$ | $\frac{1}{x^2}$ | 3 |
| E | 1   | 2               | 3 |

## 5.1 Tablas de más de una página

Cuando se hace una tabla que es demasiado larga para que quepa en una sólo página se debe usar el ambiente `longtable` en vez de `tabular`, éste se encarga de dividir la tabla entre páginas, para esto se debe agregar el paquete `longtable` en el *preámbulo*: `\usepackage{longtable}`

Ejemplo 5.4 (Tablas largas)

El código:

```
%\usepackage{longtable,multirow,booktabs}
\begin{longtable}[c]{llr} % opción [c] = centrada
\caption[Tabla grande]{Tabla grande} \label{ej1:longtable} \\ \toprule
Nombre & Apellido & Edad \\ \midrule
Julio & Cortés & 11 \\
Marco & Villalta & 13 \\
 & : & \\
Mario & Ramírez & 12 \\ \bottomrule
\end{longtable}
```

produce:

Tabla 5.1. Tabla grande

| Nombre    | Apellido   | Edad |
|-----------|------------|------|
| Julio     | Cortés     | 11   |
| Marco     | Villalta   | 13   |
| Alejandro | González   | 15   |
| Alberto   | Montalbán  | 13   |
| Pedro     | Núñez      | 12   |
| Carlos    | Montero    | 10   |
| Daniel    | Hernández  | 11   |
| Diego     | Villalobos | 14   |
| Doryan    | Loría      | 15   |
| Eduardo   | Torres     | 17   |
| Marco     | Villalta   | 13   |
| Alejandro | González   | 15   |
| Alberto   | Montalbán  | 13   |
| Pedro     | Núñez      | 12   |
| Carlos    | Montero    | 10   |
| Marco     | Villalta   | 13   |
| Alejandro | González   | 15   |
| Alberto   | Montalbán  | 13   |
| Pedro     | Núñez      | 12   |
| Carlos    | Montero    | 10   |
| Daniel    | Hernández  | 11   |

|           |            |    |
|-----------|------------|----|
| Diego     | Villalobos | 14 |
| Doryan    | Loría      | 15 |
| Daniel    | Hernández  | 11 |
| Diego     | Villalobos | 14 |
| Doryan    | Loría      | 15 |
| Marco     | Villalta   | 13 |
| Alejandro | González   | 15 |
| Alberto   | Montalbán  | 13 |
| Pedro     | Núñez      | 12 |
| Carlos    | Montero    | 10 |
| Daniel    | Hernández  | 11 |
| Diego     | Villalobos | 14 |
| Doryan    | Loría      | 15 |
| Mario     | Ramírez    | 12 |

**N** La opción **caption** de este entorno puede entrar en conflictos con otros paquetes como el paquete (del mismo nombre) **caption** (que veremos más adelante). Si usa este paquete, debería usar el comando **captionof{ }{ }** de este último paquete en vez del comando **caption** del paquete **longtable**.

El paquete **longtable** tiene comandos para definir el encabezado que se debe poner al continuar la tabla en la siguiente página y el pie de la tabla al terminar en la anterior.

- `\endfirsthead`: Define el encabezado principal de la tabla.
- `\endhead`: Define el encabezado que tendrá la tabla en las siguientes páginas.
- `\endfoot`: Define el pie que tendrá la tabla en todas las páginas excepto la última.
- `\endlastfoot`: Define el pie que tendrá la tabla en la última página.

   [Descargar archivo](#)

#### Ejemplo 5.5 (Tablas largas)

En este ejemplo usamos el paquete **longtable** junto con el paquete **booktabs**.

El código:

```
% En el preámbulo
%\usepackage{longtable,booktabs,multirow}
\begin{ longtable }[c]{llr}
\caption[Tabla de varias páginas con encabezado y pie]{Tabla de varias
páginas con encabezado y pie.}\

\toprule Nombre & Apellido & Edad \\\midrule
\endfirsthead
```

```
\multicolumn{3}{c} {{\tablename\ \thetable{} -- continua de la página
anterior}} \\
\toprule Nombre & Apellido & Edad \\ \midrule
\endhead

\midrule
\multicolumn{3}{r}{{\magenta Continúa en la siguiente página...}}\\
\midrule
\endfoot

\bottomrule
\endlasthead

Julio & Cortés & 11 \\
Marco & Villalta & 13 \\
& \vdots & \\
Mario & Ramírez & 12
\end{ longtable }
```

produce:

Tabla 5.3. Tabla de varias páginas con encabezado y pie

| Nombre                             | Apellido   | Edad |
|------------------------------------|------------|------|
| Julio                              | Cortés     | 11   |
| Marco                              | Villalta   | 13   |
| Alejandro                          | González   | 15   |
| Alberto                            | Montalbán  | 13   |
| Pedro                              | Núñez      | 12   |
| Carlos                             | Montero    | 10   |
| Daniel                             | Hernández  | 11   |
| Diego                              | Villalobos | 14   |
| Doryan                             | Loría      | 15   |
| Eduardo                            | Torres     | 17   |
| Fabián                             | Robles     | 12   |
| Fabrizio                           | Castro     | 13   |
| Gabriel                            | Gutiérrez  | 11   |
| Julio                              | Cortés     | 11   |
| Marco                              | Villalta   | 13   |
| Alejandro                          | González   | 15   |
| Alberto                            | Montalbán  | 13   |
| Pedro                              | Núñez      | 12   |
| Carlos                             | Montero    | 10   |
| Daniel                             | Hernández  | 11   |
| Diego                              | Villalobos | 14   |
| Doryan                             | Loría      | 15   |
| Diego                              | Villalobos | 14   |
| Continúa en la siguiente página... |            |      |

Tabla 5.4 – continua de la página anterior

| Nombre    | Apellido   | Edad |
|-----------|------------|------|
| Doryan    | Loría      | 15   |
| Eduardo   | Torres     | 17   |
| Fabián    | Robles     | 12   |
| Fabricio  | Castro     | 13   |
| Gabriel   | Gutiérrez  | 11   |
| Julio     | Cortés     | 11   |
| Marco     | Villalta   | 13   |
| Alejandro | González   | 15   |
| Alberto   | Montalbán  | 13   |
| Pedro     | Núñez      | 12   |
| Carlos    | Montero    | 10   |
| Daniel    | Hernández  | 11   |
| Diego     | Villalobos | 14   |
| Doryan    | Loría      | 15   |
| Eduardo   | Torres     | 17   |
| Fabián    | Robles     | 12   |
| Fabricio  | Castro     | 13   |
| Gabriel   | Gutiérrez  | 11   |
| Henry     | Solano     | 13   |
| Javier    | Ruiz       | 11   |
| Mauricio  | Torrealba  | 13   |
| Henry     | Solano     | 13   |
| Javier    | Ruiz       | 11   |
| Mauricio  | Torrealba  | 13   |
| Mario     | Ramírez    | 12   |

## 5.2 Objetos flotantes: Entornos figure y table

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 (y a veces complica) 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!]
....
 \caption{...}\label{figure:nombre}
\end{figure}

\begin{table}[h!]
....
```

```
\caption{...}\label{table:nombre}
\end{table}
```

- `[h!]` le indica a **LaTeX** que queremos la figura o la tabla, exactamente en ese lugar (h=here, esto no es tan exacto, ya que **LaTeX** en realidad lo acomoda lo más cerca posible de ese lugar). Otras opciones son `[t]=top`, `[b]=bottom`.
- `\caption{ texto}` es la etiqueta de cada objeto (numerándolo automáticamente). Se puede omitir.
- `\label` ("etiqueta") 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.

#### Ejemplo 5.6 (Entorno table)

El código:

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

p & q & $p \rightarrow q$

0 & 0 & 1

0 & 1 & 1

1 & 0 & 0

1 & 1 & 1

\end{tabular}
\caption{Tabla de verdad para

$p \rightarrow q$

\end{table}
```

produce:

| $p$ | $q$ | $p \rightarrow q$ |
|-----|-----|-------------------|
| 0   | 0   | 1                 |
| 0   | 1   | 1                 |
| 1   | 0   | 0                 |
| 1   | 1   | 1                 |

Tabla 5.5. Tabla de verdad para  $p \rightarrow q$

**Nota:** Si va a poner una figura o una tabla en el ambiente **minipage**, debería usar el siguiente formato (recuerde los ejemplos 3.9, 3.10 y 3.12).

```
\begin{figure}[h!]
\begin{minipage}
...
\end{minipage}
\end{figure}
```



## 5.3 Fuentes en tabular.

A veces es conveniente cambiar la fuente en una ambiente tabular, por ejemplo si tenemos una tabla de números sería bueno cambiar a una fuente **cmr10** para que todo quede en modo matemático sin tener que hacer esto número por número,

### Ejemplo 5.7 (Cambio de fuente)

El código:

```
\fontfamily{cmr10}\selectfont{%Fuente cmr10
\begin{tabular}{c|c}
 x & x^2+3 \\ \hline
 2 & 7
 4 & 19
\end{tabular}
}%fontfamily
```

produce:

| $x$ | $x^2 + 3$ |
|-----|-----------|
| 2   | 7         |
| 4   | 19        |

## 5.4 Color en tablas.

Para poner color en las filas o las columnas de una tabla podríamos usar el paquete **xcolor** agregamos al *preámbulo*

`\usepackage[x11names,table]{xcolor}` o `\usepackage[x11names,table]{book}`

La opción **x11names** habilita un conjunto de colores que podemos llamar por su nombre. La lista completa la puede obtener en **xcolor.pdf**. Por ejemplo,

`\color{RoyalBlue1} Texto`      Texto

`\color{LightSteelBlue1} Texto`      Texto

### Conflictos.

Si hay conflictos (“clash”) con otros paquetes, como por ejemplo con **Beamer**, podríamos poner la opción al inicio, en la clase de documento,

```
\documentclass[xcolor=table, x11names]{beamer} % obook
```

Para colorear una fila solo se agrega, al inicio de la fila,

```
\rowcolor{color}
```

Para colorear de *manera alternada* se agrega

```
\rowcolors[nfila]{color fila-impar}{color fila-par}
```

antes del inicio de la tabla.

Aquí, **nfila** es el número de fila de la primera fila en ser coloreada. Los colores de fila par e impar se pueden dejar en blanco (no se pondrá color en esa fila).

Los comandos **\columncolor** y **\cellcolor** se usan para colorear las columnas y celdas, respectivamente.

### Ejemplo 5.8

En este ejemplo, la fila inicial se colorea con el color **LightBlue2** de la opción **xllnames** del paquete **xcolor**. Esto se hace agregando **\rowcolor{LightBlue2}** al inicio de la primera fila.

También se colorean con un gris degradado a un 20% (**gray!20**) las filas pares y las impares se dejan con fondo blanco. Esto se hace agregando, antes del inicio de la tabla, la instrucción **{\rowcolors{1}{}{gray!20}}**.

El código:

```
\begin{table}[h!]
\centering
\rowcolors{1}{}{gray!20}
\begin{tabular}{ll}
\rowcolor{LightBlue2} x_{n+1} & & $|x_{n+1}-x_n|$\ \hline
1.20499955540054 & & 0.295000445\ \hline
1.17678931926590 & & 0.028210236\ \hline
1.17650193990183 & & 3.004$\times 10^{-8}$\ \hline
1.17650193990183 & & 4.440$\times 10^{-16}$\ \hline
\end{tabular}
\caption{Iteración de Newton para $x^2-\cos(x)-1=0$ con $x_0=1.5.$}
\end{table}
```

produce:

| $x_{n+1}$        | $ x_{n+1} - x_n $       |
|------------------|-------------------------|
| 1.20499955540054 | 0.295000445             |
| 1.17678931926590 | 0.028210236             |
| 1.17650193990183 | $3.004 \times 10^{-8}$  |
| 1.17650193990183 | $4.440 \times 10^{-16}$ |

**Tabla 5.6.** Iteración de Newton para  $x^2 - \cos(x) - 1 = 0$  con  $x_0 = 1.5$ .

**Ejemplo 5.9**

En este ejemplo se colorean con un 20% gris dos celdas: Simplemente agregamos `\cellcolor[gray]{0.80}` en las celdas que queremos.

El código:

```
\begin{tabular}{ll}
\rowcolor{LightBlue} x_{n+1} & $|x_{n+1}-x_n|$\ \hline
\cellcolor[gray]{0.80} 1.20499955540054 & 0.295000445\ \\
1.17678931926590 & 0.028210236\ \\
1.17650196994274 & 0.000287349\ \\
1.17650193990183 & 3.004$\times 10^{-8}$\ \\
\cellcolor[gray]{0.80} 1.17650193990183 & 4.440$\times 10^{-16}$\ \hline
\end{tabular}
```

produce:

| $x_{n+1}$        | $ x_{n+1} - x_n $       |
|------------------|-------------------------|
| 1.20499955540054 | 0.295000445             |
| 1.17678931926590 | 0.028210236             |
| 1.17650196994274 | 0.000287349             |
| 1.17650193990183 | $3.004 \times 10^{-8}$  |
| 1.17650193990183 | $4.440 \times 10^{-16}$ |

## 5.5 Rotación de texto en celdas.

Para rotar una tabla completa o simplemente el texto en las celdas. se usa el entorno

```
\begin{sideways}...\end{sideways}
```

aplicado directamente a la tabla o a la(s) celda(s). Necesitamos agregar en el preámbulo

```
\usepackage{rotating}
```

**Ejemplo 5.10**

El código:

```
\begin{sideways}
\begin{tabular}{lc}
x_{n+1} & $|x_{n+1}-x_n|$\ \hline
\cellcolor[gray]{0.80} 1.17 & 3.$\times 10^{-8}$\ \\
1.17 & 4.$\times 10^{-16}$\ \hline
\end{tabular}
\end{sideways}
```

produce:

| $x_{n+1}$ | $ x_{n+1} - x_n $   |
|-----------|---------------------|
| 1.17      | $3 \times 10^{-8}$  |
| 1.173     | $4 \times 10^{-16}$ |

El código:

```
\begin{tabular}{lc}
 x_{n+1} & $|x_{n+1} - x_n|$ \\
\cellcolor{gray}{0.80} 1.17 & 3×10^{-8} \\
1.173 & 4×10^{-16}
\end{tabular}
```

produce:

| $x_{n+1}$ | $ x_{n+1} - x_n $   |
|-----------|---------------------|
| 1.17      | $3 \times 10^{-8}$  |
| 1.173     | $4 \times 10^{-16}$ |

### Expresiones @{}.

En un ambiente tabular el separador de columnas se puede cambiar con una instrucción del tipo @{}txt{}. Este comando elimina la separación automática entre columnas y la reemplaza con el texto txt.

#### Ejemplo 5.11

En el código que sigue, r@{.}l sustituye la columna central por un punto,

El código:

produce:

```
\begin{tabular}{r@{.}l}
3 & 14159 \\
2 & 7182818 \\
0 & 577216
\end{tabular}
```

## 5.6 Unir celdas.

A veces es conveniente unir dos o más celdas para poner una leyenda un poco extensa. Para hacer esto usamos

`\multicolumn{columnas}{Alin}{texto}`

columnas : Número de columnas que abarcará la celda.

Alin : Indica la alineación del texto: l = izquierda, c = center, r = derecha.

### Ejemplo 5.12 (Unir celdas).

El código:

```
\begin{table}[h!]
\centering
\begin{tabular}{lll}
 & \multicolumn{2}{c}{Estimación del error}\\
 & \multicolumn{2}{c}{absoluto y relativo}\\
\rowcolor{LightBlue2} x_n & x_{n+1} &
 & & & $|x_{n+1}-x_n|/|x_{n+1}|$\\ \hline
-3.090721649 & 2.990721649 & 1.6717\\
-2.026511552 & 1.064210097 & 0.525143859\\
-1.205340185 & 0.821171367 & 0.681277682\\ \hline
\end{tabular}
\caption{}
\end{table}
```

produce:

| Estimación del error<br>absoluto y relativo |             |                           |
|---------------------------------------------|-------------|---------------------------|
| $x_n$                                       | $x_{n+1}$   | $ x_{n+1}-x_n / x_{n+1} $ |
| -3.090721649                                | 2.990721649 | 1.6717                    |
| -2.026511552                                | 1.064210097 | 0.525143859               |
| -1.205340185                                | 0.821171367 | 0.681277682               |

Tabla 5.7

### Usando el paquete booktabs

#### Ejemplo 5.13 (Usando “booktabs”)

Observe cómo el espaciado mejora usando los comandos `\midrule` y `\bottomrule` del paquete **booktabs**

**El código:**

```
% En el preámbulo
% \usepackage{booktabs}
\begin{table}[h!]
\centering
\begin{tabular}{lll}
& \multicolumn{2}{c}{Estimación del error}\\
& \multicolumn{2}{c}{absoluto y relativo}\\
\rowcolor{LightBlue2} x_n & x_{n+1} & \\
& & & $|x_{n+1}-x_n|/|x_{n+1}|$\\ \midrule
-3.090721649 & 2.990721649 & 1.6717\\
-2.026511552 & 1.064210097 & 0.525143859\\
-1.205340185 & 0.821171367 & 0.681277682\\ \bottomrule
\end{tabular}
\caption{}
\end{table}
```

**produce:**

| Estimación del error<br>absoluto y relativo |             |                           |
|---------------------------------------------|-------------|---------------------------|
| $x_n$                                       | $x_{n+1}$   | $ x_{n+1}-x_n / x_{n+1} $ |
| -3.090721649                                | 2.990721649 | 1.6717                    |
| -2.026511552                                | 1.064210097 | 0.525143859               |
| -1.205340185                                | 0.821171367 | 0.681277682               |

Tabla 5.8

## 5.7 Espacio vertical en las filas.

A veces el texto matemático queda muy pegado a alguno de los bordes de las celdas y necesitamos hacer un poco de espacio hacia arriba, hacia abajo o variar el ancho de la celda. Como ya vimos una solución es usar el paquete **booktabs**.

Para el caso de varias filas, se puede usar los argumentos opcionales del comando `\\`. El formato es `\\[longitud]`. Se puede usar `\smallskipamount`, `\medskipamount` y `\bigskipamount` que corresponden a los comandos de espacio vertical `\smallskip`, `\medskip` y `\bigskip`. Por supuesto, se puede usar una longitud en centímetros, etc.

**Ejemplo 5.14**

En este ejemplo tenemos una *tabla problemática*: El texto matemático está muy ajustado.

```
\begin{tabular}{l l l}\hline
$\dfrac{x}{x+1}$ & $\sqrt{x}$ & x^{2^n}\\ \hline
\end{tabular}
```

$$\frac{x}{x+1} \quad \sqrt{x} \quad x^{2^n}$$

Hay varias soluciones.

- 1 En una fila problemática se puede `\vphantom{ expresión }`. Este comando produce espacio vertical ajustada a la expresión y no produce espacio horizontal. En el caso que nos ocupa, podemos usar la misma expresión con subíndice y superíndice, por ejemplo `\vphantom{\dfrac{x}{y}_{t}^T}`, en cualquier lugar de la fila. Esto crea espacio vertical debido al subíndice y el superíndice.

El **código**:

```
\begin{tabular}{l l l}\hline
$\dfrac{x}{x+1}$ & $\sqrt{x}$ & x^{2^n} \vphantom{\dfrac{x}{y}_{t}^T} \\ \hline
\end{tabular}
```

produce:

$$\frac{x}{x+1} \quad \sqrt{x} \quad x^{2^n}$$

- 2 Una solución es usar `\\[xcm]` para reducir el espacio de una fila en blanco, arriba de la fila, y aumentar el espacio abajo.

El **código**:

```
\begin{tabular}{l l l}\hline
& & \\[-0.2cm] & & % reducir
$\dfrac{x}{x+1}$ & $\sqrt{x}$ & x^{2^n} \\[0.3cm] \hline % aumentar
\end{tabular}
```

produce:

$$\frac{x}{x+1} \quad \sqrt{x} \quad x^{2^n}$$

- 3 Otra solución puede ser usar el comando

`\specialrule{grosor}{espacio arriba}{espacio abajo}`

del paquete **booktabs**.

El **código**:

```
%\usepackage{booktabs}
\begin{tabular}{rccc}
\toprule
A & Va & Vb & R\\
\midrule
A & 1 & 2 & 3\\
\specialrule{0.4pt}{2pt}{5pt}%2pt arriba, 5pt abajo
E & $\dfrac{x}{x+1}$ & $\sqrt{x}$ & x^{2^n} \\
\specialrule{0.4pt}{5pt}{2pt}%
E & 1 & 2 & 3\\
\bottomrule
\end{tabular}
```

produce:

| A | Va              | Vb         | R         |
|---|-----------------|------------|-----------|
| A | 1               | 2          | 3         |
| E | $\frac{x}{x+1}$ | $\sqrt{x}$ | $x^{2^n}$ |
| E | 1               | 2          | 3         |

- 4 También se puede usar, de manera local o global, `\arraystretch`. El comando `\arraystretch` sirve para modificar la altura de los renglones en un entorno de tablas o matrices.

Por defecto, cada fila en un tabular o array tiene una separación estándar. Si se requiere filas más (o menos) “especiadas” o más “compactas”, se usa `\renewcommand{\arraystretch}{factor}`.

Por ejemplo

```
{ % grupo de alcance local
\renewcommand{\arraystretch}{1.5} % Escalar la altura de las filas
a 1.5 veces lo normal
\begin{longtable}{lcl}
A & & \sqrt{x^{x^2}}\\
B & & ...\\
\end{longtable}
```



```
...
\end{longtable}
} % fin del grupo de alcance local
```

- 5 También se puede usar el paquete **tabu** para controlar el espacio vertical de toda la tabla. El paquete **tabu** tiene la instrucción

```
\setlength\extrarowheight{xpt}
```

para aplicar espacio vertical de xpt en las filas.

El código:

```
\setlength\extrarowheight{5pt}
\begin{tabu}{rccc}
\firstline
Antea & Valor & Value & Valor\\
\midrule
A & 1 & 2 & 3\\
E & $\frac{x}{x+1}$ & $\sqrt{x}$ & x^{2^n} \\
E & 1 & 2 & 3\\
\lastline
\end{tabu}
```

produce:

| Antea | Valor           | Value      | Valor     |
|-------|-----------------|------------|-----------|
| A     | 1               | 2          | 3         |
| E     | $\frac{x}{x+1}$ | $\sqrt{x}$ | $x^{2^n}$ |
| E     | 1               | 2          | 3         |

## 5.8 Ancho de las columnas

En general, el entorno tabular ajusta el ancho de las columnas de acuerdo a el ancho de lo que contienen, esto hace que a veces se exceda el ancho de la página. Se puede controlar el ancho de las columnas indicándole al entorno el tamaño de cada columna. Esto se hace con la instrucción **p{*xcm*}** donde *xcm* es el ancho de la columna. Si tenemos texto, el cambio de renglón se debe forzar con el comando **\par** (fin de párrafo).

Ejemplo 5.15 (Ancho de las columnas).

En este ejemplo definimos un ambiente tabular (en realidad, longtable, para lidiar con el cambio de página en este libro) con dos columnas, la primera de 3cm y la segunda de 10cm. Esto se hace con las intrucciones (de agregar “párrafo” p{xcm}) `p{3cm}` y `p{10cm}` en las opciones de alineamiento.

Para hacer el cambio de renglón se usa el comando `\par` al final del renglón. Este comando indica el final de un párrafo y por lo tanto, el final de la línea.

El código:

```
\begin{tabular}{|p{3cm}|p{10cm}|}\hline
\textit{Representación} & \textit{Notación} \\ \hline
 $R_{4-4,2}(O_6)$ & $\par$ $R_{4-4,2}(O_6)$ &
Representación 3, en registro algebraico (R^4) en $\mathbb{R}$),
interpretación de la letra como incógnita (2), de la relación
entre volumen-altura-radio
del vaso unidad (O_6): $U = \pi 2r^2h$ & %Fin de párrafo
Representación 4, en registro algebraico (R^4) en el conjunto de
los números reales (4), interpretación de la letra como incógnita (2),
de la relación entre volumen-altura-radio del vaso unidad
(O_6): $h = U/\pi 2r^2$. \\ \hline
Resumen &
\begin{enumerate}
\item $R_{4-4,2}(O_6)$
\item $R_{4-4,2}^4(O_6)$
\item $R_{4-1,1}(O_8)$
\end{enumerate} \\ \hline
\end{tabular}
```

produce:

| Representación                         | Notación                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $R_{4-4,2}(O_6)$<br>$R_{4-4,2}^4(O_6)$ | Representación 3, en registro algebraico ( $R^4$ ) en $\mathbb{R}$ ), interpre-<br>tación de la letra como incógnita (2), de la relación entre<br>volumen-altura-radio del vaso unidad ( $O_6$ ): $U = \pi 2r^2h$<br>Representación 4, en registro algebraico ( $R^4$ ) en el conjunto<br>de los números reales (4), interpretación ... del vaso unidad<br>( $O_6$ ): $h = U/\pi 2r^2$ . |
| Resumen                                | 1. $R_{4-4,2}(O_6)$<br>2. $R_{4-4,2}^4(O_6)$<br>3. $R_{4-1,1}(O_8)$                                                                                                                                                                                                                                                                                                                      |

El siguiente ejemplo es un poco más elaborado (usar `\usepackage{graphicx}`),

**Ejemplo 5.16 (Texto e imágenes en columnas)**

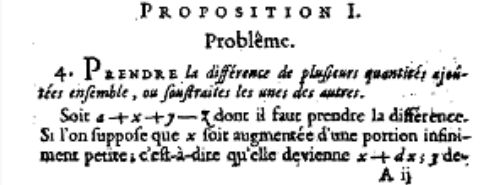
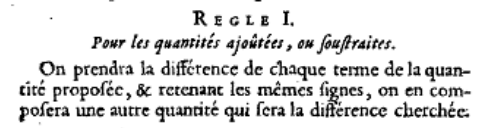
El código:

```

\begin{table}[h!]
 \centering
 \begin{tabular}{|p{6.5cm}|p{6.5cm}|}\hline
 \begin{center}
 \includegraphics[width=6.5cm]{images/Utilizacion4.pdf}
 \par\textbf{PROPOSICIÓN I}
 \par\textbf{Problema}
 \end{center}
 4. TOMAR la diferencia de varias cantidades sumadas,
 o sustraídas ... & %Cambio de columna
 \begin{center}
 \includegraphics[width=6.5cm]{images/Utilizacion5.pdf}
 \par\textbf{REGLA I}
 \par\textbf{\textit{Para las cantidades sumadas o sustraídas}}
 \end{center}
 Tomemos la diferencia de cada término de la cantidad
 propuesta, y ...\\
 \end{tabular}
\caption{La tabla muestra el modelo:...}\label{ML:tabla_escalada2}
\end{table}

```

produce:

|                                                                                                                                                                                                    |                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>PROPOSICIÓN I<br/>Problema</p> <p>4. TOMAR la diferencia de varias cantidades sumadas, o sustraídas ...</p> |  <p>REGLA I<br/>Para las cantidades sumadas o sustraídas</p> <p>Tomemos la diferencia de cada término de la cantidad propuesta, y ...</p> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Alineamiento horizontal:** `\raisebox{}{}`

Si las imágenes están en la primera columna es probable que éstas queden alineadas arriba de la caja y el texto quede alineado en el fondo de su caja. Por ejemplo, en el código

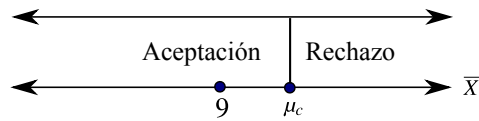
```

\begin{tabular}{|p{5cm}|c|}
 \hline
 Afirmaci\ 'o{n}: $\mu \leq 9$ \par
 $H_{\{0\}}: \mu = 9 \left(\leq \right), \quad H_{\{1\}}: \mu > 9$ %
 & \includegraphics{images/cap6_fig20}
 \hline
\end{tabular}

```

produce:

Afirmación:  $\mu \leq 9$   
 $H_0 : \mu = 9 (\leq), \quad H_1 : \mu > 9$

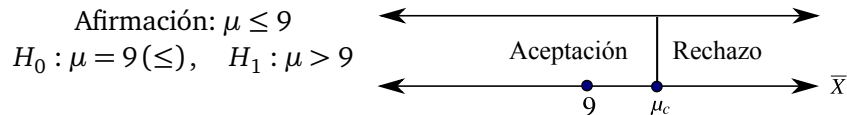


Podemos alinear horizontalmente texto y figuras de manera manual aplicando un desplazamiento hacia arriba o hacia abajo del contenido de la celda con `\raisebox{...}{...}`. En el código anterior, podríamos poner el texto en el ambiente `tabular` y la figura a la par, aplicando un desplazamiento hacia abajo (en este caso `-0.3in` es suficiente),

El código:

```
\begin{center}
\begin{tabular}{c} % Texto
 Afirmaci\{'o\}n$: \mu \leq 9$\\
 $H_{0}: \mu = 9 \left(\leq \right) , \quad H_{1}: \mu > 9$
\end{tabular}
% Bajamos el gráfico -0.3 pulgadas
\raisebox{-0.35in}{\includegraphics{images/cap6_fig20}}
\end{center}
```

produce:



Se puede usar “automáticamente” `\raisebox{-\height+\baselineskip}{...}`

También se puede usar un entorno `minipage` en el entorno `tabular`

```
\begin{tabular}{p{5cm}p{5cm}}
\begin{minipage}{5cm} Afirmaci\{'o\}n$: \mu \leq 9$\\
 $H_{0}: \mu = 9 \left(\leq \right) , \quad H_{1}: \mu > 9$
\end{minipage}
&
\begin{minipage}{5cm}\raggedleft
\includegraphics{images/cap6_fig20}
\end{minipage}
\end{tabular}
```

#### Ejemplo 5.17 (Figuras en el margen izquierdo)

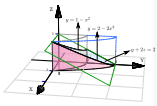
En este ejemplo el texto en la derecha no alinearía bien con la figura, una solución es bajar un poco la figura.

El código:

```
\hspace*{-2.8cm}
\begin{tabular}{p{2cm}p{13cm}}
\raisebox{-0.7cm}{\includegraphics[width=2cm]{images/exersolido21.pdf}}
& Sólido $$$
 limitado por las superficies $y = 2 - 2x^2$; $y = 1 - x^2$;
 \;; $y + 2z = 2$;
 \;; $x = 0$ $y = z = 0$;
 en el I octante.\\
\end{tabular}
```

produce: (compilado con PDFLaTeX):

Sólido Q limitado por las superficies  $y = 2 - 2x^2$ ;  $y = 1 - x^2$ ;  $y + 2z = 2$ ;  $x = 0$  y  $z = 0$ ; en el I octante.



## 5.9 Modo matemático con tabularx

A veces es conveniente usar el ambiente **tabular** habilitado para texto matemático. Esto se puede hacer con el paquete **tabularx**. Debemos poner en el preámbulo

```
\usepackage{tabularx}
```

Este paquete habilita el comando **newcolumntype** para definir columnas con contenido en modo matemático.

### Ejemplo 5.18

En este ejemplo se habilita el ambiente **tabular** para texto matemático. Observe que se usa una línea 'invisible' de altura **1cm** para crear espacio vertical en la columna donde el texto matemático quedaría algo ajustado. También se hace espacio hacia abajo con **-0.3cm**. El código:

```
%Las columnas D aceptan texto matemático centrado: |c|
\newcolumntype{D}{>{\displaystyle}c<{}}
%Las columnas M aceptan texto matemático ala izquierda: |l|
\newcolumntype{M}{>{\l}c<{}}
%Se usa 'tabular' normal.
\begin{tabular}{|D|D|D|D|M|D|M|r|}\hline
n & -1 & 0 & 1 & 2 & 3 & 5 & 5 & \\\hline
a_n & 8 & 5 & 2 & 2 & 4 & n & n & \\\hline
r_n & 0 & 1 & 2 & 5 & 22 & 444 & 444 & \\\hline
s_n & 1 & 0 & 1 & 2 & 9 & 7 & 7 & \\\hline
& & & & & & & & \\\hline
F & 0 & -2 & \frac{n}{4} & \frac{n}{4} & n^2 & 2 & 2 & \\\hline
```



Tabla 5.10. Resultados del cuestionario diagnóstico

## 5.11 Personalizar las leyendas (“caption”)

Las leyendas de las tablas y las figuras se pueden personalizar con un par de paquetes, el paquete **caption** y el paquete **floatrow** (que es una extesión del paquete **float**).

### Paquete “caption”

Con este paquete podemos personalizar las leyendas de las tablas y figuras. En este libro se usa este paquete. Las opciones escogidas son: texto **small** y la etiqueta y la numeración en negrita *rematada por un punto*. Para esto se agregó al *preámbulo*

```
\usepackage[small,bf,labelsep=period]{caption}
```

También este paquete habilita dos comandos para poner leyendas a las tablas y a las figuras: `\captionof{table}{...}` para tablas y `\captionof{figure}{...}` para figuras. Estos comandos agregan leyendas a las tablas y a las figuras sin necesidad de que esté presente un entorno. Este comando respeta los contadores de **table** y **figure**. Si se usa con el paquete **lontable** (al día de hoy) se debe usar este último comando en vez del comando **caption** de **lontable**.

### Ejemplo 5.20 (Usando el paquete “caption”)

Nótese que en este ejemplo se usa el entorno **tabular** con una leyenda pero sin usar un entorno **table** o **figure**.

El código:

```
\begin{center}
\begin{tabular}{ccl} % usa "booktabs" y "caption"
 i & x_i & & $y_i=f(x_i)$ \\ \midrule
 1 & $x_0=0$ & & 0 \\
 2 & $x_1=0.75$ & & -0.0409838 \\
 3 & $x_2=1.5$ & & 1.31799 \\ \bottomrule
\end{tabular}
\captionof{table}{Tabla usando {\tt "captionof\{table\}\{...\}"}
\end{center}
```

produce:

| $i$ | $x_i$        | $y_i = f(x_i)$ |
|-----|--------------|----------------|
| 1   | $x_0 = 0$    | 0              |
| 2   | $x_1 = 0.75$ | −0.0409838     |
| 3   | $x_2 = 1.5$  | 1.31799        |

Tabla 5.11. Tabla usando “captionof{table}{...}”

**Posición de las leyendas.** Se puede variar la posición de las leyendas, arriba, abajo, al lado, etc. y aplicar otros efectos. Para hacer esto puede, por ejemplo, ver la documentación de los paquetes **floatarrow**, **sidecap** y **threeparttable**.

## 5.12 Tablas con TikZ y tcolorbox

En el capítulo 10 se describe cómo usar los paquetes **TikZ** y **tcolorbox**. Usando estos paquetes podemos implementar tablas de apariencia agradable.

### Tablas con tcolorbox

El paquete **tcolorbox** proporciona un entorno para cajas de texto con color y enmarcados con una línea de encabezamiento. Opcionalmente, esta caja puede dividirse en una parte superior y otra inferior. El paquete **tcolorbox** puede usarse para la configuración de ejemplos, teoremas, definiciones, etc. Por ejemplo, el **código**:

```
% \usepackage{array,tabularx}
% \usepackage[table, xllnames]{xcolor}

\newcolumntype{Y}{>{\raggedleft\arraybackslash}X} % Ver tabularx
%
\tcbsset{enhanced,fonttitle=\bfseries\large,fontupper=\normalsize\sffamily,
 colback=LightCyan1,colframe=DarkOrange4,colbacktitle=DarkOrange4,
 coltitle=black,center title
}

%%-Tabla - Estilo beamer
\begin{tcolorbox}[tabularx={X||X||X},title= {\white Iteración}, beamer]
 & x_i & & $y_i=f(x_i)$ \\ \hline
A & $x_0=0$ & & 0 \\ \hline
B & $x_1=0.75$ & & -0.0409838 \\ \hline
C & $x_2=1.5$ & & 1.31799 \\
\end{tcolorbox}
```

produce:

| Iteración |              |  |                |
|-----------|--------------|--|----------------|
|           | $x_i$        |  | $y_i = f(x_i)$ |
| A         | $x_0 = 0$    |  | 0              |
| B         | $x_1 = 0.75$ |  | -0.0409838     |
| C         | $x_2 = 1.5$  |  | 1.31799        |

### Tablas con Tikz

Una forma de crear gráficos directamente con comandos **TeX** es usar **PGF/TikZ**. El paquete **TikZ** puede producir gráficos portables tanto en formato PDF como PostScript utilizando **PDFTeX**, **PDFLaTeX** o **ConTeXt** (ver 10).



Podemos usar el paquete **TikZ** para agregar sofisticación a nuestras tablas.

| $i$ | $x_i$        | $y_i = f(x_i)$ |
|-----|--------------|----------------|
| 1   | $x_0 = 0$    | 0              |
| 2   | $x_1 = 0.75$ | -0.0409838     |
| 3   | $x_2 = 1.5$  | 1.31799        |

Tabla 5.12. Tabla usando Tikz

La tabla anterior se generó con el **código**

```
\begin{dataTable}{c1l}% requiere varios paquetes
{\white i} & {\white x_i} & {\white $y_i=f(x_i)$} \\ \midrule[0pt]
1 & $x_0=0$ & 0 \\ \midrule
2 & $x_1=0.75$ & -0.0409838 \\ \midrule
3 & $x_2=1.5$ & 1.31799 \\
\end{dataTable}
\captionof{table}{Tabla usando Tikz}
```

Para que el entorno **dataTable** esté habilitado, debemos agregar en el *preámbulo* (o en archivo separado .tex) un entorno **dataTable** para generar tablas usando **Tikz**

El **código** completo es,

```
\documentclass{book}
\usepackage[T1]{fontenc}
\usepackage{amsmath,caption,array,pstricks,booktabs}
%-----Entorno DataTable-----
\usepackage{tikz}
\usetikzlibrary{calc,fit,shadows,arrows,positioning}
\pgfdeclarelayer{background}
\pgfdeclarelayer{foreground}
\pgfsetlayers{background,main,foreground}
% Data Table
\newsavebox{\dataTableContent} % Caja
\newenvironment{dataTable}[1] % Inicio nuevo entorno
{%
\begin{lrbox}{\dataTableContent}%
\begin{tabular}{#1}%
{%
\end{tabular}
\end{lrbox}
\begin{tikzpicture}
\node [inner xsep=0pt] (tbl){\usebox{\dataTableContent}};
\begin{pgfonlayer}{background}
% tabla
```

```

\draw[rounded corners=1pt,top color=gray!1,bottom color=gray!30,
 draw=black](tbl.north east) rectangle(tbl.south west);
% línea superior
\draw[rounded corners=1pt,top color=gray!10!black,
 bottom color=gray!50!black,draw=black]%
($(tbl.north west)$) rectangle
 ($(tbl.north east)-(0,1.5\baselineskip)$);
% línea inferior
\draw[rounded corners=0.25pt,fill=gray,draw=black]%
(tbl.south west) rectangle($(tbl.south east)+(0,0.05)$);
\end{pgfonlayer}
\end{tikzpicture}}

%-----Entorno DataTable-----

\begin{document}

% Usando el entorno
\begin{center}
\begin{dataTable}{c|l}%
 {\white i} & {\white x_i} & {\white $y_i=f(x_i)$} \\ \midrule[0pt]
 1 & $x_0=0$ & & 0 \\ \midrule
 2 & $x_1=0.75$ & & -0.0409838 \\ \midrule
 3 & $x_2=1.5$ & & 1.31799 \\
\end{dataTable}
\captionof{table}{Tabla usando Tikz}
\end{center}

\end{document}

```

## 5.13 Generador automático de tablas

En Internet se puede encontrar un generador automático de tablas: "Tables Generator" ([https://www.tablesgenerator.com/latex\\_tables](https://www.tablesgenerator.com/latex_tables))

Tables Generator

LaTeXHTMLTextMarkdownMediaWiki

AdvertisementFileEditTableColumnRowCellHelp

**B***I*U

Default table style

A

B

C

D

E

1

$\$a_x\$$

1

Leyenda

2

3

$\$x^2\$$

$\$\dfrac{1}{x^2}\$$

3

4

3

Tres celdas

2

3

4

1

2

3

Generate

Result (click "Generate" to refresh)

1

% Please add the following required packages to your document preamble:

2

% \usepackage[table,xcdraw]{xcolor}

3

% If you use beamer only pass "xcolor=table" option, i.e. \documentclass[xcolor=table]{beamer}

4

\begin{table}[]

5

\begin{tabular}{|l|l|l|l|l|}

6

\hline

7

$\$a_x\$$  & \multicolumn{1}{c|}{1} & & \multicolumn{3}{c|}{Leyenda} & \\ \hline

8

3 &  $\$x^2\$$  & &  $\$\dfrac{1}{x^2}\$$  & 3 & 4 & \\ \hline

9

\multicolumn{3}{|l|}{\cellcolor{HTML}{3166FF} Tres celdas}} & 2 & 3 & \\ \hline

10

1 & 2 & & 3 & & 3 & 4 & \\ \hline

11

\end{tabular}

12

\end{table}

También hay una macro para **Calc** de **LibreOffice** llamada **Calc2LaTeX**. Se puede ver las instrucciones en <https://tex.stackexchange.com/questions/179208/how-to-use-calc2latex>



## 6.1 Introducción

Además de texto corriente y texto en modo matemático, podemos insertar figuras en los formatos **.pdf**, **.png**, **.jpg**, **.eps** y **.svg** (con el paquete **svg**), o *figuras nativas*, generadas directamente con algún paquete **LaTeX** (Tiks, PSTricks, TeXDraw,...).

Otros formatos, como **.gif**, **.tiff**, etc., se pueden convertir a los formatos soportados con programas (libres) como **Inkscape**, **Gimp**, etc.

### ¿Cuál es el formato de imagen adecuado?

Los formatos **.eps** (Encapsulated PostScript), **.pdf** y **.svg** son adecuados para las figuras usuales en matemáticas. Se ven bien en pantalla y son adecuados para la impresión. Para el manejo de imágenes generales (capturas de pantalla, fotografía, etc.) es conveniente usar el formato **.png** o **.jpg**. Como dijimos, los otros formatos (**.gif**, **.wmf**,... etc.) se pueden convertir a los formatos soportados con programas, por ejemplo **Gimp**, **Inkscape** o usando el comando **convert** de **ImageMagik**.

### ¿Cómo insertar las figuras?

Aquí vamos a describir la manera fácil de insertar figuras. Todo el manejo gráfico lo vamos a hacer usando el paquete **graphicx**.

En lo que sigue, vamos a considerar las siguientes tareas,

- 1 Insertar figuras **.eps** (PostScript Encapsulado): Este es un formato de alta calidad y el de mayor soporte en **LaTeX** (aunque el formato **.pdf** ha ganado mucho terreno).
- 2 Insertar figuras **.jpg**, **.png**, **.pdf**, **.svg**, etc.
- 3 Insertar figuras cuando compilamos con **PDFLaTeX**
- 4 Convertir imágenes a otro formato con Software libre.
- 5 Extraer y/o recortar figuras de libros o de Internet.
- 6 Crear figuras nativas con **Tikz**, **LaTeXDraw**.

## 6.2 Compilando con PDFLATEX

Compilando con  
PDFLATEX

Este libro fue compilado con **PDFLaTeX** usando la distribución **TeXLive 2012**. Usa todas las figuras soportadas por **PDFLaTeX**, es decir figuras **.png**, **.pdf**, **.jpg** y **.eps**. En realidad, el formato **.eps** no está soportado, pero esta versión hace la conversión de **.eps** a **.pdf** en el proceso de compilación. Las figuras en formato **.svg** requieren el paquete **svg** o convertirlas a formato **.pdf**, por ejemplo, con algún software (como **Inkscape**).

111

### PDFLaTeX con distribuciones TeXLive 2010 o superior.

Si su distribución **TeX** es **TeXLive 2010** o superior entonces, si compila con **PDFLaTeX** puede incluir gráficos en formato **.png**, **.pdf**, **.jpg** y **.eps** sin ningún problema. Se usa el comando **\includegraphics{}**

```
\includegraphics[opciones]{ ruta y nombre de la figura }
```

Las opciones para escalar la figura son **width**, **height** y **scale**. Para rotarla se usa la opción **angle** y para 'recortarla' (clip) se usa la opción **clip**. También se puede usar el comando sin ninguna opción.

### Opciones adicionales

indexLaTeX@LaTeX El comando **includegraphics** tiene varias opciones,

| Opción                      |                                                                                                                                                                                                                      |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>width=xcm</b>            | Escala la imagen especificando el ancho deseado y manteniendo la proporción.                                                                                                                                         |
| <b>height=xcm</b>           | Escala la imagen especificando el alto deseado y manteniendo la proporción.                                                                                                                                          |
| <b>scale=x</b>              | Escala la imagen según una proporción: <b>scale=0.5</b> indica escalar la imagen a la mitad y <b>scale=2</b> escala la imagen al doble.                                                                              |
| <b>trim=lcm bcm rcm tcm</b> | Esta opción recortar la imagen: <b>lcm</b> a la izquierda, <b>bcm</b> en la parte inferior, <b>rcm</b> a la derecha, y la <b>tcm</b> en la parte superior. <b>l</b> , <b>b</b> , <b>r</b> y <b>t</b> son longitudes. |
| <b>clip</b>                 | Para que la opción <b>trim</b> trabaje se debe establecer <b>clip=true</b> .                                                                                                                                         |

**PDFLaTeX**, con **TeXLive 2010** o superior, soporta **.png**, **.pdf**, **.jpg** y si encuentra figuras **.eps** hace la conversión **.pdf**

### PDFLaTeX con otras distribuciones

**PDFLaTeX**, con **MikTeX 2.9** o con **TeXLive 2009**, solamente soporta **.png**, **.pdf**, **.jpg**. Si fuera necesario, se puede habilitar la conversión de **.eps** a **.pdf** con el paquete **epstopdf** (ver más abajo).

**N** En general, es deseable por supuesto que el texto (matemático) en una figura sea consistente con las fuentes usadas en el texto. La extensión **TextText** de **Inkscape** permite cargar los mismos paquetes de fuentes para el texto que se agrega en la figura. Si se usa **Wolfram Mathematica** para editar figuras, puede usar el paquete **MaTeX**, este paquete permite también agregar paquetes de fuentes para el texto

**Ejemplo 6.1**

En este ejemplo vamos a incluir dos gráficos, uno en el ambiente **figure** y el otro centrado y con una leyenda usando el paquete **caption**.

Estas figuras fueron implementadas primero en **Wolfram Mathematica 8** y se guardaron en formato **.pdf** y luego fueron editadas y modificadas con **Inkscape**. El texto matemático se agregó con la extensión **TeXText** de **Inkscape**.

Los gráficos están en una subcarpeta **images** de la carpeta en la que está nuestro archivo **.tex**.

El **código**:

```
\documentclass{article}
\usepackage{graphicx}
\usepackage{caption}
\usepackage[T1]{fontenc}

\begin{document} % > Compilar con PDFLaTeX

Consideremos ahora el paralelepípedo

\begin{figure}[h!] % Ambiente 'figure'
\centering % imagen sin escalar
\includegraphics{images/figura3.pdf}
\caption{Un paralelepípedo}\label{figura3}
\end{figure}

Ahora consideremos el sólido Q ...

\begin{center} % Escalada a 4cm de ancho
\includegraphics[width=5cm]{images/figura4.pdf}
\captionof{figure}{Sólido Q }
\label{figura3a}
\end{center}

\end{document}
```



Descargar archivo

produce:

Consideremos ahora ...

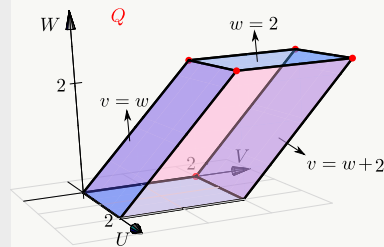


Figura 6.1. Un paralelepípedo

Ahora consideremos ...

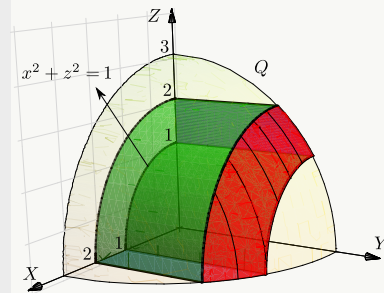


Figura 6.2. Sólido  $Q$

**Ejemplo 6.2 (PDFLaTeX con distribuciones TexLive 2010 o superior).**

En este ejemplo vamos a incluir la misma figura, pero en distintos formatos. Observe que, como estamos usando la distribución **Texlive2012** (o superior) podemos incluir figuras **.eps** sin problema. En otras distribuciones habría que hacer algunos cambios para hacer esto.

El **código**:

```
\documentclass{article}
```



Descargar archivo

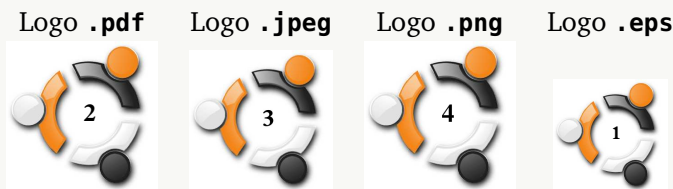
```

\usepackage[total={18cm,21cm},top=2cm, left=2cm]{geometry}
\usepackage{graphicx}

\begin{document} % Compilar con opción PDFLATEX con TeXLive2010 o superior.
\begin{tabular}{cccc}
 Logo .pdf & Logo .jpeg & Logo .png & Logo .eps\\
 \includegraphics[width=2cm]{images/ubuntu2.pdf} & % ancho 2cm
 & \includegraphics[scale=0.2]{images/ubuntu3.jpg} & % Escalada 20%
 & \includegraphics[height=2cm]{images/ubuntu4.png} & % altura 2cm
 & \includegraphics[width=1.5cm]{images/ubuntu1.eps}\\ % Conversión a pdf
\end{tabular}
\end{document}

```

produce:



### PDFLaTeX con MikTeX 2.9 o con TeXLive 2009 o menos.

Si su distribución **TeX** es **MikTeX 2.9** (al 2013 es la versión más reciente) o **TeXLive 2009** o menos, entonces si compila con **PDFLaTeX** puede incluir gráficos en formato **.png**, **.pdf**, **.jpg**. sin ningún problema. Se usa el mismo código del ejemplo anterior (excepto para figuras **.eps**).

**(N)** ¿Desea habilitar el paquete **epstopdf** en **MikTeX 2.9**? Si desea compilar con **PDFLaTeX** y habilitar la faceta de conversión sobre la marcha de figuras **.eps** a **.pdf** con **MikTeX 2.9** o **TeXLive 2009**, deberá agregar en el *preámbulo* `\usepackage{epstopdf}`. Pero también debe habilitar la opción **shell escape** para que se puede ejecutar la conversión (llamando a un programa externo). Para hacer esto se debe ir al menú de configuración del editor y en la entrada **PDFLaTeX** deberá agregar **-shell-escape** de tal manera que quede

**PDFLaTeX** -synctex=1 -interaction=nonstopmode -shell-escape  
luego debe reiniciar el editor (Recuerde que esto no es necesario en **TeXLive 2010** o más).

#### Ejemplo 6.3 (Compilando con PDFLaTeX. Opciones).

En este ejemplo se muestra una figura **.pdf** (que está en la misma carpeta que el archivo **.tex**) y el efecto de aplicar las opciones de escalamiento, rotación y 'recorte'.

El **código**:

```

\begin{tabular}{|c|c|}\hline
\includegraphics{fig.pdf}

```



```

& \includegraphics[scale=0.8]{fig.pdf} \\ \hline
\includegraphics{fig1.pdf}
& \includegraphics[scale=0.8]{fig1.pdf} \\ \hline
\includegraphics[width=0.2\textwidth]{fig.pdf}
& \includegraphics[angle=-45]{fig.pdf} \\ \hline
\includegraphics[width=0.2\textwidth]{fig.pdf}
& \includegraphics[width=2cm, angle=-45]{fig.pdf} \\ \hline
\includegraphics[width=2cm,height=4cm]{fig.pdf}
& \includegraphics[trim=2cm 2cm 4cm 0cm,clip]{fig.pdf} \\ \hline
\includegraphics[width=2cm, height=4cm]{fig.pdf}
& \includegraphics[trim = 2cm 2cm 4cm 0cm, clip]{fig.pdf} \\ \hline
\end{tabular}

```

produce:

|                                                              |                                                               |
|--------------------------------------------------------------|---------------------------------------------------------------|
| <code>\includegraphics{fig.pdf}</code>                       | <code>\includegraphics[scale=0.8]{fig.pdf}</code>             |
|                                                              |                                                               |
| <code>\includegraphics[width=0.2\textwidth]{fig.pdf}</code>  | <code>\includegraphics[angle=-45]{fig.pdf}</code>             |
|                                                              |                                                               |
| <code>\includegraphics[width=2cm,height=4cm]{fig.pdf}</code> | <code>\includegraphics[trim=2cm 2cm 4cm 0cm,clip]{...}</code> |
|                                                              |                                                               |

Para incluir figuras, sin reparar en la extensión, se debe poner en el *preámbulo*

```

\usepackage{graphicx}
\DeclareGraphicsExtensions {.pdf,.png,.jpg} % busca en este orden!

```

Esta instrucción declara las extensiones con los nombres de archivo con los que se debe probar, si están, y permite especificar el orden en el que se deben elegir los formatos gráficos cuando se incluyen archivos gráficos, dando el nombre del archivo sin la extensión. Esto es conveniente cuando se tiene el mismo archivo en distintos formatos, por ejemplo .pdf y .png, y se quiere establecer cuál formato es la prioridad (por la razón que sea).

#### Ejemplo 6.4

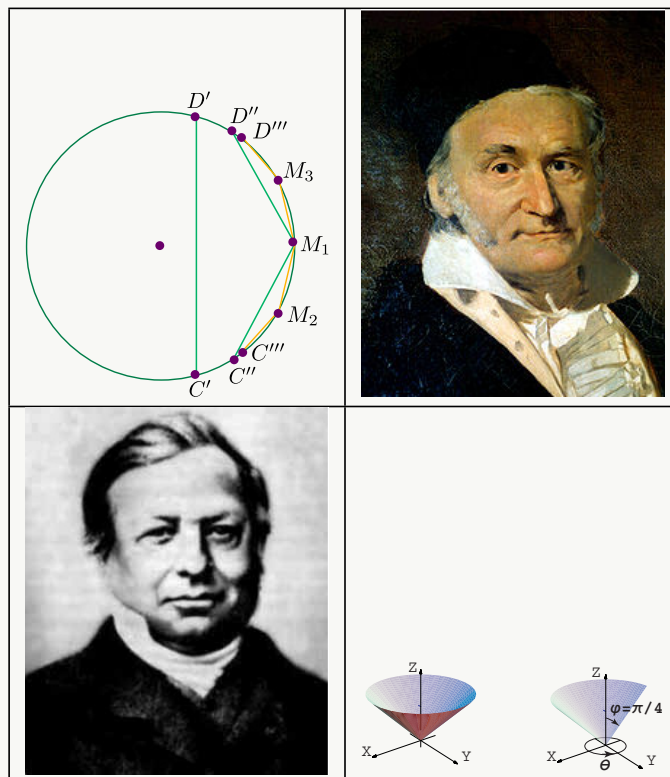
En este ejemplo vamos a insertar las figuras **geometria1.pdf**, **gauss.png**, **lioville.jpg** y la figura **coodesfericas.pdf**.

El código:

```
\documentclass {article}
\usepackage{graphicx}
\DeclareGraphicsExtensions{.pdf,.png,.jpg}% No es necesario
 % indicar extensión.

\begin{document} % Compilar PDFLaTeX
\begin{center}
\begin{tabular}{|c|c|}\hline
\includegraphics[width=4cm]{images/geometria}
&\includegraphics[width=4cm]{images/gauss}\\ \hline
\includegraphics[width=4cm]{images/lioville}
&\includegraphics[width=4cm]{images/coodesfericas}\\ \hline
\end{tabular}
\end{center}
\end{document}
```

produce:



## 6.3 Compilando con LaTeX. Figuras .eps



Si compilamos con **LaTeX**, obtendrá un archivo **.dvi** el cual podrá ver con un visualizador para este tipo de archivos (**Yap** en Windows, **Okular** en Ubuntu, por ejemplo).

**LaTeX** tiene un gran soporte para imágenes **.eps**. En muchos programas podemos guardar (o convertir) nuestros gráficos en este formato: **Inkscape**, **Mathematica**, **MatLab**, **QtOctave**, **WinPlot**, **Geogebra**, etc. Este formato es adecuado para gráficos simples y complejos, pero no es adecuado para fotos, 'pantallazos', etc.

116

### Convertir otras imágenes a formato .eps

Podemos convertir imágenes en formato **.pdf**, **.png**, **.jpg**, **.gif**, etc. a formato **.eps**; esto se puede hacer, por ejemplo con **Inkscape** o con **Gimp** (ver apéndice ??); solo debe abrir los archivos con alguno de estos programas (posiblemente editar algo adicional) y guardar como **.eps**.

### Incluir las figuras .eps

Para incluir las figuras **.eps** en su documento, se debe agregar el paquete **graphicx** en el *preámbulo* y los gráficos se incluyen con el comando `\includegraphics{}`.

`\includegraphics[opciones]{ nombre y ruta de la imagen...}`

Como es natural, los gráficos se deben *escalar* para que se acomoden al texto. El formato **.eps** es un formato vectorial, es decir, las figuras en este formato están especificadas en coordenadas, por lo tanto se pueden escalar sin pérdida de calidad. El escalamiento se puede hacer especificando el ancho **width =xcm**, y el alto **height =xcm** (juntos o ambos por separado) o especificando porcentaje de escalamiento: **scale=x** donde  $x \in [0, 1]$  es el porcentaje.

#### Ejemplo 6.5

En este ejemplo vamos a incluir el gráfico **ellipse.eps** que está en la subcarpeta **images** de la carpeta en la que está nuestro archivo **.tex**.

El código:

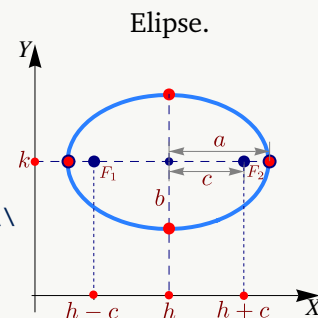
```
\documentclass{article} %Compilar -> LaTeX
\usepackage{graphicx}

\begin{document}

\begin{center}
 Ellipse.\ %Se escala a un ancho de 4cm
 \includegraphics[width=4cm]{images/ellipse.eps}
\end{center}

\end{document}
```

produce:    [Descargar archivo](#)



**Ejemplo 6.6** (Gráfico en ambiente “figure”)

El código:



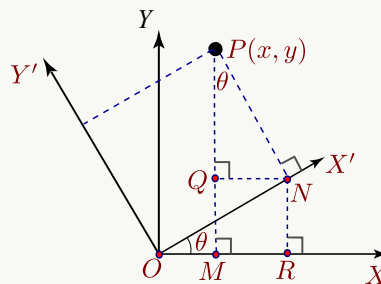
```
\documentclass{article} %Compilar -> LaTeX
\usepackage{graphicx}

\begin{document}
 Consideremos ahora una rotación de ...

 \begin{figure}[h!] % Ambiente 'figure'
 \centering % imagen sin escalar
 \includegraphics{images/figura4.eps}
 \caption{Rotación de ángulo θ }\label{figura4}
 \end{figure}
\end{document}
```

**produce:**

Consideremos ahora una rotación de ...

**Figura 6.3.** Rotación de ángulo  $\theta$ .

Podemos agregar una leyenda a la figura con el paquete **caption** sin necesidad de poner la figura en un entorno flotante. El mismo efecto se logra con el código

```
\documentclass{article} %Compilar -> LaTeX
\usepackage{graphicx}
\usepackage{caption}

\begin{document}

 Consideremos ahora una rotación de ...
 \begin{center}
 \includegraphics{images/figura4.eps}
 \captionof{figure}{Rotación de ángulo θ }\label{figura4}
 \end{center}

\end{document}
```

**Ejemplo 6.7**

Continuando con el ejemplo anterior, vamos a incluir el gráfico **ubuntu**, que está en la subcarpeta **images** de la carpeta en la que está nuestro archivo **.tex**, con un par de opciones más.

El **código**:

```
\documentclass {article} % Compilar -> LaTeX
\usepackage{graphicx}
\usepackage{caption}

\begin{document}
Logo centrado y escalado a {\tt 3cm}

\begin{center}
\includegraphics[width=3cm]{images/ubuntu.eps}
\end{center}

En la figura \ref{fig:ubuntu} se
muestra el logo de Ubuntu, centrado y
escalado.\

\begin{center}
\includegraphics[scale=0.2]{images/ubuntu.eps}
\captionof{figure}{Logo de Ubuntu}\label{fig:
ubuntu}
\end{center}
\end{document}
```

produce:

Logo centrado y escalado 3cm



En la figura 6.4 se muestra el logo de Ubuntu, centrado y escalado.



**Figura 6.4.** Logo de Ubuntu

## 6.4 De nuevo: Paquete float

Es común tener problemas en la manera como **LaTeX** acomoda los gráficos. Una manera de tomar control sobre la ubicación de los gráficos es usar el paquete **float**; para esto, agregamos en *preámbulo*,

```
\usepackage{float}
```

Ahora, en vez de digitar `\begin{table}[h!]` o `\begin{figure}[h!]`, digitamos `\begin{table}[H]` o `\begin{figure}[H]`: El gráfico o la tabla quedará donde está.

## 6.5 Paquete subfig

A veces tenemos varias figuras y nos encantaría poner una leyenda a cada una en un mismo ambiente. Esto lo podemos hacer si usamos el paquete `\usepackage{subfig}`. El siguiente ejemplo ilustra su uso.

**Ejemplo 6.8**

El código:

```
\begin{figure}
\centering
\subfloat[Converge]{\includegraphics[scale=0.6]{images/newton6}}
\subfloat[Diverge]{\includegraphics[scale=0.6]{images/newton5}}
\subfloat[Ciclo]{\includegraphics[scale=0.6]{images/newton4}}
\caption{Iteración de Newton}
\end{figure}
```

produce:

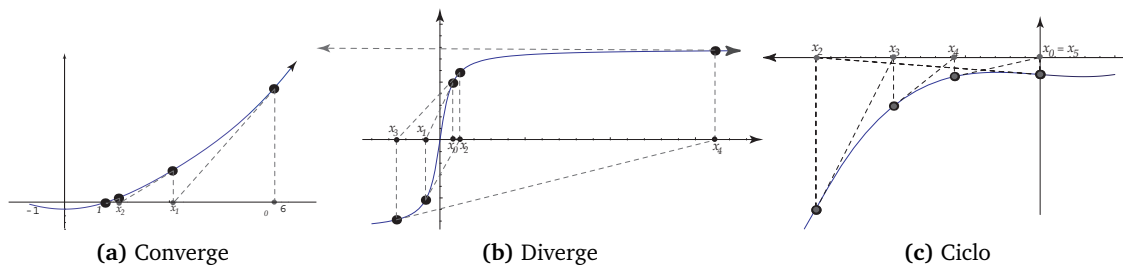


Figura 6.5. Iteración de Newton

## 6.6 Los ambientes wrapfigure y floatflt

Otros ambientes flotantes muy útiles son wrapfigure y floatflt, para poderlos utilizar se deben cargar en el preámbulo sus paquetes respectivos con las instrucciones

```
\usepackage{wrapfig} %Figuras al lado de texto
\usepackage[rflt]{floatflt} %Figuras flotantes entre el texto
```

### wrapfigure

El ambiente wrapfigure permite incluir gráficos o texto en un recuadro al lado del documento, **LaTeX** se encarga de acomodar el texto del documento alrededor del recuadro introducido. Con este ambiente se introdujo la foto de D. Knuth al inicio de este documento.

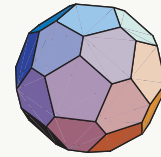
**Ejemplo 6.9**

El código:

```
\begin{wrapfigure}{r}{2.5cm}
\includegraphics{images/ML_fig11.pdf}
\end{wrapfigure}
Al incluir un recuadro con ...
```

**produce:**

Al incluir un recuadro con `\wrapfigure` se debe tomar algunas cosas en cuenta: En la definición `{r}` significa que el recuadro se introducirá a la derecha del texto, también se puede utilizar `{l}` para que sea a la izquierda. El ambiente se debe iniciar entre párrafos, es decir, es problemático escribir un ambiente `wrapfigure` en medio de un párrafo.



El recuadro será introducido justo al lado del párrafo siguiente de la definición del ambiente.

La separación del recuadro con el texto está dado por la instrucción `\columnsep` del preámbulo.

En realidad este ambiente no es “flotante”, es decir, en este caso el recuadro se introduce justo en el párrafo donde uno quiere, por lo tanto, es nuestra responsabilidad que el recuadro se “vea” bien (que no quede cortado entre páginas o cosas de este estilo); se recomienda revisar todos los gráficos o texto introducido con este comando al obtener la versión final del documento.

Este comando es frágil, por lo que no se puede utilizar dentro de otros ambientes, sin embargo, sí se puede utilizar en párrafos con multicolumnas.

Por último, el ambiente puede presentar problemas cuando el texto alrededor del recuadro no lo cubre por completo, en estos casos es mejor utilizar **parbox** o **minipage**

Note que la gran ventaja que tiene este ambiente (sobre **parbox**, por ejemplo) es que no hay que preocuparse por la cantidad de texto que hay en cada columna, **LaTeX** se encarga de la distribución de manera automática.

**floatflt**

El ambiente `floatflt` es muy similar a `wrapfigure` ya que permite la inserción de un objeto flotante rodeado de texto; en este caso **LaTeX** se encarga de acomodar el texto alrededor de él.

Para poder utilizar este ambiente se necesita incluir la librería, para esto, se coloca en el preámbulo la instrucción

```
\usepackage[rflt]{floatflt}
```

El argumento opcional `rflt` indica que, por defecto, los gráficos se colocarán a la derecha del texto, también se puede escribir `lflt` para la izquierda o `vflt` que indica que el gráfico saldrá a la derecha en páginas impares y a la izquierda en páginas pares.

Por ejemplo, el **código**:

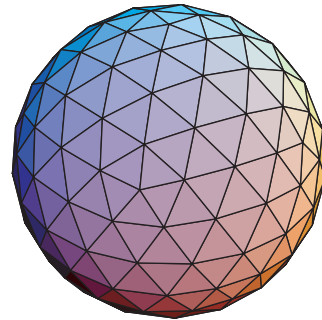
```
\begin{floatingfigure}[r]{4.5cm}
 \includegraphics{images/ML_fig12}
 \captionof{figure}{Un poliedro}
\end{floatingfigure}
```

Este ambiente `s\`olo` funciona ...

**produce:**

Este ambiente sólo funciona si se pone antes de un párrafo, la figura aparecerá lo más cerca del lugar en donde se haya escrito, esto quiere decir que **LaTeX** primero intenta poner la figura en la página actual, si no encuentra suficiente espacio vertical entonces la coloca en la página siguiente.

El argumento `[r]` es un argumento opcional que hace que el gráfico salga a la derecha del texto (no importa lo que se haya puesto al cargar la librería).



**Figura 6.6.** Un poliedro

Aunque el ambiente `floatflt` sí trabaja en páginas a doble columna se debe tener cuidado si el gráfico sobrepasa el ancho de la columna porque sino el gráfico quedará encimado sobre la columna contigua. Tampoco se debe usar el ambiente muy cerca del final de una sección, sino el gráfico quedará encimado en la sección siguiente.

Si se escribe el ambiente en el primer párrafo de una página es posible que el gráfico aparezca más abajo, es decir, el ambiente no coloca figuras al inicio de la página y, en el peor de los casos, la figura nunca aparecerá.

## 6.7 Alinear texto y figuras

Como ya habíamos indicado, el entorno **minipage** usa un punto de referencia para alinear el contenido lo que puede ser confuso. Igual puede pasar en otros entornos como **parbox** o **tabular** por ejemplo. Para controlar el alineamiento de texto y figuras o entre figuras, se puede usar el comando `\raisebox` o también el paquete **adjustbox**.

En el comando `\raisebox{distancia}{text}`, la “distancia” del primer argumento especifica la distancia que elevará el texto respecto del segundo argumento. Si es negativa, entonces baja el texto. El texto no puede contener saltos de línea. Podemos usar figuras en vez de texto.

### Ejemplo 6.10 (Usando el comando `\raisebox`)

En este ejemplo alineamos el gráfico con el texto, bajando un poco el gráfico. Por simplicidad se usa el paquete “caption”.

#### El código:

```
\documentclass{article}
\usepackage[margin=1cm, font=small]{caption}
\usepackage{graphicx}
%
\begin{document}
\begin{minipage}[t]{0.5\textwidth}
\begin{center}
```



```

\raisebox{-2.5cm}{ \includegraphics[scale=0.8]{images/ML_fig3.pdf} }
\captionof{figure}{Poliedros}
\end{center}
\end{minipage}
\hfill \begin{minipage}[t]{0.5\textwidth}
La imagen de la izquierda muestra un icosaedro junto con un
dodecaedro (figura central), los satélites son un icosaedro,
un dodecaedro y un tetraedro. Las figuras fueron generadas con
{\sc Mathematica} y maquilladas con {\it Inkscape}.
\end{minipage}
%
\end{document}

```

Produce:

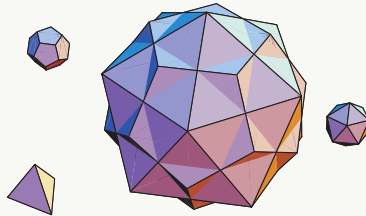


Figura 6.7. Poliedros

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

#### Ejemplo 6.11 (Usando el paquete adjustbox)

En este ejemplo alineamos el gráfico con el texto con el paquete **adjustbox**.

Podemos usar las opciones (entre otras) `valign=t` o `valign=b`. En este ejemplo usamos

```
\adjustbox{valign=t}{\begin{minipage}...\end{minipage}}
```

Por simplicidad se usa el paquete “caption”.

El código:

```

\documentclass{article}
\usepackage[margin=1cm, font=small]{caption}
\usepackage{graphicx}
%
\usepackage{adjustbox}
\begin{document}
\adjustbox{valign=t}{%
\begin{minipage}{0.5\textwidth}
\begin{center}
\includegraphics[scale=0.8]{images/ML_fig3.pdf}
\captionof{figure}{Poliedros}
\end{center}
\end{minipage}
}
\hfill \adjustbox{valign=t}{%
\begin{minipage}{0.5\textwidth}

```

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con {\sc Mathematica} y maquilladas con {\it Inkscape}.

```
\end{minipage}
}
%
\end{document}
```

Produce:

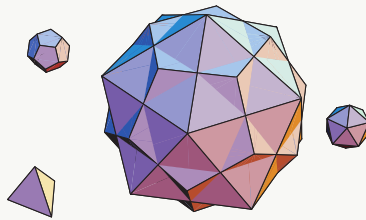


Figura 6.8. Poliedros

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

#### Ejemplo 6.12 (Varias opciones de alineamiento)

- Podemos usar la opción `export` con el paquete **adjustbox**, entonces se carga el paquete **graphicx** y se pueden usar las opciones directamente

```
\includegraphics[scale=0.7,valign=t]{...}
```

El código:

```
\documentclass{article}
\usepackage[export]{adjustbox}%
\begin{document}
\begin{center}
\fbbox{\includegraphics[scale=0.7,valign=t]{images/ML_fig3.pdf}}
\hspace{1cm}
\fbbox{\includegraphics[scale=0.3,valign=t]{images/ML_fig12}}
\captionof{figure}{Alineando figuras}
\end{center}
\end{document}
```

produce:

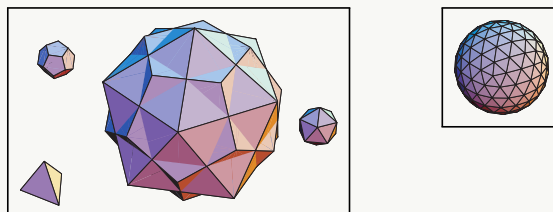


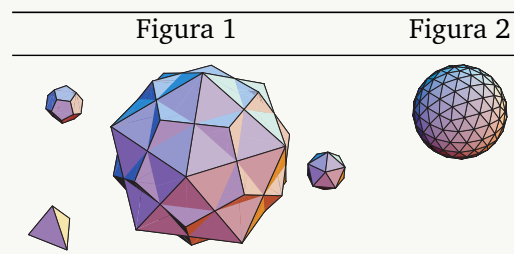
Figura 6.9. Alineando figuras

- También podemos usar el entorno **tabular** con los paquetes **adjustbox** y **booktabs**. Aquí se usa el paquete **booktabs** para controlar el espacio debajo de la primera fila con los comandos `\midrule` y `\bottomrule`

El código:

```
%\usepackage[export]{adjustbox}
%\usepackage{booktabs}
\begin{tabular}{ccc}\hline
Figura 1 & & Figura 2\\
\midrule
\includegraphics[scale=0.7,valign=t]{images/ML_fig3.pdf}
& &
\includegraphics[scale=0.3,valign=t]{images/ML_fig12} \\
\bottomrule
\end{tabular}
```

produce:

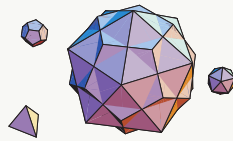


- Hay una opción que no usa los paquetes anteriores. Si queremos alinear figuras en la parte superior de un ambiente **minipage**, podemos usar el comando `\vspace{0pt}` para cambiar el punto de referencia de la figura y así **minipage** alinea en la parte superior

El código:

```
\begin{minipage}{4cm}
\vspace{0pt}
\includegraphics[scale=0.5]{images/ML_fig3.pdf}
\captionof{figure}{Poliedros}
\end{minipage}
\hfill \begin{minipage}{8cm}
\vspace{0pt}
La imagen de la izquierda muestra un icosaedro junto con un
dodecaedro (figura central), los satélites son un icosaedro,
un dodecaedro y un tetraedro. Las figuras fueron generadas con
{\sc Mathematica} y maquilladas con {\it Inkscape}.
\end{minipage}
```

produce:



**Figura 6.10.** Poliedros

La imagen de la izquierda muestra un icosaedro junto con un dodecaedro (figura central), los satélites son un icosaedro, un dodecaedro y un tetraedro. Las figuras fueron generadas con MATHEMATICA y maquilladas con *Inkscape*.

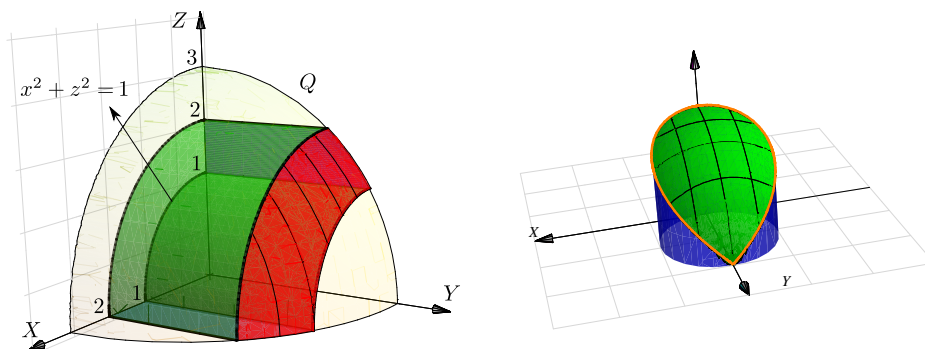


A los gráficos y las figuras se les puede hacer edición adicional usando software. En este capítulo vamos a hacer edición adicional usando **Inkscape**. Este software libre admite figuras en muchos formatos y podemos guardar en formatos como **.eps**, **.pdf** o **.png** (este último con el menú Archivo-Exportar).

**Inkscape** es un programa de gran utilidad cuando se trabaja con gráficos vectoriales. **TeXText** es una extensión para **Inkscape** que nos permiten usar código **LaTeX** dentro de **Inkscape** para componer texto matemático para nuestras figuras y gráficos. El texto **LaTeX** es compilado usando nuestra distribución **TeX** y la extensión permite usar paquetes y comandos propios.

## 7.1 Gráficos en formato vectorial

Los gráficos vectoriales usan objetos geométricos simples (puntos, líneas, curvas y polígonos) para describir, a través de ecuaciones matemáticas, imágenes digitales. Producen figuras de gran calidad y se pueden escalar sin pérdidas. En general son archivos con extensión **.svg**. Estas figuras **.svg** necesitan poca información para describir los gráficos, por ejemplo un segmento de recta solo necesita las coordenadas del punto inicial y del punto final, el grosor (thickness) y el color del borde y relleno.



**Figura 7.1.** Gráficos generadas con **Wolfram Mathematica** y guardadas en formato **.pdf** y luego editadas con **Inkscape**

Los editores de gráficos vectoriales, como **Inkscape**, son programas que permiten componer y editar imágenes en formato vectorial. Además de los archivos **.svg**, hay otros archivos que usan el modelo vectorial (**.ps**, **.eps**, **.pdf**, **.ai**, etc.) y por tanto se pueden abrir y editar sus partes. Para editar sus componentes primero debemos usar la opción “desagrupar”.

Además estos editores permiten “incrustar” otro tipo de imágenes usando la opción de “*Importar*”: Puede *importar* formatos como **.jpg**, **.png**, **.tiff** y *exporta* en formato **.png** así como muchos otros formatos basados en vectores.

En general se puede “copiar” una imagen en formato “*bimap*” (mapa de bits) digamos de una página en Internet y “pegar” en el entorno, también esta operación se puede hacer con la opción *importar e incrustar*.

## 7.2 Inkscape

**Inkscape** es un editor de gráficos vectoriales de código abierto, con capacidades similares a **Illustrator**. Es un ambiente ideal para editar los gráficos generados con programas que tengan la opción de exportar en formatos **.eps**, **.pdf**, **.svg** o algún otro formato vectorial.

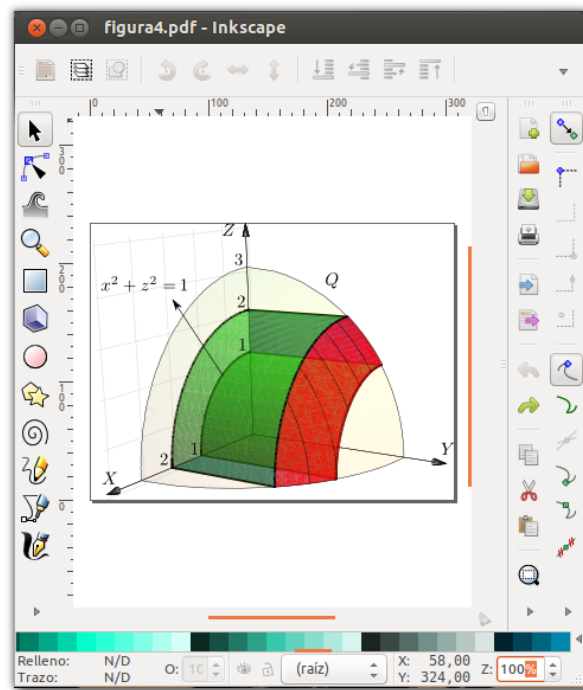


Figura 7.2. Editando un gráfico **.pdf** con **Inkscape**

### Un manual básico para Inkscape

Un manual para **Inkscape** lo puede encontrar en la revista digital Matemática, Educación e Internet (<http://www.tec-digital.itcr.ac.cr/revistamatematica/>) en

[http://www.tec-digital.itcr.ac.cr/revistamatematica/Secciones/Didactica\\_y\\_Software/RevistaDigital\\_KVindas\\_V12\\_N2\\_2012/RevistaDigital\\_Inkscape\\_KVindas.pdf](http://www.tec-digital.itcr.ac.cr/revistamatematica/Secciones/Didactica_y_Software/RevistaDigital_KVindas_V12_N2_2012/RevistaDigital_Inkscape_KVindas.pdf)

## 7.3 Extensión TexText para Inkscape

Una faceta adicional es la posibilidad de editar texto matemático **LaTeX** en el mismo entorno. Para hacer esto se puede usar la extensión **TexText**. Las instrucciones de instalación aparecen en esta misma página web <https://texttext.github.io/texttext/index.html>

Con **Inkscape** podemos abrir páginas individuales de libros en formato **.pdf** y tomar figuras y editarlas y/o modificarlas. Hay que recordar que las figuras en formato **.pdf** y **.eps** (entre otros) se pueden 'desagrupar' y modificar (vienen en formato vectorial). Esto no se puede hacer con las figuras en formatos **.jpg**, **.png**, etc. Pero la faceta principal para nuestros propósitos es que podemos agregar texto **LaTeX** en **Inkscape**.

### Agregando texto matemático con la extensión TexText

**TexText** es una extensión de **Inkscape**. Una vez instalada (para **Ubuntu** ver apéndice ?? y para **Windows** ver apéndice ??) se accede a ella con **Extensiones-TeX text**.

En la versión actual de **Inkscape** (la versión 1.0), la extensión **TexText** se encuentra en **Extensiones-Texto - TeX text**

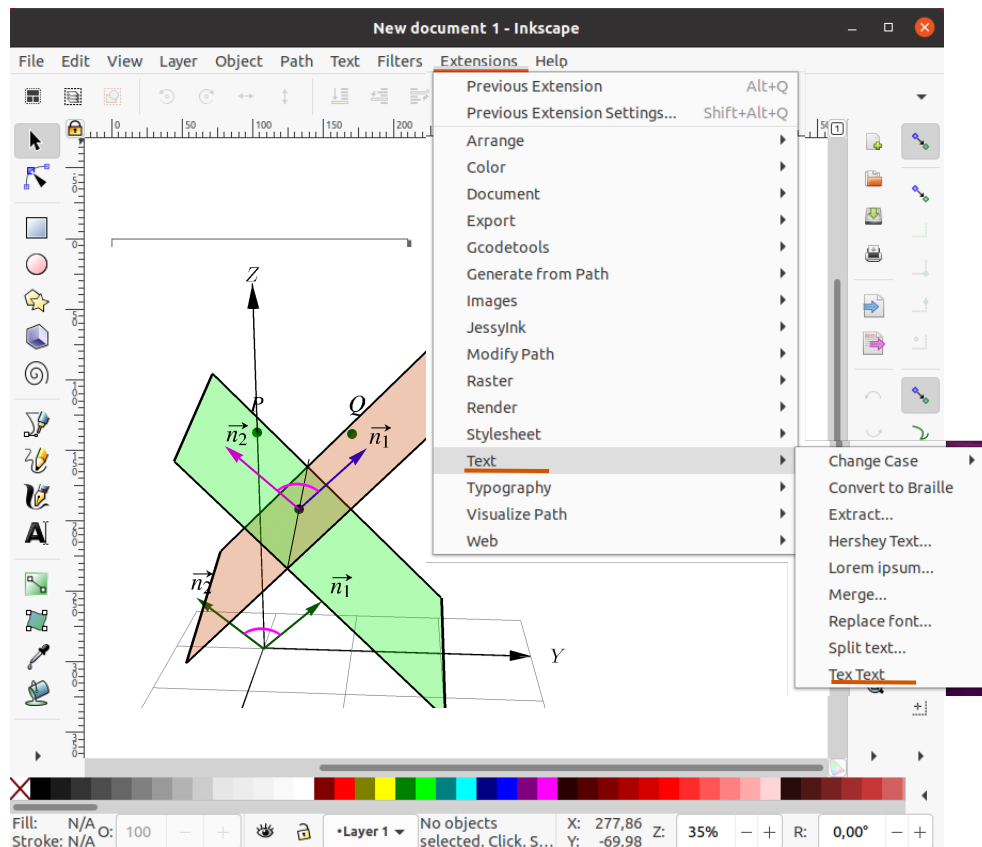


Figura 7.3. Habilitar la extensión **TexText** en el menú **Extensiones**

La figura que sigue fue elaborada con **Inkscape**, el texto matemático adicional se agregó con la extensión **TexText**. En la opción **Preamble file** se puede agregar un archivo que indica los paquetes y comandos que se desea usar.



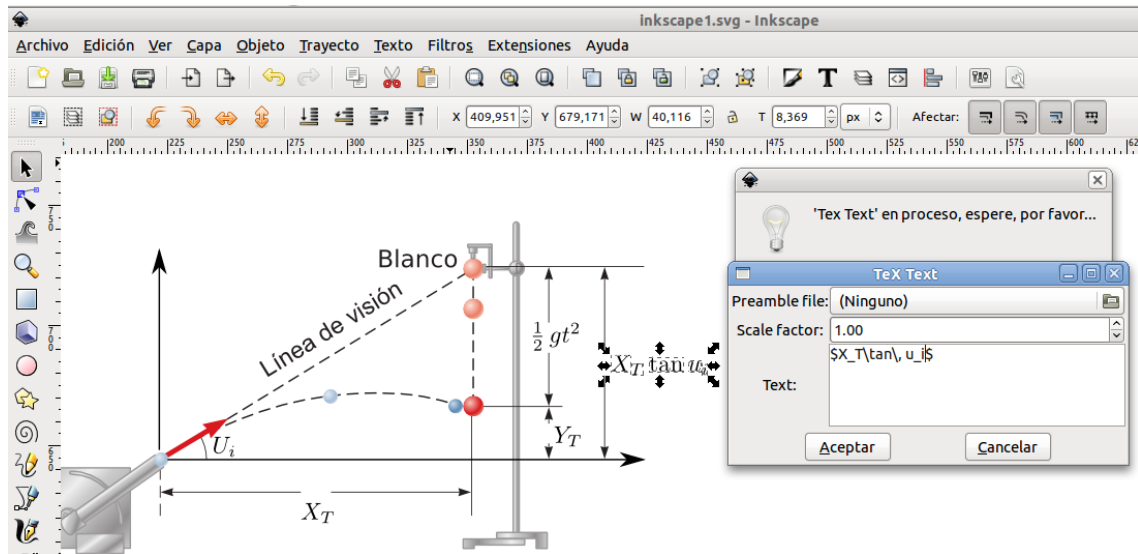


Figura 7.4. Inkscape 1.0: Editando una figura con la extensión **TexText**

Esta ventana permite código **LaTeX** básico porque no están habilitados paquetes adicionales. Si la sintaxis está bien, hacemos clic en **Aceptar** y obtenemos el texto **LaTeX**. Este texto se puede escalar y redimensionar sin perder calidad.

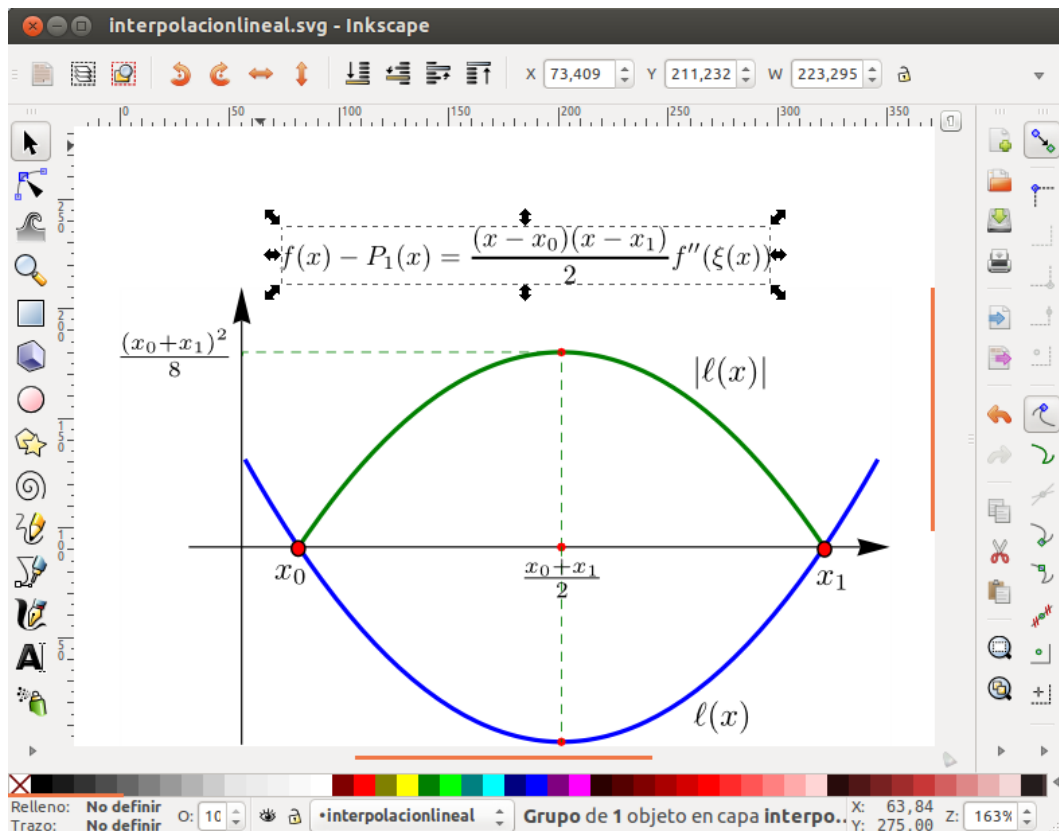


Figura 7.5. Resultado de una edición con **TexText**

### 7.3.1 Opciones

La ventana de edición **TexText** tiene dos opciones, la primera nos permite cargar un archivo de texto **.txt** (el archivo lo ponemos en algún lugar del disco duro) con llamadas a paquetes y con comandos personales y la segunda nos permite poner un factor de escala.

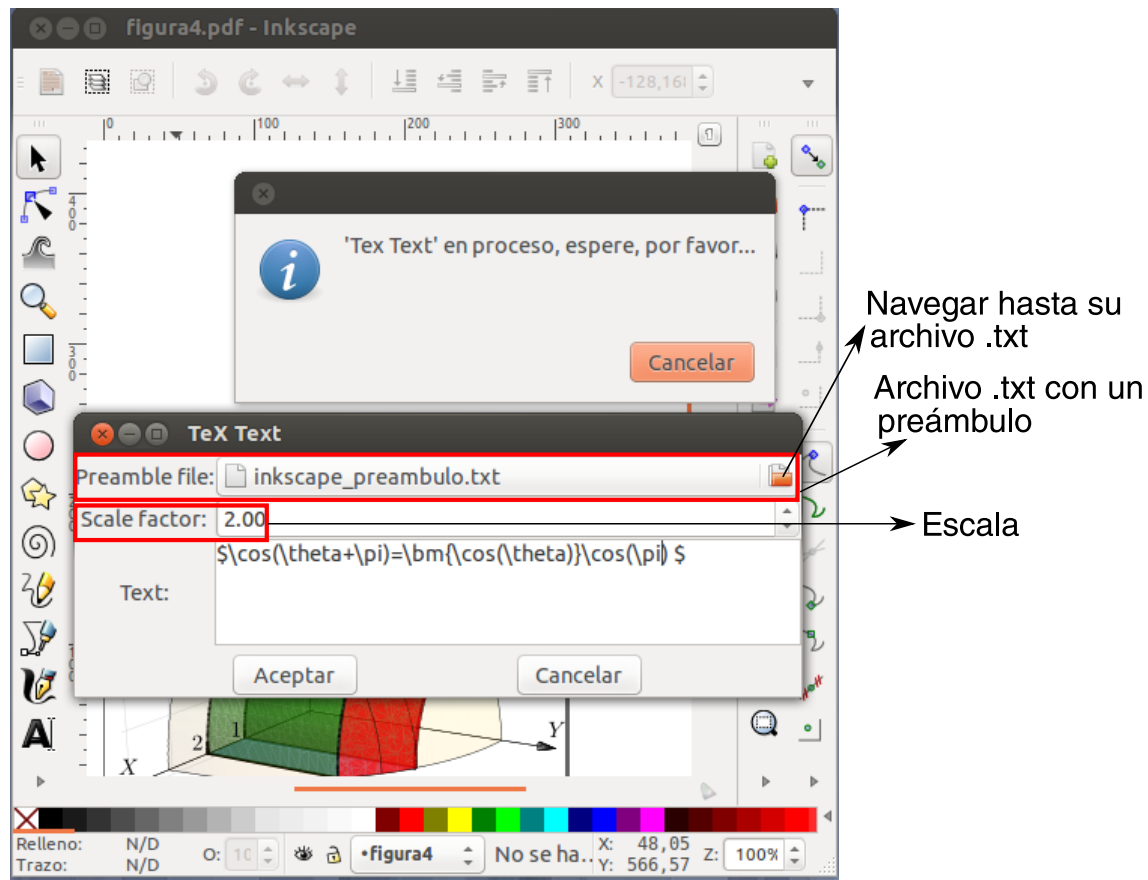


Figura 7.6. **TexText** con preámbulo y factor de escala

Un archivo para el preámbulo podría ser (lo llamamos **inkscape\_preambulo.txt**)

```
% Archivo inkscape_preambulo.txt-----
\usepackage{amsmath,amssymb,amsfonts,latexsym,stmaryrd}
\usepackage[latin1]{inputenc}
\usepackage{pslatex}
\usepackage{bm}

\newcommand{\R}{\mathbb{R}}
\newcommand{\Z}{\mathbb{Z}}
\newcommand{\Q}{\mathbb{Q}}
\newcommand{\C}{\mathbb{C}}
\newcommand{\N}{\mathbb{N}}
\newcommand{\I}{\mathbb{I}}
\newcommand{\F}{\mathbb{F}}
\newcommand{\gfrac}[2]{\displaystyle{\frac{#1}{#2}}}
\newcommand{\limite}[2]{\lim_{#1} \rightarrow #2}
```

```

\newcommand{\ds}{\displaystyle}
\newcommand{\sen}{\mathop{\rm sen}\nolimits}
\newcommand{\senh}{\mathop{\rm senh}\nolimits}
\newcommand{\arcsen}{\mathop{\rm arcsen}\nolimits}
\newcommand{\arcsec}{\mathop{\rm arcsec}\nolimits}
%-----

```

**TeXText** compila un documento **article**. En el código de la extensión se puede ver que carga tres cosas: El texto **LaTeX** que digitamos (**latex\_text**), el archivo .txt con el preámbulo (**preamble\_file**) y un factor de escala.

```

#Parte del código de la extensión TeXText.py
class PdfConverterBase(LatexConverterBase):
 def convert(self, latex_text, preamble_file, scale_factor):
 ...
\documentclass[landscape,a0]{article}
%s #inserta el preámbulo
\pagestyle{empty}
\begin{document}
\noindent
%s #inserta el código LaTeX
\end{document}
""" % (preamble, latex_text)
...

```

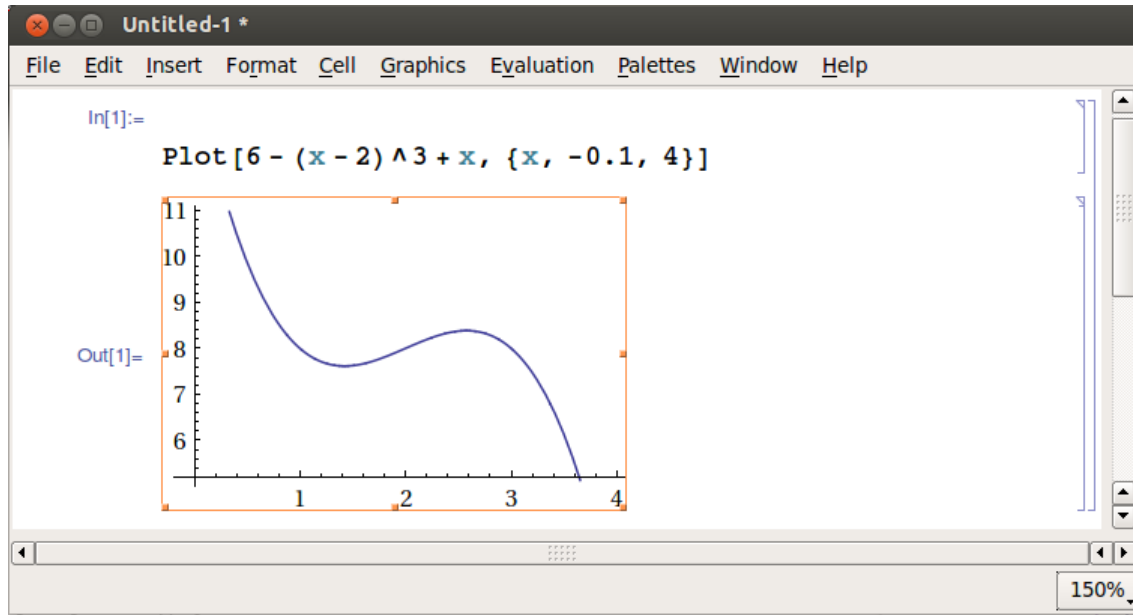
Como se observa, ya viene incluido por defecto `\begin{document}` y `\end{document}` por lo que *no se debe poner* en el archivo **.txt**

Las fórmulas se pueden reeditar seleccionándolas y llamando nuevamente a la extensión **TeX-Text** (siempre y cuando no se haya “desagrupado”).

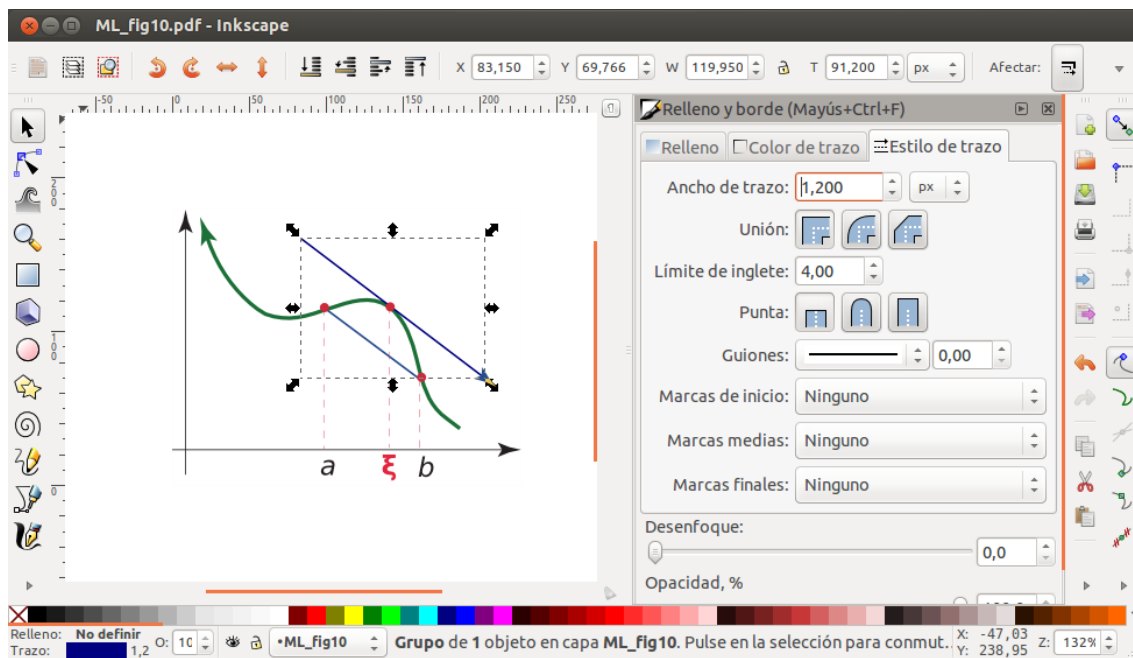
## 7.4 Edición adicional de figuras con Inkscape

En general, implementamos gráficos y figuras en algún software (**Winplot**, **Geogebra**, **Mathematica**, **MatLab**, **wxmaxima**, etc.) y guardamos en formato **.eps** o en formato **.pdf**. Luego los abrimos con **Inkscape** y los editamos y mejoramos.

Para el siguiente ejemplo, que trata sobre el teorema del valor medio para derivadas, primero implementamos la gráfica, digamos en **Mathematica** o cualquier otro software que permita graficar. Luego guardamos la figura como **.eps** o como **.pdf**, ambos formatos permiten abrir la figura con **Inkscape**, ‘desagruparla’ y mejorarla,

Figura 7.7. Gráfico en **Mathematica** (guardar como **.pdf**)

Ahora seleccionamos la figura y aplicamos **Ctrl-U** varias veces para desagrupar los componentes de la figura, luego seleccionamos las partes que nos interesan y editamos.

Figura 7.8. Abrir con **Inkscape** y editar.

### Ejemplo 7.1

La figura en este ejemplo fue implementada en **Wolfram Mathematica** y editada con **Inkscape** y guardada en formato **.pdf** (este documento se compiló con **PDFLaTeX**)

```

\begin{minipage}[b]{0.5\the}
% En el preámbulo:
%\newtheorem{teo}{Teorema} está en el preámbulo
%\usepackage{pstricks, caption}

\begin{teo}[Teorema del valor Medio]\label{tvm}
Sea $f(x)$ continua en $[a,b]$ y derivable en $]a,b[$,
entonces $\exists \xi \in]a,b[$ tal que
 $f(b)-f(a) = f'(\xi)(b-a)$.
\end{teo}

En particular, siendo $f(x)=6-(x-2)^3+x$,
 $a=2$ y $b=4$ \;;
 $\Rightarrow \xi = \frac{2}{3}(3+\sqrt{3})$.

\end{minipage} \hfill \begin{minipage}[b]{0.45\the}
\includegraphics[scale=0.7]{images/ML_fig10}.pdf
\captionof{figure}{\small Teorema del valor medio}
\label{Calculo:fig20}
\end{minipage}

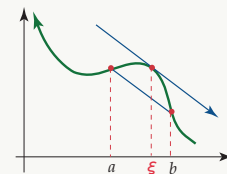
```

produce:

**Teorema 7.4.1** (Teorema del valor Medio). Sea  $f(x)$  continua en  $[a, b]$  y derivable en  $]a, b[$ , entonces  $\exists \xi \in ]a, b[$  tal que

$$f(b) - f(a) = f'(\xi)(b - a)$$

En particular, siendo  $f(x) = 6 - (x - 2)^3 + x$ ,  $a = 2$  y  $b = 4 \Rightarrow \xi = \frac{2}{3}(3 + \sqrt{3})$ .



**Figura 7.9.** Teorema del valor medio

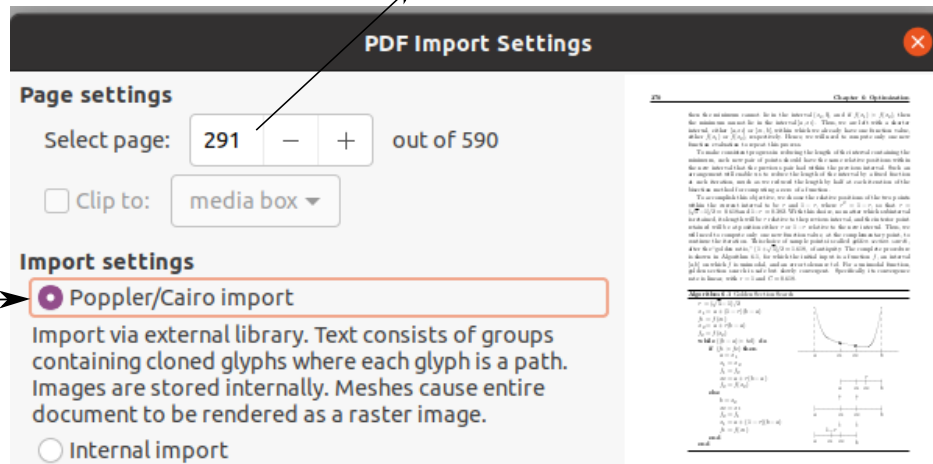
### Abrir páginas de un libro .pdf

Podemos abrir un libro .pdf con **Inkscape** y seleccionar alguna página. Esto se hace de manera natural seleccionando el libro y aplicando “Abrir con” (**Inkscape**). Para conservar el texto matemático, usamos la opción **Poppler/Cairo import**. Una vez hecho esto, la página seleccionada la podemos editar aplicando la operación de “desagrupar” (**Ctrl-U**) varias veces.

**Advertencia:** Al final, la página se debería guardar con otro nombre!

Opción para conservar  
el texto matemático

número de página



135

Figura 7.10. Abrir una página de un libro en formato .pdf

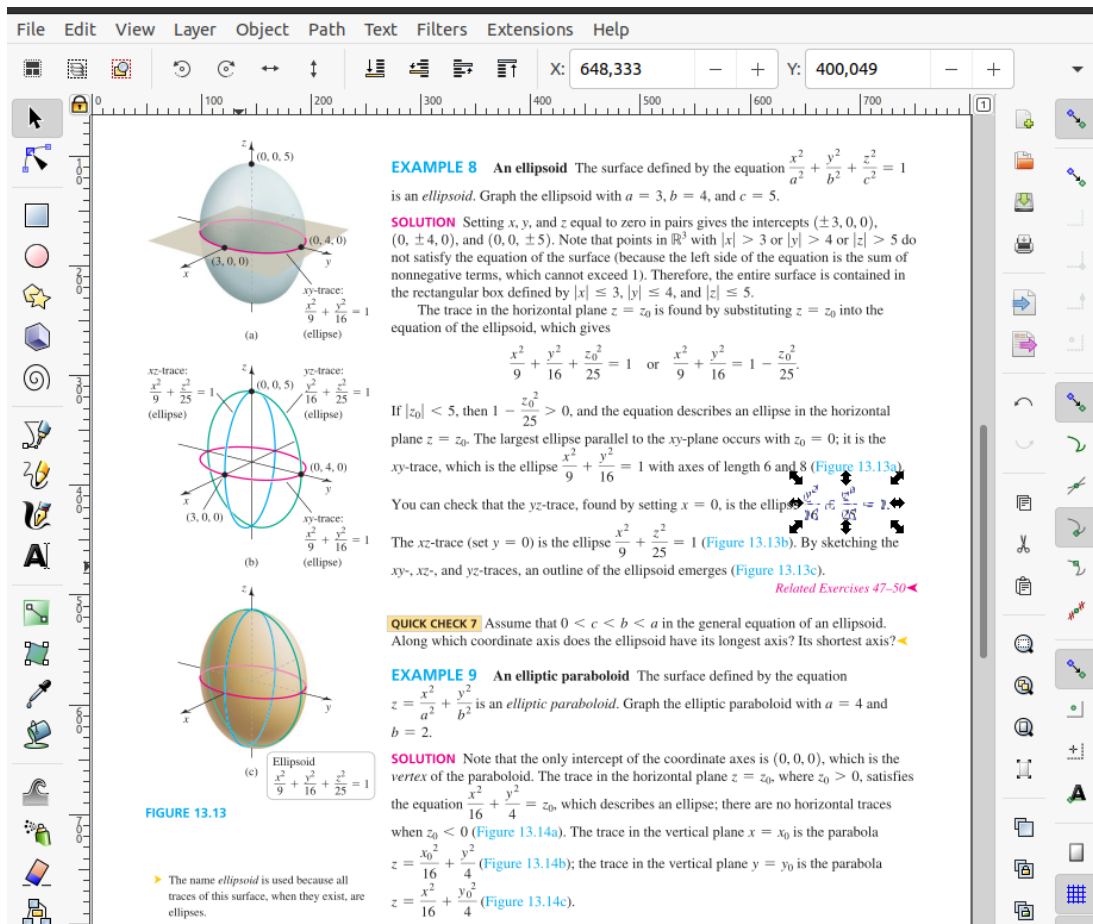


Figura 7.11. Página del libro en Inkscape con opción Poppler/Cairo import

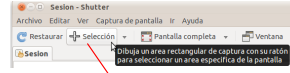
## Recortando figuras de la pantalla

Entre las opciones que tenemos para recortar figuras en la pantalla están,

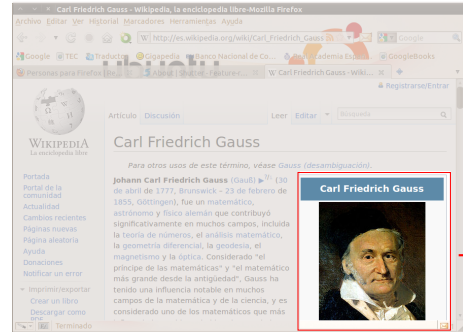
- **Shutter para Ubuntu:** Shutter es un programa de captura de pantalla con muchas características. Se puede hacer una captura de pantalla de un área específica, ventana, pantalla

completa, o incluso de un sitio web y luego aplicar diferentes efectos a la misma. Al final, exportamos la figura a **.pdf** u la guardamos en otro formato.

### Shutter



Recorte con la herramienta selección [Enter]



La selección en Shutter

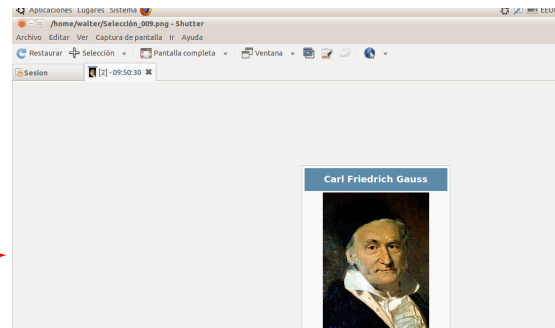


Figura 7.12. Recortando con **Shutter**

- Para **Windows** hay un dos programas de captura y edición de pantallas, similar a **Shutter**. Una es **Recortes** (Snipping Tool) y la otra es **Recortar y Anotar** (Snip & Sketch).

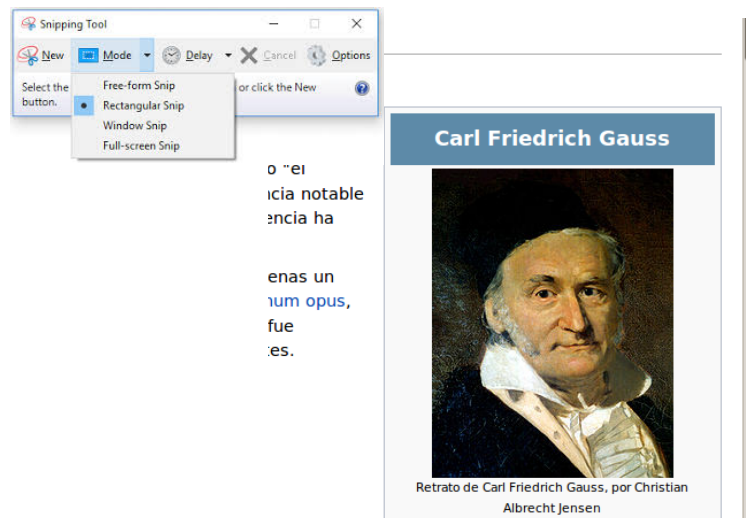


Figura 7.13. Recortando con **"Recortes"**

### Errores relacionados con "BoundingBox".

En la compilación, ya sea **LaTeX** o **PDFLaTeX**, a veces se observa el mensaje de error:

**Error: Cannot determine size of graphic (no BoundingBox)**

Esto sucede cuando una imagen no viene con las dimensiones (BoundingBox) de la caja.

La manera fácil de resolver este problema es abrir la imagen, digamos con **Inkscape** y guardar la imagen de nuevo.

Si abrimos la imagen en Adobe Illustrator, por ejemplo, para aplicar las dimensiones correctas, se debe ir a **File-Document Setup** y poner las dimensiones adecuadas para que la figura se ajuste a la región.

### Resolución de la imagen.

Los gráficos se ven bien en los formatos **.eps** o **.pdf** pero las *imágenes* (fotos, etc.) se ven mejor en formato **.png** o **.pdf**. En general, la resolución mejora si las figuras originales son lo suficientemente grandes. Cuando se gana en resolución el tamaño del PDF aumenta.





La bibliografía es una de las partes más importantes de un documento, esta permite hacer referencia a trabajos realizados anteriormente por otros autores. **LaTeX** ofrece dos formas de realizar bibliografías en un trabajo: El entorno **thebibliography** y el uso de **BibTeX**. Además hay otros sabores: **biblatex**, **biber**, **natbib**. Hay un [post](#): "bibtex vs. biber and biblatex vs. natbib" que nos puede orientar sobre estos sabores.

## 8.1 Entorno thebibliography

Para utilizar el entorno **thebibliography** se deben poner las referencias entre los comandos `\begin{thebibliography}{99} . . . \end{thebibliography}`. Cada una de las entradas de la bibliografía se pone con un comando `\bibitem{llave}`, la llave se utiliza para hacer la referencia dentro del texto.

### Ejemplo 8.1

El texto:

```
\begin{thebibliography}{99}
\bibitem{Goossens} M. Goossens; F. Mittelbach; A. Samarin.
 {\it The \LaTeX Companion}. Addison-Wesley. 1993.
\bibitem{Lamport} L. Lamport. {\it \LaTeX}. Addison-Wesley. 1996.
\end{thebibliography}
```

produce la bibliografía

### Bibliografía

- [1] M. Goossens; F. Mittelbach; A. Samarin. *The  $\text{\LaTeX}$  Companion*. Addison-Wesley. 1993.
- [2] L. Lamport.  *$\text{\LaTeX}$* . Addison-Wesley. 1996.

### Ejemplo 8.2

El texto:

En `\cite{Goossens}` y `\cite{Lamport}` se muestra como...

produce las referencias

En [1] y [2] se muestra como...

Este entorno tiene la ventaja que las referencias se escriben directamente en el documento, las desventajas son que si se tiene otro documento que hace la misma referencia entonces hay que escribir la entrada en ambos documentos y sólo tiene un único formato para presentar la bibliografía, por ejemplo, si nos piden la bibliografía siguiendo las reglas de APA entonces utilizar el entorno **thebibliography** no sería apropiado.

## 8.2 BibT<sub>E</sub>X

La segunda opción para realizar bibliografías es utilizar **BibT<sub>E</sub>X**, para este caso lo que se realiza es una “base de datos” de los libros en un archivo de texto aparte, este archivo se debe guardar en la misma carpeta del documento con extensión .bib. Este archivo se puede realizar con el Bloc de Notas en Windows o el Editor de Textos en Linux, en general funciona cualquier editor de texto plano. En el documento, donde se quiere que aparezca la bibliografía, se deben poner las instrucciones:

```
\bibliographystyle{ESTILO}
\bibliography{basededatos1[,basededatos2,...]}
```

El estilo define cómo se presentará la bibliografía, entre los estilos más populares están: plain, apalike, alpha, abbrv, unsrt. Sin embargo, hay revistas o instituciones que manejan su propio estilo, en estos casos le brindan al usuario un archivo de estilo que se copia en la carpeta del documento.

Se pueden tener varias bases de datos de bibliografía separadas, por ejemplo, se puede tener una para los libros de computación y otra para los libros de álgebra; si en algún momento se está escribiendo un artículo sobre álgebra computacional, es probable que se quiera hacer referencia a libros de ambas bases de datos, entonces en el comando `\bibliography` se ponen ambas bases.

Una de las ventajas que tiene BibT<sub>E</sub>X es que, aunque la base de datos tenga muchas referencias, en la bibliografía del documento sólo aparecen las referencias a las entradas que aparecen citadas en el texto. Si se quiere que aparezca alguna entrada aunque no se cite en el texto se agrega la línea `\nocite{Llave}` y si se quiere que todas las entradas se pongan aunque nunca se citen se debe agregar la línea `\nocite{*}`.

Cada entrada en el archivo .bib debe tener la siguiente estructura:

```
@tipo{LLave,
 propiedad1="valor1",
 propiedad2="valor2",
 ...
}
```

Donde el tipo se refiere al tipo de documento: artículo, libro, conferencia, etc. A continuación se presentan los tipos permitidos:

|            |               |             |
|------------|---------------|-------------|
| article    | incollection  | other       |
| book       | inproceedings | phdthesis   |
| booklet    | manual        | proceedings |
| conference | mastersthesis | techreport  |
| inbook     | misc          | unpublished |

La llave es la que se utiliza dentro del texto para hacer las citas con la instrucción `\cite{Llave}`.

Las propiedades se refieren a los datos que se toman en las referencias: autor, título, editorial, año, etc. Las propiedades permitidas se enuncian a continuación.

|           |              |              |
|-----------|--------------|--------------|
| address   | howpublished | number       |
| abstract  | institution  | organization |
| author    | ISBN         | pages        |
| booktitle | ISSN         | publisher    |
| chapter   | journal      | school       |
| contents  | key          | series       |
| copyright | keywords     | title        |
| crossref  | language     | url          |
| edition   | month        | volume       |
| editor    | note         | year         |

Las mismas entradas que se hicieron en la sección anterior, en BibTeX se pueden hacer en un archivo LaTeX.bib con los siguientes datos.

```
@book{Goossens,
 author="Michel Goossens and Frank Mittelbach and Alexander Samarin",
 title="The \LaTeX Companion",
 editor="Addison-Wesley",
 year="1993"
}

@book{Lamport,
 author="Leslie Lamport",
 title="\LaTeX",
 editor="Addison-Wesley",
 year="1996"
}
```

En el texto se citaría igual que en el entorno thebibliography:

En `\cite{Goossens}` y `\cite{Lamport}` se muestra como...

Y la bibliografía se mostraría con:

```
\bibliographystyle{apalike}
\bibliography{LaTeX.bib}
```

En este caso se está utilizando el estilo del APA (American Psychological Association) para citar, el resultado se muestra en la figura 8.1.

En las figuras 8.2 y 8.3 se muestran los resultados utilizando como estilo plain y alpha.

Existen muchos otros estilos con los que se puede trabajar, si se está editando un libro esta opción de cambiar de estilo es muy cómoda porque las revistas o las editoriales por lo general tienen su propio estilo y, con sólo cambiar el estilo de la bibliografía, ya el documento se adecúa.

## 8.3 Gestor de referencias bibliográficas

### 8.3.1 JabRef

Existen algunos programas que facilitan la creación y el manejo de las bases de datos que utiliza BibTeX, uno de estos programas es JabRef, entre sus principales características es que es

En [Goossens et al., 1993] y [Lamport, 1996] se muestra como...

## Bibliografía

[Goossens et al., 1993] Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley.

[Lamport, 1996] Lamport, L. (1996). *L<sup>A</sup>T<sub>E</sub>X*. Addison-Wesley.

**Figura 8.1.** Bibliografía con el estilo APA

En [1] y [2] se muestra como...

## Bibliografía

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, 1993.

[2] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X*. Addison-Wesley, 1996.

**Figura 8.2.** Bibliografía con el estilo Plain

En [GMS93] y [Lam96] se muestra como...

## Bibliografía

[GMS93] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, 1993.

[Lam96] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X*. Addison-Wesley, 1996.

**Figura 8.3.** Bibliografía con el estilo Alpha

un programa gratuito y que se puede utilizar tanto en Windows como en Linux.

Al abrir el programa se nos presenta una ventana sin nada más que el menú y las barras de

herramientas, al hacer una nueva base de datos se nos abre la base en blanco, tal como se muestra en la figura 8.4.

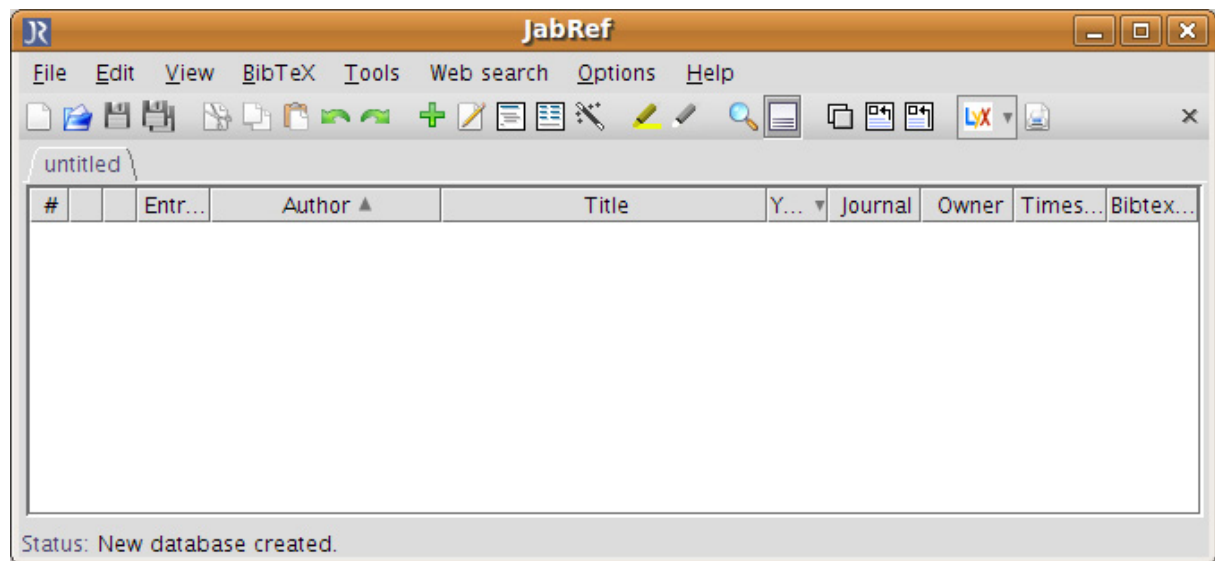


Figura 8.4. Pantalla principal de JabRef.

Para hacer una nueva entrada en la base de datos se utiliza el menú **BibTeX->New Entry** o el botón  de la barra de herramientas, a continuación sale una ventana con las opciones que hay para la referencia, tal como se muestra en la figura 8.6

Ahora aparecerá una ventana donde se pueden ir llenando los campos de la entrada, en las lengüetas de arriba se puede ir pasando entre los campos requeridos, los opcionales, los de datos generales, el abstract (resumen) y el review (revisión), la última es para ver la entrada de BibTeX en modo texto. En general se llenan los campos de los que se disponga información. Esta ventana se puede observar en la figura 8.5.

Si se abre una base de datos ya existente, aparecen las entradas arriba y la ficha bibliográfica al lado abajo de la ventana, si se quiere editar alguna de las entradas existentes se debe utilizar en el menú **BibTeX->Edit Entry** o presionar el botón  de la barra de herramientas.

Al guardar el archivo lo que hace el programa es guardarlo en modo texto con extensión **.bib** tal como se vió en la sección 8.2

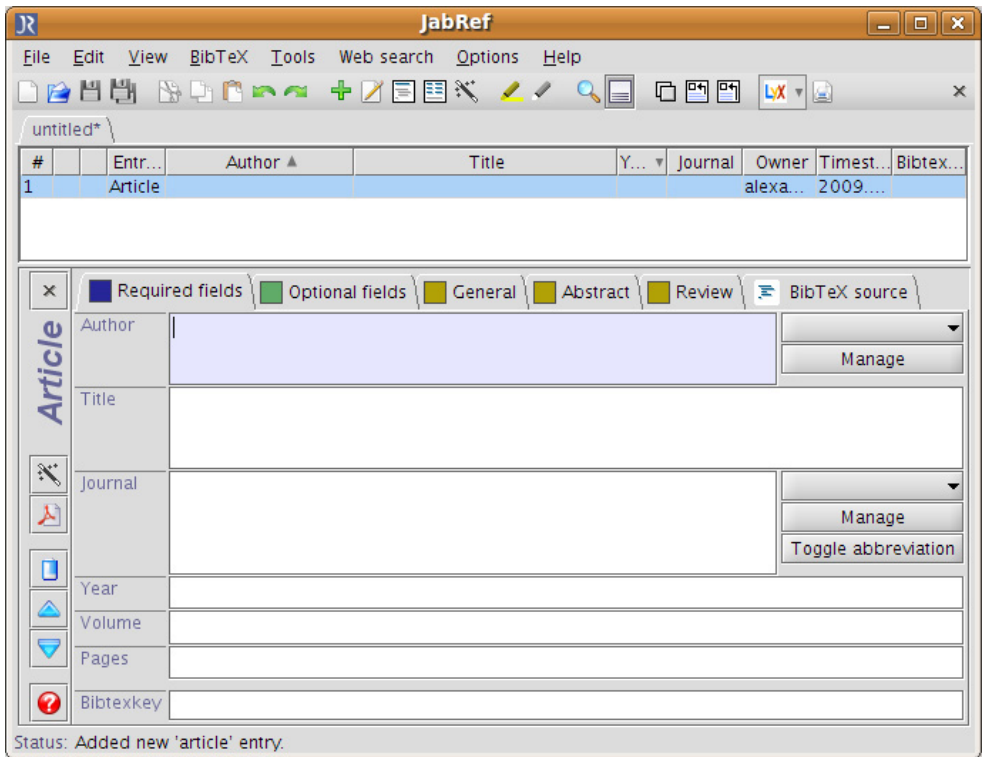


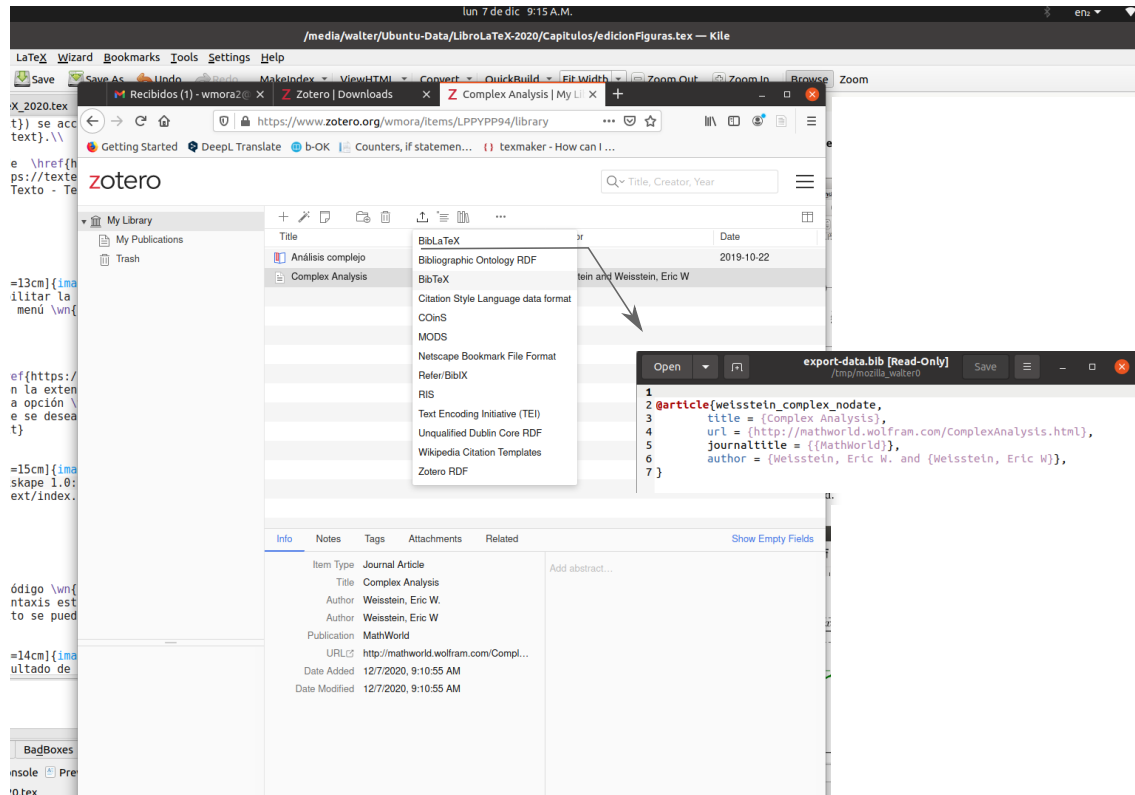
Figura 8.5. Nuevo Artículo.



Figura 8.6. Opciones para una nueva entrada en BibTeX.

### 8.3.2 Zotero

Zotero (software libre, multiplataforma) es un gestor de referencias bibliográficas, este gestor se puede integrar con el navegador web y permite encontrar los metadatos de publicaciones automáticamente (también se puede usar como aplicación). Luego se puede exportar en un formato adecuado.







Cuando escribimos un documento **LaTeX** no hay que preocuparse, en general, por el diseño del documento, **LaTeX** se encarga y aplica el diseño establecido según el “`\documentclass`”. Una vez que tenemos resuelto el contenido tenemos que fijarnos en el estilo y en el diseño editorial.

Es un hecho comprobado que una representación que funciona mejor es también percibida como más bella. La sensación de belleza viene de la facilidad de uso, es decir, lo fácil de usar generalmente es placentero. Uno no quiere leer una página mal organizada en el que no haya una jerarquía clara, queremos leer páginas placenteras que se vean claras y bien arregladas. Ahora podemos analizar si estamos comunicando de manera agradable y efectiva la información.

Nos gusta ver orden, esto crea una sensación de calma y seguridad. También les gusta ver contraste: Frecuentemente en una misma página contamos varias historias y esto puede suceder hasta en un solo párrafo o una fórmula; el contraste ayuda al lector a ver la lógica y el flujo de las ideas y a organizar la información y crea de paso un interés. Hay algunos principios que nos ayudan a definir la manera en que vamos a organizar y presentar la información, de eso se trata este capítulo.

## 9.1 Bajar la carga cognitiva

La carga cognitiva está conformada por las demandas que se imponen a la memoria de trabajo durante una observación o aprendizaje. Aquí hay dos conceptos que nos interesan, la carga cognitiva intrínseca y la carga cognitiva extrínseca.

**Carga intrínseca:** Está determinada “por la naturaleza del material y la experticia del aprendiz”

**Carga extrínseca:** La carga cognitiva extrínseca está asociada con procesos que no tienen relación directa con el aprendizaje del concepto mismo sino más bien con la lectura del material.

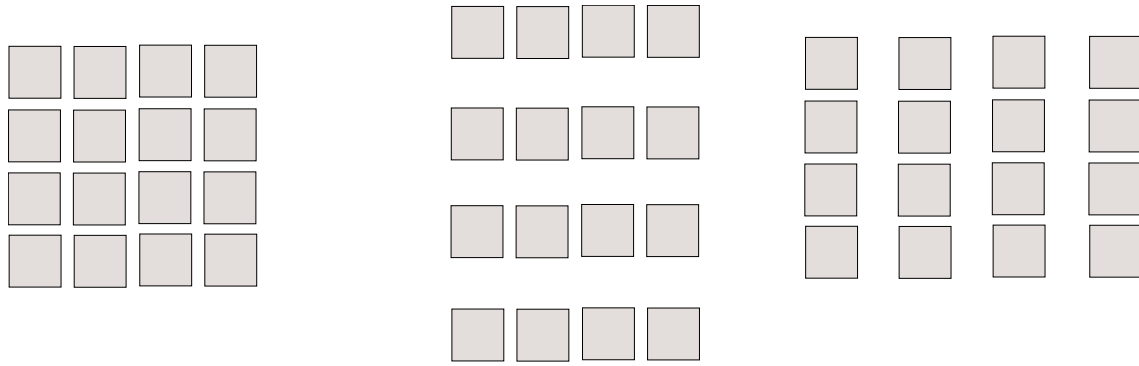
En principio no podemos hacer mucho en lo que respecta a la carga intrínseca pero sí podemos hacer algo para reducir la carga cognitiva extrínseca. El diseño editorial que nos ocupa esta orientado a reducir la carga cognitiva extrínseca presentando la información de un modo acorde con el modo en que los seres humanos percibimos el mundo.

### Las leyes Gestalt

Leyes de percepción Gestalt (‘forma’) tratan de explicar cómo el cerebro humano percibe e interpreta el mundo y forman parte del conglomerado básico de la teoría del diseño. Dos leyes nos interesan en particular, la ley de proximidad y la ley de semejanza.

**Ley de la proximidad:** El principio de proximidad dicta que los objetos más cercanos (en tiempo o espacio) se perciben como un grupo.

Por ejemplo, en la figura que sigue el arreglo de la izquierda no presenta una pertenencia clara, en el centro se acentúa la pertenencia por filas y a la derecha la pertenencia por columnas.



Ahora veamos un ejemplo en el contexto de las matemáticas: El siguiente texto presenta ambigüedad en la pertenencia,

$$\begin{aligned} \left(x + \frac{b}{2}\right)^2 - \frac{b^2}{4} + c &= \left[ x^2 + 2(x)\left(\frac{b}{2}\right) + \left(\frac{b}{2}\right)^2 \right] - \frac{b^2}{4} + c \\ &= \left[ x^2 + bx + \frac{b^2}{4} \right] - \frac{b^2}{4} + c \\ &= x^2 + bx + c \end{aligned}$$

Para mejorar la pertenencia horizontal creamos espacio,

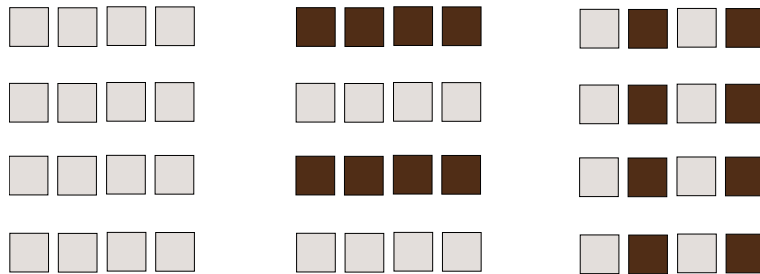
$$\begin{aligned} \left(x + \frac{b}{2}\right)^2 - \frac{b^2}{4} + c &= \left[ x^2 + 2(x)\left(\frac{b}{2}\right) + \left(\frac{b}{2}\right)^2 \right] - \frac{b^2}{4} + c \\ &= \left[ x^2 + bx + \frac{b^2}{4} \right] - \frac{b^2}{4} + c \\ &= x^2 + bx + c \end{aligned}$$

**Ley de la semejanza:** El principio de semejanza dicta que los objetos que poseen características similares de forma, color, luminosidad o tamaño aparentan pertenencia.

En la figura de abajo, el grupo de cuadrados aparenta estar dividido en dos grupos: Los oscuros y los claros.



**Proximidad versus semejanza:** Las leyes gestalt pueden reafirmarse o balancearse si se combinan. En la figura que sigue, se tiene una combinación de la ley de semejanza versus la ley de proximidad: Los cuadrados tienen una relación hacia las líneas (horizontales) más que a las columnas (verticales), en el segundo esquema esta relación se reafirma con color (ley de semejanza), sin embargo, en el tercer esquema a pesar de que la relación de proximidad no ha cambiado las columnas operan con más fuerza que las líneas, es decir la similaridad actúa por encima de la proximidad.



Como un ejemplo, en el siguiente texto se usa lo que se llama un ‘código cromático’ (provocar impacto visual mediante la manipulación del color), que permite al lector tener una clara distinción de los tipos de elementos sin ni siquiera pensar en ello.

$$\begin{aligned}
 4a^2x + 3bm - 4ab - 3max &= (4a^2x - 4ab) + (3mb - 3max) \\
 &= 4a(ax - b) + 3m(b - ax) \\
 &= 4a(ax - b) - 3m(ax - b) \\
 &= (4a - 3m)(ax - b)
 \end{aligned}$$

## 9.2 Amenidad: Los Cuatro Principios Básicos

### 9.2.1 Proximidad.

El propósito básico de la proximidad es organizar. La idea es agrupar varios items relacionados de tal manera que se visualicen como una sola unidad. Tal vez, el ejemplo más sencillo es una tarjeta de presentación. En la figura 9.1-(a), se muestra una tarjeta con varios elementos sin agrupar, en la parte (b) se muestra con los elementos agrupados según su proximidad.

### 9.2.2 Alineamiento.

El propósito del alineamiento es unificar y organizar cada página. Nada se debe colocar de manera arbitraria, los elementos deben tener conexión visual con los otros elementos.



Figura 9.1

### Alineamiento incorrecto,

Por ejemplo, para construir una tabla parcial en base  $b = 2$  módulo 13, calculamos las potencias de 2 módulo 13.

$$\begin{aligned} 2 &\equiv 2^1(\text{mod } 13), & 11 &\equiv 2^7(\text{mod } 13), \\ 4 &\equiv 2^2(\text{mod } 13), & 9 &\equiv 2^8(\text{mod } 13), \\ 8 &\equiv 2^3(\text{mod } 13), & 5 &\equiv 2^9(\text{mod } 13), \\ 3 &\equiv 2^4(\text{mod } 13), & 10 &\equiv 2^{10}(\text{mod } 13), \\ 6 &\equiv 2^5(\text{mod } 13), & 7 &\equiv 2^{11}(\text{mod } 13), \\ 12 &\equiv 2^6(\text{mod } 13), & 1 &\equiv 2^{12}(\text{mod } 13). \end{aligned}$$

Luego, ponemos la información en una tabla,

| $a$               | 1  | 2 | 3 | 4... |
|-------------------|----|---|---|------|
| $\text{Ind}_2(a)$ | 12 | 1 | 4 | 2... |

### Alineamiento correcto,

Por ejemplo, para construir una tabla parcial en base  $b = 2$  módulo 13, calculamos las potencias de 2 módulo 13.

$$\begin{aligned} 2 &\equiv 2^1(\text{mod } 13), & 11 &\equiv 2^7(\text{mod } 13), \\ 4 &\equiv 2^2(\text{mod } 13), & 9 &\equiv 2^8(\text{mod } 13), \\ 8 &\equiv 2^3(\text{mod } 13), & 5 &\equiv 2^9(\text{mod } 13), \\ 3 &\equiv 2^4(\text{mod } 13), & 10 &\equiv 2^{10}(\text{mod } 13), \\ 6 &\equiv 2^5(\text{mod } 13), & 7 &\equiv 2^{11}(\text{mod } 13), \\ 12 &\equiv 2^6(\text{mod } 13), & 1 &\equiv 2^{12}(\text{mod } 13). \end{aligned}$$

Luego, ponemos la información en una tabla,

| $a$               | 1  | 2 | 3 | 4... |
|-------------------|----|---|---|------|
| $\text{Ind}_2(a)$ | 12 | 1 | 4 | 2... |

## 9.2.3 Repetición.

La repetición es una poderosa manera de ser *consistente*. El propósito de la repetición es unificar y agregar interés visual. Los elementos repetitivos pueden ser fuentes en negrita, líneas delgadas, viñetas, encabezados, márgenes, color, fuentes, etc.

## 9.2.4 Contraste.

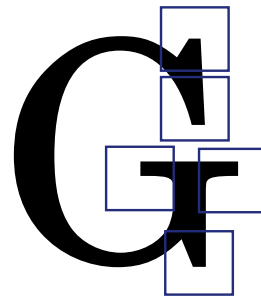
El propósito del contraste es crear interés en la página y, al mismo tiempo, ayudar en la organización. El contraste puede ser creado de varias maneras: Fuentes grandes con fuentes pequeñas, color, elementos horizontales o verticales, etc.

1. Divisibilidad por 9 : 9 divide a  $a$  si y sólo si 9 divide la suma de sus dígitos, es decir,  $9|a \iff 9|\sum_{i=0}^n a_i$
2. Divisibilidad por 3 : 3 divide a  $a$  si y sólo si 3 divide la suma de sus dígitos.
3. Divisibilidad por 2 y por 5 : tanto 2 como 5 dividen a  $a$  si y sólo si dividen  $a_0$ .

1. **Divisibilidad por 9** : 9 divide a  $a$  si y sólo si 9 divide la suma de sus dígitos, es decir,  $9|a \iff 9|\sum_{i=0}^n a_i$
2. **Divisibilidad por 3** : 3 divide a  $a$  si y sólo si 3 divide la suma de sus dígitos.
3. **Divisibilidad por 2 y por 5** : tanto 2 como 5 dividen a  $a$  si y sólo si dividen  $a_0$ .

## 9.3 Legibilidad: Cómo escoger las fuentes.

Los cuatro principios básicos de los que hablamos anteriormente son una guía para la *amenidad*. Ahora nos interesa la legibilidad. La legibilidad nos debe guiar en la selección de tipo de letra. Hay tres tipos de letra que podemos usar: **Serif** (letras con serifas o ‘remates’, como Times o Palatino), **sans serif** (letras sin serifas ‘remates’ como Helvetica o Arial) y **decorativa**. El texto **serif** se considera el más fácil de leer en los textos impresos y cansa menos la vista cuando se trata de grandes bloques, pues está diseñado para ayudar al lector a identificar y discriminar entre las letras específicas, por eso se usa en cuerpo del texto. Proporciona efecto de tranquilidad, modifica el significado del texto y añade connotaciones específicas a lo expresado. El texto **sans serif** es el segundo tipo más legible de texto y por lo general se utiliza para los títulos y subtítulos de las secciones y se aplica también a textos con fuente muy pequeña (manuales de artefactos electrónicos, medicinas, etc.) También es un texto adecuado para textos cortos en pantalla. Este tipo de fuente crea el efecto de modernidad, sobriedad, alegría y seguridad. Hay que tomar en cuenta que diferentes tipografías atraen audiencias diferentes, tanto de manera subliminal como abiertamente (piense en una tipografía para niños).



Cuando se trata de seleccionar tipos de letras para un folleto o un libro, una regla general que se menciona es la siguiente,

- ① Utilice un tipo de letra serif simple, fácilmente reconocido para el cuerpo del trabajo (Times, Times New Roman, Palatino,...).
- ② Divida el texto con títulos y subtítulos en tipo de letra sans serif (puede ser Arial o helvetica, aunque hay otras).

**LaTeX** no usa las fuentes del sistema sino que tiene sus propias fuentes. Las fuente default de **LaTeX** es **Computer Modern**, tal vez por esto sea una fuente usada en exceso. Para optimizar la calidad de impresión y también de visualización en pantalla (vía PDF), es adecuado forzar **LaTeX** para que use fuentes “postscript” (que vienen con las distribuciones actuales, por ejemplo MikTeX 2.x o TeXLive 2012). Esto se puede hacer usando algún paquete. Por ejemplo, el paquete **mathpazo** o la familia de fuentes **PSNFSS**. En la documentación de cada paquete se encuentran algunos detalles adicionales relacionados con la codificación u otra consideraciones. Algunos ejemplos son,

- El paquete **mathpazo**: La fuente default es “Palatino”.  
Agregamos en el preámbulo `\usepackage{mathpazo}`

¿Qué significa “tomar un número natural al azar”? Los naturales son un conjunto infinito, así que no tiene sentido decir que vamos a tomar un número al azar. Lo que si podemos es tomar un número de manera aleatoria en un conjunto finito  $\{1, 2, \dots, n\}$  y luego (atendiendo a la noción frecuentista de probabilidad) ver que pasa si  $n$  se hace grande (i.e.  $n \rightarrow \infty$ ).

- El paquete **mathptmx**: La fuente default es “Times”.  
Agregamos en el preámbulo `\usepackage{mathptmx}`

¿Qué significa “tomar un número natural al azar”? Los naturales son un conjunto infinito, así que no tiene sentido decir que vamos a tomar un número al azar. Lo que si podemos es tomar un número de manera aleatoria en un conjunto finito  $\{1, 2, \dots, n\}$  y luego (atendiendo a la noción frecuentista de probabilidad) ver que pasa si  $n$  se hace grande (i.e.  $n \rightarrow \infty$ ).

- El paquete **bookman**: La fuente default es “Bookman”.  
Agregamos en el preámbulo `\usepackage{bookman}`

¿Qué significa “tomar un número natural al azar”? Los naturales son un conjunto infinito, así que no tiene sentido decir que vamos a tomar un número al azar. Lo que si podemos es tomar un número de manera aleatoria en un conjunto finito  $\{1, 2, \dots, n\}$  y luego (atendiendo a la noción frecuentista de probabilidad) ver que pasa si  $n$  se hace grande (i.e.  $n \rightarrow \infty$ ).

- El paquete **newcent**: La fuente default es ‘New Century Schoolbook’.  
Agregamos en el preámbulo `\usepackage{newcent}`

¿Qué significa “tomar un número natural al azar”? Los naturales son un conjunto infinito, así que no tiene sentido decir que vamos a tomar un número al azar. Lo que si podemos es tomar un número de manera aleatoria en un conjunto finito  $\{1, 2, \dots, n\}$  y luego (atendiendo a la noción frecuentista de probabilidad) ver que pasa si  $n$  se hace grande (i.e.  $n \rightarrow \infty$ ).

- El Catálogo de Fuentes LaTeX (LaTeX Font Catalogue) tiene muestras cortas de la mayoría de las fuentes disponibles en las distribuciones TeX, con ejemplos de uso con **LaTeX**.
- En el sitio web “Stack Exchange” se pueden encontrar sugerencias de cómo combinar fuentes de texto con fuentes de tipo matemático. En particular en “[What best combination of fonts for Serif, Sans, and Mono do you recommend?](#)”

## 9.4 Color

Esta podría ser la parte más difícil del diseño. El color es una parte inseparable de nuestras vidas y es una parte de todo lo que percibimos. El color tiene un fuerte impacto en nuestras emociones y sentimientos y se puede considerar como un elemento de diseño que se puede utilizar para crear ambientes de aprendizaje mejorados.

Si vamos a usar color, lo mejor es seguir un “esquema de color”. En principio usamos un esquema de color simple: Fondo blanco con letras negras. Esquemas más avanzados involucran la combinación de varios colores. En internet podemos obtener esquemas ya hechos, como el de la figura (9.2).

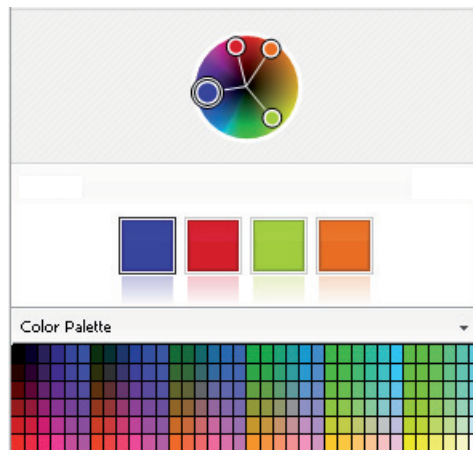


**Figura 9.2.** Esquema generado con “Color Scheme designer”

Podemos también crear esquemas personalizados basados en varios esquemas, por ejemplo

- ❶ Esquema acromático: Utiliza sólo el negro, el blanco, y los grises.
- ❷ Esquema análogo: Utiliza cualquiera de tres tonos consecutivos o cualquiera de sus tintes y matices del círculo cromático.
- ❸ Esquema complementario: Usa los opuestos directos del círculo cromático.
- ❹ Esquema complementario dividido: Consta de un tono y los dos tonos a ambos lados de su complemento.

Hay algunos sitios en Internet con esquemas de color ya hechos, como “Color Schemer Studio”, y también hay aplicaciones “online”, que nos permiten crear esquemas, como ‘Colors on the Web’ (<http://www.colorsontheweb.com>) o ‘Color Scheme designer’ (<http://colorschemedesigner.com/>)



**Figura 9.3.** Color Scheme Studio

Cuando agregamos color a una imagen, es usual observar la codificación del color (tres números) en varios modelos de color. Aquí son de interés dos modelos: RGB (acrónimo de red, green y blue) y CMYK (acrónimo de Cyan, Magenta, Yellow y Key). RGB es la elección si el material se va visualizar en pantalla y CMYK es la elección si es para impresión.

## 9.5 Editar un PDF con Inkscape

Después de generar una archivo PDF, se puede agregar detalles de diseño (color, figuras, imágenes, retoques, etc.) abriendo cada página del archivo PDF en **Inkscape**. Abrimos cada página del documento **.pdf** con **Inkscape** y para conservar el texto matemático, usamos la opción



**Poppler/Cairo import.** Una vez hecho esto, la página seleccionada la podemos editar aplicando la operación de “desagrupar” (**Ctrl-U**) varias veces.

**Advertencia:** Al final, la página se debería guardar con otro nombre para tener un respaldo.

Opción para conservar el texto matemático

número de página

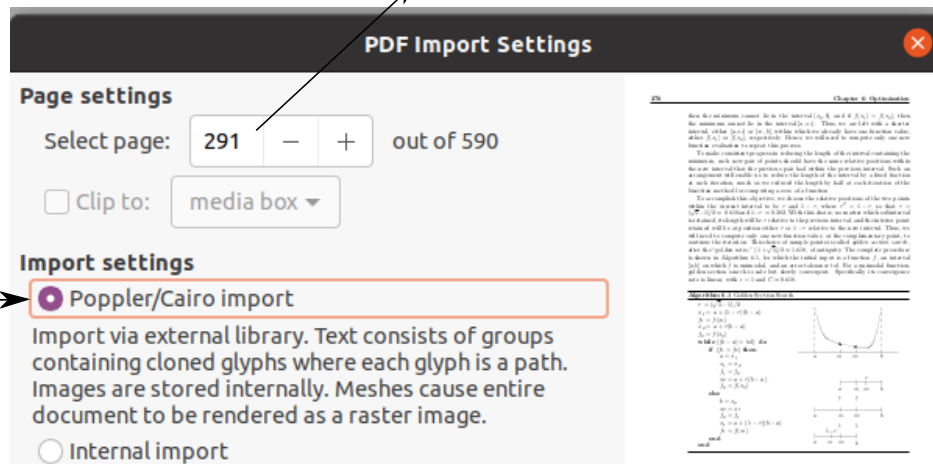


Figura 9.4. Abrir una página de un libro en formato .pdf

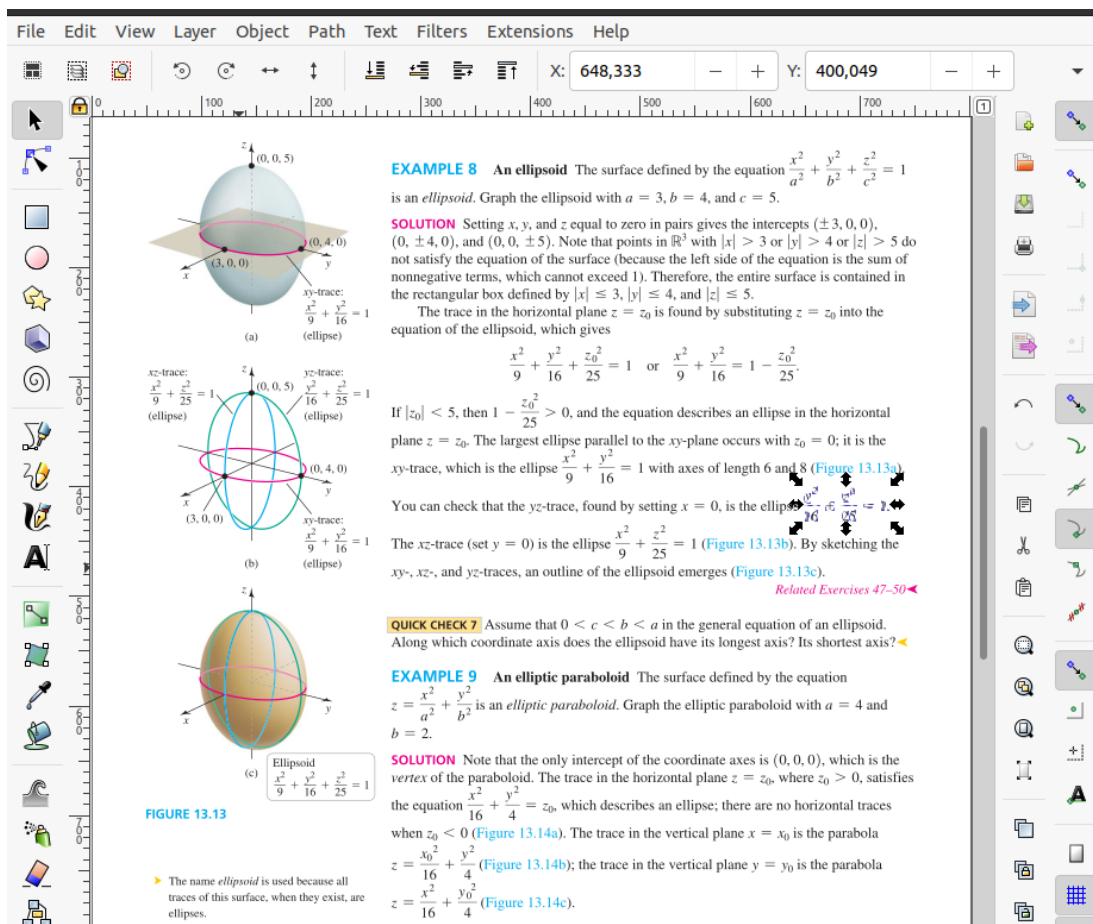


Figura 9.5. Página del libro en Inkscape con opción **Poppler/Cairo import**



Un entorno es un ambiente del tipo

```
\begin{nombre_entorno}[opciones]
...texto...
\end{nombre_entorno}
```

Podemos definir un entorno estándar, por ejemplo con el paquete **amsthm**, o podemos implementar entornos más elaborados agregando cajas a estos entornos. Por ejemplo, un entorno para teoremas se vería así,

**Teorema 10.1 (Cúbicas — Raíces reales).**

La ecuación

$$x^3 - bx + a = 0$$

tiene raíces reales si  $\frac{a^2}{4} - \frac{b^3}{27} \geq 0$ .

Las cajas de los ejemplos que usamos en este libro fueron hechas con los paquetes **Tikz** y **tcolorbox**. En realidad el segundo paquete usa el primero, pero se requieren varias librerías adicionales. Este capítulo ofrece una pequeña introducción a los paquetes **TikZ**, **tcolorbox** y **xparse**. El objetivo es implementar entornos y cajas con diseño para estos entornos.



Para este capítulo necesitamos la distribución **TeX** completa y actualizada a la fecha. Esto es necesario porque las facetas que vamos a utilizar **no** están presentes en la versiones antiguas de estos paquetes ni sus librerías.

## 10.1 Entornos simples

Para definir un entorno estándar para teoremas, definiciones, ejemplos, etc., podemos usar el paquete **amsmath**. Para definir un entorno usamos el comando

```
\newtheorem{nombre_entorno}{Encabezado}[opcional].
```

Solo debemos poner en el *preámbulo*

```
\usepackage{amsmath}
\newtheorem{unadefi}{Definición}[chapter] % Formato 3.1, 3.2
\newtheorem{unteo}{Teorema}[section] % Formato 3.1.1, 3.1.2
\newtheorem{unejem}{Ejemplo}[section]
```

Ya hora podemos usar los entornos,

El código:

```
\begin{unadefi}
 Sean p, q enteros con $q \neq 0$. Decimos que q divide a p
 si existe un entero k tal que $p=qk$.
\end{unadefi}

\begin{unteo}
 Si $a, b \in \mathbb{Z}$, y si $a|b$ y $b|a$ entonces $|a|=|b|$
\end{unteo}
```

produce:

**Definición 10.1.** Sean  $p, q$  enteros con  $q \neq 0$ . Decimos que  $q$  divide a  $p$  si existe un entero  $k$  tal que  $p = qk$ .

**Teorema 10.1.1.** Si  $a, b \in \mathbb{Z}$ , y si  $a|b$  y  $b|a$  entonces  $|a| = |b|$

## Personalización.

Podemos hacer algunos ajustes a los entornos anteriores con el comando `newtheoremstyle`. Hay varias opciones (dejar vacías las llaves indica que usamos el valor por defecto del entorno).

```
\usepackage{amsthm}
\newtheoremstyle{estiloB} % nombre del estilo
{} % espacio por encima del teorema.
{} % espacio por debajo del teorema.
{} % fuente para el cuerpo del entorno
{} % sangría
{.} % puntuacion entre la cabeza y cuerpo
{} % espacio después de encabezado
{} % Especifique manualmente el encabezado
```

Por ejemplo, podemos personalizar entornos para definición y teorema: Color azul en negrita para el encabezado y numeración en rojo y dejar una línea antes del cuerpo. Algunas cosas las podemos controlar manualmente con la última opción del comando `\newtheoremstyle`. Aquí podemos usar los comandos `\thmnumber` y `\thmnote` para controlar el color, fuente, etc., del contador del teorema y de la descripción. La numeración la podemos poner en rojo agregando `\thmnumber{\color{rojoF} #2}`. El color de la descripción del teorema (si hubiera) la controlamos con `\thmnote`. Por supuesto, podemos definir un estilo para cada entorno.

## Ejemplo 10.1 (Entornos con el paquete amsthm)

El código:



```

\documentclass{book}
\usepackage{amsthm,amsmath,amssymb,amsfonts,xcolor}
\usepackage[utf8]{inputenc}
\definecolor{azulF}{rgb}{.0,.0,.3} % Azul
\definecolor{rojoF}{RGB}{212,0,0} % Rojo
\newcommand{\Z}{\mathbb{Z}} % \Z
% Estilo B
\newtheoremstyle{estiloB}{}{}{}{}{}%
{\color{azulF}\bfseries} % fuente del encabezado
{.} % puntuación
{\newline} % espacio después del encabezado
{\thmname{#1}~\thmnumber{\color{rojoF} #2}\thmnote{~\color{azulF} (#3)}
}
%%--
\theoremstyle{estiloB}
\newtheorem{unadefiB}{Definición}[chapter]
\newtheorem{unteoB}{Teorema}[section]
%%--
% Estilo C
\newtheoremstyle{estiloC}{}{}{}{}{}%
{\color{rojoF}\bfseries}%
{.}\newline{}{}%
%%--
\swapnumbers % Intercambiar número-teorema
\theoremstyle{estiloC}
\newtheorem{unteoC}{Teorema}
%%--
\begin{document}
\begin{unadefiB}[Divisibilidad]\label{divisibilidad}
Sean a, b enteros con $b \neq 0$. Decimos que b divide a a si existe un entero c tal que $a=bc$.
\end{unadefiB}

\begin{unteoB}
Si $a, b \in \mathbb{Z}$, y si $a|b$ y $b|a$ entonces $|a|=|b|$
\end{unteoB}

\begin{unteoC}
Si $a, b \in \mathbb{Z}$, y si $a|b$ y $b|a$ entonces $|a|=|b|$
\end{unteoC}
\end{document}

```

produce:

**Definición 10.1 (Divisibilidad).**

Sean  $a, b$  enteros con  $b \neq 0$ . Decimos que  $b$  divide a  $a$  si existe un entero  $c$  tal que  $a = bc$ .

**Teorema 10.1.1.**

Si  $a, b \in \mathbb{Z}$ , y si  $a|b$  y  $b|a$  entonces  $|a| = |b|$

**1 Teorema.**

Si  $a, b \in \mathbb{Z}$ , y si  $a|b$  y  $b|a$  entonces  $|a| = |b|$

## 10.2 Entornos con xparse

Como ya dijimos, un entorno es un ambiente del tipo

```
\begin{entorno}[opciones]
 código
\end{entorno}
```

Para generar un entorno con cajas, como el que sigue

**Teorema 10.2 (Integral Gaussiana)**

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

debemos definir un nuevo entorno, en este caso un entorno llamado **teorema**, y escribir

```
\begin{teorema}[(Integral Gaussiana)]
 $$\int_{-\infty}^{\infty} \! e^{-x^2} \! , dx
 \;=\; \sqrt{\pi}$$
\end{teorema}
```

La definición de un nuevo entorno se puede hacer mediante el comando `\newenvironment`. Pero en general es mejor usar el paquete **xparse**. Para generar el entorno “teorema” que vimos antes, se requiere una combinación de los paquetes **xparse**, **tcolorbox** y **TikZ**.

Empecemos usando el paquete **xparse**. Para usar este paquete, agregamos al *preámbulo*

```
\usepackage{xparse}
```

Este paquete proporciona una interfaz de usuario fácil para especificar argumentos opcionales para entornos (en varios órdenes) incluyendo comandos y macros. Para ello se define el entorno utilizando `\NewDocumentEnvironment`, mientras que para macros regulares se utiliza `\NewDocumentCommand`. El primero tiene la siguiente sintaxis:

```
\NewDocumentEnvironment{nombre}{argumentos}{Inicio entorno}{Fin del entorno}
```

Los “argumentos” son una secuencia de especificaciones para el entorno que pueden entre-  
mezclar argumentos *opcionales* y *obligatorios*, según sea necesario, por ejemplo tenemos

**m**: un argumento obligatorio (el argumento va entre { })

**+m**: un argumento obligatorio que acepta párrafos (el argumento va entre { })

**o**: un argumento opcional (el argumento va entre [])

**O{default}**: un argumento opcional similar a **o**, pero devuelve **default** si no se da el  
argumento (el argumento va entre [])

### Ejemplo 10.2 (Implementando un entorno)

Para definir un entorno para una “definición” con contadores en rojo, una descripción y  
contenido como argumentos opcionales: usaríamos `O{}O{}`, de esta manera, si no hay  
argumento, el default es “nada”. Además algo de espacio antes y después del entorno,  
ponemos en el *preámbulo*

```
% En el preámbulo
\usepackage{xparse,xcolor}
\newcounter{midefi}[chapter] % Contador 1.1, 1.2,... en rojo
\renewcommand{\themidefi}{\color{red}\thechapter.\arabic{midefi}}

\NewDocumentEnvironment{midefinicion}{O{}O{}}{%2 argumentos opcionales
 %Inicio del entorno
 \bigskip
 \begin{minipage}{\textwidth}
 % Definición - número - descripción (opcional)
 \textbf{Definición \; \themidefi \; \small\sffamily #1}}

 {% Parámetro #2: Contenido del entorno
 #2
 \end{minipage}
 \bigskip
 }
}
```

Luego podemos usar el entorno en el cuerpo del documento, el **código**:

```
\documentclass{book}
\usepackage{amsthm,amsmath,amssymb,amsfonts,xcolor}
\usepackage[utf8]{inputenc}
\newcommand{\Z}{\mathbb{Z}}

\usepackage{xparse}
\newcounter{midefi}[chapter] % Contador 1.1, 1.2,... en rojo
```

```

\renewcommand{\themidefi}{\color{red}\thechapter.\arabic{midefi}}
\NewDocumentEnvironment{midefinicion}{0}{0}{\%2 argumentos opcionales
 %Inicio del entorno
 \bigskip
 \begin{minipage}{\textwidth}
 % Definición - número - descripción (opcional)
 \textbf{Definición \; \themidefi \; \small\sffamily #1}}

 }{\% Parámetro #2: Contenido del entorno
 #2
 \end{minipage}
 \bigskip
 }%

\begin{document}
 \begin{midefinicion}[(Principio de inducción).]
 Si P es una fórmula en la aritmética de Peano, entonces
 $(P(0) \wedge (\forall n)(P(n) \rightarrow P(n+1)) \rightarrow (\forall n)(P(n)))$
 En particular, P puede ser la fórmula $(m < n) \rightarrow Q(m)$ y a este
 caso
 algunas personas le llaman "Inducción fuerte".
 \end{midefinicion}

\end{document}

```

produce:

**Definición 10.0 (Principio de inducción).**

Si  $P$  es una fórmula en la aritmética de Peano, entonces

$$(P(0) \wedge (\forall n)(P(n) \rightarrow P(n+1)) \rightarrow (\forall n)(P(n)))$$

En particular,  $P$  puede ser la fórmula  $(m < n) \rightarrow Q(m)$  y a este caso algunas personas le llaman "Inducción fuerte".

## Comandos con opciones con xparse

Podemos usar `\NewDocumentCommand` para crear comandos con (o sin) opciones de una manera natural. La sintaxis es

```
\NewDocumentCommand{\nombre}{\código} %Sin parámetros
```

```
\NewDocumentCommand{\nombre}{parámetros}{código}
```

Por ejemplo, veamos el código de un comando para simplificar el ambiente `minipage` en dos columnas. Por default, usamos `0.45\textwidth` para cada columna.

```
%%- En el preámbulo -----
```



```
\usepackage{xparse}
\NewDocumentCommand{\minpage}{ 0{0.45} +m 0{0.45} +m }{%
 \begin{minipage}{#1\textwidth}
 #2
 \end{minipage}%
 \hfill \begin{minipage}{#3\textwidth}
 #4
 \end{minipage}%
}
```

Ahora ya podemos usar el entorno como\

```
\mimpage{...texto 1...}{...texto... }
\mimpage[0.3]{...texto 1...}{...texto...}% 0.3\textwidth 1ra columna
\mimpage{...texto 1...}[0.4]{...texto...}% 0.4\textwidth 2da columna
\mimpage[0.3]{...texto 1...}[0.4]{...texto... }
```

## 10.3 Crear figuras nativas con TikZ

Como decíamos antes, las cajas para los entornos las haremos con los paquetes **xparse**, **TikZ** y **tcolorbox**. El orden lógico requiere seguir con el paquete **TikZ**.

El ambiente **picture** de **LaTeX** es un ambiente especial para insertar figuras implementadas con comandos relativamente simples. Las figuras generadas en el ambiente **picture** de **LaTeX** quedan insertadas de manera automática en el documento. Programar los gráficos permite tener un control absoluto y preciso sobre todos los detalles, realizar gráficos sencillos es también muy rápido. Por otra parte, hay nuevo lenguaje que aprender, no tiene una interfaz gráfica y el código (por más sencillo que sea) no permite ver inmediatamente como se verá finalmente el gráfico.

Existen varios editores que permiten hacer figuras y generan el código **LaTeX**, listo para introducirlo en nuestro documento, por ejemplo **Inkscape**.

### Paquete Tikz

Este es un paquete para crear gráficos para documentos **LaTeX** usando un ambiente '**tikzpicture**' y comandos especiales para dibujar líneas, curvas, rectángulos, etc. Muy adecuado para trabajar con presentaciones Beamer.

La documentación la puede ver en

<http://mirrors.ucr.ac.cr/CTAN/graphics/pgf/base/doc/pgfmanual.pdf>

Si desea hacer un documento PDF, tenga en cuenta que los gráficos permanecen si compila con **PDFLaTeX** (se puede usar con Beamer) no así con **dvi**  $\rightarrow$  **pdf**.

Para usar el paquete se debe poner en el *preámbulo*

```
\usepackage{tikz}
```

La versatilidad de este paquete le permite crear gráficos hasta en el mismo texto usando el comando `\tikz`. Por ejemplo, podemos crear un círculo anaranjado como este: ● con el código

```
...como este: \tikz \fill[orange] (1ex,1ex) circle (1ex); con...
```

Aquí el “;” es necesario.

Para crear figuras complejas podemos usar el ambiente “**tikzpicture**”. Este ambiente crea una caja rectangular invisible con una coordenada inferior izquierda ajustada al dibujo. Por defecto la unidad de medida es el centímetro.

```
\begin{tikzpicture}[opciones]
 comandos tikz...
\end{tikzpicture}
```

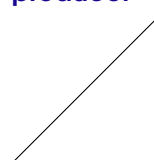
### 10.3.1 Comando draw

El comando **draw[opciones]** se usa para dibujar segmentos usando coordenadas.

El código:

```
\begin{tikzpicture}
 \draw (0,0) -- (2,2); % segmento
\end{tikzpicture}
```

produce:

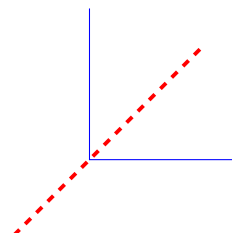


Entre las opciones que tenemos para **draw** están: Grosor con **[line width=xcm]** o con **thin**, **thick** **ultra thick** color, punteados con **[dashed]** o **[dotted]**

El código:

```
\begin{tikzpicture} % ejes
 \draw[thin, blue] (0,2) -- (0,0) -- (2,0);
 % segmento
 \draw[line width=0.051cm, red, dashed]
 (-1,-1) -- (1.5,1.5);
\end{tikzpicture}
```

produce:



### Representación gráfica de una función con TikZ

Para realizar el gráfico de una función usamos el comando **plot**. La variable  $x$  se escribe en el código como  $\backslash x$  y el dominio  $[a, b]$  se especifica con la opción **domain=a:b** del comando **draw**.

Para poner etiquetas se usa `node[]`.

Por ejemplo, para realizar el gráfico de la función  $y = 3x + 1$  con  $x \in [-2, 2]$ ; se debería poner en el código

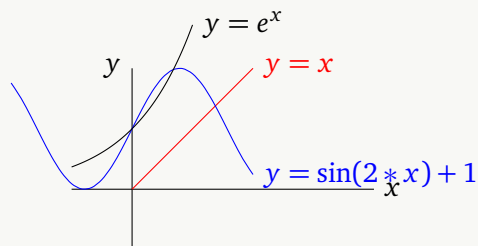
```
\draw[domain = -2:2] plot (\x,{3*\x+1}) node[right] {$y = 3x+1$};
```

### Ejemplo 10.3

El código:

```
\begin{tikzpicture}[scale=0.8] % Escalamiento de la figura 80%
\draw (-1,0) -- (4,0) node[right] {x}; % Ejes
\draw (0,-1) -- (0, 2) node[left] {y};
% Dominio: domain = a:b
\draw[smooth, domain = 0:2, color=red] plot (\x,\x) node[right] {$y = x$}
};
% \x r indica que x se mide en radianes
\draw[smooth, domain = -2:2, color=blue] plot (\x,{sin(2*\x r)+1})
node[right] {$y = \sin(2*x)+1$};
\draw[smooth, domain = -1:1, color=black] plot (\x,{exp(\x)}) node[
right] {$y = e^x$};
\end{tikzpicture}
```

produce:



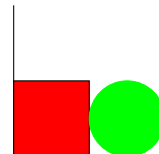
### Rectángulos y círculos con draw

Podemos usar `draw` y sus opciones para dibujar rectángulos y círculos.

El código:

```
\begin{tikzpicture} % ejes
\draw[thin] (0,2) -- (0,0) -- (2,0);
% rectángulo borde negro, relleno rojo, esquina (0,0)
\draw[black, fill=red] (0,0) rectangle (1,1);
% círculo verde, centro =(1.5,.5) y radio 0.5cm
\draw[green, fill=green] (1.5,0.5) circle [radius
=0.5];
\end{tikzpicture}
```

produce:



### 10.3.2 Nodos

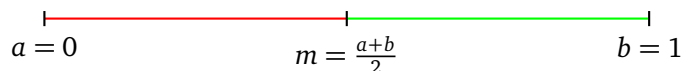
Para agregar texto (u otros objetos) a las figuras **TikZ** usamos nodos. Los nodos usualmente son rectángulos o círculos (u otra figura) con algo en su interior. Existen variadas opciones para su posicionamiento. En general, esperamos algo como

```
node[opciones]...{texto}
```

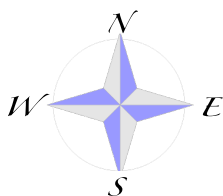
Por ejemplo, el **código**:

```
\begin{tikzpicture}[xscale=0.8] % Escalamiento 80%
 % Segmento de 0 a 1
 \draw[-][draw=red, thick] (0,0) -- (.5,0);
 \draw[-][draw=green, thick] (.5,0) -- (1,0);
 % Texto debajo ("below") del segmento
 \draw [thick] (0,-.1) node[below]{$a=0$} -- (0,0.1);
 \draw [thick] (0.5,-.1) node[below]{$m=\frac{a+b}{2}$} -- (0.5,0.1);
 \draw [thick] (1,-.1) node[below]{$b=1$} -- (1,0.1);
\end{tikzpicture}
```

produce:



#### Posicionar nodos usando “anclas”



Cuando posicionamos un nodo en *alguna coordenada*, el nodo es centrado en esta coordenada por defecto, es decir tiene “anclaje” en su centro. Hay otras maneras de posicionar los nodos. Por ejemplo, usando “anclas” (“anchors”). Cuando declaramos la opción **anchor=** punto cardinal, el punto de anclaje (*la coordenada*) se mueve al punto cardinal.

Por ejemplo, podemos poner un nodo en el punto (1,1) y otro nodo con un texto en este mismo punto pero anclado en el sur, es decir, la coordenada **(1,1)** quedará en el sur de la caja del nodo.

El **código**:

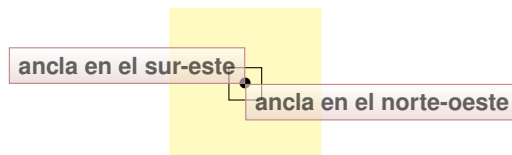
```
\tikzset{ % Lista de opciones de nodo
 % opción draw= color borde
 % opacity = porcentaje transparencia
 % opción fill = color relleno
 opciones/.style={%
 draw= red!50!black!50,
 top color = white,
 bottom color = red!50!black!20,
```

```

 fill opacity = 0.6,
 rectangle, font=\small\bf\sffamily}
}
\begin{center}
\begin{tikzpicture}
 \draw[draw=yellow!30,fill=yellow!30] (0,0) rectangle(2,2);
 \node[draw=black,rectangle, anchor=center] at (1,1) {\bullet};
 \node[opciones, anchor=south east] at (1,1) {ancla en el sur-este};
 \node[opciones, anchor=north west] at (1,1) {ancla en el norte-oeste};
\end{tikzpicture}
\captionof{figure}{Nodos rectangular con anclaje en el sureste y noroeste}
\end{center}

```

produce:



**Figura 10.1.** Nodos rectangular con anclaje en el sureste y noroeste

También podemos anclar un nodo en alguna posición respecto a la página actual,

### Usando librerías

Hay librerías **TikZ** especiales para hacer muchas cosas. Más adelante vamos a usar algunas de estas librerías. Por ahora, vamos a usar una par de librerías de ejemplo.

#### Ejemplo 10.4 (Usando las librerías “matrix” y “backgrounds”)

Vamos a usar las librerías **matrix** y **backgrounds** para colorear submatrices y entradas en una matriz  $M$ .

El código:

```

\documentclass{book}
\usepackage{amsthm,amsmath,amssymb,amsfonts,pstricks,xparse}
\usepackage[utf8]{inputenc}
\newcommand{\Z}{\mathbb{Z}}
\usepackage{tikz}
\usetikzlibrary{matrix,backgrounds}
\pgfdeclarelayer{wfondo}
\pgfsetlayers{wfondo,background,main}
\NewDocumentCommand{\iluminar}{O{blue!40} m m}{% rectángulo
\draw[#1, fill=#1] (#2.north west) rectangle(#3.south east);}

\begin{document}

```

```

\begin{center}
\raisebox{1cm}{\mathrel{=};}
\begin{tikzpicture}
\matrix (m)[matrix of math nodes,left delimiter=(,right delimiter=)]
{
1 & 2 & -1 & 2 \\
0 & 1 & 3 & 2 \\
0 & 0 & 1 & 0 \\
0 & 3 & 0 & 5 \\
}; % punto y coma!
% Iluminar de submatriz y elementos
\begin{pgfonlayer}{wfon}
% Iluminar submatriz desde m_{2,2} hasta m_{4,4}
\illuminate[green!30]{m-2-2}{m-4-4}
% Iluminar elementos de una lista
\foreach \element in {m-4-1,m-3-2,m-2-3,m-1-4}{
\illuminate[violet!30]{\element}{\element} }
% Iluminar elemento m_{1,1}
\illuminate[gray!30]{m-1-1}{m-1-1}
\end{pgfonlayer}
\end{tikzpicture}
\end{center}
\end{document}

```

produce:

$$M = \begin{pmatrix} 1 & 2 & -1 & 2 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 5 \end{pmatrix}$$

### 10.3.3 Nodos y flechas.

A veces es buena idea usar flechas para relacionar objetos en una página. Usando el paquete `tikz` y la librería `arrows` podemos crear un *marcador* para llevar una flecha de un nodo con un objeto, identificado con algún nombre, hasta otro nodo con otro objeto, identificado con otro nombre. El proceso puede involucrar varios nodos, por supuesto. El comando que creamos es `\marcar`. Tiene tres argumentos, `\marcar{color}{identificador}{objeto}`. Este comando marca objetos, luego con un entorno `tikzpicture` creamos la flecha, con ángulo de salida `outer` y ángulo de llegada `in`.

Por ejemplo:  $\textcircled{a}x^2 + bx + c = \textcircled{a}(x - x_1)(x - x_2)$ , se obtiene con el código (completo)

```

\documentclass{article}
\usepackage{amsmath}
\usepackage{tikz}

```

```

\usetikzlibrary{arrows}
%\marcar{color}{identificador}{objeto}
\newcommand{\marcar}[3]{%
 \tikz[overlay,remember picture,baseline=-2pt] \node[circle,#1,draw,
 text=black, inner sep=1pt] (#2) { #3};}

\begin{document}
 Por ejemplo:\;\;\;
 \marcar{red}{nodo1}{\mathit{a}} \mathit{x}^2 + \mathit{b}\mathit{x} + \mathit{c} \;\;=\;\;
 \marcar{red}{nodo2}{\mathit{a}} \mathit{x}(\mathit{x}-\mathit{x}_1)(\mathit{x}-\mathit{x}_2)\mathit{x}.

 %--Flechas-----
 \begin{tikzpicture}[overlay, remember picture]
 % color ángulos
 \draw[arrows=->,red] (nodo1) to [out=50, in=90](nodo2);
 \end{tikzpicture}
 %-----
\end{document}

```

**Ejemplo 10.5**

En este ejemplo vamos a “pintar” entradas  $ij$  de una matriz con un comando `\pintar`, y también vamos a agregar nodos y flechas para comentar alguna entrada de la matriz marcándolas con el comando `\tikznode`. Este ejemplo requiere usar `\usetikzlibrary{babel}` para eliminar conflictos con el paquete **babel**. Un ejemplo más detallado lo puede encontrar en [31]

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

Fila 1, Columna 1

Fila  $m$ , Columna  $n$

El código completo es

```

\documentclass{article}
\usepackage{tikz,amsmath,colortbl}

%% -----
\usetikzlibrary{babel} %--Necesario para evitar conflictos con babel--
%\pintar
\newcommand\pintar{\cellcolor{clight2}}
% marcador \tikznode
\definecolor{clight2}{RGB}{212, 237, 244}%
\newcommand\tikznode[3][]%
 {\tikz[remember picture,baseline=(#2.base)]

```

```

\newnode[minimum size=0pt,inner sep=0pt,#1](#2){#3};%
}
\tikzset{>=stealth}
%% -----

\begin{document}
\left[
\begin{array}{rrrrr}
\pintar \tikznode{pp1}{a_{11}} & a_{12} & a_{13} & \cdots & a_{1n} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
a_{m1} & a_{m2} & a_{m3} & \cdots & \pintar \tikznode{pc1}{a_{mn}}
\end{array}
\right]

\begin{tikzpicture}[remember picture,overlay,cyan,rounded corners]
\draw[<-, shorten < =1pt] (pp1)
-- +(0.6,0) % Línea a la derecha
|- +(6,0.4) % Segmento curvo para la flecha
coordinate (pp) % Recordar posición
node[right] {Fila 1, Columna 1};
\draw[<-,shorten <= 2pt] (pc1)
-- +(0,-0.5) % Línea abajo
coordinate (pc1') % Recordar posición
-- (pc1'-|pp) %
node[right] {Fila m, Columna n};
\end{tikzpicture}

\end{document}

```

## 10.4 Cajas con el paquete tcolorbox

Con el paquete **tcolorbox** podemos crear entornos con cajas. Una de las facetas más útiles es la posibilidad de controlar automáticamente la división de las cajas cuando se pasa de una página a otra o cuando la caja se extiende entre varias páginas.

Para usar este paquete debemos tener actualizada la distribución **TeX**. Vamos a usar la versión actual: **4.42** de **2020**.

Para cargar el paquete (y eventualmente algunas librerías que se usan en este libro) se agrega en el *preámbulo*

```

\usepackage{tikz,tkz-tab}%
\usetikzlibrary{matrix,arrows, positioning,shadows,shadings,backgrounds,
 calc, shapes, tikzmark}
\usepackage{tcolorbox, empheq}%

```



```
\tcbuselibrary{skins,breakable,listings,theorems}
```

## Cajas básicas con “tcolorbox”

Una caja básica usa el ambiente **tcolorbox**

```
\begin{tcolorbox}[opciones]
 código...
\end{tcolorbox}
```

Por ejemplo, el **código**:

```
\begin{tcolorbox}[colback=gray!5!white, % Color del fondo
 colframe=black!75!white, % Color borde
 title=Identities in number theory]
Identities:
\begin{minipage}{7cm}

\begin{enumerate}[font=\sffamily\bfseries,label=I.\arabic*), series=Is]
\item $11^2=121$
\item $(1+1)^2=1+2+1$
\item $12^2=144$
\item $(1+2)^2=1+4+4$
\end{enumerate}

\end{minipage} \hfill \begin{minipage}{7cm}

\begin{enumerate}[label=I.\arabic*), resume*=Is]
\item $3^3 + 4^4 + 3^3 + 5^5 = 3435$
\item $2^4 = 4^2$
\item $2592=2^59^2$
\item $10^2+11^2+12^2=13^2+14^2$
\end{enumerate}

\end{minipage}
\end{tcolorbox}%
```

produce:

### Identities in number theory

Identities:

I.1)  $11^2 = 121$

I.5)  $3^3 + 4^4 + 3^3 + 5^5 = 3435$

I.2)  $(1+1)^2 = 1+2+1$

I.6)  $2^4 = 4^2$

I.3)  $12^2 = 144$

I.7)  $2592 = 2^59^2$

I.4)  $(1+2)^2 = 1+4+4$

I.8)  $10^2 + 11^2 + 12^2 = 13^2 + 14^2$

## Opciones para títulos de caja

El paquete **tcolorbox** describe en su documentación, entre cientos de cosas, varias opciones para implementar el título de una caja (sección 7.2 de la documentación). Los dos ejemplos que siguen son “entornos” y deben usar el comando `\newtcolorbox`, por eso el código está al final de esta sección.

### Identities

$$\text{I.1)} \quad 11^2 = 121$$

$$\text{I.2)} \quad (1 + 1)^2 = 1 + 2 + 1$$

$$\text{I.3)} \quad 12^2 = 144$$

$$\text{I.4)} \quad (1 + 2)^2 = 1 + 4 + 4$$

$$\text{I.5)} \quad 3^3 + 4^4 + 3^3 + 5^5 = 3435$$

$$\text{I.6)} \quad 2^4 = 4^2$$

$$\text{I.7)} \quad 2592 = 2^5 9^2$$

$$\text{I.8)} \quad 10^2 + 11^2 + 12^2 = 13^2 + 14^2$$

$$\text{I.1)} \quad 11^2 = 121$$

$$\text{I.2)} \quad (1 + 1)^2 = 1 + 2 + 1$$

$$\text{I.3)} \quad 12^2 = 144$$

$$\text{I.4)} \quad (1 + 2)^2 = 1 + 4 + 4$$

$$\text{I.5)} \quad 3^3 + 4^4 + 3^3 + 5^5 = 3435$$

$$\text{I.6)} \quad 2^4 = 4^2$$

$$\text{I.7)} \quad 2592 = 2^5 9^2$$

$$\text{I.8)} \quad 10^2 + 11^2 + 12^2 = 13^2 + 14^2$$

### Identities

## Cajas con “tcbbox”

**tcbbox** crea una caja ajustada al ancho de su contenido. Se puede usar la mayoría de las opciones de las cajas **tcolorbox**. La sintaxis es

```
\tcbbox[opciones]{ contenido }
```

Las opciones se pueden declarar, en el *preámbulo* o antes del código, usando

```
\tcbset{NombreOpciones/.style={ op1,op2,...,opn}}
```

de tal manera que en el código solo usemos

```
\tcbbox[NombreOpciones]{ contenido }
```

y aún así, podemos agregar más opciones usando comas.

**Ejemplo 10.6** (Cajas “tcbbox” en el texto)

`\tcbbox` se puede usar en texto corriente, pero hay que ajustar la caja con varias opciones: La opción `|tcbbox raise base|` centra la caja con el texto, la opción **nobeforeafter** elimina el espacio antes y después de la caja mientras que la opción **extrude by=x** expande la caja hacia arriba y hacia abajo o la contrae si  $x < 0$ ; en este caso, la usamos para “adelgazar” la caja.

El código:

```
El discriminante del polinomio ax^2+bx+c es
\tcbbox[tcbbox raise base, nobeforeafter,
extrude by=-2mm]{\(\Delta = b^2 - 4ac\)}\}
```

produce:

El discriminante del polinomio  $ax^2 + bx + c$  es  $\Delta = b^2 - 4ac$

Por supuesto, en vez de estar escribiendo estas opciones, se pueden declarar antes (en el *preámbulo*)

```
\tcbset{opteqA/.style={%
\tcbbox raise base,
nobeforeafter,
extrude by=-2mm,
colback=red!50!black!20,
colframe=red!50!black!20}}
```

De esta manera escribimos `\tcbbox[opteqA]{\(\Delta = b^2 - 4ac\)}` para obtener  $\Delta = b^2 - 4ac$

**Ejemplo 10.7** (Cajas “tcbboxmath” para ecuaciones)

`\tcbboxmath` se puede usar en ambientes de ecuaciones para resaltar partes de una ecuación. Como antes, primero indicamos las opciones.

El código:

```
%\newcommand{\dpr}[2]{\dfrac{\partial #1}{\partial #2}}
\tcbset{opteqB/.style={% opteqB = Opciones habilitadas
nobeforeafter,
extrude by=-2mm,
colback=red!50!black!20,
colframe=red!50!black!20}}
\begin{eqnarray}%
```

```

z_t &= & \tcbboxmath[opteqB]{\dpr{z}{x}}x'(t)
 +\tcbboxmath[opteqB]{\dpr{z}{y}}y'(t)
\end{eqnarray}%

\tcbset{opteqC/.style={% opteqC = Opciones habilitadas
 skin=beamer,
 extrude by=-2mm,
 nobeforeafter,
 colback=blue!60!green!10!white,
 colframe=LightBlue4!50!black}}
\begin{eqnarray}%
z_x &= & \tcbboxmath[opteqC]{\dpr{z}{u}}u_x
 +\tcbboxmath[opteqC]{\dpr{z}{v}}v_x
\end{eqnarray}%

```

produce:

$$z_t = \frac{\partial z}{\partial x} x'(t) + \frac{\partial z}{\partial y} y'(t) \quad (10.1)$$

$$z_x = \frac{\partial z}{\partial u} u_x + \frac{\partial z}{\partial v} v_x \quad (10.2)$$

#### Ejemplo 10.8 (Cajas para arreglos)

Aquí tenemos dos opciones: Usar directamente `\tcolorbox` y crear una caja que contenga el arreglo o, usar el paquete `empheq` y hacer una caja más ajustada.

Primera opción: Usar `\tcolorbox`. El código:

```

\tcbset{opteqC/.style={skin=beamer,colback=red!10!white}}

\begin{tcolorbox}[opteqC]
\begin{eqnarray}
A &=& \dfrac{\partial u}{\partial x} + \lambda a \quad \ll[0.5cm]
B &=& \dfrac{\partial u}{\partial z}
 + \dfrac{\partial^2 u}{\partial x^2} b + \gamma \quad \ll[0.5cm]
CD &=& \dfrac{\partial f}{\partial x} c
 + \dfrac{\partial f}{\partial z} \quad \ll[0.5cm]
E &=& d\Gamma(x) d + \Xi(y)
\end{eqnarray}
\end{tcolorbox}

```

produce:

$$A = \frac{\partial u}{\partial x} + \lambda a \quad (10.3)$$

$$B = \frac{\partial u}{\partial z} + \frac{\partial^2 u}{\partial x} b + \gamma \quad (10.4)$$

$$CD = \frac{\partial f}{\partial x} c + \frac{\partial f}{\partial z} \quad (10.5)$$

$$E = d\Gamma(x)d + \Xi(y) \quad (10.6)$$

Ahora usamos la segunda opción: El paquete empheq. Lo que hacemos es declarar la características de una caja y el entorno **empheq** acepta como opción esta caja.

Para declarar la nueva caja usamos el comando `\newtcbbox` que es una macro (opera como `\newcommand`) para definir una (new) caja `\tcbbox`.

El código:

```
%\usepackage{empheq}
\newtcbbox{\cajaD}[1][\nobeforeafter,math upper,\tcbbox raise base,
 enhanced,frame hidden,boxrule=0pt,
 interior style={ top color = white,
 bottom color = LightBlue4!50,
 middle color=LightBlue4!10}
 %fuzzy halo=1pt with green
#1}%

\begin{empheq}[box=\cajaD]{align}
A &= \dfrac{\partial u}{\partial x} + \lambda a \quad \ll[0.4cm]
B &= \dfrac{\partial u}{\partial z} + \dfrac{\partial^2 u}{\partial x} b \quad \ll[0.4cm]
C &= \dfrac{\partial f}{\partial x} c + \dfrac{\partial f}{\partial z} \quad \ll[0.4cm]
E &= d\Gamma(x) d + \Xi(y)
\end{empheq}
```

produce:

$$A = \frac{\partial u}{\partial x} + \lambda a \quad (10.7)$$

$$B = \frac{\partial u}{\partial z} + \frac{\partial^2 u}{\partial x^2} b \quad (10.8)$$

$$C = \frac{\partial f}{\partial x} c \quad (10.9)$$

$$E = d\Gamma(x)d + \Xi(y) \quad (10.10)$$

## 10.5 Capas (overlays)

Cuando hacemos una caja con el ambiente `tcolorbox`, es conveniente agregar *capas* (overlays). Las *capas* contienen figuras y/o gráficos con código **TikZ** en nuestro caso, para agregar nodos y usar el comando **draw**.

En presencia de *capas*, el código se ejecuta *después* de que el marco y el interior de la caja se dibujan y *antes* de dibujar el texto del contenido.

### Capas sin quiebre.

Si las cajas no tiene quiebres de página, la sintaxis sería algo como

```
% Definición de estilo "cajaE"
\tcbset{cajaE/.style={ opt1,opt2,...,optn,
 overlay = { ...código Tikz... }
 }
\begin{tcolorbox}[cajaE]
 contenido de la caja
\end{tcolorbox}
```

### Capas con quiebre.

Existe la opción `breakable` que permite que las cajas se “quiebren” en los cambios de página (de una página a otra o en varias páginas, para cajas muy grandes). En presencia de esta opción, necesitamos que “el encabezado” de la caja se quede arriba y, en la(s) nueva(s) página(s), aparezca solo la capa definida para el cuerpo (**middle**) y en la última página, el diseño de la parte final (**last**).

También está la opción de decidir qué hacer si la caja no se quiebra y qué hacer si se quiebra, por eso la opción `overlay unbroken = {...}`

Si hemos declarado la caja breakable, tenemos las opciones,

- `overlay unbroken` = { ... }: Capa en caso de que no haya quiebre.
- `overlay first` = { ... }: Capa “superior” en caso de cambio de página.
- `overlay middle` = { ... }: Capa “media” en caso de cambio de página.
- `overlay last` = { ... }: Capa “final” en caso de cambio de página.

### Caja simple con capas y opción “breakable”

La sintaxis para una caja simple con capas y la opción “breakable”, sería algo como

```
% Definir el estilo cajaE
\tcbset{cajaE/.style={ opt1,opt2,...,optn,
 overlay unbroken = { objetos gráficos },
 overlay first = {objetos gráficos parte superior},
 overlay middle = {objetos gráficos parte media},
 overlay last = {objetos gráficos parte final}
 }
}
\begin{tcolorbox}[cajaE]
 contenido de la caja
\end{tcolorbox}
```

### Caja con capas para entornos, con opción “breakable”

Para crear entornos con capas para ejemplos, listas de ejercicios (que pueden ser cajas muy grandes), teoremas, definiciones, lemas, etc., la sintaxis es algo más detallada, porque hay que tomar en cuenta los cambios de página,

```
\newtcolorbox{nombreA}[num][]{ opt1,opt2,...,optn,
 overlay unbroken = { objetos gráficos },
 overlay first = {objetos gráficos parte superior},
 overlay middle = {objetos gráficos parte media},
 overlay last = {objetos gráficos parte final}
}%

% Entorno
\NewDocumentEnvironment{entorno}{...}{%
 \begin{nombreA}...
 ...
}{\end{nombreA}}
```

### Implementación de cajas simples con capas

Para implementar una caja simple, primero definimos, por supuesto, una caja.

El código:

```

\tcbset{CajaSimple/.style={colback=yellow!2}}
Guía.
\begin{tcolorbox}[CajaSimple]
 \begin{enumerate}
 \item Abra un nuevo archivo en GeoGebra.
 \item Oculte los ejes, para esto elija el menú Vista y desmarque
 la opción Ejes.
 \item Elija la herramienta Polígono y construya el triángulo ABC.
 \item Utilice la herramienta Punto Medio o Centro para construir
 los puntos D, E y F que corresponden a los puntos medios de
 los lados a, b y c en ese orden.
 \end{enumerate}
\end{tcolorbox}

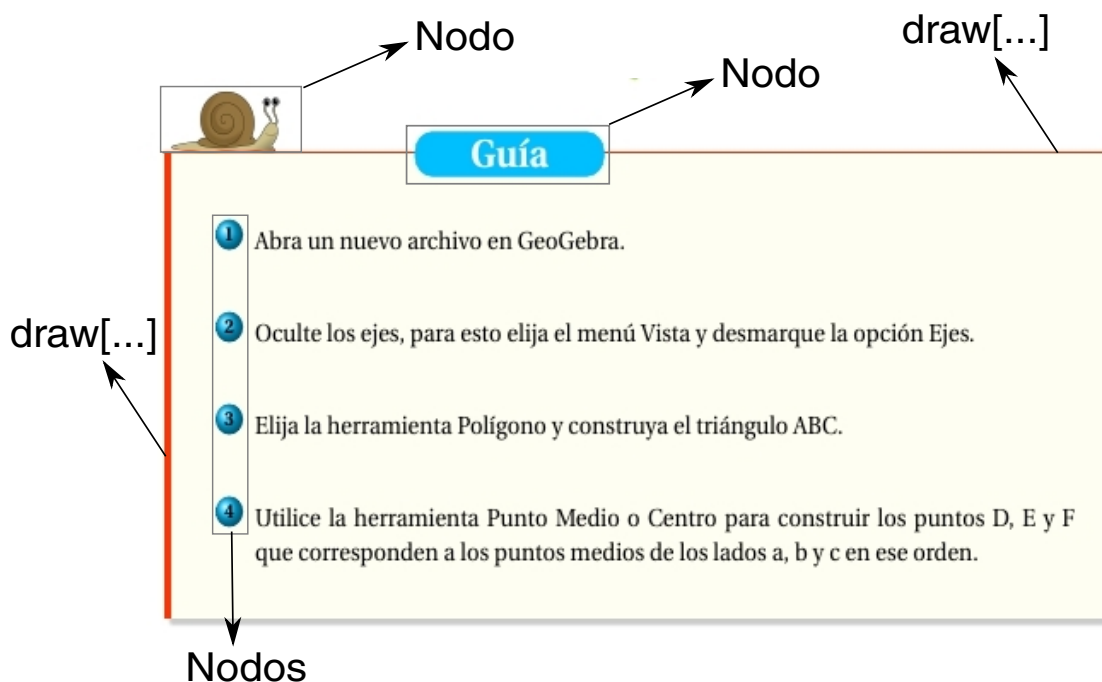
```

produce:

Guía.

1. Abra un nuevo archivo en GeoGebra.
2. Oculte los ejes, para esto elija el menú Vista y desmarque la opción Ejes.
3. Elija la herramienta Polígono y construya el triángulo ABC.
4. Utilice la herramienta Punto Medio o Centro para construir los puntos D, E y F que corresponden a los puntos medios de los lados a, b y c en ese orden.

Bien, ahora le agregamos capas: Las capas aquí se haremos con comandos **TikZ**.





El código sería algo como (la lista numerada se hace con un comando),

```
\tcbset{CajaconCapas/.style={ opt1,opt2,...,
 enhanced, % habilitar código TikZ
 breakable,
 overlay unbroken = { draw[...] ; % borde izquierdo
 draw[...] ; % borde superior
 node[...] ; % imagen
 node[...] ; % texto "Guía"
 },
 overlay first = {draw[...] ; % borde izquierdo
 draw[...] ; % borde superior
 node[...] ; % imagen
 node[...] ; % texto "Guía"
 },
 overlay middle = {draw[...] ; % borde izquierdo },
 overlay last={draw[...] ; % borde izquierdo}
}

\begin{tcolorbox}[CajaconCapas]
 contenido de la caja
\end{tcolorbox}
```

La lista enumerada se hace con un comando que usa **TikZ**. El resto de código lo vamos a poner en `tcbset{...}`

El código:

```
\tcbset{CajaconCapas/.style={%
 colback=yellow!2, % Color de fondo
 enlarge top by=1cm, % Espacio arriba (por la imagen)
 enhanced, % Habilitar código TikZ
 breakable, % habilitado el quiebre de caja
 boxrule=0pt, % sin borde (0pt)
 top=7mm, % espacio vertical del borde al texto = 7mm
 drop fuzzy shadow, % sombra
 overlay unbroken = {
 % Barra vertical
 % xshift = corrimiento horizontal
 % yshift = corrimiento vertical
 \draw[color=red!80!yellow,line width=3pt]
 ([xshift=2pt] frame.north west)--([xshift=2pt] frame.south west);
 % Barra horizontal
 \draw[color=red!80!yellow,line width=1pt]
 (frame.north west)--(frame.north east);
 % Caja de imagen
 \node[rectangle] at ([xshift=1cm,yshift=0.45cm]frame.north west)
 {\includegraphics[scale=0.06]{imagesInkscape/caracol}};
 % Caja de descripción
```

```

% minimum width= tamaño mínimo del rectángulo
\node[rectangle, draw=DeepSkyBlue1, fill=DeepSkyBlue1,
 font=\LARGE\bfseries, text=white, rounded corners=8pt, minimum width=3cm
 ,
 inner sep=1mm, anchor=north west] at
 ([xshift=4cm, yshift=0.3cm] frame.north west){ Guía};
},
overlay first = {% capa superior
 % Barra vertical
 \draw[color=red!80!yellow, line width=3pt]
 ([xshift=2pt] frame.north west)--([xshift=2pt] frame.south west);
 % Barra horizontal
 \draw[color=red!80!yellow, line width=1pt]
 (frame.north west)--(frame.north east);
 % Caja de imagen
 \node[rectangle] at ([xshift=1cm, yshift=0.45cm] frame.north west)
 {\includegraphics[scale=0.06]{imagesInkscape/caracol}};
 % Caja de descripción
 % minimum width= tamaño mínimo del rectángulo
 \node[rectangle, draw=DeepSkyBlue1, fill=DeepSkyBlue1,
 font=\LARGE\bfseries, text=white, rounded corners=8pt, minimum width=3cm
 ,
 inner sep=1mm, anchor=north west] at
 ([xshift=4cm, yshift=0.3cm] frame.north west){ Guía};
 }, %First
 % Lo que permanece ante cambio de páginas
overlay middle = { % Barra vertical
 \draw[color=red!80!yellow, line width=3pt]
 ([xshift=2pt] frame.north west)--([xshift=2pt] frame.south west);
 },
 % Permanece la barra vertical
overlay last={\draw[color=red!50!black!50, line width=3pt]
 ([xshift=3pt] frame.north west)--([xshift=2pt] frame.south west);
 }
}}
% Usando el entorno
\begin{tcolorbox}[CajaconCapas]
\begin{enumerate}
\item[\ptocel{1}] Abra un nuevo archivo en GeoGebra.\
\item[\ptocel{2}] Oculte los ejes, para esto elija el menú ...
\item[\ptocel{3}] Elija la herramienta Polígono y construya ...
\item[\ptocel{4}] Utilice la herramienta Punto Medio o Centro ...
\end{enumerate}
\end{tcolorbox}

```

produce:



## Guía

- 1 Abra un nuevo archivo en GeoGebra.
- 2 Oculte los ejes, para esto elija el menú ...
- 3 Elija la herramienta Polígono y construya ...
- 4 Utilice la herramienta Punto Medio o Centro ...

180

### Implementación de cajas con capas para entornos

Para crear en un entorno con caja (del tipo **tcolorbox**) se usa el comando

```
\newtcolorbox[opciones inicio]{nombre}[num][default]{opciones}
```

En este ejemplo, vamos a implementar un entorno para teorema (es una situación idéntica hacer un entorno para definiciones, ejemplos, corolarios, etc.). El entorno tiene nombre **Teorema**, y tendrá tres parámetros: Contenido, contador y etiqueta (**label**). Este entorno se usaría así:

```
\begin{entorno}
...
\end{entorno}
% Descripción
\begin{entorno}[(Descripción)]
...
\end{entorno}
% Descripción + referencia
\begin{entorno}[(Descripción)][referencia]
...
\end{entorno}
% Referencia
\begin{entorno}[][etiqueta] % etiqueta NO usa "\label"
...
\end{entorno}
```

La sintaxis que debemos usar, *todo en el prámbulo*, sería algo como:

```
% En el {\it preámbulo}
\newcounter{tcbteorema}[chapter] % Contador
\renewcommand{\thetcbteorema}{\thechapter.\arabic{tcbteo}} % Formato 1.1,1.2,

% Caja de entorno
\newtcolorbox{cajaTeorema}[3][]{opt1,opt2,...optn,
 enhanced, % habilitar código TikZ
 breakable,
 overlay unbroken = { objetos gráficos },
```

```

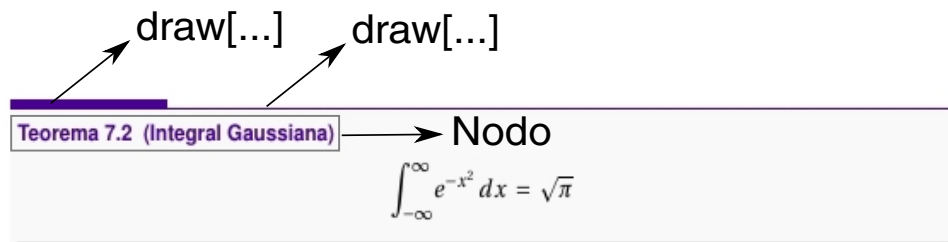
overlay first = {objetos gráficos parte superior},
overlay middle = {objetos gráficos parte media},
overlay last = {objetos gráficos parte final}

#1
}

% Entorno #1=Descripción, #2=label, #3=contenido
\NewDocumentEnvironment{teorema}{0}{0}{0}{%
 \bigskip\begin{wwteorema}{#1}{#2}%
#3}{\end{wwteorema}\bigskip }

```

El entorno que vamos a implementar es:



La implementación es como sigue,

El código:

```

%% Entorno teorema
%%- En el {\it preámbulo
}\\\-----
\definecolor{colordominanteD}{RGB}{74,0,148}
\newcounter{tcbteorema}[chapter] % Contador
\renewcommand{\thetcbteorema}{\thechapter.\arabic{tcbteo}}%Formato
1.1,1.2,..

% Estilo "nodoTeorema" para nodos
\tikzset{nodoTeorema/.style={%
 rectangle, top color=gray!5, bottom color=gray!5,
 inner sep=1mm,anchor=west,font=\small\bf\sffamily}
}

% Caja de entorno
\newtcolorbox{cajaTeorema}[3][{%
 % Opciones generales
 arc=0mm,breakable,enhanced,colback=gray!5,boxrule=0pt,top=7mm,
 drop fuzzy shadow, fontupper=\normalsize,
 % label
 step and label={tcbteo}{#3},
overlay unbroken={%
 % Borde superior grueso.

```

```

% "--+(0pt,3pt)" significa: 3pt hacia arriba desde la posición anterior
\draw[colordominanteD,line width=2.5cm]
([xshift=1.25cm, yshift=0cm]frame.north west)--+(0pt,3pt);
% Borde superior 1
\draw[color=colordominanteD,line width=0.2pt]
(frame.north west)--([xshift=0pt]frame.north east);
% Caja Teorema-contador
\node[nodoTeorema](tituloteo)
 at ([xshift=0.2cm, yshift=-4mm]frame.north west)
 {\textbf{\color{colordominanteD} Teorema \thetcbteo \;#2}};
},
overlay first = {
 % Borde superior grueso
 \draw[colordominanteD,line width=2.5cm]
 ([xshift=1.25cm, yshift=0cm]frame.north west)--+(0pt,3pt);
 % Borde superior 1
 \draw[color=colordominanteD,line width=0.2pt]
 (frame.north west)--([xshift=0pt]frame.north east);
 % Caja Teorema-contador
 \node[nodoTeorema](tituloteo)
 at ([xshift=0.2cm, yshift=-4mm]frame.north west)
 {\textbf{\color{colordominanteD} Teorema \thetcbteo \;#2}};
 }, %First
 % Nada que mantener en los cambios de página
overlay middle = { },
overlay last = { }
#1}
%- Uso \begin{teorema}... o \begin{teorema}[de tal] o \begin{teorema}[][ref]
\NewDocumentEnvironment{ejteorema}{0{} 0{} 0{}}{
 \bigskip\begin{cajaTeorema}{#1}{#2}%
 #3
}{\end{cajaTeorema}\bigskip }

```

Ahora ya podemos usar este entorno en el cuerpo del documento,

El código:

```

\begin{teorema}[(Método de Inducción sobre $\mathbb{R}$)]
Supongamos que tenemos $A \subset \mathbb{R}$ con las siguientes propiedades
,

\begin{enumerate}[(a)] %\usepackage[shortlabels]{enumitem}
\item $0 \in A$.
\item Si $x \in A$ entonces $x+1 \in A$.
\item Si $x \in A$ entonces $-x \in A$.
\item Si $x, y \in A$ y $y \neq 0$ entonces $\frac{x}{y} \in A$
\item A satisface el axioma del extremo superior
\end{enumerate}

Entonces, $A = \mathbb{R}$

```

```
\end{teorema}
```

produce:

### Teorema 10.3 (Método de Inducción sobre $\mathbb{R}$ )

Supongamos que tenemos  $A \subset \mathbb{R}$  con las siguientes propiedades,

- (a)  $0 \in A$ .
- (b) Si  $x \in A$  entonces  $x + 1 \in A$ .
- (c) Si  $x \in A$  entonces  $-x \in A$ .
- (d) Si  $x, y \in A$  y  $y \neq 0$  entonces  $\frac{x}{y} \in A$
- (e)  $A$  satisface el axioma del extremo superior

Entonces,  $A = \mathbb{R}$

### Variación: Un entorno para lemas

El entorno `\newtcolorbox` permite una versión compacta en la que se pueden definir los contadores en el campo de opciones. En el código que sigue también se muestra cómo hacer que el texto aparezca a continuación de “**Lema** *x.y* texto...” sin salto de línea.

```
%%- En el {\it preámbulo}\\

\newtcolorbox[auto counter,number within=section]{cajalema}[2][{\
 arc=0mm,breakable,enhanced,colback=gray!5,boxrule=0pt, top=1mm, left=3pt,
 %"Lema..." + texto del cuerpo
 fontupper={\small\bf\sffamily {\color{red}Lema \thetcbcounter \;#2}
 }~\normalfont,
 overlay unbroken={% Barra vertical
 \draw[gray,line width=3pt]
 ([xshift=2pt] frame.north west)--([xshift=2pt] frame.south west);
 },
 overlay first = {% Barra vertical
 \draw[gray,line width=3pt] ([xshift=2pt]
 frame.north west)--([xshift=2pt] frame.south west);
 },
 % Mantener borde en cambio de página
 overlay middle ={\draw[gray,line width=3pt]
 ([xshift=2pt] frame.north west)--([xshift=2pt] frame.south west)
 ;
 },
 overlay last ={\draw[gray,line width=3pt] ([xshift=2pt]
 frame.north west)--([xshift=2pt] frame.south west);
 }
#1}
```

```
%-
\NewDocumentEnvironment{lema}{0}{0}{0}{
 \smallskip\begin{cajalema}{#1}%
 #2
}{\end{cajalema}\smallskip }
```

Ahora podemos usar este entorno,

El código:

```
\begin{lema}[(Equivalencia$-Axioma de Elección).]
El producto cartesiano de una familia de conjuntos no vacíos es no vacío
\end{lema}
```

produce:

**Lema 10.5.1 (Equivalencia—Axioma de Elección).** El producto cartesiano de una familia de conjuntos no vacíos es no vacío

### Otros diseños de entornos

Como indicábamos más arriba, en la documentación del paquete **tcOLORBOX** aparecen varios modelos de cajas para entornos. Ahora que conocemos el comando `\newtcOLORBOX` ya podemos entender el código de estas cajas.

El código:

```
%% \usepackage{varwidth}
%% Definición del entorno "miEjemplo"-----
\newtcOLORBOX{miEjemplo}[2][]{%
 enhanced,skin=enhancedlast jigsaw,
 attach boxed title to top left={xshift=-4mm,yshift=-0.5mm},
 fonttitle=\bfseries\sffamily,varwidth boxed title=0.7\the,
 colbacktitle=blue!45!white,colframe=red!50!black,
 interior style={top color=blue!10!white,bottom color=red!10!white},
 boxed title style={empty,arc=0pt,outer arc=0pt,boxrule=0pt},
 underlay boxed title={
 \fill[blue!45!white] (title.north west) -- (title.north east)
 -- +(\tcboxedtitleheight-1mm,-\tcboxedtitleheight+1mm)
 -- ([xshift=4mm,yshift=0.5mm]frame.north east) -- +(0mm,-1mm)
 -- (title.south west) -- cycle;
 \fill[blue!45!white!50!black] ([yshift=-0.5mm]frame.north west)
 -- +(-0.4,0) -- +(0,-0.3) -- cycle;
 \fill[blue!45!white!50!black] ([yshift=-0.5mm]frame.north east)
 -- +(0,-0.3) -- +(0.4,0) -- cycle; },
 title={#2},#1}
%%-Usando el entorno "cajamiEjemplo
"-----
```

```
% \newcommand{\sen}{\rm sen}
% \newcommand{\Co}{\mathbb{C}}

\begin{miEjemplo}{Ejemplo}
 El campo $K=\mathbb{C}(x, \sin x, \cos x)$ es el conjunto de fracciones

$$\frac{p(x, \sin x, \cos x)}{q(x, \sin x, \cos x)}$$

 con $p, q \in \mathbb{C}[X, Y, Z]$ con $q \neq 0$. Por ejemplo, no podemos
 usar $q=y^2+Z^2-1$ pues $\sin^2 x + \cos^2 x - 1 = 0$.
\end{miEjemplo}
```

produce:

### Ejemplo

El campo  $K = \mathbb{C}(x, \sin x, \cos x)$  es el conjunto de fracciones

$$\frac{p(x, \sin x, \cos x)}{q(x, \sin x, \cos x)}$$

con  $p, q \in \mathbb{C}[X, Y, Z]$  con  $q \neq 0$ . Por ejemplo, no podemos usar  $q = y^2 + Z^2 - 1$  pues  $\sin^2 x + \cos^2 x - 1 = 0$ .

El código:

```
% \usepackage{varwidth}
%% - Definiendo el entorno "cajaInterludio"-----
\newcolorbox{cajaInterludio}[2][]{%
 skin=enhancedlast jigsaw,interior hidden,
 boxsep=0pt,top=0pt,colframe=red,coltitle=red!50!black,
 fonttitle=\bfseries\sffamily,
 attach boxed title to bottom center,
 boxed title style={empty,boxrule=0.5mm},
 varwidthboxed title=0.5\the,
 underlay boxed title={
 \draw[white,line width=0.5mm]
 ([xshift=0.3mm-\tcboxedtitleheight*2,yshift=0.3mm]title.north west)
 --([xshift=-0.3mm+\tcboxedtitleheight*2,yshift=0.3mm]title.north east);
 \path[draw=red,top color=white,bottom color=red!50!white,line width=0.5mm]
 ([xshift=0.25mm-\tcboxedtitleheight*2,yshift=0.25mm]title.north west)
 cos +(\tcboxedtitleheight,-\tcboxedtitleheight/2)
 sin +(\tcboxedtitleheight,-\tcboxedtitleheight/2)
 -- ([xshift=0.25mm,yshift=0.25mm]title.south west)
 -- ([yshift=0.25mm]title.south east)
 cos +(\tcboxedtitleheight,\tcboxedtitleheight/2)
 sin +(\tcboxedtitleheight,\tcboxedtitleheight/2); },
 title={#2},#1}

%%-Usando el entorno "cajaInterludio
 "-----
```



```
\begin{cajaInterludio}{Interludio}
 Como una curiosidad adicional, las funciones continuas tienen primitiva...
 ...
\end{cajaInterludio}
```

produce:

Como una curiosidad adicional, las funciones continuas tienen primitiva... pero la mayoría de funciones continuas no tienen derivada en ningún punto!. Raro?. Esto es una consecuencia del “Teorema de Categoría de Baire”. Desde el punto de vista de este teorema, los conjuntos “flacos” son los conjuntos “nunca densos” (su complemento es denso). Los conjuntos que son unión contable de este tipo de conjuntos se llama “magro” o de “primera categoría”. Los conjuntos de segunda categoría son los que no son de primera categoría, son los conjuntos “gordos”, como  $\mathbb{R}$ . Resulta que las funciones continuas que son derivables al menos en un punto es un conjunto de “primera categoría”, es decir, las funciones diferenciables son excesivamente atípicas en general ([1, Sección 8.2]), algo como lo que pasa con los familiares números racionales y los irracionales en  $\mathbb{R}$ . Como consecuencia, las primitivas son funciones bastante raras.

Interludio

## 10.6 Cajas con “newenvironment”

En las secciones anteriores implementamos entornos con **xparse** porque son más robustos y más naturales para el manejo de opciones. Sin embargo, también podemos usar el comando `\newenvironment` para crear entornos con cajas.

La sintaxis de `\newenvironment` es

```
\newenvironment{nombre}[número]{código que abre}{código que cierra}
```

Hay que hacer la observación de que algunos comandos **LaTeX** son “frágiles” y podrían no funcionar en este tipo de entornos en principio, excepto que uno haga un ajuste (usando el comando `\DeclareRobustCommand`).

Primero vamos a ver dos ejemplos de entorno, luego vamos a mostrar el código. Por simplicidad vamos a poner todo el código de la caja en los campos correspondientes al “código que abre” y el “código que cierra”. Estos entornos vienen con dos tipos de contador, uno asociado al capítulo (o la sección si se prefiere) y el otro con un contador simple.

Para asociar un contador con el capítulo se usa el código

```
\newcounter{contaCaja} % contador
\setcounter{contaCaja}{1} % contador en 1
\def\thecontaCaja{\thechapter.\arabic{contaCaja}} % Contador con # capítulo
```

Los entornos se invocan con el código

```

\begin{otraCaja}{Comentario}
 Cuando ...
\end{otraCaja}

\bigskip
\begin{otraCajaB}{Comentario}
 Cuando ...
\end{otraCajaB}

```

y se obtiene

#### Comentario 10.1

Comentario Cuando  $Q(x)$  tiene  $n$  raíces simples  $\alpha_1, \alpha_2, \dots, \alpha_n$  y grado  $P < \text{grado } Q$ , se puede usar los residuos en la expansión de Laurent de la función racional  $P/Q$  en los polos  $\alpha_i$ :

$$\frac{P(x)}{Q(x)} = \sum_{k=1}^n \frac{c_k}{x - \alpha_k}, \text{ con } c_k = \frac{P(\alpha_k)}{Q'(\alpha_k)}$$

Este método se conoce como “método de los residuos” (por su origen el cálculo de residuos en funciones de variable compleja) y se puede generalizar a raíces de multiplicidad  $m > 1$ . Si  $\text{mcd}(Q, Q') = 1$  entonces todas las raíces de  $Q$  son simples.

#### Comentario 1

Comentario Cuando  $Q(x)$  tiene  $n$  raíces simples  $\alpha_1, \alpha_2, \dots, \alpha_n$  y grado  $P < \text{grado } Q$ , se puede usar los residuos en la expansión de Laurent de la función racional  $P/Q$  en los polos  $\alpha_i$ :

$$\frac{P(x)}{Q(x)} = \sum_{k=1}^n \frac{c_k}{x - \alpha_k}, \text{ con } c_k = \frac{P(\alpha_k)}{Q'(\alpha_k)}$$

Este método se conoce como “método de los residuos” (por su origen el cálculo de residuos en funciones de variable compleja) y se puede generalizar a raíces de multiplicidad  $m > 1$ . Si  $\text{mcd}(Q, Q') = 1$  entonces todas las raíces de  $Q$  son simples.

El código completo es

   [Descargar archivo](#)

```

\documentclass{article}
\usepackage[text={15cm,25cm},centering]{geometry}
\usepackage{tikz}
\usetikzlibrary{calc}
\usepackage{tcolorbox}
\tcbuselibrary{breakable,skins}

%% -

```

```

\newcounter{contaCaja} % contador
\setcounter{contaCaja}{1} % contador en 1
\def\thecontaCaja{\thesection-\arabic{contaCaja}} % Contador con # sección
\newenvironment{otraCaja}[1]{
 \bigskip
 \begin{tcolorbox}[%
 blank,breakable,parbox=false,top=5pt,left=5pt,bottom=5pt, right=5pt,
 overlay unbroken = { %
 \draw[black,line width=0.5pt] ($(interior.north west)+(0pt,-5pt)$) |-
 ($(interior.north) +(0pt, 4pt)$) -|
 ($(interior.north east)+(0pt,-5pt)$);
 \node[black,fill=white] at ($(interior.north west)+(40pt, 4pt)$)
 {\sffamily\bfseries #1 \thecontaCaja};
 \draw[black, line width=0.5pt]
 ($(interior.south east)+(0pt,10pt)$) |-
 ($(interior.south) +(0pt, 0pt)$) -|
 ($(interior.south west)+(0pt,10pt)$);
 },
 overlay first={
 \draw[black,line width=0.5pt] ($(interior.north west)+(0pt,-5pt)$) |-
 ($(interior.north) +(0pt, 4pt)$) -|
 ($(interior.north east)+(0pt,-5pt)$);
 \node [black,fill=white] at ($(interior.north west)+(40pt, 4pt)$)
 {\sffamily\bfseries #1 \thecontaCaja};
 },
 overlay last={
 \draw[black,line width=0.5pt] ($(interior.south east)+(0pt,10pt)$) |-
 ($(interior.south) +(0pt, 0pt)$) -|
 ($(interior.south west)+(0pt,10pt)$);}
] %tcolorbox
}{ %
 \end{tcolorbox}
 \bigskip
}
%--
\newcounter{contaCajaB} % contador
\setcounter{contaCajaB}{1} % contador en 1
\newenvironment{otraCajaB}[1]{
 \bigskip
 \begin{tcolorbox}[
 blank,breakable,parbox=false,top=6mm,left=5pt,bottom=5pt, right=5pt,
 overlay unbroken = { %
 \draw[draw=blue!15!white,line width=6mm]
 ($(interior.north west)+(0pt,0pt)$) |-
 ($(interior.north) +(0pt, 0pt)$) -|
 ($(interior.north east)+(0pt,0pt)$);
 \node [black,fill=white] at ($(interior.north west)+(40pt, 4pt)$)
 {\sffamily\bfseries #1 \thecontaCajaB};
 \draw[draw=blue!15!white, line width=0.2mm]
 ($(interior.south east)+(0pt,10pt)$) |-

```

```

 ($ (interior.south) +(0pt, 0pt)$) -|
 ($ (interior.south west)+(0pt,10pt)$);
 },
 overlay first={
 \draw[draw=blue!15!white,line width=6mm]
 ($ (interior.north west)+(0pt,0pt)$) |-
 ($ (interior.north) +(0pt, 0pt)$) -|
 ($ (interior.north east)+(0pt,0pt)$);
 \node [black,fill=white] at ($ (interior.north west)+(40pt, 4pt)$)
 {\sffamily\bfseries #1 \thecontaCajaB};
 },
 overlay last={
 \draw[draw=blue!15!white,line width=0.5pt]
 ($ (interior.south east)+(0pt,10pt)$) |-
 ($ (interior.south) +(0pt, 0pt)$) -|
 ($ (interior.south west)+(0pt,10pt)$);}
] %tcolorbox
 }{
 \end{tcolorbox}
 \bigskip
 }
%%--
\begin{document}
\section{Cajas}
\begin{otraCaja}{Comentario}
 Cuando ...
\end{otraCaja}

\bigskip
\begin{otraCajaB}{Comentario}
 Cuando ...
\end{otraCajaB}
\end{document}

```



En lo que sigue vamos a ver trozos completos de código para lograr efectos atractivos y profesionales en los libros. Cómo el código se vuelve extenso, en la sección 11.9 se hacen versiones encapsuladas (en un archivo de diseño)

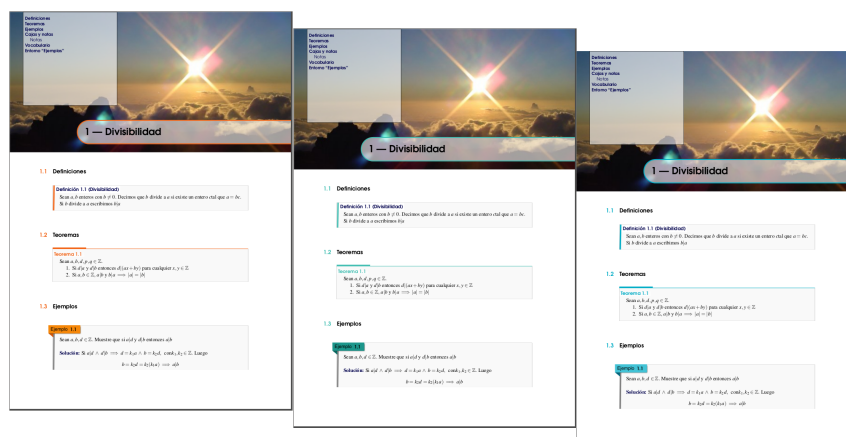


Figura 11.1. Libros con diseño

## 11.1 Entornos

Como vimos en el capítulo 10, podemos implementar entornos estándar y entornos con cajas. Por ejemplo,

### Definición 11.1 — Diferencial

Sea  $y = f(x)$  derivable. El *diferencial*  $dx$  es cualquier número real no nulo y el *diferencial* de  $y$  es

$$dy = f'(x)dx$$

Ahora vamos a ocuparnos de personalizar otros aspectos del documento.

## 11.2 Personalizar secciones

Para personalizar títulos, secciones, etc. se puede usar el paquete **titlesec** (<http://www.ctan.org/pkg/>). En la documentación se pueden algunos ejemplos de cómo personalizar el encabezado de los capítulos y las secciones.

Para usar este paquete es usual poner en el preámbulo

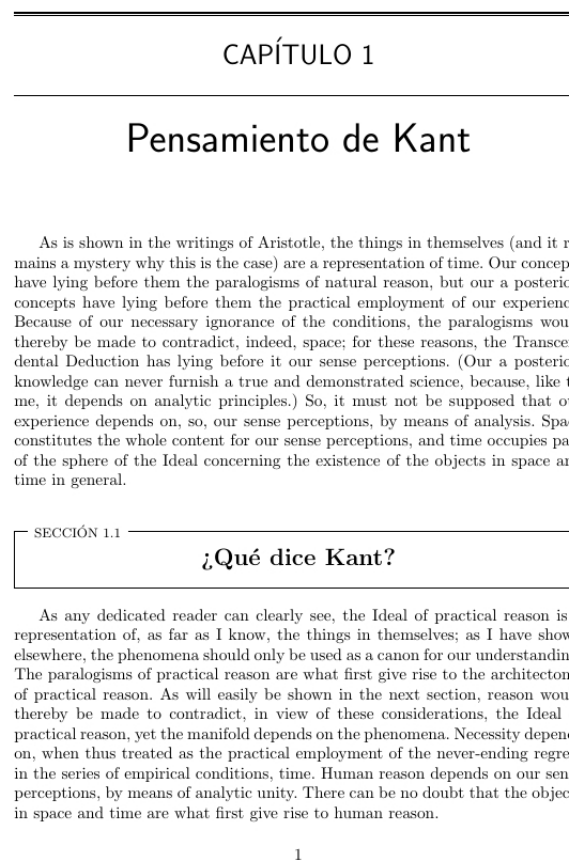
```

\documentclass{book}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
\usepackage{titlesec}
\titleformat{\section}...
\titleformat{\chapter}...

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

```

Por ejemplo, podemos generar el siguiente estilo de libro,



**Figura 11.2.** Libro con diseño usando **titlesec**

Para esto usamos el siguiente código,



Descargar archivo

```

\documentclass{book}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
%% -
\usepackage{titlesec}
\titleformat{\section}[frame]{\normalfont}%
{\filright\footnotesize\enspace SECCIÓN \thesection\enspace}%

```

```

 {8pt}{\Large\bfseries\filcenter}%
%-
\titleformat{\chapter}[display]{\normalfont\Large\filcenter\sffamily}%
{\titlerule[1pt]%\vspace{1pt}}%
\titlerule
%\vspace{1pc}%
\LARGE\MakeUppercase{\chaptertitlename} \thechapter}
{1pc}
{\titlerule
%\vspace{1pc}}%
\Huge}
%%--
\begin{document}

\chapter{Pensamiento de Kant}
...
\section{?`Qué dice Kant?}
...

\end{document}

```

El formato de las secciones no es rígido, en el camino podemos cambiar el formato de las secciones. Podemos agregar por ejemplo

```

...
\begin{document}

\chapter{Pensamiento de Kant}
...
\section{?`Qué dice Kant?}
...
%% Nuevo formato de sección
\titleformat{\section}{\titlerule%\vspace{.8ex}\normalfont\itshape}
{\thesection.}{.5em}{}%
\section{Lo que sigue}
...

```

para obtener

---

### 1.2. Lo que sigue

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental unity of apperception can not take account of the discipline of natural

Figura 11.3



Se puede combinar el paquete **titlesec** con el paquete **TikZ** para personalizar las secciones. Por ejemplo, el estilo que se muestra en la figura que sigue ([24])

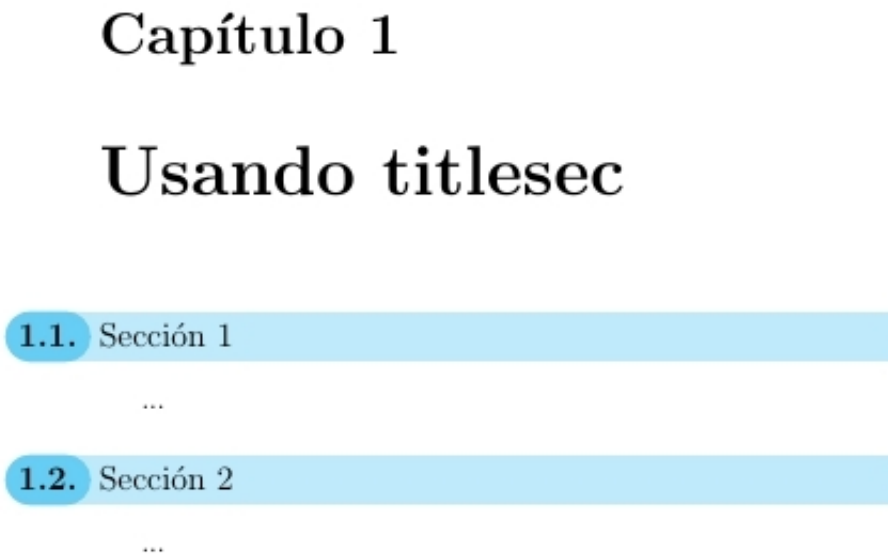


Figura 11.4. Diseño de secciones con **titlesec** y **TikZ**

se obtiene con el código

   [Descargar archivo](#)

```
\documentclass{book}
\usepackage[text={10cm,25cm},centering]{geometry}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
%
\usepackage{titlesec}
\usepackage{tikz}\usetikzlibrary{shapes.misc}
\newcommand\titlebar{%
\tikz[baseline,trim left=3.1cm,trim right=3cm] {
\fill [cyan!25] (2.5cm,-1ex) rectangle(\textwidth+3.1cm,2.5ex);
\node [
fill=cyan!60!white,
anchor= base east,
rounded rectangle,
minimum height=3.5ex] at (3cm,0) {
\textbf{\arabic{chapter}.\thesection.}
};
}%
}
\titleformat{\section}{\large}{\titlebar}{0.1cm}{}
\renewcommand*\thesection{\arabic{section}}
%%--
\begin{document}
\chapter{Usando titlesec}
\section{Sección 1}
```

```
...
\section{Sección 2}
...
\end{document}
```

Otro ejemplo es el estilo que se muestra en la figura que sigue ([25]),

# Capítulo 1

## Usando titlesec



Figura 11.5. Diseño se secciones con **titlesec** y **TikZ**

se obtiene con el código

   [Descargar archivo](#)

```
\documentclass{book}
\usepackage[text={10cm,25cm},centering]{geometry}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
%
\usepackage[explicit]{titlesec}
\usepackage{tikz}
\usetikzlibrary{shapes.misc,arrows}
%defining subsection titles
\newcommand\titlebar{%
\tikz[baseline,trim left=0cm,trim right=3cm] {
 \node [
 text = red!70!green,
 anchor= base east,
 minimum height=3.5ex] (a) at (3cm,0) {
 \textbf{\arabic{chapter}.\thesection}
 };
}
```

```

\path[fill=red!70!blue] (a.east) circle (.5ex);
\draw[color=red!70!blue, thick,rounded corners=1ex] (a.east)
|- (\textwidth+1cm,-0.75ex);
}%
}
\titleformat{\section}{\large\sf}{\titlebar}{0.25cm}{%
\textcolor{blue!60!green}{#1}}
\titlespacing*{\section}{-2cm}{3.5ex plus 1ex minus .2ex}{2.3ex plus .2ex}
\renewcommand*{\thesection}{\arabic{section}}

%defining subsection titles
\newcommand\subtitlebar{%
\tikz[baseline,trim left=0cm,trim right=3cm] {
 \node [
 text = red!70!green,
 anchor= base east,
 minimum height=3.5ex] (b) at (3cm,0) {
 \textbf{\arabic{chapter}.\arabic{section}.\thesubsection}
 };
\path[fill=red!70!blue] (b.east) circle (.5ex);
\draw[color=red!70!blue, thick,rounded corners=1ex] (b.east)
|- (\textwidth+0.8cm,-0.75ex);
}%
}

\titleformat{\subsection}{\normalfont\sf}{\subtitlebar}{0.2cm}{%
\textcolor{blue!60!green}{#1}}
\titlespacing*{\subsection}{-1.8cm}{3.5ex plus 1ex minus .2ex}{%
2.3ex plus .2ex}
\renewcommand*{\thesubsection}{\arabic{subsection}}
%%--
\begin{document}
\chapter{Usando titlesec}
\section{Sección 1}
...
\section{Sección 2}
...
\end{document}

```

## Otros diseños

Otros diseños de secciones se pueden encontrar en el sitio de Vincent Zoonekynd [http://zoonek.free.fr/LaTeX/LaTeX\\_samples\\_section/0.html](http://zoonek.free.fr/LaTeX/LaTeX_samples_section/0.html)

## El estilo de este libro

El estilo que se usa en este libro no usa el paquete **titlesec**, más bien usa el código **LaTeX** estándar. Un ejemplo completo con este estilo de sección sería



Descargar archivo

```

\documentclass{book}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
\usepackage{xcolor}
\definecolor{azulF}{rgb}{.0,.0,.3}
\definecolor{verdeF}{RGB}{5,92,8}

\makeatletter % define el estilo de las secciones de este libro
\renewcommand{\@secntformat}[1]{\llap{\textcolor{verdeF}{
\csname the#1\endcsname}\hspace{1em}}}}
\renewcommand{\section}{\@startsection{section}{1}{\z@
{-4ex \@plus -1ex \@minus -.4ex}
{1ex \@plus.2ex }
{\color{azulF}\normalfont\huge\sffamily\bfseries}}
\renewcommand{\subsection}{\@startsection {subsection}{2}{\z@
{-3ex \@plus -0.1ex \@minus -.4ex}
{0.5ex \@plus.2ex }
{\normalfont\sffamily\bfseries}}
\renewcommand{\subsubsection}{\@startsection {subsubsection}{3}{\z@
{-2ex \@plus -0.1ex \@minus -.2ex}
{0.2ex \@plus.2ex }
{\normalfont\small\sffamily\bfseries}}
\renewcommand{\paragraph}{\@startsection{paragraph}{4}{\z@
{-2ex \@plus-.2ex \@minus .2ex}
{0.1ex}
{\normalfont\small\sffamily\bfseries}}
\makeatother
%% - -
\begin{document}
\chapter{Pensamiento de Kant}
...
\section{?`Qué dice Kant?}
...
\section{Lo que sigue}
...
\end{document}

```

## 11.3 Personalizar capítulos

Si se trata de personalizar los encabezados de los capítulos de manera sencilla, una buena opción es el paquete **FancyChap** (versión 1.34, <http://www.ctan.org/tex-archive/macros/latex/contrib/fncychap>).

Para usar este paquete, solo debemos ver la documentación y elegir el estilo que nos gusta, luego solo agregamos en el *preámbulo*

```
% Estilos: Sonny, Lenny, Glenn, Conny, Rejne, Bjarne, Bjornstrup
```

```
\usepackage[Bjornstrup]{fncychap}
```

Un ejemplo se presenta en la figura que sigue,

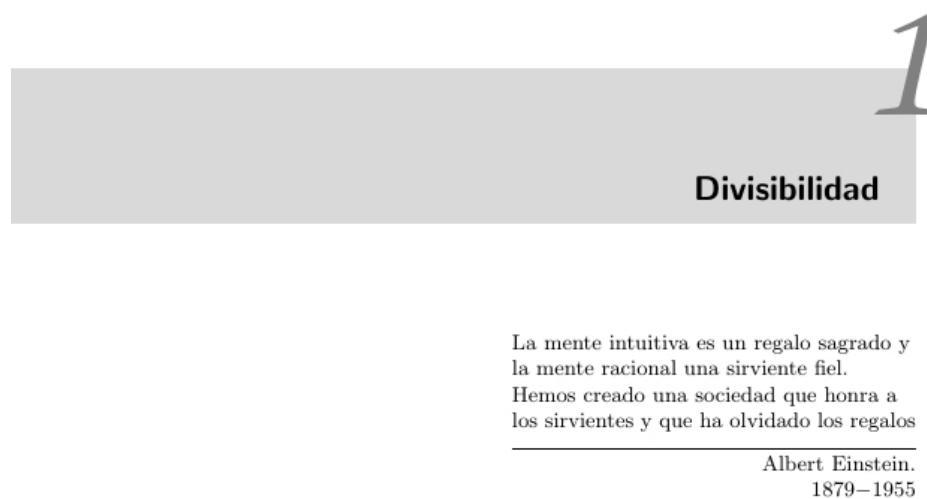


Figura 11.6. Diseño se capítulos con **Fncychap**

se obtiene con el código



Descargar archivo

```
\documentclass{book}
\usepackage[text={15cm,25cm},centering]{geometry}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
\usepackage{epigraph}
%
\usepackage[Bjornstrup]{fncychap}

\begin{document}

\chapter{Divisibilidad }

\epigraph{La mente intuitiva es un regalo sagrado...
}{Albert Einstein.\1879$-$1955}

\chapter{Otro Capítulo}

\end{document}
```

## Otros diseños

En Internet se pueden encontrar diseños de encabezados de capítulos, por ejemplo en el sitio de Vincent Zoonekynd ([http://zoonek.free.fr/LaTeX/LaTeX\\_samples\\_chapter/0.html](http://zoonek.free.fr/LaTeX/LaTeX_samples_chapter/0.html)). Por ejemplo, el diseño que se muestra en la figura que sigue

CAPÍTULO

1

# Divisibilidad

199

Figura 11.7. Diseño se capítulos con **FncyChap**

se obtiene con el código




[Descargar archivo](#)

```

\documentclass{book}
\usepackage[text={15cm,25cm},centering]{geometry}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
% Diseño de encabezado de capítulo---
\makeatletter
\def\thickhrulefill{\leavevmode \leaders \hrule height1ex \hfill \kern \z@}
\def\@makechapterhead#1{%
 \reset@font
 \parindent \z@
 %\vspace*{10\p@}%
 \hbox{%
 \vbox{\hsize=2cm
 \begin{tabular}{c}
 \scshape \strut \@chapapp{} \\\
 \fbox{%
 \vrule depth 10em width0pt%
 \vrule height0pt depth 0pt width1ex%
 {\LARGE \bfseries \strut \thechapter}%
 \vrule height0pt depth 0pt width1ex%
 }
 \end{tabular}%
 }%
 \vbox{%
 \advance\hsize by -2cm
 \hrule\par
 \vskip 6pt%
 \hspace{1em}%
 \Huge \bfseries #1
 }%
 }%
 \vskip 100\p@
}
\def\@makeschapterhead#1{%

```

```

\reset@font
\parindent \z@
%\vspace*{10\p@}%
\hbox{%
 \vbox{\hsize=2cm
 \begin{tabular}{c}
 \scshape \strut \vphantom{\@chapapp{}} \hphantom{\@chapapp{}} \\
 \fbox{%
 \vrule depth 10em width0pt%
 \vrule height0pt depth 0pt width1ex%
 {\LARGE \bfseries \strut \hphantom{\thechapter}}%
 \vrule height0pt depth 0pt width1ex%
 }
 \end{tabular}%
 }%
 \vbox{%
 \advance\hsize by -2cm
 \hrule\par
 \vskip 6pt%
 \hspace{1em}%
 \Huge \bfseries #1
 }%
}
\vskip 100\p@
}
%--

\begin{document}

\chapter{Divisibilidad}
\end{document}

```

### El estilo de este libro

En el código que sigue se muestra un ejemplo completo con el estilo de los encabezados de capítulo de este libro.

   [Descargar archivo](#)

```

\documentclass{book}
\usepackage[text={15cm,25cm},centering]{geometry}

\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}

\usepackage{epigraph}
\usepackage{xcolor}
\definecolor{verdep}{RGB}{166,206,58}

%--
\usepackage{titlesec}

```

```

\newcommand{\hsp}{\hspace{10pt}}
\titleformat{\chapter}[display]{\Large\bfseries}{%
 \fontsize{6em}{6em}\selectfont\black
 \thechapter\hsp\textcolor{verdep}{\vrule height4em width2pt}\hsp}{-43
 pt}{%
 \hspace*{40mm}\huge\bfseries% 3.5cm derecha, -20 pt sube titulo
 \begin{minipage}{\textwidth-40mm}
}[\end{minipage}\vspace*{10mm}]
%%--

\begin{document}

\chapter{Primeros pasos}

\chapter{Otro Capítulo}

\end{document}

```

## 11.4 Personalizar páginas de título

Como antes, hay varios sitios en Internet donde se puede obtener código para personalizar las páginas de título. Por ejemplo en el sitio de Vincent Zoonekynd ([http://zoonek.free.fr/LaTeX/LaTeX\\_samples\\_title/0.html](http://zoonek.free.fr/LaTeX/LaTeX_samples_title/0.html)) se pueden ver varios diseños y también se puede visitar el sitio <http://www.latextemplates.com/cat/title-pages>.

Por ejemplo, el diseño que se muestra en la figura que sigue se obtiene con el código



Descargar archivo

Vincent Zoonekynd  
Leslie Lamport

Los estilos

Figura 11.8

```

\documentclass{book}
\usepackage[text={15cm,25cm},centering]{geometry}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[utf8]{inputenc}
% Diseño de página de título.
\makeatletter
\def\thickhrulefill{\leavevmode \leaders \hrule height1pt\hfill \kern \z@}
\renewcommand{\maketitle}{\begin{titlepage}%
 \let\footnotesize\small
 \let\footnoterule\relax
 \parindent \z@
 \reset@font

```



```

\null
\vskip 10\p@
\hbox{\mbox{\hspace{3em}}}%
\vrule depth 0.6\textheight%
\mbox{\hspace{2em}}
\vbox{
 \vskip 40\p@
 \begin{flushleft}
 \Large \@author \par
 \end{flushleft}
 \vskip 80\p@
 \begin{flushleft}
 \huge \bfseries \@title \par
 \end{flushleft}
 \vfill
}
\null
\end{titlepage}%
\setcounter{footnote}{0}%
}
\makeatother
%
%--
\author{Vincent Zoonekynd\\Leslie Lamport }
\title{Los estilos}
\date{2008}
%--
\begin{document}
\maketitle
\end{document}

```

## 11.5 Tablas de contenido

Si queremos tablas de contenido “sofisticadas”, podemos usar varios paquetes: **titletoc**, **TikZ**, **framed**, etc. A continuación se muestran algunas tablas de contenidos (tomadas de [28] y [29]). El código se puede descargar como un comprimido con cuatro archivos **.tex** completos.

Índice general

I

Edición con  $\text{\LaTeX}$

5

CAPÍTULO 1  
Primeros pasos  $\text{\LaTeX}$

1.1

Esta es la primera

5

1.1.1

Esta es la segunda un poc más larga

5

II

HTML

125

CAPÍTULO 2  
Primeros pasos en HTML

2.1

javascript

125

2.1.1

No es java

125

Índice general

CHAPTER 1

(TÍTULO CAPÍTULO 1)

1.1

(Título section 1)

(Título sub-section 1) — 3 • (Título sub-section 2) — 3

1.2

(Título section 2)

(Título sub-section 1) — 3 • (Título sub-section 2) — 3

CHAPTER 2

(TÍTULO CAPÍTULO 2)

2.1

(Título section 1)

(Título sub-section 1) — 5 • (Título sub-section 2) — 5

2.2

(Título section 2)

(Título sub-section 1) — 5 • ( sub-section 2) — 5

Índice general

5

CAPÍTULO 1  
(Título capítulo 1)

1.1.

(Título Sección1)

.....

1.1.1.

(Título sub-Sección1)

.....

1.1.2.

(Título sub-Sección2)

.....

1.2.

(Título Sección2)

.....

1.2.1.

(Título sub-Sección1)

.....

1.2.2.

(Título sub-Sección2)

.....

7

CAPÍTULO 2  
(Título Capítulo 2)

2.1.

(Título Sección1)

.....

2.1.1.

(Título sub-Sección1)

.....

2.1.2.

(Título sub-Sección2)

.....

2.2.

(Título Sección2)

.....

2.2.1.

(Título sub-Sección1)

.....

2.2.2.

( sub-Sección2)

.....

Índice general

CHAPTER 1

(TITLE CHAPTER 1)

1.1

(title section 1)

(title sub-section 1) — 3 • (title sub-section 2) — 3

1.2

(title section 2)

(title sub-section 1) — 3 • (title sub-section 2) — 3

CHAPTER 2

(TITLE CHAPTER 2)

2.1

(title section 1)

(title sub-section 1) — 5 • (title sub-section 2) — 5

2.2

(title section 2)

(title sub-section 1) — 5 • (title sub-section 2) — 5

Mini-tabla de contenidos

A veces se usa una minitabla de contenidos en cada capítulo. Esto se puede hacer con el paquete `minitoc`

Capítulo 1

Divisibilidad

Contenido de este capítulo

1.1. Algoritmo de la división

.....

3

1.1.1. Números Primos

.....

3

1.2. Números primos

.....

3

1.2.1. Criba de Eratóstenes

.....

3

1.3. MCD y MCM

.....

3

1.1. Algoritmo de la división

Figura 11.9. Mini-tabla de contenidos en cada capítulo

El resultado de la figura se obtuvo con el código:



Descargar archivo

203

```

\documentclass[12pt,a4paper,twoside,openright]{report}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[english,spanish,es-tabla]{babel}
\usepackage[charter,cal=cmcal]{mathdesign}
%
\usepackage[spanish]{minitoc}
\mtcsettitle{minitoc}{Contenido de este capítulo}
\usepackage[hidelinks]{hyperref} %después de minitoc
%
\dominitoc
\begin{document}
 \tableofcontents
 \chapter{Divisibilidad}
 \minitoc % Minitabla de contenidos
 \section{Algoritmo de la división}
 \subsection{Números Primos}
 \section{Números primos}
 \subsection{Criba de Eratóstenes}
 \section{MCD y MCM}
 \chapter{Congruencias}
 \minitoc % Minitabla de contenidos
 \section{Congruencias módulo m}
 \subsection{Trucos de divisibilidad}
 \section{Teorema Chino del resto}
 \subsection{Ejercicios}
 \section{Congruencias de Orden Superior}
\end{document}

```

## 11.6 Encabezados de página

Se puede usar el paquete **fancyhdr** para personalizar los encabezados de página y los pie de página; para cargarlo se pone en el preámbulo

```
\usepackage{fancyhdr}
```

Por ejemplo, podemos poner cabeceras en helvética (como en la figura que sigue),

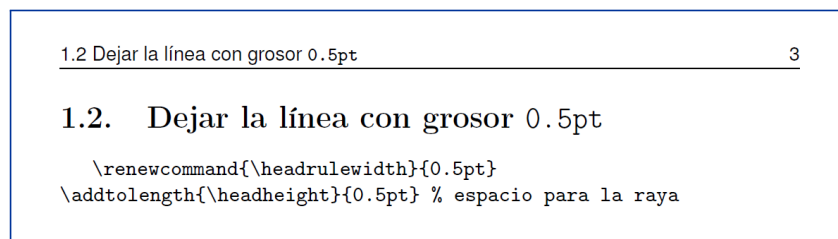


Figura 11.10. Cabecera en helvética y con línea

## 1.1. Eliminar la línea

```
\renewcommand{\headrulewidth}{0pt}
```

Figura 11.11. Cabecera en helvética y sin línea

El código que se usó para generar estas cabeceras es el siguiente,    [Descargar archivo](#)

```
\documentclass{book}
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{helvet}
% Fuente helvética
\newcommand{\helv}{\fontfamily{phv}\fontsize{9}{11}\selectfont}
%% -
\usepackage{fancyhdr}
% Usar el estilo definido
\pagestyle{fancy}
% Definir las marcas: capítulo.sección -----
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % borra cabecera y pie actuales
% El número de página
\fancyhead[LE,R0]{\helv\thepage} %L=Left, R=right, 0=Odd (impar),E=Even págs pares
% "Marcas" a la derecha e izquierda del encabezado
\fancyhead[LO]{\helv\rightmark}
\fancyhead[RE]{\helv\leftmark}
\renewcommand{\headrulewidth}{0pt} % Sin raya. Con raya?: cambiar {0} por {0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % espacio para la raya
\fancypagestyle{plain}{%
\fancyhead{} % elimina cabeceras y raya en páginas "plain"
\renewcommand{\headrulewidth}{0pt}}

\begin{document}
\chapter{Cabeceras con el paquete fancyhdr}
\newpage
\section{Eliminar la línea}
\renewcommand{\headrulewidth}{0pt}
\newpage
\section{Dejar la línea con grosor {\tt 0.5pt}}
\renewcommand{\headrulewidth}{0.5pt} % grosor 0.5pt
\addtolength{\headheight}{0.5pt} % espacio para la raya
\end{document}
```

Usando este mismo paquete también se podría incluir una figura en todas las cabeceras, digamos a la derecha, con el código,

```
%% ...
\usepackage{fancyhdr}
```

```

\usepackage{graphicx}
...
\rhead{\setlength{\unitlength}{1mm}}
\begin{picture}(0,0)
 % Figura
 \put(-10,2){\includegraphics[width=10mm]{images/ubuntu.png}}
 % Texto
 \put(-92,5){
 \parbox[t]{90mm}{
 \begin{flushright}
 \begin{scriptsize}
 \textsf{Ubuntu}
 \end{scriptsize}
 \end{flushright}
 }
 }
\end{picture}}%
\begin{document}
...

```

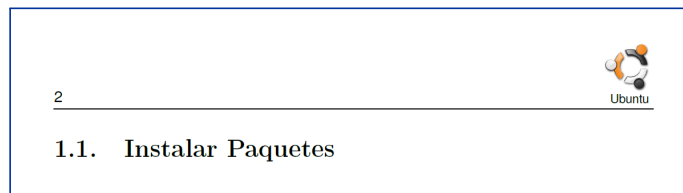


Figura 11.12. Incluir un figura en la cabecera.

### Figuras (logos) en el pie de página

Una manera fácil de agregar figuras en el pie de página es usando el comando **fancyfoot** y el paquete **TikZ**. Podríamos agregar el siguiente código en el preámbulo

```

\fancyhf{}%
\fancyfoot[LE,R0]{\tikz[baseline={(0,0)},anchor=center]
 \node[draw,inner sep=0pt,
 outer sep=0pt,label={[xshift=-.13cm,yshift=0.1cm]center:
 %\footnotesize\thepage}}]
 {\includegraphics{nombre_figura}};}%

```

Un logo centrado en el pie de página se obtiene poniendo `\fancyfoot[C]` en vez de `\fancyfoot[LE,R0]`.

### Más opciones para los encabezados

Hay una gran variedad de posibilidades para los encabezados combinando varios paquetes. Por ejemplo, puede visitar [30]

### Encabezados de este libro

Los encabezados de este libro usan los paquetes **fancyhdr**, **TikZ**, **tikzpagenodes** y **textpos**.

El **código** es (ver la sección 11.9 para tener el código completo)

```
\documentclass{book}
% ... El código completo está más adelante: sección 11.9
% ... Ver sección "Plantillas LaTeX"
% ...
% ...
\usepackage{fancyhdr}
% Números de página en rectángulos y capítulo. Posicionar los nodos
\usepackage[absolute]{textpos}
 \setlength{\TPHorizModule}{10mm}%
 \setlength{\TPVertModule}{10mm}%
 \textblockorigin{0mm}{0mm}% Origen: esquina superior izquierda

%...
%-----
% Decoración de cabeceras
% Texto en secciones
\renewcommand{\sectionmark}[1]{\markright{\sffamily\normalsize\thesection
\hspace{5pt}#1}{}}
\fancyhf{}
% Páginas con la sección a la izquierda y en rectángulo con bordes difusos
\usepackage{tikzpagenodes}
\usetikzlibrary{decorations.pathmorphing,calc,shadows.blur,shadings}
\pgfmathsetseed{1}

\pgfdeclaredecoration{irregular fractal line}{init}
{
 \state{init}[width=\pgfdecoratedinputsegmentremainingdistance]
 {
 \pgfpathlineto{\pgfpoint{random*\pgfdecoratedinputsegmentremainingdistance{
 %
 (random*\pgfdecorationsegmentamplitude-0.02)}%
 *\pgfdecoratedinputsegmentremainingdistance}}
 \pgfpathlineto{%
 \pgfpoint{\pgfdecoratedinputsegmentremainingdistance}{0pt}}}}
}

\tikzset{
 paper/.style={draw=black!10, blur shadow, shade=bilinear interpolation,
 lower left=black!20, upper left=black!15, upper right=white,
 lower right=black!10},
 irregular border/.style={decoration={irregular fractal line, amplitude
 =0.2},
 decorate,
 },
 ragged border/.style={decoration={random steps, segment length=7mm,
 amplitude=2mm}, decorate,
 }
}
```

```
%L=Left, Odd Even - Decoración en encabezado
\begin{fancyhead}[LO,LE]{\rightmark {\color{grisF}
\href{http://www.tec-digital.itcr.ac.cr/revistamatematica/}{\sffamily
\normalsize \;(http://www.tec-digital.itcr.ac.cr/revistamatematica/).}}
\begin{tikzpicture}[remember picture,overlay]
\fill[verdep,opacity=0.7]
 decorate[irregular border]{decorate{decorate{%
 decorate{decorate[ragged border]{
 ($ (current page header area.south east -|current page.east) %
 - (0, random*5mm)$) -- ($ (current page header area.south west -|
 current page.west) - (0, random*5mm)$)
 }}}}}
 -- (current page.north west) -- (current page.north east) -- cycle;
\end{tikzpicture}
\end{tikzpicture}
}% Fin decoración cabeceras
%...
```

## 11.7 Insertar una portada

Aunque podríamos hacer una portada usando el paquete **TikZ**, parece más sencillo hacer una portada, por ejemplo con **Inkscape**, guardarla en formato **.pdf** y luego insertarla usando el paquete **pdfpages**.

Hay que recordar que las fuentes es mejor, a la hora de guardar en formato **.pdf**, convertirlas a “contornos” (en el menú **Trayecto - Objeto a trayecto**) para que no haya sustitución de fuentes. También se puede exportar la portada como un archivo **.png**

En este libro, se editó la portada con **Inkscape** y se insertó con el código

```
\documentclass{book}
%Insertar páginas pdf
\usepackage{pdfpages}

\begin{document}
% Insertar Portada.pdf
\includepdf[pages=-]{images/Portada
}
% ...
```

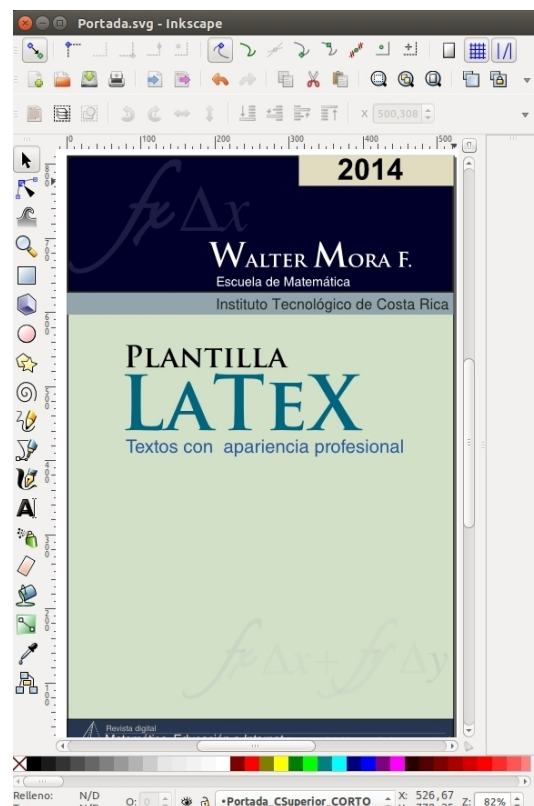


Figura 11.13. Portada en Inkscape

## 11.8 Listas de ejercicios con solución

Hay varios paquetes para hacer listas de ejercicios, por ejemplo los paquetes **answers**, **ans**, **exercise**, **ExSol**, **exsheets** y **probsoln**.

La documentación de estos paquetes se puede obtener en <http://www.ctan.org/>. En esta sección solo vamos a describir dos paquetes: **answers** y **ans**.

### 11.8.1 Paquete answers

Para hacer listas de ejercicios con solución se puede usar el paquete **answers** (hay un paquete para exámenes en **CTAN-exam**), para cargarlo se pone en el preámbulo `\usepackage{answers}`. El paquete requiere crear entornos y crear, abrir y cerrar archivos externos (en la carpeta de trabajo) con las preguntas y las soluciones de los ejercicios (archivos **ans1**, **ans2**, etc).

En el **código** que sigue se muestra un ejemplo.



#### Ejemplo 11.1 (Paquete answers)

```
\documentclass[12pt]{article}
\usepackage{amsmath,amssymb,amsfonts,latexsym}
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[shortlabels]{enumitem}

%answers-----
\usepackage{answers}
\newtheorem{ejer}{}[section]\begin{ejer}...Ejercicio... \end{ejer}
\Newassociation{sol}{Solution}{ans} %\begin{sol}... \end{sol}
%-----
\newcommand{\sen}{\mathop{\rm sen}\nolimits}
\newcommand{\R}{\mathbb{R}}
\newcommand{\Z}{\mathbb{Z}}
%-----
\begin{document}
\chapter{...}
\section{...}
%-
\Opensolutionfile{ans}[ans1] % Abrir primera lista de ejercicios
\section{Ejercicios}%Inicio lista #1
\begin{ejer}\rm Resolver $|\cos(\theta)|=1$ con $\theta \in \mathbb{R}$, $\theta \in \mathbb{R}$
\begin{sol}
 $|\cos(\theta)|=1 \iff \theta=k\pi, \quad k \in \mathbb{Z}$
\end{sol}
\end{ejer}
%- Ejercicio sin solución
\begin{ejer}\rm Resolver
\begin{enumerate}[a.]
\item $|\sin(\theta)|=1$ con $\theta \in \mathbb{R}$, $\theta \in \mathbb{R}$
\item $|\sin(\theta)|=1$ con $\theta \in \mathbb{R}$, $\theta \in \mathbb{R}$
\end{enumerate}
\end{ejer}
```



```

\end{ejer}
\Closesolutionfile{ans} % Cerrar primera lista de ejercicios

\Opensolutionfile{ans}[ans2] % Segunda lista de ejercicios
\section{Ejercicios}%Inicio lista #2
\begin{ejer}\rm Resolver $|\tan(\theta)|=1$ con $\theta \in \mathbb{R}$, \R.$\%roman
 \begin{sol}
 {\bf Sugerencia:} Mmmmm
 \end{sol}
\end{ejer}
%-----
\begin{ejer} Resolver $|\sec(\theta)|=1$ con $\theta \in \mathbb{R}$
 \begin{sol}
 $|\sec(\theta)|=1 \Rightarrow \dots$
 \end{sol}
\end{ejer}
\Closesolutionfile{ans} % Cerrar segunda lista de ejercicios
%Imprimir las soluciones
\section*{Soluciones del capítulo 1}
\input{ans1} % soluciones de la primera lista de ejercicios
\input{ans2} % soluciones de la segunda lista de ejercicios
\end{document}

```

La salida sería:

### 1.1. Ejercicios

- 1 Primer ejercicio.
- 2 Segundo ejercicio.

### Solución de los ejercicios del capítulo 1

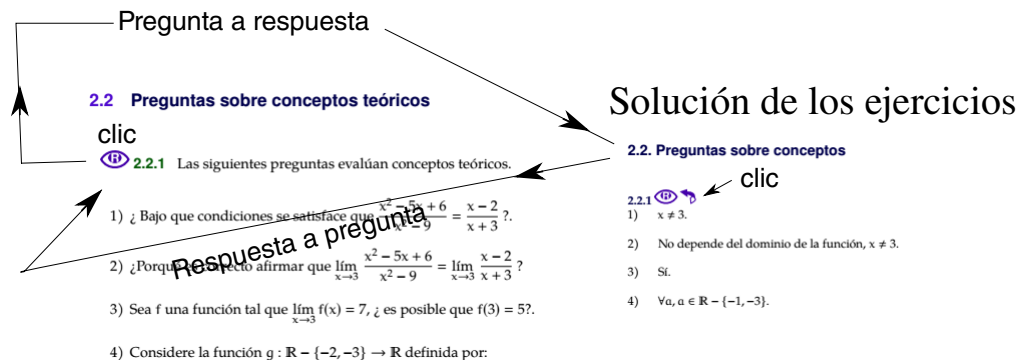
- 1 Solución del ejercicio 1
- 2 Solución del ejercicio 2

## Personalización del paquete

Para el caso de *libros* de texto, podemos simplificar el mecanismo para crear estas listas de ejercicios con entornos y comandos sin tener que estar explícitamente creando y abriendo archivos **ans**. Por ejemplo, podríamos definir un comando

```
\ejersol{ ..pregunta..}{... respuesta (opcional)...}
```

Adicionalmente, se puede saltar de la pregunta a la respuesta y viceversa



La definición sería

```
%\usepackage{answers}
\newcommand{\ejersol}[2]{%
\begin{ejer}
#1\scantokens{\begin{solu}#2\end{solu}}
\end{ejer}}
```

de tal manera que el código de las listas de ejercicios sea algo como



```
\documentclass{book}
\usepackage{answers}
...% otros paquetes
\begin{document}
\chapter{...}
\section{...}
% lista de ejercicios de la sección
\subsection{Ejercicios}
\ejersol{ pregunta 1\\ }{ respuesta1 }
\ejersol{ pregunta 2\\ }{ sugerencia }
\ejersol{ pregunta 3\\ }{ } %sin respuesta
.
.
.
%-----
% Final del documento
% Imprimir soluciones
\soluciones % Obligatorio
\solucionesCap{1}
\solucionesCap{2} %Capítulos con listas de ejercicios
...
\end{document}
```

El código que se obtiene en la la liga [Descargar archivo](#) corresponde a un libro ya preparado para recibir preguntas y respuestas de manera automática. Una descripción de la

manera de hacer esto la puede encontrar en

[http://www.tec-digital.itcr.ac.cr/revistamatematica/Secciones/Matematica\\_Algoritmos\\_Programacion/RevistaDigital\\_WMora\\_V14\\_N1\\_2013/RevistaDigital\\_walter\\_V14\\_n1\\_2013.pdf](http://www.tec-digital.itcr.ac.cr/revistamatematica/Secciones/Matematica_Algoritmos_Programacion/RevistaDigital_WMora_V14_N1_2013/RevistaDigital_walter_V14_n1_2013.pdf)

### 11.8.2 Paquete “ans”

El paquete **ans** ofrece una manera aparentemente más sencilla de crear listas de ejercicios, siempre y cuando *se respeten las reglas* del paquete!. Por ejemplo, el paquete requiere usar `\par` para cambiar de párrafo y se debe dejar un renglón al final para indicar que la respuesta ya finalizó.

La apariencia del código para las listas de ejercicios con este paquete sería algo como



Descargar archivo

```
\documentclass{book}
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{amsmath,amssymb,amsfonts,latexsym,cancel,stmaryrd,amsthm}

%-----
% Puede poner el archivo "ans.sty" en su carpeta de trabajo
\usepackage{ans} % > Debe haber al menos un ejercicio por cada sección
% > Use \exer: si usted NO va a contestar la pregunta
% > Use \exera: si usted va a contestar la pregunta
% > \subanswer ...(seguido de \exera). Use \par
% para cambiar de párrafo y dejar un renglón al final,
% éste se usa para delimitar la respuesta.
% > Subproblemas: puede usar \exer y \subexer o
% \exera y \subexera
% > \exer\annot{hard}

\makeanswers
\artsec
\makesolutions
%-----
\begin{document}
\chapter{I}
\section{Lista de ejercicios con solución }

%--Inicio de la lista de ejercicios
\exercises
\exera Determine la ecuación canónica de la parábola $y=2x^2-4x+1$.
\answer $2x^2-4x+1=y \implies 2(x-1)^2= y+1 \implies (x-1)^2=\frac{1}{2}(y+1)$.
\exera Determine la ecuación canónica de las siguientes parábolas,
```

```

\subexera $-9\,y^2 - 8\,x - 3=0$\
\subanswer $y^2=-\frac{8}{9}(x+\frac{3}{8})$
% Se debe dejar un renglón para indicar el final de la respuesta

\subexera $y^2+2 y-4 x=7$\
\subanswer $(y + 1)^2 = 4 (x + 2)$
%-

\subexera $x^2+2 x-2 y+5=0$\
\subanswer $(x + 1)^2 = 2 (y - 2)$
%-

\exer Determine la ecuación canónica de la parábola $y=4x^2-5x+1.$

\leexer
%--Fin de la lista de ejerciciosde esta sección
...

```

## 11.9 Plantillas LaTeX para libros

Para evitar el exceso de código en el preámbulo lo mejor es poner todo este código en un archivo aparte e invocarlo con el comando `\input` o implementar un archivo de estilo `.sty`

En general tendríamos que iniciar el documento con

```

\documentclass{book}
\input{Archivo_Diseno_Libro.tex}
...
\begin{document}
...
\end{document}

```

Para tal efecto hemos preparado cinco plantillas. Cada plantilla viene con un archivo de código para el preámbulo `Archivo_Diseno_Libro...` y un manual `.tex` que puede servir de plantilla para un archivo `.tex` o usar simplemente como referencia.

### Advertencia.

Las siguientes plantillas usan la versión más reciente del paquete **tcolorbox** (entre otros paquetes recientes), por lo tanto *debe actualizar los paquetes de sus distribución TeX* o instalar manualmente este paquete (ver la sección 10.4 del capítulo 10).

### La plantilla de este libro.

La plantilla para este libro viene con un código relativamente simple, es algo como

```
\documentclass[11pt,fleqn,x11names,table]{book}
% Dimensiones y márgenes.
\usepackage[text={15cm,25cm},centering,headsep=20pt,top=0.8in,
 bottom = 0.8in,letterpaper,showframe=false]{geometry}

%-- Paquetes y código de diseño-----
\input{Archivo_Disenio_Libro_LaTeX_RevistaMatematica_ITCR}
%-----
\usepackage{pdfpages} %Insertar páginas pdf
%%--
\begin{document}
% Insertar Portada.pdf
\includepdf[pages=-]{images/Portada}
\pagestyle{empty}

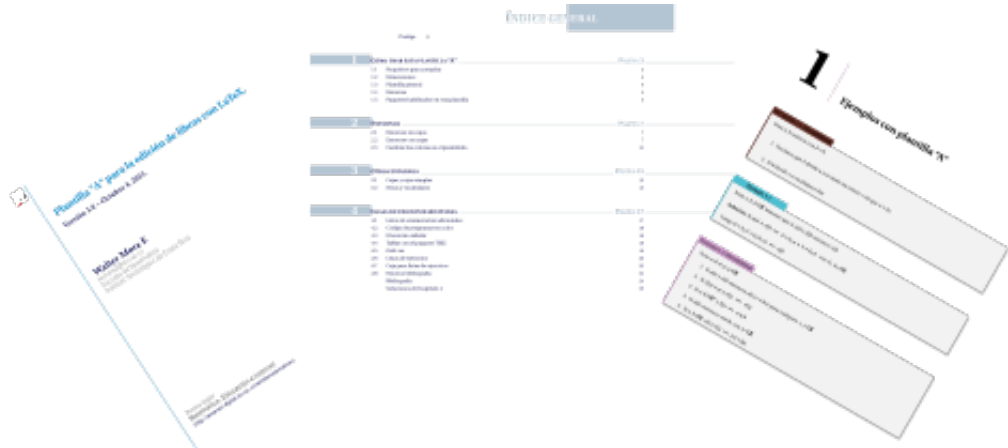
\titulo{% Autor (\fnte es un comando para un tipo de fuente especial)
 \huge\bfseries\color{wcolornotas} Alexánder Borbón A., Walter Mora F.}{
 %
 %pre-título
 \color{verdeF}Edición de Textos Científicos
 }{%
 %Título principal
 {\fontsize{80}{1} \selectfont La\TeX{}}
 }{%
 %Adicional
 Composición, Diseño Editorial, Gráficos,\,
 Inkscape, Tikz y Presentaciones Beamer
 }

\chapter{...}
```

Plantilla A

Una idea general de esta plantilla se presenta en la figura que sigue,

   [Descargar archivo](#)



Plantilla B

Una idea general de esta plantilla se presenta en la figura que sigue,

   Descargar archivo

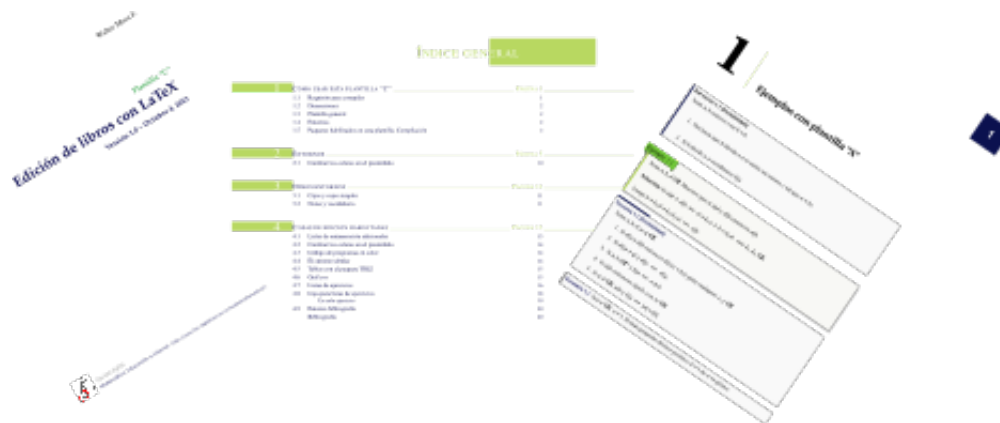


215

## Plantilla C

Una idea general de esta plantilla se presenta en la figura que sigue,

   Descargar archivo



## Plantilla D

Una idea general de esta plantilla se presenta en la figura que sigue,

   Descargar archivo



### 11.9.1 Otras Plantillas (templates) $\text{\LaTeX}$

Otras plantillas ("Templates")  $\text{\LaTeX}$  se pueden obtener en Internet.

- 1 **LaTeX Templates** (<http://www.latextemplates.com/>) es un sitio con varias plantillas. Por ejemplo,

|                                 |                         |                         |
|---------------------------------|-------------------------|-------------------------|
| Tareas (Assignments)            | Libros (Books)          | Calendarios (Calendars) |
| Hojas de vida (Curricula Vitae) | Ensayos (Essays)        | Laboratorios            |
| Presentaciones (Presentations)  | Reportes de Laboratorio | Cartas (Letters)        |
| Títulos (Title Pages)           | Tesis (Theses)          | etc.                    |

- 2 **LaTeX for Humans** (<http://latexforhumans.wordpress.com/>). Entre otras cosas, en este sitio hay muchas plantillas de tesis gratuitas. En particular, se pueden encontrar plantillas de tesis de universidades Europeas y americanas.

- 3 **\howTeX** (<http://www.howtotex.com/>). Este es un sitio con muchas plantillas  $\text{\LaTeX}$  de propósito general.

Por supuesto, las búsquedas en Internet sobre alguna plantilla en particular seguramente nos dará muchos resultados.

## 11.10 Libro con algoritmos y programas

### 11.10.1 Paquete `algorithm2e` (versión 2020.)

Este es un paquete adecuado para describir claramente algoritmos de programación. Debemos poner en el *preámbulo*

```
\usepackage[ruled,vlined,lined,linesnumbered,algochapter,spanish]{algorithm2e}
```

Los comandos más frecuentes son

```
\If{ condición }{ código }
\ElseIf{ condición }{ código }{ else ... código }
\For{ condición }{ código }
\While{ condición }{ código }
\Repeat{ condición ("Until") }{ código }
\; se usa para el cambio de línea.
```

En los siguientes ejemplos se muestra cómo usar **If**, **For**, **While**, etc.

El **código**:

```
% Nuevo en la versión 2020
% Incompatible con versiones antiguas
\begin{algorithm}[H] % H = forzar está posición
\caption{Máximo común divisor}\label{ML:Algorithm1}
\SetAlgoLined
```

```

\KwData{$a,\, b \in \mathbb{N}.}
\KwResult{MCD$(a,b)}
\LinesNumbered
\SetAlgoVlined
$c = |a|,\; d = |b|;
\While{$d \neq 0}{
 $r = \text{rem}$(c,d);
 $c = d;
 $d = r;
}
\Return MCD$(a,b)=|c|;
\end{algorithm}

```

produce:

---

**Algoritmo 11.1:** Máximo común divisor

---

**Data:**  $a, b \in \mathbb{N}$ .

**Result:**  $\text{MCD}(a, b)$

```

1 $c = |a|, d = |b|;$
2 while $d \neq 0$ do
3 $r = \text{rem}(c, d);$
4 $c = d;$
5 $d = r;$
6 return $\text{MCD}(a, b) = |c|;$

```

---

El código:

```

% Nuevo en la versión 5.0
% Incompatible con versiones antiguas
\begin{algorithm}[H]
\caption{Inverso Multiplicativo mod m .}
\SetAlgoLined
\KwData{$a \in \mathbb{Z}_m$}
\KwResult{$a^{-1} \pmod{m}$, si existe.}
\LinesNumbered
\SetAlgoVlined
Calcular x, t tal que $xa+tm=\text{MCD}(a,m)$;
\If{$\text{MCD}(a,m)>1$}{ $a^{-1} \pmod{m}$; m no existe}
\Return $\text{rem}\backslash, (x,m).$
\end{algorithm}

```

produce:



**Algoritmo 11.2:** Inverso Multiplicativo mod  $m$ .**Data:**  $a \in \mathbb{Z}_m$ **Result:**  $a^{-1} \bmod m$ , si existe.

```

1 Calcular x, t tal que $xa + tm = \text{MCD}(a, m)$;
2 if $\text{MCD}(a, m) > 1$ then
3 | $a^{-1} \bmod m$ no existe
4 else
5 | return $\text{rem}(x, m)$.
```

Por ejemplo,

**Algoritmo 11.3:** Criba de Eratóstenes**Data:**  $n \in \mathbb{N}$ **Result:** Primos entre 2 y  $n$ 

```

1 $\text{max} = \left\lceil \frac{(n-3)}{2} \right\rceil$;
2 boolean $\text{esPrimo}[i]$, $i = 1, 2, \dots, \text{max}$;
3 for $j = 1, 2, \dots, \text{max}$ do
4 | $\text{esPrimo}[j] = \text{True}$;
5 $i = 0$;
6 while $(2i + 3)(2i + 3) \leq n$ do
7 | $k = i + 1$;
8 | if $\text{esPrimo}[i]$ then
9 | while $(2k + 1)(2i + 3) \leq n$ do
10 | $\text{esPrimo}[(2k + 1)(2i + 3) - 3]/2] = \text{False}$;
11 | $k = k + 1$;
12 | $i = i + 1$;
13 Imprimir;
14 for $j = 1, 2, \dots, \text{max}$ do
15 | if $\text{esPrimo}[j] = \text{True}$ then
16 | Imprima j
```

Se obtiene con el **código**:

```

% Nuevo en la versión 5.0 (2013)
% Incompatible con versiones antiguas
\begin{algorithm}[H]
\caption{Criba de Eratóstenes} \label{CribaEraostenes}
\SetAlgoLined
\KwData{n \in $\mathbb{N}$}
\KwResult{Primos entre 2 y n}
\LinesNumbered
max$=\left\lceil \frac{(n-3)}{2} \right\rceil$;
boolean esPrimo$[i]$, $i=1,2,\dots,max$;
\SetAlgoVlined
\For{$j=1,2,\dots,max$}{ esPrimo$[j]=$True;}
$i=0$;
\While{$(2i+3)(2i+3) \leq n$}{
```

```

 $k=i+1$\;
 \If{\rm esPrimo$[i]}{
 \While{$(2k+1)(2i+3) \leq n$}
 {
 esPrimo$[((2k+1)(2i+3)-3)/2]=$False\;
 $k=k+1$\;
 }}
 $i=i+1$\;
\Imprimir\;
\For{$j=1,2,\dots,max$}{
\If{\rm esPrimo$[j]=$True}{Imprima j } }
\end{algorithm}

```

## 11.11 Color para el código

Para agregar jerarquía al código se debe usar indentación y color. Hay varios paquetes para agregar color al código, por ejemplo **minted**, **verbments** y **listings**. Los paquetes **minted** y **verbments** usan programas externos (**Python** y **Pygments**) y requiere instalar estos programas. **Listings** no requiere nada adicional. En este libro se usa **listings** (y un poco de **minted**).

Con estos paquetes se elige el lenguaje (**C++**, **java**, **latex**, **etc.**) y permiten también, entre otras opciones, usar código  $\text{\LaTeX}$  para generar texto matemático para documentar el código.

### 11.11.1 Paquete Listings

Este paquete para resaltar código viene incluido en **TeXLive** y en **MikTeX** y no necesita una instalación adicional. Para usarlo debemos agregar en el *preámbulo*,

```
\usepackage{listings}
```

El entorno es como sigue,

```

\begin{lstlisting}[opciones]
... código...
\end{lstlisting}

```

También se puede usar un comando muy útil para incluir código en el texto. El comando es `\lstinline` y se usa como el comando `\verb`.

Las opciones se pueden declarar en el *preámbulo*, como se puede ver en el ejemplo que sigue,

#### Ejemplo 11.2 (Usando Listings).

En este código se muestra como usar el paquete **listings**    [Descargar archivo](#)

```

\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage{amsmath,amsfonts,amssymb}
\usepackage{xcolor}

```

```
%-----
\usepackage{listings}%
\lstset{ %
 language=Pascal, % lenguaje
 basicstyle=\bfseries\ttfamily,
 keywordstyle=\color{blue},
 commentstyle=\color{brown},
 backgroundcolor=\color{gray!90},
 showstringspaces=false
}

%-----
\begin{document}

Usamos la instrucción \lstinline|for i:= maxint to 0 do|.\

\begin{lstlisting}

 for i:= maxint to 0 do
 begin{
 do nothing
 }
 end;
 Write('Case insensitive');
 Write('Pascal keywords');

\end{lstlisting}
\end{document}
```

**produce:**

Usamos la instrucción `for i:= maxint to 0 do`.

```
for i:= maxint to 0 do
begin{
 do nothing
}
end;
Write('Case insensitive');
Write('Pascal keywords');
```

En este libro usamos **listings** con el paquete **beramono** (para las fuentes) con las opciones,

```
\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage{fourier}
\usepackage[scaled=0.83]{beramono}
\usepackage{amsmath,amsfonts,amssymb}
```

```

\usepackage{xcolor}
\lstset{ %
 language={LaTeX}TeX}, % lenguaje
 basicstyle=\bfseries\ttfamily,
 keywordstyle=\color{blue},
 commentstyle=\color{brown},
 backgroundcolor=\color{gray!15},
 showstringspaces=false,
 flexiblecolumns=true,
 stringstyle=\ttfamily\color{blue},
 extendedchars=true,
 emph={rm,bf,it,sf}, %...
 literate=%
 *{$}{{{ \color{red}}$}}1 % produce $en rojo
 {$$}{{{ \color{red}}$ $}}1
 % No dejar renglones en blanco
}%
\begin{document}
...
\end{document}

```

### 11.11.2 Paquete minted

#### Instalar minted en Ubuntu

Para usar este paquete en **Ubuntu** debemos hacer algunos cosas previamente.

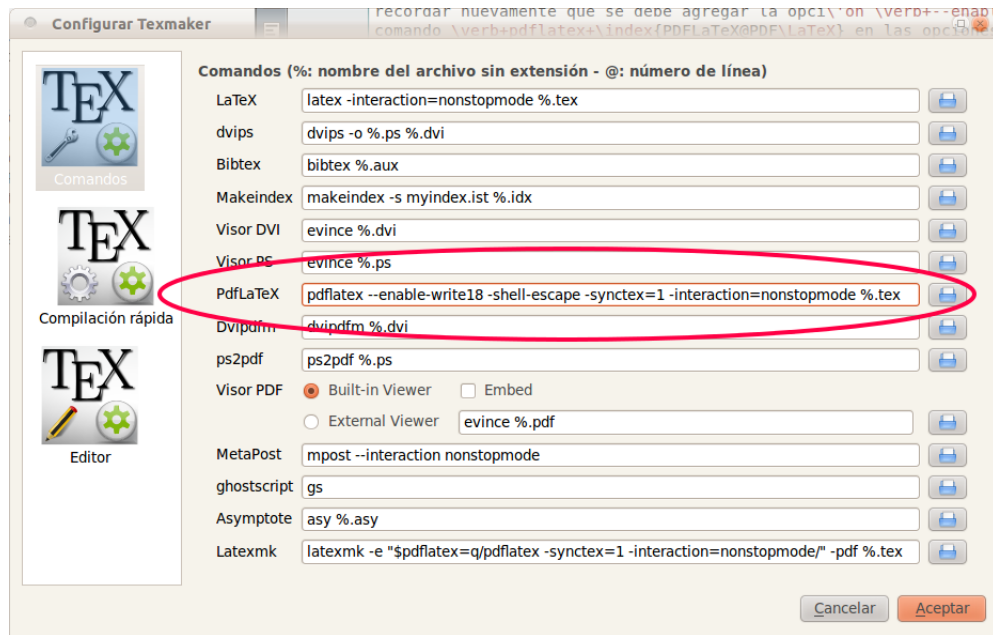
- Instalar **Pygments**: Se ejecuta en una terminal el código

```
sudo sudo apt-get install python-pygments
```

- **minted.sty** ya está presente en las distribuciones **TeXLive** y **MikTeX**. Si no lo tiene lo puede descargar [aquí](#). Ver el apéndice ??.
- Habilitar **shell-scape**: En la configuración de su editor debe buscar la opción **PdfLaTeX** (o **LaTeX**) y agregar **-shell-escape** de tal manera que quede

```
PdfLaTeX -synctex=1 -interaction=nonstopmode -shell-escape '%source'
```

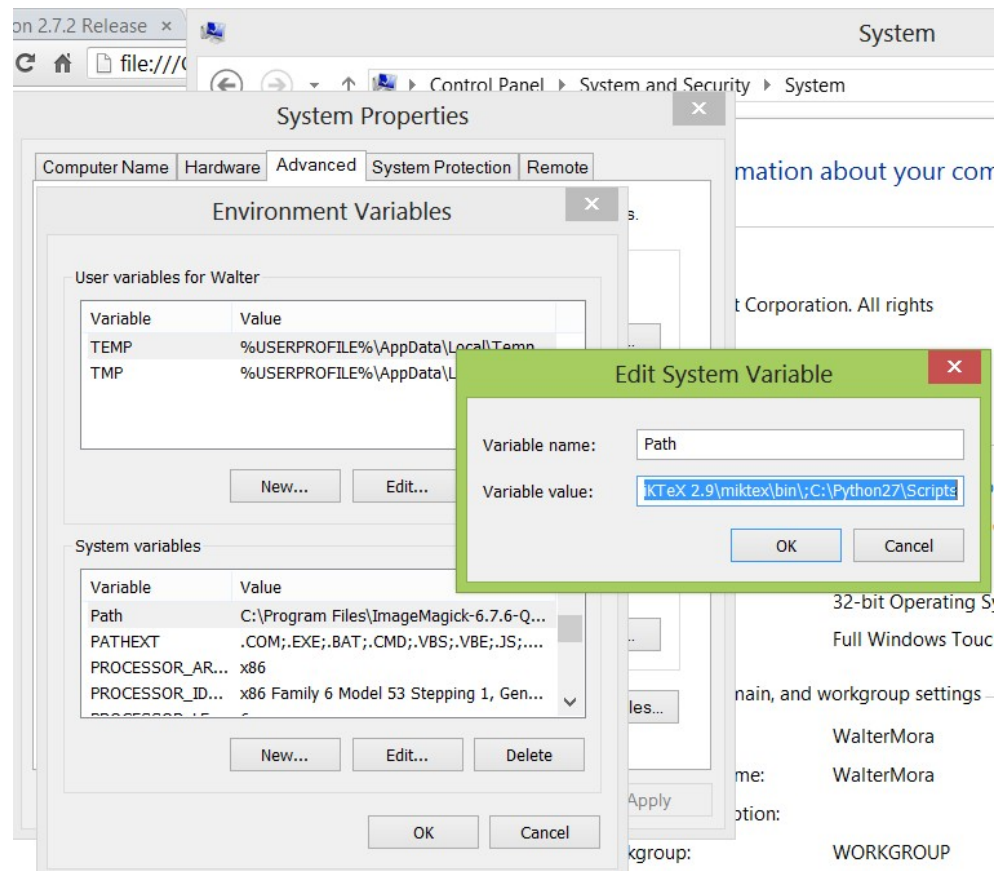
Por ejemplo, en **TeXMaker** se vería así,



## Instalar en Windows

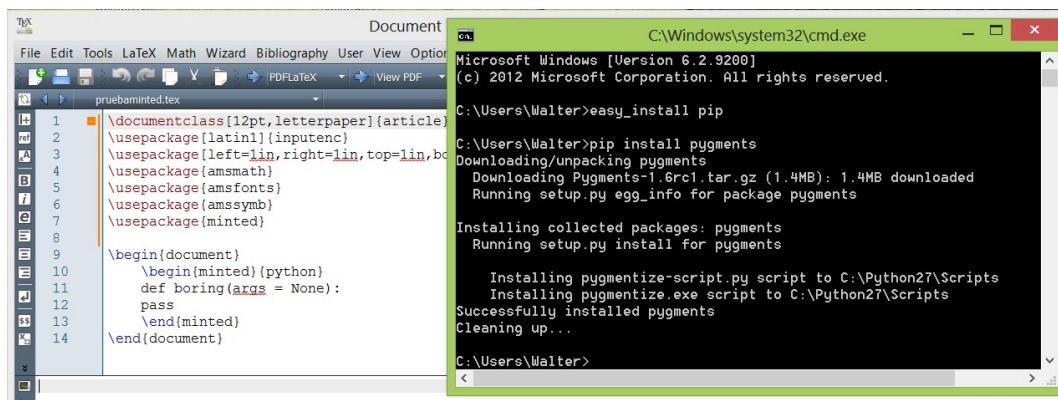
Para usar este paquete en **Windows** debemos hacer algunas cosas previamente (estas instrucciones siguen funcionando en Windows 10, instalar otra versión de Python requiere otros ajustes).

- Instalar **Python 2.7.2** desde <http://www.python.org/>. Se ejecuta **python-2.7.2.msi**
- Descargar y ejecutar **distribute\_setup.py** en [http://python-distribute.org/distribute\\_setup.py](http://python-distribute.org/distribute_setup.py)
- Editar **Variables de Entorno**: Ir al '**Panel de Control**', luego navegar las ventanas en este orden: **System and Security**, **System**, **Advanced Systems Settings** y finalmente **Advanced**, en esta última ventana presiona **Environment Variables**. En esta ventana va a la subventana **System variables**, selecciona **Path** y presiona el botón **Edit**. Al final de la línea pone un 'punto y coma' (;) y agrega ;C:\Python27\Scripts y cierra todas las ventanas presionando OK.



- Abrir una terminal (en Windows se puede buscar como **cmd** desde la campo de búsqueda en el botón Inicio o el ícono de búsqueda en Windows 8)
- En la terminal se ejecuta el siguiente código (uno a la vez, con Enter),  

```
easy_install pip
pip install pygments
```



- Habilitar **shell-scape**: En la configuración de su editor debe buscar la opción PdfLaTeX (o LaTeX) y agregar **-shell-escape** de tal manera que quede

PdfLaTeX -synctex=1 -interaction=nonstopmode -shell-escape '%source'

- **minted.sty** ya está presente en las distribuciones **TeXLive** y **MikTeX**. Si no lo tiene lo puede descargar [aquí](#). Ver el apéndice ??.

Una vez completada la instalación, se debe poner en el preámbulo `\usepackage{minted}`. Adicionalmente se puede cambiar el *estilo* (que es el que define el esquema de color). Por ejemplo, en este libro usamos el estilo **vs**. Para cambiar el tema default y usar este tema, se pone en el preámbulo `\usemintedstyle{vs}`. El entorno es el siguiente,

```
\begin{minted}[opciones]{lenguaje}
...
código
...
\end{minted}
```

La opción **mathescape** permite introducir código **LaTeX** *solo después* del símbolo de comentario del lenguaje que se está usando.

En el ejemplo que sigue, el lenguaje que se declara es **python**. La opción **mathescape** permite aplicar texto en modo matemático en el entorno y *debe ir precedido por el símbolo de comentario del lenguaje*. El paquete **minted** usa la librería **Pygments** así que si quiere saber más de **minted** hay que ver la documentación del paquete **minted** y la documentación de **Pygments**.

En el ejemplo que sigue se muestran otras opciones.

### Ejemplo 11.3 (Usando minted).

El texto:



```
\documentclass{article}
\usepackage{amsmath}
\usepackage[utf8]{inputenc}
\usepackage{minted} % minted requiere Python, Pygments
 % y habilitar 'shell escape'
\usemintedstyle{vs} % Estilo vs

\begin{document}
\begin{minted}[mathescape, frame=lines, fontseries=b,
 rulecolor=\color{gray}]{python}

Suma de los primeros naturales,
Returns $\sum_{i=1}^n i$

def sum_from_one_to(n):
 r = range(1, n + 1)
 return sum(r)
```

```
\end{minted}
\end{document}
```

produce:

```
Suma de los primeros naturales,
Returns $\sum_{i=1}^n i$
def sum_from_one_to(n):
 r = range(1, n + 1)
 return sum(r)
```

### 11.11.3 Paquete verbments

La instalación sigue los mismos pasos que la instalación de **minted**. Este paquete es parecido a **minted** pues está basado también en **Pygments**, así que las opciones serán familiares. Tiene la ventaja de que el código se divide si queda entre dos páginas y se puede poner el tema directamente en las opciones. También las opciones se pueden declarar al inicio del documento. Para usar el paquete debemos poner en el preámbulo `\usepackage{verbments}`. El entorno sería algo como,

```
\begin{pyglist}[language=java,caption={Nombre del programa},otras opts]
... código
\end{pyglist}
```

Cuando se usan muchas opciones, como en el ejemplo que sigue, es mejor declararlas antes con los comandos **fvset** y **plset**.

**Nota:** Igual que **minted**, puede agregar texto en modo matemático. El texto matemático debe de ir precedido por el símbolo usado para los comentarios, en el caso de Java sería `// $ ....$`). Debe agregar la opción **mathescape=true**.

**Nota:** Este paquete viene con la versión **TeXLive** más reciente. Si tiene una versión anterior, debe obtener la versión del archivo **verbments.sty** del 2011 o la más reciente. El archivo **verbments.sty** se puede obtener en <http://texcatalogue.sarovar.org/entries/verbments.html>

#### Ejemplo 11.4 (Usando verbments).

Este ejemplo supone que se va a usar el lenguaje **java**. El código:

```
\documentclass{article}
\usepackage{verbments}%Requiere Python, Pygments y Habilitar 'shell escape'
\definecolor{verbmentsbgcolor}{rgb}{0.9764, 0.9764, 0.9762}
\definecolor{verbmentscaptionbgcolor}{rgb}{0.1647, 0.4980, 1}
\begin{document}
%opciones verbments

\fvset{frame=bottomline,framerule=0.01cm}
\plset{language=java,texcl=true,style=vs,%
 listingnamefont=\sffamily\bfseries\color{white},%
```



```

 bgcolor=verbmentsbgcolor,captionfont=\sffamily\color{white},%
 captionbgcolor=verbmentscaptionbgcolor, listingname=\textbf{Programa}}
%Entorno -----
\begin{pyglist}[caption={Imprimir.java}]
import javax.swing.*;
import java.awt.*;
 public class app_prg1 extends JApplet
 { public void init(){}
 public void paint (Graphics g){
 g.drawString(" 3 +46 = "+(3+46),30, 30);}
 }
\end{pyglist}
\end{document}

```

produce:

#### Programa 11.1: Imprimir.java

```

import javax.swing.*;
import java.awt.*;
 public class app_prg1 extends JApplet
 { public void init(){}
 public void paint (Graphics g){
 g.drawString(" 3 +46 = "+(3+46),30, 30);}
 }

```

## 11.12 LaTeX, R y Knitr

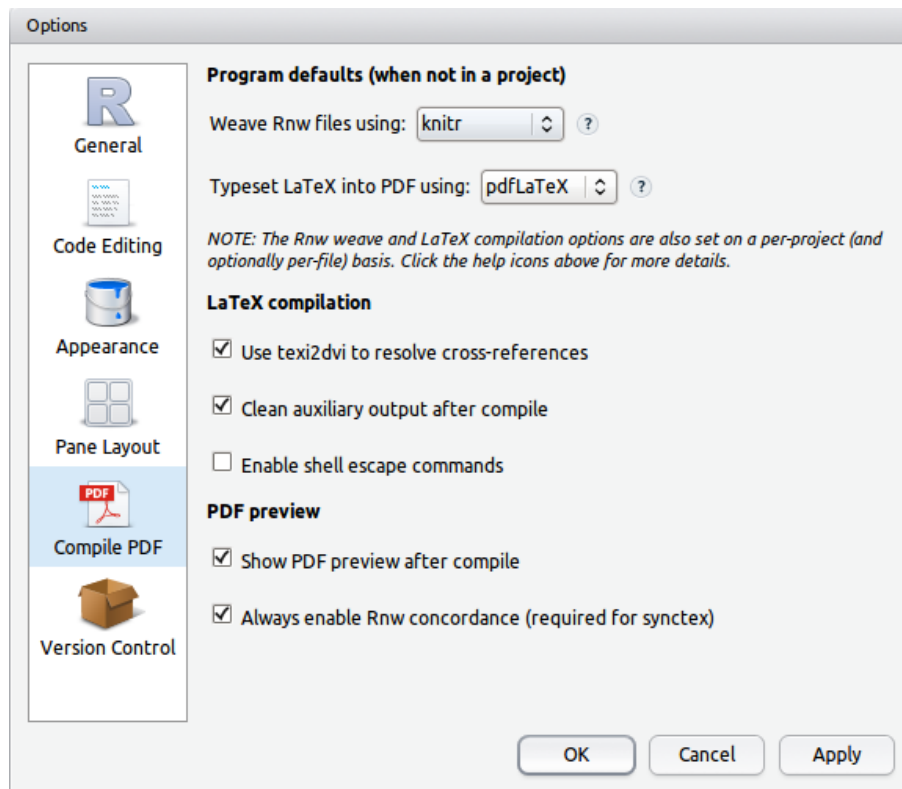
El paquete **knitr** permite combinar código **R** y figuras en documentos **LaTeX**. Estos documentos son archivos **.Rnw**.

El proceso de “tejer” (weaving) un archivo **.Rnw**, implica la ejecución de código **R** insertado en el documento **LaTeX**. El paquete **knitr** se puede usar integrado con **RStudio**.

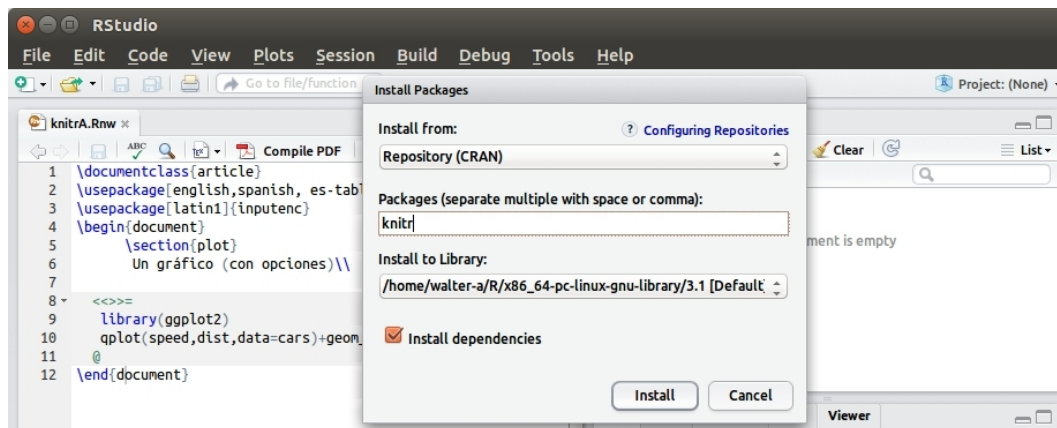
**Knitr** extrae código **R** en el documento de entrada, lo evalúa y escribe los resultados en el documento de salida. Hay dos tipos de código **R** que podemos insertar en un documento **LaTeX**, trozos (chunks) de código (como párrafos separados) y código **R** en línea.

### Instalar el paquete Knitr

El código **LaTeX** y los trozos (“chunks”) de código **R** insertados, se editan en un archivo **.Rnw**. Hay que indicarle a **RStudio** que use **knitr** para correr e incluir (“weaving”) estos archivos. Debemos ir al menú **Tools - Global Options - Swave** y elegir en la pestaña **Weave Rnw files using** la opción **knitr**.



Luego instalamos el paquete **knitr** de la manera usual: **Tools-install Packages**.



## Editar

Para editar en **RStudio**, se abre un archivo nuevo “**R Sweave**” (o un archivo **.Rnw**) y los trozos de código **R** se insertan entre las líneas

```
<< opciones >>=
... código R
@
```

Por ejemplo, si instalamos el paquete **ggplot2**, podemos compilar el código

```
% Abrir en RStudio y guardar como nombre.Rnw
```

```

\documentclass{book}
\usepackage[english,spanish, es-tabla]{babel}
\usepackage[utf8]{inputenc}
\begin{document}
 \chapter{Knitr}
 \section{plot}
 Un gráfico
 % Insertar código R
 <<=>=
 library(ggplot2)
 qplot(speed,dist,data=cars)+geom_smooth()
 @
\end{document}

```

y obtenemos

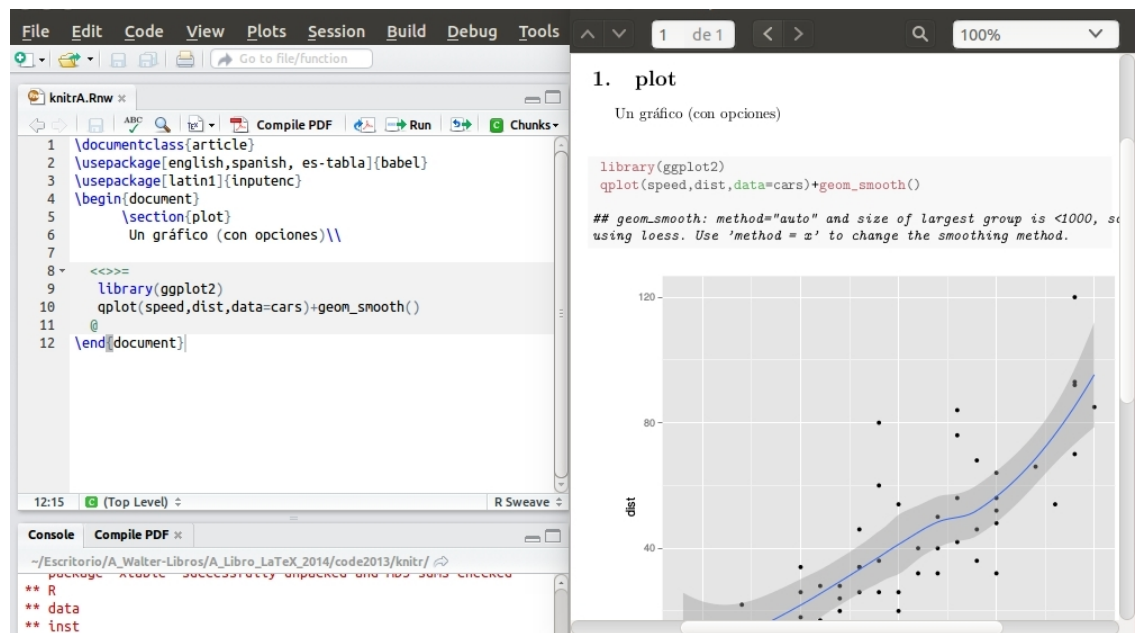


Figura 11.14. LaTeX y R en RStudio con Knitr

También podemos agregar opciones al gráfico del ejemplo anterior,

```

<<echo=FALSE, out.width=5, fig=TRUE, include=TRUE>>=
library(ggplot2)
qplot(speed,dist,data=cars)+geom_smooth()
@

```

Las opciones permitidas se pueden ver en <http://yihui.name/knitr/options/>.

### Variables y código R en el texto.

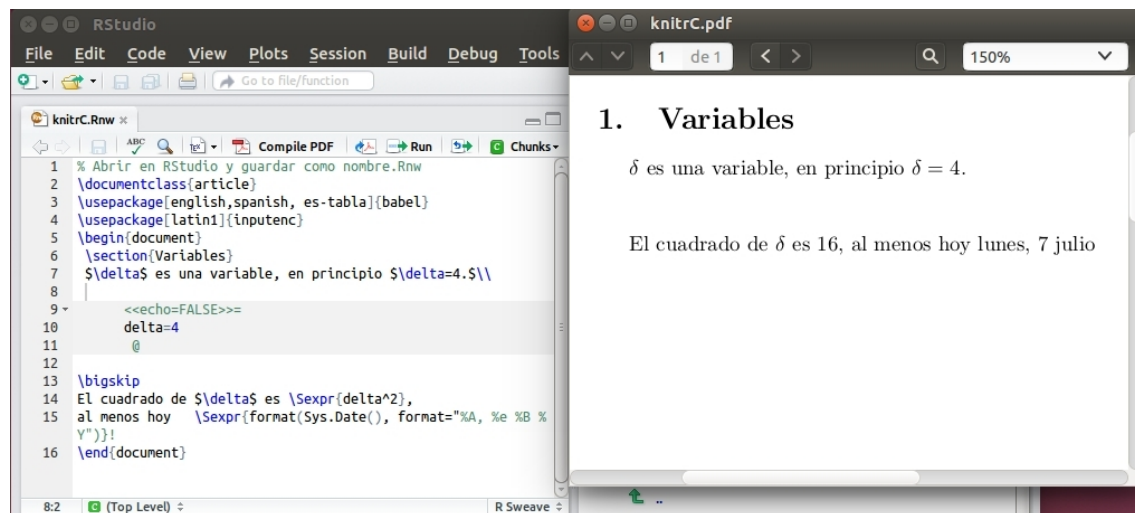
El siguiente código muestra cómo se combina el código **LaTeX** y el código **R** en otro nivel. El comando `\Sexpr` ejecuta código **R** en el texto, en este caso una operación con una variable

definida previamente ( $\delta$ ).



```
% Abrir en RStudio y guardar como nombre.Rnw
\documentclass{book}
\usepackage[english,spanish, es-tbla]{babel}
\usepackage[utf8]{inputenc}
\begin{document}
\section{Variables}
 δ es una variable, en principio $\delta=4$.

<<echo=FALSE>>=
delta=4
@
\bigskip
El cuadrado de δ es δ^2 ,
al menos hoy $\text{format}(\text{Sys.Date()}, \text{format}=\ "%A, \%e \%B \%Y")$!
\end{document}
```

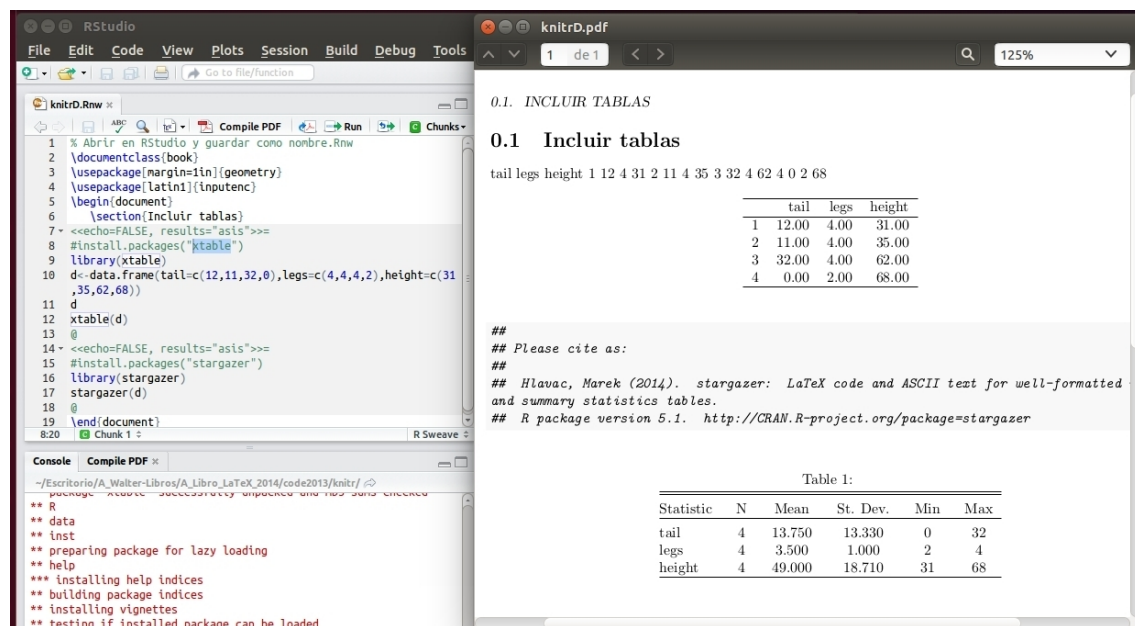


## Incluir tablas con xtable y stargazer

**stargazer** es un paquete **R** que crea código **LaTeX** para tablas de regresión correctamente formateadas, así como cuadros estadísticos de resumen.

```
% Abrir en RStudio y guardar como nombre.Rnw
% Instalar los paquetes xtable y stargazer
\documentclass{article}
\usepackage[margin=1in]{geometry}
\usepackage[utf8]{inputenc}
\begin{document}
\section{Incluir tablas}
<<echo=FALSE, results="asis">>=
#install.packages("xtable")
library(xtable)
```

```
d<-data.frame(tail=c(12,11,32,0),legs=c(4,4,4,2),height=c(31,35,62,68))
d
xtable(d)
@
<<echo=FALSE, results="asis">>=
#install.packages("stargazer")
library(stargazer)
stargazer(d)
@
\end{document}
```



## Archivos de diseño

Al compilar un archivo **.Rnw** se produce también un archivo **.tex**. Para agregar diseño al documento **.tex** se puede usar este último archivo, agregar el paquete de diseño, y compilar en un editor dedicado (Texmaker, Kile, etc.).

Al agregar el archivo de diseño en el preámbulo hay que tener el cuidado de que el nombre no tenga guiones bajos.

```
\documentclass{book}
\usepackage[margin=1in]{geometry}
\input{RevistaMatematicaITCREstiloLibroLaTeXB.tex} % sin guiones bajos...
%\usepackage[english,spanish, es-tabla]{babel}
%\usepackage[utf8]{inputenc}
\begin{document}
```



## 12.1 Introducción

“**Beamer**” es una clase **LaTeX** (`\documentclass{beamer}`) que se usa para generar transparencias para presentaciones (al estilo Power Point). Se compila con **LaTeX+dvips** o **PDFLaTeX** y se usa código **LaTeX** estándar.

La versión actual (diciembre 2013) es **Beamer 3.33**. **Beamer** viene incluido en **TeXLive** y en **MikTeX**. Si tiene una instalación completa, puede usar **Beamer** sin problemas. La documentación la puede encontrar en ‘The Beamer Class’.

Un documento **Beamer** consiste de una sucesión de marcos (*frames*). En el caso más simple, un marco solo contiene una transparencia. Un ejemplo de transparencia (usando el tema personalizado) se ve en la figura 12.1.

Instituto Tecnológico de Costa Rica  
Método de Newton.

Escuela de Matemática - CRV  
CIDSE

- 1 ¿Qué significa “suficientemente cercano” a  $x^*$ ?
- 2 Se puede aclarar en el sentido del teorema

**Teorema**

Sea  $x^*$  un cero simple de  $f$  y sea  $I_\varepsilon = \{x \in \mathbb{R} : |x - x^*| \leq \varepsilon\}$ .  
 Asumamos que  $f \in C^2(I_\varepsilon)$  y que  $\varepsilon$  es lo suficientemente pequeño de tal manera que en este intervalo  $|f'(x)| \geq m > 0$  y existe  $M > 0$  tal que  $|f''(x)| \leq M$ . Si en el método de Newton, algún  $x_j$  satisface

$$x_j \in I_\varepsilon, \quad y \quad |x_j - x^*| < 2 \frac{m}{M} \quad (1)$$

entonces  $\lim_{n \rightarrow \infty} x_n = x^*$

W. Mora F.      Métodos Híbridos      3 / 13

Figura 12.1. Transparencia **Beamer**.

## 12.2 Un documento Beamer

La estructura general de un posible documento **Beamer** podría ser,

```
\documentclass{beamer}
\usepackage{amsmath,amssymb,amsfonts,latexsym,stmaryrd}
\usepackage[latin1]{inputenc}
```

```

\usepackage[T1]{fontenc}
% Conversión eps to pdf, requiere habilitar shell-escape.
% Texlive 2010 o superior No lo necesita
%\usepackage{epstopdf}

%\DeclareGraphicsExtensions{.pdf,.png,.jpg}
\usefonttheme{professionalfonts} % fuentes de LaTeX
\usetheme{Warsaw} % Tema escogido en este ejemplo
\setbeamercovered{transparent} % Velos

\newtheorem{teo}{Teorema}
\newtheorem{ejemplo}{Ejemplo}
\newtheorem{defi}{Definición}
\newtheorem{coro}{Corolario}
\newtheorem{prueba}{Prueba}

\begin{document}

\title{Factorización en $Z_p[x]$ y en $Z[x].\$\\}$
\subtitle{Parte I}
\author{{\bf Prof. Walter Mora F.}\\
 {Escuela de Matemática, ITCR}\\
 {http://www.tec-digital.itcr.ac.cr/revistamatematica/}\\
 \vspace*{0.5cm}}
\date{Junio 2007}
\frame{\titlepage}

\section{Primera sección}

\begin{frame}{Título de este marco}
 % texto
\end{frame}

\begin{frame}{Título de este marco}
 % texto
\end{frame}

\end{document}

```

## Compilar

- Un documento **Beamer** se puede compilar directamente con **PDFLaTeX** o con **LaTeX-dvips**
  - Si instaló la distribución **TeXLive 2009** o menos o **MikTeX2.9** o menos y si compila con **PDFLaTeX**, no habrá problemas con las figuras en formatos **.pdf**, **.jpg**, **.png**.  
  
Si necesita usar figuras **.eps** podría agregar el paquete **epstopdf** y habilitar **shell escape**. Ver apéndice ??.
  - Si instaló la distribución **TeXLive 2010** o superior, y si compila con **PDFLaTeX**, no habrá problemas con las figuras en formatos **.pdf**, **.jpg**, **.png**, **.eps**.



En ambos casos podría, si le parece conveniente, declarar las extensiones para incluir figuras sin hacer referencia a la extensión.

```
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

- Si compila con **LaTeX** puede usar figuras **.eps** (hay paquetes que permiten habilitar otros formatos como **.svg**, etc.)

La primera transparencia del código anterior se muestra en la figura 12.2

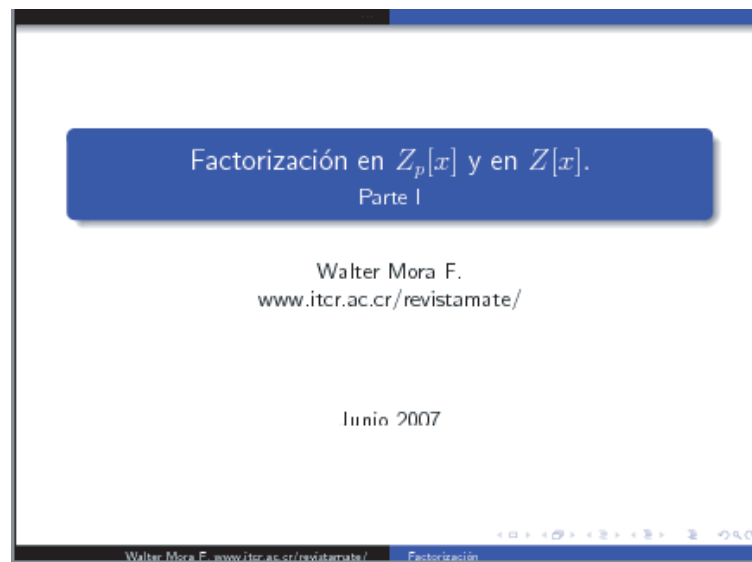


Figura 12.2. Transparencia **Beamer**. Tema Warsaw

- **Beamer** carga (por default) el paquete '**graphicx**' para el soporte de gráficos.
- La lista parcial de temas que viene con **Beamer** es

```
\usetheme{Bergen} \usetheme{Hannover}
\usetheme{Boadilla} \usetheme{Luebeck}
\usetheme{Warsaw} \usetheme{AnnArbor}
```

- Los temas de la versión actual (2020) se pueden ver en la documentación del paquete, también en <https://hartwork.org/beamer-theme-matrix/> y hay más temas en <https://es.overleaf.com/gallery/tagged/presentation>
- En Internet hay otros tantos temas **Beamer** disponibles, usualmente temas de particulares o instituciones pero de distribución libre. Los temas se pueden personalizar (ver [18]).

## 12.3 Marcos (frames)

El uso más común de un marco (frame) es poner una lista de ítems

### Ejemplo 12.1

```
\begin{frame}
 \frametitle{Campo Galois $GF(p^r)$ }
 \framesubtitle{Resumen}
 \begin{enumerate}
 \item Todo dominio integral {\em finito} es un campo\\
 \item Si F es un campo con q elementos, y a es un elemento no nulo de F , entonces $a^{q-1}=1$ \\
 \item Si F es un campo con q elementos, entonces cualquier $a \in F$ satisface la ecuación $x^q - x = 0$
 \end{enumerate}
\end{frame}
```

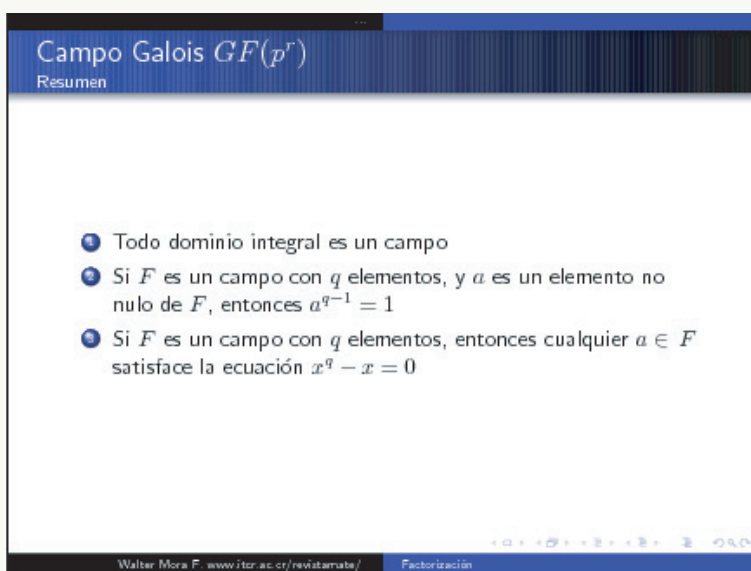


Figura 12.3. Marco Beamer. Tema Warsaw

## 12.4 Velos (overlays)

En una presentación puede ser deseable que los ítems vayan apareciendo uno a la vez, mientras los otros permanecen con un *velo* (formalmente “una capa”). Esto se puede lograr agregando la opción [ $\langle + - \rangle$ ] a los entornos **enumerate** o **itemize**. Un ejemplo se muestra en la figura 12.4.

## Ejemplo 12.2

```

\begin{frame}
 \frametitle{Campo Galois $GF(p^r)$ }
 \framesubtitle{Resumen}
 \begin{enumerate}[<+>] % <- Nueva opción
 \item Sea F un campo con q elementos y a un elemento no nulo de F . Si n es el orden de a , entonces $n|(q-1)$.
 \item Sea p primo y $m(x)$ un polinomio irreducible de grado r en $\mathbb{Z}_p[x]$. Entonces la clase residual $\mathbb{Z}_p[x]/\equiv_{m(x)}$ es un campo con p^r elementos que contiene $\mathbb{Z}_p$ y una raíz de $m(x)$.
 \item Sea F un campo con q elementos. Entonces $q=p^r$ con p primo y $r \in \mathbb{N}$
 \end{enumerate}
\end{frame}

```

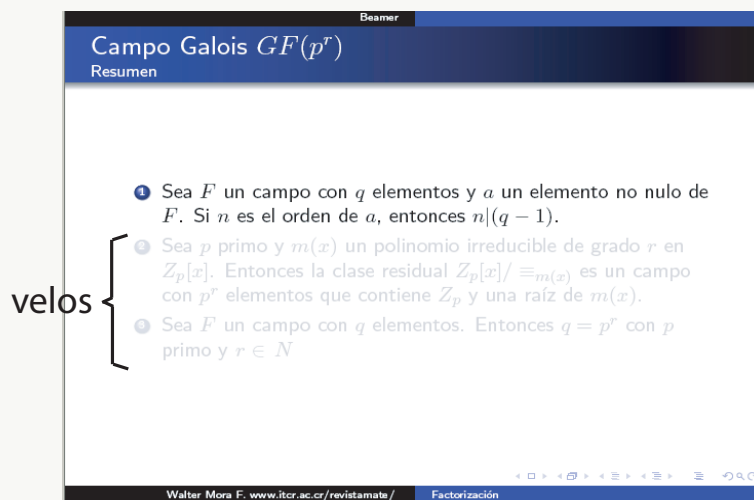


Figura 12.4. Marco Beamer con dos “velos”.

## Opciones &lt;i-&gt; y \uncover&lt;i-&gt;

En vez de usar la opción [ $<+>$ ] en el entorno **enumerate** (o **itemize**), se puede agregar un comportamiento un poco más dinámico usando las opciones <i-> y \uncover<i->{texto}.

Con estas opciones podemos controlar la secuencia en la que se presentan distintas líneas en una transparencia.

- \item<i-> indica que este ítem se presenta en la  $i$ -ésima transparencia. En la práctica se puede ver como la misma transparencia con un velo menos.
- \uncover<i->{texto} indica que este **texto** se presenta en la  $i$ -ésima transparencia

Por ejemplo, en la siguiente transparencia, se quiere mostrar inicialmente una ecuación y la solución, ambos ítems se marcan con <1->. Después se muestra el procedimiento, que corresponde a los ítems <2->, <3->, <4->

## Ejemplo 12.3

```
\begin{frame}{Ejemplo}
 \begin{enumerate}
 \item<1-> $x^4 - x = 0$ % <1->
 \item<2-> $x(x^3 - 1) = 0$
 \item<3-> $x = 0 \vee x^3 - 1 = 0$
 \item<4-> $x = 0 \vee x = \sqrt[3]{1}$
 \item<1-> $\Rightarrow x = 0, x = 1$ % <1->
 \end{enumerate}
\end{frame}
```

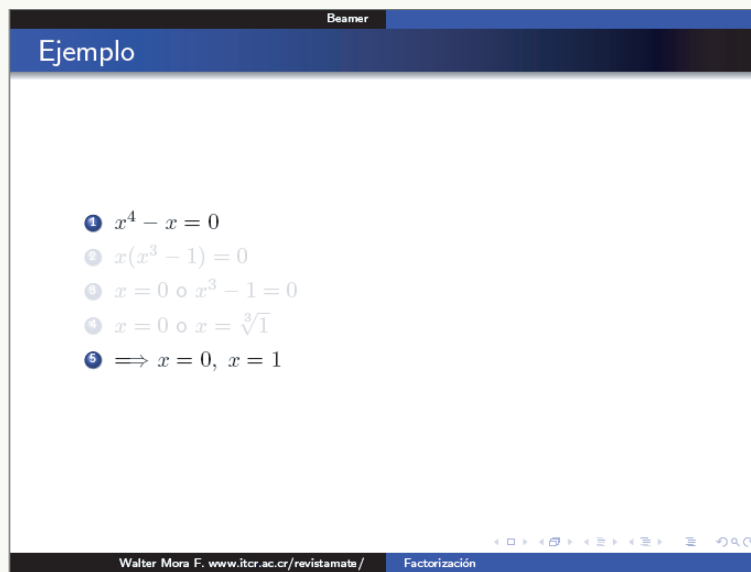


Figura 12.5. Marco Beamer con overlays.

- Un ejemplo del comando `\uncover` se presenta más adelante, en la sección 12.9.

### Opción <i-!alert@ i>

Esta opción se usa igual que en los casos anteriores. Lo nuevo es que la nueva transparencia descubre en color rojo el nuevo ítem.

## Ejemplo 12.4

```
\begin{frame}{Ejemplo}
 \begin{enumerate}
 \item<1-> $x^4 - x = 0$
 \item<2-> $x(x^3 - 1) = 0$
 \item<3-> $x = 0 \vee x^3 - 1 = 0$
 \item<4-> $x = 0 \vee x = \sqrt[3]{1}$
 \item<5-> $\Rightarrow x = 0, x = 1$
 \end{enumerate}
\end{frame}
```

```

\item<1-|alert@1> $x^4 - x = 0$ % <-
\item<2-|alert@2> $x(x^3 - 1) = 0$
\item<3-|alert@3> $x = 0$ \;$ o $ \;$ $x^3 - 1 = 0$
\item<4-|alert@4> $x = 0$ \;$ o $ \;$ $x = \sqrt[3]{1}$
\item<1-|alert@1> $\Rightarrow x = 0, \; x = 1$ % <-
\end{enumerate}
\end{frame}

```

**Nota:** Si solo queremos el efecto de ‘alerta’ en cada ítem, podemos poner

```

\begin{frame}{Ejemplo}
\begin{enumerate}[<+-| alert@+>] % <- opción
\item $x^4 - x = 0$
\item $x(x^3 - 1) = 0$
\item $x = 0$ \;$ o $ \;$ $x^3 - 1 = 0$
\item $x = 0$ \;$ o $ \;$ $x = \sqrt[3]{1}$
\item $\Rightarrow x = 0, \; x = 1$
\end{enumerate}
\end{frame}

```

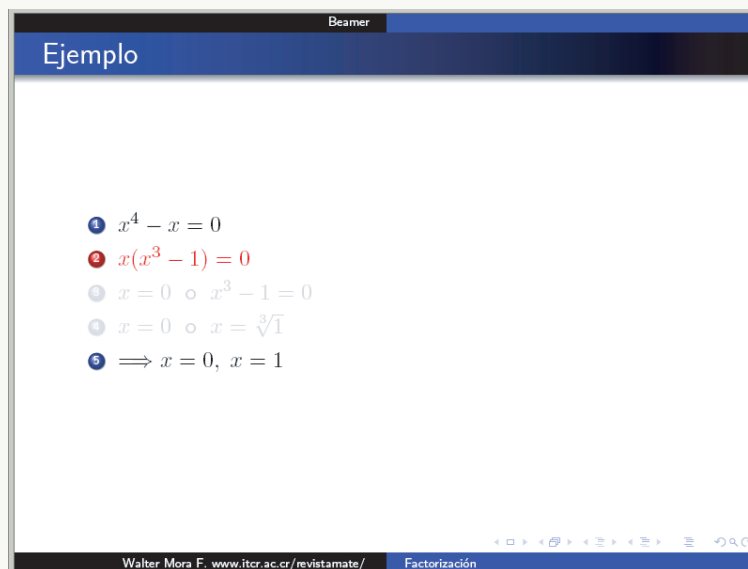


Figura 12.6. Marco **Beamer** con opciones `<i-|alert@ i>`

## 12.5 Comando pause

Para crear un velo, se puede usar `\pause`. Si solo se usa una vez, se cubre la parte del marco que está después de `\pause` y se puede usar varias veces en el marco si queremos fragmentar los velos (ver el ejemplo 12.5)

## 12.6 Entornos para teoremas, definición, etc.

Ya habíamos puesto en el preámbulo nuestras definiciones para los entornos de Teorema, Definición, etc. Estos entornos se usan igual que **enumerate**

239

### Ejemplo 12.5

```
\begin{frame}{Campo Galois $GF(p^r)$ }
\begin{teo} %definido en el preámbulo
 Sea F un campo y $P(x)$ mónico en $F[x]$, grado $P(x) \geq 1$.
 ...
\end{teo}
\pause % <---- PAUSA
\begin{ejemplo} % Entorno definido en el preámbulo
 Sea $P(x) = x^3 - 2 \in \mathbb{Q}[x]$. $P(x)$ es irreducible. Aunque tiene una
 raíz en $\mathbb{R}$, a saber $2^{1/3}$, $\mathbb{R}$ no es un campo de escisión para P .
\end{ejemplo}
\end{frame}
```

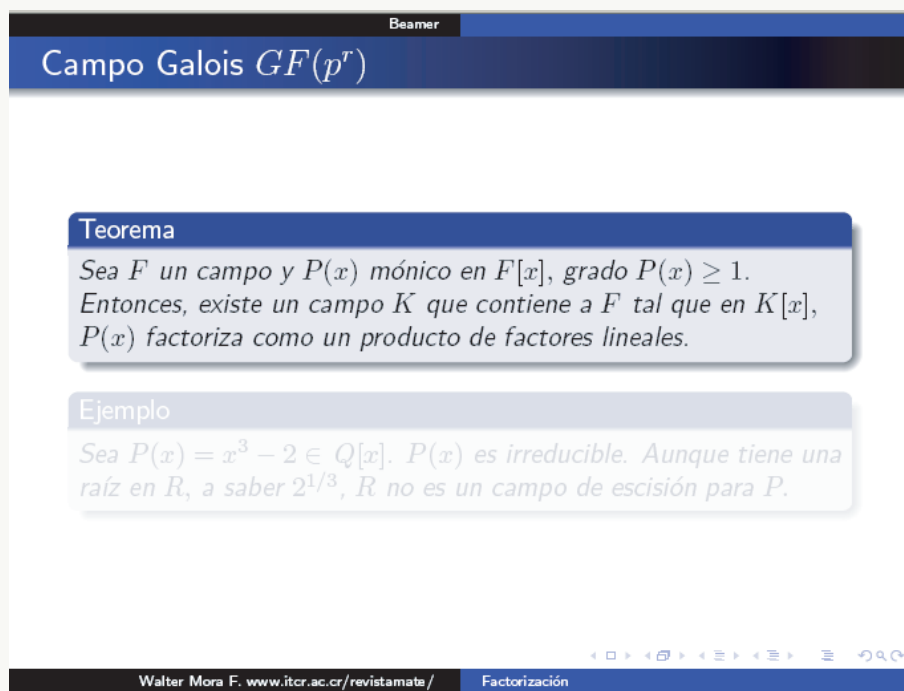


Figura 12.7. Entornos Teorema y Ejemplo.

**Nota:** Beamer tiene su propio entorno para ejemplos, teoremas, definiciones, etc.

El idioma se puede cambiar en el archivo (el paquete **babel** podría presentar choques con Beamer) `... \tex\latex\beamer\base\beamerbasetheorems.sty`

## 12.7 Blocks.

La numeración y la etiqueta de los entornos **teorema**, **definición**, etc., son adecuados para notas de clase, libros o artículos, es decir, en contextos donde el público puede acceder al material escrito porque la numeración de teoremas es algo complicado de recordar (recuerden el sufrido libro de Jean Dieudonne, “Foundations of Modern Analysis”). En una presentación es mejor indicar las cosas con nombre fáciles de recordar, como “Teorema de Rolle”, etc. El entorno **Block** es solo un caja sin etiqueta de entorno y sin numeración; la ventaja es que nos permite poner nombres arbitrarios a las cajas. Veamos un ejemplo,

### Ejemplo 12.6

```
\begin{frame}{Nodos igualmente espaciados}
\begin{block}{Diferencias hacia adelante}
\begin{eqnarray*}
\Delta^0 y_k &:=& y_k, \\
&\dots & \\
\end{eqnarray*}
\end{block}
\end{frame}
```

The figure shows a Beamer presentation slide. The slide has a title bar 'Nodos igualmente espaciados'. Inside, there is a block titled 'Diferencias hacia adelante' containing the following mathematical formulas:

$$\begin{aligned} \Delta^0 y_k &:= y_k, \\ \Delta^1 y_k &= y_{k+1} - y_k, \\ \Delta^2 y_k &= y_{k+2} - y_{k+1} - y_{k+1} + y_k = y_{k+2} - 2y_{k+1} + y_k, \\ &\dots \\ \Delta^n y_k &= \sum_{j=0}^n (-1)^j \binom{n}{j} y_{k+n-j} \end{aligned}$$

At the bottom of the slide, there is a navigation bar with standard Beamer navigation icons.

Figura 12.8. Block.

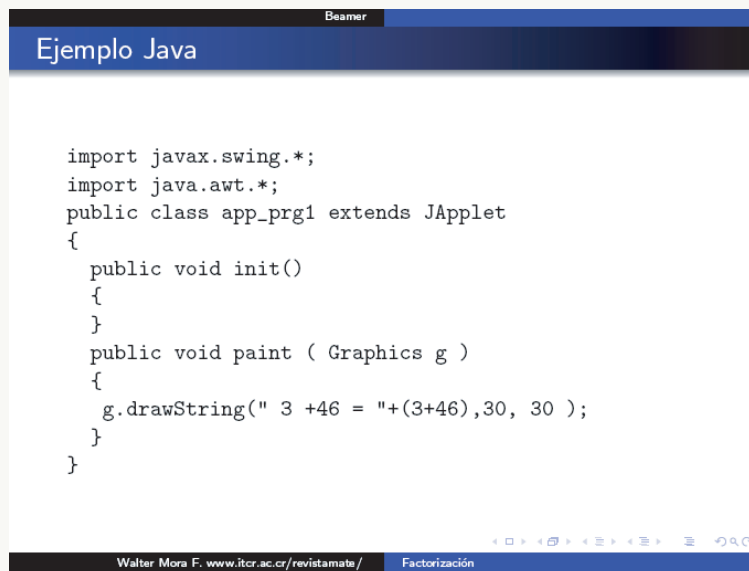
## 12.8 Opción fragile

Se debe usar la opción **fragile** en un marco que contiene **verbatim**, **algorithm2e**, etc. Un ejemplo típico, es presentar el código de un programa (ver figura 12.9),

241

Ejemplo 12.7

```
\begin{frame}[fragile]
\frametitle{Ejemplo Java}
\begin{verbatim}
import javax.swing.*;
import java.awt.*;
public class app_prg1 extends JApplet
{public void init(){
public void paint (Graphics g)
{g.drawString(" 3 +46 = "+(3+46),30, 30);}
}
\end{verbatim}
\end{frame}
```

Figura 12.9. Opción **fragile**.

## 12.9 Entornos para código de programas

### 12.9.1 Entorno semiverbatim

A veces es adecuado mostrar el código de un programa en bloques de una manera no necesariamente lineal. Para esto usamos recubrimientos y un ambiente **semiverbatim**.

- `\alert<i>{texto}` para poner el **texto** en rojo.



- `\uncover<i->{...}` para controlar la secuencia de recubrimientos dentro del entorno `semiverbatim`
- `\visible<i->{texto}` para mostrar **texto** en la transparencia *i* pero fuera del entorno `semiverbatim`. Se podrían usar los siguientes comandos,

## Ejemplo 12.8

```

\begin{frame}[fragile] % "fragile" es obligatorio
\frametitle{Un algoritmo para buscar números primos}
\begin{semiverbatim}
 \uncover<1->{\alert<0>{int main (void)}}
 \uncover<1->{\alert<0>{\{\}}}
 \uncover<1->{\alert<1>{ \alert<4>{std::}vector is_prime(100,true)}}
 \uncover<1->{\alert<1>{ for (int i = 2; i < 100; i++)}}
 \uncover<2->{\alert<2>{ if (is_prime[i])}}
 \uncover<2->{\alert<0>{ \{\}}}
 \uncover<3->{\alert<3>{ \alert<4>{std::}cout << i << " ";}}
 \uncover<3->{\alert<3>{ for (int j = 1; j < 100;}}
 \uncover<3->{\alert<3>{ is_prime [j] = false, j+=i);}}
 \uncover<2->{\alert<0>{ \{\}}}
 \uncover<1->{\alert<0>{ return 0;}}
 \uncover<1->{\alert<0>{\{\}}}
\end{semiverbatim}
\visible<4->{Notar el uso de \alert{\texttt{std::}}.}
\end{frame}

```

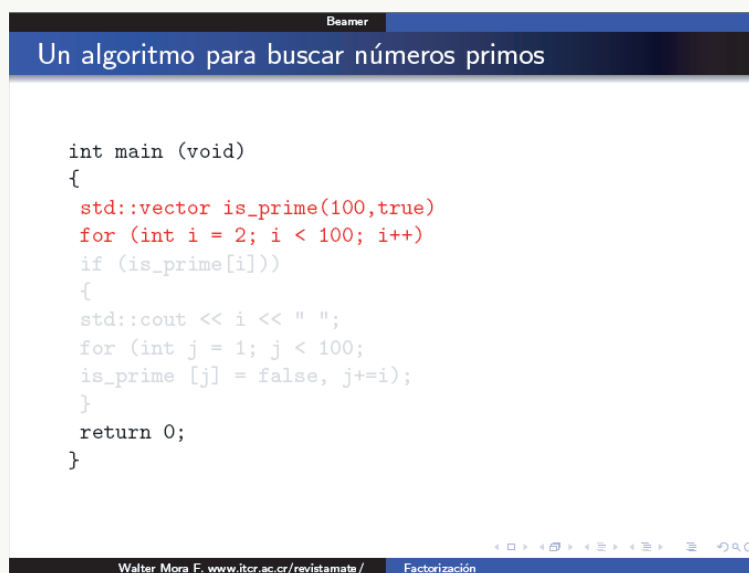


Figura 12.10. semiverbatim

## 12.9.2 Iluminar código de lenguajes de programación.

Para agregar color de manera automática al código de un lenguaje de programación se puede usar los paquetes **minted**, **verbments** o **listings**. Los dos primeros, posiblemente los más eficientes, necesitan instalar programas externos. **listings** se puede usar directamente.

Los detalles de la instalación de **minted** y **verbments** (Windows-Ubuntu) los puede ver en la sección respectiva.

Los tres entornos se pueden usar con **Beamer**. Por ejemplo, para usar **minted** ponemos en el preámbulo `\usepackage{minted}`. En Windows podría usar `\usepackage{lstlisting}`.

### Ejemplo 12.9

```
\begin{frame}[fragile]{Programación java} % fragile

\begin{minted}[frame=lines, rulecolor=\color{gray}]{java}

import javax.swing.*;
import java.awt.*;
public class app_prg1 extends JApplet
{ public void init(){
 public void paint (Graphics g){
 g.drawString(" 3 +46 = "+(3+46),30, 30);}
}

\end{minted}

\end{frame}
```

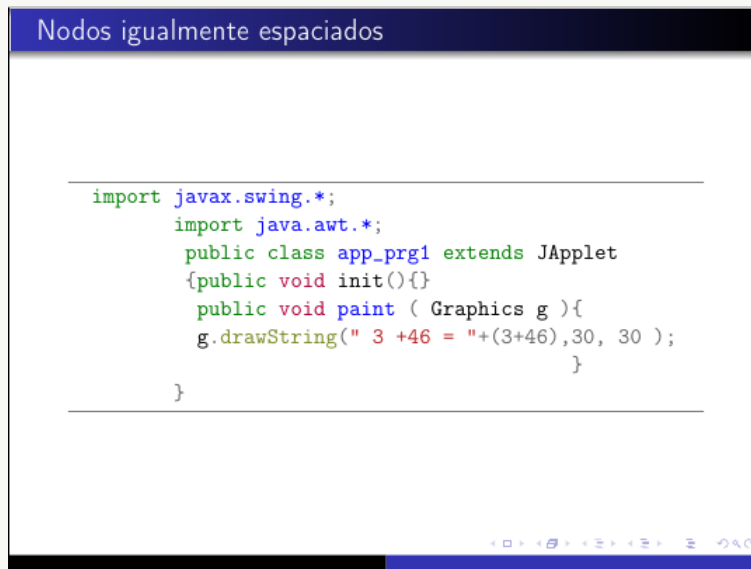


Figura 12.11. Paquete **minted** en **Beamer**

## 12.10 Beamer y el paquete algorithm2e

En esta sección vamos a ver un ejemplo en el que se usa el paquete **algorithm2e** (ver sección 11.10.1) en un entorno **frame**. Para este ejemplo, se puso en el preámbulo

%En el preámbulo

```
\usepackage[ruled,vlined,lined,linesnumbered,algosection,spanish]{algorithm2e}
```

Observe que necesitamos la opción **fragile** para **frame** y la opción **[H]** para **algorithm**.

### Ejemplo 12.10

```
\begin{frame}[fragile]
\begin{algorithm}[H] % <- necesario
\SetAlgoLined
\KwData{$(x_0, y_0), \dots, (x_m, y_m)$}
\KwResult{Coeficientes $\alpha_0, \alpha_1, \dots, \alpha_m$
en la base $\{X^{(0)}, \dots, X^{(n)}\}$. }
\LinesNumbered
$a_0 = y_0$;
$s = \alpha_j - \alpha_0$;
$f = x_j - x_0$;
\SetAlgoVlined
...
\Return $\alpha_j = s/f$;
\end{algorithm}
\end{frame}
```

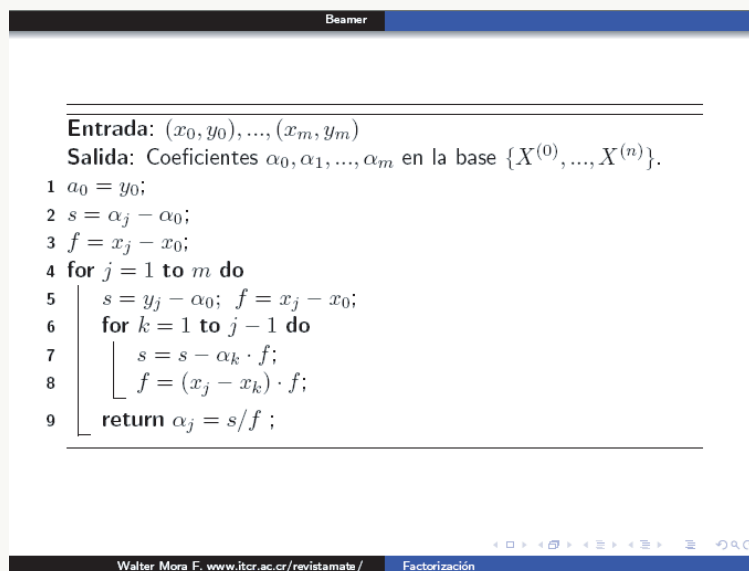


Figura 12.12. Beamer con algorithm2e

## 12.11 Gráficos

La inclusión de gráficos se hace de la misma manera que un documento **LaTeX** usual (ver capítulo 7). Como dijimos antes,

- ❶ Si instaló la distribución **TeXLive 2009** o menos o **MikTeX2.9** o menos y si compila con **PDFLaTeX**, no habrá problemas con las figuras en formatos **.pdf**, **.jpg**, **.png**. Si necesita usar figuras **.eps** podría agregar el paquete **epstopdf** y habilitar **shell escape**. Ver ??.
- ❷ Si instaló la distribución **TeXLive 2010** o superior, y si compila con **PDFLaTeX**, no habrá problemas con las figuras en formatos **.pdf**, **.jpg**, **.png**, **.eps**. En ambos casos podría, si le parece conveniente, declarar las extensiones para incluir figuras sin hacer referencia a la extensión.

```
\DeclareGraphicsExtensions{.pdf,.png,.jpg}
```

- ❸ Si compila con **LaTeX** puede usar figuras **.eps** (hay paquetes que permiten habilitar otros formatos como **.svg**, etc.)

### Ejemplo 12.11

```
% Requiere paquetes graphicx y caption
\begin{frame}{Superficie SS y proyección}
 \centering
 \includegraphics{images/B_fig12.pdf} %compilando PDFLaTeX
 \captionof{figure}{Superficie SS }\label{Beamer:fig12}
\end{frame}
```

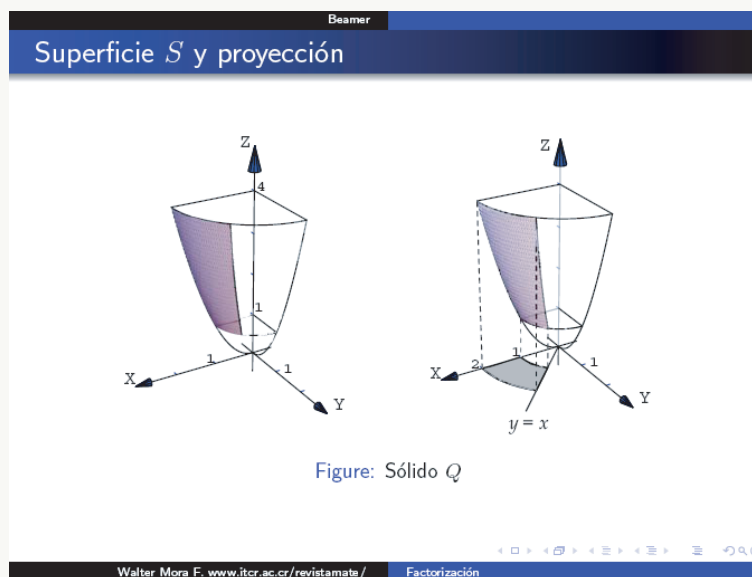


Figura 12.13. Incluir un gráfico

## 12.12 Ligas y botones.

Digamos que queremos poner un botón para ir a la transparencia  $j$  desde la transparencia  $i$  y, además, poner un botón en la transparencia  $j$  de retorno.

Para esto usamos ligas simples o botones: `\beamergotobutton` y `\beamerreturnbutton`.

Cada marco debe tener una identificación de marco y un marco destino.

```
\hyperlink{identificación del marco}{botón + texto}
```

```
\hypertarget<2>{identificación del destino}{}
```

### Ejemplo 12.12

```
\begin{frame}{MARCO 1}
 \hyperlink{MARCO1}{\beamergotobutton{Ir a Marco 2}}
 \hypertarget<2>{MARCO2}{}
\end{frame}
\begin{frame}{MARCO 2}
 \hyperlink{MARCO2}{\beamerreturnbutton{Regresar a Marco 1}}
 \hypertarget<2>{MARCO1}{}
\end{frame}
```

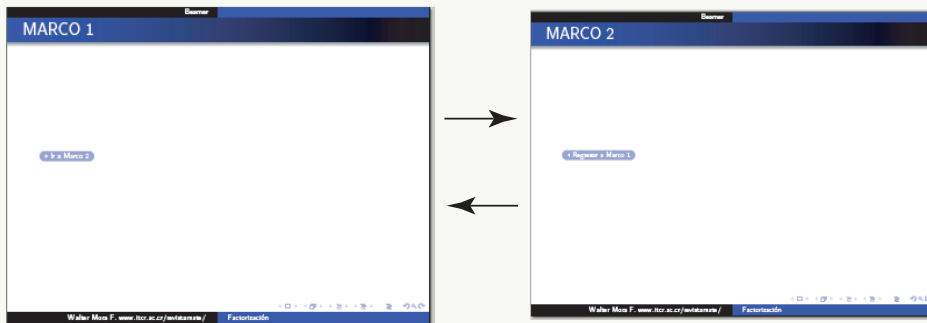


Figura 12.14. Ligas

- También se puede incorporar ligas *sin* botones,

```
\begin{frame}{MARCO 3}
 \hyperlink{MARCO3}{Ir a Marco 4}
 \hypertarget<2>{MARCO4}{}
\end{frame}
```

```
\begin{frame}{MARCO 4}
 \hyperlink{MARCO4}{Regresar a Marco 3}
 \hypertarget<2>{MARCO3}{}
\end{frame}
```

## 12.13 Efectos de Transición. Color

Un efecto de transición de una transparencia A a una transparencia B, se puede lograr poniendo el comando respectivo en cualquier parte de la transparencia B. El efecto se logra ver solo a pantalla completa.

### Otros efectos

- `\transblindshorizontal`
- `\transblindsvertical<2,3>`
- `\transboxin`
- `\transboxout`
- `\transglitter<2-3>[direction=90]`
- Se pueden incluir películas, animaciones, etc. con `\usepackage{multimedia}`, etc.

En el ejemplo que sigue, además de poner un efecto de transición vamos a crear un entorno **tabular** con las filas con color azul, específicamente **ZurichBlue**. Necesitamos hacer dos cosas en el preámbulo para que todo esto funcione,

- `\documentclass[xcolor=pdfTeX,table]{beamer}`. La opción “**table**” le informa a **Beamer** que el paquete **colortbl** debe ser cargado para poder usar la opción `\rowcolors`
- `\definecolor{ZurichBlue}{rgb}{.255,.41,.884}`. Con esto definimos lo que será nuestro **ZurichBlue**. En el código que sigue, el color se pone en distintos porcentajes.

#### Ejemplo 12.13

```
%\documentclass[xcolor=pdfTeX,table]{beamer}
%\definecolor{ZurichBlue}{rgb}{.255,.41,.884}
\begin{frame}
 \transdissolve % <--- Efecto de transición
 \begin{center}
 \rowcolors{1}{ZurichBlue!20}{ZurichBlue!5} %Porcentaje de color
 \begin{tabular}{|l|l|c|}\hline
 Enteros & \& de -2^{63} a $2^{63}-1$ \\ \hline
 ...
 \end{tabular}
 \end{center}
\end{frame}
```

| Marco B       |              |                                                   |
|---------------|--------------|---------------------------------------------------|
| Enteros       | long         | de $-2^{63}$ a $2^{63} - 1$                       |
|               | int          | de $-2^{31}$ a $2^{31} - 1$                       |
|               | short        | de $-2^{16}$ a $2^{16} - 1$                       |
|               | byte         | de $-2^7$ a $2^7 - 1$                             |
| Coma flotante | float        | de $3.4 \times 10^{-38}$ a $3.4 \times 10^{38}$   |
|               | double       | de $1.7 \times 10^{-308}$ a $1.7 \times 10^{308}$ |
| Caracteres    | char         |                                                   |
| boolean       | true o false |                                                   |

Figura 12.15. Transición y Color

## 12.14 Ligas a Documentos Externos

Para hacer ligas a documentos externos podemos usar el comando

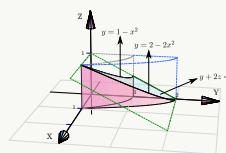
```
\href{http://...}{ texto}
```

### Ejemplo 12.14

El código:

```
\parbox{3cm}{
\href{http://www.tec-digital.itcr.ac.cr/revistamatematica/cursos-linea/
3D-Web/exersolido21.html}{
\includegraphics[width=3cm]{images/exersolido21}}
}\parbox{12cm}{Sólido Q_{14} limitado por las
superficies $y=2-2x^2$; $y=1-x^2$; $y+2z=2$; $x=0$ y $z=0$; en el
I octante.}\}
```

produce: (puede hacer clic sobre el gráfico)



Sólido  $Q_{14}$  limitado por las superficies  $y = 2 - 2x^2$   
 $y = 1 - x^2$ ;  $y + 2z = 2$ ;  $x = 0$  y  $z = 0$ ; en el I octante.

- Liga a un documento en el disco duro, por ejemplo

El código del programa está [aquí](#)

```
El código del programa está
\href{file:///C:/MiJava/programa1.java}{\underline{aquí}}
```

- Otras ligas:

Prof. Walter Mora F. ([wmora2@gmail.com](mailto:wmora2@gmail.com))

En la revista digital Matemática, Educación e Internet encontrará [el Manual de LaTeX](#), en la liga “Libros”

```
Prof. Walter Mora F.
{\href{mailto:wmora2@gmail.com}{(wmora2@gmail.com)}}\
En la revista digital Matemática, Educación e Internet
encontrará \href{http://www.tec-digital.itcr.ac.cr/revistamatematica/}
{\underline{el Manual de LaTeX}}, en la liga ``Libros''\
```

## 12.15 Animaciones

Se puede preparar una animación simplemente desplegando una secuencia de gráficos, por ejemplo

---

```
\begin{frame}
\frametitle{Mi animación}
\begin{figure}[t]
\centering
\includegraphics<1>[scale=0.2]{images/picture_1.pdf}
\includegraphics<2>[scale=0.2]{images/picture_2.pdf}
\includegraphics<3>[scale=0.2]{images/picture_3.pdf}
\includegraphics<4->[scale=0.2]{images/picture_4.pdf}
\end{figure}
\end{frame}
```

---

## 12.16 Multicolumnas.

Las cajas que se realizan en una diapositiva con beamer (teoremas, definiciones, bloques) se pueden acomodar en varias columnas, esto usualmente no se utiliza mucho en presentaciones pero sí se utiliza bastante si lo que se quieren hacer son posters o panfletos (tal como se muestra en el capítulo 13).

Para realizar multicolumnas con beamer se utilizan los entornos **columns** y **column**, lo mejor es definir el tamaño de las columnas de acuerdo al ancho del documento, observe el siguiente ejemplo.



## Ejemplo 12.15

```

\begin{frame}{Multicolumnas}
\begin{columns}[t]
\begin{column}{.48\linewidth}
\begin{block}{Bloque 1}
Bloque de la primera columna
\end{block}
\end{column}
\begin{column}{.48\linewidth}
\begin{block}{Bloque 2}
Bloque en la segunda columna
\end{block}
\end{column}
\begin{column}{.48\linewidth}
\begin{block}{Bloque 3}
Bloque en la segunda columna
\end{block}
\end{column}
\end{columns}
\end{frame}

```

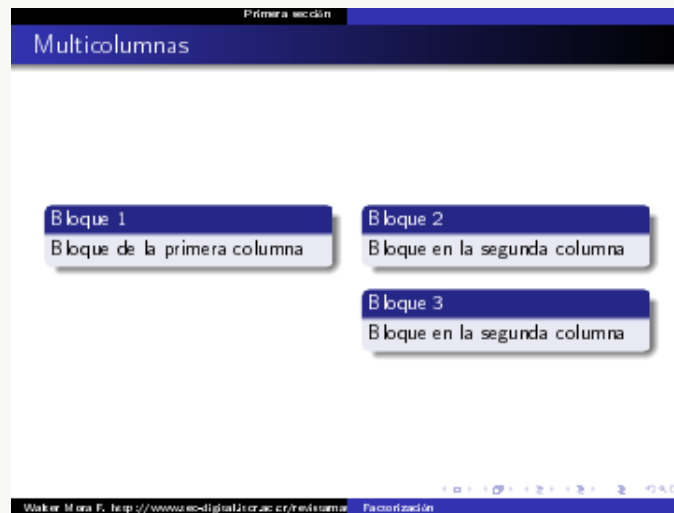


Figura 12.16. Multicolumnas.

En este caso se hicieron dos columnas dentro del ambiente `columns`, el ancho de cada una de ellas se definió como `.48\linewidth` que es un poquito menos que el ancho de la página, esto se hace para dejar un espacio pequeño entre las columnas y la diapositiva. Si se hacen tres columnas y se quiere que todas tengan el mismo ancho se puede utilizar `.3\linewidth` para cada una de ellas.

## 12.17 Color y otros ajustes

Aunque los temas ya tienen un buen diseño de color, podría ser que uno quiera personalizar algunas cosas sin tocar el código fuente del tema. En el siguiente código se muestra como hacer cambios en el color y el idioma. Los colores que se muestran son solo para efectos de ilustrar.

```
\documentclass[12pt]{beamer}
\usepackage[latin1]{inputenc}
\usetheme{Warsaw}

%-- Cambios en el idioma -----
\uselanguage{spanish}
\languagepath{spanish}
\deftranslation[to=spanish]{Theorem}{Teorema}
\deftranslation[to=spanish]{theorem}{Teorema}
\deftranslation[to=spanish]{Example}{Ejemplo}
\deftranslation[to=spanish]{example}{Ejemplo}
%...
%-----

%-- Cambios en el color -----
% Define el color:
\definecolor{miazul}{rgb}{0.137,0.466,0.741}
% Lo aplica al fg="foreground"
\setbeamercolor{structure}{fg=miazul}
% Define el color:
\definecolor{mivioleta}{RGB}{216,203,230}
% Lo aplica al bg="background"
\setbeamercolor{background canvas}{bg=mivioleta}

% Otras opciones....
% Color de fuente
\setbeamercolor{normal text}{fg=white}
% Background de los entornos teorema, defi, etc.
\setbeamercolor{block body}{fg=black,bg=blue!40}
%-----

\begin{document}
%...
\begin{frame}{Series alternadas}
% ...
\end{frame}
%...
\end{document}
```



## 13.1 Introducción

Con  $\text{\LaTeX}$  es posible realizar posters y trípticos (panfletos o brochures).

Para realizar los posters se puede utilizar la clase “**Beamer**” esta es una clase de  $\text{\LaTeX}$  que se usa para generar transparencias para presentaciones al estilo Power Point (ver el capítulo 12). Para hacer los posters se utiliza el paquete **beamerposter**, un ejemplo se puede ver en la figura 13.1.

**7º CIEMAC Congreso Internacional sobre la Enseñanza de la Matemática Asistida por Computadora**

**Instituto Tecnológico de Costa Rica**  
Diciembre, 2011

El Instituto Tecnológico de Costa Rica en respuesta a su compromiso con el desarrollo del país y con su vocación constante de apoyo integral con la educación costarricense tiene el agrado de invitar a profesores y profesores de matemática de educación media y a maestros y maestras de primaria al 7º Congreso Internacional sobre Enseñanza de la Matemática Asistida por Computadora. Este congreso está orientado a ofrecer a los educadores y educadoras del país tres días de reflexión y capacitación sobre temas relacionados con la matemática y su enseñanza. La actividad es organizada por la Escuela de Matemática del Instituto Tecnológico de Costa Rica.

**Temas de trabajo:**  
El trabajo académico se centrará en los siguientes ejes temáticos:  
• Área de Software para Educación.  
• Área de Desarrollo del Razonamiento.  
• Área de evaluación de software.  
Se desarrollará fundamentalmente a través de conferencias y talleres de trabajo.

**Participación:**  
El congreso contará con diversas exposiciones tanto nacionales como extranjeras, entre las más destacadas están:  
• Ricardo Hernández (Estados Unidos)  
• Edson Delfino (Costa Rica)  
• Norma Nájera (Estados Unidos)  
• José Luis Lapáez (España)  
• Mario Martín (Costa Rica)  
• Félix Sepúlveda (México)  
• Víctor Palencia (México)  
• Norma Goria (México)  
• Teresa Carrillo (México)

**Modalidades de participación:**  
Las modalidades para los trabajos son:  
• Talleres.  
• Cursos cortos.  
• Conferencias.  
Le invitamos a enviar sus propuestas al correo [absc@itcr.ac.cr](mailto:absc@itcr.ac.cr), pueden ser talleres de 2, 4 o 6 horas ya sean sobre software o un taller de evaluación y diseño de actividades curriculares, ponencias que demuestren resultados de investigaciones o reportes de experiencias exitosas en el uso de tecnología en los procesos de enseñanza aprendizaje de la matemática, estas tienen una duración 30 minutos o bien de una hora si el ponente lo justifica. Además se desarrollarán otras actividades tendientes a fomentar en los docentes el interés por utilizar software educativo en matemática y en general para ayudarlos a crear profesionalmente mediante la actualización en temas de relevancia como resolución de problemas y uso de software en educación. La fecha límite para el envío de propuestas es el 15 de septiembre del 2011.

**Comité organizador:**  
M.Sc. Mario Martín Sánchez, ITCR (coordinador)  
M.Sc. Corbin Páez Páez, ITCR  
M.Sc. Alexander Borbón Alajó, ITCR  
M.Sc. Jorge Marge Falas, ITCR  
M.Sc. Cindy Calvo, ITCR  
Lic. Rebecca Silva Ortega

| Hora                | Miércoles 16                         | Jueves 17                  | Viernes 18                       |
|---------------------|--------------------------------------|----------------------------|----------------------------------|
| 8:00 am a 9:00 am   | Inscripción, inauguración y desayuno | Conferencia plenaria       | Conferencia plenaria             |
| 9:00 am a 10:00 am  | Conferencia inaugural                | Conferencia plenaria       | Conferencia plenaria             |
| 10:00 am a 10:30 am | Refrigerio                           |                            |                                  |
| 10:30 am a 12:30 pm | Talleres y ponencias                 | Talleres y ponencias       | Cierre y entrega de certificados |
| 12:30 pm a 1:30 pm  | Almuerzo                             |                            |                                  |
| 1:30 pm a 3:00 pm   | Talleres y ponencias                 | Talleres y ponencias       |                                  |
| 3:00 pm a 3:30 pm   | Refrigerio                           |                            |                                  |
| 3:30 pm a 5:00 pm   | Talleres y ponencias                 | Talleres y ponencias       |                                  |
| 5:00 pm a 6:00 pm   | Actividades de integración           | Actividades de integración |                                  |

| Tipo de participación   | Inscripción temprana | Inscripción general |
|-------------------------|----------------------|---------------------|
| Ponente nacional        | Q10 000              | Q25 000             |
| Ponente extranjero      | \$50                 | \$60                |
| Estudiante nacional     | Q12 000              | Q17 000             |
| Estudiante extranjero   | \$60                 | \$70                |
| Participante nacional   | Q20 000              | Q25 000             |
| Participante extranjero | \$75                 | \$100               |

**Observaciones:**  
• Los precios de ponente son para aquellos expositores que desean participar en todo el evento y recibir título de participación.  
• La cuota de inscripción incluye: 5 refrigerios (mañana y tarde), todo del evento, CD de memoria del evento, título de participación, gafete, programa de actividades, calendario y derecho a asistencia a las actividades incluidas en el programa.

**Información adicional:**  
Si desea recibir información adicional sobre el congreso o tiene alguna duda del evento, por favor escribir a [ciemac@itcr.ac.cr](mailto:ciemac@itcr.ac.cr) o a Mario Martín a [marin@itcr.ac.cr](mailto:marin@itcr.ac.cr)  
Escuela de Matemática  
Instituto Tecnológico de Costa Rica  
Cartago, Costa Rica  
Teléfono: (506) 2552-5333 / (506) 2550-2225, Fax: (506) 2550-2493  
Apartado Postal: 159-7050 Cartago, Costa Rica

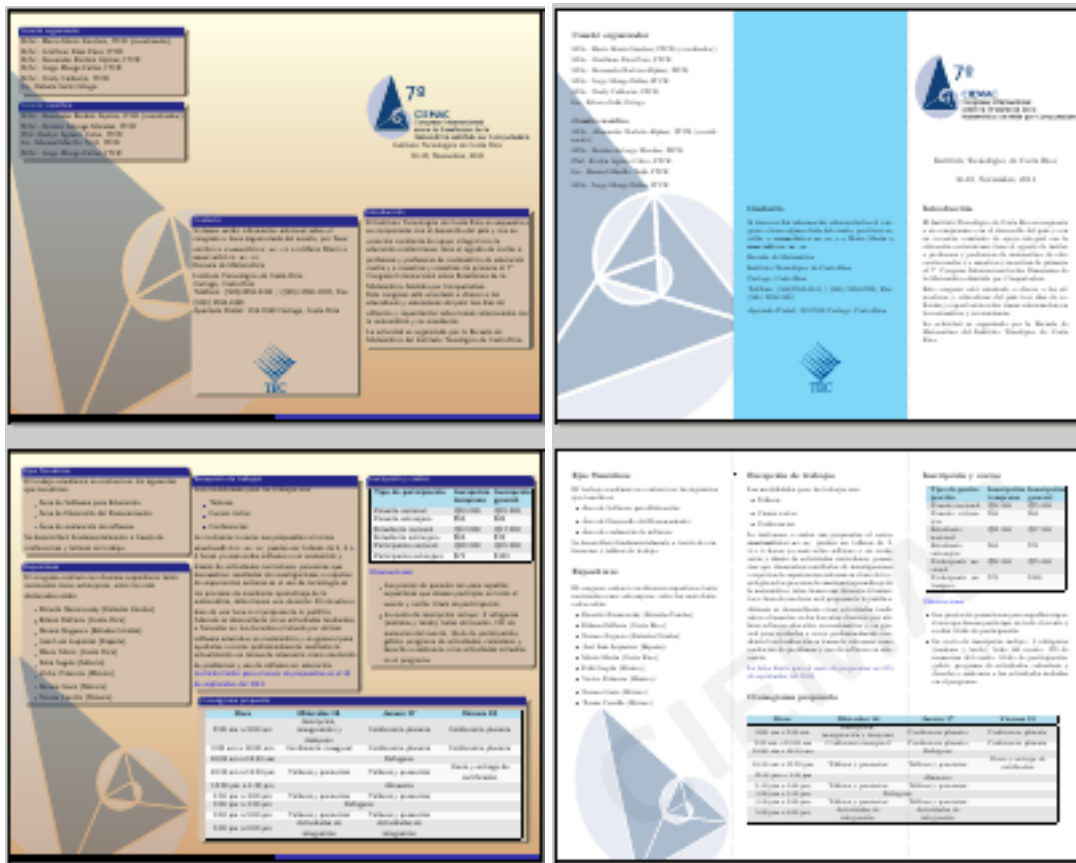
Figura 13.1. Ejemplo de poster con beamer.

Para los panfletos (brochures) también se puede utilizar la clase **beamer** o la clase **leaflet** que es especializada para ello, en el libro se va a explicar la creación de panfletos con ambas clases.

Un ejemplo de brochure utilizando beamer se puede observar en la figura 13.2a, otro ejemplo utilizando leaflet se puede observar en la figura 13.2b.

## 13.2 Posters

Existen distintas clases para realizar posters en  $\text{\LaTeX}$ , entre las más comunes están **a0poster**, **sciposter** y **beamer**.



(a) Panfleto con beamer

(b) Panfleto con leaflet

Figura 13.2. Panfletos (brochures o trípticos)

En el libro se mostrará la forma en que se puede realizar un poster utilizando la clase **beamer** con el paquete **beamerposter**. Se mostrará sólo la utilización de esta clase ya que se considera bastante sencilla de implementar y con un acabado final muy bueno.

La estructura general de un posible documento de  $\text{\LaTeX}$  para producir un poster con Beamer podría ser,



```
\documentclass{beamer}
\usepackage[orientation=portrait,size=a0,scale=1.4,debug]{beamerposter}

\mode<presentation> {
 \usetheme{Warsaw}
}

\usefonttheme[onlymath]{serif}
\boldmath

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

```

\usepackage{amsmath,amsthm, amssymb, latexsym}

\title[Posters]{\VERYHuge Haciendo un poster en \LaTeX}
\author[Alex]{\huge Alexander Borb\on Alp\lzar}
\institute[ITCR]{\Large Instituto Tecnol\ogico de Costa Rica}
\date[Enero-Febrero, 2013]

\begin{document}
\begin{frame}[plain]{}
 \maketitle
 \vfill
 \begin{block}{\large Fontsizes}
 \bc
 {\tiny tiny}\par
 {\scriptsize scriptsize}\par
 {\footnotesize footnotesize}\par
 {\normalsize normalsize}\par
 {\large large}\par
 {\Large Large}\par
 {\LARGE LARGE}\par
 {\veryHuge veryHuge}\par
 {\VeryHuge VeryHuge}\par
 {\VERYHuge VERYHuge}\par
 \end{block}
 \vfill
 \begin{columns}[t]
 \begin{column}{.48\the}
 \begin{block}{Introducci\on}
 En este art\lculo...
 \end{block}
 \end{column}
 \begin{column}{.48\the}
 \begin{block}{Secci\on 2}
 \begin{itemize}
 \item item 1 y $\int f(x) dx$
 \item item 2
 \end{itemize}
 \end{block}
 \begin{block}{Secci\on 3}
 \begin{itemize}
 \item item 1
 \item item 2
 \end{itemize}
 $\int f(x) dx$
 \end{block}
 \end{column}
 \end{columns}
\end{frame}
\end{document}

```

Lo cual, al compilarlo con **PDFLaTeX** produce un poster sencillo, dicho poster se muestra en la 13.3.



Figura 13.3. Poster sencillo. Tema Warsaw

- Como se puede notar el tipo de documento es **beamer**, pero en este caso se utiliza el paquete **beamerposter** para extenderlo de forma tal que permita trabajar con un tamaño de página grande. Dentro de las opciones del paquete se utilizaron:
  - **orientation**: Esta se refiere a la orientación de la hoja, puede recibir portrait o landscape ya sea si la página se quiere vertical u horizontal.
  - **size**: Se refiere al tamaño de la hoja, para posters los más comunes son a0, a1 y a2, sin embargo también se puede utilizar a3 y a4. en la siguiente tabla se muestran las medidas para estos tamaños:

| Tamaño | Dimensiones      |
|--------|------------------|
| a0     | 84.1cm x 118.9cm |
| a1     | 59.4cm x 84.1cm  |
| a2     | 42cm x 59.4cm    |
| a3     | 29.7cm x 42cm    |
| a4     | 21cm x 29.7cm    |

- **escale**: Es la escala a la cual se presentarán los elementos en el poster, el tamaño normal es 1, entre mayor sea el número los elementos saldrán más grandes.

- Posteriormente se indica que se debe utilizar el tema Warsaw en modo presentación.

```
\mode<presentation> {
 \usetheme{Warsaw}
}
```

Tal como se vió en el capítulo 12 sobre beamer, con sólo cambiar el tema se puede obtener una nueva combinación de colores, esta es una gran ventaja ya que se pueden obtener posters muy distintos sólo cambiando el tema. En Internet se pueden conseguir temas exclusivos para posters. Recuerde además que los temas se pueden personalizar y se pueden crear temas propios.

- Las líneas

```
\usefonttheme[onlymath]{serif}
\boldmath
```

le dan formato al texto matemático poniéndole como fuente **serif** en negrita.

- Lo siguiente es definir el título, autor, institución y fecha para que beamer realice automáticamente el título del poster

```
\title[Posters]{\VERYHuge Haciendo un poster en \LaTeX}
\author[Alex]{\huge Alexander Borb\on Alp\lzar}
\institute[ITCR]{\Large Instituto Tecnol\ogico de Costa Rica}
\date{Enero-Febrero, 2013}
```

Los datos opcionales (que se dan entre corchetes) usualmente se utilizan para otros lugares en el poster, por lo general en la zona de abajo.

En algunos temas de **beamer**, sobre todo los que están pensados para **beamerposter** el título sale automáticamente en el **frame**, si esto no ocurre se debe poner el comando **maketitle**.

- En el cuerpo del documento lo que se hace es un sólo **frame** (una sola diapositiva), en esta se hará todo el poster.

```
\begin{frame}[plain]{}
...
\end{frame}
```

La opción **plain** se utiliza para que no salga la barra de arriba ni la de abajo que usualmente aparece en las diapositivas.

Para acomodar el poster se utilizan los **bloques** y las **multicolumnas** de beamer (tal como se vieron en las secciones 12.7 y 12.16). Las imágenes se agregan de la forma usual.

- El fondo del poster se puede cambiar con alguno de los comandos

```
\beamertemplateshadingbackground{color1}{color2}
\beamersetaveragebackground{color}
\beamertemplatesolidbackgroundcolor{color}
```



Donde `\beamertemplateshadingbackground` realiza un degradado desde el *color1* hasta el *color2*. Los otros dos comandos sirven para poner un color sólido de fondo.

Así en el siguiente ejemplo se agregó el comando `\beamertemplateshadingbackground{brown!70}{yellow!10}` en el preámbulo, además se agregó el logo del TEC con los comandos

```
\vspace*{-10cm}
\begin{center}
 \includegraphics[scale=2]{LogoTECMemorias}
\end{center}
```

esto justo después de hacer el título con `\maketitle`. El resultado se puede observar en la figura 13.4.



Figura 13.4. Fondo de color e imagen.

## 13.3 Trípticos (Panfleots o Brochures)

Los trípticos (panfletos o brochures) se pueden realizar utilizando la clase **beamer** o la clase **leaflet**, esta última es una clase especializada para este tipo de documentos.

### 13.3.1 Trípticos con beamer

Para realizar un **tríptico con beamer** se debe crear una presentación con dos diapositivas, cada una de ellas del tamaño del papel que se quiera y se divide cada página en tres columnas.

Se debe tomar en cuenta que el orden en que aparecerán las páginas en las hojas al doblar el panfleto es el que se muestra en la figura 13.5.

|   |   |   |
|---|---|---|
| 5 | 6 | 1 |
| 2 | 3 | 4 |

Figura 13.5. Orden de las páginas en un panfleto.

Al utilizar beamer este ordenamiento de las páginas no es automático sino que se debe realizar de forma manual.

El **código**: mínimo necesario para realizar un panfleto con beamer es:



Descargar

archivo

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

\usetheme{Warsaw}
\setbeamertemplate{navigation symbols}{}
\usepackage{geometry}
\geometry{landscape, letterpaper}

\usepackage[utf8]{inputenc}
\begin{document}
 \begin{frame}[fragile]{}
 \begin{columns}[t]
 \begin{column}{0.33\textwidth}
 Texto de la p\`agina 5
 \end{column}
 \begin{column}{0.33\textwidth}
 Texto de la p\`agina 6
 \end{column}
 \begin{column}{0.33\textwidth}
 \includegraphics[scale=0.5]{LogoCIEMAC}
 Texto de la p\`agina 1
 \end{column}
 \end{columns}
 \end{frame}
 \begin{frame}[fragile]{}
 \begin{columns}[t]
 \begin{column}{0.33\textwidth}
 Texto de la p\`agina 2
 \end{column}
 \begin{column}{0.33\textwidth}
 \begin{block}{Bloque 1}
```

```

 Texto del bloque
 \end{block}
\end{column}
\begin{column}{0.33\textwidth}
 Texto de la p\`agina 4
\end{column}
\end{columns}
\end{frame}
\end{document}

```

El resultado de este **código**: se puede observar en la imagen 13.6.

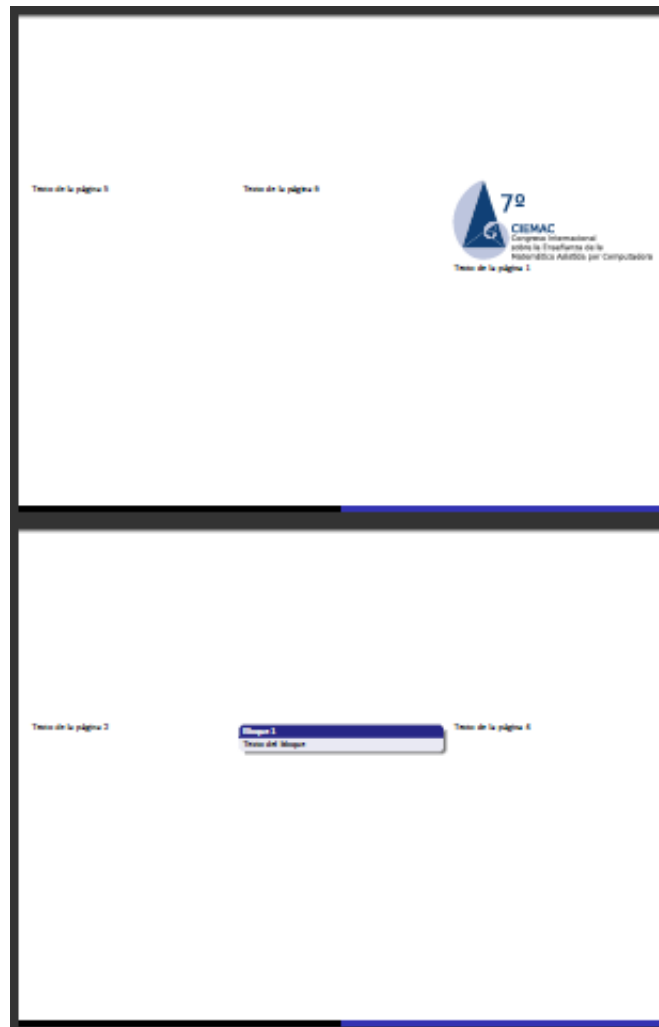


Figura 13.6. Panfleto con **código**: mínimo.

- Como se nota la clase que se utiliza es **beamer** con el tema **Warsaw**, de igual forma con sólo cambiar el tema ya se obtendría un estilo distinto para las cajas y los colores.

Para definir el tamaño y el formato de la página se puede utilizar el paquete **geometry**, en este caso se definió el tamaño carta con una orientación horizontal.

```

\usepackage{geometry}
\geometry{landscape, letterpaper}

```

- Para eliminar la barra de navegación que beamer coloca por defecto en la esquina inferior derecha se puede utilizar el **código**:

```
\setbeamertemplate{navigation symbols}{}
```

- Si se quisiera poner un fondo con un degradado bonito se puede hacer con el **código**:

```
\beamertemplateshadingbackground{brown!70}{yellow!10}
```

También es posible poner una imagen como fondo, para esto, si se quiere colocar la imagen Fondo.pdf se puede utilizar el **código**:

```
\usebackgroundtemplate{
 \includegraphics[width=\paperwidth, height=\paperheight,
 keepaspectratio]{Fondo}
}
```

- En cada columna se pueden colocar bloques, texto, imágenes, etc. En general cualquier elemento que se puede utilizar en una presentación con beamer. En el ejemplo se puso un bloque en la página 3 del brochure y una imagen en la primera página.
- Se debe recordar que al utilizar el paquete **inputenc** se debe verificar si el documento está en la codificación **utf8** ó **latin1** (ISO-8859-1).

### 13.3.2 Trípticos con leaflet

La clase leaflet es una clase especializada para realizar panfletos, en el caso de leaflet la segunda página aparece al revés para imprimir directamente en una impresora que imprima por ambos lados de la página. Esta distribución se puede observar en la figura 13.7.

|   |   |   |
|---|---|---|
| 5 | 6 | 1 |
| 4 | 3 | 2 |

Figura 13.7. Orden de las páginas en un panfleto con leaflet.

Sin embargo, si no se quiere que la segunda página aparezca de esta forma se debe colocar la opción **notumble**.

El **código**: mínimo para realizar un panfleto con leaflet es:

```
\documentclass[10pt, notumble, letterpaper]{leaflet}

\usepackage[utf8]{inputenc}
\pagestyle{empty}

\title{7° Congreso Internacional sobre Enseñanza de la
Matemática Asistida por Computadora}
```

```

\author{Instituto Tecnol\ogico de Costa Rica}
\date{16-18, Noviembre, 2011 }

\CutLine*{1}% linea punteada sin tijeras
\CutLine*{3}% linea punteada sin tijeras
\CutLine*{4}% linea punteada sin tijeras
\CutLine{6}% linea punteada con tijeras

\begin{document}

\maketitle

\thispagestyle{empty}

\section{Secci'on 1}

Texto en la secci'on 1

\newpage

\section{Secci'on 2}

Texto en la secci'on 2

\newpage

\section{Secci'on 3}

Texto en la secci'on 3

\newpage

\section{Secci'on 4}

Texto en la secci'on 4

\newpage

\section{Secci'on 5}

Texto en la secci'on 5

\newpage

\section{Secci'on 6}

Texto en la secci'on 6

\section{Secci'on 7}

```

Texto en la sección 7

`\end{document}`

El resultado al compilar este código: es el panfleto sencillo que se muestra en la figura 13.8.

263

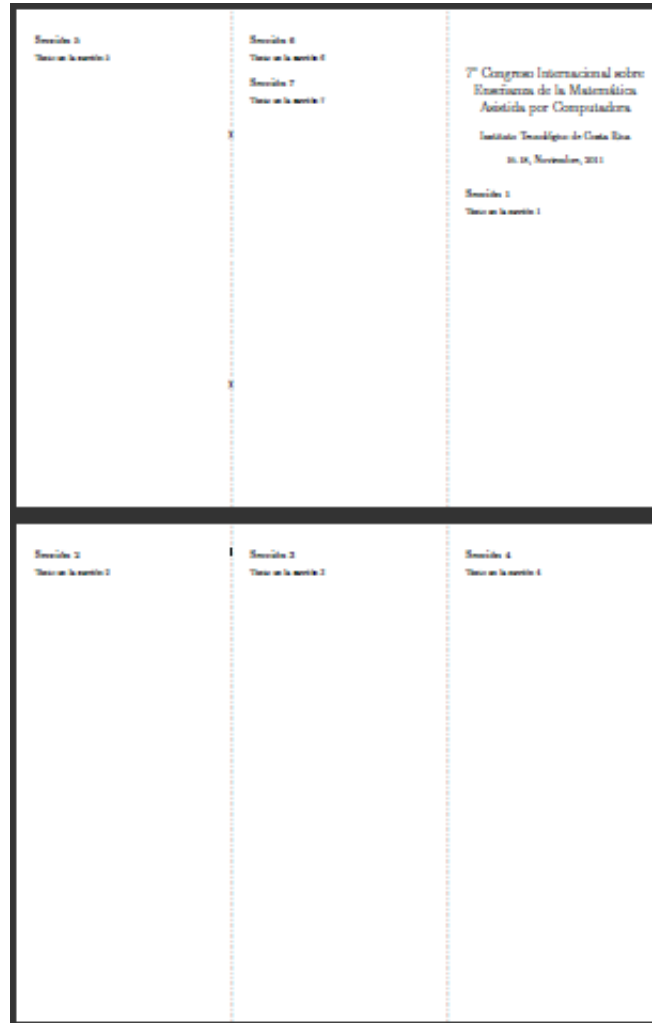


Figura 13.8. Panfleto mínimo con leaflet.

- Como se puede notar en el código: y el resultado, usar **leaflet** tiene la ventaja que las seis páginas del panfleto se acomodan automáticamente, en el código: simplemente se escribe de manera continua utilizando título, secciones, subsecciones, etc., en el momento que se acabe una página se continúa automáticamente en la página siguiente y si se quiere pasar de página de forma manual se puede utilizar el comando `\newpage`.
- La clase **leaflet** hace diferencia entre las dos páginas grandes del panfleto y las páginas pequeñas dentro de las grandes. En la figura 13.7 se notan que se tienen dos páginas grandes, en la primera página grande están las páginas pequeñas uno, cinco y seis, mientras que en la segunda página grande están las páginas pequeñas dos, tres y cuatro.

Así, por ejemplo el **código**:

`\CutLine*{1}% línea punteada sin tijeras`

```
\CutLine*{3}% línea punteada sin tijeras
\CutLine*{4}% línea punteada sin tijeras
\CutLine{6}% línea punteada con tijeras
```

Lo que hace es trazar las líneas punteadas que separan las páginas pequeñas, el comando `\CutLine*` hace una línea punteada sin tijeras y `\CutLine` hace una línea punteada con tijeras.

En este caso se hicieron líneas punteadas sin tijeras en las páginas pequeñas 1, 3 y 4, y una línea punteada con tijeras en la página pequeña 6.

- La clase **leaflet** tiene el comando `\AddToBackground` que permite agregar elementos tales como imágenes o marcas de agua en el fondo de las páginas grandes y las páginas pequeñas. Todos los elementos que se agreguen en el fondo con este comando se deben definir en el preámbulo del documento.

Para agregar un elemento en una página pequeña se utiliza `\AddToBackground{pagina}` y si es en una página grande se usa `\AddToBackground*{pagina}`

Para poder usar colores para la marca de agua (y en el caso de usar tablas de colores) se puede agregar en el preámbulo el paquete **xcolor**

```
\usepackage[dvipsnames,usenames,xl1names,table]{xcolor}
\definecolor{LIGHTGRAY}{gray}{.95}
```

Ahora, por ejemplo, si se quiere agregar la imagen LogoCIEMAC.pdf en la página pequeña 1 se puede utilizar el **código**:

```
\AddToBackground{1}{% Fondo de la p'agina peque~na 1
\put(10,450){\includegraphics[scale=0.5]{LogoCIEMAC}}}
```

El comando `\put(10,450)` indica el lugar de la página pequeña 1 donde se pondrá la imagen, la posición (0,0) es en la esquina inferior izquierda de la página y las unidades están en pixeles.

Esta imagen se utilizará como el título del panfleto, para hacer que haya espacio para esta imagen se puede definir el título como `\title{\vspace*{5cm}}`. Si se quiere colorear el fondo de una de las páginas pequeñas se puede utilizar el **código**:

```
\AddToBackground{6}{% Fondo de la p'agina peque~na 6
\put(0,0){\textcolor{Cyan!50}{\rule{\paperwidth}{\paperheight}}}}
```

Este coloca en la página pequeña 6 un rectángulo del tamaño de la página completa (con `\rule{\paperwidth}{\paperheight}`) de color **Cyan** degradado en un 50%.

Con este comando también se pueden poner rectángulos pequeños o líneas de colores cambiando el ancho y el alto de la regla, por ejemplo, para una línea se puede utilizar `\rule{\paperwidth}{3}`

También se puede poner un texto como marca de agua de alguna de las páginas, por ejemplo, para agregar el texto CIEMAC al fondo de la página grande 2 se puede utilizar el **código**:

```
\AddToBackground*{2}{% Fondo de la p'agina grande 2
\put(\LenToUnit{.5\paperwidth},\LenToUnit{.5\paperheight}){
\makebox(0,0)[c]{
\resizebox{.9\paperwidth}{!}{\rotatebox{35.26}{
\textsf{\textbf{\textcolor{LIGHTGRAY}{CIEMAC}}}}}}}
```

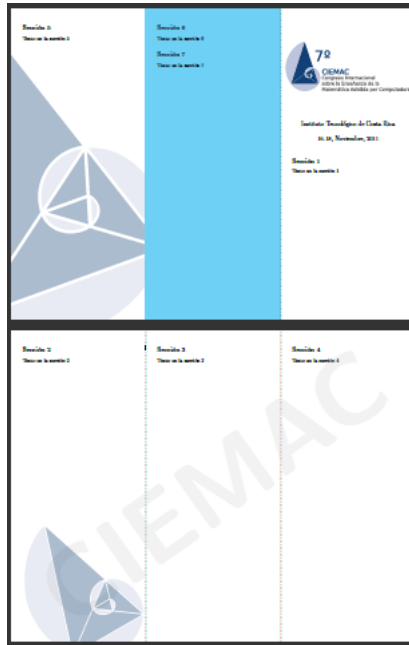
El comando `\LenToUnit` convierte la unidad dada a pixeles. `\makebox` crea una caja (en principio de tamaño 0x0), `\resizebox` le cambia el tamaño a la caja para hacerla un poco más pequeña que el tamaño total de la página, `\rotatebox` rota la caja (en este caso 35.26 grados). Por último la palabra CIEMAC se puso en serif, negrita y con color gris claro (color que se definió en el preámbulo).

Para terminar, se pueden agregar algunas imágenes rotadas al fondo de las páginas grandes 1 y 2.

```
\AddToBackground*{1}{% Fondo de la p'agina grande 1
\put(0,\LenToUnit{.5\paperheight}){
\makebox(0,0)[c]{
\resizebox{.9\paperwidth}{!}{\rotatebox{20}{
\includegraphics[scale=1.5]{IconoCIEMACTransparente}}}}}}
\AddToBackground*{2}{% Fondo de la p'agina grande 2
\put(130,100){
\makebox(0,0)[c]{
\resizebox{.4\paperwidth}{!}{\rotatebox{30}{
\includegraphics[scale=1]{IconoCIEMACTransparente}}}}}}
```

El resultado de agregar todos estos elementos se puede ver en la figura 13.9.





**Figura 13.9.** Agregando imágenes de fondo con leaflet.



Hay varias herramientas de conversión de **LaTeX** a **HTML**, pero usualmente funcionan bien solo con documentos sencillos y con un conjunto moderado de paquetes.

Si tenemos documentos más complejos, una salida **HTML** fiel y correcta, requiere una herramienta de conversión que nos permita cierta configuración.

En la tabla se puede ver algunas opciones para la conversión de un archivo con nombre `archivo.tex`. Las herramientas se aplican con un comando desde la *terminal* (consola).

| Herramienta        | Comando de uso (ejemplo)                   |                                                              |
|--------------------|--------------------------------------------|--------------------------------------------------------------|
| ☆ Pandoc           | <a href="http://pandoc.org">pandoc.org</a> | <code>pandoc archivo.tex -s -katex -o salida.html</code>     |
| ☆ lwrap (lwrapmk)  | <a href="#">CTAN lwrap</a>                 | <code>lualatex -shell-escape archivo.texlwrapmk html</code>  |
| ★ make4ht (tex4ht) | <a href="#">CTAN make4ht</a>               | <code>make4ht -u -c config.cfg archivo.tex "mathjax"</code>  |
| ☆ latex2html       | <a href="#">CTAN latex2html</a>            | <code>latex2html -split +0 -no_navigation archivo.tex</code> |

**Tabla 14.1.** Herramientas para convertir **LaTeX** a **HTML**.

La opción más configurable para documentos complejos es Make4ht. Pero tiene una curva de aprendizaje un poco alta. La solución que presentamos aquí es usar *una plantilla ya preparada*. Esta plantilla usa entornos basados en el paquete `mdframed`.

## 14.1 Make4ht

Un documento `.tex` se puede compilar de varias maneras. Podemos tener una salida bimodal: un documento PDF y un documento HTML (para añadir contenido interactivo).

El formato HTML ofrece accesibilidad, interactividad y un mayor alcance en el entorno digital actual. Generar el HTML desde un archivo LaTeX permite aplicar revisiones y/o modificaciones al archivo `.tex` sin necesidad de tener que volver a editar manualmente la salida HTML.

Mientras el formato PDF es ideal para impresión y citación formal, el HTML se adapta a cualquier dispositivo y permite integrar elementos dinámicos como visualizaciones, actividades interactivas y contenido multimedia. Este enfoque promueve la ciencia abierta y la educación digital al hacer el contenido más flexible y accesible.

Para obtener un documento PDF, compilamos con PDFLaTeX. Para traducir el documento `.tex` y obtener un sitio web, con su archivo de estilo `.css`, compilamos con Make4ht. El archivo `.css` se modifica y/o se amplía con un archivo personalizado `config.cfg`. En la Figura 14.1 se muestra el proceso de manera simplificada.



**Figura 14.1.** Compilar con PDFLaTeX o con Make4ht (esquema simplificado)

Tenemos un comprimido [plantilla.zip](#) con un archivo `.tex` y que también contiene un archivo de configuración ya implementado `config.cfg`, todo listo para usar.

Un artículo completo de cómo usar la plantilla se puede ver en

W. Mora-Flores. (2025) Generar una Web Interactiva con LaTeX (usando Make4ht e IA)

La plantilla `.tex` está implementado con código que ejecutan tanto PdfLaTeX como Make4ht. Pero hay partes en que solo nos interesa una salida HTML y otras en las que solo nos interesa la salida PDF, para esto, además del texto común, usamos comandos que PDFLaTeX ignora y que Make4ht sí compila y viceversa.

Make4ht se enfoca en la estructura del contenido y, si no hay errores de compilación fatales, entrega un archivo HTML correcto pero con un diseño simple. Este artículo es un ejemplo (y al mismo tiempo es una plantilla) de cómo agregar diseño web moderno y *responsivo*, de manera manual, a través de un archivo de configuración.

Este documento considera las *tareas comunes* en la implementación de material didáctico en matemáticas y agrega un archivo de configuración para la salida moderna y correcta en HTML. La manera de configurar Make4ht para lograr este objetivo se describe en la sección ??

### 14.1.1 Una plantilla.tex configurada

Tenemos un comprimido [plantilla.zip](#) que viene con un archivo `plantilla.tex`, con figuras y un archivo de configuración `config.cdf` (entre otras cosas) que PDFLaTeX compila bien y Make4ht traduce bien a HTML.

Asumimos que tenemos una instalación TeX *completa* y actualizada al 2025 (las más populares son TeXLive, MikeTeX y MacTeX). Esta actualización ya viene con Make4ht. Esto es necesario porque varias cosas funcionan bien con la última actualización, sino podríamos tener errores de compilación o comportamientos inesperados.

En el código podemos ver el inicio de la `plantilla.tex`

```
\documentclass[fleqn,oneside]{article} % Puede usar también book, etc.
\input{Paquetes/WebPreambuloyEntornos.tex}
\input{Paquetes/ComandosdelUsuario.tex}
\title{El título}
\author{El autor}
```

```

\date{}
\ifdefined\HCode\Configure{HEAD}{% Insertar en el html<head></head>
 \HCode{<link rel="stylesheet" href="css/toc-toggle.css" />}
}\fi
%
\begin{document}
\maketitle
\solopdf{% PDFLaTeX->pdf
 \tableofcontents
}

\section{Introducción}
%... código LaTeX usual...
\end{document}

```

Podemos agregar paquetes y comandos personales, en el archivo `ComandosdelUsuario.tex` que está en la carpeta “Paquetes”. Make4ht soporta una lista amplia de paquetes esenciales: Incluye los paquetes que usualmente usamos. Mantenga los comandos personales “simples”.

Hay varios paquetes como Nicematrix, Polynomial, hhline, etc. que *no* son soportados por Make4ht. Si llamamos paquetes no soportados por Make4ht desde un archivo “ComandosdelUsuario.tex”, hay que “envolverlos” con el código

```

% Para compilar con Windows | Linux no tiene este problema
\ifdefined\HCode\else
% Paquetes
% no soportados por Make4ht
\fi

```

Si tenemos código en el archivo `.tex` con paquetes no soportados por Make4ht, como veremos más abajo, usamos el comando `\solopdf{...}` para que Make4ht los ignore y, en el `.html`, se pueden incluir imágenes o podemos usar código equivalente (simplificado) en `.tex`, que sea soportado por Make4ht y usar el comando `\solomk{...}` (Make4ht lo compila pero PDFLaTeX lo ignora), o insertar directamente un equivalente en código HTML, con el comando `\insertarhtml{...}`. Use la IA como asistente, es buena idea.

La estructura de la carpeta de trabajo se muestra en la Figura 14.2.

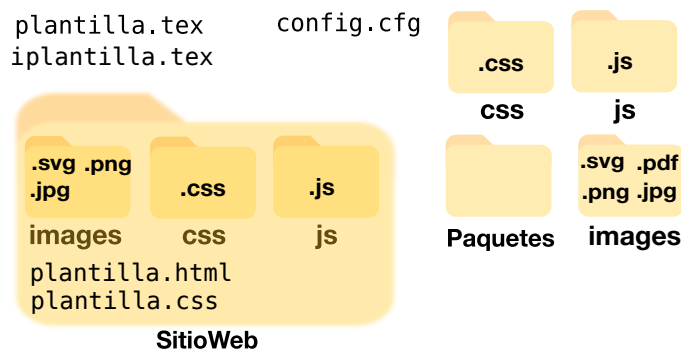


Figura 14.2. Carpeta de trabajo (plantilla.zip)

### 14.1.2 Compilar esta plantilla

Make4ht ofrece varias opciones de compilación a través de un archivo de configuración y/o con opciones entre comillas. En este documento mayormente compilamos en la terminal | cmd | consola, la línea de compilación se muestra en la el código

---

```
make4ht --utf8 -c config.cfg -f html5 -d SitioWeb plantilla.tex
 "mathml,svg,Gin-dim"
% opción limpiar: agregar -m clean (y quitar después)
% opción debug: agregar -a debug
```

---

**N** En esta plantilla, el archivo `config.cfg` puede cargar el script MathJax más personalizado, si se ocupan paquetes y comandos adicionales.

1. La opción `-c config.cfg` “inyecta” nuestro código, configuración de entornos y código de estilo CSS, en el archivo `plantilla.css`, de tal manera que prevalezca sobre el código CSS que genera Make4ht.
2. La opción `-d SitioWeb` inserta en `SitioWeb`, en la carpeta de trabajo, los archivos `.css` y HTML (si no está la carpeta, la crea).

En esta plantilla tenemos algunos comandos personalizados (están en `WebPreamble\Entornos.tex`):

|                                                 |                                                                               |
|-------------------------------------------------|-------------------------------------------------------------------------------|
| <code>\solopdf{#1}:</code>                      | PDFLaTeX ejecuta el código y Make4ht lo ignora.                               |
| <code>\solomk{#1}:</code>                       | Make4ht ejecuta el código ... y PDFLaTeX lo ignora.                           |
| <code>\insertarhtml{#1}:</code>                 | Código HTML <i>puro</i> con Make4ht (PDFLaTeX lo ignora)                      |
| <code>\ifdefined\HCode ... \fi:</code>          | Make4ht ejecuta el código y PDFLaTeX lo ignora.                               |
| <code>\ifdefined\HCode ... \else... \fi:</code> | PDFLaTeX ejecuta el código después ... de <code>\else</code> .                |
| <code>\linea:</code>                            | Una línea gris delgada, que es apropiada para HTML y PDF                      |
| <code>\bigskip:</code>                          | Para Make4ht se ha definido como un <code>&lt;div&gt;</code> de cierta altura |

**Ejemplo.** En el código que sigue se muestra cómo incluir una figura `.pdf` con PDFLaTeX e incluir la misma figura en formato `.svg` en HTML (Make4ht puede hacer la conversión automática de `.pdf` a `.svg`, pero depende del sistema, las librerías instaladas y del tipo de archivo `.pdf`).

---

```
\solopdf{% PDFLaTeX-> pdf
\begin{center}
\includegraphics[scale=0.35]{images/sumaRiemann.pdf}
```

```

\captionof{figure}{Figura {\tt .pdf}}
\end{center}
}
\solomk{% Mkae4ht -> html
\begin{center}
\includegraphics{images/sumaRiemann.svg}
\captionof{figure}{Figura {\tt .svg}}
\end{center}
}

```

---

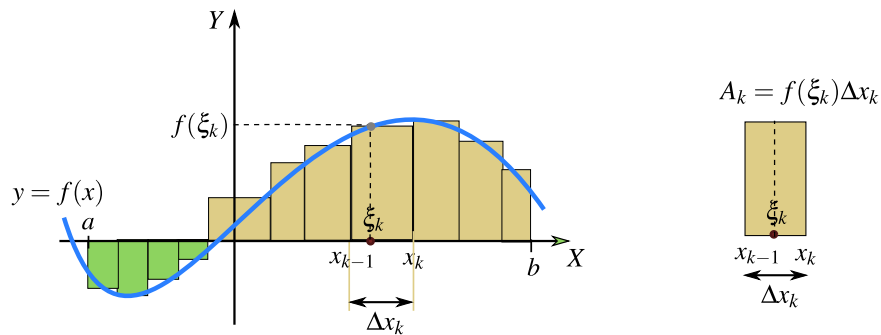


Figura 14.3. Figura .pdf

En este caso, la figura sumaRiemann.svg ha sido escala adecuadamente (con Inkscape) para la salida .html. Hay varios programas para escalar figuras.

### 14.1.3 Un sitio web de ejemplo

Un sitio web completo se puede generar con LaTeX y Make4ht, incluyendo menús de navegación. Recordemos que el propósito es poder modificar el contenido .tex sin tener que modificar nada en el HTML (pues se genera automáticamente). Es posible usar la IA para obtener código .tex, de acuerdo a alguna especificación, que se traduzca a HTML o código HTML puro, que se pueda insertar directamente con nuestro comando `\insertarhtml{...}`.

Un ejemplo se puede ver en [Cálculo Superior Interactivo](#)



Índice

Lugar geométrico

Ecuación Canónica

Ejercicios

### 3. Hipérbolas

Una Hipérbola, en un plano, se puede definir de varias maneras. En nuestro caso solo consideramos una definición como "lugar geométrico" y otra definición en términos de una ecuación general.

**Definición 1 (La hipérbola como lugar geométrico.).**

En un plano, una hipérbola es el lugar geométrico de todos los puntos  $Q$  tales que el valor absoluto de la diferencia de sus distancias a dos puntos fijos del plano,  $F_1$  y  $F_2$ , (llamados *focos*), es constante. Si llamamos a la constante  $2a$ , entonces

$$|d(Q, F_1) - d(Q, F_2)| = 2a \text{ con } 0 < 2a < d(F_1, F_2)$$

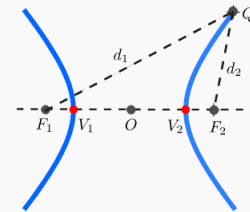


Figura 1: Hipérbola

Pensar | Observar

Figura 14.4. Sitio iCS





---

Última versión actualizada y *comprimido* con los ejemplos de este libro:  
<https://tecdigital.tec.ac.cr/servicios/revistamatematica/Libros/>  
Esta versión: Marzo, 2022.

---

# 15

## Algunos Principios Básicos para Explorar Código (La)TeX

Aunque **(La)TeX** es relativamente fácil como software para edición, **(La)TeX** como language de programación tiene partes complicadas, empezando porque no se parece a los lenguajes de programación usuales y tiene sus “idiosincrasias”: Los conceptos de “tokens” y “expansión” de **TeX** son bastante ajenos a las experiencias de la gente con un lenguaje de programación.

En Internet se pueden encontrar trozos de código para resolver problemas específicos, por ejemplo en [TeX - LaTeX Stack Exchange](#). Este capítulo solo pretende mencionar algunas cosas muy básicas del lenguaje con el propósito de poder explorar y/o modificar estos trozos de código **(La)TeX**.

El código que encontremos puede estar escrito de varias formas: en “plain **TeX**”, en **LaTeX**, o **LaTeX** usando algún o algunos paquetes. No hay problema en usar código **TeX** en documentos **LaTeX**.

### 15.1 Variables (registros)

En **TeX** hay varios tipos de registros, algunos ya están definidos (primitivas del lenguaje) y también se pueden definir nuevos registros. Entre los registros más usuales están

- Contadores: En **TeX** los contadores nuevos se declaran con `\newcount\nombre` y en **LaTeX** se declaran con `\newcounter{nombre}`
- Dimensiones: En **TeX** las dimensiones nuevas se declaran con `\newdimen\nombre` y en **LaTeX** se declaran con `\newlength{nombre}`
- Glue: **TeX** utiliza un modelo de cajas y “glue” (gotas de goma: espacios en blanco de diferente tamaño). Las cajas contienen objetos tipográficos, como texto, texto matemático e imágenes, y la “goma en gotas” son espacios flexibles que pueden estirarse y/o encogerse en cantidades que están bajo el control del usuario. “glue” es una dimensión (distancia) con posiblemente componentes de estiramiento y/o encogimiento. En **TeX** las nuevas variables de tipo “glue” se declaran con `\newskip`. Variables primitivas tipo “glue” son `\hskip`, `\vskip`, `\mskip`, etc. En general tienen tres parámetros

```
\hskip <dimen1> plus <dimen2> minus <dimen3>
```

```

\vskip <dimen1> plus <dimen2> minus <dimen3>
\mskip <dimen1> plus <dimen2> minus <dimen3>
...
etc.
% Nuevo
\newskip\miskip = 10mm plus 12 mm minus 3mm

```

- Listas de “tokens” (fichas): En **TeX** las listas nuevas se declaran con `\newtoks` nombre, por ejemplo

```

\newtoks\mistoks \mistoks = {abcd Ef}

```

**Cálculos con registros.** En plain **TeX** se pueden hacer cálculos con registros (numéricos o aún simbólicos) y *hay que comprender y ser muy cuidadoso de las reglas sintácticas para cada tipo de registro* (< number>, <dimen>, <skip>, <toks>)

En particular, `<number>` puede ser un contador, un caracter declarado con `\chardef`, un número entero o decimal, `numexpr <...>`. De esta manera tenemos tres operaciones básicas

- Sumar: `\advance <registro>by <valor>`. El <registro> es tipo contador o tipo dimensión. El valor es <number> o <dimen> dependiendo del tipo de <registro>
- Multiplicar: `\multiply <registro>by <number>`.
- Dividir: `\divide <registro>by <number>`. Si <registro> es tipo <number> entonces el resultado se trunca.

En **(La)TeX** no hay muchas opciones de cálculo, pero se pueden hacer cálculos “naturales” o también más avanzados con paquetes como `sagetex`, `fp`, `calc`, `luacode`, `pgfmath`, etc.

## Contadores (variables entera)

**(La)TeX** usa contadores para capítulos, secciones, teoremas, etc. Estos contadores se pueden usar con otros propósitos para algunos cálculos simples (en aritmética entera). Un contador es una variable entera. En **LaTeX**, `\newcounter{newcnt}` crea la variable entera con nombre `newcnt` (este nombre solo admite letras), inicializada en 0. Además se crea `\thnewcnt` y con este comando se imprime el valor del contador `newcnt`. Para cambiar el valor de `newcnt` a  $n$  usamos `\setcounter{newcnt}{n}`

La primitiva `\the` de **TeX** toma un registro (digamos un parámetro interno) y convierte su valor a una “string”, para imprimir.

Los contadores son variables *globales* y en el interior de los entornos (grupos) `\begin{entorno}... \end{entorno}` se comporta de manera *local*. Si queremos un comportamiento como variable global en un entorno debemos agregar `\global`

| Contadores en LaTeX                           |                                                                                                                                                       |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\newcounter{newcnt}</code>              | contador de nombre newcnt inicializado en 0. En principio se define el contador interno <code>\c@newcnt</code> que mantiene el valor numérico actual. |
| <code>\thnewcnt</code>                        | imprime <code>\arabic{newcnt}</code>                                                                                                                  |
| <code>\setcounter{newcnt}{num}</code>         | asigna num como valor global de newcnt                                                                                                                |
| <code>\addtoresetcounter{\newcnt}{num}</code> | incrementa newcnt en num (puede ser negativo).                                                                                                        |
| <code>\value{newcnt}</code>                   | valor de newcnt. Se usa en cualquier contexto en que <b>LaTeX</b> espere un número.                                                                   |
| <code>\stepcounter{newcnt}</code>             | incrementa newcnt en uno                                                                                                                              |
| <code>\refstepcounter{newcnt}</code>          | incrementa newcnt en uno                                                                                                                              |

También están los registros `\muskip` (“glue” en modo matemático) y `\box` (cajas).

También en “plain TeX” se puede escribir `\newcount\newctr` y para cambiar el valor podemos escribir `\newctr=n` y usamos `\the\newctr` para imprimir el valor.

| Contadores en “plain TeX”          |                                                       |
|------------------------------------|-------------------------------------------------------|
| <code>\newcount\newctr</code>      | contador de nombre newctr                             |
| <code>\newctr=n</code>             | asigna el valor n a newctr                            |
| <code>\the\newctr</code>           | imprime el valor de newctr                            |
| <code>\advance\newctr by k</code>  | suma k a newctr                                       |
| <code>\multiply\newctr by k</code> | multiplica el valor de newctr por k (división entera) |
| <code>\divide\newctr by k</code>   | divide el valor de newctr por k                       |

### Ejemplo 15.1

En este ejemplo creamos una variable (contador) `ansj` de dos maneras, le asignamos como valor el número de capítulo y luego imprimimos como texto “El número actual es” seguido del valor de `ansj` aumentado en 5

El **código: LaTeX**

```
\newcounter{ansj}
\setcounter{ansj}{\thechapter}
El número actual es \theansj
\setcounter{ansj}{\theansj+5}
El número actual es \theansj
```

**produce:**

El número actual es 15

El número actual es 20

El **código: TeX**

```
\newcount\ansj
\ansj=\thechapter
El número actual es \number\ansj
\advance\ansj by 5
El número actual es \number\ansj
El número actual es \the\ansj
```

**produce:**

El número actual es 15

El número actual es 20

El número actual es 20

## Expansión.

La “expansión” es, en general, “el proceso de sustitución del contenido de las macros” en lugar de la llamada de las macros.

Por ejemplo, si definimos `\newcommand\test{mi texto}`, entonces cuando aparece el comando `\test` este comando expande a “mi texto”. Las complicaciones vienen cuando en vez de “mi texto”, hay expresiones más complejas. Ver por ejemplo [What is "expansion"?](#).

Bien, `\value{newcnt}` *expande* al contador interno `\c@newcnt`. De esta manera, `\advance`, `\multiply`, `\divide` se puede usar con contadores creados con `\newcounter` usando `\value`

### código: LaTeX

```
\newcounter{x}
\setcounter{x}{5} % o \value{x}=5
\advance\value{x} by 5
\multiply\value{x} by 2
```

El valor actual es `\thex`

### código: TeX

```
\newcount\x
\x=5
\advance\x by 5
\multiply\x by 2
```

El valor actual es `\the\x`

En ambos casos `\thex` y `\the\x` imprimen 20.

## Variable global, variable local.

Los contadores son variables globales. Si se modifican en un entorno, su valor es local en ese entorno. Fuera del entorno recobra su valor global.

Por ejemplo, el **código**:

```
\newcounter{x}
\setcounter{x}{5}
\advance\value{x} by 5 %x=10

\begin{minipage}{6cm}
\multiply\value{x} by \value{x}
%x=10*10=100
```

En este entorno el valor actual es `\thex`  
`\end{minipage}`

El valor global actual es `\thex` `%x=10`

```
\newcounter{x}
\setcounter{x}{5}
\advance\value{x} by 5 %x=10

\begin{tabular}{l}
\multiply\value{x} by \value{x}
%x=10*10=100
```

En este entorno el valor actual es `\thex\`  
`\end{tabular}`

El valor global actual es `\thex` `%x=10`

### produce:

En este entorno el valor actual es 100

El valor global actual es 10

---

El comando `\global` deshabilita el comportamiento local en los entornos

El código:

```
\newcounter{x}
\setcounter{x}{5}
\advance\value{x} by 5 %x=10
\begin{tabular}{l}
\global\multiply\value{x} by \value{x} %x=100
El valor actual (global) es \thex\
\end{tabular}
```

El valor actual es \thex %x=100

produce:

El valor actual (global) es 100

El valor actual es 100

### Ejemplo 15.2 ( $n \bmod m$ )

La expresión  $n \bmod m$  calcula el residuo  $r$  de dividir  $n$  por  $m$  (división entera), con  $0 \leq r < m$ . Se tiene  $n = mk + r$  con  $k = n/m$  (división entera) y entonces el residuo es

$$r = n - (n/m) * m$$

El comando `\nmodm{a}{b}` devuelve el residuo de dividir  $a$  por  $b$ , es decir  $a \bmod b$

```
\newcount\residuo
\newcommand{\nmodm}[2]{\residuo=#1 % n
\divide\residuo by #2 % n/m (división entera)
\multiply\residuo by #2 % m*(n/m)
\multiply\residuo by -1 % -m*(n/m)
\advance\residuo by #1 % n-m*(n/m)
\the\residuo}
```

De esta manera

- `\nmodm{512}{7}` produce: 1
- `\nmodm{7}{7}` produce: 0
- `\nmodm{11}{4}` produce: 3

## Aritmética con `\numexpr`

Podemos hacer aritmética simple usando `\numexpr`. Este comando funciona como un contador. Para este comando, la división no es división entera, más bien una división redondeada al entero más cercano.

Usualmente usamos `\the\numexpr... \relax`. El comando `\relax` se usa aquí para evitar que `\numexpr` lea más cosas y se confunda (como si fueran cosas que también intervienen en el cálculo). Por ejemplo

- `\the\numexpr 1 + 2 * 3 \relax` produce: 7
- `\the\numexpr 11/4 \relax` produce: 3 (pues  $11/4 \approx 2.75$ )

- `\newcounter{dn}\setcounter{dn}{5}\the\numexpr 5*\thedn+2\relax` produce: 27

- Podemos definir un comando para la función módulo si permitimos que el residuo pueda tomar valores enteros negativos (no hay problema en eso, aunque no es la función **mod** estándar)

```
\newcommand{\amodb}[2]{\the\numexpr #1-#2*(#1/#2)\relax} %división por redondeo
```

De esta manera

- `\amodb{512}{7}` produce: 1
- `\amodb{7}{7}` produce: 0
- `\amodb{11}{4}` produce: -1 (en efecto:  $11 = 4 * 3 - 1$ )

Por supuesto, lo podríamos “arreglar” así

```
\newcommand{\amodb}[2]{\newcount\aa \aa =#1 \divide\aa by #2\relax
 \the\numexpr#1-#2*\aa\relax}
```

De esta manera

- `\amodb{512}{7}` produce: 1
- `\amodb{7}{7}` produce: 0
- `\amodb{11}{4}` produce: 3
- Podemos usar una listas de enteros  $\{2, 3, \dots, 8\}$  para generar una matriz triangular  $8 \times 8$  con el paquete **tikz**

El código:

```
\documentclass[tikz]{article}
\begin{document}
 \begin{tikzpicture}[scale=1]
 \foreach \n in {1,...,8} {
 \foreach \m in {1,...,\numexpr9-\n} {
 \draw (\m,-\n) node{$a_{\n\m}$};
 }
 }
\end{tikzpicture}
\end{document}
```

produce:

$a_{11}$     $a_{12}$     $a_{13}$     $a_{14}$     $a_{15}$     $a_{16}$     $a_{17}$     $a_{18}$   
 $a_{21}$     $a_{22}$     $a_{23}$     $a_{24}$     $a_{25}$     $a_{26}$     $a_{27}$   
 $a_{31}$     $a_{32}$     $a_{33}$     $a_{34}$     $a_{35}$     $a_{36}$   
 $a_{41}$     $a_{42}$     $a_{43}$     $a_{44}$     $a_{45}$   
 $a_{51}$     $a_{52}$     $a_{53}$     $a_{54}$   
 $a_{61}$     $a_{62}$     $a_{63}$   
 $a_{71}$     $a_{72}$   
 $a_{81}$

## Longitudes (dimensiones)

Las “longitudes” determinan varias dimensiones de los documentos, y también son unidades de distancia relativas a algunos elementos del documento.

Las unidades básicas son mm, cm, pt, inch, em (aproximadamente el ancho de una ‘M’ en la fuente actual), ex (aproximadamente la altura de una “x” en la fuente actual), etc.

En **LaTeX** `\newlength` asigna un nuevo registro (con valor 0cm) que puede ser usado en cualquier lugar y se le puede asignar valores con `\setlength` y es global. Dentro de un entorno `\setlength` actúa localmente.

Hay longitudes predefinidas (primitivas), como `\textwidth`, `\baselineskip`, `\columnsep`, etc. Y en este caso solo usamos `\setlength` o también asignamos valores de manera directa:

```

\setlength{\columnsep}{1in} o \columnsep=1in

\includegraphics[width = 0.25\textwidth]{lion-logo}

```

En código de bajo nivel (plain **TeX**) se usa `\newdimen`

| Dimensiones en LaTeX                     |                                                                                                                                                                                                    |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\newlength{milong}</code>          | Define un nuevo comando milong de longitud, con valor 0 cm.                                                                                                                                        |
| <code>\setlength{milong}{valor}</code>   | Asigna un valor al comando de longitud milong. El comando milong puede ser una primitiva; por ejemplo podemos cambiar el valor de <code>\parskip</code> con <code>\setlength{\parskip}{1mm}</code> |
| <code>\addtolength{milong}{valor}</code> | Incrementa el valor del comando milong en valor                                                                                                                                                    |
| <code>\settowidth{milong}{texto}</code>  | Asigna el valor de un comando milong como la longitud de texto. <code>\settowidth{milong}{\it Mi idea}</code> asigna a milong el ancho del texto <code>Mi idea</code>                              |



|                           |                                                                                                                                                                                                                 |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\settoheight</code> | Comando que actúa como un ajuste a la altura. Por ejemplo, <code>\settoheight{\parskip}{Idea}</code> asigna al comando <code>\parskip</code> la altura de la letra “I”                                          |
| <code>\settodepth</code>  | Comando que actúa como un ajuste a la altura. Por ejemplo, <code>\settodepth{\parskip}{pato}</code> asigna al comando <code>\parskip</code> la distancia en la que letra “p” se extiende por debajo de la línea |

El valor de un comando creado con `\newlength` puede ser cambiado en cualquier momento. Esto también es cierto para algunos de los parámetros de longitud, mientras que otros deben cambiarse sólo en el preámbulo y algunos nunca deben cambiarse.

`< dimen >` es lo que **TeX** espera ver cuando necesita indicar una dimensión (positiva o negativa).

| Dimensiones en “plain TeX”         |                                                                                                                                    |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <code>\dimen &lt;number&gt;</code> | Registro para almacenar dimensiones (numerado de <code>\dimen0</code> a <code>\dimen255</code> . Los primeros 10 están reservados. |
| <code>\newdimen\midimen</code>     | Inicializa una nueva variable de tipo “dimensión” <code>\midimen</code>                                                            |

### Ejemplo 15.3 (Dimensiones con LaTeX).

Este es un ejemplo general que define una dimensión `\wleng`, luego se le asigna un valor (global) y este valor se redefine en un entorno (asignación local)

El código:

```
\documentclass{article}
\begin{document}
\newlength{\wleng}
\setlength{\wleng}{30pt} % actúa globalmente

Una regla {\tt 40pt x 30 pt} \rule{40pt}{\wleng}

{% @<--- \begingroup
\setlength{\wleng}{10pt} % redefine \wleng localmente, en este grupo
Una regla {\tt 40pt x 10 pt} \textcolor{magenta}{\rule{40pt}{\wleng}}
}% @<--- \endgroup

En un entorno \latex+tikzpicture+: línea de largo 0.8\textwidth-1cm

\begin{tikzpicture}
\setlength{\wleng}{0.8\textwidth-1cm} % redefine \wleng localmente
\draw [gray,thick] (0,0) -- (\wleng,0) node[right]
{\verb+0.8\textwidth-1cm+};
\end{tikzpicture}
```

Ahora volvemos al valor global: Una regla `{\tt 40pt x 30 pt} \rule{40pt}{\wlength}`  
`\end{document}`

produce:

Una regla 40pt x 30 pt 

Una regla 40pt x 10 pt 

En un entorno `tikzpicture`: línea de largo `0.8\textwidth-1cm`

 `0.8\textwidth-1cm`

Ahora volvemos al valor global: Una regla 40pt x 30 pt 

**Plain TeX.** Con las dimensiones podemos aplicar las operaciones `\advance`, `\multiply` y `\divide`, siempre que lo hagamos usando las reglas de manera correcta. Imprimimos con `\the` y el resultado se expresa en unidades pt

#### Ejemplo 15.4 (Dimensiones con plain TeX).

El código:

```
\newdimen\wdim \wdim = 0.4pt
1.) Valor de \verb+\wdim+ es \the\wdim

\advance\wdim by 2pt
2.) Valor de \verb+\wdim+ es \the\wdim

\multiply\wdim by 2
3.) Valor de \verb+\wdim+ es \the\wdim

\divide\wdim by 5
4.) Valor de \verb+\wdim+ es \the\wdim
```

produce:

- 1.) Valor de `\wdim` es 0.4pt
- 2.) Valor de `\wdim` es 2.4pt
- 3.) Valor de `\wdim` es 4.79999pt
- 4.) Valor de `\wdim` es 0.95999pt

#### Aritmética con `\dimexpr`

Con `\dimexpr` podemos hacer operaciones con `+`, `-`, `*` y `/` con paréntesis, es decir, `(/)`.

Por ejemplo, las siguientes operaciones son legales

El código:

```
1.) \the\dimexpr 1pt + 1pt\relax
2.) \the\dimexpr 1pt * 10\relax
3.) \the\dimexpr (1pt + 1pt) / (1 + 1)\relax
4.) \the\dimexpr 5pt * (3 / 2) \relax
5.) \the\dimexpr 5pt * (4 / 3) \relax
```

produce:

```
1.) 2.0pt
2.) 10.0pt
3.) 1.0pt
4.) 10.0pt
5.) 5.0pt
```

- Observe que `\the\dimexpr 10 * 1pt\relax` no es legal y produce un error.
- Mientras `\numexpr` produce enteros, `\dimexpr` produce dimensiones. Dentro de `\numexpr` una dimensión se convierte en un entero que representa su valor en “scaled points” sp (1sp = 1/65536pt)

En el código que sigue, la expresión

```
\dimexpr \numexpr \A * \A / \B \relax sp
```

convierte la dimension resultante a sp.

El código:

```
\newlength\A
\newlength\B
\newlength\C
\setlength\A{10pt}
\setlength\B{5pt}
\setlength\C{\dimexpr \numexpr \A * \A / \B \relax sp\relax}
Imprimir:\\ \the\C
```

produce:

Imprimir:  
20.0pt

## 15.2 Condicionales

Primero, como vamos a comparar cosas, debemos observar los “códigos”. Cada carácter que **TeX** lee, se le asocia un código ASCII y/o un código de categoría (catcode)

- character code: Internamente, **TeX** representa los caracteres con un número entero (código del carácter). La tabla de códigos de caracteres está basada en la tabla ASCII de 7 bits para números menores de 128. Por ejemplo, la letra H tiene código ASCII 72 y un espacio en blanco tiene código ASCII 32.
- catcode: Cuando **TeX** analiza el texto de entrada, asigna a cada carácter leído un código de categoría (catcode). Por ejemplo, la llave { se le asigna categoría 1, las letras del alfabeto se les asigna categoría 11, y más importante, \ se le asigna categoría 0. Los caracteres con categoría 0 se llaman “escape character” porque le dice a **TeX** que “salte a un modo de lectura” en el que tiene que analizar con cuidado que sigue, pues se trata del nombre de un comando (“control sequence”), no de texto ordinario. A veces es conveniente, en ciertos entornos, hacer que una letra, por ejemplo, tenga “catcode 0”.

Si escribimos `\catcode'\|=0` esto hace que `\` y `|` actúen como “caracteres de escape” y, entonces, `\micomando` y `|micomando` hacen lo mismo.

**TeX** tiene más de 15 comandos tipo `if`. La estructura general es

```
\if...<token 1><token 2><verdadero: texto>\fi
\if...<token 1><token 2><verdadero: texto>\else<falso: texto>\fi
```

`\if` Comprueba la igualdad de los códigos de los caracteres.

---

```
\if <token 1><token 2>{texto} [\else {texto}] \fi
```

El uso de `\if` puede ser muy complicado porque tiene sus sutilezas. La primera cosa importante es que `\if` expande los tokens hasta que encuentra los dos siguientes tokens *no expandibles*. `<token 1>` y `<token 2>` pueden ser un carácter o comandos. El test del `if` es verdadero si `<token 1>` y `<token 2>` se expanden al mismo código de carácter.

Si comparamos comandos que expanden a un solo carácter, no hay problema.

```
\def\miletra{x} %comando \miletra expande a x
\if a\miletra % compara 'a' con 'x' (\miletra se expandió)
a es \verb+\miletra+
\else
a no es \verb+\miletra+
\fi
% Resultado: " a no es \miletra"

\def\mletrab{x}
\if \mletrab\miletra{Son la misma}\fi
% Resultado: "Son la misma"

% Aquí no obtiene el resultado esperado
\def\{10}\def\{10}
\if\{10}\{10}{Verdadero}\else{Falso!}\fi %\if compara 10: es falso!
% Resultado: Falso!

%Pero...
\def\{11}\def\{10}
\if\{11}\{10}{Verdadero}\else{Falso!}\fi %\if compara 11: es verdadero!
% Resultado: 10Verdadero

% Más claramente...
\def\p{sb23c}
\if s\p{Verdadero}\else{Falso!}\fi % \p expande a sb23c
% aquí el if compara ss y el resultado es: b23cVerdadero
```

`\ifcat` Comprueba la igualdad de los códigos de categoría (catcode).

---

Con `\catcode`` asignamos un código de categoría. Por ejemplo, 1 es el código de categoría para abrir un grupo, `\catcode`<=1` hace que “<” actúe como inicio de grupo, es decir como “{”.

```
\catcode`<=1 % hace que < sea inicio de grupo
\let\b=< % hace a \b el caracter < (ahora es inicio de grupo)

%comparamos catcode de < y {
\ifcat <{{\verb+<+ es inicio de grupo}\fi %

% Resultado: < es inicio de grupo

%comparamos catcode de \b y {
\ifcat \b{{\verb+\b+ es inicio de grupo}\fi

% Resultado: \b es inicio de grupo

% Por defecto, '<' tiene catcode 12
\catcode`<=12 % Deshacer, '<' vuelve a ser "menor que"
```

`\ifx` Comprueba la igualdad de la expansión de dos macros (hasta un primer nivel), o la igualdad del código de carácter y el código de categoría.

---

```
\def\a{10}\def\b{10}
\ifx\a\b{Verdadero}\else{Falso!}\fi

% Resultado "Verdadero"

\def\es{10} \def\esd{\es}
%\es expande a 10\\
%\esd expande a \es y luego a 10 \\
%\ifx\es\esd, en un 1er nivel de expansión, compara 10 y \esd (falso)
\ifx\es\esd{Verdadero}\else{Falso!}\fi

% Resultado: Falso!
```

`\ifcase` Casos

---

```
\ifcase <number><case 0>\or<case 1>\or ... \or<case n>{texto}\else{texto} \fi
```

Se procesa la rama dada por el registro “<number>” y pasa al “`\else`” cuando no existe una rama con un “número” dado. No procesa algo si no se declara “`\else texto`”.

```
\ifcase\month\or
 January\or February\or March\or April\or May\or June\or
 July\or August\or September\or October\or November\or December\fi
\space {\magenta \number\day, \number\year}
```

al día de hoy **produce:**

September 8, 2025

**\ifnum** Comprueba las relaciones entre números (>, =, <)

```
\newcounter{num}\setcounter{num}{11}
Comparar enteros: \
\ifnum\value{num} > 50 {verdadero, es grande}
 \else {falso, es pequeño }\fi

% Resultado: falso, es pequeño
```

**\ifodd** Comprueba si un número es impar

```
\newcount\nlinea \nlinea=11
Comparar enteros: \
\ifodd\number\nlinea {\advance\nlinea by 1 \the\nlinea}\fi %true

% Resultado: 12
```

**\ifhmode** Comprueba si el modo actual es el modo horizontal (posiblemente restringido)

**\ifvmode** Comprueba si el modo actual es el modo vertical (posiblemente interno)

**\ifmmode** Comprueba si el modo actual es el modo matemático (posiblemente interno)

**\ifinner** Comprueba si el modo actual es un modo interno

**\ifdim** Compara dos dimensiones.

```
\if <dimen_1><Relación><dimen_2>{texto} [\else {texto}] \fi
```

<dimen\_1> y <dimen\_2> solo pueden ser constantes (como 2pt), registros de dimensión (como \dimen7) o parámetros de dimensión (como \parindent)

```
\documentclass{book}
\begin{document}

\dimen0 = 1000pt \ifdim \dimen0 > 3pt true\else false\fi

%Resultado: true

\newdimen\mydim \mydim=125pt
\ifdim \mydim > 100pt \relax
 \ifdim \mydim < 150pt \relax
 Está en el rango%
 \else
 Está arriba de 150pt%
 \fi
\else
```

```

 Está abajo de 100pt%
\fiEstá abajo de 100pt%
\fi

%Resultado: Está en el rango%
\end{document}

```

|                       |                                                                          |
|-----------------------|--------------------------------------------------------------------------|
| <code>\ifvoid</code>  | Comprueba si un registro de caja está vacío                              |
| <code>\ifhbox</code>  | Comprueba si un registro de caja contiene una caja horizontal            |
| <code>\ifvbox</code>  | Comprueba si un registro de caja contiene una caja vertical              |
| <code>\ifof</code>    | Comprueba si el flujo de entrada ha finalizado o si el archivo no existe |
| <code>\iftrue</code>  | Una prueba que siempre es verdadera                                      |
| <code>\iffalse</code> | Una prueba que siempre es falsa.                                         |

Los comandos `\iftrue` y `\iffalse` se usan principalmente en la definición de macros para dar seguimiento del resultado de una macro que ejecuta alguna prueba.

|                     |                        |
|---------------------|------------------------|
| <code>\newif</code> | Crear una nueva prueba |
|---------------------|------------------------|

Podemos usar `\newif\ifTalCosa` para declarar `\ifTalCosa` como una variable booleana en el siguiente sentido:

- `\newif\ifTalCosa` define `\ifTalCosa = \iffalse`
- `\TalCosatrue` define a `\ifTalCosa=\iftrue`
- `\TalCosafalse` define a `\ifTalCosa=\iffalse`

```

\newif\ifTalCosa

\TalCosatrue % ahora \ifTalCosa=\iftrue
\ifTalCosa
Se cumple tal cosa
\else
No se cumple tal cosa
\fi

% Resultado: Se cumple tal cosa

```

**Paquete adicionales.** A veces se usa el paquete “`ifthen`” para ampliar el manejo de condicionales, por ejemplo con

```
\ifthenelse{<test>}{<then cláusula>}{<else cláusula>}
\whiledo{<test>}{<while cláusula>}
```

En la condición de `\ifthenelse` se puede usar expresiones con `\( \)`, `\AND`, `\OR`, `\NOT`, `\equal`

Pero para algo más eficiente se usa el paquete “`etoolbox`”

#### Ejemplo 15.5

El **código**:

```
\documentclass{article}
\usepackage[english,spanish,es-noshorthands]{babel}
% es-noshorthands % <-- desactiva «< >>» como comillas. TeXLive 2025
\usepackage{ifthen}
\begin{document}
\newcounter{num}\setcounter{num}{11}

Contador num "no es menor que 2" y "es menor o igual que 11"?
\ifthenelse{ \value{num} > 1 \AND \value{num} < 12 }{Verdadero}{Falso}
\end{document}
```

**produce:**

Contador num “no es menor que 2” y “es menor o igual que 11”? Verdadero

## 15.3 Ciclos

Los ciclos `whilenum` y `for` incluyen el símbolo “@” en su definición, entonces se deben usar entre `\makeatletter ... \makeatother`.

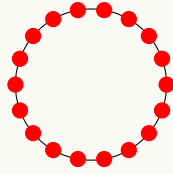
Tenemos:

- `\loop` {comandos} `\ifnum` condición `\repeat`
- `\@whilenum` condición `\do` { comandos }
- Del paquete **ifthen** tenemos `\whiledo`{condición}{ comandos } que ejecuta repetidamente los comandos mientras la condición sea true
- `\@for`\elcomando:={list}\do{comandos}

#### Ejemplo 15.6

En este ejemplo vamos a implementar una circunferencia con un punto (pequeño círculo) cada 20 grados. Para eso usaremos `\@whilenum`





Una manera fácil de dibujar círculos es usar el paquete `tikz`. Un círculo de centro  $(a, b)$  y radio  $r$  cm se dibuja con (el fondo es blanco, así que es como una circunferencia)

```
\begin{tikzpicture}
 \draw (a,b) circle (r cm);
\end{tikzpicture}
```

Para usar las funciones seno y coseno y evaluarlas en ángulos, debemos cargar adicionalmente la librería `calc`.

Los puntos sobre el círculo grande son círculos con centro  $(r \cos t, r \sin t)$ , es decir,

$$({r*\cos(\value{ang})},{r*\sin(\value{ang})})$$

donde `ang` es un contador que va variando de 20 en 20.

@ es tratada como una letra “normal” solamente en paquetes y clases y se usa normalmente para proteger nombres reservados por el propietario en el paquete o la clase. Fuera de estos ambientes, si aparece @, debemos usar el entorno

```
\makeatletter...\makeatother
```

para “hacer @ una letra normal” (make at letter)

El código de la figura anterior es

```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{calc}

\begin{document}

\begin{tikzpicture}
 \makeatletter
 \newcounter{ang} %0
 \draw (0,0) circle (3cm);
 \@whilenum\value{ang} <360 \do{

 \draw[red,fill=red] ({3*cos(\value{ang})},{3*sin(\value{ang})})
 circle (0.1cm);

 \addtocounter{ang}{20}
 }
 \makeatother
\end{tikzpicture}
\end{document}
```

```
\end{tikzpicture}

\end{document}
```

### Ejemplo 15.7 (Tabla de multiplicación módulo $m$ )

En el ejemplo 15.2 vimos como implementar la función  $\text{nmodm}\{n\}\{m\}$  que calcula el residuo  $n \bmod m$ . Usamos la fórmula

$$r = n - (n/m) * m$$

La multiplicación módulo  $m$  de  $a$  por  $b$  es  $(a * b) \bmod m$ . Entonces para hacer una tabla de multiplicar, solo necesitamos calcular

$$\text{nmodm}\{i * j\}\{m\} \text{ con } i = 1, 2, \dots, m - 1 \text{ y } j = 1, 2, \dots, m - 1$$

Pr ejemplo, la tabla de multiplicar  $\bmod 5$ : El código:

```
\documentclass{article}
\usepackage{ifthen}

\begin{document}
\newcount\residuo
\newcommand{\nmodm}[2]{\residuo=#1
\divide\residuo by #2
\multiply\residuo by #2
\multiply\residuo by -1
\advance\residuo by #1
$\!\!\!\!\$the\residuo}

\def\yy{&}
\newcounter{k}
\begin{tabular}{l|llll}
* & 1 & 2 & 3 & 4\\\hline
\setcounter{k}{1}
\whiledo{\value{k} < 5}{
\thek & \multiply\value{k} by 1\relax\nmodm{\value{k}}{5}
\yy
\multiply\value{k} by 2\relax\nmodm{\value{k}}{5}
\yy
\multiply\value{k} by 3\relax\nmodm{\value{k}}{5}
\yy
\multiply\value{k} by 4\relax\nmodm{\value{k}}{5}\%
\ifnum\value{k}=4\hline\end{tabular}\fi
\stepcounter{k}
}%
\end{document}
```

produce:

| * | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 4 | 1 | 3 |
| 3 | 3 | 1 | 4 | 2 |
| 4 | 4 | 3 | 2 | 1 |

Observe: En el entorno `tabular` el valor de `k` es local (`\value{k}`), pero `\stepcounter{k}` cambia el valor global de `k` (los valores en la primera columna)

## 15.4 Macros

### Tokens

**TeX** lee el documento que hemos editado y todo lo convierte a “tokens”. Los tokens son caracteres individuales y comandos (“control sequences”) tales como `\abc`, `\hskip`, etc. y como `\$`, `\!`, etc. **TeX** convierte los tokens en números enteros. Una descripción más específica y amigable se puede ver en la [documentación de Overleaf](#)

### Macros

Una macro es un comando (“control sequence”) definido con `\def`, `\gdef`, `\edef` o `\xdef`. En **LaTeX** usamos `\newcommand`. Las macros soportan hasta 9 argumentos.

|                    |                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\def</code>  | Define una macro                                                                                                                          |
| <code>\edef</code> | Define una macro de tal manera que los comandos (“control sequences”) en el texto de sustitución se expanden cuando se hace la definición |
| <code>\gdef</code> | Equivalente a <code>\global\def</code>                                                                                                    |
| <code>\xdef</code> | Equivalente a <code>\global\edef</code>                                                                                                   |

La estructura general de una macro es

```
< macro primitiva><macro nombre><parámetros>{ <texto para reemplazar>}
```

- **macro primitiva:** Son `\def`, `\gdef`, `\edef` o `\xdef`
- **macro nombre:** Nombre de la macro, como `mimacro`. Solo admite letras. Podríamos usar `\csname ... \endcsname` para usar otros caracteres en el nombre, entre otras cosas. Ver más adelante.
- **parámetros:** Puede omitirse, sino, contiene `#1`, `#2`, ..., `#9` y talvez otros tokens
- **texto para reemplazar:** Es el cuerpo de la macro, es una serie de tokens en las que se “inyectan” los argumentos cuando se procesa (expande) la macro. Los argumentos se introducen en las posiciones indicadas por los parámetros de la macro utilizados en la definición original.

### Ejemplo 15.8

Definamos `\mimacro`

```
\def\mimacro A#1B{"Hola #1"}
```

Aquí solo está el parámetro `#1` y `A` y `B` son delimitadores. Usaríamos la macro como

`\mimacro AJuanB` que produce “Hola Juan”.

Por supuesto, podríamos poner

```
\def\mimacro#1#2{"#1, #2"}
```

y ejecutar con `\mimacro{Hola}{Juan}`.

## `\expandafter`

`\expandafter` es una instrucción que invierte el orden de la “expansión”. El término “expansión” significa *reemplazar* la macro y sus argumentos por el “texto para reemplazar” (si hubiera).

### Ejemplo 15.9

Si tenemos `\def\mimacro{ABC}`, entonces el “texto para reemplazar” es `ABC` y la expansión de `\mimacro` es `ABC`

Un poco más formalmente, el comando `\expandafter` va seguido de varios tokens,

```
\expandafter <tokena> <token1> ... <tokenk>@
```

al ejecutar, `tokena` se guarda sin expansión. Luego se analiza `token1`, si es una macro, *se expande* (la macro y sus argumentos se reemplaza por el “texto de reemplazo”). Si `token1` es una primitiva no pasa nada, excepto que sea una de las primitivas `\expandafter`, `\csname` ... `\endcsname`, `\the` o llave que abre. Luego `tokena` se pega de nuevo delante de los tokens generados en los pasos anteriores, y el procesamiento continúa con `tokena`

### Ejemplo 15.10

```
\def\mimacroB[#1]{"Hola #1"}
\def\nombre{[Juan]}
```

La llamada `\mimacroB\nombre` *no funciona* porque `\mimacro` espera su primer argumento entre `[ ]` y eso pasa solo si expandimos `\nombre`. Tendremos un error “Use the `\mimacroB` doesn’t match its definition”

Entonces debemos primero expandir `\nombre`, ponemos

```
\expandafter\mimacroB\nombre % expande 1ro \nombre a ["Juan"]
```

### Ejemplo 15.11

Consideremos el código

```
\def\las{s}
\expandafter\uppercase{\las} % Resultado: s en minúscula!
```

La idea parece ser primero expandir `\las` de tal manera que quede `\uppercase{s}`, pero en `\expandafter\uppercase{\las}`, nos brincamos `\uppercase` y lo que sigue es `{`, que no es expandible, entonces no pasa nada, queda solo `\uppercase{\las}`. Pero `\uppercase` es una primitiva que espera un “texto general” y entonces no afecta a `\las`.

Para expandir primero `\las` debemos escribir

```
\def\las{s}
\uppercase\expandafter{\las} % Resultado: S
```

## `\csname ... \endcsname`

`\csname ... \endcsname` son primitivas (**TeX**) que nos permite, entre otras cosas, definir nombres de macros y correr estas macros (“`csname=control sequence name`”). Permite construir comandos cuyos nombres contienen por ejemplo, puntos o dos puntos o números y también comandos que se expanden cuando se define o se usa el comando. Ambos son útiles si queremos construir un nombre de comando a partir de varios datos, o crear comando sobre la marcha.

- Podemos utilizar `\csname` para averiguar si un comando ha sido definido y hacer algo. Usamos la instrucción

```
\expandafter\ifx\csname algún comando \endcsname\relax<haga esto>\else<haga
aquello>\fi
```

Esta instrucción salta el `\ifx` y si el comando no ha sido definido, se obtiene `\relax`.

- A diferencia de los comandos hechos con `\def`, los comandos pueden ser definidos con `\csname ... \endcsname` de tal manera que incluyan cosas que no sean letras (con la excepción de `%`). Podemos por ejemplo definir un comando de nombre “`di@- -#&ga`”:

```
\documentclass{article}
\expandafter\def\csname di@- -#&ga\endcsname{Digo @- -\#\&ga }

\begin{document}
% llamada a la macro "di@- -#&ga"
```

```
\csname di@- -#&ga\endcsname % Produce: Digo @- -#&ga
\end{document}
```

- Podemos llamar datos por prefijo. Por ejemplo si los datos de una persona son nombre, celular y email y si queremos usar estos nombres como prefijos entonces tenemos que definir estos nombres nombre, celular, email de la siguiente manera

```
\documentclass{article}
\newcommand\datoUsuario[4]{%#1 es el usuario
 \expandafter\newcommand\csname nombre#1\endcsname{{\bf #2}}
 \expandafter\newcommand\csname celular#1\endcsname{{\bf #3}}
 \expandafter\newcommand\csname email#1\endcsname{{\bf #4}}
}
\begin{document}

\datoUsuario{JP}{Don Juan}{25555555}{DonJuan@prueba.com}

\begin{enumerate}
\item Nombre: \nombreJP
\item Número de celular: \celularJP
\item Correo: \emailJP
\end{enumerate}
\end{document}
```

Esto produce

- Nombre: **Don Juan**
- Número de celular: **25555555**
- Correo: **DonJuan@prueba.com**

- Con `\csname... \endcsname` podemos crear comandos del tipo `\foo#1`, `\foo#2`, .... (como `\label{ }`, contadores, etc).

Por ejemplo, creamos un comando para escribir los datos un usuario por número de cédula. Algo como `\Usuario{10505683}`

Con `\newcommand` y con `\def` no podemos poner números en el nombre de un comando, pero si lo podemos hacer con `\csname... \endcsname`

El nombre del usuario lo podemos poner como un número usando

```
\expandafter\newcommand\csname user-#1\endcsname
```

Y luego llamamos la macro que define el usuario con `\csname user-#1\endcsname`

En este ejemplo vamos a usar un `\newcommand` para crear un comando que haga la llamada `\csname user-#1\endcsname`

```
\documentclass{article}
\newcommand{\DefUsuario}[4]{%
 \expandafter\newcommand\csname user-#1\endcsname{
 \begin{enumerate}
 \item Nombre: #2
 \item Número de celular: #3
 \item Correo: #4
 \end{enumerate}
 }
}
\newcommand{\Usuario}[1]{\csname user-#1\endcsname}

\begin{document}
%Base de datos
\DefUsuario{10505683}{Don Juan Pérez Gamboa}{85555555}{DonJuan@prueba.com}
...
%Llamada
\Usuario{10505683}
\end{document}
```

El resultado es

- a) Nombre: Don Juan Pérez Gamboa
  - b) Número de celular: 85555555
  - c) Correo: DonJuan@prueba.com
- Podemos hacer un comando `\midim{nombredim}{valor}` para definir una dimensión. Es comando `\midim` usa `\csname` para poder definir cada nombre `nombredim` de la dimensión y cada valor.

```
\newcommand{\midim}[2]{%
 \expandafter\newlength\csname #1\endcsname %nombre
 \expandafter\setlength\csname #1\endcsname{#2}%valor
}

%Define dimensión "rlargo"
\midim{rlargo}{5cm}
%aplicación
\textcolor{magenta}{\rule{\rlargo}{1pt}}
```

Produce

---

- Si usamos el paquete `answers` para listas de ejercicios con respuesta, en cada capítulo debemos abrir y cerrar un archivo que contiene las soluciones de las listas ejercicios del capítulo. Esto se hace así (ver el artículo "Personalizar un entorno para listas de ejercicios con el paquete `answers`")

```

\documentclass{book}
\Newassociation{solu}{Soln}{ans}
...
\chapter{A}
...
%Abrir archivo ans1 para las respuestas
\Opensolutionfile{ans}[ans1]

Secciones, listas de ejercicios, secciones,...
...
\Closesolutionfile{ans} %Cerrar ans1

\chapter{B}
...

Abrir archivos con las respuestas
\input{ans1}\input{ans2}...
\end{document}

```

Podemos automatizar este proceso, redefiniendo el estilo book de tal manera que cada vez que se abra un capítulo, se abra y se cierre la lista de ejercicios automáticamente.

En este caso usamos `\csname... \endcsname` para verificar si están presentes los comandos `\Closesolutionfile` y `\Opensolutionfile`, sino, procedemos a abrir y cerrar.

```

...
\newcount\ansj % contador de listas ans j
\Newassociation{solu}{Soln}{ans}
\ansj=\thechapter
...
% Infiltrar chapter
\renewcommand*\chapter{%
\expandafter\ifx\csname Closesolutionfile\endcsname \relax
\else\Closesolutionfile{ans}\fi
\expandafter\ifx\csname Opensolutionfile\endcsname \relax
\else\Opensolutionfile{ans}[ans\number\ansj]\advance\ansj by 1\fi
...
}

```

Recuerde que el código completo está en el artículo "Personalizar un entorno para listas de ejercicios con el paquete answers"





## Bibliografía

- [1] S. Abbot. *“Understanding Analysis”*. Springer. 2000.
- [2] Andrew Mertz y William Slough. *“Beamer by Example”*. En <http://www.tug.org/pracjourn/2005-4/mertz/mertz.pdf>
- [3] A. Hendrickson. *“The Joys of \csname...\endcsname”*. TUGboat, Volume 33 (2012), No. 2.
- [4] J. Bezos. *“Tipografía”*. <http://www.tex-tipografia.com/archive/tipos.pdf>
- [5] C. Feuersanger. *“Notes On Programming in TEX”*. 2021. <http://pgfplots.sourceforge.net/TeX-programming-notes.pdf>
- [6] N. Drakos, R. Moore. *“The LaTeX2HTML Translator”*. En <http://cbl.leeds.ac.uk/>
- [7] F. Hernández. *“Elementos de Infografía para la Enseñanza Matemática”*. Revista digital Matemática, Educación e Internet ([www.cidse.itcr.ac.cr/revistamate/](http://www.cidse.itcr.ac.cr/revistamate/)). Vol. 11, No 1. Agosto-Diciembre 2010.
- [8] Gilles Bertrand. *“Preparing a presentation (Beamer)”*. En <http://www.rennes.enst-bretagne.fr/~gbertran/>
- [9] G. Grätzer. *The New Standard L<sup>A</sup>T<sub>E</sub>X*. Personal T<sub>E</sub>XInc. California. 1998.
- [10] I. Strizver. *“Type rules! : the designer’s guide to professional typography”*. John Wiley & Sons, Second edition. 2006.
- [11] Keith Reckdahl. *“Using Imported Graphics in L<sup>A</sup>T<sub>E</sub>X2”*. <ftp://ftp.tex.ac.uk/tex-archive/info/epslatex.pdf>
- [12] KiJoo Kim. *“Beamer v3.0 Guide”*. En <http://faq.ktug.or.kr/wiki/uploads/>
- [13] Hahn, J. *L<sup>A</sup>T<sub>E</sub>X for everyone*. Prentice Hall, New Jersey, 1993.
- [14] L. Lamport. *“L<sup>A</sup>T<sub>E</sub>X”*. Addison-Wesley. 1996.
- [15] M. Goossens; F. Mittelbach; A. Samarin. *“The L<sup>A</sup>T<sub>E</sub>X Companion”*. Addison-Wesley. 1993.
- [16] R. Willians. *“Non-Designer’s Design Books”*. Third Edition. Peachpit Press. 2008.
- [17] L. Seidel. *“LaTeXtoHTML”*. En <http://apolo.us.es/CervanTeX/>
- [18] Till Tantau. *User Guide to the Beamer Class, Version 3.07* En <http://latex-Beamer.sourceforge.net>, 2007.

- [19] The LaTeX Font Catalogue. En <http://www.tug.dk/FontCatalogue/utopia-md/>
- [20] Hahn, J. “ $\text{\LaTeX}$  for everyone”. Prentice Hall, New Jersey, 1993.
- [21] V. Eijkhout. “*TeX by Topic: A Texnician’s Reference*”. Addison-Wesley. 1992.
- [22] “TeXText”. <http://pav.iki.fi/software/texttext/>. Consultada el 3 de Enero 2013.
- [23] Joseph Wright. “From `\newcommand` to `\NewDocumentCommand` with xparse”. <https://www.tug.org/TUGboat/tb31-3/tb99wright.pdf>. Consultada el 10 de Enero 2014.
- [24] TeX Stack Exchange. “How to place a shaded box around a section label and name”. <http://tex.stackexchange.com/questions/34288/>. Consultada el 23 de Mayo 2014.
- [25] TeX Stack Exchange. “How to make section like this”. <http://tex.stackexchange.com/questions/147344/>. Consultada el 23 de Mayo 2014.
- [26] TeX Stack Exchange. “Customizing chapter style with tikz”. <http://tex.stackexchange.com/questions/160320/>.
- [27] TeX Stack Exchange. “How I do I make enumerated lists in columns”. <http://tex.stackexchange.com/questions/303912/>
- [28] TeX Stack Exchange. “Pretty Table of Contents”. <http://tex.stackexchange.com/questions/35825/>. Consultada el 11 de junio 2014.
- [29] TeX Stack Exchange. “How to customize the table of contents using TikZ?”. <http://tex.stackexchange.com/questions/19796/>. Consultada el 11 de junio 2014.
- [30] TeX Stack Exchange. “Alternatives to the horizontal line in fancyhdr”. <http://tex.stackexchange.com/questions/113937>. Consultada el 11 de junio 2014.
- [31] TeX Stack Exchange. “Problem with babel and tikz using draw”. <http://tex.stackexchange.com/questions/166772/problem-with-babel-and-tikz-using-draw>. Consultada en abril, 2017.

- \, 18
- \\, 21
- ¿, 12
- #, 18
- \$, 18
- %, 18
- &, 18
- á, 12
- é, 12
- í, 12
- ó, 12
- ú, 12
- TeX, 254
- ^, 18
- \_, 18
- LaTeX, 8, 12, 13, 15
- PDFLaTeX, 8, 9
- ©, 48
- ', 48
- +, 93
- \$\$, 3, 8, 8, 32, 52, 61, 62, 64, 64, 69, 134
- \$, 3, 8, 8, 10, 15, 32, 34, 52, 53, 56, 56, 57, 62–64, 134, 164, 235–240, 244, 245, 247, 248
- &, 66, 67, 69–74, 247
- &=, 73, 74
- \, 64, 75
- \,, 64
- \:, 64
- \;, 64
- \[, 61, 63, 64, 66, 67
- \\, 66, 67, 69–74
- \], 61, 63, 64, 66, 67
- \{...\}, 62
- ^, 3, 15, 57, 164, 217, 219, 225
- \_, 57, 57, 59–61, 63, 65, 70, 78, 91, 240
- LaTeX, 162
- ñ, 12
- ~, 18
- PDF
  - Archivo, 137
- a0, 256
- a0poster, 254
- a1, 256
- a2, 256
- a3, 256
- a4, 256
- \above, 59
- \abstract, 44
- Acentos, 12
  - Modo matemático, 63
- \acute, 63
- \addcontentsline, 44
- AddToBackground\*\AddToBackground\*, 264, 265
- AddToBackground\AddToBackground, 264
- \addtolength, 205
- Adobe Courier, 50
- Adobe Helvetica, 50
- Adobe Illustrator, 137
- Adobe Times, 50
- \aleph, 75
- \alert, 241, 242
- Alineamiento, 70, 149
- Alineamiento horizontal, 100
- \alph, 37
- \alpha, 74, 244
- \amalg, 75
- Amenidad, 149, 151
- amsmath
  - Delimitadores, 61
- \angle, 75
- \appendix, 210
- \approx, 75
- \approxeq, 76
- \arabic, 37, 41
- \arcsen, 56

- `\arrayrulecolor`, 84
- Arreglos, **66**
- article, 13
- article.cls, 14
- `\ast`, **75**
- `\asymp`, **75**
- `\atop`, **58, 59**, 60, 62
- `\author`, **43**, 43, 233
- author\author, 255, 257, 263
- 
- `\backsim`, **76**
- `\backsimeq`, **76**
- `\backslash`, **75**
- `\bar`, **63**
- Barras horizontales, **62**
- Beamer, 1, 163, **232**, 233, 234, 239, **253**, 253, 254, 258
  - Alert, 237
  - algorithm2e, 244
  - Animaciones, **249**
  - Blocks, 240
  - Botones, **246**
  - Código de programas, 241
  - Color, 247
  - Efectos de transición, 247
  - Entornos, 239
  - Figuras, 233, 245
  - fragile, 241, 244
  - frame, **232**, 244
  - Gráficos, 245
  - Ligas, **246**, 246
  - Ligas documentos externos, 248
  - Marcos(frames), 235
  - minted, **243**
  - Multicolumnas, 249
  - Pause, 238
  - semiverbatim, 241
  - Temas, 234
  - Transparencias, 246
  - Velos(overlays), 235
- beamer, 254, 256–261
  - bloques, 257
  - multicolumnas, 257
  - navigation symbols, 261
- `\beamergetobutton`, **246**
- beamerposter, 253, 254, 256, 257
  - escale, 256
  - orientation, 256
  - size, 256
- `\beamerreturnbutton`, **246**
- 
- beamertemplateshadingbackground\beamertemplateshadingbackground, **261**
- `\because`, **76**
- `\begin`
  - algorithm, 217, 219, 244
  - align, **73**
  - align\*, **73, 74**
  - array, 61, **66**, 66, 67, 69
  - block, **240**
  - Bmatrix, 70
  - bmatrix, 70
  - center, **21**, 21, 22, 24, 32, 78, 99, 107, 247
  - column\*, 250
  - columns, **250**
  - document, 3, 8, 11, 15, 20, 43, 45, 48, 49, 52, 56, 76, 107, 116, 118, 205, 210, 224, 226, 233
  - ejemplo, 239
  - enumerate, 34–37, 41, 79, 236–238
  - eqnarray, **71**, 71–73
  - eqnarray\*, **71**, 71, 72, 240
  - equation, **65**, 65
  - ex, **210**
  - figure, 24, **88**, 88, 89, 118, 119, 134, 245, 249
  - floatingfigure, **121**
  - flushright, 206
  - frame, **233**, **235–241**, 242–247, 249, **250**
  - itemize, 35, 36
  - longtable, 85, 87
  - minipage, 23, 24, 32, 79, 80, 89, 134
  - minted, **224**, **225**, 243
  - multicols, **23**, 23
  - multline, **73**
  - multline\*, **74**
  - pgfonlayer, 107
  - picture, 206
  - pmatrix, 70
  - preface, 44
  - pyglist, **225**, **226**
  - scriptsize, 206
  - semiverbatim, **242**
  - sideways, 92, 93
  - smallmatrix, 70
  - sol, **210**
  - subequation, **65**
  - table, **88**, 89, 91, 92, 94, 99, 103, 118
  - tabular, 79, 83, 84, 89–95, 98, 99, 102, 103, 116, 118, 247

- tabularx, 107
- teo, 134
- teorema, 239
- thebibliography, 43, 44, **139**
- tikzpicture, 107, **163**, **164**
- Verbatim, 53
- verbatim, 29, 241
- Vmatrix, 70
- vmatrix, 70
- wrapfigure, **119**
- begin\begin
  - block, 255, 260
  - column, 255, 260
  - columns, 255, 260
  - document, 255, 260, 263
  - frame, 255, 257, 260
  - itemize, 255
- \beta, **74**
- \bf, **19**, 21, 34–36, 41
- BIB
  - Archivo, 140
- \bibitem, **139**, 139
- \sbibitem, 43, 44
- Bibliografía, **42**, 44, **139**, 140
- \bibliography, **140**, **141**
- \bibliographystyle, **140**, **141**
  - abbrv, **140**
  - alpha, **140**
  - apalike, **140**
  - plain, **140**
  - unsrt, **140**
- BibTeX, 44, **139**, **140**, 140, 141, 143
  - APA, **141**
  - Base de datos, 140, 143
  - Propiedades, **141**
  - Tipos, **140**
- \bigcirc, **75**
- \Bigl, **61**
- \biggl, **61**
- \Biggr, **61**
- \biggr, **61**
- \Bigl, **61**
- \bigl, **61**
- \Bigr, **61**
- \bigr, **61**
- \bigtriangledown, **75**
- \bigtriangleup, **75**
- \binom, 32, 240
- Bloc de notas, 47
- \blue, 51, 78
- Bold, 50
- Bold extended, 50
- \boldmath, **63**
- boldmath\boldmath, 255, 257
- \boldsymbol, 59
- book.cls, 14
- paquete booktabs, 98
- \bot, **75**
- \bottomrule, 84
- \bottonrule, 85, 87, 95
- BoundingBox
  - Errores, 136
- \bowtie, **75**
- Brochures, 253, 258
- brochures, 261
- \bullet, 37, **75**
- C++, 219
- Código cromático, 149
- Cabeceras, 204
- Cajas, 22
- Cajas de color, 30
- \cal, 34
- \cup, 34
- \cap, 34, 62, **75**
- Caps and small caps, 50
- \caption, 24, 88, **89**, 89, 91, 92, 99, 103, 134
- \captionof{figure}, 118, 119, 121, 217, 219, 245
- Caracteres especiales, 18
- Carga cognitiva, **147**
- Carga cognitiva extrínseca, **147**, 147
- Carga cognitiva intrínseca, **147**
- \catcode, 53
- \cdot, 33, 59, **75**, 78
- \cdots, 59, **60**, 70
- \cellcolor, 92, 93
- \centering, 79, 91, 92, 94, 99, 103, 119, 245, 249
- Centrado
  - Modo matemático, **64**
- Centrar texto, 21
- \chapter, 43, **44**, 205
- \char, 49
- \chi, **74**
- \choose, 59
- \circ, **75**
- circ\circ, 263
- \circeq, **76**
- Citas bibliográficas, **139**

`\cite`, **44**, **139**, **140**, 141  
`\cline`, **83**, 83  
`\Closesolutionfile`, **210**  
`\clubsuit`, **75**  
`\cmidrule`, 95  
CMYK, 30, **153**  
Color, 30, **152**, 153  
`\color`, 32, 41, 53, 107  
    LightSteelBlue1, 90  
    red, 78, 79  
    RoyalBlue1, 90  
    white, 41  
Color Schemer Designer, 153  
Color Schemer Studio, 153  
Columnas, 22  
`\columnsep`, 120  
Comillas, 12  
Compile, **8**, 8, 10  
    LaTeX , 8, **9**  
    PDFLaTeX , 8, **9**, 9  
    LaTeX , 9  
    TeXMaker, 9  
    xelatex, 52  
Computer Modern Math Extensions, 50  
Computer Modern Math Italic, 50  
Computer Modern Math Symbols, 50  
Computer Modern Roman, 50  
Computer Modern Sans, 50  
Computer Modern Typewriter, 50  
Condensed, 50  
`\cong`, **75**  
`\const`, 107  
Contadores automáticos, 65  
    Ecuaciones, 65  
    Teoremas, 65  
Contenido, 42  
Contraste, 150  
`\coord`, 80  
`\cos`, 57, 64, 69  
Cuerpo del documento, 8, **15**, 15  
`\cup`, 62, **75**  
`\curlyeqprec`, **76**  
`\curlyeqsucc`, **76**  
CutLine\*\CutLine\*, 263, 264  
CutLine\CutLine, 263, 264  
`\cyan`, 33  
  
`\dagger`, **75**  
`\dashv`, **75**  
`\date`, **43**, 43, 233  
  
date\date, 255, 257, 263  
`\ddagger`, **75**  
`\ddots`, **60**, 70  
\DeclareGraphicsExtensions, 116, 234, 245  
`\def`  
    `\max`, 56  
    `\min`, 56  
`\definecolor`, 30, **226**, **247**  
definecolor\definecolor, 264  
Definiciones, 44  
Delimitadores, 60  
`\Delta`, 34, **74**, 240  
`\delta`, **74**  
`\dfrac`, **58**  
`\diamond`, **75**  
`\diamondsuit`, **75**  
Diseño editorial, **147**, 147  
`\displaystyle`, **57**, 57, 59, 60, 62, 64, 78,  
    95, 98, 103, 225  
Distribuciones, **3**, 3  
`\div`, **75**  
`\documentclass`, 2, 147  
    article, 8, 11, **15**, 15, 20, 49, 52, 56, 107,  
        116, 118, 224, 226  
    beamer, 232, **233**, 247  
    book, 2, 3, 45, 205, 210  
    memoir, 49  
    report, 43, 48  
documentclass\documentclass  
    beamer, **255**, **260**  
    leaflet, **263**  
`\doteq`, **75**  
`\doteqdot`, **76**  
`\dotfill`, **28**, 28  
`\dots`, 240  
`\down`, 107  
`\downarrow`, 107  
`\draw`, 107, 164  
DVI  
    Archivo, **9**, 9, 116  
    Visor, 8  
    Visualizador, 116  
DVI→PDF, 163  
  
Editar, 8  
Editor de textos, 47  
Editores, **3**, 3  
Efectos especiales, 20  
`\eIf`, **216**, 217  
Ejemplos, 44

- `\ell`, **75**
- `\em`, **19**, 35, 36
- `\emptyset`, 34, **75**
- `\end`
  - algorithm, 217, 219, 244
  - align, **73**
  - align\*, **73**, **74**
  - array, 61, **66**, 66, 67, 69
  - block, **240**
  - Bmatrix, 70
  - bmatrix, 70
  - center, **21**, 21, 22, 24, 32, 78, 99, 107, 247
  - column\*, 250
  - columns, **250**
  - document, 3, 8, 11, 15, 20, 43, 45, 48, 52, 56, 76, 107, 116, 118, 205, 210, 226, 233
  - ejemplo, 239
  - enumerate, 34–37, 41, 79, 236–238
  - eqnarray, **71**, 71–73
  - eqnarray\*, **71**, 71, 72, 240
  - equation, **65**, 65
  - ex, **210**
  - figure, 24, **88**, 88, 89, 118, 119, 134, 245, 249
  - floatingfigure, **121**
  - flushright, 206
  - frame, **233**, **235–241**, 242–247, 249, **250**
  - itemize, 35, 36
  - longtable, 85, 87
  - minipage, 23, 24, 32, 79, 80, 89, 134
  - minted, **224**, **225**, 243
  - multicols, **23**, 23
  - multline, **73**
  - multline\*, **74**
  - pgfonlayer, 107
  - picture, 206
  - pmatrix, 70
  - preface, 44
  - pyglist, **225**, **226**
  - scriptsize, 206
  - semiverbatim, **242**
  - sideways, 92, 93
  - smallmatrix, 70
  - sol, **210**
  - subequation, **65**
  - table, **88**, 89, 91, 92, 94, 99, 103, 118
  - tabular, 79, 83, 84, 89–95, 98, 99, 102, 103, 116, 118, 247
  - tabularx, 107
  - teo, 134
  - teorema, 239
  - thebibliography, 43, 44, **139**
  - tikzpicture, 107, **163**, **164**
  - Verbatim, 53
  - verbatim, 29, 241
  - Vmatrix, 70
  - vmatrix, 70
  - wrapfigure, **119**
- `end\end`
  - block, 255, 260
  - column, 255, 260
  - columns, 255, 260
  - document, 255, 260, 263
  - frame, 255, 257, 260
  - itemize, 255
- Enfático, **19**
- `\ensuremath`, 107
- Enumerado, 33
  - Tikz, 40
- enumerate, **33**, 36, 40, 44
- EPS
  - Archivo, 110, 116, 137
  - Convertir, 116
- `\epsilon`, **74**
- eqnarray, 45
- `\eqslantgtr`, **76**
- `\eqslantless`, **76**
- equation, 45
- `\equiv`, **75**
- Error de sintaxis, **10**
- Español, 13
- Espacio
  - Modo matemático, **64**
- Espacio horizontal, 21
- Espacio vertical, 21
- Esquemas de color, 153
  - Acromático, **153**
  - Análogo, **153**
  - Complementario, **153**
  - Complementario dividido, **153**
- `\eta`, **74**
- `\exists`, **75**, 134
- Expresiones de dos niveles, 58
- `\setlength\extrarowheight`, 98
- `\fallingdotseq`, **76**
- `\fancyhead`, 205
- `\fancyhf`, 205



- \fancypagestyle, **205**
- \fbox, **22**, 22, 35, 36
- \fboxsep, 32
- \fcolorbox, 32
- Figuras, 1, 110
  - Insertar, 110
- figure, 45
- \fill
  - orange, 163
- \flat, **75**
- floatflt, **119**, **120**, 121
- \fontencoding, **49**, 50, 51
- \fontfamily, **49**, 50, 51, 79, 90–92, 205
  - cmex, 50
  - cmm, 50
  - cmr, 50
  - cmss, 50
  - cmsy, 50
  - cmtt, 50
  - pcr, 50
  - phv, 50
  - ptm, 50
- Fonts, 18
- \fontseries, **50**, 50, 51
  - b, 50
  - bx, 50
  - c, 50
  - m, 50
  - sb, 50
- \fontshape, **50**
  - it, 50
  - n, 50
  - sc, 50
  - sl, 50
- \fontsize, 20, **50**, 79, 205
- \footnote, 29, 53
- \footnotesize, **19**, 41
- footnotesize\footnotesize, 255
- \For, **216**, 219
- \forall, **75**
- Formatos, 110
- \frac, 10, **58**, **59**, 69, 73, 78, 95, 98, 103, 219
- Fracciones, 58
- \frame, 233
- frame\frame, 255
- \framebox, **22**, 27
- \framesubtitle, **235**, **236**
- \frametitle, **235**, **236**, 241, 242, 249
- \frown, **75**
- Fuentes, 8, **18**, 18
  - Arial, **151**
  - Atributos, 49
  - Bookman, **152**
  - Como escoger, 151
  - Computer Modern, **151**
  - Decorativa, **151**
  - encoding, 49
  - family, 49
  - Helvética, 204
  - Helvetica, **151**
  - New Century Schoolbook, **152**
  - Palatino, **151**
  - Sans serif, **151**
  - series, 49
  - Serif, **151**
  - shape, 49
  - size, 49
  - tabular, **90**
  - Tamaños, **18**, 19, 48
  - Times, **151**, **152**
  - Times New Roman, **151**
  - Tipos, **18**, 18
- \fvset, **225**, **226**
- \Gamma, **74**
- \gamma, **74**
- Geogebra, 116
- geometry, 260
- geometry\geometry, **260**
- \geq, 61, **75**
- \geqq, **76**
- \geqslant, **76**
- \gfrac, 78
- \gg, **75**
- \ggg, **76**
- GIF
  - Archivo, 110, 116
- Gimp, 116
- Gráficos, 1, 8, 110
  - Escalar, 116
- \gtrapprox, **76**
- \gtrdot, **76**
- \gtreqless, **76**
- \gtreqqless, **76**
- \gtrless, **76**
- \gtrsim, **76**
- \hat, **63**
- \hbar, **75**
- \hbox, 63

- `\headheight`, 16, 205
- `\headsep`, 16
- `\heartsuit`, 75
- `\helv`, 205
- `\hfill`, 23, 28, 28, 29, 79, 80, 134
- `\hline`, 83, 83, 89–95, 98, 99, 103, 247
- `\href`, 248, 249
- `\hrulefill`, 16, 28, 28
- `\hspace`, 21, 21, 22, 33, 64, 102
- `\hspace*`, 21, 33
- `\Huge`, 19
- `\huge`, 19
- `huge\huge`, 257
- `\hyperlink`, 246
- `\hypertarget`, 246
  
- Idioma, 12
- `\If`, 216, 219
- `\iiint`, 59
- `\iint`, 59
- `\Im`, 75
- Imagen
  - Resolución, 137
- `\imagetop`, 102, 103
- `\imath`, 63, 75
- `\in`, 62, 64, 69, 75, 134, 217, 219, 235, 239
- `\includegraphics`, 24, 27, 79, 99, 101–103, 111, 114, 116, 116, 118, 119, 121, 134, 206, 245, 248, 249
  - angle, 114
  - clip, 111
  - graphicx, 110
  - height, 111, 114
  - scale, 111, 114, 116, 134
  - trim, 111, 114
  - width, 111, 114, 116, 116
- `includegraphics\includegraphics`, 258, 260, 261, 264, 265
- index, 46
- `\index`, 43, 45, 46–47
  - see, 46, 47
  - seealso, 47
  - `textbf`, 46
  - `textit`, 46
- Índice alfabético, 45–47
- `\infty`, 32, 75
- Inglés, 12
- Inkscape, 1, 116, 129, 133, 137, 153, 153
  - Edición de figuras, 127, 132
  - TeXText, 129
- `\input`, 48, 76, 210
- `inputenc`, 49, 261
  - latin1, 49
  - utf8, 12, 52
- `institute\institute`, 255, 257
- `\int`, 57, 59, 60, 64
- `int\int`, 255
- Integrales, 59
- Internet, 153
- `\intertext`, 73
- `\iota`, 74
- `\it`, 19
- Itálica, 19
- Italic, 50
- `\item`, 33, 34–37, 41, 235–238
  - label, 41
- `\itembolazules`, 41
  
- JabRef, 141
- Java, 219, 225
- `\jmath`, 75
- `\Join`, 75
- JPG
  - Archivo, 110, 111, 116
  
- `\kappa`, 74
- Kile, 4, 52
- Kindle, 49
- Knuth, D., 2
- `\KwData`, 217, 244
- `\KwResult`, 217, 244
  
- Líneas, 28
- `\label`, 45, 88, 89, 89, 99, 103, 118, 134, 217, 219, 245
- `\Lambda`, 74
- `\lambda`, 74
- Lamport, L., 2
- `\Large`, 19, 41
- `\large`, 19, 107
- `LARGE\Large`, 255
- `Large\Large`, 255
- `large\large`, 255, 257
- LaTeX, 1, 139, 219, 232, 253
- LaTeX, 56, 64, 65, 110, 116, 119–121
- LaTeX, 18, 20, 23, 43, 46, 48, 49, 52
- LaTeX, 136, 147, 151
- LaTeX
  - Internet, 268
- LaTeX, 2, 2–5
  - Convertir, 5

$\LaTeX$ -dvips, 232–234, 245  
 LaTeXDraw, 110  
 latin1, 261  
`\ldots`, 60, 80  
 Leaflet, 261, 263  
 leaflet, 253, 258  
`\left`, 59, 60, 61, 61, 62, 64, 66, 67, 69, 219  
`\leftmark`, 205  
`\Leftrightarrow`, 78  
 Legibilidad, 151  
 Lenguajes de Programación  
     Código, 219  
`LenToUnit\LenToUnit`, 265  
`\leq`, 64, 75, 219  
`\leqq`, 76  
`\leqslant`, 76  
`\lessapprox`, 76  
`\lessdot`, 76  
`\lesseqgtr`, 76  
`\lesseqqgtr`, 76  
`\lessgtr`, 76  
`\lesssim`, 76  
 Letras griegas, 74  
 Leyes Gestalt, 147, 149  
     Ley de la semejanza, 148  
     Ley de proximidad, 148  
`\lim`, 59, 79  
`\LinesNumbered`, 217, 219  
`\linesnumbered`, 244  
`\linewidth`, 134  
 Linux, 3, 47, 142  
     Editor de textos, 140  
 Listas de ejercicios, 209  
 listings, 219  
`\ll`, 75  
 Llaves, 62  
     Horizontales, 62  
`\lll`, 76  
`\ln`, 61  
`\log`, 52, 65, 72  
`\Longleftrightarrow`, 34  
`\Longrightarrow`, 69, 72, 78, 237, 238  
`\arraystretch`, 98  
`\longtable`, 85  
 Los cuatro principios básicos, 149  
 LyX, 4  
 Márgenes, 8  
 Mac, 3  
 MacTeX, 3  
 Make4ht (Traducción a HTML), 268  
`makebox\makebox`, 265  
`makeindex`, 46  
`\makeindex`, 43, 45  
`\maketitle`, 43, 43  
`maketitle\maketitle`, 258, 263  
`\marginnote`, 33  
`\markboth`, 205  
`\markright`, 16, 205  
`\mathbb`, 76  
 Mathematica, 116  
`\mathtop`, 56  
 MatLab, 116  
 Matrices, 66, 69  
`\arraystretch`, 98  
`\max`, 62  
`\mbox`, 56, 61, 67, 67, 71–73, 78, 217, 219  
 Medium, 50  
`\mid`, 75  
`\midrule`, 84, 85, 87, 95  
 MikTeX, 56  
 MiKTeX, 14  
 MikTeX, 52, 151, 232  
 MikTeX, 3, 5  
`\min`, 62  
 minipage, 32, 120  
 minitoc, 203  
 minted, 219, 225  
     python, 225  
`mode\mode`  
     babel, 255  
`\models`, 75  
 Modo matemático, 56  
 Modulación, 48  
`\mp`, 75  
`\mpage`, 79, 80  
`\mu`, 74  
 multicols, 23  
`\multicolumn`, 94, 95, 103  
  
`\N`, 60, 64, 217  
`\nabla`, 33, 75  
`\natural`, 75  
`\neg`, 75  
 Negrita, 19  
     Modo matemático, 63  
`\neq`, 59, 60, 75, 217  
`\Newassociation`, 210  
`\newcolumnntype`, 102, 103  
`\newcommand`, 76

- \arcsec, 43
- \arcsen, 43, 56
- \bc, 78
- \be, 79
- \bt, 79
- \colr, 79
- \ds, 78
- \ec, 78
- \ee, 79
- \et, 79
- \fhv, 79
- \gfrac, 78
- \helv, 205
- \imp, 78
- \itembolazules, 41
- \limite, 79
- \mpage, 79
- \proy, 78
- \sen, 43, 56
- \sii, 78
- \sumauk, 78
- \wvec, 78
- \wvecb, 78
- \newcommand\*, 107
- \newcommandx\*
  - \coord, 80
  - \mpage, 80
- \newpage, 16, 205
- newpage\newpage, 263
- \newtheorem, 43
  - Corolario, 233
  - Definicion, 233
  - Ejemplo, 233
  - ex, 210
  - Prueba, 233
  - teo, 134
  - Teorema, 233
- \ni, 75
- \nocite, 140
- \node, 41, 107
- \nonumber, 71, 72
- Normal, 50
- \normalfont, 51
- \normalsize, 19, 107
- normalsize\normalsize, 255
- \not
  - <, 75
  - =, 75
  - >, 75
  - \approx, 75
  - \cong, 75
  - \equiv, 75
  - \geq, 75
  - \leq, 75
  - \prec, 75
  - \preceq, 75
  - \sim, 75
  - \simeq, 75
  - \sqsubseteq, 75
  - \sqsupseteq, 75
  - \subset, 75
  - \subseteq, 75
  - \succ, 75
  - \succeq, 75
  - \supset, 75
  - \supseteq, 75
- Notas al pie de página, 29
- notumble, 261
- \nu, 74
- Nuevos comandos, 76
- Numeración, 8
- Objetos flotantes, 83, 88
- \oddsidemargin, 15, 16, 43
- \odot, 75
- \oint, 59
- Okular, 116
- \Omega, 74
- \omega, 74
- \ominus, 75
- OML, 49
- OOoBasic, 54
- \opensolutionfile, 210
- Operadores binarios, 74
- \oplus, 75
- \oslash, 75
- \otimes, 75
- \over, 58, 59, 60, 64, 72, 134
- \overbrace, 63
- Overleaf, 4
- \overline, 34, 62
- \overrightarrow, 76, 78
- \overset, 59
- Página
  - Ancho, 8
  - Largo, 8
- Párrafo, 20
- \pagebreak, 48
- \pagestyle, 205

- empty, **16**
- fancy, **205**
- myheadings, **16**
- pagestyle\pagestyle, **263**
- Panfleto, **263**
- Panfletos, **253, 258, 261**
- paperheight\paperheight, **261, 264, 265**
- paperwidth\paperwidth, **261, 264, 265**
- \par, **99**
- \parallel, **75**
- \parbox, **27, 28, 120, 206, 248**
- \parindent, **16, 43**
- \parskip, **16**
- \partial, **75**
- Paso de línea, **20, 21**
- \pause, **238, 239**
- PDF
  - Archivo, **9, 9, 110, 111, 116, 133, 137, 151, 153, 163**
  - Archivos, **30**
  - Editor, **153**
  - Visor, **8**
- PDFLaTeX, **163**
- \pdfdeclarelayer, **107**
- PDFLaTeX, **110, 111**
- PDFTeX, **232, 233, 256**
- PDFLaTeX, **133, 136**
- PDFLaTeX, **46**
- \pdfsetlayers, **107**
- \perp, **75**
- \Phi, **74**
- \phi, **74**
- \Pi, **74**
- \pi, **64, 69, 74**
- picture, **162**
- \plot, **163**
- \plset, **225, 226**
- \pm, **75**
- \pmb, **59, 64**
- PNG
  - Archivo, **110, 111, 116, 137**
- Posters, **253, 254**
- PostScript, **151**
- Potencias, **57**
- Preámbulo, **8, 11, 13, 15, 15, 16, 33, 45, 48, 50, 76, 79, 90, 92, 114, 118, 120, 134, 152, 163, 216, 225, 244, 247**
- \prec, **75**
- \precapprox, **76**
- \preccurlyeq, **76**
- \preceq, **75**
- \precsim, **76**
- \prime, **75**
- \printindex, **43, 45, 46**
- \prod, **59**
- \propto, **75**
- \protect, **41**
- Proximidad, **149**
- \proy, **78**
- \Psi, **74**
- \psi, **74**
- PStricks, **110**
- \put, **206**
- put\put, **264, 265**
- pyglist
  - java, **225, 226**
- Pygments, **224, 225**
- python, **224**
- QtOctave, **116**
- \quad, **74**
- \R, **34, 56, 64**
- Raíces, **59**
- \raisebox{}{}{} , **100**
- \Re, **34, 75**
- Recortar figuras, **135**
- \red, **134**
- \ref, **45, 89, 118**
- Referencias bibliográficas, **44**
- Relaciones, **75**
  - Negación, **75**
- \renewcommand
  - \abstractname, **13**
  - \appendixname, **13, 14**
  - \baselinestretch, **16**
  - \bibname, **14**
  - \chaptermark, **205**
  - \chaptername, **14**
  - \contentsname, **13, 14**
  - \figurename, **13, 14**
  - \footrulewidth, **205**
  - \headrulewidth, **205**
  - \labelenumi, **37**
  - \labelenumii, **37**
  - \labelenumiii, **37**
  - \labelenumiv, **37**
  - \partname, **13, 14**
  - \refname, **13**
  - \sectionmark, **205**

- `\tablename`, 13, 14
- `\Repeat`, **216**
- Repetición, 150
- report.cls, 14
- resizebox\resizebox, 265
- `\Return`, 217, 219, 244
- RGB, 30, **153**
- `\rhead`, 206
- `\rho`, **74**
- `\right`, 59, **60**, **61**, 61, 62, 64, 66, 67, 69, 219
- `\Rightarrow`, 134
- `\rightarrow`, 59, 79, 83, 89
- `\rightmark`, 205
- `\risingdotseq`, **76**
- `\rm`, **19**, 56, 78, 210, 217, 219
- Roman, 19
- `\Roman`, 37
- rotatebox\rotatebox, 265
- `\rowcolor`, 91, 92, 94, 95
  - LightBlue2, 91
- `\rowcolors`, 91, 95, 107, **247**
- `\rule`, **28**, 29, 98, 103
- rule\rule, 264
- Símbolos, 8
- Símbolos
  - Adicionales, 76
  - amssymb, 76
  - Especiales, 75
  - Otros, 75
- Símbolos matemáticos, **74**
- Sans Serif, 19
- `\sc`, **19**, 43
- `\scalebox`, **103**, 103
- sciposter, 254
- `\scriptsize`, **19**
- scriptsize\scriptsize, 255
- Secciones, **42**
  - Personalizar, 191
- `\section`, 43, **44**, 210
- `\selectfont`, 20, 50, 51, 79, 90–92, 205
- Semi-bold, 50
- `\sen`, 56, 57
- `\setbeamercovered`, **233**
- setbeamertemplate\setbeamertemplate, **260**, **261**
- `\setcounter`, 43, **65**, 65, 210
- `\setlength`, 206
  - `\columnsep`, 23
- `\SetAlgoLined`, 217
- `\SetLine`, 244
- `\setmainfont`, 52
- `\setminus`, **75**
- `\SetVLine`, 244
- `\SetAlgoVlined`, 217, 219
- `\sf`, **19**
- `\sharp`, **75**
- Shutter, **135**
- sideways, 93
- `\Sigma`, **74**
- `\sigma`, **74**
- `\sim`, **75**
- `\simeq`, **75**
- `\sl`, **19**
- Slanted, 19, 50
- `\small`, **19**, 33, 43
- Small Caps, 19
- `\smile`, **75**
- Software libre, 110
- Sombreros
  - Modo matemático, **63**
- `\spadesuit`, **75**
- `\specialrule`, 84
- `\sqcap`, **75**
- `\sqcup`, **75**
- `\sqrt`, 8, **60**, 72, 95, 98, 134, 237, 238
- `\sqsubset`, **76**
- `\sqsubseteq`, **75**
- `\sqsupset`, **76**
- `\sqsupseteq`, **75**
- `\stackrel`, 59
- `\star`, **75**
- Subíndices, 57
- `\subfloat`, 119
- Subrayado, 19
- Subsecciones
  - Personalizar, 191
- `\subsection`, 43, **44**
- `\Subset`, **76**
- `\subset`, 22, **75**
- `\subseteq`, 34, **75**
- `\subseteqq`, **76**
- `\substack`, 59
- `\subsubsection`, 43, **44**
- `\subtitle`, 233
- `\succ`, **75**
- `\succapprox`, **76**
- `\succcurlyeq`, **76**
- `\succeq`, **75**
- `\succsim`, **76**

- `\sum`, 57, 59, 61, 78, 225, 240
- `\sumauk`, 78
- Superíndices, 57
- `\Supset`, 76
- `\supset`, 75
- `\supseteq`, 75
- `\supseteqq`, 76
- `\surd`, 75
- svg
  - Archivo, 110
- Título, 42
- Tablas, 83
  - Ancho de columnas, 98
  - Color, 90
  - Escalar, 103
  - Espaciado, 95
  - Modo matemático, 102
  - Rotar texto, 92
  - Tikz, 105
- table, 45
- `\tableofcontents`, 43, 43
- paquete tabu, 98
- tabular, 247
- `\arraystretch`, 98
- Tamaño Natural, 57
- `\tau`, 74
- Teoremas, 44
- TEX
  - Archivo, 9, 9
- $\TeX$ , 9, 49, 52
- TeX, 2, 2–4
- TeXDraw, 110
- $\TeX$ Live, 151, 225, 232
- TeX Live, 52, 56
- TeX Live, 3, 5
- TeXMaker, 3, 4, 9, 52
  - Opciones, 47
- TeXstudio, 4
- `\textbf`, 19, 99, 107
- `\textbf\textbf`, 265
- `\textcolor`, 107
- `\textcolor\textcolor`, 264, 265
- textcomp, 54
- `\textheight`, 15, 15, 43, 48, 56
- `\textit`, 19, 99
- Texto matemático, 8, 18, 219
- Texto normal, 18
- `\textquotesingle`, 54
- `\textsf`, 206
- `\textsf\textsf`, 265
- `\texttt`, 19
- `\textwidth`, 15, 15, 23, 24, 43, 48, 56, 79, 80, 107, 114
- `\textwidth\textwidth`, 260
- `\tfrac`, 58
- `\thepage`, 205
- `\therefore`, 76
- `\thesection`, 205
- `\Theta`, 74
- `\theta`, 74
- `\thickapprox`, 76
- `\thicksim`, 76
- `thispagestyle\thispagestyle`, 263
- Tikz, 1, 40, 41, 106, 110, 162, 163
  - Figuras nativas, 162
  - Tablas, 105
- `\tikz`, 41, 163
- tikzpicture, 162
- `\times`, 53, 75, 91–93
- `\tiny`, 19
- tiny\tiny, 255
- `\title`, 43, 43, 233
- title\title, 255, 257, 263, 264
- `\titlepage`, 233
- todonotes, 33
- `\top`, 75
- `\topmargin`, 15, 16, 43, 48, 56
- `\toprule`, 84, 85, 87, 95
- Trípticos, 253, 258, 261
- `\transblindshorizontal`, 247
- `\transblindsvertical`, 247
- `\transboxin`, 247
- `\transboxout`, 247
- `\transdissolve`, 247
- `\transglitter`, 247
- Transparencias, 232, 253
- Tres puntos consecutivos, 60
- `\triangle`, 75
- `\triangleleft`, 75
- `\triangleq`, 76
- `\triangleright`, 75
- `\trianglerighteq`, 76
- `\tt`, 19, 118, 205, 247
- Typewriter, 19
- Ubuntu, 3, 5, 14, 52, 116, 135
- `\unboldmath`, 63
- `\uncover`, 236, 242
- `\underbrace`, 63

- `\underline`, **19**, **249**
- `Unir celdas`, **94**
- `\unitlength`, **206**
- `\up`, **107**
- `\uparrow`, **107**
- `\uplus`, **75**
- `\Upsilon`, **74**
- `\upsilon`, **74**
- `usebackgroundtemplate\usebackgroundtemplate`, **261**
- `\usefonttheme`, **233**
- `usefonttheme\usefonttheme`, **255**, **257**
- `\usemintedstyle`, **224**
- `\usepackage`
  - `algorithm2e`, **216**, **244**
  - `amsmath`, **8**, **11**, **15**, **16**, **43**, **56**, **107**, **210**, **233**
  - `amsmath`, **3**, **8**, **11**, **15**, **16**, **43**, **56**, **73**, **107**, **210**, **233**
  - `amssymb`, **8**, **11**, **15**, **16**, **43**, **56**, **107**, **210**, **233**
  - `answers`, **209**, **210**
  - `anyfontsize`, **20**
  - `babel`, **13**, **13**, **56**, **205**, **210**
  - `bookman`, **51**, **152**
  - `calligra`, **50**
  - `cancel`, **43**, **56**
  - `cmbright`, **51**
  - `colortbl`, **107**
  - `concrete`, **51**
  - `enumitem`, **41**
  - `epstopdf`, **116**, **234**, **245**
  - `eulervm`, **51**
  - `fancyhdr`, **204**, **205**
  - `fancyvrb`, **53**, **53**
  - `fix-cm`, **20**
  - `float`, **118**
  - `floatflt`, **119**
  - `fontenc`, **3**, **12**, **43**, **50**, **56**, **205**, **210**, **233**
  - `fourier`, **51**
  - `geometry`, **11**, **11**
  - `graphicx`, **8**, **11**, **15**, **16**, **43**, **116**, **116**, **118**
  - `helvet`, **51**, **205**
  - `inputenc`, **11**, **12**, **15**, **16**, **52**, **56**, **107**, **205**, **210**, **233**
  - `latexsym`, **8**, **11**, **15**, **16**, **43**, **56**, **107**, **210**, **233**
  - `lstlisting`, **243**
  - `makeidx`, **43**, **45**
  - `makeidx`, **45**
  - `marginnote`, **33**
  - `mathpazo`, **51**, **151**
  - `mathptmx`, **51**, **152**
  - `minted`, **224**, **243**
  - `multicol`, **23**, **23**
  - `multimedia`, **247**
  - `newcent`, **51**, **152**
  - `palatino`, **51**
  - `pslatex`, **51**
  - `pstricks`, **33**
    - `black`, **33**
    - `blue`, **33**
    - `cyan`, **33**
    - `darkgray`, **33**
    - `gray`, **33**
    - `green`, **33**
    - `lightgray`, **33**
    - `magenta`, **33**
    - `red`, **33**
    - `white`, **33**
    - `yellow`, **33**
  - `pxfonts`, **51**
  - `rotating`, **92**
  - `stmaryrd`, **107**, **233**
  - `subfig`, **118**
  - `tabularx`, **102**, **107**
  - `textcomp`, **48**
  - `tikz`, **41**, **107**
  - `txfonts`, **51**
  - `verbments`, **225**, **226**
  - `wrapfig`, **119**
  - `xargs`, **79**
  - `xcolor`, **30**, **90**
  - `xltxtra`, **52**
- `usepackage\usepackage`
  - `amsmath`, **255**
  - `amssymb`, **255**
  - `amsthm`, **255**
  - `babel`, **255**
  - `beamerposter`, **255**
  - `geometry`, **260**
  - `inputenc`, **255**, **260**, **263**
  - `latexsym`, **255**
  - `xcolor`, **264**
- `\usetheme`, **233**
- `usetheme\usetheme`, **255**
  - `Warsaw`, **257**
- `\usetikzlibrary`, **107**
- `utf8`, **261**



- `\varepsilon`, 74
- `\varphi`, 74
- `\varpi`, 74
- `\varrho`, 74
- `\varsigma`, 74
- `\vartheta`, 74
- `\vartriangleright`, 76
- VBA, 54
- `\vdash`, 75
- `\vdots`, 60, 66, 70
- `\vec`, 33, 63, 78
- `\vee`, 75
- Ver
  - Resultado, 8
- `\verb`, 205
- `\verb`, 18
- Verbatim
  - codes, 53
  - commandchars, 53
  - fontfamily, 53
  - formatcom, 53
  - frame, 53
  - resetmargins, 53
  - xleftmargin, 53
  - xrightmargin, 53
- `\VerbatimFootnotes`, 53
- verbments, 225
  - Texto matemático, 225
- verbments.sty, 225
- VERYHuge\VERYHuge, 255, 257
- VeryHuge\VeryHuge, 255
- veryHuge\veryHuge, 255
- `\visible`, 242
- `\vrule`, 95
- `\vspace*`, 233
- `\vspace`, 22, 22
- `\vspace*\vspace*`, 264
- `\vv`, 63
- `\wedge`, 34, 75
- `\While`, 216, 217, 219
- `\widehat`, 76
- `\widetilde`, 76
- Windows, 3, 47, 116, 136, 142
  - Bloc de notas, 140
- Winplot, 116
- Winshell, 4
- `\wp`, 75
- `\wr`, 75
- wrapfigure, 119, 120
- Writer, 5
- Writer2LaTeX, 5
- `\wvec`, 78
- `\x`, 163, 164
- XeLaTeX, 52
- `\xi`, 74
- `\xi`, 74, 134
- Yap, 116
- `\Z`, 69
- `\zeta`, 74