

Segunda edición

El universo

LATEX

Rodrigo De Castro Korgi

Versión interactiva del libro
Incluye software



Facultad de Ciencias



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Departamento de Matemáticas

EL UNIVERSO L^AT_EX

EL UNIVERSO L^AT_EX

Rodrigo De Castro Korgi

Departamento de Matemáticas
Facultad de Ciencias
Universidad Nacional de Colombia
Sede Bogotá

EL UNIVERSO L^AT_EX

© Rodrigo De Castro Korgi
Profesor Asociado

Departamento de Matemáticas
Facultad de Ciencias
Universidad Nacional de Colombia

© Universidad Nacional de Colombia
Facultad de Ciencias
Departamento de Matemáticas

Segunda edición, 2003
Segunda reimpresión, 2007
Tercera reimpresión, 2008
Bogotá, Colombia

ISBN 958-701-060-4

Impresión: Universidad Nacional de Colombia, Unibiblos
dirunibiblio_bog@unal.edu.co
Bogotá, Colombia

Diagramación en L^AT_EX : el autor
Diseño de carátula: Andrea Kratzer

Catalogación en la publicación Universidad Nacional de Colombia

68.22544
D291u De Castro Korgi, Rodrigo
El universo L^AT_EX / Rodrigo De Castro Korgi. – Bogotá:
Universidad Nacional de Colombia. Facultad de Ciencias.
Departamento de Matemáticas, 2003
xiv, 470 p. : il.

ISBN 958-701-060-4

1. L^AT_EX (programa para computador) 2. Escritura computarizada
3. Tipos (imprenta) – Símbolos matemáticos 4. Composición tipográfica
automatizada.
I. Tít.

Índice General

Prólogo a la segunda edición	ix
Contenido del CD	xi
1. T_EX, L_AT_EX y L_AT_EX 2_ε	1
2. Estructura de un documento L_AT_EX	4
2.1. L _A T _E X-nica de procesamiento de documentos	4
2.2. Tipos de comandos L _A T _E X	7
2.3. Tipos de documentos L _A T _E X	10
2.4. El preámbulo de un documento	13
2.5. El estilo article (artículo)	13
2.6. El estilo book (libro)	17
2.7. El estilo report (reporte o informe)	18
2.8. El uso de paquetes	18
2.9. El paquete babel	20
2.10. El formato de página	21
2.11. Unidades de medida y longitudes en L _A T _E X	26
2.12. Numeración de las páginas	27
2.13. Modificaciones del formato de página	28
2.13.1. El paquete layout	28
2.13.2. Modificación de los parámetros	30
2.14. Páginas con una o dos columnas	32
3. Herramientas de edición básicas	33
3.1. Tildes y acentos	33
3.2. Puntuación	34
3.3. Tipos de letra y fuentes	37
3.4. Tamaño de la letra	39
3.5. Combinaciones de tipo y tamaño de letra	40
3.6. Caracteres especiales y otros símbolos	42
3.7. Espacio horizontal	43
3.8. Sangrías	44

3.9. Espacio vertical	44
3.10. Control sobre cambios de página	46
3.11. Texto subrayado	47
3.12. Citas	47
3.13. Texto centrado	49
3.14. Texto cargado a la izquierda o a la derecha	50
3.15. Cajas	50
3.15.1. Cajas con una sola línea de texto	51
3.15.2. Cajas con párrafos	52
3.15.3. Parámetros para cajas con líneas	55
3.15.4. Traslación vertical de cajas	55
3.15.5. Cajas invisibles y cajas de anchura nula	56
3.16. Mini-páginas	57
3.17. Notas al pie de página	58
3.18. Notas marginales	61
3.19. Listas con incisos o items	62
3.19.1. Los entornos <code>itemize</code> y <code>enumerate</code>	62
3.19.2. El entorno <code>description</code>	63
3.19.3. Listas encajadas en listas	64
3.19.4. Listas con el paquete <code>enumerate</code>	65
3.20. El entorno <code>verbatim</code>	67
3.21. El paquete <code>verbatim</code>	68
3.22. Líneas horizontales y verticales	69
3.23. Relleno con espacio en blanco, puntos, líneas o flechas	69
3.24. Definición de comandos nuevos	72
3.24.1. Comandos simples	73
3.24.2. Almacenamiento de cajas	73
3.25. Contadores y referencias cruzadas	75
3.25.1. Modificación de un contador	77
3.25.2. Estilos de numeración para contadores	78
3.25.3. Contadores creados por el usuario	79
3.26. Referencias bibliográficas	80
3.27. Rastreo de claves asignadas por el usuario	84
3.27.1. Rastreo con el paquete <code>showkeys</code>	84
3.27.2. Rastreo con el archivo <code>lablst.tex</code>	85
4. Matemáticas	86
4.1. Modo matemático	86
4.2. Los paquetes <code>amsmath</code> y <code>amssymb</code>	87
4.3. Galería de símbolos matemáticos	88
4.4. Recomendaciones para la correcta escritura en matemáticas	95

4.5. Subíndices y superíndices	97
4.6. Puntos suspensivos	98
4.7. Fracciones	98
4.8. Coeficientes binomiales	99
4.9. Raíces	100
4.10. Funciones y operadores básicos	101
4.11. Texto en expresiones matemáticas	102
4.12. Sumas	103
4.13. Integrales	105
4.14. Operadores grandes	107
4.15. Fórmulas en cajas	108
4.16. Acentos en modo matemático	109
4.17. Micro-espacios	111
4.18. Operadores con límites inferiores	112
4.19. Relaciones de congruencia	114
4.20. Paréntesis y símbolos de agrupación	115
4.21. Casos	118
4.22. Matrices	119
4.23. Matrices con el entorno <code>array</code> de LATEX	121
4.24. Tipos de letra en modo matemático	122
4.25. Símbolos en negrilla	124
4.26. Tamaño de los símbolos	126
4.27. Símbolos sobre símbolos	128
4.28. Definición de nuevos comandos	129
4.29. Separación de expresiones matemáticas en el margen derecho	132
4.30. Alineación y numeración de fórmulas	133
4.30.1. División de fórmulas con <code>multline</code>	134
4.30.2. Alineaciones con <code>gather</code>	135
4.30.3. Alineaciones con <code>align</code>	136
4.30.4. Alineaciones con <code>split</code>	139
4.30.5. Alineaciones con <code>aligned</code> y <code>gathered</code>	141
4.30.6. Alineaciones con <code>flalign</code>	144
4.30.7. Alineaciones con el entorno <code>eqnarray</code> de LATEX	145
4.30.8. Espaciamiento vertical en alineaciones	146
4.30.9. Control sobre cambios de página en alineaciones	147
4.31. Opciones para la numeración de fórmulas	147
4.31.1. Colocación y numeración de fórmulas	147
4.31.2. Jerarquía de la numeración	148
4.31.3. Numeración forzada	148

4.31.4. Numeración subordinada	149
4.31.5. Referencias cruzadas	150
4.31.6. Ajustes en la posición de los números	150
4.32. Teoremas y estructuras relacionadas	150
4.32.1. Opciones de <code>\newtheorem</code>	152
4.32.2. El comando <code>\newtheorem</code> en el paquete <code>amsthm</code> . .	153
4.32.3. El entorno <code>proof</code> del paquete <code>amsthm</code>	154
4.33. Diagramas conmutativos con <code>amscd</code>	156
4.34. Diagramas conmutativos con <code>pb-diagram</code>	158
4.35. Los estilos <code>amsart</code> y <code>amsbook</code>	164
5. Tablas	167
5.1. Tablas con líneas	169
5.2. Tablas con párrafos	170
5.3. Tablas con filas especiales	171
5.4. Simplificación del formato de una tabla	172
5.5. Líneas horizontales con <code>\cline</code>	172
5.6. Líneas verticales con <code>\vline</code>	173
5.7. Espacio horizontal adicional	173
5.8. Espacio vertical adicional	174
5.9. Grosor de las líneas de una tabla	176
5.10. @-expresiones	177
5.11. Inserción y numeración de tablas	177
5.12. Tablas con el paquete <code>array</code>	178
5.13. Tablas y texto circundante	181
5.14. Tablas extensas, tablas a color y rotación de tablas	182
6. Manejo de documentos grandes	183
6.1. Partes o divisiones en un documento <code>LATEX</code>	183
6.2. Numeración de las partes o divisiones	184
6.3. El documento raíz	185
6.4. Apéndices	187
6.5. Redefinición de títulos o rótulos	188
6.6. Tabla de contenido o índice general	190
6.7. Índices de tablas y figuras	191
6.8. Elaboración manual de índices alfabéticos	193
6.9. Elaboración de índices alfabéticos con el programa <i>MakeIndex</i>	194
6.9.1. El uso del comando <code>\index</code>	196
6.9.2. Rastreo de los comandos <code>\index</code>	202
6.9.3. Opciones del programa <i>MakeIndex</i>	203

6.9.4. Índices múltiples con el paquete <code>multind</code>	204
6.10. Bibliografías con el programa BIBTEX	205
6.10.1. Formato de un archivo ‘ <code>—.bib</code> ’	207
6.10.2. Estilos bibliográficos	212
6.10.3. Estilos bibliográficos autor-año	214
7. El ambiente gráfico de LATEX 2ϵ	216
7.1. PostScript y <code>dvips</code>	217
7.2. El paquete <code>color</code>	218
7.2.1. Definición de nuevos colores	218
7.2.2. Texto y cajas en color	219
7.3. Los paquetes <code>graphics</code> y <code>graphicx</code>	221
7.3.1. Aumento a escala de objetos	221
7.3.2. Reflexión de objetos	223
7.3.3. Rotación de objetos	223
7.3.4. Inclusión de gráficas externas en documentos LATEX	226
7.4. Gráficas y tablas como objetos flotantes	235
7.4.1. Problemas con la inserción de objetos flotantes . .	236
7.4.2. Objetos flotantes horizontalmente centrados . . .	238
7.5. Epígrafes y numeración de tablas y gráficas	239
7.6. El entorno <code>picture</code> de LATEX 2ϵ	242
7.6.1. Grillas con el comando <code>\graphpaper</code>	243
7.6.2. El comando <code>\put</code>	244
7.6.3. Líneas rectas con <code>\line</code>	245
7.6.4. Flechas con <code>\vector</code>	246
7.6.5. Círculos con <code>\circle</code>	246
7.6.6. El comando <code>\oval</code>	247
7.6.7. Cajas	248
7.6.8. El comando <code>\shortstack</code>	249
7.6.9. Curvas cuadráticas de Bézier	250
7.6.10. El comando <code>\multiput</code>	251
8. Otras herramientas de LATEX 2ϵ	252
8.1. El estilo <code>slides</code> para transparencias	252
8.2. El estilo <code>letter</code> para cartas	257
8.3. El entorno <code>list</code>	260
8.4. Definición de nuevos entornos	262
8.5. Longitudes elásticas	264
8.6. Comandos internos de LATEX	264
8.7. Modificación de los comandos seccionales	265
8.8. Documentos LATEX por correo electrónico	267

9. El uso de otras fuentes en documentos L^AT_EX 2_ε	269
9.1. Atributos de las fuentes	269
9.2. Los archivos de fuentes	271
9.3. La colección AE de fuentes PostScript	272
9.4. La colección PSNFSS de fuentes PostScript	273
9.4.1. Las familias de fuentes de PSNFSS	273
9.4.2. Descripción de los paquetes de PSNFSS	276
9.4.3. El paquete <code>mathptmx</code>	277
9.4.4. El paquete <code>mathpazo</code>	279
9.4.5. El paquete <code>pifont</code>	281
9.4.6. Codificación de las fuentes de la colección PSNFSS .	283
9.5. Comandos para cambiar las familias de fuentes en un documento	284
9.6. Acceso a una fuente con los comandos <code>\newfont</code> y <code>\DeclareFixedFont</code>	285
9.7. Acceso a los símbolos de una fuente	286
10. Documentos L^AT_EX interactivos	289
10.1. Archivos en formato PDF	290
10.2. El paquete <code>hyperref</code>	292
10.2.1. Opciones de <code>hyperref</code>	293
10.2.2. Enlaces internos y externos	296
10.2.3. Inclusión de gráficas	297
10.2.4. Acciones de menú en Acrobat Reader o Adobe Acrobat	298
10.2.5. Control sobre la lista de <i>Marcadores (Bookmarks)</i> .	300
10.3. Los programas <code>pdfL^AT_EX</code> y <code>pdfl^AT_EX</code>	301
10.3.1. Configuración de <code>pdfL^AT_EX</code>	301
10.3.2. Algunos comandos primarios de <code>pdfL^AT_EX</code>	303
10.4. Ejemplo de un documento L ^A T _E X interactivo en formato PDF	306
11. Gráficas con el paquete PSTricks	308
11.1. El programa <code>PSTricks</code> y su instalación	308
11.2. Las características básicas de una gráfica en <code>PSTricks</code> . .	311
11.3. El uso de colores en <code>PSTricks</code>	313
11.4. Líneas rectas	314
11.5. Polígonos	317
11.6. Circunferencias y arcos de circunferencia	318
11.7. Elipses	320
11.8. Parábolas	321

11.9. Relleno y sombreado de regiones. Parte I	321
11.10. Sombras	326
11.11. Puntos	327
11.12. Los comandos “estrella”	328
11.13. Texto enmarcado	329
11.14. Colocación y rotación de objetos	331
11.15. Colocación de etiquetas	334
11.16. Ejes coordenados	336
11.17. Grillas	339
11.18. Repeticiones	341
11.19. Objetos gráficos personalizados	343
11.20. Trazado de curvas por interpolación I	344
11.21. Trazado de curvas por interpolación II	348
11.22. Trazado de funciones a partir de su ecuación	351
11.23. Curvas paramétricas	357
11.24. Relleno y sombreado de regiones. Parte II	359
11.25. Otras coordenadas	365
11.26. Grafos: nodos y conectores de nodos	367
11.26.1. Tipos de nodos	367
11.26.2. Conectores de nodos	368
11.26.3. Etiquetas para los conectores de nodos	373
11.27. Árboles	377
11.28. Resortes y zigzags	383
11.29. Texto a lo largo de curvas	386
11.30. Texto con efectos especiales	387
12. Gráficas con el paquete <i>P<small>i</small>C<small>T<small>E</small>X</small></i>	388
12.1. El paquete <i>P<small>i</small>C<small>T<small>E</small>X</small></i> y su instalación	388
12.2. Características de una gráfica en <i>P<small>i</small>C<small>T<small>E</small>X</small></i>	389
12.3. Grillas	391
12.4. Rectas y polígonos	392
12.5. Circunferencias y arcos de circunferencia	393
12.6. Elipses y arcos de elipse	393
12.7. El comando <i>\put</i>	394
12.8. Rectángulos e histogramas	396
12.9. El comando <i>\multiput</i>	397
12.10. Letras y palabras apiladas	398
12.11. Ejes coordenados	399
12.12. Trazado de curvas	404
12.13. Grosor de líneas y curvas	407
12.14. Líneas y curvas punteadas	408

12.15. Líneas y curvas a trozos	410
12.16. Flechas	414
12.17. Rotaciones	416
12.18. Gráficos de barras	418
12.19. Sombreado de regiones	420
13. Otros paquetes importantes	426
13.1. El paquete <code>fancyhdr</code>	426
13.2. El paquete <code>multicol</code>	429
13.3. El paquete <code>fncychap</code>	430
13.4. El paquete <code>fancybox</code>	432
13.5. El paquete <code>colortbl</code>	434
13.6. El paquete <code>longtable</code>	437
13.7. El paquete <code>caption2</code>	438
13.8. El paquete <code>picinpar</code>	440
13.9. El paquete <code>lscape</code>	443
13.10. El paquete <code>subfiles</code>	443
Apéndice A	445
A.1 Mensajes de error de <code>LATEX</code>	445
A.2 Mensajes de error de <code>TEX</code>	446
Apéndice B. Configuración de <code>LATEX</code> para el uso de <code>babel</code>	448
Apéndice C. La estructura de directorios TDS y la instalación de paquetes <code>LATEX</code>	449
Bibliografía	451

Prólogo a la segunda edición

Para producir libros, artículos y otros documentos técnicos de calidad artística debemos ingresar al universo L^AT_EX. Basado en el programa núcleo T_EX, L^AT_EX se ha convertido en el estándar tipográfico del mundo científico y ha creado una distinguida clase de T_EX-nócratas cuyos miembros se cuentan por decenas de miles. El éxito de la primera edición de *El Universo L^AT_EX* ratifica el éxito continuo de L^AT_EX a nivel mundial.

La segunda edición mantiene los propósitos y la filosofía que inspiraron la primera: servir de guía de aprendizaje y manual de referencia para la versión estándar de L^AT_EX, denominada L^AT_EX 2_ε, y para numerosos paquetes que se ejecutan en su ambiente. No nos contentamos simplemente con presentar una lista escueta de comandos, lo cual requeriría solo un puñado de páginas, sino que ilustramos el uso de los distintos comandos por medio de abundantes ejemplos. Puesto que el usuario utiliza L^AT_EX para producir documentos concretos, la manera más eficiente de aprender el programa es a través del examen de ejemplos concretos, lejos de las descripciones sintácticas abstractas. Nuestra presentación de todas las instrucciones de L^AT_EX y los demás paquetes de macros es, no obstante, completa y rigurosa.

Lo nuevo en la segunda edición

El texto de la primera edición ha sido sometido a una revisión exhaustiva: se han corregido ciertas imprecisiones, así como errores ortográficos y tipográficos, muchos de los cuales le fueron señalados al autor por lectores atentos. Se han expandido las explicaciones de las herramientas básicas de L^AT_EX 2_ε y de la mayoría de los paquetes. Se ha actualizado la presentación de paquetes como *babel* y *amsmath*, teniendo en cuenta los cambios que éstos han tenido desde la primera edición del libro. La presentación de las fuentes y paquetes de la colección PSNFSS también se ha actualizado.

En esta edición se describen algunos tópicos y paquetes L^AT_EX que no fueron mencionados en la primera edición, entre los que se incluyen:

1. Los paquetes básicos de L^AT_EX 2_ε, *enumerate*, *longtable*, *lscape*, *flafter*, y *showkeys*.

2. Los paquetes **amscd** y **pb-diagram** para diagramas commutativos.
3. El paquete **hyperref** y el programa **pdflatEX** para la producción de documentos **LATEX** interactivos.
4. Los paquetes **fancybox** (macros adicionales para cajas con bordes), **fncychap** (nuevos estilos para la presentación de capítulos) y **picinpar** (para la inserción de “ventanas” en párrafos normales).
5. Comandos primarios de **TEX** y algunos de los llamados comandos internos de **LATEX**, útiles en documentos corrientes.

La adición más notoria es la inclusión de un CD con abundante software y con una versión interactiva del libro. El contenido del CD se describe en las páginas xi a xiv.

Agradecimientos

Debo agradecer los comentarios positivos y las valiosas observaciones que recibí por parte de muchos lectores de la primera edición. Agradezco también a Juan Lesmes por su lectura minuciosa de todo el manuscrito de la nueva edición.

La concepción, diseño y elaboración de la versión electrónica de *El universo LATEX* estuvieron a cargo de mi sobrino, el compositor Federico García De Castro. Su colaboración en este proyecto ha sido realmente invaluable y dispongo de pocas palabras para manifestarle mi gratitud.

De manera muy especial quiero agradecer a Gustavo Rubiano, Director de la Oficina de Publicaciones de la Facultad de Ciencias, por sus acertadas sugerencias, su apoyo continuo y sus ingentes esfuerzos en procura de una edición de gran calidad.

Esta edición se publica bajo el auspicio del Departamento de Matemáticas y la Facultad de Ciencias de la Universidad Nacional de Colombia, sede de Bogotá, instituciones a las que expreso mi gratitud y reconocimiento.

Correo electrónico del autor

Los **TEX**-nócratas interesados están invitados a entrar en contacto con el autor, escribiendo a la dirección rdcastro@matematicas.unal.edu.co, para expresar sus comentarios, plantear preguntas o inquietudes y compartir sus **TEX**-periencias en el universo **LATEX**.

Contenido del CD

Versión electrónica de *El Universo LATEX*

La presente edición de *El Universo LATEX* incluye, en el CD adjunto, una versión electrónica o virtual del libro, en formato PDF interactivo. Para acceder a la versión electrónica se hace *click* en el archivo *Universo.pdf*, ubicado en la raíz del CD. Se recomienda usar la versión 5.0, o posterior, del programa gratuito *Acrobat Reader* (la versión 5.1 en español, para *Windows 98/2000/XP*, se incluye también en el CD).

La versión electrónica conlleva un triple propósito:

1. Servir de resumen del libro, enriqueciendo y complementando la presentación del material. Es éste el propósito fundamental; la versión electrónica coincide con la versión impresa en la distribución del contenido por capítulos y secciones, pero posee lujo prohibidos en el papel: el uso ilimitado de colores y la interactividad de los enlaces y el hipertexto. No obstante, la versión electrónica no reemplaza al libro mismo ya que no contiene exposiciones detalladas de todos los tópicos y, en el caso de los más delicados, el usuario es remitido al libro para las explicaciones pertinentes.
2. Servir de referencia para los ejemplos del libro. Una de las características centrales de *El Universo LATEX* es la abundancia de ejemplos para los diferentes comandos presentados. La versión electrónica incluye prácticamente todos los ejemplos de los capítulos 1 a 12 del libro, y el lector puede utilizarlos como modelos o guías para sus propios documentos.
3. Ilustrar las posibilidades de *pdfLATEX*. En la actualidad, la publicación no se limita a documentos impresos en papel; con *pdfLATEX*, un documento *LATEX* cualquiera se puede convertir en un elegante y flexible documento interactivo para diversos usos, como presentaciones, conferencias, material didáctico o documentos destinados a la *Web*.

Esperamos que esta versión electrónica de *El Universo L^AT_EX* sirva al lector como pauta o inspiración para documentos similares. El programa pdfL^AT_EX propiamente dicho se describe con todo detalle en el Capítulo 10, junto con el paquete `hyperref`, herramienta de gran utilidad para la definición de enlaces interactivos y el control del archivo PDF final.

Software incluido en el CD

El CD también incluye software de dominio público y software de evaluación (conocido como *shareware*) que puede ser de utilidad para los usuarios de L^AT_EX. Recordamos al lector que los programas de tipo *shareware* requieren registro o licencia después del período de evaluación. A continuación se describen los programas incluidos en el CD, con indicaciones sobre su instalación.



MiKTeX, versión 2.2. Implementación gratuita de T_EX para Windows 95/98/2000/ME/NT/XP/. Incluye, entre otros, T_EX, L^AT_EX, pdfT_EX, pdfL^AT_EX, dvips, BIBT_EX, *MakeIndex*, fuentes CM (*bitmaps* y PostScript), visor YAP y todos los paquetes estándares de L^AT_EX 2_E. Versión descargada de la página www.miktex.org.

Instalador: `Setup.exe`.



WinEdt, versión 5.2. (*Shareware*; período de evaluación: 31 días). Completa y versátil interfaz gráfica para Windows 95/98/2000/ME/NT/XP/. Está previamente configurado para MiKTeX, aunque se puede reconfigurar para otras implementaciones de T_EX. Posee herramientas muy variadas como soporte para múltiples diccionarios, definición de macros, rastreo de errores, acceso a programas externos, etc. Versión descargada de la página www.winedt.com.

Instalador: `Setup.exe` (instalarlo *después* de MiKTeX).



TeXnicCenter, versión 1 Beta 6.01. Interfaz para Windows 95/98/2000/ME/NT/XP/, diseñada exclusivamente para documentos L^AT_EX. Se asemeja a WinEdt pero es completamente gratuito. Está previamente configurado para MiKTeX, aunque se puede instalar con otras implementaciones de T_EX. Versión descargada de la página www.ToolsCenter.org.

Instalador: `TXCSetup_1Beta6_01.exe` (instalarlo *después* de MiKTeX).



WinShell for \TeX , versión 2.2.1. Otra interfaz gratuita para *Windows* 95/98/2000/NT/. Es más modesto que WinEdt y $\text{\TeX}nicCenter$ pero más fácil de usar. Posee interfaces en varios idiomas, incluyendo el español. Está previamente configurado para MiK \TeX , aunque se puede instalar con otras implementaciones de \TeX . Versión descargada de la página www.winshell.de.

Instalador: **WinShell221.exe** (instalarlo *después* de MiK \TeX).



Diccionarios para editores compatibles (como WinEdt). Se incluyen: español, inglés, francés, portugués, alemán e italiano. Para detalles sobre la instalación de diccionarios en WinEdt se puede consultar la plantilla Help.



Oz \TeX , versión 4.0. (*Shareware*). Implementación de \TeX y L \TeX para Macintosh.



Ghostscript, versión 7.04. Interpretador de archivos PostScript (se describe en la sección 7.1). Versión para *Windows* descargada de la página www.cs.wisc.edu/~ghost/.

Instalador: **gs704w32.exe**.



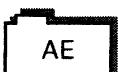
GSview, versión 4.3. Interfaz para Ghostscript. Versión para *Windows* descargada de la página www.cs.wisc.edu/~ghost/.

Instalador: **gsv43w32.exe**.



Adobe Acrobat Reader, versión 5.1 en español, para *Windows* 98/2000/XP. Visor de archivos PDF con interfaz para navegación e impresión. Versión gratuita descargada de la página www.adobe.com.

Instalador: **AcroReader51_ESP.exe**.



Colección AE de fuentes PostScript. Estas fuentes se describen en la sección 9.3.



Colección PSNFSS de fuentes PostScript, versión 8.2. Estas fuentes se describen en la sección 9.4.



Numerosos paquetes L \TeX , distribuidos en subcarpetas; el nombre de cada subcarpeta coincide con el nombre del paquete. Se incluyen *todos* los paquetes L \TeX mencionados o descritos en el libro y muchos otros que pueden ser de utilidad para el lector. Información general sobre paquetes se encuentra en la sección 2.8; una guía para la instalación de nuevos paquetes se presenta en el Apéndice C. Los paquetes han sido descargados de la página Web de CTAN (*Comprehensive \TeX Archive Network*); véase más información sobre CTAN en el Capítulo 1.



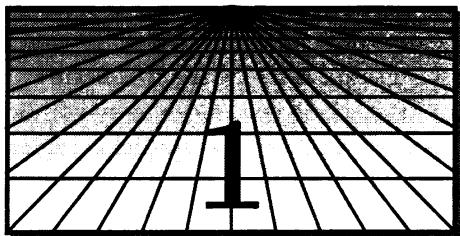
Contiene los patrones de partición silábica de numerosos idiomas. Sobre la configuración y uso del paquete **babel** de L^AT_EX, véase la sección 2.9 y el Apéndice B.



Colección de más de 150 estilos bibliográficos (archivos **bst**) y numerosas bases de datos bibliográficas (archivos **bib**) para el programa B^IB_TE_X (descrito detalladamente en la sección 6.10).



Ejemplo de un documento L^AT_EX interactivo en formato PDF. Para detalles sobre este ejemplo, véase la sección 10.4.



TeX, L^AT_EX y L^AT_EX 2_ε

TeX: pronúnciese tek o tej. El creador de TeX, Donald Knuth, dice: "si Ud. lo pronuncia correctamente, la pantalla del computador debe quedar empañada".

Durante casi una década, Donald Knuth¹, de la Universidad de Stanford, trabajó en el diseño de un sistema de procesamiento de textos basado en un conjunto de instrucciones o comandos tipográficos básicos. Una vez finalizado el programa núcleo, en 1983, Knuth escribió: "TeX es un nuevo sistema para la producción de hermosos libros, especialmente libros que contengan muchas expresiones matemáticas". El nombre TeX representa el prefijo griego $\tau\epsilon\chi$, que significa arte, y que aparece en palabras como técnica o tecnología. El logo mismo conlleva la idea de lo que se puede hacer con el programa: manipular símbolos y colocarlos en las posiciones deseadas, utilizando únicamente un conjunto básico de instrucciones computacionales. TeX es capaz de producir su propio logo simplemente a partir del comando \TeX, que está definido a su vez con comandos primarios (¡trate Ud. de producir la expresión TeX con un procesador de texto corriente!). El nombre del programa hace énfasis en su propósito fundamental: la generación de textos *técnicos* de calidad artística.

El programa TeX, y sus extensiones posteriores, se hicieron rápidamente populares hasta convertirse en el estándar tipográfico actual en matemáticas, ciencias e ingeniería. La biblia de TeX, escrito por el propio Knuth bajo el título *The TeX-book*, ha tenido dieciocho ediciones desde su aparición en 1986. Una de las características más notables de TeX es que un documento fuente se escribe únicamente con los símbolos del teclado y puede ser procesado en múltiples plataformas computacionales, desde micro-computadores IBM compatibles, Macintosh o Linux, hasta estaciones de trabajo VMS o super-computadores Cray, para producir el mismo documento final. Un documento fuente TeX, dada su portabilidad y tamaño, es de fácil transmisión por correo electrónico.

¹El lector encontrará una semblanza de Donald Knuth en la página 442.

Especialmente popular es el paquete \LaTeX , creado por Leslie Lamport en 1985, y que consiste en una serie de macros o rutinas \TeX que facilitan las tareas de edición. \LaTeX tiene comandos muy cómodos y elegantes para la creación de tablas, listas, bibliografías, índices, referencias cruzadas, etc, y permite al usuario concentrarse en la *estructura* del documento en vez de los detalles puramente \TeX -nicos.

La popularidad de \TeX y \LaTeX ha creado una serie de personajes exóticos en el mundo académico:

\TeX -nico: Conocedor de todas las minucias, argucias y misterios de \TeX .

\TeX -perto: Persona familiarizada con los comandos básicos de \TeX , capaz de programar rutinas o macros \TeX .

\TeX -nócrata: Persona que ha asimilado la filosofía \TeX y que es un usuario asiduo de \TeX o \LaTeX (aunque desconoce o no le interesa el funcionamiento “interno” de los comandos del programa).

El propósito de este libro es crear \TeX -nócratas proficientes en \LaTeX . Creemos que \LaTeX es el ambiente ideal para aquellas personas interesadas en utilizar las herramientas más sofisticadas de edición de textos \TeX -nicos para obtener documentos de calidad suprema.

Haciendo un poco de historia, podemos considerar con algún detalle las más importantes extensiones de \TeX ; esto nos permite entender por qué \LaTeX ha llegado ha tener tanta prominencia en la comunidad mundial de \TeX -nócratas.

\LaTeX 2.09. Fue durante muchos años la versión estándar de \LaTeX , hasta que se hizo obsoleta en junio de 1994 con la aparición de $\text{\LaTeX}_2\epsilon$. Su principal debilidad fue siempre su limitada capacidad para la edición de expresiones matemáticas.

\AMS-TeX . Paquete de macros \TeX diseñado en 1987, con el auspicio de la AMS, *American Mathematical Society*, para facilitar construcciones matemáticas y uniformizar la producción de artículos o libros de contenido esencialmente matemático. Es independiente y totalmente incompatible con \LaTeX .

\AMS-LaTeX . Primer intento de integración de \AMS-TeX y \LaTeX , realizado principalmente por Frank Mittelbach, Rainer Schöpf y Michael Downes; apareció en 1990. Creó un cierto caos en la comunidad de \TeX -nócratas debido a que el procesamiento de un documento \AMS-LaTeX requería usualmente la re-configuración, en el computador local, del formato normal de procesamiento de \LaTeX .

$\text{\LaTeX} 2\epsilon$. Revisión completa de la versión \LaTeX 2.09; uno de sus propósitos centrales fue la integración dentro de un ambiente único de \LaTeX , \AMS-\TeX y \AMS-\LaTeX . La idea fundamental de $\text{\LaTeX} 2\epsilon$ es que toda futura adición o extensión a \LaTeX se haga por medio de “paquetes” individuales, que el usuario puede invocar por medio de la instrucción `\usepackage{...}`. De esta manera se pone punto final a la proliferación de dialectos incompatibles. Como ejemplo, \AMS-\LaTeX es ahora simplemente el paquete `amsmath`, al cual se accede con `\usepackage{amsmath}`.

$\text{\LaTeX} 3$. Será la próxima versión unificada de \LaTeX para el nuevo milenio; de hecho, $\text{\LaTeX} 2\epsilon$ es el primer paso en la consolidación de dicho programa. Se pretende hacer una profunda revisión a la interfaz de diseño de nuevos paquetes con el fin de facilitar su construcción y su uso. El proyecto $\text{\LaTeX} 3$ se adelanta bajo la dirección de Frank Mittelbach, Chris Rowley y Rainer Schöpf.

pdf \TeX y pdf \LaTeX . Extensión de \TeX (\LaTeX) capaz de generar directamente PDF a partir de un documento fuente \TeX (\LaTeX). Este proyecto fue iniciado y desarrollado principalmente por H n Th  Th n en el a o 2000. Por su r pido acceso, su seguridad, su capacidad interactiva y su portabilidad, el formato PDF se ha convertido en uno de los preferidos para la publicaci n, tradicional y electr nica, en los c rculos acad micos y cient ficos.

Tanto usuarios como fan ticos pueden visitar la red mundial de software \TeX , conocida como CTAN (*Comprehensive \TeX Archive Network*). Los servidores de CTAN contienen implementaciones para pr cticamente todos los sistemas operativos, as  como paquetes de macros \TeX y \LaTeX , y muchos programas  tiles. El software disponible es de dominio p blico o de tipo *shareware*. All  se encuentra tambi n toda la informaci n actualizada sobre la marcha del proyecto $\text{\LaTeX} 3$. Los tres servidores centrales de CTAN, dotados de poderosos motores de b squeda, son:

En EE.UU: <http://www.ctan.org/>

En Inglaterra: <http://www.tex.ac.uk/tex-archive/>

En Alemania: <http://www.dante.de/cgi-bin/ctan-index/>

Para los usuarios de habla hispana puede ser de inter s la p gina Web de *Cervan \TeX* , el Grupo de Usuarios de \TeX Hispanohablantes, localizada en <http://www.cervantex.org/>. Seg n sus promotores, el grupo busca intercambiar experiencias sobre \TeX y promover su uso en Hispanoam rica.



Estructura de un documento L^AT_EX

2.1. L^AT_EX-nica de procesamiento de documentos

Trabajar con L^AT_EX requiere primero crear un archivo de entrada (en inglés, *input file*), o documento fuente, al que se le debe dar (preferiblemente) la extensión **tex**; éste debe ser un archivo de *texto llano* o archivo ASCII¹. El archivo de entrada puede ser creado con *cualquier* editor capaz de almacenar documentos de texto llano, usualmente con la opción ‘guardar como texto’. Muchas implementaciones de T_EX y L^AT_EX tienen sus propios editores de texto incorporados.

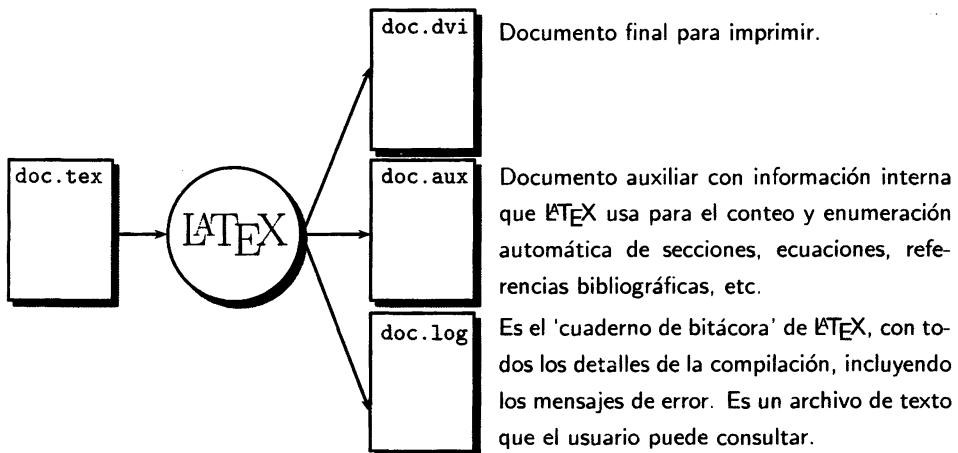


FIGURA 2.1. L^AT_EX en acción.

L^AT_EX compila o procesa o compone (en inglés, *typeset*) el documento de entrada y produce (si el procesamiento es exitoso) un archivo, que tiene el mismo nombre, pero con extensión **dvi**. Este último documento ‘—.dvi’

¹La sigla ASCII proviene de *American Standard Code for Information Interchange*. Los símbolos ASCII constituyen los caracteres que son interpretados de la misma manera por todos los computadores y todos los sistemas operativos.

es el que puede ser visualizado en pantalla o impreso; contiene el texto ya *formateado* junto con la información sobre las fuentes necesarias, pero de tal forma que es *independiente* de las características de la impresora o el monitor utilizados². LATEX crea además dos archivos de control con extensiones **aux** y **log**, respectivamente. La Figura 2.1 resume la LATEX-nica de procesamiento.

El documento de entrada ‘**—.tex**’ se escribe utilizando libremente todos los caracteres del teclado, *excepto* los siguientes once:

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

Estos caracteres tienen un significado *especial* para LATEX y son, por consiguiente, de uso reservado. Entre ellos se destaca el ‘*backslash*’ \ ya que todo comando o instrucción en TEX y en LATEX comienza con dicho símbolo. La Tabla 2.1 muestra, a manera de ejemplo, algunas de las instrucciones requeridas por LATEX para producir ciertos símbolos o expresiones.

Instrucción	Resultado obtenido
\dag	†
\S	§
\copyright	©
\TeX	TEX
\LaTeX	LATEX
\$\pi\$	π
\$\in\$	\in
\$\infty\$	∞
\$\partial\$	∂
\$\aleph\$	\aleph
\$\forall\$	\forall

TABLA 2.1. Algunos comandos LATEX.

LATEX procesa el texto de entrada en uno de tres modos:

- Modo normal o modo de párrafo (en inglés, *paragraph mode*). Es el modo de procesamiento más natural para el usuario: el texto se separa en renglones, párrafos y páginas.

²La extensión **dvi** significa precisamente *device independent*. El archivo ‘**—.dvi**’ es independiente del dispositivo de salida pero requiere de un ‘*driver*’ o manejador para la impresora y de un ‘*previewer*’ o visualizador para el monitor, ambos usualmente incluidos en las implementaciones de TEX y LATEX.

- Modo ID o modo Izquierda-Derecha (en inglés, *LR mode*). Es similar al modo normal, excepto que LATEX escribe indefinidamente de izquierda a derecha y nunca comienza un nuevo renglón. Son relativamente pocos los comandos que hacen que LATEX ingrese al modo ID y lo advertiremos explícitamente en cada caso. Un ejemplo típico son los comandos de la subsección 3.15.1 (página 51) para cajas.
- Modo matemático (en inglés, *math mode*). Modo al cual entra LATEX cuando encuentra el símbolo \$ o algo como \begin{equation}. En tales casos, LATEX procesa el texto que sigue, no como texto normal, sino como símbolos matemáticos. Obsérvense, por ejemplo, las seis últimas instrucciones de la Tabla 2.1. Las numerosas herramientas TEX-nicas para el procesamiento de expresiones matemáticas se presentarán detalladamente en el Capítulo 4.

Al escribir un documento, el usuario tiene que acostumbrarse a la manera peculiar como LATEX (y también TEX) maneja los espacios en blanco, en cualquiera de los tres modos de procesamiento mencionados. LATEX ignora —en el archivo fuente— cualquier espacio que siga a otro espacio; es decir, un espacio es lo mismo que 10 espacios o 100 espacios. Por ejemplo, las dos instrucciones

Esfuerzo	mental
Esfuerzo	mental

producen ambas el mismo resultado: **Esfuerzo mental**. La razón para este comportamiento es muy importante: TEX y LATEX controlan automáticamente el espaciamiento entre símbolos y entre palabras en el documento final. Esto es especialmente útil y eficiente en expresiones matemáticas. No obstante, LATEX tiene instrucciones para añadir e incluso eliminar espacio horizontal (véase la sección 3.7). Por ejemplo, el espacio promedio ocupado por una letra minúscula se obtiene con la instrucción _ donde _ es el resultado de presionar la barra espaciadora. Así, si en el documento fuente escribimos **Esfuerzo__mental**, obtendremos: **Esfuerzo mental**.

Para comenzar un nuevo párrafo se deja una línea en blanco. Pero, de manera análoga a lo que sucede con el espaciamiento horizontal, una línea en blanco es lo mismo que 10 o 100 líneas en blanco: en ambos casos se inicia un nuevo párrafo sin espacio vertical adicional. Los espacios en blanco al comienzo de un párrafo son también ignorados por completo.

Las diferentes instrucciones de LATEX para controlar el espacio vertical se presentarán en el Capítulo 3, especialmente en la sección 3.9.

☞ LATEX ignora el símbolo % y todo lo que aparece a su derecha. Se puede usar entonces % para incluir comentarios personales en el archivo fuente ‘‘.tex’’. Dichos comentarios no aparecerán en el documento final ‘‘.dvi’’.

☞ Para obtener explícitamente los símbolos especiales

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

el usuario debe utilizar comandos LATEX específicos (sección 3.6).

☞ Los errores sintácticos cometidos al escribir el documento fuente ‘‘.tex’’ son parte de la vida diaria de los usuarios de TEX y LATEX. Al encontrar un error, LATEX detiene el procesamiento del archivo de entrada y emite un mensaje de error, el cual no siempre es del todo útil para ubicar exactamente el sitio del error. LATEX-pericia acumulada es la única manera de disminuir el volumen de errores y de interpretar acertadamente los mensajes de error. Información detallada sobre los mensajes de error más frecuentes se encuentra en el Apéndice A.

2.2. Tipos de comandos LATEX

Todo comando LATEX consta del símbolo \ seguido de una secuencia de una o más letras, mayúsculas o minúsculas. Como se explicará en la sección 3.24, el usuario también puede definir sus propios comandos, siempre con la misma restricción: el símbolo \ debe estar seguido de una cadena de letras, sin números ni otros símbolos del teclado.

Podemos distinguir cuatro tipos de instrucciones o comandos:

Comandos simples: son de la forma \comando y se usan principalmente para producir símbolos particulares, como los mostrados en la Tabla 2.1, o efectos directos. El comando \dotfill, por ejemplo, se usa para relleno horizontal con puntos (véase la sección 3.23).

Sobre los comandos simples hay que hacer la siguiente importante advertencia: los espacios en blanco que aparezcan después de las letras de un \comando cualquiera son ignorados por LATEX. Supóngase, por ejemplo, que en el documento de entrada escribimos algo como

\LaTeX tiene muchos usuarios devotos.

Obtendremos:

LATEXtiene muchos usuarios devotos.

El espacio en blanco que sigue al comando \LaTeX se ha perdido; en

tales situaciones debemos escribir `\LaTeX` o `\LaTeX{}` o `\LaTeX__` en lugar de simplemente `\LaTeX`, para que el espacio en blanco siguiente sea tenido en cuenta. Esta misma norma se aplica a todos los comandos que produzcan símbolos o expresiones y que deban ir seguidos de un espacio en blanco.

Comandos con uno o más argumentos: son de la forma

`\comando{...}{...}...{...}`

donde los `{...}` representan los valores concretos asumidos por los argumentos. Por ejemplo, `\underline{...}` es el comando de un argumento utilizado para subrayar texto. Así, `\underline{palabra}` da lugar a palabra (véase la sección 3.11). El comando `\frac{...}{...}` de dos argumentos se utiliza (en modo matemático) para escribir fracciones. Los dos argumentos son, por supuesto, el numerador y el denominador. Así, `$\frac{x}{1+y}$` produce $\frac{x}{1+y}$ (véase la sección 4.7).

Algunos comandos tienen uno o más argumentos de opciones, escritos entre paréntesis angulares `[]`. Por lo general, un argumento opcional precede a los argumentos obligatorios, en la forma

`\comando[opciones]{...}{...}...{...}`

Un ejemplo típico de comando con argumento opcional es

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

presentado en la siguiente sección (sección 2.3).

Declaraciones globales: son comandos simples (es decir, comandos sin argumentos) cuyo alcance se delimita con corchetes o llaves exteriores `{...}`. Por ejemplo, la instrucción `\it{...}` hace que la parte escrita en `...` aparezca en letra cursiva o itálica (véase la sección 3.3 y la Tabla 3.1). Al omitir los corchetes exteriores, una declaración global permanece vigente por el resto del documento, a menos que esté incluida en otro grupo delimitado por llaves `{ }` o en uno de los “entornos” mencionados en el siguiente inciso.

Otra manera de suspender una declaración global se consigue emitiendo una declaración contraria. Así por ejemplo, la declaración `\large` hace que se incremente en lo sucesivo el tamaño de la letra (sección 3.4) pero podemos retornar al tamaño de letra normal escribiendo la declaración `\normalsize`.

Entornos: son construcciones de la forma:

```
\begin{entorno}
:
\end{entorno}
```

Entre los numerosos entornos existentes en LATEX podemos citar, como ejemplos, el entorno **center** para centrar texto u otro material (sección 3.13), el entorno **enumerate** para producir listas enumeradas (sección 3.19), el entorno **tabular** para tablas (Capítulo 5) y el entorno **equation** para desplegar, centrar y numerar ecuaciones (sección 4.1).

Como se señaló antes, las declaraciones globales escritas dentro de un entorno sólo tienen efecto local; es decir, hasta que LATEX encuentra la instrucción **\end{entorno}**. Por ejemplo, si recurrimos al entorno **quote** para presentar una cita (sección 3.12) y queremos que ésta aparezca en letra cursiva o itálica, podemos escribir:

```
\begin{quote}
\it ...
\end{quote}
```

La declaración **\it** para forzar la letra itálica permanece vigente únicamente hasta la aparición de **\end{quote}**.

Un error muy frecuente entre los usuarios de LATEX, tanto principiantes como experimentados, es olvidar escribir **\end{entorno}**, lo cual da lugar a un mensaje de error como el siguiente:

```
\begin{entorno} on input line ... ended by \end{document}.
```

Para evitar contratiempos, se recomienda escribir desde un comienzo las dos instrucciones **\begin{entorno}** y **\end{entorno}** cuando se desee usar un **entorno** particular.

En los argumentos (obligatorios y opcionales) de los comandos no se deben dejar espacios en blanco. Para comenzar un nuevo renglón sin añadir espacio extra se escribe el símbolo **%** en el extremo derecho, antes de iniciar el nuevo renglón. El símbolo **%** hace que LATEX ignore todo lo que está a su derecha, incluyendo los espacios en blanco. De esta forma, una instrucción LATEX puede ocupar varias líneas en el documento fuente. Esta es una situación bastante frecuente; véase como ilustración el tercer ejemplo de la página 11.

2.3. Tipos de documentos LATEX

Cuando LATEX procesa un archivo de entrada espera encontrar ciertas instrucciones básicas que determinan su estructura. Así, cualquier documento LATEX debe empezar con el comando³

\documentclass{...}

Entre las llaves {} se especifica el *estilo* o *clase* de documento que se intenta escribir. LATEX 2ε posee cinco clases básicas⁴, a saber:

- **article** estilo artículo
- **book** estilo libro
- **report** estilo reporte o informe
- **letter** estilo carta
- **slides** estilo transparencias.

En este capítulo presentamos una descripción global de los estilos **article**, **book** y **report**; información más completa sobre el manejo de documentos grandes aparece en el Capítulo 6. Los estilos **slides** y **letter** se consideran detalladamente en el Capítulo 8.

El contenido o *cuerpo del documento* propiamente dicho aparece entre los comandos **\begin{document}** y **\end{document}**:

```
\documentclass[...]{...}
:
\begin{document}
cuerpo del documento  ⇒  {   :
:
\end{document}
```

LATEX ignora todo lo que aparece debajo de la instrucción **\end{document}**.

La instrucción **\documentclass** puede ir acompañada de un argumento de *opciones*, escritas entre paréntesis angulares [], en la forma

\documentclass[opciones]{...}

Las opciones más importantes son las que controlan el tamaño de la letra, el tamaño y la orientación del papel, el número de columnas (1 ó 2) y la

³En LATEX 2.09 y versiones anteriores, el estilo o clase de documento se invocaba con **\documentstyle{...}**.

⁴La American Mathematical Society ha diseñado dos estilos especiales, **amsart** y **amsbook**, que son modificaciones de los estilos **article** y **book**, respectivamente. Estos estilos se describen en la sección 4.35.

impresión en una o en ambas caras del papel. En la Tabla 2.2 aparecen las opciones disponibles y las que LATEX escoge por defecto en ausencia de una declaración explícita.

Ejemplo Los comandos mínimos para un artículo con letra de tamaño 12 puntos:

```
\documentclass[12pt]{article}
\begin{document}
...
\end{document}
```

Ejemplo Los comandos mínimos para un libro a dos columnas, con letra de tamaño 11 puntos, en el que se permita que los diferentes capítulos comiencen en hojas de numeración par o impar (`openany`), y escrito en versión de prueba (`draft`):

```
\documentclass[11pt,twocolumn,openany,draft]{book}
\begin{document}
...
\end{document}
```

Dentro de los paréntesis angulares [], las diferentes opciones se pueden combinar en cualquier orden. Así, también podemos escribir lo anterior en la forma:

```
\documentclass[draft,openany,twocolumn,11pt]{book}
\begin{document}
...
\end{document}
```

Ejemplo Como se señaló al final de la sección 2.2, no se permite dejar espacios en blanco dentro de los argumentos de un comando. Si es necesario iniciar un nuevo renglón, hay que escribir el símbolo % en el extremo derecho para impedir que se añada espacio no deseado. De esta forma, una instrucción puede ocupar dos o más renglones en el documento fuente. Por ejemplo:

```
\documentclass[11pt,twocolumn,openany,notitlepage,draft,%
legalpaper]{book}
\begin{document}
...
\end{document}
```

Característica	Opciones válidas
Tamaño de la letra	<code>10pt 11pt 12pt</code> . Por defecto: <code>10pt</code>
Tamaño del papel	<code>letterpaper</code> (8.5×11 pulgadas, opción por defecto). <code>legalpaper</code> (8.5×14 pulgadas). <code>executivepaper</code> (7.25×10.5 pulgadas). <code>a4paper</code> (21×30 centímetros). <code>a5paper</code> (15×21 centímetros). <code>b5paper</code> (18×25 centímetros).
Orientación del papel	<code>portrait landscape</code> Opción por defecto: <code>portrait</code> (vertical). Con la opción <code>landscape</code> (horizontal) se intercambian el alto y el ancho de todas las páginas.
Número de columnas	<code>onecolumn twocolumn</code> Opción por defecto: <code>onecolumn</code> . En el estilo <code>letter</code> (carta), la opción <code>twocolumn</code> no está disponible.
Página separada para el título	<code>titlepage notitlepage</code> En los estilos <code>book</code> y <code>report</code> el título aparece en una página separada; en el estilo <code>article</code> la opción por defecto es <code>notitlepage</code> .
Impresión en una o dos caras	<code>oneside twoside</code> En el estilo <code>book</code> se usa por defecto <code>twoside</code> : la numeración de las páginas impares aparece a la derecha y la de los pares aparece a la izquierda. En los estilos <code>article</code> y <code>report</code> el formato automático es <code>oneside</code> .
Tipo de impresión	<code>draft final</code> Por defecto: <code>final</code> . La opción <code>draft</code> produce una ‘caja negra’ ■ de advertencia en los sitios en los que se ha excedido el margen derecho.
Primera página de cada capítulo	<code>openright openany</code> En el estilo <code>book</code> la opción por defecto es <code>openright</code> : la primera página de cada capítulo aparece a la derecha, en una página de numeración impar. En el estilo <code>report</code> la opción por defecto es <code>openany</code> . Estas opciones sólo están disponibles en los estilos <code>book</code> y <code>report</code> .

TABLA 2.2. Las opciones de `\documentclass` para los estilos `article`, `book`, `report`, `letter`.

2.4. El preámbulo de un documento

La parte de un documento fuente comprendida entre `\documentclass{...}` y `\begin{document}` se llama preámbulo (en inglés, *preamble*).

preámbulo \Rightarrow $\left\{ \begin{array}{c} \text{\textbackslash documentclass[...]\{...} \\ \\ \text{\begin{document}} \end{array} \right.$

El preámbulo es una “región” importante porque allí se deben colocar aquellos comandos que modifiquen los parámetros establecidos por defecto para un documento. Por ejemplo, para cambiar el tamaño de las márgenes, el espaciamiento entre renglones, el espaciamiento entre párrafos, los encabezados de las páginas, etc, hay que añadir determinadas instrucciones que obliguen a L^AT_EX a modificar el formato estándar declarado en `\documentclass[...]{...}`. Algunas de estas instrucciones solamente pueden aparecer en el preámbulo; de lo contrario, o son ignoradas o no afectan la totalidad del documento o conducen a mensajes de error.

2.5. El estilo **article** (artículo)

La estructura básica de un documento en el estilo `article` es la siguiente:

```
\documentclass[opciones]{article}
\title{...}
\author{...}
\date{...}
\begin{document}
\maketitle
\section{...}
.....
\subsection{...}
.....
\section{...}
.....
\subsection{...}
.....
\end{document}
```

Presentamos en la página 15 una reproducción del archivo `ejem.dvi`, ejemplo concreto de un sencillo documento L^AT_EX escrito (por dos prestigiosos

autores) con el estilo `article`. El documento fuente `ejem.tex` aparece en la Tabla 2.3.

```
\documentclass[10pt]{article}
\title{Un artículo muy aburrido}
\author{Fernando Fernandez Consuegra\thanks{Con el patrocinio de Colciencias.}\And Domingo Dominguez Sinsuegra\thanks{Sin el patrocinio de Colciencias.}}
\date{Enero 15 del 2000}
\begin{document}
\maketitle
\begin{abstract}
Se presentan los resultados de una exhaustiva investigación.
\end{abstract}
Comenzamos por decir que realmente no tenemos mucho por decir ...
\section{Primera sección}
Los temas tratados en esta sección pueden resultar un tanto ...
\subsection{Primera subsección}
El tema tratado aquí, dada su complejidad, amerita una ...
\subsubsection{Primer tema de la subsección}
Este es un tema muy conocido y no lo discutiremos más.
\subsubsection{Segundo tema de la subsección}
Este es un tema muy complicado y lo discutiremos en ...
\section{Segunda sección}
Esta es la segunda sección del presente artículo. Es ...
\end{document}
```

TABLA 2.3. Archivo fuente, `ejem.tex`, del artículo impreso en la página siguiente (Figura 2.2).

Observaciones generales sobre el estilo `article`:

- El comando `\maketitle` hace que se produzcan las líneas para título, autor y fecha; al omitirlo, no aparece ninguna de ellas. Por lo tanto, podemos crear documentos muy sencillos, sin título ni autor, simplemente omitiendo la instrucción `\maketitle`.
- Nótese que `\maketitle` se coloca *después* de `\begin{document}`, a diferencia de `\title`, `\author` y `\date`, que aparecen en el preámbulo.
- LATEX separa los títulos largos en dos o más renglones, pero se puede usar `\And` dentro de `\title{...}` para forzar separaciones en el título.
- Los nombres de dos o más autores se separan con `\and`; para que aparezcan en renglones diferentes se separan con `\And`.

Un artículo muy aburrido

Fernando Fernandez Consuegra*
Domingo Dominguez Sinsuegra†

Enero 15 del 2000

Abstract

Se presentan los resultados de una exhaustiva investigación.

Comenzamos por decir que realmente no tenemos mucho por decir, pero para mayor claridad dividiremos nuestro tema en secciones.

1 Primera sección

Los temas tratados en esta sección pueden resultar un tanto abstrusos. Por consideraciones humanitarias subdividiremos la sección.

1.1 Primera subsección

El tema tratado aquí, dada su complejidad, amerita una división adicional.

1.1.1 Primer tópico de la subsección

Este es un tópico muy conocido y no lo discutiremos más.

1.1.2 Segundo tópico de la subsección

Este es un tópico muy complicado y lo discutiremos en nuestro próximo artículo.

2 Segunda sección

Esta es la segunda sección del presente artículo. Es más interesante que la anterior ya que es la última.

*Con el patrocinio de Colciencias.

†Sin el patrocinio de Colciencias.

FIGURA 2.2. Imagen reducida al 70 % de su tamaño original del archivo ejem.dvi producido con el documento fuente ejem.tex de la Tabla 2.3.

- Se puede utilizar el comando `\author{...}` para escribir afiliaciones de los autores o instituciones, separando cada renglón con `\backslash`.
- Si se omite el comando `\date{...}`, LATEX imprime de todas maneras la fecha del día actual (la fecha vigente en el computador). Para eliminar completamente la fecha se debe escribir `\date{}`.
- La instrucción `\thanks{...}` se puede utilizar en el argumento de los comandos `\author`, `\title` y `\date` para producir notas al pie de página con agradecimientos, direcciones electrónicas u otro tipo de información sobre los autores o el artículo mismo.
- Por defecto, LATEX deja márgenes superior e izquierdo de una pulgada. Para otras opciones establecidas por defecto, véase la Tabla 2.2.
- Por defecto, las páginas aparecen numeradas en la parte inferior y no tienen ningún encabezado en la parte superior. No obstante, el usuario puede incluir encabezados, con la numeración de páginas en la parte superior, por medio del comando `\pagestyle` (véase la sección 2.10). El usuario puede hacer otras modificaciones al formato de página pre-establecido; esto se explica en la sección 2.13. También se puede modificar el formato para los títulos de las secciones (tamaño, tipo de letra, justificación, etc); véase al respecto la sección 8.7.
- Las unidades o divisiones más importantes en el estilo `article` son las secciones, subsecciones y sub-subsecciones, creadas con los comandos `\section{...}`, `\subsection{...}` y `\subsubsection{...}`, respectivamente. Las dos primeras son numeradas automáticamente por LATEX, como se aprecia en el ejemplo de la página anterior, pero podemos controlar esta numeración (véase la sección 6.2). LATEX también permite crear secciones y subsecciones no numeradas por medio de los “comandos estrella” `\section*{...}` y `\subsection*{...}`.
- El entorno ‘`abstract`’ para el resumen del artículo tiene la sintaxis

```
\begin{abstract}
    Texto
\end{abstract}
```

y se debe colocar después de `\maketitle`. El resumen aparece en letra más pequeña, inmediatamente antes del texto del artículo, o en una página separada si se usa la opción `titlepage` (Tabla 2.2). Podemos hacer que LATEX imprima la expresión ‘Resumen’, en vez de ‘Abstract’, usando ya sea el paquete `babel` (sección 2.9) o instrucciones específicas (véase la sección 6.5).

2.6. El estilo book (libro)

En el estilo **book** se dispone del comando `\chapter{...}`, para capítulos. La estructura básica de un documento en el estilo **book** es la siguiente:

```
\documentclass[opciones]{book}
\title{...}
\author{...}
\date{...}
\begin{document}
\maketitle
\chapter{...}
.....
\section{...}
.....
\chapter{...}
.....
\section{...}
.....
\end{document}
```

Observaciones generales sobre el estilo **book**:

- El comando `\maketitle` hace que se produzcan, en una página separada, las líneas para título, autor y fecha; al omitirlo, no se imprime la página del título.
- L^AT_EX imprime, en la página del título, la fecha del día actual (vigente en el computador) aun si no se usa el comando `\date{...}`. Para eliminar completamente la fecha se debe escribir `\date{}`.
- Por defecto, cada capítulo comienza en una página de numeración impar (a mano derecha), a menos que se use la opción `openany` (véase la Tabla 2.2), y se generan encabezados con los títulos de los capítulos (a mano izquierda) y de las secciones (a mano derecha). Tales encabezados se pueden suprimir o modificar (véase la sección 2.10). Los formatos para los títulos de secciones y capítulos también se pueden modificar; véase al respecto la sección 8.7.
- Los capítulos se numeran automáticamente pero es posible crear capítulos no numerados por medio del comando estrella `\chapter*{...}`.
- El entorno `abstract` no está disponible en el estilo **book**.
- L^AT_EX tiene herramientas especiales para manipular eficientemente documentos grandes y mecanismos para la generación automática de tablas de contenido, índices y bibliografía (véase el Capítulo 6).

2.7. El estilo **report** (reporte o informe)

El estilo **report** tiene la misma estructura del estilo **book**, con las siguientes diferencias (véase también la Tabla 2.2):

- El estilo **report** está diseñado para impresión a una sola cara (opción **oneside**).
- Los capítulos pueden comenzar en páginas de numeración par o impar (opción **openany**).
- Los números de las páginas aparecen centrados en la parte inferior y no hay encabezados en la parte superior, aunque éstos se pueden forzar usando la instrucción **\pagestyle** (sección 2.10).
- El entorno **abstract** sí está disponible; el ‘**abstract**’ o resumen se imprime en una página independiente, no numerada, adicional a la página del título.

2.8. El uso de paquetes

Un **paquete** (en inglés, *package*) es un conjunto de macros (es decir, instrucciones **TEX** o **LATEX**) diseñado para simplificar las tareas de edición o agregar nuevos recursos. Para acceder a un **paquete** determinado se coloca la instrucción

\usepackage{paquete}

en el preámbulo del documento (sección 2.4). Los paquetes propiamente dichos son archivos con extensión **sty**; muchos de ellos tienen el aval del Proyecto **LATEX3** y se incluyen en las distribuciones estandarizadas de **LATEX 2 ε** . Tales paquetes vienen pre-instalados en la mayoría de las implementaciones de **TEX** y **LATEX** y en lo sucesivo nos referiremos a ellos como “paquetes estándares de **LATEX 2 ε** ”.

Algunos paquetes son muy populares y tienen un status “semi-oficial” pero no vienen instalados en implementaciones corrientes. El usuario debe aprender a instalarlos, lo cual depende tanto de la plataforma computacional utilizada como de la implementación local. En el apéndice C se presenta una guía para la instalación de paquetes.

Para saber si un **paquete** determinado está o no localmente instalado, basta colocar en el preámbulo del documento fuente la instrucción **\usepackage{paquete}**. Si **LATEX** no puede acceder al **paquete** en cuestión, emitirá el mensaje de error: **File ‘paquete.sty’ not found**.

Algunos de los paquetes que se describirán a lo largo de este libro son:

amsmath. Paquete que contiene las herramientas T_EX-nicas más sofisticadas para el manejo de expresiones matemáticas. Diseñado bajo el auspicio de la *American Mathematical Society*, es la versión actualizada para L_AT_EX 2_ε de los macros anteriormente distribuidos como *AMS-TEX* y *AMS-LATEX*. Se describe en el Capítulo 4.

graphicx. Paquete estándar de L_AT_EX 2_ε para transformaciones geométricas de objetos (aumento a escala, rotaciones y reflexiones) y para la inclusión en documentos L_AT_EX de gráficas creadas con software externo. Se describe en la sección 7.3.

color. Paquete estándar de L_AT_EX 2_ε que permite la inclusión de textos en color o textos sobre fondos de color. Se describe en la sección 7.2.

inputenc. Paquete estándar de L_AT_EX 2_ε que permite el uso directo de caracteres acentuados (como á, é, ñ, ö) en documentos L_AT_EX. Se describe en la sección 3.1.

babel. Paquete estándar de L_AT_EX 2_ε para generar documentos en idiomas diferentes del inglés; se describe en la siguiente sección.

Ejemplo Los comandos mínimos para un artículo con letra de tamaño 12 pt, en el que se cargan los paquetes **amsmath**, **color** y **graphicx**:

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\usepackage{color}
\usepackage{graphicx}
\begin{document}
.....
\end{document}
```

También se puede lograr lo anterior escribiendo una sola vez la instrucción **\usepackage{...};** los diferentes paquetes se separan con comas:

```
\documentclass[12pt]{article}
\usepackage{amsmath,color,graphicx}
\begin{document}
.....
\end{document}
```



Todos los paquetes mencionados o descritos en el presente libro hacen parte de las distribuciones de L_AT_EX 2_ε o, en caso contrario, se incluyen en el CD adjunto, en la carpeta /Paquetes.

2.9. El paquete **babel**

Este paquete **babel**⁵ ayuda a generar documentos LATEX en idiomas diferentes al inglés. Una vez se especifica un idioma, **babel** utiliza los patrones de partición silábica (en inglés, *hyphenation patterns*) del idioma declarado para dividir correctamente las palabras en el margen derecho⁶. Los títulos en el interior del documento (como ‘Resumen’, ‘Capítulo’, ‘Bibliografía’, etc) aparecen en el idioma escogido. El idioma o idiomas deseados en el documento se incluyen como opciones de `\usepackage{babel}`, en la forma `\usepackage[idiomas]{babel}`; los idiomas se separan con comas. Prácticamente todos los idiomas europeos están actualmente disponibles en el ambiente **babel**. Algunas opciones son: `spanish`, `portuguese`, `french`, `german`, `italian`, `greek`, `catalan`, `esperanto`.

Ejemplo Los comandos mínimos para un libro escrito en español, con acceso a los paquetes `amsmath` y `babel`:

```
\documentclass{book}
\usepackage{amsmath}
\usepackage[spanish]{babel}
\begin{document}
.....
\end{document}
```

En un mismo documento se pueden declarar varios idiomas; esto último es útil si, por ejemplo, en el documento aparecen citas extensas en diferentes idiomas. El *último* idioma declarado es el idioma principal del documento. Para indicarle a **babel** el cambio de idioma dentro de un mismo documento se usa la instrucción `\selectlanguage{...}`, instrucción que se puede usar múltiples veces en el documento.

Ejemplo Para acceder a los idiomas español y alemán en un mismo documento (estilo `book`) escribimos:

```
\documentclass{book}
\usepackage[german,spanish]{babel}
\begin{document}
.....
\end{document}
```

El idioma español (`spanish`) es el idioma principal del documento por ser el último declarado al cargar `babel`. Para cambiar al idioma alemán escribimos `\selectlanguage{german}`.

⁵Paquete estándar de LATEX 2 ε , escrito por Johannes Braams.

⁶Véase la ‘Nota TEX-nica’ al final de esta sección.

En caso de duda, el comando `\languagename` permite saber cuál es el idioma actualmente vigente en el documento; el nombre del idioma aparecerá escrito en el documento ‘`—.dvi`’.

La opción `spanish` del paquete `babel`. Cuando se ejecuta la instrucción `\usepackage[spanish]{babel}`, `babel` consulta los archivos `spanish.sty` y `spanish.ldf`⁷. El contenido y mantenimiento de la versión española están a cargo del grupo *CervanTeX* (Capítulo 1). A finales del año 2000, los citados archivos `spanish` sufrieron extensos cambios que afectan el aspecto final de documentos *LATEX*, más allá de cuestiones puramente gramaticales o lingüísticas. A lo largo del presente libro llamaremos la atención del lector sobre los cambios estructurales introducidos por el uso de la opción `spanish` de `babel`, con respecto a los formatos estándares de *LATEX*.

Nota *TeX-nica*. Los patrones de partición en sílabas de los diferentes idiomas utilizados por `babel` son archivos con extensión `tex`. Las implementaciones de *LATEX* se distribuyen con patrones para algunos idiomas, pero no basta cargar `babel` por medio de `\usepackage` para tener acceso a esos patrones. *LATEX* debe ser re-configurado corriendo de nuevo *INITeX*; los detalles de este procedimiento se explican en el Apéndice B.

③ En la carpeta `/Babel/hyphen/` del CD adjunto se incluyen los patrones de partición silábica para una extensa colección de idiomas. La implementación *MiKTeX* para *Windows* tiene ya instalados esos archivos, pero para que `babel` tenga acceso a los idiomas deseados, hay que activar los nombres en la plantilla `Languages` de *MiKTeX Options* y actualizar (‘Refresh Now’) la base de datos en la plantilla `General`. Procediendo de esta manera, *MiKTeX* automáticamente re-configura *LATEX* con *INITeX*.

2.10. El formato de página

LATEX divide la página en sectores, como se muestra en la Figura 2.3. Las dimensiones de cada una de las regiones dependen del estilo de documento (`book`, `article`, etc); el contenido de los encabezados y el pie de página se controlan con `\pagestyle{...}`. Los márgenes superior e interior son, por defecto, de una pulgada (más un pequeño espacio extra que depende de cada estilo⁸). Las notas al pie de página (sección 3.17) *no* aparecen en el sector ‘Pie de página’ sino en la parte inferior del cuerpo del documento.

⁷ Algo similar ocurre al invocar otro idioma; la extensión `ldf` es una abreviación de *language definition file*.

⁸ Véase al respecto la sección 2.13, en la que se explica, además, cómo el usuario puede modificar las dimensiones de los parámetros que controlan el formato de página.

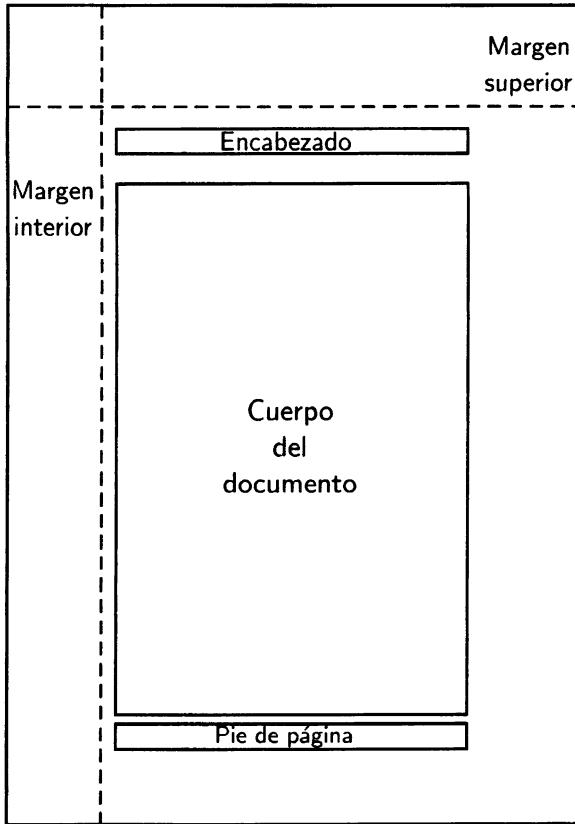


FIGURA 2.3. Formato básico de página.

El comando `\pagestyle{formato}` tiene la siguiente sintaxis:

`\pagestyle{formato}`

donde el *formato* puede ser uno de los siguientes:

- `empty`
- `plain`
- `headings`
- `myheadings`

Las características de estos formatos se presentan a continuación.

empty. Es el formato más simple: no hay encabezados ni páginas numeradas (*LATEX* realiza el conteo de las páginas pero no imprime los números).

plain. No hay encabezados y los números de las páginas aparecen centrados en el sector ‘Pie de página’. En los estilos `article` y `report` es el formato

escogido por defecto (en ausencia de `\pagestyle`).

headings. En el estilo `book` es el formato escogido por defecto (en ausencia de `\pagestyle`). Genera encabezados con los números de las páginas y con información adicional que depende del estilo del documento, a saber:

- En el estilo `article`, con la opción de impresión a una cara (opción por defecto), los encabezados contienen títulos de secciones, determinados por los comandos `\section{...}`. Pero si se usa la opción de impresión a dos caras (`twoside`), los encabezados son títulos de secciones a mano izquierda y títulos de subsecciones a mano derecha.
- En el estilo `book` los encabezados de las páginas de numeración par (a mano izquierda) contienen el título del capítulo actual y las páginas de numeración impar (a mano derecha) contienen títulos de secciones, determinados por los comandos `\section{...}`.
- El estilo `report` establece por defecto impresión a una sola cara y los encabezados contienen títulos de capítulos, no de secciones.

myheadings. Genera encabezados con los números de las páginas y con el texto que el usuario escoge por medio de las instrucciones `\markboth` o `\markright`, cuya sintaxis es:

```
\markboth{encabezado a izquierda}{encabezado a derecha}
\markright{encabezado a derecha}
```

l^AT_EX utiliza los encabezados para todas las páginas, a partir de aquélla en la que se encuentra alguna de estas dos instrucciones. El comando `\markright{...}` se usa principalmente para documentos con la opción de impresión a una cara, caso en el cual las páginas se consideran como páginas a mano derecha. En realidad, `\markright{...}` tiene el mismo efecto que `\markboth{}{...}`.

Las instrucciones `\markboth` y `\markright` pueden aparecer múltiples veces en un documento, pero el sitio más apropiado para colocarlas es inmediatamente después de comandos divisionales como `\chapter` o `\section`.

Ejemplo Los comandos que aparecen a continuación corresponden a un documento escrito en el estilo `article`, tamaño de 11 puntos, para impresión a dos caras, y cuyos encabezados contienen a mano derecha el título del

artículo y a mano izquierda su autor. Estos encabezados difieren de los que se obtendrían por defecto usando simplemente `\pagestyle{headings}`. En este documento se accede además a los paquetes `graphicx` y `amsmath`.

```
\documentclass[11pt,twoside]{article}
\usepackage{graphicx,amsmath}
\pagestyle{myheadings}
\markboth{Autor del artículo}{Título del artículo}
\begin{document}
...
\end{document}
```

A continuación detallamos importantes aspectos sobre el formato de página y los encabezados.

Formato de una página aislada. Cuando se desea cambiar el formato de una página aislada es mejor usar la instrucción

`\thispagestyle{formato}`

donde las opciones de `formato` son las mismas cuatro ya explicadas: `empty`, `plain`, `headings` o `myheadings`.

Páginas no numeradas. Es posible que la primera página de un documento aparezca numerada incluso si se ha usado la instrucción `\pagestyle{empty}`. La explicación de este fenómeno radica en el hecho de que el comando `\maketitle` emite a su vez el comando `\pagestyle{plain}`, forzando el número en la primera página. Por consiguiente, para eliminar por completo la numeración en todas las páginas del documento se debe escribir `\pagestyle{empty}` en el preámbulo y `\thispagestyle{empty}` inmediatamente después de `\maketitle`.

Comandos estrella. Los comandos `\chapter*{...}`, `\section*{...}` y `\subsection*{...}` no son tenidos en cuenta por LATEX para los encabezados.

Encabezados demasiado largos. Sucele a menudo que algunos títulos —de capítulos, secciones o de todo el documento— son demasiado largos para ser usados como encabezados. O, simplemente, el usuario desea que el encabezado difiera del título real. En tales casos el usuario puede indicarle a LATEX las abreviaciones deseadas para reemplazar en los encabezados los títulos reales; esto se hace por medio

del argumento opcional que tienen los comandos `\chapter`, `\section` y `\subsection`, en la forma

```
\chapter[título modificado]{título real}
\section[título modificado]{título real}
\subsection[título modificado]{título real}
```

La expresión *título modificado* aparece en la tabla de contenido o índice general del documento (véase la sección 6.6).

Encabezados vacíos. Si un libro está dividido en capítulos pero éstos no tienen secciones, los encabezados generados por L^AT_EX con la instrucción `\pagestyle{headings}` son vacíos en las páginas a mano derecha (excepto por los números de las páginas). Esta anomalía también se presenta en el estilo `article`, impreso a dos caras, si el artículo no contiene subsecciones. En tales situaciones el usuario puede optar por incluir manualmente los encabezados con alguno de los comandos `\markright` o `\markboth`. Esto se ilustra en el siguiente ejemplo.

Ejemplo Supóngase que tenemos un artículo que contiene varias secciones pero no subsecciones; como se explicó arriba, los encabezados a mano derecha aparecerán vacíos si se imprime a dos caras con `\pagestyle{headings}`. Para modificar tal formato, podemos optar por incluir encabezados con el título del artículo a mano izquierda y con el título de la sección actual a mano derecha. Esto se consigue incluyendo una instrucción `\markboth` después de cada comando `\section{...}`, como se ilustra a continuación.

```
\documentclass[12pt,twoside]{article}
\pagestyle{myheadings}
\begin{document}
.....
\section{título de la primera sección%}
\markboth{título del artículo}{título de la primera sección}
.....
\section{título de la segunda sección%}
\markboth{título del artículo}{título de la segunda sección}
.....
\end{document}
```



El paquete `fancyhdr`, incluido en el disco adjunto, permite crear encabezados o pies de página “vistosos” en los que aparezcan, por ejemplo, líneas horizontales decorativas. Este paquete, muy popular y confiable, se describe en la sección 13.1. Un paquete relacionado es `fncychap`, el cual permite modificar la presentación de los capítulos (para los estilos `book` y `report`); se describe en la sección 13.3.

2.11. Unidades de medida y longitudes en LATEX

LATEX reconoce sólo ciertas unidades de medida: las unidades básicas de TEX. Todas las dimensiones se deben presentar en las unidades de la Tabla 2.4. LATEX acepta dimensiones tanto positivas como negativas; por ejemplo, `1.5cm`, `8pt`, `-0.3cm`, `14mm`, `-2em`. En lugar del punto decimal se puede escribir una coma; así, `1.5cm` y `1,5cm` son dimensiones correctas. Sin embargo, `0` no es una dimensión legítima ya que no contiene unidades de medida; la longitud nula se debe especificar con `0pt` o `0cm`.

Unidades TeX	Abreviatura	Significado
Pulgadas	<code>in</code>	Usual
Centímetros	<code>cm</code>	Usual
Milímetros	<code>mm</code>	Usual
Puntos	<code>pt</code>	<code>1pt</code> = ancho de un punto ≈ 0.35 mm
Picas	<code>pc</code>	<code>1pc</code> = 12 puntos
Eines	<code>em</code>	<code>1em</code> = ancho de una 'M' en la fuente vigente
Equis	<code>ex</code>	<code>1ex</code> = altura de una 'x' en la fuente vigente

TABLA 2.4. Unidades de medida en TEX y LATEX.

La unidad `pt` es muy usada en LATEX ya que permite expresar distancias, tanto pequeñas como grandes, sin usar decimales. `1 pt` equivale a `0.035146 cm`; la siguiente igualdad es muy útil como referencia:

$$1 \text{ cm} = 28.5 \text{ pt}$$

Ciertos comandos de LATEX, llamados *parámetros de longitud*, controlan longitudes concretas, tales como la anchura de los márgenes o la separación entre los párrafos. Para establecer o cambiar las longitudes determinadas por dichos parámetros se usa la instrucción `\setlength`. Concretamente, por medio de

`\setlength{\paramt}{longitud}`

el parámetro de longitud `\paramt` adquiere la *longitud* indicada. En la sección 2.13 se ilustra el uso de `\setlength` para la modificación de los parámetros que controlan el formato de página.

Algunos parámetros de TEX y LATEX utilizan *longitudes elásticas*, las cuales se pueden expandir o contraer una determinada magnitud, a diferencia de las normales, también llamadas *longitudes rígidas*. Para mayores detalles sobre este tópico, véase la sección 8.5.

2.12. Numeración de las páginas

Por defecto, L^AT_EX numera las páginas usando números arábigos 1, 2, ..., pero el usuario puede utilizar otros tipos de numeración por medio del comando

```
\pagenumbering{estilo de numeración}
```

Los *estilos de numeración* disponibles son:

`\pagenumbering{arabic}`. Se numeran las páginas con números arábigos; numeración utilizada por defecto.

`\pagenumbering{roman}`. Se numeran las páginas con números romanos en minúscula: i, ii, iii, iv, Si se usa el paquete `babel`, opción `spanish` (versión posterior al año 2000) la numeración obtenida es I, II, III,... (véase al respecto la nota T_EX-nica al final de esta sección).

`\pagenumbering{Roman}`. Se numeran las páginas con números romanos en mayúscula: I, II, III, IV, ...

`\pagenumbering{alph}`. Se numeran las páginas con letras minúsculas: a, b, c, d, ...

`\pagenumbering{Alph}`. Se numeran las páginas con letras mayúsculas: A, B, C, D, ...

En un mismo documento se pueden usar varios `\pagenumbering{...}` para cambiar el tipo de numeración; sin embargo, cada vez que se use, el conteo de páginas se reinicia en 1.

L^AT_EX lleva el conteo de las páginas por medio de un *contador* (en inglés, *counter*), que puede ser controlado o modificado por el usuario (véase la sección 3.25).

Nota T_EX-nica. En versiones del paquete `babel`, opción `spanish`, posteriores al año 2000, la numeración romana minúscula i, ii, iii, ... es reemplazada por la numeración en versalitas I, II, III, ..., en todas las instancias en las que L^AT_EX recurra al estilo-contador `roman`. Para restituir la numeración romana al estilo estándar de L^AT_EX hay que agregar las siguientes instrucciones en el documento fuente, después de `\begin{document}`:

```
\makeatletter
\def\@roman{\romannumeral #1}
\makeatother
```

Sobre el significado de las instrucciones `\makeatletter` y `\makeatother`, véase la sección 8.6.

2.13. Modificaciones del formato de página

Esta es una sección de referencia para los lectores interesados en hacer modificaciones a los formatos de página pre-establecidos por defecto con las declaraciones `\documentclass{...}` y `\pagestyle{...}`.

Las dimensiones que determinan el formato de página están controladas por varios parámetros (Figura 2.4) que adquieren por defecto ciertos valores, según el estilo declarado en `\documentclass{...}`. En la Tabla 2.5 aparecen los valores por defecto para los estilos `article`, `book` y `report`, y para los tamaños de letra 10 pt, 11 pt y 12 pt. Estos parámetros se pueden modificar, *desde el preámbulo del documento*, por medio de instrucciones apropiadas, tal como lo explicaremos en la presente sección.

Parámetro	Impresión a dos caras			Impresión a una cara		
	10pt	11pt	12pt	10pt	11pt	12pt
<code>\textwidth</code>	12.1 cm	12.6 cm	13.7 cm	12.1 cm	12.6 cm	13.7 cm
<code>\textheight</code>	19.3 cm	19 cm	19.2 cm	19.3 cm	19 cm	19.2 cm
<code>\topmargin</code>	9 mm	9 mm	9 mm	9 mm	9 mm	9 mm
<code>\oddsidemargin</code>	1.5 cm	1.3 cm	7 mm	2.2 cm	1.9 cm	1.4 cm
<code>\evensidemargin</code>	2.9 cm	2.6 cm	2 cm	2.2 cm	1.9 cm	1.4 cm
<code>\headheight</code>	4 mm	4 mm	4 mm	4 mm	4 mm	4 mm
<code>\headsep</code>	9 mm	9 mm	9 mm	9 mm	9 mm	9 mm
<code>\footskip</code>	1 cm	1 cm	1 cm	1 cm	1 cm	1 cm
<code>\marginparwidth</code>	3.7 cm	3.5 cm	3 cm	3.2 cm	3 cm	2.4 cm
<code>\marginparsep</code>	4 mm	3.5 mm	3.5 mm	4 mm	3.5 mm	3.5 mm

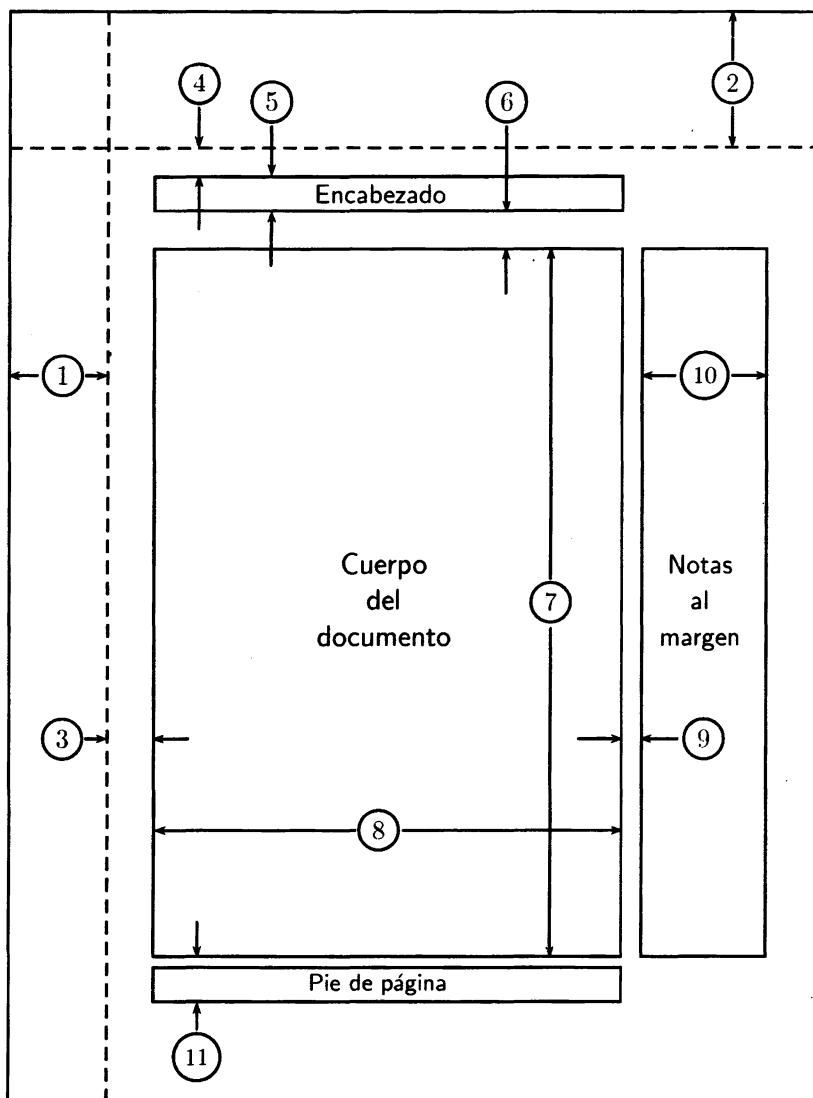
TABLA 2.5. Valores por defecto de los parámetros que controlan el formato de página (tamaño carta) en los estilos `article`, `book` y `report`.

2.13.1. El paquete `layout`

El paquete `layout`⁹ se puede utilizar para ver los valores de los parámetros que controlan el formato de página del documento que se está procesando. Una vez cargado el paquete, con `\usepackage{layout}`, podemos escribir la instrucción `\layout` en cualquier sitio, después de `\begin{document}`. El archivo ‘‘.dvi’’ mostrará, en una página separada, un diagrama enteramente similar al de la Figura 2.4, con los valores exactos de los 11 parámetros. Tal esquema le ayudará al usuario en las decisiones sobre las modificaciones deseadas.

Los textos del diagrama se pueden obtener en varios idiomas. Por ejemplo, si se carga el paquete en la forma `\usepackage[spanish]{layout}`,

⁹Paquete estándar de LATEX 2 ε , escrito por Kent McPherson.



- | | |
|--------------------------|-------------------|
| ① una pulgada + \hoffset | ⑥ \headsep |
| ② una pulgada + \voffset | ⑦ \textheight |
| ③ \oddsidemargin | ⑧ \textrwidth |
| ③ \evensidemargin | ⑨ \marginparsep |
| ④ \topmargin | ⑩ \marginparwidth |
| ⑤ \headheight | ⑪ \footskip |

FIGURA 2.4. Parámetros que controlan el formato de página.

los textos se obtendrán en español. Otras opciones son: `english` (opción por defecto), `french` y `portuguese`. Las dimensiones aparecerán en unidades `pt` o ‘puntos’ (sección 2.11).

2.13.2. Modificación de los parámetros

El margen interior (① en la Figura 2.4) y el margen superior (② en la Figura 2.4) son, por defecto, de 1 pulgada en *todos* los documentos LATEX, pero se pueden agrandar o reducir por medio de los comandos primarios `TEX`, `\hoffset` y `\voffset`.

\hoffset Corrección horizontal; debe usarse solamente en el preámbulo del documento y puede ser positiva o negativa. Afecta a todas las páginas del documento.

Ejemplo La instrucción `\hoffset0.5in`, colocada en el preámbulo del documento, establece un margen interior de 1 pulgada + 0.5 pulgada = 1.5 pulgadas, para todas las páginas del documento.

Ejemplo La instrucción `\hoffset-1cm`, colocada en el preámbulo del documento, establece un margen interior de 1 pulgada - 1 cm = 1.54 cm, para todas las páginas del documento.

\voffset Corrección vertical; debe usarse solamente en el preámbulo del documento y puede ser positiva o negativa. Afecta a todas las páginas del documento.

Ejemplo La instrucción `\voffset-0.5in`, colocada en el preámbulo del documento, establece un margen superior de 1 pulgada - 0.5 pulgada = $\frac{1}{2}$ pulgada, para todas las páginas del documento.

Los parámetros ③ a ⑪ mostrados en la Figura 2.4 se pueden modificar con la instrucción `\setlength` (sección 2.11), como se indica a continuación. Advertimos al lector que estos cambios solamente se pueden efectuar en el preámbulo y afectan a todas las páginas del documento.

\textwidth El ancho del cuerpo del documento.

Ejemplo `\setlength{\textwidth}{13cm}` establece un ancho de 13 cm para el cuerpo del documento.

\textheight La altura del cuerpo del documento (sin el encabezado ni el pie de página).

Ejemplo `\setlength{\textheight}{20cm}` establece una altura de 20 cm para el cuerpo del documento.

\topmargin Espacio extra añadido encima del encabezado (④ en la Figura 2.4).

Ejemplo `\setlength{\topmargin}{5mm}` establece un espacio extra de 5 mm encima del encabezado, lo cual es una reducción del espacio establecido por defecto (Tabla 2.5). Esto quiere decir que el margen real en la parte superior de la página es 1 pulgada + 5 mm (sin contar el efecto de `\voffset`).

Ejemplo `\setlength{\oddsidemargin}{0pt}` elimina completamente el espacio extra encima del encabezado; esto quiere decir que el margen real en la parte superior de la página es de exactamente 1 pulgada (sin contar el efecto de `\voffset`).

\oddsidemargin Espacio extra añadido en la parte lateral del cuerpo del documento (③ en la Figura 2.4). Si se escoge impresión a dos caras (`twoside`) afecta solamente las páginas de numeración impar; de lo contrario, afecta a todas las páginas.

Ejemplo `\setlength{\oddsidemargin}{2mm}` establece un espacio extra de 2 mm, lo cual es una reducción del espacio establecido por defecto (Tabla 2.5). Esto quiere decir que el margen real a la izquierda, en las páginas de numeración impar, es 1 pulgada + 2 mm (sin contar el efecto de `\hoffset`).

Ejemplo `\setlength{\oddsidemargin}{0pt}` elimina completamente el espacio extra; esto quiere decir que el margen real a la izquierda, en las páginas de numeración impar, es de exactamente 1 pulgada (sin contar el efecto de `\hoffset`).

\evensidemargin Espacio extra añadido en la parte lateral del cuerpo del documento (③ en la Figura 2.4). Si se escoge impresión a dos caras (`twoside`) afecta solamente las páginas de numeración par; de lo contrario, afecta a todas las páginas.

Ejemplo `\setlength{\evensidemargin}{0pt}` elimina completamente el espacio extra; esto quiere decir que el margen lateral en las páginas de numeración par es de exactamente 1 pulgada (sin contar el efecto de `\hoffset`).

\headsep Espacio entre la base del encabezado y el tope del cuerpo del documento (⑥ en la Figura 2.4).

Ejemplo `\setlength{\headsep}{1cm}` establece una separación de 1 cm entre la base del encabezado y el tope del cuerpo del documento. Puesto que el valor por defecto de este parámetro es de 9 mm (Tabla 2.5), la separación se amplía ligeramente.

\headheight Altura del encabezado.

Ejemplo `\setlength{\headheight}{6mm}` establece una altura de 6 mm para el encabezado, que es ligeramente mayor que la altura de 4 mm establecida por defecto (Tabla 2.5).

\footskip Distancia vertical entre la base del cuerpo del documento y la base del pie de página (⑪ en la Figura 2.4).

Ejemplo `\setlength{\footskip}{1.5cm}` establece una distancia vertical de 1.5 cm entre la base del cuerpo del documento y la base del pie de página, que es mayor que la distancia de 1 cm establecida por defecto (Tabla 2.5).

\marginparwidth Controla el ancho de las notas marginales (⑩ en la Figura 2.4). Véase también la sección 3.18.

Ejemplo `\setlength{\marginparwidth}{2.2cm}` establece un ancho de 2.2 cm para todas las notas marginales.

\marginparsep Controla la distancia que separa las notas marginales del texto principal (⑨ en la Figura 2.4). Véase también la sección 3.18.

Ejemplo `\setlength{\marginparsep}{5mm}` establece una distancia de 5 mm entre el texto principal y las notas marginales.

2.14. Páginas con una o dos columnas

Una de las opciones de `\documentclass` es `twocolumn` (Tabla 2.2), para documentos escritos a dos columnas. No obstante, para este tipo de documentos se recomienda emplear el paquete `multicol` (descrito en la sección 13.2), que brinda recursos adicionales.

El comando `\twocolumn` permite incluir páginas a dos columnas en documentos normales. Al respecto, hay que tener presente lo siguiente:

- Al encontrar la instrucción `\twocolumn`, LATEX inicia siempre una nueva página, escrita a dos columnas. El formato a dos columnas permanece vigente hasta la aparición de la instrucción `\onecolumn`, la cual hace que LATEX termine la página actual e inicie una nueva, en el formato de una columna. Esto significa, en particular, que en una misma página no se pueden combinar los formatos a una y dos columnas (lo que sí es posible con el paquete `multicol`).
- El comando `\twocolumn` tiene un argumento opcional: con la instrucción `\twocolumn[texto]`, LATEX inicia una nueva página, escrita a dos columnas, colocando el *texto* en el primer renglón y a todo lo ancho de la página.



Herramientas de edición básicas

En este capítulo presentamos las herramientas básicas de edición que L^AT_EX posee para el manejo de textos.

3.1. Tildes y acentos

Símbolo	Instrucción	Símbolo	Instrucción	Símbolo	Instrucción
á	\'a	à	\`a	œ	\oe
é	\'e	â	\^a	(E	\OE
í	\'{\i}	ä	\\"a	æ	\ae
ó	\'o	ă	\u{a}	Æ	\AE
ú	\'u	ă	\v{a}	ø	\o
ñ	\~n	á	\r{a}	Ø	\O
Ñ	\~N	à	\.{a}	ß	\ss
í	\i	ä	\d{a}	ł	\l
j	\j	ã	\H{a}	Ł	\L
¿	?‘	ā	\^{\{a}}	ő	\t{\oo}
¡	!‘	ā	\={a}		
		a	\b{a}		
		ą	\c{a}		

Los diferentes acentos exhibidos en la tabla anterior sobre la letra a (á, à, â, ä, ...), se pueden aplicar a cualquier otra letra, mayúscula o minúscula. Obsérvese que para lograr la i con tilde (í) primero eliminamos el punto de la i (con \i) y luego acentuamos este último símbolo en la forma \'{\i}. Análogamente, j se obtiene a partir de \^{\j}.

Dependiendo de la plataforma (*Windows*, Linux, Macintosh, etc) y de la codificación local de fuentes, símbolos como á, é, í, ó, ú, ñ, ö, ç, æ se pueden obtener directamente del teclado (adecuadamente configurado) o por medio de códigos específicos. Estos símbolos se pueden usar libre y directamente

en documentos L^AT_EX si en el preámbulo del documento fuente ‘—.tex’ se carga el paquete `inputenc` con la opción `latin1`, en la forma:

```
\usepackage[latin1]{inputenc}
```

`inputenc` es un paquete estándar de L^AT_EX 2_E; algunas de sus opciones son:

`latin1` codificación para idiomas de Europa occidental.

`latin2` codificación para idiomas de Europa oriental.

`latin3` codificación para Catalán, Gallego y Esperanto.

Si se carga la opción `spanish` del paquete `babel` (sección 2.9) en la forma

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

se pueden usar en documentos L^AT_EX las siguientes abreviaciones para los acentos del castellano:

Entrada	Salida
'a 'e 'i 'o 'u "u	á é í ó ú ü
'A 'E 'I 'O 'U "U 'N	Á É Í Ó Ú Ü Ñ

3.2. Puntuación

Puntos. Después del punto final de una oración, L^AT_EX deja un espacio mayor que el espacio usual entre palabras (excepto si el punto final está precedido por una letra mayúscula). Pero hay puntos que no representan el final de una oración; en tales casos se debe usar `. \` en vez del simple punto para lograr el **espaciamiento correcto**.

Ejemplo

Entrada	Salida
Dr.\ Pasteur	Dr. Pasteur
Univ.\ Estatal	Univ. Estatal
Pág.\ 321 del Vol.\ 6	Pág. 321 del Vol. 6

Un punto precedido por una letra mayúscula no es interpretado por L^AT_EX como el final de una oración sino como el punto de una abreviatura. Si realmente es un punto final, es necesario añadir `\@` antes del punto para lograr el espaciamiento adicional entre oraciones.

Ejemplo El rey habló ante la ONU. Al día siguiente abdicó.

El rey habló ante la ONU\@. Al día siguiente abdicó.

Puntos suspensivos. Las instrucciones `\dots` y `\ldots` producen ... que tiene un mejor aspecto que los tres simples puntos ...

Guiones. LATEX distingue tres tipos de guiones:

- En palabras compuestas: no-lineal, pseudo-métrica.
- En rangos: Isaac Newton (1642–1727), páginas 58–63.
- Como signos de puntuación: Riemann —sucesor de Gauss y Dirichlet en Göttingen— murió a los cuarenta años de edad.

Los tres tipos de guiones se obtienen por medio de “ligaduras”, en la siguiente forma¹:

Entrada	Salida
-	-
--	--
---	---

Comillas. Las ‘comillas simples’ se obtienen con las teclas ‘ y ’, respectivamente. Para las comillas “inglesas” hay que usar dos veces la tecla ‘ (para abrir las) y dos veces la tecla ’ (para cerrarlas)¹. Tenemos que evitar escribir palabras “indecentes”, con la tecla ”, porque obtenemos resultados “indecentes”.

Si se usa el paquete `babel`, opción `spanish`, disponemos de las llamadas «comillas francesas» o *guillemets* (también conocidas como comillas latinas o angulares), por medio de los siguientes comandos²:

« \guillemotleft » \guillemotright

Estas comillas se pueden obtener más sencillamente con las combinaciones "< y >", respectivamente.

Unión de palabras. Se usa el símbolo ~ para sugerirle a LATEX que no separe determinadas palabras en el margen derecho. Algo como “C. F. Gauss” no luce bien; para evitarlo escribimos C.F.~Gauss en el archivo de entrada. Otros ejemplos: `secciones 1~a~7`, `mayo~5`, `II~Guerra Mundial`.

Sílabas. Usando el paquete `babel`, con el idioma escogido (sección 2.9), la partición silábica en el margen derecho es casi siempre la correcta. En ocasiones, sin embargo, LATEX y `babel` son incapaces de decidir sobre la partición de una palabra y debemos indicarla manualmente. Esto se hace en el documento fuente escribiendo, por ejemplo,

¹Para el caso en que el teclado o las fuentes usadas no permitan fácil acceso, LATEX tiene también comandos propios para guiones y comillas (véase la sección 3.6).

²Los comandos `\guillemotleft` y `\guillemotright` también están disponibles si se usa LATEX con fuentes que tengan la codificación T1. Véase al respecto la sección 9.1.

`eu\cli\dia\no`, en vez de `euclidiano`, `re\ti\cu\lo`, en vez de retículo, para lograr la separación adecuada.

Una manera más eficiente es indicar desde un comienzo la separación correcta de las palabras problemáticas para que L^AT_EX las divida correctamente *cada vez que aparezcan*. Esto se logra con la instrucción `\hyphenation{...}`. Se usa en la forma

```
\hyphenation{eu-cli-dia-no dia-go-na-li-za-ble ul-tra-fil-tro
abe lia-no ho-meo-mor-fo}.
```

Las palabras se separan con espacios en blanco, sin signos de puntuación. La instrucción se debe colocar en el preámbulo del documento. Desafortunadamente, `\hyphenation{...}` no admite símbolos con acentos como `\'a` o `\'i` (ni aun con el paquete `inputenc`). En tales casos, debemos escribir la partición manualmente o usar L^AT_EX con fuentes que tengan una codificación (en inglés, *encoding*) diferente de la estándar, conocida como codificación OT1 (véase la sección 9.1).

Espaciamiento francés. Después de los cuatro símbolos de puntuación `. : ! ?`, L^AT_EX añade un pequeño espacio, adicional al usual espaciamiento entre palabras (excepto en el caso en que un punto esté precedido de una letra mayúscula; véase el apartado ‘Puntos’ en la página 34). La declaración global `\frenchspacing` elimina ese espacio adicional; es decir, con `\frenchspacing` el espacio entre palabras es aproximadamente el mismo, incluso después de los signos de puntuación. Esta declaración permanece vigente hasta que sea contrarrestada con `\nonfrenchspacing`.

Un detalle importante: la opción `spanish` del paquete `babel` activa la declaración `\frenchspacing`.

Ordinales y volados. Para ordinales y expresiones pequeñas en la parte superior del renglón (también conocidas como “volados”) la opción `spanish` del paquete `babel` brinda las abreviaciones `"a`, `"A`, `"o`, `"O` y el comando `\sptext{...}`.

Ejemplo

Entrada	Salida
<code>1"a sesión</code>	<code>1.^a sesión</code>
<code>5"A Sinfonía</code>	<code>5.^A Sinfonía</code>
<code>2"o capítulo</code>	<code>2.^o capítulo</code>
<code>3\sptext{er} examen</code>	<code>3.^{er} examen</code>

3.3. Tipos de letra y fuentes

El manejo de fuentes en LATEX 2 ε difiere considerablemente del que se empleaba en LATEX 2.09 y versiones anteriores. Las fuentes que usa LATEX por defecto (y con las cuales está escrito el presente libro) son las llamadas fuentes CM (*Computer Modern Fonts*), diseñadas por el propio Donald Knuth, el creador de TEX³. Con los siguientes comandos se accede a los principales tipos de letra disponibles:

- romana normal \textrm{...}
- sans serif \textsf{...}
- mono-espaciada (*typewriter*) \texttt{...}
- cursiva o itálica \textit{...}
- negrilla \textbf{...}
- inclinada (*slanted*) \textsl{...}
- versalitas (*small caps*) \textsc{...}

Ejemplo.

\textrm{Lea esta frase, por favor.}	Lea esta frase, por favor.
\textit{Lea esta frase, por favor.}	<i>Lea esta frase, por favor.</i>
\textbf{Lea esta frase, por favor.}	Lea esta frase, por favor.
\textsl{Lea esta frase, por favor.}	<i>Lear esta frase, por favor.</i>
\textsf{Lea esta frase, por favor.}	Lear esta frase, por favor.
\textsc{Lea esta frase, por favor.}	LEA ESTA FRASE, POR FAVOR.
\texttt{Lea esta frase, por favor.}	Lear esta frase, por favor.

Podemos hacer múltiples cambios de fuente en nuestros documentos, como se ilustra a continuación.

Ejemplo.

Todo lo dicho es *sumamente* importante y lo que viene también lo es.

```
\textbf{Todo} lo dicho es \textit{sumamente} importante y lo que
viene \textsl{tambi\'en lo es.}
```

Los anteriores comandos se pueden combinar para obtener tipos de letra con diferentes atributos. Por ejemplo, \textit{...} se puede combinar con \textbf{...} para obtener letra cursiva negrilla.

³En el Capítulo 9 se explica cómo se pueden reemplazar las fuentes CM estándares por otras fuentes, en especial por las fuentes PostScript de las colecciones AE y PSNFSS.

Ejemplo

Esta frase está escrita en la fuente cursiva negrilla.

```
\textbf{\textit{Esta frase está escrita en la fuente cursiva negrilla.}}
```

Esta frase está escrita en la fuente sans serif inclinada.

```
\textsf{\textsl{Esta frase está escrita en la fuente sans serif inclinada.}}
```

Esta frase está escrita en la fuente inclinada negrilla.

```
\textsl{\textbf{Esta frase está escrita en la fuente inclinada negrilla.}}
```

Al combinar tipos de letra, como en el ejemplo anterior, el orden es irrelevante; así, la combinación `\textbf{\textit{...}}` tiene el mismo efecto que `\textit{\textbf{...}}`. Se advierte al lector que no todas las combinaciones de los comandos básicos dan lugar a nuevas fuentes. Por ejemplo, las fuentes CM no tienen versalitas en negrilla. Cuando el tipo de letra solicitado no está disponible, L^AT_EX realiza una sustitución de fuentes y emite una advertencia (que el usuario puede leer en el archivo ‘`—.log`’) con el siguiente aspecto:

`LaTeX Font Warning: Font shape ... not available ...`

Cada uno de los tipos de letra básicos tiene su modo *enfático*, obtenido por medio de `\emph{...}`.

Ejemplo

Todo lo dicho es sumamente importante.

Todo lo dicho es \emph{sumamente} importante.

Todo lo dicho es sumamente importante.

```
\textsf{Todo lo dicho es \emph{sumamente} importante.}
```

Todo lo dicho es sumamente importante.

```
\textbf{Todo lo dicho es \emph{sumamente} importante.}
```

Todo lo dicho es sumamente importante.

```
\textit{Todo lo dicho es \emph{sumamente} importante.}
```

Todo lo dicho es sumamente importante.

```
\texttt{Todo lo dicho es \emph{sumamente} importante.}
```

Aparte de los mencionados comandos, también existen declaraciones globales para cambios de fuente (Tabla 3.1); su alcance está delimitado por corchetes exteriores `{...}`⁴. Los comandos de la forma `\text--{...}`, a diferencia de las declaraciones globales, también se pueden usar en modo matemático (véase la sección 4.11).

Declaración global	corresponde a
<code>\rm ...</code> o <code>\rmfamily ...</code>	<code>\textrm{ ... }</code>
<code>\sf ...</code> o <code>\sffamily ...</code>	<code>\textsf{ ... }</code>
<code>\tt ...</code> o <code>\ttfamily ...</code>	<code>\texttt{ ... }</code>
<code>\bf ...</code> o <code>\bfseries ...</code>	<code>\textbf{ ... }</code>
<code>\it ...</code> o <code>\itshape ...</code>	<code>\textit{ ... }</code>
<code>\sl ...</code> o <code>\slshape ...</code>	<code>\textit{ ... }</code>
<code>\sc ...</code> o <code>\scshape ...</code>	<code>\textit{ ... }</code>
<code>\em ...</code>	<code>\emph{ ... }</code>

TABLA 3.1. Comandos y declaraciones globales para cambios de fuente.

3.4. Tamaño de la letra

El tamaño de la letra en documentos L^AT_EX escritos con los estilos `article`, `book`, `report` y `letter` es, por defecto, de 10 puntos (10pt), con las opciones adicionales `[11pt]` y `[12pt]`⁵. Los tamaños y tipos de letra en los títulos globales y en los títulos de secciones y subsecciones cambian automáticamente. No obstante, L^AT_EX ofrece toda una gama de tamaños con las siguientes declaraciones globales:

```
\tiny \scriptsize \footnotesize \small
      \normalsize
\large \Large \LARGE \huge \Huge
```

Estas declaraciones cambian el tamaño pero no el tipo de letra y su alcance se delimita con corchetes exteriores. Si se omiten los corchetes de delimitación, la declaración permanece vigente por el resto del documento, a menos que esté incluida en otro grupo delimitado por llaves `{ }` o en un entorno.

⁴Las declaraciones globales de dos letras, `\bf`, `\sc`, `\tt`, etc, son las mismas usadas en L^AT_EX 2.09.

⁵Los estilos `amsart` y `amsbook` ofrecen dos tamaños adicionales: `[8pt]` y `[9pt]` (véase la sección 4.35).

Ejemplo

\tiny Cogito, ergo sum	Cogito, ergo sum
\scriptsize Cogito, ergo sum	Cogito, ergo sum
\footnotesize Cogito, ergo sum	Cogito, ergo sum
\small Cogito, ergo sum	Cogito, ergo sum
\normalsize Cogito, ergo sum	Cogito, ergo sum
\large Cogito, ergo sum	Cogito, ergo sum
\Large Cogito, ergo sum	Cogito, ergo sum
\LARGE Cogito, ergo sum	Cogito, ergo sum
\huge Cogito, ergo sum	Cogito, ergo sum
\Huge Cogito, ergo sum	Cogito, ergo sum

Se debe tener presente que la anterior es una gama *relativa* de tamaños; así por ejemplo, el tamaño producido por medio de {\normalsize ... } en un documento con la instrucción \documentclass[12pt]{article} es de 12 puntos, y \small produce el tamaño inmediatamente inferior: 11 puntos. Los tamaños exhibidos en el ejemplo anterior corresponden al caso en que \normalsize es 11 puntos, ya que el presente libro fue escrito con la opción [11pt].

3.5. Combinaciones de tipo y tamaño de letra

Los comandos para cambio de tipo y tamaño de letra se pueden combinar de manera natural. Para esta clase de combinaciones se pueden usar los comandos \text{--{...}} o las declaraciones globales \--family, \--shape y \--series (Tabla 3.1), pero hay que evitar usar las formas simplificadas \it, \bf, \sf, etc, porque con éstas no se pueden combinar todos los atributos de las fuentes.

Ejemplo

\textit{\tiny Lea esta frase}	<i>Lea esta frase</i>
\textsf{\small Lea esta frase}	Lea esta frase
\textsf{Lea {\large \emph{esta}} frase}	Lea <i>esta</i> frase
\textbf{\textit{\Large Lea esta frase}}	<i>Lea esta frase</i>
\textsc{\Large Lea esta frase}	LEA ESTA FRASE

Ejemplo Aquí se usan las declaraciones globales de la Tabla 3.1 para combinar los atributos de las fuentes. Compárese con el ejemplo anterior.

{\tiny\itshape Lea esta frase}	<i>Lea esta frase</i>
{\small\sffamily Lea esta frase}	Lea esta frase
{\large\sffamily Lea \emph{esta} frase}	Lea esta frase
{\Large\bfseries\itshape Lea esta frase}	<i>Lea esta frase</i>
{\Large\scshape Lea esta frase}	LEA ESTA FRASE

No todos los tipos de letra están disponibles en todos los tamaños. Cuando una determinada fuente no existe en el tamaño solicitado, L^AT_EX realiza una sustitución por una fuente similar y emite una advertencia (que el usuario puede leer en el archivo ‘*—.log*’). La letra en versalitas (\textsc o \scshape), por ejemplo, no existe en versión negrita.

Si una combinación particular de tipo y tamaño de letra va a ser usada con frecuencia, es recomendable asignarle un comando de acceso, por medio de \DeclareTextFontCommand. Más precisamente, la instrucción

\DeclareTextFontCommand{\fuente}{especificaciones}
--

define un nuevo comando, \fuente, con el cual se obtiene la combinación de tipo y tamaño de letra declarada en las *especificaciones*. El comando \fuente tiene un argumento y se usa en la misma forma que los comandos \text--{...} de la sección 3.3; así, para obtener un determinado *texto* en la nueva \fuente, escribimos \fuente{*texto*}. El nombre \fuente es escogido por el usuario; en la expresión “*fuente*” solamente se admiten letras, minúsculas o mayúsculas, pero no dígitos ni otros símbolos del teclado.

Es preciso advertir que \DeclareTextFontCommand sólo se puede usar en el preámbulo del documento; en caso contrario, se recibe el siguiente mensaje de error:

! LaTeX Error: Can be used only in preamble.

Ejemplo A la combinación de los tres atributos \Large, \itshape y \bfseries podemos asignarle el nombre \Laritbf por medio de

\DeclareTextFontCommand{\Laritbf}{\Large\itshape\bfseries}
--

Al escribir \Laritbf{!‘Lea esta frase, por favor!’}, por ejemplo, obtenemos:

¡Lea esta frase, por favor!

3.6. Caracteres especiales y otros símbolos

Como se mencionó en la sección 2.1, ciertos símbolos del teclado son de uso reservado para L^AT_EX. Para obtenerlos explícitamente se deben utilizar instrucciones L^AT_EX específicas, como se indica a continuación (véase también la sección 3.20).

Símbolo	Instrucción	Símbolo	Instrucción
{	\{	\$	\\$
}	\}	%	\%
&	\&	#	\#
\	\textbackslash		\textbar
-	_	'	\textasciicircum
~	\textasciitilde	^	\textasciicircum
~	\sim	\land	\land

En la siguiente tabla aparecen algunos símbolos de uso general (el comando \textcircled{ } tiene un argumento). En la sección 4.3 se presentan numerosos símbolos adicionales, incluidos en el paquete **amssymb**.

Símbolo	Instrucción	Símbolo	Instrucción
†	\dag	£	\pounds
‡	\ddag	•	\textbullet
§	\S	¶	\P
©	\copyright	®	\textcircled{a}
®	\textregistered	™	\texttrademark

En las secciones 3.1 y 3.2 se mencionaron las llamadas ligaduras para guiones, comillas y los símbolos invertidos de admiración e interrogación. L^AT_EX 2_E posee comandos propios para casos en los que el teclado o las fuentes usadas no permitan tales ligaduras:

Ligadura	Símbolo	Instrucción
---	—	\textemdash
--	—	\textendash
!‘	¡	\textexclamdown
?‘	¿	\textquestiondown
“	”	\textquotedblleft
”	”	\textquotedblright
‘	‘	\textquoteright
,	,	\textquoteright

3.7. Espacio horizontal

En la sección 2.1 se explicó, a grandes rasgos, cómo L^AT_EX maneja el espacio horizontal. Las instrucciones básicas para añadir pequeños espacios horizontales aparecen en la Tabla 3.2. En modo matemático existen comandos adicionales para micro-espacios positivos y negativos, frecuentemente útiles en el retoque de expresiones matemáticas (véase la sección 4.17).

\u	Produce el espacio promedio ocupado por una letra, siendo \u el resultado de presionar la barra espaciadora.
\,	Produce un micro-espacio cuya anchura es: \ll .
\quad	Produce un espacio horizontal de amplitud igual a la anchura de la letra M (en la fuente vigente). Equivale a 6 micro-espacios.
\quad\quad	Produce un espacio horizontal equivalente a dos \quad.

TABLA 3.2. Pequeños espacios horizontales.

Para añadir espacio horizontal de longitud determinada se usa la instrucción

\hspace{longitud}

La longitud deseada se debe presentar solamente en las unidades de medida aceptadas por L^AT_EX (sección 2.11) y puede ser negativa, lo que equivale a “retroceder espacio” en la cantidad especificada.

Ejemplo.

Año: Mes: Día:

Año: \hspace{2cm}Mes: \hspace{4cm}Día: \hspace{3cm}.

Hay que advertir que un espacio en blanco antes o después de \hspace se tiene en cuenta, es decir, se añade a la distancia especificada en el argumento de \hspace. Esto se aprecia en el siguiente ejemplo.

Ejemplo. Compare:

Izquierda	Derecha	Izquierda \hspace{1cm} Derecha
Izquierda	Derecha	Izquierda \hspace{1cm} Derecha
Izquierda	Derecha	Izquierda \hspace{1cm} Derecha

L^AT_EX ignora la instrucción \hspace{...} al comienzo de un renglón o de una página; en tales casos se debe usar la versión estrella, \hspace*{...}.

La instrucción \hfill se usa para llenar con espacio en blanco el espacio horizontal disponible. Su uso se explica en la sección 3.23.

3.8. Sangrías

Al comienzo de cada párrafo, L^AT_EX deja automáticamente una sangría (en inglés, *indentation*); para eliminarla se debe usar la instrucción `\noindent` al comienzo del párrafo. Por ejemplo, al inicio del presente párrafo hemos usado `\noindent`, siguiendo una tradicional norma tipográfica —de indiscutible origen anglosajón— que recomienda no usar sangría en el primer párrafo de un capítulo o de una sección (en general, se recomienda eliminar la sangría después de cualquier material que aparezca desplegado o centrado).

La sangría con la que se inicia cada párrafo se controla con `\parindent` y su anchura se puede cambiar usando la instrucción `\setlength` (sección 2.11).

Ejemplo `\setlength{\parindent}{1cm}` establece una sangría de 1 cm al comienzo de todo párrafo, a partir del sitio en el que aparece la instrucción.

Ejemplo `\setlength{\parindent}{0pt}` elimina completamente la sangría al comienzo de todo párrafo, a partir del sitio en el que aparece la instrucción.

3.9. Espacio vertical

Como se explicó en la sección 2.1, para comenzar un nuevo párrafo se deja una línea en blanco y una línea en blanco es lo mismo que 10 o 100 líneas en blanco: en ambos casos se inicia un nuevo párrafo sin espacio vertical adicional. Otra manera de comenzar un nuevo párrafo es por medio de la instrucción `\par`. En la Tabla 3.3 aparecen otras instrucciones relacionadas.

Para añadir espacio vertical de *longitud* determinada podemos usar alguna de las dos instrucciones siguientes:

\[\[longitud] L^AT_EX inicia inmediatamente un nuevo renglón (sin sangría), insertando un espacio vertical de *longitud* dada, antes del nuevo renglón. El texto que precede a esta instrucción no es justificado a la derecha. El argumento `[longitud]` es opcional; es decir, con `\\\` simplemente se inicia un nuevo renglón (sin sangría). Ejemplos concretos son: `\[3mm]`, `\[2cm]`, `\[5pt]`.

\vspace{longitud} L^AT_EX inserta un espacio vertical de *longitud* especificada. Si la instrucción aparece en medio de un párrafo, el espacio se inserta después del renglón que la contiene. Ejemplos concretos son: `\vspace{3mm}`, `\vspace{2cm}`, `\vspace{5pt}`.

\par	Inicia un nuevo párrafo (con sangría).
\\\	Inicia una nueva línea o renglón (sin sangría). El texto que precede a \\\ no es justificado a la derecha.
\newline	Tiene el mismo efecto que \\.
\linebreak	Justifica el renglón actual, es decir, estira proporcionalmente todos los caracteres hasta tocar el margen derecho y comienza un nuevo renglón, sin sangría.
\nolinebreak	Impide que LATEX inicie un nuevo renglón en el sitio en el que se escribe esta instrucción. De hecho, \nolinebreak es más una sugerencia que una orden pues LATEX la ignora si no puede acomodar adecuadamente el texto.
\smallskip	Inserta un pequeño espacio vertical antes de comenzar un nuevo párrafo. Este comando debe ir precedido de una línea en blanco o de la instrucción \par; de lo contrario será ignorado.
\medskip	Equivale a dos \smallskip; debe ir precedido de una línea en blanco o de la instrucción \par.
\bigskip	Equivale a tres \smallskip; debe ir precedido de una línea en blanco o de la instrucción \par.

TABLA 3.3. Inicio de párrafos y espacio vertical.

LATEX ignora las instrucciones \vspace{...} y \\[...] al comienzo de una página; en tales casos se debe usar la versión estrella, \vspace*{...}. Así por ejemplo, si al inicio de una página queremos un espacio en blanco de 5 cm (adicional al margen superior), escribimos \vspace*{5cm}.

La distancia entre renglones y la distancia entre párrafos están predefinidas para cada estilo de documento pero pueden ser modificadas, si el usuario así lo desea, recurriendo a las dos instrucciones siguientes:

\baselinestretch: Controla la distancia entre renglones (en párrafos normales) y afecta todo el documento; se cambia por medio de la instrucción \renewcommand, *sólo* en el preámbulo del documento. LATEX cambia automáticamente este parámetro cuando el usuario escoge una de las opciones de tamaño de letra [11pt] o [12pt].

Ejemplo \renewcommand{\baselinestretch}{1.35} incrementa el espaciamiento normal entre renglones en un 35 %.

\parskip Controla el espacio entre párrafos; por defecto, dicho espacio es igual al espacio entre renglones. Se puede cambiar múltiples veces dentro de un mismo documento usando la instrucción \setlength.

Ejemplo \setlength{\parskip}{2mm} establece una distancia *adicional* de 2 mm entre párrafos.

3.10. Control sobre cambios de página

Para controlar cambios de página, L^AT_EX tiene las instrucciones básicas mostradas en la Tabla 3.4.

\newpage	Inicia una nueva página.
\pagebreak	Justifica verticalmente el contenido de la página añadiendo espacio adicional entre los párrafos (no entre los renglones) y comienza una nueva página.
\clearpage	Es similar a \newpage excepto que las tablas o figuras que estén bajo el alcance de los entornos table o figure , y que no hayan sido colocadas por L ^A T _E X, se imprimen en una o más hojas separadas. Sobre la inserción de tablas y figuras véase la sección 7.4.
\cleardoublepage	Funciona como \clearpage para documentos con la opción twoside (Tabla 2.2). L ^A T _E X añade toda una hoja en blanco adicional, si es necesario, para que la siguiente página de texto tenga numeración impar.
\nopagebreak	Impide que L ^A T _E X inicie una nueva página en el sitio en el que aparece esta instrucción.

TABLA 3.4. Comandos para cambios de página.

Cuando se usa la opción **twocolumn** de **\documentclass** o la instrucción **\twocolumn** (véase la sección 2.14), los comandos **\newpage** y **\pagebreak** no inician una nueva página sino una nueva columna, pero los comandos **\clearpage** y **\cleardoublepage** sí inician una nueva página, añadiendo una columna vacía en la página actual, de ser necesario.

La declaración **\flushbottom** hace que la altura de la parte impresa de todas las páginas (completas) sea la misma, mientras que **\raggedbottom** tiene el efecto contrario: permite que la altura de la parte impresa varíe un poco de página a página. Por defecto, L^AT_EX utiliza **\raggedbottom**, excepto con la opción **twoside** (Tabla 2.2). En particular, en el estilo **book** las páginas tienen todas la misma altura, y para lograrlo L^AT_EX añade espacio vertical entre párrafos si es necesario. El usuario puede modificar la acción pre-determinada colocando la declaración apropiada en el preámbulo o en cualquier otra parte del documento. Estas son declaraciones globales, es decir, permanecen vigentes hasta que sean contrarrestadas por la declaración contraria.

Al preparar la versión *definitiva* de un documento se necesita algunas veces extender el contenido de una página particular en uno, dos o más renglones. Con **\nopagebreak** es a veces imposible lograrlo ya que, más

que una instrucción, `\nopagebreak` es una sugerencia que L^AT_EX ignora si no puede acomodar adecuadamente el texto. L^AT_EX 2 _{ε} posee dos comandos específicamente diseñados para forzar páginas más largas. La instrucción

```
\enlargethispage{longitud}
```

incrementa el tamaño de la página en la cantidad especificada por *longitud* y afecta solamente la página en la cual aparece. Si la cantidad especificada en *longitud* es negativa, el tamaño de la página se reduce. La instrucción

```
\enlargethispage*{longitud}
```

funciona de manera similar pero más eficiente: elimina proporcionalmente el espacio entre párrafos u otro espacio en blanco disponible, maximizando así la cantidad de texto en la página.

La manera más práctica de extender el tamaño de una página particular es añadir una cantidad generosa de espacio, con `\enlargethispage*{5cm}`, por ejemplo, y escribir luego la instrucción `\newpage` exactamente en el sitio donde se desea terminar la página.

3.11. Texto subrayado

Para subrayar texto se utiliza la instrucción `\underline{...}`, la cual se puede usar en los tres modos de procesamiento de L^AT_EX: modo normal, modo ID y modo matemático (sección 2.1).

Ejemplo

L^AT_EX tiene muchos comandos y se deben aprender poco a poco.

```
\LaTeX\ tiene \underline{muchos} comandos y se deben aprender
\underline{poco a poco}.
```

3.12. Citas

L^AT_EX tiene dos entornos para citas: `quote` y `quotation`:

```
\begin{quote}
texto de la cita
\end{quote}
```

```
\begin{quotation}
texto de la cita
\end{quotation}
```

Las citas aparecen desplegadas, centradas y con sangrías a la izquierda y a la derecha. L^AT_EX añade un pequeño espacio vertical antes y después de la

cita. En ambos entornos, el inicio de un párrafo se obtiene, como es usual, con una línea en blanco o con el comando \par.

La diferencia fundamental entre los dos entornos es la siguiente: aparte de la sangría doble (a izquierda y a derecha), con **quotation** cada párrafo se inicia con una sangría adicional mientras que con **quote** no hay sangría extra sino un espacio vertical adicional entre los párrafos. Por consiguiente, **quote** se puede usar para una cita corta o una sucesión de citas cortas, y **quotation** para citas extensas que se extiendan por varios párrafos.

Ejemplo Aquí se usa el entorno **quote** para una sucesión de citas cortas.

Aprendemos por experiencia que los seres humanos nunca aprenden nada por experiencia. *George Bernard Shaw*

La diferencia entre el genio y la estupidez es que el genio tiene límites. *Napoleón Bonaparte*

La mayoría de la gente prefiere morir antes que pensar ... ¡y lo consiguen! *Bertrand Russell*

```
\begin{quote}
```

Aprendemos por experiencia que los seres humanos nunca aprenden nada por experiencia. \textit{George Bernard Shaw}\par

La diferencia entre el genio y la estupidez es que el genio tiene límites. \textit{Napoleón Bonaparte}\par

La mayoría de la gente prefiere morir antes que pensar \ldots ! 'y lo consiguen! \textit{Bertrand Russell}

```
\end{quote}
```

Ejemplo Se ha usado el entorno **quotation** para presentar la siguiente cita, tomada de las *Analectas* de Confucio. Obsérvese la sangría al inicio de cada párrafo, adicional a las sangrías laterales.

Lo que busca el hombre superior se halla en él mismo; lo que busca el hombre vulgar se encuentra en los demás.

El hombre superior está en armonía con los demás, pero no siempre de acuerdo con ellos. El hombre vulgar se pone de acuerdo con los demás, pero no está en armonía con ellos.

El hombre superior es fácil de servir y difícil de complacer. El hombre vulgar, en cambio, es difícil de servir y fácil de complacer.

```
\begin{quotation}
```

Lo que busca el hombre superior se halla en él mismo; lo ... \par

El hombre superior está en armonía con los demás, pero no ... \par

El hombre superior es fácil de servir y difícil de complacer ...

```
\end{quotation}
```

3.13. Texto centrado

Para centrar una línea de texto se utiliza la instrucción `\centerline{...}`⁶.

Ejemplo

Lea esta frase, por favor.

```
\centerline{Lea esta frase, por favor.}
```

Para centrar más de una línea de texto se usa el entorno `center`. L^AT_EX añade un pequeño espacio vertical antes y después del texto centrado, e inicia un nuevo renglón cada vez que alcanza el margen derecho, sin dividir palabras.

Ejemplo

“El sentido común es la cosa mejor repartida del mundo, ya que cada uno piensa estar tan bien provisto de él, que incluso los que son difíciles de contentar no suelen desear más del que poseen”. Descartes

```
\begin{center}
‘‘El sentido común es la cosa mejor repartida del mundo, ya que
cada uno piensa estar tan bien provisto de él, que incluso los que
son difíciles de contentar no suelen desear más del que poseen’’.
Descartes
\end{center}
```

Ejemplo Se ilustra cómo el inicio de renglones se puede forzar con `\``.

“El sentido común es la cosa mejor repartida del mundo,
ya que cada uno piensa estar tan bien provisto de él,
que incluso los que son difíciles de contentar
no suelen desear más del que poseen”. Descartes

```
\begin{center}
‘‘El sentido común es la cosa mejor repartida del mundo,\``
ya que cada uno piensa estar tan bien provisto de él,\``
que incluso los que son difíciles de contentar\``
no suelen desear más del que poseen’’. Descartes
\end{center}
```

Además del entorno `center`, L^AT_EX posee la declaración `\centering`. Se usa para centrar el material (texto, tablas, gráficas, etc) que aparezca dentro de otros entornos, como `minipage` (sección 3.16), `table` o `figure` (sección 7.4).

⁶L^AT_EX procesa el argumento de `\centerline` en modo ID (sección 2.1).

3.14. Texto cargado a la izquierda o a la derecha

Para cargar una línea de texto a la izquierda se utiliza la instrucción `\leftline{...}`, y a la derecha `\rightline{...}`⁷. Para más de una línea de texto existen los entornos `flushleft` y `flushright`. L^AT_EX inicia un nuevo renglón cada vez que alcanza el margen derecho, sin dividir palabras, pero se puede forzar el comienzo de un renglón con `\``.

Ejemplo

Si quieras que el futuro sea diferente
del presente debes conocer el pasado.
Baruch Spinoza (1632-1677)

```
\begin{flushright}
Si quieras que el futuro sea diferente\\
del presente debes conocer el pasado.\\
Baruch Spinoza (1632--1677)
\end{flushright}
```

El texto en el entorno `flushleft` pierde su justificación en el margen derecho mientras que en el entorno `flushright` el texto queda cargado a la derecha y no justificado a la izquierda. Para eliminar la justificación del texto en los márgenes existen también declaraciones:

`\raggedleft` Elimina la justificación a la izquierda; es el análogo del entorno `flushright`.

`\raggedright` Elimina la justificación a la derecha; es el análogo del entorno `flushleft`.

Estos dos comandos se usan principalmente dentro de otros entornos (como `quote`) o dentro de `\parbox` (véase el último ejemplo de la página 54).

3.15. Cajas

Una caja (en inglés, *box*) es un elemento que L^AT_EX trata como si fuera una sola letra: no importa cuán grande es, L^AT_EX nunca la divide en partes. Podemos crear cajas para recuadros —con marcos visibles o invisibles— que contengan texto o fórmulas, o para producir mini-párrafos con dimensiones diferentes de las del documento. Podemos colocar luego las cajas unas al lado de otras como lo haríamos con letras o texto normal.

⁷Al igual que sucede con `\centerline`, L^AT_EX procesa los argumentos de `\leftline` y `\rightline` en modo ID (sección 2.1). Los tres son comandos T_EX primarios.

3.15.1. Cajas con una sola línea de texto

El texto contenido en estas cajas es escrito por L^AT_EX en modo ID (sección 2.1). Como es usual, los argumentos obligatorios se encierran entre llaves { } y los opcionales entre paréntesis angulares []. Estas cajas pueden tener bordes visibles o invisibles y anchura automática o escogida por el usuario.

\mbox{*texto*}. Caja con bordes invisibles que contiene al *texto*.

\fbox{*texto*}. Caja con bordes visibles que contiene al *texto*.

\makebox[*ancho*][*justificación*]{*texto*}. Caja con bordes invisibles que contiene al *texto*; el usuario puede especificar el *ancho* de la caja, y la posición del *texto* dentro de ésta se controla con el parámetro [*justificación*].

\framebox[*ancho*][*justificación*]{*texto*}. Como \makebox pero con bordes visibles.

Los parámetros válidos para [*justificación*] son:

- [c] Texto centrado; opción pre-determinada.
- [l] Texto cargado a la izquierda.
- [r] Texto cargado a la derecha.
- [s] Texto extendido o “estirado” (el ancho de la caja); se extiende el espacio entre palabras, no el espacio entre letras.

El *ancho* en \makebox y \framebox puede ser cualquier dimensión T_EX; si se suprime este parámetro opcional, la caja adquiere el ancho del *texto* contenido en ella. L^AT_EX calcula y almacena las cuatro dimensiones del argumento *texto* mostradas en la Tabla 3.5.

\width	Anchura natural del <i>texto</i> .
\height	Altura natural del <i>texto</i> , medida desde la base del renglón (en inglés, <i>baseline</i>), que es el sitio donde descansan las letras sin perfil inferior, como a, b, c.
\depth	Distancia desde la base del renglón hasta el extremo inferior del <i>texto</i> .
\totalheight	\height + \depth

TABLA 3.5. Dimensiones naturales del argumento *texto* de cajas L^AT_EX.

En el argumento [*ancho*] de \makebox y \framebox se puede usar cualquier fracción de las dimensiones presentadas en la Tabla 3.5 ($1.8\width$, $2.3\width$, $1.5\depth$, etc). Esto se ilustra en el siguiente ejemplo.

Ejemplo El ancho de las siguientes cajas es 1.5\width , es decir, 1.5 veces el ancho del texto contenido en ellas.

¡No fume más!	<code>\framebox[1.5\width]{!‘No fume más!’}</code>
¡No fume más!	<code>\framebox[1.5\width][l]{!‘No fume más!’}</code>
¡No fume más!	<code>\framebox[1.5\width][r]{!‘No fume más!’}</code>
¡No fume más!	<code>\framebox[1.5\width][s]{!‘No fume más!’}</code>

3.15.2. Cajas con párrafos

Si la caja contiene uno o más párrafos, en vez de un sólo renglón de texto, se debe usar la instrucción `\parbox`, en la forma:

<code>\parbox[posición][altura][posición interior]{ancho}{texto}</code>

El significado de los argumentos es el siguiente:

`{texto}` es el contenido de la caja (argumento obligatorio).

`{ancho}` es la longitud horizontal de la caja (argumento obligatorio).

`[posición]` se refiere a la posición de la caja propiamente dicha con respecto al texto exterior que la rodea, es decir, la alineación vertical de la caja. Sus parámetros válidos son:

- [c] Caja centrada; opción pre-determinada.
- [t] Caja cargada hacia su “tope” o parte superior.
- [b] Caja cargada hacia su “base” o parte inferior.

`[altura]` es la longitud vertical de la caja. Puede ser cualquier dimensión \TeX , incluyendo fracciones de las dimensiones de la Tabla 3.5.

`[posición interior]` se refiere a la posición del `texto` dentro de la caja misma. Este argumento únicamente tiene efecto cuando se especifica el argumento `[altura]`; sus parámetros válidos son:

- [c] `texto` verticalmente centrado, opción pre-determinada.
- [t] `texto` cargado hacia arriba.
- [b] `texto` cargado hacia abajo.

Es muy importante tener en cuenta que si se usa el argumento opcional `[posición]`, también se debe usar `[altura]`, y viceversa. Por consiguiente, para que el argumento `[posición interior]` tenga efecto, los otros dos argumentos opcionales `[posición]` y `[altura]` también deben estar presentes.

Los párrafos que aparecen dentro de un `\parbox` no llevan sangría, pero el usuario puede modificar el tamaño de la sangría escribiendo en el argumento `texto` de `\parbox` la declaración `\setlength{\parindent}{...}` véase la sección 3.8).

Ejemplo

Clasificación de los grupos cíclicos

Todo grupo cíclico finito de orden n es isomorfo al grupo aditivo de los enteros módulo n .

Todo grupo cíclico infinito es isomorfo al grupo aditivo de los enteros.

```
\begin{center}
\textbf{Clasificación de los grupos cíclicos}\[0.2cm]
\parbox{4cm}{Todo grupo cíclico finito de orden $n$ es isomorfo al grupo aditivo de los enteros módulo $n$.}
\hspace{1.5cm}
\parbox{4.7cm}{Todo grupo cíclico infinito es isomorfo al grupo aditivo de los enteros.}
\end{center}
```

Ejemplo

Para reflexionar:

Si quieres que el futuro sea diferente del presente debes conocer el pasado.

Baruch Spinoza

```
\makebox[4cm][l]{Para reflexionar:}
\fbox{\parbox[t][1.4cm][6.1cm]{Si quieres que el futuro sea diferente del presente debes conocer el pasado.\\
\rightline{Baruch Spinoza}}}
```

Ejemplo

Para reflexionar:

Si quieres que el futuro sea diferente del presente debes conocer el pasado.

Baruch Spinoza

```
\makebox[1.5\width][l]{Para reflexionar:}
\framebox[1.2\width]{\parbox[c][1.6\height][6.1cm]{Si quieres que el futuro sea diferente del presente debes conocer el pasado.\\
\rightline{Baruch Spinoza}}}
```

Ejemplo Aquí se usan los tres parámetros opcionales de \parbox; para apreciar mejor el resultado, cada \parbox se ha incluido dentro de un \fbox.

Confucio dijo a sus discípulos: “aprender sin pensar es inútil, pensar sin aprender es peligroso.”

Confucio dijo a sus discípulos: “aprender sin pensar es inútil, pensar sin aprender es peligroso.”

Confucio dijo a sus discípulos: “aprender sin pensar es inútil, pensar sin aprender es peligroso.”

```
\fbox{\parbox[c][3.5cm][t]{3.3cm}{\small Confucio dijo a sus
discípulos: ‘‘aprender sin pensar es inútil, pensar sin aprender
es peligroso.’’}}\quad
\fbox{\parbox[c][3.5cm][c]{3.3cm}{\small Confucio dijo a sus
discípulos: ‘‘aprender sin pensar es inútil, pensar sin aprender
es peligroso.’’}}\quad
\fbox{\parbox[c][3.5cm][b]{3.3cm}{\small Confucio dijo a sus
discípulos: ‘‘aprender sin pensar es inútil, pensar sin aprender
es peligroso.’’}}
```

Ejemplo Se ilustra la eliminación de la justificación a la derecha, con \raggedright, y a la izquierda, con \raggedleft (sección 3.14), dentro de \parbox.

Macondo era entonces una aldea de veinte casas de barro y cañabrava construidas a la orilla de un río de aguas diáfanas que se precipitaban por un lecho de piedras pulidas, blancas y enormes como huevos prehistóricos.

```
\parbox{7.5cm}{\raggedright Macondo era . . . . . prehistóricos.}
```

Macondo era entonces una aldea de veinte casas de barro y cañabrava construidas a la orilla de un río de aguas diáfanas que se precipitaban por un lecho de piedras pulidas, blancas y enormes como huevos prehistóricos

```
\parbox{7.5cm}{\raggedleft Macondo era . . . . . prehistóricos.}
```

3.15.3. Parámetros para cajas con líneas

Los comandos `\fbox` y `\framebox` para cajas con líneas tienen dos parámetros de control: `\fboxrule` y `\fboxsep`; ambos se pueden cambiar con `\setlength`, en cualquier parte del documento.

`\fboxrule` Controla el grosor de las líneas; por defecto es 0.4 pt.

Ejemplo `\setlength{\fboxrule}{1pt}` hace que el grosor de las líneas sea de 1 pt.

`\fboxsep` Controla la distancia entre los bordes o líneas de la caja y su contenido. Por defecto, tal distancia es de 3 pt.

Ejemplo `\setlength{\fboxsep}{10pt}` establece una distancia de 10 pt entre las líneas de la caja y su contenido.

3.15.4. Traslación vertical de cajas

Con el comando

`\raisebox{distancia}{texto}`

el `texto` se eleva la `distancia` especificada. El parámetro `distancia` puede ser negativo, en cuyo caso, el `texto` desciende con respecto a la base del renglón (que es el sitio donde descansan las letras sin perfil inferior, como a, b, c)⁸.

Ejemplo

Algunas palabras pueden *elevarse* y otras *descender* del renglón.

Algunas palabras pueden `\raisebox{1.5ex}{\em elevarse}` y otras `\raisebox{-1.5ex}{\em descender}` del renglón.

Ejemplo Presentamos una versión simplificada del logo T_EX. La letra E desciende 0.6 unidades ex (véase la Tabla 2.4). Puesto que la unidad 'ex' es relativa a la fuente vigente, al cambiar el tamaño de la letra, las distancias relativas se mantienen.

T_EX `{\large\bfseries T\raisebox{-0.6ex}{E}X}`

T_EX `{\Large\bfseries T\raisebox{-0.6ex}{E}X}`

T_EX `{\huge\bfseries T\raisebox{-0.6ex}{E}X}`

⁸LaTeX procesa el argumento `texto` de `\raisebox` en modo ID (sección 2.1).

3.15.5. Cajas invisibles y cajas de anchura nula

Los comandos de la Tabla 3.6 son comandos primarios \TeX para “fantasmas”, es decir, cajas invisibles; estos comandos se pueden usar también en documentos \LaTeX . Su uso se ilustrará en secciones posteriores; \vphantom se puede usar, por ejemplo, para añadir espacio vertical, pero no espacio horizontal, en filas de tablas (véase la sección 5.8).

\strut	Produce una caja invisible de anchura nula cuya altura es suficiente para incluir las letras de alto perfil (como f, h, l) y las de bajo perfil (como p, g, q) de la fuente vigente.
$\text{\phantom}\{text\}$	Produce una caja invisible que contiene al <i>text</i> .
$\text{\hphantom}\{text\}$	Produce una caja invisible con el ancho del <i>text</i> pero de altura y profundidad nulas.
$\text{\vphantom}\{text\}$	Produce una caja invisible con la altura del <i>text</i> pero de ancho nulo.

TABLA 3.6. Comandos \TeX para cajas invisibles.

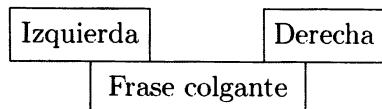
Ejemplo Las dos primeras cajas no tienen exactamente la misma altura porque la palabra ‘izquierda’ tiene una letra de bajo perfil (la letra q) mientras que la palabra ‘derecha’ no tiene letras de bajo perfil. Las dos cajas del renglón siguiente sí tienen exactamente la misma altura porque se ha usado en ellas un \strut .

izquierda	derecha	$\text{\fbox}\{izquierda\} \text{\fbox}\{derecha\}$
izquierda	derecha	$\text{\fbox}\{\text{\strut}\ izquierda\} \text{\fbox}\{\text{\strut}\ derecha\}$

Las cajas \LaTeX de ancho nulo creadas con $\text{\makebox}[0pt][...]\{...\}$ son muy útiles para colocar texto u otro material arriba o debajo de otras cajas. \LaTeX “cree” que la anchura de la caja es nulo pero en el argumento obligatorio $\{...\}$ se puede colocar cualquier material (incluyendo otra caja) que se puede trasladar horizontalmente, con \hspace , o verticalmente, con \raisebox .

Ejemplo La caja con la ‘Frase colgante’ está incluida en una caja con ancho nulo cuyo argumento obligatorio contiene comandos para desplazamiento hacia la izquierda y hacia abajo. Puesto que la anchura de dicha caja es nula, la instrucción $\text{\hspace}\{1.5cm\}$ tiene efecto exactamente después,

y en el mismo renglón, que la caja ‘Izquierda’. Las tres cajas `\framebox` utilizadas tienen un `\strut` para que sean de la misma altura.



```

\begin{center}
\framebox[1.2\width]{\strut Izquierda}\makebox[0pt][l]{%
\hspace{-8mm}\raisebox{-7mm}{\framebox[1.2\width]{\strut Frase \%%
colgante}}}\hspace{1.5cm}\framebox[1.2\width]{\strut Derecha}
\end{center}

```

Ejemplo Para dibujar la siguiente “carita feliz”, a la caja que contiene los ojos (`\mbox{\^o}` `\^o`) le sigue una caja de ancho nulo cuyo argumento obligatorio contiene los comandos para desplazar hacia la izquierda y hacia abajo el símbolo \smile , obtenido con `\$\\smile$`.



```

\fbox[1.5\width]{\mbox{\^o} \^o}\makebox[0pt]{\hspace{-3ex}\raisebox{-2ex}{$\smile$}}}

```

LATEX permite almacenar cajas que sean de uso frecuente en un documento. Esto simplifica las labores de edición y agiliza la compilación. Véase al respecto la sección 3.24.2 en la página 73.

3.16. Mini-páginas

El entorno `minipage` cumple una función similar a la de `\parbox` pero se usa para bloques de texto más extensos. Dentro de una mini-página se pueden usar los comandos de edición de una página normal, incluyendo notas al pie de página (sección 3.17). Para colocar una nota al pie de una gráfica o de una tabla, lo más apropiado es entonces incluir la gráfica o la tabla dentro de una mini-página.

La sintaxis de `minipage` es:

```

\begin{minipage}[posición][altura][posición interior]{ancho}
  texto
\end{minipage}

```

donde los argumentos tienen exactamente el mismo significado que para la instrucción `\parbox` (véase la sección 3.15.2 en la página 52).

3.17. Notas al pie de página

Para notas al pie de página, L^AT_EX posee la instrucción

\footnote{*texto de la nota*}

Las notas se numeran consecutivamente 1, 2, 3, etc. Por ejemplo, en este párrafo hemos colocado dos notas insulsas: primera⁹ y segunda¹⁰, por medio de

```
... dos notas insulsas: primera\footnote{Una nota corta.} y
segunda\footnote{Una nota más larga.}, por medio ...
```

Para el conteo de las notas, L^AT_EX utiliza un *contador* (véase la sección 3.25) que se incrementa en 1 cada vez que aparece \footnote{...}. En el estilo **article** las notas se enumeran consecutivamente en todo el documento, pero en los estilos **book** y **report** se enumeran independientemente en cada capítulo. Podemos forzar el número de la nota con el argumento opcional que posee \footnote{}, en la forma

\footnote[*número*]{*texto de la nota*}

donde *número* debe ser un entero positivo. Al usar el argumento opcional [*número*], el contador de las notas *no* se incrementa.

Otros símbolos para las notas al pie de página. L^AT_EX puede usar símbolos en vez de números para señalar las notas al pie de página. Los nueve símbolos reservados para ello, en el orden en que aparecen en las notas, son los siguientes¹¹:

* † ‡ § ¶ || ** †† ‡‡

Para hacer que L^AT_EX use los anteriores símbolos en las notas al pie de página hay que escribir la siguiente instrucción

\renewcommand{\thefootnote}{\fnsymbol{footnote}}

Hay que tener muy presente que al usar dicha instrucción, el número máximo de notas al pie de página es 9 (en todo el documento, en el estilo **article**, o por capítulo, en los estilos **book** y **report**). Si se excede el número máximo de notas, se recibe el siguiente mensaje de error:

! LaTeX Error: Counter too large.

Por consiguiente, el usuario debe llevar un conteo manual de las notas y cuando se hayan completado nueve, tiene que escribir la instrucción

⁹Una nota corta.

¹⁰Una nota más larga.

¹¹Si se usa una versión del paquete **babel**, opción **spanish**, posterior al año 2000, no aparecen estos símbolos sino secuencias de asteriscos: *, **, ***, ****, ...

```
\setcounter{footnote}{0}
```

con la cual el contador de notas toma de nuevo el valor 0. Esto permite escribir nuevas notas con `\footnote`, las cuales aparecerán demarcadas una vez más en la secuencia *, †, ‡,

También podemos forzar los símbolos para las notas, con el argumento opcional de `\footnote`,

```
\footnote[número]{texto de la nota}
```

teniendo en cuenta que el `número` hace referencia a la siguiente secuencia:

<i>número</i>	1	2	3	4	5	6	7	8	9
<i>símbolo</i>	*	†	‡	§	¶		**	††	‡‡

Al usar el argumento opcional `[número]`, en la forma indicada, el contador de las notas *no* se incrementa.

Redefiniendo el comando `\thefootnote`, L^AT_EX permite otros estilos de numeración para las notas al pie de página. Las opciones disponibles son:

`\renewcommand{\thefootnote}{\roman{footnote}}`. Las notas aparecen demarcadas en la secuencia i, ii, iii, Si se usa una versión del paquete `babel`, opción `spanish`, posterior al año 2000, la numeración obtenida es I, II, III, ... (véase la nota T_EX-nica de la página 27).

`\renewcommand{\thefootnote}{\Roman{footnote}}`. Las notas aparecen numeradas con números romanos grandes I, II, III, IV, ...

`\renewcommand{\thefootnote}{\alph{footnote}}`. Las notas aparecen demarcadas en la secuencia alfabetica a, b, c, d, ...

`\renewcommand{\thefootnote}{\Alph{footnote}}`. Las notas aparecen demarcadas en la secuencia alfabetica A, B, C, D, ...

`\renewcommand{\thefootnote}{\arabic{footnote}}`. Las notas aparecen numeradas con los números arábigos 1, 2, 3, ..., que es la numeración pre-determinada. Usamos esta instrucción para re-establecer la numeración estándar de notas cuando ésta haya sido modificada previamente en el documento.

Notas al pie de página en sitios no permitidos. Hay sitios en los que no es permitido usar `\footnote`; por ejemplo, dentro de cajas creadas con `\mbox`, `\makebox`, `\fbox`, `\framebox` o `\parbox`. Tampoco dentro de otro `\footnote` ni dentro de las tablas creadas con el entorno `tabular` (Capítulo 5). La única instrucción aceptada en los argumentos de tales comandos es

\footnotemark, la cual produce la “marca” o pequeño número que señala la nota. El texto mismo de la nota se escribe con \footnotetext{...}, instrucción que se debe colocar en la misma página pero fuera del alcance de los comandos de la caja o tabla. Tanto \footnotemark como \footnotetext admiten el argumento opcional [*número*] para forzar los números de las notas. Se usa en la misma forma que en el caso de \footnote.

Ejemplo El comando \footnotemark para demarcar la nota al pie de página aparece bajo el alcance \parbox, pero \footnotetext, para el texto de la nota, aparece fuera de \parbox{8cm}{...}.

El día Martes y el mes de Marzo estaban dedicados a Marte¹², dios romano de la guerra.

```
\parbox{8cm}{El día Martes y el mes de Marzo estaban dedicados a  
Marte\footnotemark, dios romano de la guerra.}  
\footnotetext{Llamado Ares por los griegos.}
```

Notas al pie de página en mini-páginas. Las notas al pie de página en una mini-página (sección 3.16) aparecen en la parte inferior de ésta y no en la parte inferior de la página actual. Se numeran, por defecto, con letras minúsculas cursivas *a*, *b*, *c*, etc.

Ejemplo Dos notas al pie de una mini-página.

El dios romano del comercio era Mercurio, palabra cuya raíz^a, también aparece en mercancía, en Miércoles^b y en mercurio, aquel metal cuya fluidez evoca la movilidad del mensajero de los dioses.

^aMerx.

^bMercurii dies, el día de Mercurio.

```
\begin{minipage}{9.5cm}  
El dios romano del comercio era Mercurio, palabra cuya  
raíz\footnote{Merx.}, también aparece en mercancía, en  
Miércoles\footnote{Mercurii dies, el día de Mercurio.} y en  
mercurio, aquel metal cuya fluidez evoca la movilidad del  
mensajero de los dioses.  
\end{minipage}
```

El contador de notas en mini-páginas es diferente del contador de notas en el documento principal. Para escribir una nota al pie de la página principal, desde una mini-página, hay que recurrir a los dos comandos \footnotemark y \footnotetext, mencionados en el inciso anterior.

¹²Llamado Ares por los griegos.

3.18. Notas marginales

\LaTeX tiene la capacidad de escribir notas en los márgenes, con el comando

`\marginpar{texto de la nota}`

El texto de la nota aparece a la altura del renglón en el cual se escribe `\marginpar` y, por defecto, con el tipo y tamaño de letra del documento:

Ejemplo Aquí aparece la nota marginal más famosa en la historia de la ciencia, en el tamaño de letra `\tiny`.

He encontrado una asombrosa demostración pero este margen es muy pequeño para contenerla.

Aquí aparece la nota marginal más `\marginpar{\tiny He encontrado una asombrosa demostración pero este margen es muy pequeño para contenerla.}` famosa en la historia de la ciencia, en el tamaño ...

Las notas marginales aparecen en el margen exterior cuando se usa la opción `twoside` de impresión a dos caras (Tabla 2.2), en el margen derecho cuando se usa la opción `oneside` de impresión a una sola cara (opción pre-determinada en el estilo `article`), y aparecen en el margen más cercano cuando se usa la opción `twocolumn` de impresión a dos columnas. La instrucción `\reversemarginpar` obliga a \LaTeX a imprimir todas las notas sucesivas en el margen opuesto al establecido por defecto, y la instrucción `\normalmarginpar` hace que se impriman normalmente.

El ancho de las notas y la distancia que las separa del texto principal se pueden modificar, con respecto a los valores establecidos por defecto, cambiando los parámetros `\marginparwidth` y `\marginparsep` (véase la sección 2.13). Para acomodar la nota marginal que aparece en esta página hemos reducido el ancho a 1.2 cm, colocando en el preámbulo del documento la instrucción `\setlength{\marginparwidth}{1.2cm}`.

\LaTeX no maneja las notas marginales de manera muy eficiente. Algunos de los problemas que se pueden presentar son:

- La división de palabras no es óptima. Como alternativa, se puede usar `\raggedleft` o `\raggedright` (sección 3.14), o se pueden forzar los renglones con `\backslash\backslash`.
- Una nota marginal nunca es dividida por \LaTeX en dos páginas; así, una nota puede extenderse más allá del borde inferior de la página. El usuario debe hacer ajustes cambiando la posición del comando `\marginpar` en el documento fuente, o usando `\vspace` negativo (sección 3.9) en el argumento de `\marginpar`.
- Notas marginales cercanas pueden traslaparse. La mejor solución en tales casos es proceder por ensayo y error, cambiando la posición de los comandos `\marginpar`.

3.19. Listas con incisos o items

L^AT_EX tiene tres entornos básicos para listas con incisos o items: **itemize**, **enumerate** y **description**. El paquete **enumerate** tiene una extensión muy flexible y útil de ellos, para casi todos los gustos y caprichos (véase la sección 3.19.4 en la página 65). Además, el usuario puede diseñar su propio estilo de listas con el entorno **list** de L^AT_EX, descrito en la sección 8.3.

3.19.1. Los entornos **itemize** y **enumerate**

```
\begin{itemize}
\item Texto
:
\item Texto
\end{itemize}
```

```
\begin{enumerate}
\item Texto
:
\item Texto
\end{enumerate}
```

En el entorno **itemize**, los diferentes items aparecen demarcados con el símbolo **•** y en el entorno **enumerate** se enumeran consecutivamente en la forma 1., 2., 3., Advertimos al lector que si se usa una versión del paquete **babel**, opción **spanish**, posterior al año 2000, la viñeta obtenida con el entorno **itemize** es **■** en vez de **•**.

El comando **\item** de ambos entornos tiene un argumento opcional para las viñetas: **\item[viñeta]**. Así por ejemplo, **\item[a]** da lugar a la viñeta a), **\item[\textbf{(b)}]** da lugar a la viñeta (b), etc.

Ejemplo Obsérvese que el párrafo que sigue a **\end{itemize}** aparece automáticamente sin sangría.

Aristóteles pensaba que hay tres clases de felicidad:

- La felicidad de quien vive de diversiones y placeres.
- La felicidad de quien vive como ciudadano libre y responsable.
- La felicidad de quien vive como filósofo y pensador.

Pensaba que era verdaderamente feliz sólo quien podía combinar equilibradamente las tres clases de felicidad.

Aristóteles pensaba que hay tres clases de felicidad:

```
\begin{itemize}
\item La felicidad de quien vive de diversiones y placeres.
\item La felicidad de quien vive como ciudadano libre y responsable.
\item La felicidad de quien vive como filósofo y pensador.
\end{itemize}
```

Pensaba que era verdaderamente feliz sólo quien ...

Ejemplo

Las fuerzas fundamentales de la naturaleza son:

1. La *fuerza gravitatoria*. Es la más débil de las cuatro y es siempre una fuerza de atracción.
2. La *fuerza electromagnética*. Es experimentada por partículas con carga; puede ser de atracción o de repulsión.
3. La *interacción nuclear fuerte*. Mantiene unidos los núcleos atómicos.
4. La *interacción nuclear débil*. Controla procesos como la desintegración radioactiva.

\noindent Las fuerzas fundamentales de la naturaleza son:

```
\begin{enumerate}  
  \item La {\em fuerza gravitatoria.} Es la más débil ...  
  \item La {\em fuerza electromagnética.} Es experimentada por ...  
  \item La {\em interacción nuclear fuerte.} Mantiene unidos los ...  
  \item La {\em interacción nuclear débil.} Controla procesos ...  
\end{enumerate}
```

3.19.2. El entorno **description**

Se usa de manera similar a los anteriores, excepto que podemos colocar “etiquetas descriptivas” para cada ítem, en la forma `\item[etiqueta]`. Las etiquetas aparecen en negrilla, como se observa en el siguiente ejemplo.

Ejemplo

Los cuatro satélites de Júpiter descubiertos por Galileo son:

Europa: es el menor de los cuatro, un poco más pequeño que la Luna.

Io: tiene más o menos el tamaño de la Luna.

Ganimedes: es el mayor satélite del sistema solar. De hecho, es más grande que el planeta Mercurio, aunque sólo tiene las tres quintas partes de la masa de éste.

Calisto: es el más exterior de los cuatro y el tercero en tamaño.

Los cuatro satélites de Júpiter descubiertos por Galileo son:

```
\begin{description}  
  \item[Europa:] es el menor de los cuatro, ...  
  \item[Io:] tiene más o menos el tamaño de la Luna.  
  \item[Ganimedes:] es el mayor satélite del sistema solar. ...  
  \item[Calisto:] es el más exterior de los cuatro y ...  
\end{description}
```

3.19.3. Listas encajadas en listas

L^AT_EX admite cuatro niveles de listas encajadas, lo que es más que suficiente para la mayoría de documentos. Además, los entornos `itemize`, `enumerate` y `description` se pueden encajar unos en otros. Con `itemize`, L^AT_EX utiliza cuatro viñetas (\bullet , $-$, $*$, \cdot) para demarcar los items en cada uno de los niveles. Si usa una versión del paquete `babel`, opción `spanish`, posterior al año 2000, las cuatro viñetas son: \blacksquare , \bullet , \circ y \diamond . Esto se aprecia en el siguiente ejemplo.

Ejemplo

- Primer ítem del primer nivel.
- Segundo ítem del primer nivel, con una lista encajada.
 - Primer ítem del segundo nivel, con una lista encajada.
 - Primer ítem del tercer nivel.
 - Segundo ítem del tercer nivel, con una lista encajada.
 - Primer ítem del cuarto nivel.
 - Segundo ítem del cuarto nivel.
 - Tercer ítem del cuarto nivel.
 - Segundo ítem del segundo nivel.
 - Tercer ítem del segundo nivel.
- Tercer ítem del primer nivel.

```
\begin{itemize}
\item Primer ítem del primer nivel.
\item Segundo ítem del primer nivel, con una lista encajada.
\begin{itemize}
\item Primer ítem del segundo nivel, con una lista encajada.
\begin{itemize}
\item Primer ítem del tercer nivel.
\item Segundo ítem del tercer nivel, con una lista encajada.
\begin{itemize}
\item Primer ítem del cuarto nivel.
\item Segundo ítem del cuarto nivel.
\item Tercer ítem del cuarto nivel.
\end{itemize}
\end{itemize}
\end{itemize}
\item Segundo ítem del segundo nivel.
\item Tercer ítem del segundo nivel.
\end{itemize}
\item Tercer ítem del primer nivel.
\end{itemize}
```

3.19.4. Listas con el paquete `enumerate`

Para listas enumeradas y encajadas en otras listas es recomendable el uso del paquete `enumerate`. Con el entorno básico `enumerate` de L^AT_EX, se pueden obtener listas similares pero el usuario no tiene mucho control sobre las viñetas obtenidas en cada nivel (además, el tipo de numeración utilizado en la versión estándar de L^AT_EX 2_ε difiere del que se obtiene con el paquete `babel`, opción `spanish`).

El paquete `enumerate`¹³ posee también el entorno `enumerate` el cual tiene, a diferencia de su homólogo de L^AT_EX, un argumento opcional:

```
\begin{enumerate}[estilo]
  Items
\end{enumerate}
```

El *estilo* puede ser uno de los siguientes parámetros:

Parámetro	Viñetas obtenidas
A	A, B, C, D, ...
a	a, b, c, d, ...
I	I, II, III, IV, ...
i	i, ii, iii, iv, ...
1	1, 2, 3, 4, ...

Los parámetros anteriores pueden estar rodeados de símbolos de agrupación o texto adicional. Si alguno de los símbolos A, a, I, i, 1 no ha de ser considerado como parámetro en el argumento *estilo*, se debe encerrar entre corchetes, en la forma {A}, {i}, etc. Si se usa el paquete `babel`, opción `spanish`, el parámetro i produce las viñetas I, II, III, ... en lugar de i, ii, iii, ... (véase la nota T_EX-nica de la página 27).

Los siguientes ejemplos ilustran la versatilidad del paquete `enumerate`, al cual se accede en la forma usual: `\usepackage{enumerate}`.

Ejemplo

E-1 Primera división

```
\begin{enumerate}[E-1]
\item Primera división
```

E-2 Segunda división

```
\begin{enumerate}
\item Segunda división
```

(a) Primer sub-ítem

```
\begin{enumerate}[(a)]
\item Primer sub-item
```

(b) Segundo sub-ítem

```
\item Segundo sub-item
```

(c) Tercer sub-ítem

```
\item Tercer sub-item
```

E-3 Tercera división

```
\begin{enumerate}
\item Tercera división
\end{enumerate}
```

¹³Paquete estándar de L^AT_EX 2_ε, escrito por David Carlisle.

Ejemplo

Noción 1. Célebre

- A) Egregio
- B) Epónimo
- C) Perínclito

Noción 2. Elocuente

- A) Persuasivo
- B) Grandilocuente
- C) Lenguaraz

```
\begin{enumerate}[\{\sf Noción\} 1.]
\item Célebre
\begin{enumerate}[A]
\item Egregio
\item Epónimo
\item Perínclito
\end{enumerate}
\item Elocuente
\begin{enumerate}[A]
\item Persuasivo
\item Grandilocuente
\item Lenguaraz
\end{enumerate}
\end{enumerate}
```

Ejemplo

Rama I. India

I-1 Hindú

I-2 Bengalí

I-3 Sánscrito

(A) Lenguas griegas

- Ático
- Jónico
- Dórico

(B) Lenguas romances

- ★ Latín
- ★ Italiano
- ★ Francés
- ★ Español
- ★ Portugués

Rama II. Germánica

Rama III. Celta

Rama IV. Eslava

```
\begin{enumerate}[\{Rama\} I.]
\item India
\begin{enumerate}[\{I\}-1]
\item Hindú
\item Bengali
\item Sánscrito
\begin{enumerate}[(A)]
\item Lenguas griegas
\begin{enumerate}[$\bullet$]
\item \'Atico
\item Jónico
\item Dórico
\end{enumerate}
\end{enumerate}
\item Lenguas romances
\begin{enumerate}[$\star$]
\item Latín
\item Italiano
\item Francés
\item Español
\item Portugués
\end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

3.20. El entorno `verbatim`

El entorno `verbatim` reproduce texto digitado, imitando una máquina de escribir tradicional. Su sintaxis es:

```
\begin{verbatim}
  texto
\end{verbatim}
```

El contenido completo del `texto` es impreso en la fuente mono-espaciada *typewriter*, tal cual como fue digitado. Esto quiere decir que los caracteres reservados, como \$, \, %, etc, pierden el significado especial que tienen en L^AT_EX y se imprimen literalmente. Los espacios digitados con la barra espaciadora y con la tecla **Enter** son tenidos en cuenta, tal como sucede en una máquina de escribir (pero la tecla **Tab** sólo produce un espacio normal).

Para una sola línea de texto, L^AT_EX posee la instrucción `\verb"..."`. Todo lo que aparece entre las comillas " se imprime literalmente, en la fuente *typewriter*¹⁴. En lugar del símbolo de comillas se pueden usar otros como !, +, =, pero no letras, ni llaves { }, ni el símbolo *.

Ejemplo

Entrada	Salida
<code>\verb"\$135 menos el 10%"</code> .	\$135 menos el 10%.
<code>\verb!\" y \newline!</code> .	\" y \newline.

El entorno `verbatim` y la instrucción `\verb` tienen su respectiva “versión estrella”, `verbatim*` y `\verb*`. Se comportan exactamente como sus homólogos `verbatim` y `\verb`, excepto que cada espacio en blanco produce como salida el símbolo `\` (espacio visible).

Ejemplo

Entrada	Salida
<code>\verb*"a b"</code>	a\ b
<code>\verb*"uno dos tres fin"</code>	uno\ dos\ tres\ fin

Tanto `verbatim` como `\verb` tienen la siguiente limitación importante: no pueden aparecer en el argumento de ningún otro comando L^AT_EX (aunque sí pueden aparecer dentro de otros entornos).

El comando `\verb` y el entorno `verbatim` son útiles para simular código de lenguajes de programación o ilustrar su sintaxis; los hemos usado en este libro para presentar los comandos de L^AT_EX.

¹⁴El argumento de `\verb` es escrito por L^AT_EX en modo ID.

3.21. El paquete **verbatim**

Los entornos **verbatim** y **verbbatim*** de L^AT_EX, presentados en la sección anterior, pueden conducir al temido mensaje de error

! TeX capacity exceeded, sorry [...].

si el texto bajo el alcance de los entornos es muy extenso. Esto se debe al hecho de que el texto incluido en estos entornos es almacenado por L^AT_EX *antes* de terminar el procesamiento del documento, lo que puede originar un súbito agotamiento de memoria. En tales situaciones, el usuario puede usar el paquete **verbatim**¹⁵, el cual redefine los entornos originales para superar dicha limitación. Los dos entornos mantienen la misma sintaxis y sólo hay un pequeño precio que pagar: no se puede escribir texto en el renglón que contiene la instrucción `\end{verbatim}` porque será ignorado.

El paquete **verbatim** tiene otras dos herramientas de gran utilidad:

- El entorno **comment**.
- El comando `\verbatiminput`.

El entorno **comment** es una especie de “entorno anti-**verbatim**”, que se usa en la forma

```
\begin{comment}
  texto
\end{comment}
```

y el cual hace que L^AT_EX ignore todo el *texto* bajo su alcance. Como su nombre lo indica, este entorno es útil para incluir en el documento fuente comentarios personales o notas provisionales, que pueden extenderse por varios párrafos o incluso por varias páginas. Tales comentarios serán ignorados por L^AT_EX durante el procesamiento del documento y no se mostrarán en el documento final ‘*—.dvi*’.

El comando `\verbatiminput{archivo}` hace que L^AT_EX imprima el contenido entero de un *archivo* externo, en la fuente mono-espaciada *typewriter*. Esto es muy útil para transcribir literalmente programas computacionales sin tener que copiarlos en el documento L^AT_EX. Así por ejemplo, con la instrucción `\verbatiminput{C:/Programas/Pascal/prog25}`, se transcribirá el contenido completo del archivo *prog25* ubicado en el subdirectorio *C:/Programas/Pascal*.

¹⁵Paquete estándar de L^AT_EX 2_E, escrito por Rainer Schöpf. Se carga en la forma usual: `\usepackage{verbatim}`.

3.22. Líneas horizontales y verticales

La instrucción `\rule` crea cajas llenas de tinta con las cuales podemos formar líneas rectas del grosor deseado; su sintaxis exacta es:

```
\rule[levantamiento]{ancho}{grosor}
```

siendo obligatorios los argumentos `{ancho}` y `{grosor}`. El argumento opcional `[levantamiento]` se usa para subir (si es positivo) o bajar (si es negativo) la línea, con respecto a la base del renglón, que es el sitio donde las letras sin perfil inferior (como a, b, c) descansan.

Primera línea: 

Segunda línea: 

Tercera línea: 

Cuarta línea: 

```
Primera línea: \rule{5cm}{1pt}\par
Segunda línea: \rule[2mm]{8cm}{2pt}\par
Tercera línea: \rule{1pt}{6mm}\par
Cuarta línea: \rule{8mm}{3mm}
```

3.23. Relleno con espacio en blanco, puntos, líneas o flechas

Los comandos descritos en esta sección no son comandos propios de LATEX sino comandos primarios TEX.

Las instrucciones `\hfill`, `\dotfill` y `\hrulefill` rellenan el espacio horizontal disponible con espacio en blanco, puntos y líneas, respectivamente. TEX hace el relleno en la base del renglón. Las instrucciones `\leftarrowfill` y `\rightarrowfill` rellenan con flechas, que apuntan hacia la izquierda y hacia la derecha, respectivamente; el relleno se hace a media altura, no sobre la base del renglón.

Los ejemplos que siguen muestran cómo, en presencia de estas instrucciones, el espacio disponible se distribuye proporcionalmente.

Ejemplo

Año:_____ Mes:_____ Día:_____.

```
\parbox{7cm}{Año:\hrulefill\ Mes:\hrulefill\ Día:\hrulefill.}
```

Ejemplo

```
Principio          Fin
Principio          Fin
```

```
\noindent Principio\hfill Fin\\
\makebox[6cm]{Principio\hfill Fin}
```

Ejemplo

A —————→12 cm————— B

```
\noindent A\rightarrowarrowfill 12 cm\leftarrowarrowfill B
```

Ejemplo

1	2	3	4	5
1.....	2.....	3.....	4.....	5
1_____	2_____	3_____	4_____	5
1_____	2.....	3_____	4.....	5
1_____	2_____	3_____	4_____	5

```
\noindent 1\hfill 2\hfill 3\hfill 4\hfill 5\\
1\dotfill 2\dotfill 3\dotfill 4\dotfill 5\\
1\hrulefill 2\hrulefill 3\hrulefill 4\hrulefill 5\\
1\hrulefill 2\dotfill 3\hrulefill 4\dotfill 5\\
1\rightarrowfill 2\rightarrowfill 3\leftarrowfill 4\leftarrowfill 5
```

Ejemplo El siguiente despliegue es similar al del ejemplo anterior, excepto que aparece dentro de una mini-página. Obsérvese cómo los 8 cm asignados se distribuyen proporcionalmente.

1	2	3	4	5
1.....	2.....	3.....	4.....	5
1_____	2_____	3_____	4_____	5
1_____	2.....	3_____	4.....	5
1_____	2_____	3_____	4_____	5

```
\begin{minipage}{8cm}
1\hfill 2\hfill 3\hfill 4\hfill 5\\
1\dotfill 2\dotfill 3\dotfill 4\dotfill 5\\
1\hrulefill 2\hrulefill 3\hrulefill 4\hrulefill 5\\
1\hrulefill 2\dotfill 3\hrulefill 4\dotfill 5\\
1\rightarrowfill 2\rightarrowfill 3\leftarrowfill 4\leftarrowfill 5
\end{minipage}
```

Los comandos de relleno horizontal pueden aparecer consecutivamente en un mismo renglón, con lo cual se puede distribuir el espacio de manera muy variada.

Ejemplo

Nombre Apellidos Idioma

```
Nombre \dotfill\dotfill Apellidos \dotfill\dotfill\dotfill
\hfill\hfill Idioma \dotfill
```

Ejemplo El comando `\hfill` es ignorado por L^AT_EX al comienzo y al final de un renglón; para añadir espacio en blanco en tales casos, usamos `\hspace*` (sección 3.7).

Salida 6:30 AM Llegada 2 PM

```
\hspace*{1cm}Salida\dotfill\dotfill\dotfill 6:30 AM\hfill\hfill
Llegada\dotfill\dotfill\dotfill 2 PM \hspace*{1cm}
```

El comando `\vfill` representa la versión vertical de `\hfill`: el espacio vertical disponible se rellena con espacio en blanco. También podemos usar varios `\vfill` consecutivos para añadir más espacio.

Ejemplo El comando `\vfill` es ignorado al comienzo de una página pero no de una mini-página, como se muestra en el presente ejemplo. Los 8 cm de espacio vertical disponibles en esta mini-página se distribuyen por medio de varios `\vfill`. Obsérvese el uso de dos `\vfill` consecutivos.

*El ingenioso hidalgo
Don Quijote de la Mancha*

Compuesto por
MIGUEL DE CERVANTES

Madrid
Año 1605

```
\fbox{%
\begin{minipage}[c][8cm]{6cm}
\centering
\vfill
{\it El ingenioso hidalgo\\
Don Quijote de la Mancha}
\vfill \vfill
Compuesto por \\
{\sc Miguel de Cervantes}
\vfill
Madrid\\
Año 1605
\end{minipage}}
```

3.24. Definición de comandos nuevos

LATEX permite al usuario definir sus propios comandos. Se pueden distinguir tres clases de definiciones:

- Abreviaciones de nombres de comandos ya existentes.
- Definiciones de comandos nuevos, con o sin argumentos.
- Re-definiciones de comandos ya existentes.

En los dos primeros casos usamos la instrucción `\newcommand`, en el tercero usamos `\renewcommand` (estos dos comandos tienen la misma sintaxis). Ya habíamos ilustrado el uso de `\renewcommand` en casos concretos: para cambiar el espaciamiento entre renglones en un documento (sección 3.9) y para cambiar los símbolos que señalan las notas al pie de página (sección 3.17).

En su forma más general, la instrucción `\newcommand` tiene la siguiente sintaxis:

`\newcommand{\nombre}[n][defecto]{definición}`

donde los argumentos `[n]` y `[defecto]` son opcionales; el primero se usa cuando se quiere que `\nombre` tenga `n` argumentos obligatorios y el segundo se reserva como argumento opcional para `\nombre` (para ejemplos de comandos con argumentos véase la sección 4.28). El `nombre` mismo del nuevo comando puede contener una o más letras, mayúsculas o minúsculas, pero no dígitos ni otros símbolos del teclado. Si el comando `\nombre` ya existe, LATEX emitirá el mensaje de error

! LaTeX Error: Command \nombre already defined.

Para evitar esta situación incómoda, el usuario puede usar a cambio la instrucción `\providetcommand` cuya sintaxis es exactamente igual a la de `\newcommand`. Si el nuevo comando definido ya existe, LATEX utilizará la definición existente e ignorará la nueva, pero si se trata de un comando previamente indefinido, LATEX utilizará la nueva definición. En otras palabras, el efecto de `\providetcommand` es exactamente el mismo de `\newcommand`, pero evitando posibles conflictos con los nombres de comandos ya existentes.

Es buena idea colocar todas las instrucciones de la forma `\newcommand` y `\providetcommand` en el preámbulo del documento; esto facilita las labores de revisión y sirve de modelo para futuros documentos.

3.24.1. Comandos simples

Se usan principalmente para simplificar nombres de comandos ya existentes en L^AT_EX o en paquetes de uso frecuente. Estos comandos simples se definen en la forma:

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

A continuación presentaremos algunos ejemplos; para ejemplos adicionales y ejemplos de comandos con argumentos remitimos al lector a la sección 4.28.

Ejemplo Si aparece repetidas veces el comando `\smallskip` en un documento, bien vale la pena abreviarlo con `\sk`, o algo similar. Para ello escribimos la instrucción

```
\newcommand{\sk}{\smallskip}.
```

Ejemplo Podemos abreviar las instrucciones de entornos de uso frecuente; por ejemplo, `\begin{itemize}` lo podemos simplificar como `\bi` y `\end{itemize}` como `\ei`. Las definiciones precisas serían:

```
\newcommand{\bi}{\begin{itemize}}
\newcommand{\ei}{\end{itemize}}.
```

Ejemplo En la definición del siguiente comando se usa la instrucción `\fbox` (sección 3.15).

```
\newcommand{\Nota}{\fbox{\bf Nota}}
```

La instrucción `\Nota` producirá **Nota** como salida.

Ejemplo Las instrucciones `\normalsize`, `\footnotesize` y `\scriptsize` para tamaños de letra se pueden abreviar, por ejemplo, con `\normal`, `\Small` y `\SMALL`, respectivamente.

```
\newcommand{\normal}{\normalsize}
\newcommand{\Small}{\footnotesize}
\newcommand{\SMALL}{\scriptsize}.
```

Con estas definiciones, la gama de diez tamaños de letra en documentos L^AT_EX (sección 3.4) es más fácil de recordar y de usar: `\tiny`, `\SMALL`, `\Small`, `\small`, `\normal`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`.

3.24.2. Almacenamiento de cajas

Los comandos nuevos, definidos por medio de `\newcommand`, le ahorran tiempo al usuario pero no a L^AT_EX porque todo comando se ejecuta completamente cada vez que se invoca. L^AT_EX ofrece, como alternativa, la posibilidad de almacenar cajas de manera permanente para que sean usadas

solamente cuando se invoquen. L^AT_EX procesa estas cajas *una sola vez*: en el momento de ser definidas. Esto es útil para expresiones que se repitan múltiples veces en un documento; almacenadas como cajas, contribuyen a disminuir el tiempo de procesamiento, especialmente en documentos extensos.

Un nuevo espacio de almacenamiento en memoria para una caja se reserva en la forma

```
\newsavebox{\nombre}
```

donde el `nombre` tiene las mismas restricciones que para los comandos definidos con `\newcommand`: puede incluir una o más letras, mayúsculas o minúsculas, pero no dígitos ni otros símbolos del teclado. Si el comando `\nombre` ya existe, L^AT_EX emitirá un mensaje de error. El contenido de la nueva caja `\nombre` se almacena por medio de:

```
\savebox{\nombre}[ancho][justificación]{texto}
```

donde los argumentos opcionales tienen el mismo significado que para el comando `\makebox` (sección 3.15.1). El `ancho` puede ser cualquier dimensión T_EX, incluyendo fracciones de las dimensiones de la Tabla 3.5 (página 51).

También existe la versión sin argumentos opcionales de `\savebox`, a saber:

```
\sbox{\nombre}{texto}
```

Para imprimir el contenido de una caja almacenada como `\nombre`, es decir, para usar la caja, se escribe la instrucción `\usebox{\nombre}`.

Ejemplo Podemos definir y almacenar la caja `\obs` que consiste en la siguiente expresión: Observación. Los comandos de definición son

```
\newsavebox{\obs}
\sbox{\obs}{\underline{\sf Observación}}
```

Cada vez que queramos obtener la expresión 'Observación' escribimos en el documento fuente la instrucción `\usebox{\obs}`.

Ejemplo En un ejemplo de la página anterior se utilizó `\newcommand` para definir el comando `\Nota`, el cual produce la caja `\Nota`. Alternativamente, podemos proceder definiendo

```
\newsavebox{\Nota}
\sbox{\Nota}{\fbox{\bf Nota}}
```

Cada vez que queramos obtener la caja `\Nota` escribimos la instrucción `\usebox{\Nota}`.

3.25. Contadores y referencias cruzadas

Una de las características más sofisticadas y a la vez útiles de L^AT_EX es su habilidad para contar elementos de un documento tales como capítulos, secciones, páginas, figuras, ecuaciones, etc, y permitir luego hacer referencia a esos números utilizando solamente la *clave* o *etiqueta* (en términos T_EX-nicos, *label*) que el usuario utilizó para ellos. Los *números* que aparecen en expresiones como “en la sección 7.12”, “véase la Figura 4”, “en la página 58” pueden ser determinados *automáticamente* por L^AT_EX; de tal manera que si un documento sufre modificaciones posteriores con la inserción o remoción de nuevas secciones, figuras, ecuaciones, etc, L^AT_EX re-enumera los elementos del documento para producir los números correctos de las referencias cruzadas.

En la Tabla 3.7 aparecen los contadores (en inglés, *counters*) predefinidos por L^AT_EX. Los contadores de las tres primeras columnas coinciden con los entornos o comandos (sin el símbolo \) del mismo nombre, excepto `mpfootnote`, que es el contador de notas al pie de mini-páginas (sección 3.16). Los contadores de la última columna corresponden a los diferentes niveles del entorno `enumerate`.

<code>part</code>	<code>paragraph</code>	<code>figure</code>	<code>enumi</code>
<code>chapter</code>	<code>subparagraph</code>	<code>table</code>	<code>enumii</code>
<code>section</code>	<code>page</code>	<code>footnote</code>	<code>enumiii</code>
<code>subsection</code>	<code>equation</code>	<code>mpfootnote</code>	<code>enumiv</code>
<code>subsubsection</code>			

TABLA 3.7. Contadores básicos pre-definidos por L^AT_EX.

Cada vez que se utilice uno de los entornos o comandos referidos en la Tabla 3.7, podemos emitir (de manera opcional) la instrucción

`\label{clave}`

donde la *clave* es una etiqueta escogida enteramente por el usuario. Dicha *clave* puede ser cualquier secuencia de letras y números, y es de “uso interno”, es decir, no aparece impresa en el documento final. Podemos luego hacer referencia al *número* del elemento al cual se le asignó la *clave* usando el comando

`\ref{clave}`

Además, L^AT_EX “sabe” en qué página aparece el elemento en cuestión. Se puede hacer referencia a la página por medio de la instrucción

`\pageref{clave}`

Ejemplo Supongamos que vamos a iniciar una nueva sección en un documento escrito con el estilo `article`, denominada “Grupos solubles”. Para hacer referencia posterior a ella, usamos un nombre clave fácil de recordar (por ejemplo, ‘solubles’). Escribimos entonces en el archivo fuente

```
\section{Grupos solubles}\label{solubles}
```

y cuando queramos hacer referencia al número de la sección o a la página en la cual ésta se inicia escribimos, por ejemplo,

```
... en la sección \ref{solubles}, página \pageref{solubles}, se  
mencionó que ...
```

para obtener algo como

... en la sección 4, página 73, se mencionó que ...

Ejemplo La siguiente nota al pie de página, creada con `\footnote`, tiene una etiqueta, asignada con `\label{...}`.

```
...\footnote{\label{recientes} Investigaciones recientes en  
varias partes del mundo han mostrado ... } ...
```

Se puede hacer referencia posterior al número de la nota; por ejemplo,

```
... como se señaló en la nota \ref{recientes}, hay evidencia...
```

para obtener algo como

... como se señaló en la nota 14, hay evidencia ...

Algunas observaciones importantes sobre contadores y referencias cruzadas:

1. Es necesario procesar o compilar por lo menos dos veces un documento para obtener los valores correctos generados por las instrucciones `\ref` y `\pageref`. Esto se debe al hecho de que L^AT_EX utiliza los valores correspondientes al *anterior* procesamiento del documento, que pueden no reflejar aún los nuevos cambios. En particular, esto significa que al procesar un documento por primera vez no se obtendrá ningún número en absoluto al usar `\ref` o `\pageref`; en los sitios correspondientes L^AT_EX produce la señal de advertencia `???`. Dicha advertencia simplemente le indica al usuario que debe procesar el documento fuente una vez más.
2. Todos los contadores tienen valores enteros (pueden ser positivos o negativos) y se incrementan en 1 cada vez que aparece el comando o entorno respectivo. Los contadores pre-definidos tienen, por lo general, valor inicial 0. Así por ejemplo, el contador `chapter` de capítulos (en los estilos `book` y `report`) adquiere el valor 1 al aparecer por primera vez la instrucción `\chapter{...}`.

3. El valor actual de un *contador* determinado es controlado con la instrucción `\thecontador`. Por ejemplo, `\thechapter` imprime el valor actual del contador `chapter`, `\thefootnote` imprime el valor actual del contador `footnote`, `\thepage` imprime el valor actual del contador `page`, etc. Hay que observar que la instrucción `\thesection` en el estilo `book` produce dos números (por ejemplo, 3.5): el primero es el número del capítulo y el segundo el de la sección propiamente dicha. Similarmente, la instrucción `\thesubsection` genera tres números (por ejemplo, 3.5.2) correspondientes a los valores actuales de capítulo, sección y subsección (véase al respecto la sección 6.2).

Ejemplo Vamos a colocar la siguiente frase en el documento fuente del presente libro:

Estamos leyendo en la página `\thepage`, sección
`\thesection\` del Capítulo `\thechapter`.

Obtenemos exactamente lo siguiente:

Estamos leyendo en la página 77, sección 3.25 del Capítulo 3.

4. Podemos llevar un control o rastreo de todas las claves asignadas con `\label`, y de las referencias cruzadas hechas con `\ref` y `\pageref`, lo cual es muy útil en documentos extensos con numerosas claves. Para los detalles véase la sección 3.27.
5. El comando `\newtheorem` permite definir entornos para teoremas y estructuras similares, con su respectivo contador; para más detalles véase la sección 4.32. El usuario puede definir también sus propios contadores, tal como se explicará en la sección 3.25.3.
6. La instrucción `\label{clave}` puede aparecer en cualquier parte de un documento, no necesariamente en presencia de los entornos o comandos referidos en la Tabla 3.7, o de otros contadores. Esto sirve para hacer referencia a una página determinada del documento por medio de la instrucción `\pageref{clave}`.

3.25.1. Modificación de un contador

El valor actual de un *contador*, pre-definido por L^AT_EX o creado por el usuario (sección 3.25.3), se puede modificar por medio de las instrucciones:

`\setcounter{contador}{número}`. El *contador* adquiere el valor indicado por *número*, que debe ser un entero (positivo o negativo).

`\addtocounter{contador}{número}`. El valor actual del *contador* se incrementa en la cantidad indicada con *número*, que debe ser un entero (positivo o negativo).

`\stepcounter{contador}`. El valor actual del *contador* se incrementa en uno. Se usa principalmente en la definición de nuevos contadores por parte del usuario (véase la sección 3.25.3).

Ejemplo Queremos modificar el contador de capítulos para que el siguiente capítulo de nuestro documento (escrito con el estilo *book*) sea el número 6. Le asignamos al contador *chapter* el valor 5 (en caso de tener un valor diferente), de tal manera que al aparecer la instrucción `\chapter`, el capítulo creado tenga el número 6. Las instrucciones necesarias son:

```
\setcounter{chapter}{5}
\chapter{Título del capítulo}
```

Ejemplo Queremos que la primera página de un documento tenga el número 15. Después de `\begin{document}` colocamos la siguiente instrucción: `\setcounter{page}{15}`. A diferencia de los demás contadores, el contador *page* se incrementa después de que el número de la página ha sido generado; por lo tanto, el valor establecido con `\setcounter{page}` será el número que tendrá la página en la que aparece dicha instrucción.

Ejemplo Para que las notas al pie de página (sección 3.17) tengan numeración 0, 1, 2, ..., en vez de 1, 2, 3, ... colocamos la instrucción

```
\setcounter{footnote}{-1}
```

antes de escribir el primer comando `\footnote{...}`.

3.25.2. Estilos de numeración para contadores

LATEX emplea por defecto numeración arábiga para imprimir los valores de la mayoría de los contadores. Como excepciones podemos mencionar la numeración de apéndices en los estilos *book* y *report* (sección 6.4), que se numeran en la forma A, B, C,..., y las notas al pie de una mini-página (sección 3.16), que se numeran en la forma a, b, c, Hay cinco estilos básicos de numeración para contadores, a saber:

`\arabic{contador}`. Imprime los valores del *contador* con números arábigos: 1, 2, 3, ...

`\roman{contador}`. Imprime los valores del *contador* con los números romanos i, ii, iii, iv, Con la opción *spanish* de *babel* la numeración obtenida es I, II, III, IV, ... (véase la nota TEX-nica de la página 27).

\Roman{contador}. Imprime los valores del *contador* con números romanos en mayúscula: I, II, III, IV,

\alph{contador}. Imprime los valores del *contador* con letras minúsculas: a, b, c, d,

\Alph{contador}. Imprime los valores del *contador* con letras mayúsculas: A, B, C, D,

Para modificar el estilo de numeración de un determinado *contador* escribimos

```
\renewcommand{\thecontador}{nuevo estilo}
```

El nuevo estilo puede consistir en uno o más de los cinco estilos básicos, con o sin texto adicional.

Ejemplo \renewcommand{\thechapter}{\Roman{chapter}} hace que los capítulos aparezcan numerados con números romanos en mayúscula. Si se quiere que éste sea el tipo de numeración para todos los capítulos, el lugar más natural para colocar esta instrucción es el preámbulo del documento (en los estilos *book* o *report*).

Ejemplo Para que las secciones de un documento en el estilo *article* aparezcan numeradas con letras mayúsculas A, B, C, ... escribimos en el preámbulo: \renewcommand{\thesection}{\Alph{section}}.

Ejemplo Para que las páginas de un documento en el estilo *book* sean numeradas en la forma IV-28 (donde IV representa el número del capítulo y 28 el número de la página) colocamos la instrucción

```
\renewcommand{\thepage}{\Roman{chapter}--\arabic{page}}
```

en el preámbulo del documento. Esta instrucción controlará la numeración de las páginas pero en el título de la primera página de cada capítulo y en los encabezados aparecerán números arábigos, a menos que incluyamos también la instrucción \renewcommand{\thechapter}{\Roman{chapter}}.

3.25.3. Contadores creados por el usuario

Con la instrucción

```
\newcounter{cont}
```

LATEX crea un nuevo contador llamado *cont*. La expresión *cont* puede ser cualquier secuencia de letras que no corresponda a un contador ya existente.

Por lo general, el usuario crea nuevos contadores en conjunción con nuevos comandos o entornos. Esto se ilustra en el siguiente ejemplo.

Ejemplo Vamos a crear la instrucción `\nota` para producir notas numeradas: siempre que se escriba el comando `\nota` debe aparecer la expresión ‘NOTA’ seguida del número de la nota en negrita. Esto significa que la definición del nuevo comando `\nota` debe incluir un contador. A dicho contador lo llamaremos (de manera muy natural) ‘nota’, y utilizaremos la instrucción `\stepcounter` (sección 3.25.1) para incrementar en 1 el valor del contador con cada aparición de `\nota`. Una vez creado el contador `nota`, la instrucción `\thenota` produce el valor actual de tal contador (véase la observación 3 de la página 77 para la explicación de la instrucción `\thecontador`).

Los dos comandos requeridos para lograr este cometido son:

```
\newcounter{nota}
\newcommand{\nota}{\stepcounter{nota}{\bf NOTA \thenota.}}
```

Al escribir `\nota` por primera vez obtenemos **NOTA 1.** Al hacerlo por segunda vez obtenemos **NOTA 2.**, y así sucesivamente.

3.26. Referencias bibliográficas

Las normas académicas de mayor aceptación recomiendan la inclusión de una bibliografía o lista de referencias bibliográficas al final de todo documento. El estilo bibliográfico manejado por L^AT_EX es el llamado “estilo numérico”, corrientemente usado en la literatura matemática, en el que la lista de publicaciones y las citas bibliográficas en el interior del documento aparecen en la forma [1], [2], [3], En lugar de números se pueden usar etiquetas como [Scott], [Páez-Díaz], [A-D 1965], pero *siempre* aparecen entre paréntesis angulares. Este rígido estilo de L^AT_EX no es compatible con las normas más tradicionales de las ciencias naturales y las ciencias humanas, en las que predomina el denominado “estilo autor-año”: no hay etiquetas en la lista final de publicaciones y las citas internas se hacen sin paréntesis o con paréntesis circulares; por ejemplo, (Scott, 1992) o Scott (1992).

Si el usuario desea aprovechar las enormes ventajas que representa el uso del comando `\cite` de L^AT_EX para las citas bibliográficas, pero desea alejarse del *estilo* estándar de presentación, debe recurrir al programa BIBT_EX, el cual es también aconsejable para manejar bibliografías grandes. Con dicho programa, descrito detalladamente en la sección 6.10, se puede escoger un estilo bibliográfico por medio de la instrucción `\bibliographystyle{...}`.

Con el entorno `thebibliography` de L^AT_EX se puede elaborar manualmente una lista bibliográfica en la que las diferentes publicaciones estén

identificadas con “claves”. El usuario usa dichas claves, en el interior del documento, para hacer referencias cruzadas a la bibliografía, por medio del comando `\cite`.

El entorno `thebibliography` tiene la sintaxis:

```
\begin{thebibliography}{etiqueta más ancha}
\bibitem{clave1} Datos de la primera referencia bibliográfica
\bibitem{clave2} Datos de la segunda referencia bibliográfica
:
\end{thebibliography}
```

Los nombres `clave1`, `clave2`, ... son escogidos enteramente por el usuario; deben ser secuencias de letras y números. Para hacer referencia dentro del documento mismo al ítem denominado `clave` se usa la instrucción `\cite{clave}`. El argumento `{etiqueta más ancha}` es obligatorio ya que L^AT_EX necesita reservar suficiente espacio para las etiquetas del margen izquierdo de la bibliografía.

El uso de `claves` explícitas en los ítems bibliográficos es opcional. Así, si el usuario quiere presentar en la bibliografía una publicación determinada, sin citarla en el interior documento, puede escribir

```
\bibitem{} Datos bibliográficos
```

Las diferentes referencias bibliográficas se obtienen numeradas en la forma [1], [2], [3], ..., pero el usuario puede escoger otras etiquetas escribiendo

```
\bibitem[etiqueta]{clave}
```

La instrucción `\begin{thebibliography}` produce además un título para la bibliografía: **References** en el estilo `article` y **Bibliography** en los estilos `book` y `report`. Al usar el paquete `babel`, opción `spanish`, los títulos obtenidos son **Referencias** y **Bibliografía**, respectivamente. Incluso sin usar el paquete `babel`, podemos forzar otro título para la lista bibliográfica final, re-definiendo los parámetros que controlan dichos nombres; en el estilo `article` tal parámetro es `\refname` y en los estilos `book` y `report`, `\bibname` (véase al respecto la sección 6.5). Se cambian utilizando `\renewcommand`, en la siguiente forma:

```
\renewcommand{\refname}{Título deseado} (Estilo article)
```

```
\renewcommand{\bibname}{Título deseado} (Estilos book y report)
```

La gran ventaja de usar `\cite` para las citas bibliográficas en el interior del documento es que si la bibliografía es modificada por el usuario, añadiendo o eliminando ítems, L^AT_EX actualiza correctamente la numeración y las referencias cruzadas (al igual que con los comandos `\label` y `\ref` presentados en la sección 3.25).

Ejemplo La siguiente es una lista bibliográfica típica en un documento escrito con el estilo `article`. El número 9 que aparece en el argumento `{9}` indica que hay menos de 10 items en la lista. Recalcamos que dicho argumento hace alusión a la *etiqueta más ancha*, no al número de items. Obsérvese que para la primera publicación no se usó ninguna *clave* (`\bibitem{}`) ya que el autor no tiene intenciones de citarla en el interior del documento.

```
\renewcommand{\refname}{Bibliografía}
\begin{thebibliography}{9}
\bibitem{} Frey, G. \textit{Links between stable elliptic
curves and certain diophantine equations}, Annales
universitatis Saraviensis, \textbf{1} (1986), 1--40.
\bibitem{wiles1} Wiles, Andrew, \textit{Modular curves and certain
class group}, Invent. Math. \textbf{58} (1980), 1--35.
\bibitem{wiles2} Wiles, Andrew, \textit{Modular elliptic curves
and Fermat's Last Theorem}, Annals of Mathematics \textbf{142}
(1995), 443--551.
\bibitem{taylor-wiles} Taylor, Richard and Wiles, Andrew,
\textit{Ring-theoretic properties of certain Hecke algebras},
Annals of Mathematics \textbf{142} (1995), 553--572.
\end{thebibliography}
```

Lo anterior produce la siguiente lista al final del documento:

Bibliografía

- [1] Frey, G. *Links between stable elliptic curves and certain diophantine equations*, Annales universitatis Saraviensis, 1 (1986), 1–40.
- [2] Wiles, Andrew, *Modular curves and certain class group*, Invent. Math. **58** (1980), 1–35.
- [3] Wiles, Andrew, *Modular elliptic curves and Fermat's Last Theorem*, Annals of Mathematics **142** (1995), 443–551.
- [4] Taylor, Richard and Wiles, Andrew, *Ring-theoretic properties of certain Hecke algebras*, Annals of Mathematics **142** (1995), 553–572.

Dentro del documento se hace referencia a estas publicaciones con el comando `\cite`; por ejemplo, la siguiente frase en el archivo de entrada

La parte esencial de la demostración aparece en `\cite{wiles2}` y las correcciones necesarias se encuentran en `\cite{taylor-wiles}`.

producirá lo siguiente en el documento final:

La parte esencial de la demostración aparece en [3] y las correcciones necesarias se encuentran en [4].

Ejemplo La bibliografía del ejemplo anterior se puede modificar con etiquetas no-numéricas. Obsérvese que la etiqueta más ancha, T-W, se incluye en el argumento de \begin{thebibliography}.

```
\renewcommand{\refname}{Referencias bibliográficas}
\begin{thebibliography}{T-W}
\bibitem[F]{wiles1} Frey, G. \textit{Links between stable elliptic curves and certain diophantine equations}, Annales universitatis Saraviensis, \textbf{1} (1986), 1--40.
\bibitem[Wa]{wiles1} Wiles, Andrew, \textit{Modular curves and certain class group}, Invent. Math. \textbf{58} (1980), 1--35.
\bibitem[Wb]{wiles2} Wiles, Andrew, \textit{Modular elliptic curves and Fermat's Last Theorem}, Annals of Mathematics \textbf{142} (1995), 443--551.
\bibitem[T-W]{taylor-wiles} Taylor, Richard and Wiles, Andrew, \textit{Ring-theoretic properties of certain Hecke algebras}, Annals of Mathematics \textbf{142} (1995), 553--572.
\end{thebibliography}
```

Referencias bibliográficas

- [F] Frey, G. *Links between stable elliptic curves and certain diophantine equations*, Annales universitatis Saraviensis, **1** (1986), 1–40.
- [Wa] Wiles, Andrew, *Modular curves and certain class group*, Invent. Math. **58** (1980), 1–35.
- [Wb] Wiles, Andrew, *Modular elliptic curves and Fermat's Last Theorem*, Annals of Mathematics **142** (1995), 443–551.
- [T-W] Taylor, Richard and Wiles, Andrew, *Ring-theoretic properties of certain Hecke algebras*, Annals of Mathematics **142** (1995), 553–572.

Al escribir \cite{wiles2} en el documento fuente, se obtendrá [Wb] como salida, y \cite{taylor-wiles} dará lugar a [T-W].

El comando \cite tiene dos interesantes facetas adicionales:

- Se pueden citar varias referencias simultáneamente, separándolas con comas. Así, \cite{wiles1,taylor-wiles} (ejemplo inmediatamente anterior) producirá [Wa,T-W].
- Se pueden incluir anotaciones variadas en la cita misma por medio del comando opcional de \cite: \cite[anotación]{...}. En el ejemplo inmediatamente anterior, la instrucción \cite[pag. 8]{wiles1} dará lugar a [Wa, pag. 8] mientras que \cite[Remark 5]{wiles2} producirá como salida [Wb, Remark 5].

3.27. Rastreo de claves asignadas por el usuario

Las claves asignadas con `\label{clave}` para las referencias cruzadas (sección 3.25) y `\bibitem[...]{clave}` para los items bibliográficos (sección 3.26), junto con los comandos paralelos `\ref`, `\pageref` y `\cite`, son una gran ayuda en la escritura de documentos. Ya no es necesario memorizar los números asignados a secciones, ecuaciones, teoremas, referencias bibliográficas, etc, y lo más importante: el usuario puede introducir cambios en el documento, con la seguridad de que la numeración y las referencias cruzadas serán actualizadas correctamente por L^AT_EX. Pero, como precio, ¡hay que memorizar las palabras clave utilizadas! Y esto puede ser fuente de errores o confusiones, especialmente en documentos extensos.

Para ayudarnos a seguir el rastro de claves hay dos útiles recursos, descritos a continuación.

3.27.1. Rastreo con el paquete `showkeys`

Invocando el paquete `showkeys`¹⁶ se produce una versión provisional del documento principal ‘—.dvi’ en el que se señalan, con recuadros en las márgenes o con texto intercalado entre los renglones, los sitios en los que están escritos los comandos `\label`, `\ref`, `\pageref`, `\bibitem` y `\cite`.

El paquete se carga en la forma usual, `\usepackage{showkeys}`, y al procesar el documento fuente se obtiene lo siguiente:

1. Los argumentos de los comandos `\ref`, `\pageref` y `\cite` aparecen escritos en los sitios originales, en letra más pequeña, sobre una línea horizontal colocada en el espacio que separa los renglones, en la forma ilustrada en el presente párrafo con una `clave` particular.
2. Aparecen recuadros en los márgenes con los nombres de las claves asignadas con `\label` y `\bibitem`. Puesto que `showkeys` trata de no alterar el formato del documento, algunas veces los recuadros se pueden traslapar con el texto principal.

ejem5

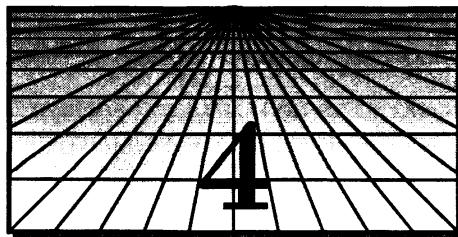
Ejemplo En este párrafo hemos colocado la clave `\label{ejem5}`, para una futura referencia, y hemos escrito la instrucción `\ref{s:historia}` para hacer referencia a una sección anterior del documento a la cual le habíamos asignado la clave ‘`s:historia`’ con `\label{s:historia}`. Al procesar el documento con el paquete `showkeys`, obtenemos una caja en el margen con la clave `ejem5` y el texto intercalado con el argumento de `\ref`.

¹⁶Paquete estándar de L^AT_EX 2_E, escrito por David Carlisle.

3.27.2. Rastreo con el archivo **lablst.tex**

En las distribuciones corrientes de L^AT_EX 2_< se incluye **lablst.tex**, archivo que produce, una vez procesado, la lista de las claves definidas por el usuario con **\label** y **\bibitem**, a partir de la información contenida en el documento auxiliar ‘**—.aux**’. El archivo final **lablst.dvi** se puede imprimir, como referencia. El siguiente es el procedimiento concreto utilizado:

1. Ubicar el archivo **lablst.tex** en el computador local; la ubicación exacta depende de la implementación de T_EX y L^AT_EX (ya sea comercial o de dominio público) que el usuario disponga.
2. Hacer *una copia* del archivo **lablst.tex** y colocarla en el mismo subdirectorio o carpeta en el que se encuentra el documento fuente.
3. Procesar **lablst.tex** como un documento L^AT_EX normal, escribiendo los datos que el programa pida de manera interactiva: el nombre del documento fuente (sin la extensión **.tex**), el estilo utilizado en **\documentclass{ }** y los paquetes cargados con **\usepackage{ }**. De hecho, no es necesario escribir *todos* los paquetes utilizados, solamente aquéllos que definan comandos usados en títulos de capítulos, secciones, etc (entre ellos el paquete **inputenc**, empleado para las letras acentuadas). Los nombres de los diferentes paquetes se separan con comas. Si en el documento se ha usado el paquete **babel** con la opción **spanish**, por ejemplo, se debe escribir **spanish** en la lista; no se debe escribir **babel**.
4. El documento **lablst.dvi** obtenido contiene la lista de las claves asignadas, bajo los títulos *Logical labels within sections* y *Bibliography logical labels*, junto con los números de las páginas en los que aparecen las definiciones originales.



Matemáticas

4.1. Modo matemático

Como se mencionó en la sección 2.1, uno de los tres modos en los que L^AT_EX procesa el texto de entrada es el denominado *modo matemático* (en inglés, *math mode*). Las dos maneras más simples de invocar el modo matemático son:

$\$ \dots \$$ para expresiones matemáticas no desplegadas,

$\backslash [\dots \backslash]$ para expresiones matemáticas desplegadas y centradas.

Por ejemplo, $\$f(x,y)=ax+by\$$ produce $f(x,y) = ax + by$ mientras que la instrucción $\backslash [f(x,y)=ax+by.\backslash]$ da lugar a:

$$f(x,y) = ax + by.$$

Equivalentemente, podemos invocar el modo matemático por medio de los entornos explícitos:

`\begin{math} \dots \end{math}`

para expresiones matemáticas no desplegadas, y

`\begin{displaymath} \dots \end{displaymath}`

para expresiones matemáticas desplegadas y centradas. Obviamente, las formas $\dots \$$ y $\backslash [\dots \backslash]$ son las más cómodas.

Los lectores con T_EX-periencia previa pueden estar acostumbrados a usar $\$\dots \$$ para desplegar expresiones matemáticas. No se recomienda su uso en documentos L^AT_EX (aunque no se recibirá ningún mensaje de error al hacerlo) debido a que la distribución del espacio en blanco alrededor del despliegue no es satisfactoria en todos los casos.

El entorno

`\begin{equation} \dots \end{equation}`

también invoca el modo matemático y produce una expresión desplegada, centrada y numerada en el margen derecho. La numeración de fórmulas es un tópico que se trata en detalle en la secciones 4.30 y 4.31.

En modo matemático, L^AT_EX ignora (en el archivo fuente) todos los espacios. Por ejemplo, `$a + b = c$` produce lo mismo que `$a+b=c$` y lo mismo que `$ a + b = c $`, a saber: $a + b = c$. Este proceder tiene una justificación fundamental: T_EX y L^AT_EX controlan automáticamente el espaciamiento de los símbolos en el documento final, así como el tamaño y el tipo de letra. El usuario sólo debe cuidarse por respetar la sintaxis de los diferentes comandos matemáticos; L^AT_EX se encarga de traducirlos para generar expresiones con los más altos estándares tipográficos.

Ejemplo En la afirmación exhibida a continuación aparecen algunos de los símbolos matemáticos listados en la sección 4.3.

Para todo $\varepsilon > 0$ existe $\delta > 0$ tal que $|x - a| < \delta$ implica $|f(a) - f(x)| < \varepsilon$.

Para todo `$\backslash varepsilon > 0$` existe `$\delta > 0$` tal que `$|x-a| < \delta$` implica `$|f(a)-f(x)| < \varepsilon$`.

4.2. Los paquetes `amsmath` y `amssymb`

El software anteriormente conocido como *AMS-T_EX* y *AMS-L^AT_EX* ha sido incorporado al ambiente L^AT_EX 2_ε como un paquete, `amsmath`, al cual se accede colocando la instrucción

`\usepackage{amsmath}`

en el preámbulo del documento. Una versión preliminar de este paquete había sido distribuida como `amstex`, paquete obsoleto desde 1996.

El paquete `amsmath` invoca a su vez una serie de paquetes subsidiarios, todos ellos incluidos en las distribuciones típicas de L^AT_EX 2_ε. Uno de los paquetes que acompañan a `amsmath` es el paquete de símbolos `amssymb`, que contiene la extensa colección de símbolos matemáticos de la AMS, conocida como *AMS Fonts*. Debe ser cargado por separado, en la forma usual: `\usepackage{amssymb}`. Este paquete invoca a su vez al paquete `amsfonts`. La lista completa de todos los símbolos de estas colecciones aparece en la sección 4.3.

 Se advierte al lector que la mayoría de las construcciones y símbolos matemáticos que se presentan en este capítulo requieren de `amsmath` y sus paquetes subsidiarios (versión 2.0 o posterior). La implementación MiK_TE_X incluida en el CD adjunto contiene las versiones más recientes.

4.3. Galería de símbolos matemáticos

Los símbolos que aparecen en las tablas de esta sección se deben usar en modo matemático; algunos pertenecen al paquete `amssymb` o al pequeño paquete `latexsym`, lo cual se indica explícitamente en cada caso. Estos dos paquetes se pueden cargar simultáneamente escribiendo:

```
\usepackage{amssymb, latexsym}
```

Los demás símbolos pertenecen, o bien a la colección básica de símbolos T_EX, o bien al paquete `amsmath`.

Relaciones binarias básicas					
\leq	<code>\leq</code> ó <code>\le</code>	\geq	<code>\geq</code> ó <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>

Operadores binarios básicos					
\cdot	<code>\cdot</code>	\triangleleft	<code>\triangleleft</code>	\triangleright	<code>\triangleright</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\star	<code>\star</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\ast	<code>\ast</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\circ	<code>\circ</code>
\vee	<code>\vee</code> ó <code>\lor</code>	\wedge	<code>\wedge</code> ó <code>\land</code>	\bullet	<code>\bullet</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\diamond	<code>\diamond</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\uplus	<code>\uplus</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\amalg	<code>\amalg</code>
\triangle	<code>\bigtriangleup</code>	\bigtriangledown	<code>\bigtriangledown</code>	\wr	<code>\wr</code>
\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>	\div	<code>\div</code>

Operadores binarios del paquete `amssymb`

\dotplus	<code>\dotplus</code>	\centerdot	<code>\centerdot</code>
\ltimes	<code>\ltimes</code>	\rtimes	<code>\rtimes</code>
\Cup	<code>\Cup</code>	\Cap	<code>\Cap</code>
\veebar	<code>\veebar</code>	\barwedge	<code>\barwedge</code>
\boxplus	<code>\boxplus</code>	\boxminus	<code>\boxminus</code>
\boxtimes	<code>\boxtimes</code>	\boxdot	<code>\boxdot</code>
\curlyvee	<code>\curlyvee</code>	\curlywedge	<code>\curlywedge</code>
\intercal	<code>\intercal</code>	\divideontimes	<code>\divideontimes</code>
\smallsetminus	<code>\smallsetminus</code>	\doublebarwedge	<code>\doublebarwedge</code>
\circleddash	<code>\circleddash</code>	\circledcirc	<code>\circledcirc</code>
\leftthreetimes	<code>\leftthreetimes</code>	\circledast	<code>\circledast</code>
\rightthreetimes	<code>\rightthreetimes</code>		

Para tener acceso a estos símbolos es necesario escribir la instrucción `\usepackage{amssymb}` en el preámbulo del documento.

Relaciones y operadores binarios del paquete `latexsym`

\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\Join	<code>\Join</code>
\lhd	<code>\lhd</code>	\rhd	<code>\rhd</code>	\Box	<code>\Box</code>
\unlhd	<code>\unlhd</code>	\unrhd	<code>\unrhd</code>	\Diamond	<code>\Diamond</code>

Para tener acceso a estos símbolos es necesario escribir la instrucción `\usepackage{latexsym}` en el preámbulo del documento. Estos símbolos del paquete `latexsym` también están disponibles en el paquete `amssymb` y tienen los mismos nombres, a excepción de `\unlhd` y `\unrhd`, que corresponden a `\vartriangleleft` y `\vartriangleright`, respectivamente.

Esquinas

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>
\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>

Estos símbolos requieren el paquete `amssymb`.

Flechas

\leftarrow	<code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\rightsquigarrow	<code>\leadsto</code> ^a	\updownarrow	<code>\updownarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code> ó <code>\iff</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookleftarrow	<code>\hookleftarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\Updownarrow	<code>\Updownarrow</code>

^a El símbolo \rightsquigarrow pertenece al paquete `latexsym`.

Flechas del paquete `amssymb`

\upharpoonleft	<code>\upharpoonleft</code>	\upharpoonright	<code>\upharpoonright</code>
\downharpoonleft	<code>\downharpoonleft</code>	\downharpoonright	<code>\downharpoonright</code>
\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>
\leftleftarrows	<code>\leftleftarrows</code>	\rightrightarrows	<code>\rightrightarrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>
\upuparrows	<code>\upuparrows</code>	\downdownarrows	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>
\leftrightsquigarrow	<code>\leftrightsquigarrow</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\multimap	<code>\multimap</code>		

Para tener acceso a estos símbolos es necesario escribir la instrucción `\usepackage{amssymb}` en el preámbulo del documento.

Relaciones binarias del paquete <code>amssymb</code>			
\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>
\lll	<code>\lll</code>	\ggg	<code>\ggg</code>
\lessim	<code>\lessim</code>	\gtrsim	<code>\gtrsim</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>
\lesseqgqtr	<code>\lesseqgqtr</code>	\gtreqqless	<code>\gtreqqless</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>
\subsetneqq	<code>\subsetneqq</code>	\supsetneqq	<code>\supsetneqq</code>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>
\backepsilon	<code>\backepsilon</code>	\sqsubset	<code>\sqsubset</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\blacktriangleright	<code>\blacktriangleright</code>

Para tener acceso a estos símbolos es necesario escribir la instrucción `\usepackage{amssymb}` en el preámbulo del documento.

Relaciones y flechas negadas del paquete <code>amssymb</code>			
$\not\leq$	<code>\nless</code>	$\not\geq$	<code>\ngtr</code>
$\not\leqslant$	<code>\lneq</code>	$\not\geqslant$	<code>\gneq</code>
$\not\leqq$	<code>\lneq</code>	$\not\geqq$	<code>\ngeq</code>
$\not\leqslant$	<code>\lneqslant</code>	$\not\geqslant$	<code>\ngeqslant</code>
$\not\leqslant$	<code>\lneqq</code>	$\not\geqq$	<code>\gneqq</code>
$\not\parallel$	<code>\lvertneqq</code>	$\not\geqslant$	<code>\gvertneqq</code>
$\not\leqslant$	<code>\lneqq</code>	$\not\geqslant$	<code>\ngeqq</code>
$\not\sim$	<code>\lnsim</code>	$\not\approx$	<code>\gnsim</code>
$\not\approx$	<code>\lnapprox</code>	$\not\approx$	<code>\gnapprox</code>
$\not\prec$	<code>\nprec</code>	$\not\succ$	<code>\nsucc</code>
$\not\preceq$	<code>\npreceq</code>	$\not\succeq$	<code>\nsuccceq</code>
$\not\precneq$	<code>\precneqq</code>	$\not\succneq$	<code>\succneqq</code>
$\not\precnsim$	<code>\precnsim</code>	$\not\sim$	<code>\succnsim</code>
$\not\precnapprox$	<code>\precnapprox</code>	$\not\approx$	<code>\succnapprox</code>
$\not\subsetneq$	<code>\subsetneq</code>	$\not\supsetneq$	<code>\supsetneq</code>
$\not\varsubsetneq$	<code>\varsubsetneq</code>	$\not\varsupsetneq$	<code>\varsupsetneq</code>
$\not\subsetneqq$	<code>\subsetneqq</code>	$\not\supsetneqq$	<code>\supsetneqq</code>
$\not\leftarrow$	<code>\leftarrow</code>	$\not\rightarrow$	<code>\rightarrow</code>
$\not\Leftarrow$	<code>\Leftarrow</code>	$\not\Rightarrow$	<code>\Rightarrow</code>
$\not\in$	<code>\notin</code>	\neq	<code>\neq</code>

Para tener acceso a estos símbolos es necesario escribir la instrucción `\usepackage{amssymb}` en el preámbulo del documento. El comando `\not` de L^AT_EX permite negar símbolos arbitrarios; por ejemplo, `\not\in` produce \notin (que es ligeramente diferente del símbolo `\notin`: $\not\in$).

Miscelánea de símbolos		
\hbar \hbar	\imath \imath	\jmath \jmath
\Re \Re	\Im \Im	∂ \partial
\forall \forall	\exists \exists	\surd \surd
$'$ '	\prime \prime	\emptyset \emptyset
∇ \nabla	\triangle \triangle	\Box \Box
\bot \bot	\top \top	\angle \angle
\diamondsuit \diamondsuit	\heartsuit \heartsuit	\clubsuit \clubsuit
\spadesuit \spadesuit	\flat \flat	\natural \natural
\sharp \sharp	\wp \wp	\neg \neg ó \lnot
∞ \infty	\diamond \diamond	ℓ \ell
\smallint \smallint	\mho \mho ^a	\And \And ^b
\pm \pm	\mp \mp	

^a Requiere el paquete amsmath

^b Requiere el paquete latexsym

Miscelánea de símbolos del paquete amssymb		
\eth \eth	\hslash \hslash	\Bbbk \Bbbk
\nexists \nexists	\blacksquare \blacksquare	\circledS \circledS
\square \square	\measuredangle \measuredangle	\complement \complement
\lozenge \lozenge	\varnothing \varnothing	\Game \Game
\triangledown \triangledown	\blacklozenge \blacklozenge	\bigstar \bigstar
\vartriangle \vartriangle	\diagup \diagup	\sphericalangle \sphericalangle
\Finv \Finv	\diagdown \diagdown	\backprime \backprime
\checkmark \checkmark	\maltese \maltese	\yen \yen
\blacktriangle \blacktriangle	\blacktriangledown \blacktriangledown	

Para tener acceso a estos símbolos es necesario escribir la instrucción `\usepackage{amssymb}` en el preámbulo del documento.

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

^a Requiere el paquete `amssymb`

Letras griegas mayúsculas				
Δ	<code>\Delta</code>	Δ	<code>\varDelta</code>	^a
Λ	<code>\Lambda</code>	Λ	<code>\varLambda</code>	^a
Π	<code>\Pi</code>	Π	<code>\varPi</code>	^a
Ψ	<code>\Psi</code>	Ψ	<code>\varPsi</code>	^a
Φ	<code>\Phi</code>	Φ	<code>\varPhi</code>	^a
Σ	<code>\Sigma</code>	Σ	<code>\varSigma</code>	^a
Θ	<code>\Theta</code>	Θ	<code>\varTheta</code>	^a
Υ	<code>\Upsilon</code>	Υ	<code>\varUpsilon</code>	^a
Ξ	<code>\Xi</code>	Ξ	<code>\varXi</code>	^a
Ω	<code>\Omega</code>	Ω	<code>\varOmega</code>	^a
Γ	<code>\Gamma</code>	\digamma	<code>\digamma</code>	^b
		Γ	<code>\varGamma</code>	^a

^a Requiere el paquete `amsmath`

^b Requiere el paquete `amssymb`

Letras hebreas			
\aleph	<code>\aleph</code>	\beth	<code>\beth</code>
\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>

Estos símbolos, excepto `\aleph`, requieren el paquete `amssymb`.

4.4. Recomendaciones para la correcta escritura en matemáticas

Las herramientas matemáticas de L^AT_EX y paquetes como **amsmath**, **amsthm**, **amssymb**, **amscd** y **pb-diagram**, presentadas a lo largo de este extenso capítulo, son múltiples y sofisticadas. En esta sección presentamos algunas recomendaciones generales, resaltando ciertos errores típicos que conviene evitar.

1. Siguiendo normas tipográficas tradicionales, los nombres de funciones básicas como cos, log, exp, y operadores como lim, sup, inf, etc se escriben en letra romana (vertical). L^AT_EX tiene comandos para las funciones y los operadores más frecuentes; los citados arriba, por ejemplo, se obtienen con `\cos`, `\log`, `\exp`, `\lim`, `\sup` e `\inf`. El usuario puede además definir sus propios comandos para nuevas funciones y operadores. Detalles: secciones 4.10 y 4.18.

<code>\$\cos x+\cos y\$</code>	<i>cosx + cosy</i>	Incorrecto
<code>\\cos x+\\cos y\$</code>	<i>cos x + cos y</i>	Correcto

2. Antes y después de una expresión matemática desplegada, ya sea con `\[... \]`, con `\begin{equation} ... \end{equation}` o con entornos similares, L^AT_EX deja un pequeño espacio vertical adicional. Por consiguiente, no se debe añadir más espacio ni antes ni después de los despliegues. En particular, no se deben dejar renglones en blanco alrededor de despliegues:

Correcto	Incorrecto
<i>Texto anterior</i> <code>\[</code> <i>Expresión matemática</i> <code>\]</code> <i>Texto posterior</i>	<i>Texto anterior</i> <code>\[</code> <i>Expresión matemática</i> <code>\]</code> <i>Texto posterior</i>

Hay una razón adicional por la cual se deben evitar los renglones en blanco antes y después de las expresiones matemáticas desplegadas: tradicionalmente el párrafo que sigue a un despliegue no lleva sangría mientras que un renglón en blanco hace que el párrafo siguiente se inicie con sangría. Detalles: sección 3.8.

3. Para dividir fórmulas o expresiones matemáticas muy grandes en dos o más renglones hay que seguir normas de separación y alineación precisas. Una fórmula matemática grande *no* se separa simplemente dividiéndola en varios `\[\cdots\]` consecutivos. Detalles: sección 4.30.
4. El tipo de letra utilizado por L^AT_EX en modo matemático difiere del utilizado en el discurso usual. Para incluir texto corriente en expresiones matemáticas desplegadas se debe usar el comando `\text{...}`. Detalles: sección 4.11.
5. Recomendaciones sobre el uso correcto de ciertos símbolos y expresiones matemáticas:

- (a) No confundir los símbolos `< y >`, usados para ‘menor’ y ‘mayor’, respectivamente, con los paréntesis angulares `(y)`, usados para producto interno, generación de estructuras, etc.

<code>\$<u,v>\$</code>	$\langle u, v \rangle$	Incorrecto
<code>\$\langle u,v\rangle\$</code>	$\langle u, v \rangle$	Correcto

- (b) No confundir la barra vertical `|`, usada para valor absoluto en expresiones como $|a + b| \leq |a| + |b|$ (`||a+b||\leq ||a||+||b||`) con la relación binaria `\mid` que produce una barra vertical `|` rodeada de espacio horizontal adicional a izquierda y derecha. En general, los símbolos para relaciones binarias (véanse las tablas de la sección 4.3) producen automáticamente espacio extra a su alrededor.

<code>\$\{x P(x)\}\$</code>	$\{x P(x)\}$	Incorrecto
<code>\$\{x\mid P(x)\}\$</code>	$\{x \mid P(x)\}$	Correcto

n divide a m se denota $n|m$
`n divide a m se denota $n|m$`

n divide a m se denota $n \mid m$
`n divide a m se denota $n\mid m$`

- (c) Hay que distinguir entre los comandos `\cdots` y `\ldots` para puntos suspensivos horizontales, los cuales producen puntos centrados e inferiores, respectivamente. Se deben usar de acuerdo con la posición de los símbolos adyacentes. Detalles: sección 4.6.

<code>\$1+2+\ldots+n\$</code>	$1 + 2 + \dots + n$	Incorrecto
<code>\$1+2+\cdots+n\$</code>	$1 + 2 + \cdots + n$	Correcto

6. L^AT_EX escoge automáticamente el tamaño de los símbolos en las expresiones matemáticas. El usuario sólo debe cambiar el tamaño en ocasiones muy especiales y con mucha cautela. Detalles: sección 4.26.
 7. En la escritura matemática tradicional se enumeran teoremas, lemas, definiciones, ecuaciones, etc. Para obtener el máximo provecho de las facilidades de edición de L^AT_EX, hay que acostumbrarse desde un comienzo a usar el mecanismo de los contadores, evitando la numeración explícita. Los comandos requeridos son: `\label`, `\ref` y `\pageref`. Si el documento fuente sufre modificaciones posteriores, L^AT_EX actualiza automáticamente la numeración y las referencias cruzadas, lo cual no sucede si se emplea la numeración manual. Detalles: secciones 3.25 y 4.30.

Por el Teorema 5 se concluye ... Incorrecto

Por el Teorema \ref{clave} se concluye ... Correcto

8. En inglés, tradicionalmente se usa el ‘punto decimal’ (como en 3.1416) pero en el mundo hispanohablante se prefiere a veces la ‘coma decimal’ (como en 3,1416). En modo matemático, el valor pre-determinado por la opción `spanish` del paquete `babel` (posterior al año 2000) es la coma decimal, pero se puede forzar el punto decimal en todo el documento escribiendo la instrucción `\decimalpoint` en el preámbulo.

4.5. Subíndices y superíndices

Los superíndices se obtienen con el símbolo `^` y los subíndices con `_`. Estos caracteres del teclado están reservados por L^AT_EX exclusivamente para esta función y solamente se pueden usar en modo matemático.

Ejemplo

$\$e^y\$$	e^y	$\$A_a b\$$	A_a^b
$\$x^{y^2}\$$	x^{y^2}	$\$A^b a\$$	A_a^b
$\$x^{y_2}\$$	x^{y_2}	$\$F_i * \$$	F_i^*
$\$x^{2y}\$$	x^{2y}	$\$(a+b)^n\$$	$(a+b)^n$
$\$xF_y\$$	$x F_y$	$\$(a+b)^{n+1}\$$	$(a+b)^{n+1}$

4.6. Puntos suspensivos

En modo matemático, los comandos básicos para puntos suspensivos son:

\ldots	\vdots
\cdots	\ddots

\cdots produce puntos suspensivos centrados \cdots y \ldots da lugar a los puntos \dots (la l representa ‘lower’).

El paquete **amsmath** tiene varios comandos que producen puntos suspensivos horizontales, según el contexto, a saber:

\dotsc	Puntos en presencia de comas.
\dotsb	Puntos en presencia de operadores o relaciones binarias.
\dotsm	Puntos en presencia de multiplicaciones o productos.
\dotso	Puntos en otros contextos.

Ejemplo

$\$A_1,A_2,A_3,\dotsc\$$	A_1, A_2, A_3, \dots
$\$A_1+A_2+A_3+\dotsb\$$	$A_1 + A_2 + A_3 + \dots$
$\$A_1A_2A_3\dotsm\$$	$A_1 A_2 A_3 \dots$

4.7. Fracciones

La instrucción $\frac{\dots}{\dots}$ para fracciones tiene dos argumentos: numerador y denominador. Las fracciones se obtienen en dos tamaños: tamaño texto (algo como $\frac{a}{b}$) y en tamaño de despliegue (algo como $\frac{a}{b}$). L^AT_EX escoge el tamaño dependiendo de la situación concreta en la que aparece el comando \frac , pero con el paquete **amsmath** podemos forzar el tamaño: $\tfrac{\dots}{\dots}$, para tamaño texto, y $\dfrac{\dots}{\dots}$, para tamaño de despliegue.

Ejemplo

$\$\\frac{a}{b}\$$	$\frac{a}{b}$	(tamaño texto)
$\$[\\frac{a}{b}]\$$	$\frac{a}{b}$	(tamaño de despliegue)
$\$\\tfrac{1}{1+x^2}\$$	$\frac{1}{1+x^2}$	(tamaño texto)
$\$\\dfrac{1}{1+x^2}\$$	$\frac{1}{1+x^2}$	(tamaño de despliegue)

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

$$\$ \frac{x/2+y/2}{1+x/y} \$ \quad \frac{x/2 + y/2}{1 + x/y}$$

Con respecto a la última fracción, debemos recordar que el símbolo `/` no es un símbolo reservado para L^AT_EX y se puede usar sin restricciones tanto en modo normal como en modo matemático.

Para fracciones continuas tenemos el comando `\cfrac{...}{...}`, el cual produce mejores resultados que el uso directo de `\frac`.

Ejemplo

$$\frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$$

```
\[
  \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cdots}}}
]
```

Si usamos `\frac` en lugar de `\cfrac` obtenemos lo siguiente:

$$\frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$$

4.8. Coeficientes binomiales

Los coeficientes binomiales se obtienen con el comando de dos argumentos `\binom{...}{...}`, de comportamiento similar a `\frac{...}{...}`. También existen las versiones `\tbinom` y `\dbinom` para forzar el tamaño texto y el de despliegue, respectivamente.

Ejemplo

$$\$ \binom{n}{k} \$ \quad \binom{n}{k}$$

$$\$ \dbinom{m}{i+j} \$ \quad \binom{m}{i+j}$$

$$\$ [\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}] \$ \quad \binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

4.9. Raíces

Para raíces se usa la instrucción `\sqrt[índice]{radicando}`; el argumento opcional *índice* se emplea para raíces no cuadradas.

Ejemplo

<code>\$\sqrt[\beta]{R}\$</code>	$\sqrt[\beta]{R}$
<code>\$\sqrt{a+5}\$</code>	$\sqrt{a + 5}$
<code>\$\sqrt[n]{1+\sqrt{1+x}}\$</code>	$\sqrt[n]{1 + \sqrt{1 + x}}$
<code>\$\sqrt[3]{\frac{x}{x+1}}\$</code>	$\sqrt[3]{\frac{x}{x+1}}$
<code>\sqrt[\beta]{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}}</code>	$\sqrt[\beta]{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}}$

Como se puede apreciar en $\sqrt[\beta]{R}$, la colocación del índice de la raíz no es siempre satisfactoria. El paquete `amsmath` tiene instrucciones para ajustar la posición del índice de la raíz:

- `\leftroot{n}` mueve el índice de la raíz *n* micro-espacios hacia la izquierda (si *n* es positivo) y hacia la derecha (si *n* es negativo).
- `\uproot{n}` mueve el índice de la raíz *n* micro-espacios hacia arriba (si *n* es positivo) y hacia abajo (si *n* es negativo).

Estos comandos se usan como parte del argumento *índice* de `\sqrt`.

Ejemplo La instrucción `\sqrt[\leftroot{-1}\uproot{2}]{\beta}{R}` produce el siguiente resultado: $\sqrt[\beta]{R}$.

Varios radicales colocados en un mismo renglón pueden lucir “desajustados” debido a las diferentes alturas de los radicandos. Obsérvese, por ejemplo:

$$\sqrt{x} + \sqrt{y} + \sqrt{z}$$

Para corregir esta pequeña anomalía, el paquete `amsmath` posee el comando `\smash` que permite “recortar” la altura del radicando (con la opción `[t]`) o su profundidad (con la opción `[b]`), ajustando uniformemente los radicales que aparezcan en un mismo renglón.

Ejemplo Compare:

$\sqrt{x} + \sqrt{y} + \sqrt{z}$	<code>\$\sqrt{x} + \sqrt{y} + \sqrt{z}\$</code>
$\sqrt{x} + \sqrt{y} + \sqrt{z}$	<code>\$\sqrt{x} + \sqrt{\smash[b]{y}} + \sqrt{z}\$</code>

4.10. Funciones y operadores básicos

Hay funciones matemáticas básicas (como \log , \ln , \cos , \exp) y operadores (como \arg , \dim , \ker) que tradicionalmente se escriben en letra normal para distinguirlos de las variables (como x , a , y), escritas en itálicas. Los TEX-nócratas cuidadosos están acostumbrados a usar los comandos especiales que **TEX** y **LATEX** tienen para tales casos: en la Tabla 4.1 aparecen las funciones y operadores pre-definidos.

<code>\sin</code>	<code>sin</code>	<code>\sinh</code>	<code>sinh</code>	<code>\arcsin</code>	<code>arcsin</code>	<code>\exp</code>	<code>exp</code>
<code>\cos</code>	<code>cos</code>	<code>\cosh</code>	<code>cosh</code>	<code>\arccos</code>	<code>arccos</code>	<code>\dim</code>	<code>dim</code>
<code>\tan</code>	<code>tan</code>	<code>\tanh</code>	<code>tanh</code>	<code>\arctan</code>	<code>arctan</code>	<code>\arg</code>	<code>arg</code>
<code>\cot</code>	<code>cot</code>	<code>\coth</code>	<code>coth</code>	<code>\log</code>	<code>log</code>	<code>\hom</code>	<code>hom</code>
<code>\sec</code>	<code>sec</code>			<code>\ln</code>	<code>ln</code>	<code>\deg</code>	<code>deg</code>
<code>\csc</code>	<code>csc</code>			<code>\lg</code>	<code>lg</code>	<code>\ker</code>	<code>ker</code>

TABLA 4.1. Funciones y operadores básicos.

Estos comandos no sólo producen los nombres de los respectivos operadores y funciones en letra romana normal sino que dejan un espacio horizontal adecuado a izquierda y a derecha. En el siguiente ejemplo se enfatiza el uso correcto de funciones y operadores.

Ejemplo

<code>\$cos\alpha+cos\beta\$</code>	$\cos\alpha + \cos\beta$	Incorrecto
<code>\$\$\cos\alpha+\cos\beta\$\$</code>	$\cos\alpha + \cos\beta$	Correcto
<code>\$1+2log a\$</code>	$1 + 2\log a$	Incorrecto
<code>\$\$1+2\log a\$\$</code>	$1 + 2 \log a$	Correcto
<code>\$-x exp(x+y)\$</code>	$-x \exp(x + y)$	Incorrecto
<code>\$\$-x\exp(x+y)\$\$</code>	$-x \exp(x + y)$	Correcto
<code>\$m dim_F V=n\$</code>	$m \dim_F V = n$	Incorrecto
<code>\$\$m\dim_F V=n\$\$</code>	$m \dim_F V = n$	Correcto
<code>\$V^*\simeq hom(V,F)\$</code>	$V^* \simeq \hom(V, F)$	Incorrecto
<code>\$\$V^*\simeq \hom(V,F)\$\$</code>	$V^* \simeq \hom(V, F)$	Correcto

En la sección 4.18 se presentan los operadores con límites inferiores y se explica cómo el usuario puede definir funciones y operadores nuevos que preserven las normas de tipo de letra y espaciamiento.

4.11. Texto en expresiones matemáticas

Como lo señalamos en la sección 4.1, en modo matemático todos los espacios son ignorados (en el documento fuente) y los símbolos se imprimen en itálicas. Por tal razón, el paquete `amsmath` nos brinda el comando `\text{texto}` para incluir *texto* en modo matemático. Tal instrucción se utiliza principalmente en los despliegues ya que en el discurso corriente el modo matemático está delimitado por los símbolos $\$$. En los despliegues es usual utilizar los comandos `\text{a}, \quad` o `\quad\quad` para añadir espacio adicional.

Ejemplo

$$0 < a_n < \frac{1}{n} \quad \text{para todo número natural } n \geq 1.$$

```
\[
 0 < a_n < \frac{1}{n} \quad \text{para todo número natural } n \geq 1.
\]
```

Ejemplo

Sea f la función

$$f(x) = 1 + \log x + \gamma \quad (\gamma \text{ es la constante de Euler})$$

definida en el semi-eje real positivo.

Sea f la función

```
\[ f(x)=1+\log x+\gamma \quad (\gamma \text{ es la constante de Euler}) \]
definida en el semi-eje real positivo.
```

Los siguientes comandos para cambio de tipo de letra (sección 3.3),

```
\textit{...}      \textit{...}
\textbf{...}     \textsl{...}
\textsf{...}     \textsc{...}
\texttt{...}
```

también se pueden usar en modo matemático, lo que nos permite utilizar distintos tipos de letra para el texto que aparezca acompañando expresiones matemáticas. En particular, `\textit{...}` es una alternativa al comando `\text{...}` (si queremos que el texto se obtenga en letra romana normal).

4.12. Sumas

La instrucción `\sum_{límite inferior}^{límite superior}` produce símbolos sumatorios en dos tamaños diferentes: \sum (tamaño texto) y \sum (tamaño de despliegue), dependiendo del contexto. En el tamaño texto los límites de la suma aparecen situados en la parte lateral.

Ejemplo

<code>\$\sum_{n=1}^{\infty} a_n\$</code>	$\sum_{n=1}^{\infty} a_n$ (tamaño texto)
<code>\[\sum_{n=1}^{\infty} a_n \]</code>	$\sum_{n=1}^{\infty} a_n$ (tamaño de despliegue)

La colocación de los límites inferior y superior de la suma es automática pero se puede controlar con los modificadores `\nolimits` (los límites aparecen al lado derecho del símbolo \sum) y `\limits` (los límites aparecen encima y debajo del símbolo \sum).

Ejemplo

<code>\$\sum_{n=1}^{\infty} a_n\$</code>	$\sum_{n=1}^{\infty} a_n$ (texto)
<code>\$\sum\limits_{n=1}^{\infty} a_n\$</code>	$\sum_{n=1}^{\infty} a_n$ (texto)
<code>\[\sum_{n=1}^{\infty} a_n \]</code>	$\sum_{n=1}^{\infty} a_n$ (despliegue)
<code>\[\sum\nolimits_{n=1}^{\infty} a_n \]</code>	$\sum_{n=1}^{\infty} a_n$ (despliegue)

Para sumas con varias líneas en el límite inferior, el paquete `amsmath` tiene la instrucción `\substack{... \\ ...}`.

Ejemplo

<code>\[</code>	
<code>\sum_{\substack{0 < j < m \\ 1 < i \leq n}} A_{ij}</code>	$\sum_{\substack{0 < j < m \\ 1 < i \leq n}} A_{ij}$
<code>\]</code>	
<code>\[</code>	
<code>\sum_{\substack{j=k \\ i=k-1}}^{n,m} c_i \alpha_j</code>	$\sum_{\substack{j=k \\ i=k-1}}^{n,m} c_i \alpha_j$
<code>\]</code>	

Una alternativa a `\substack` es el entorno `subarray`¹, el cual permite escoger la alineación de los renglones del límite inferior de la suma. Las posibles alineaciones son: `l` (a la izquierda) y `c` (centrado).

Ejemplo Compárese con el ejemplo anterior. Aquí los renglones del límite inferior de la suma se han alineado a la izquierda.

$$\sum_{\substack{j=k \\ i=k-1}}^{n,m} c_j \alpha_i$$

```
\[
  \sum_{\begin{array}{l} j=k \\ i=k-1 \end{array}}^{n,m} c_{\{j\}} \alpha_{\{i\}}
\]
```

El comando `\sideset` del paquete `amsmath` tiene un propósito bastante peculiar: colocar símbolos en uno cualquiera de los cuatro extremos del operador \sum . Se usa en la forma:

```
\sideset{anteriores}{posteriores}
```

donde el primer argumento contiene los símbolos *anteriores* a \sum y el segundo argumento los símbolos *posteriores*. En ambos argumentos hay que usar los indicadores de posición `_` y `^`.

Ejemplo

$$\sum' a_n$$

$$\sum_*^* a_n$$

$$\sum_{n=1}^\infty a_n$$

```
\[
  \sum' a_n
  \sum_*^* a_n
  \sum_{n=1}^\infty a_n
]
```

Tanto los modificadores `\limits` y `\nolimits` como los comandos `\substack`, `\subarray` y `\sideset` se pueden usar con integrales (sección 4.13) y con operadores grandes (sección 4.14).

El paquete `amsmath` tiene la opción `nosumlimits` para forzar a L^AT_EX a escribir *todos* los límites (superiores e inferiores) de los símbolos sumatorios en la parte lateral, incluso en las expresiones desplegadas. Se invoca en la forma `\usepackage[nosumlimits]{amsmath}`.

¹El entorno `subarray` tiene una sintaxis similar a la del entorno `array` (sección 4.23) usado para matrices.

4.13. Integrales

La instrucción `\int_{...}^{...}` tiene una sintaxis similar a la de `\sum` y produce símbolos integrales en dos tamaños diferentes, dependiendo del contexto. Cuando alguno de los límites de la integral consta de un solo símbolo no son necesarias las llaves `{ }` de agrupación. Todos los comandos para integrales aparecen en la Tabla 4.2.

Instrucción	Tamaño obtenido	
	Texto	Despliegue
<code>\smallint</code>	\int	\int
<code>\int</code>	\int	\int
<code>\iint</code>	\iint	\iint
<code>\iiint</code>	\iiint	\iiint
<code>\iiiint</code>	\iiiint	\iiiint
<code>\oint</code>	\oint	\oint
<code>\idotsint</code>	$\int \cdots \int$	$\int \cdots \int$

TABLA 4.2. Comandos para integrales.

Ejemplo

<code>\$\smallint_a^b f\$</code>	$\int_a^b f$
<code>\$\int_a^b f\$</code>	$\int_a^b f$
<code>\[\int_a^b f \]</code>	$\int_a^b f$
<code>\$\int_{[a,b]} f\$</code>	$\int_{[a,b]} f$
<code>\[\int_{[a,b]} f \]</code>	$\int_{[a,b]} f$

Con cualquiera de los comandos de la Tabla 4.2 se puede usar el modificador `\limits` para forzar la colocación de los límites de la integral en la parte superior o en la parte inferior, según sea el caso, del símbolo integral.

Ejemplo Compárese la colocación de los límites de las integrales, con o sin el uso del modificador `\limits`.

<code>\int_0^1 \sqrt{t}(1+t^2)dt</code>	$\int_0^1 \sqrt{t}(1+t^2)dt$
<code>\int\limits_0^1 \sqrt{t}(1+t^2)dt</code>	$\int_0^1 \sqrt{t}(1+t^2)dt$
<code>\oint_{(0,0)}^{(1,1)} f \cdot d\alpha</code>	$\oint_{(0,0)}^{(1,1)} f \cdot d\alpha$
<code>\oint\limits_{(0,0)}^{(1,1)} f \cdot d\alpha</code>	$\oint_{(0,0)}^{(1,1)} f \cdot d\alpha$
<code>\iiint_D(R) F</code>	$\iiint_{D(R)} F$
<code>\iiint\limits_{D(R)} F</code>	$\iiint_{D(R)} F$
<code>\idotsint_S P(\beta) dS</code>	$\int \cdots \int_S P(\beta) dS$
<code>\idotsint\limits_S P(\beta) dS</code>	$\int \cdots \int_S P(\beta) dS$

Si el usuario desea que el modificador `\limits` esté siempre vigente, puede utilizar la opción `intlimits` del paquete `amsmath`. De esta manera, al invocar el paquete `amsmath` en la forma

`\usepackage[intlimits]{amsmath}`

todos los límites de las integrales aparecerán encima y/o debajo de los símbolos integrales. Por defecto, el paquete `amsmath` se carga con la opción contraria: `nointlimits`, con la cual los límites aparecen a la derecha del símbolo integral.

4.14. Operadores grandes

Los operadores \prod , \coprod , \bigcup , \bigcap , \bigoplus , \bigodot , \bigotimes , \bigvee , \bigwedge , \sqcup tienen una sintaxis similar a la de \sum .

Tamaño obtenido			Tamaño obtenido		
Instrucción	Texto	Despliegue	Instrucción	Texto	Despliegue
<code>\sum</code>	Σ	\sum	<code>\biguplus</code>	\uplus	\biguplus
<code>\prod</code>	Π	\prod	<code>\bigodot</code>	\odot	\bigodot
<code>\coprod</code>	\coprod	\coprod	<code>\bigoplus</code>	\oplus	\bigoplus
<code>\bigcup</code>	\bigcup	\bigcup	<code>\bigotimes</code>	\bigotimes	\bigotimes
<code>\bigcap</code>	\bigcap	\bigcap	<code>\bigvee</code>	\bigvee	\bigvee
<code>\bigsqcup</code>	\sqcup	\bigcup	<code>\bigwedge</code>	\bigwedge	\bigwedge

TABLA 4.3. Operadores grandes.

Ejemplo

$\$ \prod_{i \in I} X_i \$$	$\prod_{i \in I} X_i$
$\[\prod_{i \in I} X_i \]$	$\prod_{i \in I} X_i$
$\[\coprod_{i=1}^n A_i \]$	$\coprod_{i=1}^n A_i$
$\$ \bigcup_{i=1}^{\infty} A_i \$$	$\bigcup_{i=1}^{\infty} A_i$
$\[\bigcup_{i=1}^{\infty} A_i \]$	$\bigcup_{i=1}^{\infty} A_i$
$\[\bigvee_{k=1}^{n+1} P_k \]$	$\bigvee_{k=1}^{n+1} P_k$
$\$ \bigotimes_{i=0}^m V_i \$$	$\bigotimes_{i=0}^m V_i$

Los operadores grandes de la Tabla 4.3 también admiten los modificadores `\limits` y `\nolimits` para cambiar la posición de los límites. Los comandos `\substack` y `\sideset`, así como el entorno `subarray`, mencionados en la sección 4.12, también se pueden usar con estos operadores.

Ejemplo

<code>\$\bigcup_{n=1}^{\infty} a_n\$</code>	$\bigcup_{n=1}^{\infty} a_n$
<code>\$\bigcup\limits_{n=1}^{\infty} a_n\$</code>	$\bigcup_{n=1}^{\infty} a_n$
<code>\bigwedge_{\substack{j=2i \\ i>0}} X_{ij}</code>	$\bigwedge_{\substack{j=2i \\ i>0}} X_{ij}$

La opción `nosumlimits` del paquete `amsmath` —mencionada al final de la sección 4.12— para forzar a L^AT_EX a escribir *todos* los límites (superiores e inferiores) en la parte lateral, incluso en las expresiones desplegadas, afecta también a los operadores de la Tabla 4.3. Se invoca en la forma `\usepackage[nosumlimits]{amsmath}`.

4.15. Fórmulas en cajas

Una *expresión* matemática se puede encerrar en una caja por medio de `\fbox{$expresión$}` (véase la sección 3.15); es necesario que la expresión vaya entre signos \$. Con el paquete `amsmath` podemos usar, en modo matemático, la instrucción `\boxed{expresión}`, que no exige que la *expresión* se escriba entre signos \$. Por ejemplo, $\log xy = \log x + \log y$ se obtiene a partir de `$\boxed{\log xy=\log x+\log y}$` mientras que

$$\boxed{\log xy = \log x + \log y}$$

es el resultado de `\boxed{\log xy=\log x+\log y}`.

Es frecuentemente útil encerrar una expresión matemática en una caja con bordes invisibles, por medio de `\mbox{...}` (sección 3.15). Utilizamos este recurso cuando, por ejemplo, queremos aplicar a una fórmula o expresión un cierto comando que no se puede usar en modo matemático. Esto se ilustrará en la sección 4.26, en relación con las declaraciones globales `\small`, `\large`, etc para cambio de tamaño.

4.16. Acentos en modo matemático

Los comandos para acentuar símbolos, presentados en la sección 3.1, solamente se pueden usar en texto corriente y no en modo matemático. En modo matemático se deben usar los siguientes comandos de un argumento.

Comando	Acento	Ejemplo
<code>\acute{ }</code>	'	<code>\acute{a}</code> á
<code>\grave{ }</code>	`	<code>\grave{a}</code> à
<code>\hat{ }</code>	^	<code>\hat{a}</code> â
<code>\check{ }</code>	~	<code>\check{a}</code> ã
<code>\tilde{ }</code>	~	<code>\tilde{a}</code> ã
<code>\bar{ }</code>	-	<code>\bar{a}</code> ã
<code>\vec{ }</code>	-	<code>\vec{a}</code> ã
<code>\dot{ }</code>	.	<code>\dot{a}</code> ã
<code>\ddot{ }</code>	..	<code>\ddot{a}</code> ã

Estos son comandos básicos de L^AT_EX y el paquete `amsmath` tiene versiones propias de todos ellos, con los mismos nombres, excepto que la primera letra es mayúscula: `\Acute{ }`, `\Grave{ }`, `\Hat{ }`, `\Check{ }`, `\Tilde{ }`, `\Bar{ }`, `\Vec{ }`, `\Dot{ }`, `\Ddot{ }`. Fueron definidos en `amsmath` para corregir el comportamiento deficiente que tienen los comandos originales (con respecto a los acentos dobles²). Dicha anomalía se ilustra a continuación.

Ejemplo Compare:

<code>\$\hat{\hat{A}}\$</code>	Acceptable	✓
<code>\$\hat{\hat{\hat{A}}}\$</code>	Inaceptable	✗
<code>\$\tilde{\tilde{\tilde{A}}}\$</code>	Acceptable	✓
<code>\$\tilde{\tilde{\bar{A}}}\$</code>	Inaceptable	✗
<code>\$\vec{\vec{\vec{A}}}\$</code>	Acceptable	✓
<code>\$\vec{\vec{\bar{A}}}\$</code>	Inaceptable	✗
<code>\$\dot{\dot{\dot{A}}}\$</code>	Acceptable	✓
<code>\$\dot{\dot{\hat{A}}}\$</code>	Inaceptable	✗

²Para la versión 2.0 (y posteriores) de `amsmath`, los comandos con letra inicial mayúscula y letra inicial minúscula han sido unificados. Es decir, ambos se comportan satisfactoriamente con los acentos dobles.

El paquete `amsmath` tiene además los comandos `\ddot{...}` y `\dddot{...}` para colocar puntos triples y cuádruples sobre un símbolo.

Ejemplo

$$\$\\ddot{u} \$ \quad \ddot{u}$$

$$\$\\ddot{Q} \$ \quad \ddot{Q}$$

$$\$\\dddot{u} \$ \quad \dddot{u}$$

$$\$\\dddot{Q} \$ \quad \dddot{Q}$$

Para “condecoraciones” que abarquen uno o más símbolos disponemos de los comandos exhibidos en la siguiente tabla.

Comando	Ejemplo
\widehat{ }	\widehat{x}
	\widehat{xy}
	\widehat{xyz}
\widetilde{ }	\widetilde{x}
	\widetilde{xy}
	\widetilde{xyz}
\underline{ }	$\underline{x+y+z}$
\overline{ }	$\overline{x+y+z}$
\overrightarrow{ }	\overrightarrow{AB}
	$\overrightarrow{x+y+z}$
\overleftarrow{ }	$\overleftarrow{x+y+z}$
	$\overleftarrow{x+y+z}$
\underrightarrow{ }^a	$\underrightarrow{x+y+z}$
	$\underrightarrow{x+y+z}$
\underleftarrow{ }^a	$\underleftarrow{x+y+z}$
	$\underleftarrow{x+y+z}$
\overleftrightarrow{ }^a	$\overleftrightarrow{x+y+z}$
	$\overleftrightarrow{x+y+z}$
\underleftrightarrow{ }^a	$\underleftrightarrow{x+y+z}$
	$\underleftrightarrow{x+y+z}$

^aRequiere el paquete `amsmath`

El tamaño máximo que se puede obtener con `\widehat{...}` y `\widetilde{...}`, exhibido en los ejemplos de la tabla anterior. Las líneas horizontales y las flechas son, por el contrario, arbitrariamente extendibles.

4.17. Micro-espacios

Como lo hemos reiterado con anterioridad, TeX maneja con criterio propio el espaciamiento en expresiones matemáticas; el resultado obtenido es perfectamente aceptable en todos los casos. No obstante, en ocasiones el usuario puede dar retoques de finura añadiendo o eliminando pequeños espacios. El comando `\,` (mencionado en la sección 3.7) produce un micro-espacio; en modo matemático disponemos de él y de otros comandos:

- `\,` Micro-espacio cuya anchura es: `ll`.
- `\:` Equivale a dos micro-espacios; su anchura es: `lll`.
- `\;` Equivale a tres micro-espacios; su anchura es: `lll`.
- `\!` Produce un micro-espacio negativo, lo que equivale a “retroceder” un espacio de anchura igual a la producida por un micro-espacio.

Ejemplo Las diferenciales (dx , dy , etc) de las integrales lucen mejor separadas con micro-espacios. Compárese:

Entrada	Salida
<code>\[\int_a^b f(x) dx \]</code>	$\int_a^b f(x) dx$
<code>\[\int_a^b f(x), dx \]</code>	$\int_a^b f(x) dx$
<code>\$\iint f(x,y) dx dy\$</code>	$\iint f(x,y) dx dy$
<code>\$\iint f(x,y), dx, dy\$</code>	$\iint f(x,y) dx dy$

Ejemplo Para resultados óptimos, podemos añadir micro-espacios después de raíces y factoriales, si éstos van seguidos de paréntesis u otros símbolos, y al usar algunos acentos como flechas o barras (sección 4.16). Cuando el símbolo `/` se usa para fracciones, la expresión final frecuentemente luce mejor introduciendo micro-espacios negativos. Compárese:

Entrada	Salida	Entrada	Salida
<code>\sqrt{2}x</code>	$\sqrt{2}x$	<code>f(\vec{x})</code>	$f(\vec{x})$
<code>\sqrt{2}\,,x</code>	$\sqrt{2} x$	<code>f(\vec{x}\,,)</code>	$f(\vec{x})$
<code>2! 15!</code>	$2!15!$	<code>x/\log x</code>	$x/\log x$
<code>2!\,,15!</code>	$2! 15!$	<code>x/\!\log x</code>	$x/\log x$
<code>n!(n+1)!</code>	$n!(n + 1)!$	<code>\overrightarrow{AB}</code>	\overrightarrow{AB}
<code>n!\,(n+1)!</code>	$n! (n + 1)!$	<code>\overrightarrow{AB\,:}</code>	\overrightarrow{AB}

4.18. Operadores con límites inferiores

Algunos operadores admiten “límites” inferiores que, dependiendo del tamaño utilizado (texto o despliegue), aparecen a la derecha o debajo del operador. Los límites se escriben como subíndices, en la forma $_{\{\dots\}}$. Por ejemplo, $\lim_{x \rightarrow a} f(x)$ produce lo siguiente

$$\begin{array}{lll} \$\lim_{x \rightarrow a} f(x)\$ & \lim_{x \rightarrow a} f(x) & \text{(tamaño texto)} \\ \backslash [\lim_{x \rightarrow a} f(x) \] & \lim_{x \rightarrow a} f(x) & \text{(tamaño de despliegue)} \end{array}$$

En la Tabla 4.4 aparecen los operadores que se comportan como \lim . Si queremos apartarnos del resultado obtenido por defecto, utilizamos los modificadores \limits o \nolimits para forzar los límites, ya sea debajo o a la derecha del respectivo operador.

\Pr	\Pr	\sup	\sup	\projlim^a	\projlim
\gcd	\gcd	\inf	\inf	\injlim^a	\injlim
\det	\det	\lim	\lim	\varlimsup^a	\varliminf^a
\min	\min	\limsup	\limsup	\varliminf^a	\varlimsup^a
\max	\max	\liminf	\liminf	\varinjlim^a	\varprojlim^a

^a Requiere el paquete **amsmath**

TABLA 4.4. Operadores que admiten “límites”.

Ejemplo

$\$ \max_{x \in [a,b]} g(x) \$$	$\max_{x \in [a,b]} g(x) $
$\backslash [\max_{x \in [a,b]} g(x) \]$	$\max_{x \in [a,b]} g(x) $
$\$ \max \limits_{x \in [a,b]} g(x) \$$	$\max_{x \in [a,b]} g(x) $
$\$ \inf \limits_{p \notin P} L(f,p) \$$	$\inf_{p \notin P} L(f,p)$
$\$ \limsup_{n \rightarrow \infty} a_n \$$	$\limsup_{n \rightarrow \infty} a_n$
$\backslash [\limsup_{n \rightarrow \infty} a_n \]$	$\limsup_{n \rightarrow \infty} a_n$
$\$ \varlimsup_{n \rightarrow \infty} a_n \$$	$\overline{\lim}_{n \rightarrow \infty} a_n$
$\backslash [\varliminf_{n \rightarrow \infty} a_n \]$	$\underline{\lim}_{n \rightarrow \infty} a_n$

Con las versiones del paquete `babel`, opción `spanish`, posteriores al año 2000, se dispone de los siguientes comandos para funciones:

<code>\sen</code>	<code>sen</code>	<code>\arcsen</code>	<code>arc sen</code>
<code>\tg</code>	<code>tg</code>	<code>\arctg</code>	<code>arc tg</code>
<code>\cotg</code>	<code>cotg</code>	<code>\senh</code>	<code>senh</code>
<code>\cosec</code>	<code>cosec</code>	<code>\tgh</code>	<code>tgh</code>

Además, algunos de los operadores de la Tabla 4.4 aparecen acentuados:

<code>\max</code>	<code>máx</code>	<code>\lim</code>	<code>lím</code>	<code>\liminf</code>	<code>lím inf</code>
<code>\min</code>	<code>mín</code>			<code>\limsup</code>	<code>lím sup</code>

Pero los operadores que requieren del paquete `amsmath` no se obtendrán con tildes. No obstante, el paquete `amsmath` permite definir nuevas funciones y operadores con la declaración `\DeclareMathOperator`, la cual permite al usuario incluir, si lo desea, tildes y otros acentos en los nombres de los operadores. Con el comando `\unaccentedoperators` se eliminan por completo las tildes predefinidas por la opción `spanish` de `babel`.

Para definir nuevas funciones u operadores, el paquete `amsmath` dispone del comando:

`\DeclareMathOperator{\nombre}{definición}`

El nombre de la función u operador así definido aparecerá escrito en letra normal (y no en itálicas) en cualquier expresión matemática, y tendrá el espaciamiento adecuado. Si queremos, además, que el operador definido admita “límites” (como `\sup`, `\lim`, `\max`, etc) debemos usar la versión estrella:

`\DeclareMathOperator*{\nombre}{definición}`

Los operadores así definidos también admiten los modificadores `\limits` y `\nolimits` para cambiar la posición del límite inferior.

Las declaraciones `\DeclareMathOperator` y `\DeclareMathOperator*` pueden aparecer únicamente en el preámbulo del documento (después de cargar el paquete `amsmath`!); en caso contrario, se recibirá un mensaje de error.

Ejemplo Si no estamos usando la opción `spanish` del paquete `babel`, podemos “españolizar” la función seno definiendo `\sen` con la declaración:

`\DeclareMathOperator{\sen}{sen}`

Esta definición nos permite escribir expresiones como

$$\$ \sen^2(x) + \cos^2(x) = 1 \$ \quad \text{sen}^2(x) + \cos^2(x) = 1$$

$$\$ \lim_{x \rightarrow 0} \frac{\sen x}{x} = 1 \$ \quad \lim_{x \rightarrow 0} \frac{\sen x}{x} = 1$$

Ejemplo Vamos a definir el operador \Max, de tal manera que produzca la expresión ‘Máx’ y admita “límites” (como lo hace el comando estándar \max). Puesto que ni ‘a’ ni á se pueden usar en modo matemático, debemos utilizar el comando \acute{a} (sección 4.16) para obtener la a con tilde. La definición requerida es:

```
\DeclareMathOperator*{\Max}{M\acute{a}x}
```

con la cual podemos escribir expresiones como:

$$\$ \Max_{x \in [a,b]} |g(x)| \$ \quad \text{Máx}_{x \in [a,b]} |g(x)|$$

$$\backslash [\ Max_{x \in [a,b]} |g(x)| \] \quad \text{Máx}_{x \in [a,b]} |g(x)|$$

4.19. Relaciones de congruencia

LATEX tiene dos comandos para relaciones de congruencia con módulo: \bmod y \pmod. El paquete amsmath añade las versiones \mod y \pmod. Se diferencian entre sí por la manera como la expresión ‘mod’ es presentada³.

Ejemplo

\$a \equiv b \bmod{n}\$	$a \equiv b \pmod{n}$
\$a \equiv b \mod{n}\$	$a \equiv b \pmod{n}$
\$a \equiv b \pmod{n}\$	$a \equiv b \pmod{n}$
\$a \equiv b \pmod{n}\$	$a \equiv b \pmod{n}$
\$n \equiv m+1 \bmod{p^2-1}\$	$n \equiv m+1 \pmod{p^2-1}$
\$n \equiv m+1 \mod{p^2-1}\$	$n \equiv m+1 \pmod{p^2-1}$
\$n \equiv m+1 \pmod{p^2-1}\$	$n \equiv m+1 \pmod{p^2-1}$

$$a \equiv_\ell b \pmod{H} \quad \text{si y sólo si} \quad ab^{-1} \in H$$

```
\[
  a \equiv_\ell b \pmod{H} \quad \text{si y sólo si} \\
  ab^{-1} \in H
]
```

³Si se usa el paquete babel, opción spanish (versión posterior al año 2000), se obtendrá mód con los comandos \bmod y \pmod de LATEX. El acento se puede eliminar declarando con anterioridad \unaccentedoperators.

4.20. Paréntesis y símbolos de agrupación

Los paréntesis () y [] se obtienen con las respectivas teclas; su tamaño se puede ampliar (en modo matemático), como lo explicaremos en esta sección. L^AT_EX tiene además otros símbolos de agrupación (en inglés, *delimiters*) para expresiones matemáticas, a saber:

{ \{	} \}
< \langle	> \rangle
ó \vert	\ ó \Vert
[\lfloor] \rfloor
[\lceil] \rceil

Para agrandar el tamaño de los símbolos de agrupación, de tal manera que abarquen la expresión que encierran, se usan los modificadores `\left` y `\right`. Éstos se deben usar en pares; usar uno y no el otro conduce a un mensaje de error. De ser necesario, se usa `\left.` o `\right.` para completar el par. En el siguiente ejemplo se ilustra tal situación.

Ejemplo

<code>\left(\frac{a}{b}, \frac{c}{d}\right)</code>	$\left(\frac{a}{b}, \frac{c}{d}\right)$
<code>\left\{ \frac{1+a}{1+a} : a \in A \right\}</code>	$\left\{ \frac{1}{1+a} : a \in A \right\}$
<code>\left(\frac{1-n}{1+\frac{1}{n}} \right)^n</code>	$\left(\frac{1-n}{1+\frac{1}{n}} \right)^n$
<code>\left. \frac{dy}{dx} \right _{x=b} = b+1</code>	$\left. \frac{dy}{dx} \right _{x=b} = b+1$

Desafortunadamente, `\left` y `\right` no producen siempre resultados satisfactorios porque los símbolos de agrupación obtenidos pueden ser excesivamente grandes o demasiado pequeños (véanse los ejemplos que aparecen después de la Tabla 4.5). Para remediar esta deficiencia, L^AT_EX tiene los modificadores de tamaño `\big`, `\Big`, `\bigg` y `\Bigg` con los cuales el usuario puede producir símbolos de agrupación en el tamaño que estime más conveniente. El paquete `amsmath` tiene una versión mejorada de estos modificadores, en pares izquierda-derecha:

`\bigl \bigr \Bigl \Bigr \biggl \biggr \Biggl \Biggr`

En la Tabla 4.5 se pueden observar los tamaños concretos obtenidos (con la opción [11pt] de `\documentclass`). A diferencia de sus homólogos de L^AT_EX, éstos funcionan bien con fuentes de todos los tamaños y también se pueden usar con las flechas ↑, ↓, ↕, ↑↑, ↓↓, ↕↗ (sección 4.3), así como con los símbolos / y \.

Tamaño normal	<code>\bigl</code> <code>\bigr</code>	<code>\Bigl</code> <code>\Bigr</code>	<code>\biggl</code> <code>\biggr</code>	<code>\Biggl</code> <code>\Biggr</code>
()	()	()	()	()
[]	[]	[]	[]	[]
{ }	{ }	{ }	{ }	{ }
$\langle \rangle$	$\langle \rangle$	$\langle \rangle$	$\langle \rangle$	$\langle \rangle$
[]	[]	[]	[]	[]
[]	[]	[]	[]	[]

TABLA 4.5. Los modificadores de tamaño para símbolos de agrupación.

Ejemplo Compare:

$$|x - y| + |y - z| \quad \$\left| |x-y|+|y-z| \right| $$$

$$\bigl|x - y\bigr| + \bigl|y - z\bigr| \quad \$\bigl| |x-y|+|y-z| \bigr| $$$

Ejemplo Compare:

$$\left(\frac{1-n}{1+\frac{1}{n}} \right)^n \quad \left[\left(\frac{1-n}{1+\frac{1}{n}} \right)^n \right]$$

Ejemplo Compare:

$$\left| \left\langle \sum_i x_i \alpha_i, \beta \right\rangle^{1/2} \right| \quad \left| \left[\left(\sum_i x_i \alpha_i, \beta \right)^{1/2} \right] \right|$$

Ejemplo El uso de `\left` y `\right` también da lugar a resultados insatisfactorios al escribir en tamaño texto una expresión como $\left| \frac{a'}{b'} \right|$ (obtenida a partir de `$\left| \frac{a'}{b'} \right|`). Las barras verticales son excesivamente largas y alteran inadecuadamente el espaciamiento normal entre renglones; en su lugar debemos usar el modificador `\big`, en la forma `$\big| \frac{a'}{b'} \big|`, para obtener $\left| \frac{a'}{b'} \right|$.

El comando `\overbrace{expresión}` coloca una llave o corchete horizontal encima de una *expresión*; sobre la llave misma se puede colocar otra expresión, con la sintaxis de superíndice: `\overbrace{...}^{\dots}`. Similarmente, `\underbrace` coloca una llave debajo de una expresión; bajo tal llave se puede colocar alguna expresión adicional, con la sintaxis de subíndice: `\underbrace{...}_{\dots}`.

Ejemplo

`$\overbrace{x+y+z}$`

$$\overbrace{x+y+z}$$

`$\underbrace{A \times \cdots \times A}_{\{n\} \text{ factores}} \rightarrow B$`

$$\underbrace{A \times \cdots \times A}_{n \text{ factores}} \rightarrow B$$

$$\underbrace{a + \cdots + a}_{n \text{ veces}} + 1 + \cdots + 1 + \underbrace{b + \cdots + b}_{m \text{ veces}}$$

$k+n+m$ sumandos

```
\left[ \underbrace{\overbrace{a+\dots+a}^{\text{n veces}}}_{k+n+m \text{ sumandos}} + 1 + \overbrace{b+\dots+b}^{\text{m veces}} \right]_{\{k+n+m\}}
```

4.21. Casos

Las construcciones con “casos” son muy corrientes en matemáticas; el paquete `amsmath` posee el entorno `cases` para presentar tales situaciones. Se usa en la forma:

```
\begin{cases}
  ... & ...
  ....
  ... & ...
\end{cases}
```

En la última fila no es necesario usar `\\"\\` aunque hacerlo no conduce a ningún mensaje de error. El entorno `cases` produce un corchete exterior `{` de tamaño adecuado.

Ejemplo

$$f_n(x) = \begin{cases} -x^2 + n, & \text{si } x < 0 \text{ y } n \text{ es par,} \\ \alpha + x, & \text{si } x > 0, \\ x^2, & \text{en otros casos.} \end{cases}$$

```
\[ f_n(x)=
\begin{cases}
-x^2+n, & \text{\text{si } $x<0$ y $n$ es par}, \\
\alpha+x, & \text{\text{si } $x>0$,} \\
x^2, & \text{\text{en otros casos.}}
\end{cases} \]
```

Ejemplo

$$\begin{cases} x \cdot \infty = \infty \cdot x = \infty, & 0 < x \leq \infty \\ x \cdot \infty = \infty \cdot x = -\infty, & -\infty \leq x < 0 \\ x \cdot (-\infty) = (-\infty) \cdot x = -\infty, & 0 < x \leq \infty \\ x \cdot (-\infty) = (-\infty) \cdot x = \infty, & -\infty \leq x < 0 \end{cases}$$

```
\[
\begin{cases}
x\cdot\infty=\infty\cdot x=\infty, & 0 < x \leq \infty \\
x\cdot\infty=\infty\cdot x=-\infty, & -\infty \leq x < 0 \\
x\cdot(-\infty)=(-\infty)\cdot x=-\infty, & 0 < x \leq \infty \\
x\cdot(-\infty)=(-\infty)\cdot x=\infty, & -\infty \leq x < 0
\end{cases}
\]
```

4.22. Matrices

El paquete `amsmath` posee seis entornos para matrices, con una sintaxis muy natural: las diferentes componentes de la matriz se separan con `&` y las filas con `\backslash`. El número de símbolos `&` debe ser el mismo en cada fila y, por defecto, las componentes de la matriz aparecen centradas en sus respectivas columnas. En la última fila de la matriz no es necesario usar `\backslash` y los símbolos de agrupación exteriores adquieren el tamaño adecuado. Los seis entornos son:

<code>\begin{matrix} ... \end{matrix}</code>	para matrices sin paréntesis,
<code>\begin{pmatrix} ... \end{pmatrix}</code>	para matrices entre (),
<code>\begin{bmatrix} ... \end{bmatrix}</code>	para matrices entre [],
<code>\begin{vmatrix} ... \end{vmatrix}</code>	para matrices entre ,
<code>\begin{Bmatrix} ... \end{Bmatrix}</code>	para matrices entre { },
<code>\begin{Vmatrix} ... \end{Vmatrix}</code>	para matrices entre .

Ejemplos

```
\begin{pmatrix}
1 & 2 \\
2 & -3
\end{pmatrix}
```

$$\begin{pmatrix} 1 & 2 \\ 2 & -3 \end{pmatrix}$$

```
\begin{pmatrix}
\lambda - 1 & 2 & -1 \\
2 & \lambda - 3 & 4 \\
1 & 0 & \lambda + 1
\end{pmatrix}
```

$$\begin{pmatrix} \lambda - 1 & 2 & -1 \\ 2 & \lambda - 3 & 4 \\ 1 & 0 & \lambda + 1 \end{pmatrix}$$

```
\begin{bmatrix}
0 & & & \\
1 & 0 & & \\
1 & -1 & 0 & \\
1 & 3 & -2 & 4
\end{bmatrix}
```

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & -1 & 0 & \\ 1 & 3 & -2 & 4 \end{bmatrix}$$

```
\begin{bmatrix}
0 & i & -i & 1 \\
1 & 0 & i & -1 \\
i & -1 & 0 & -i \\
x_1 & x_2 & x_3 & x_4
\end{bmatrix}
```

$$\begin{bmatrix} 0 & i & -i & 1 \\ 1 & 0 & i & -1 \\ i & -1 & 0 & -i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Máximo número de columnas. Los citados entornos para matrices tienen la siguiente limitación: admiten un máximo de 10 columnas por matriz (aunque no hay limitación sobre el número de filas). El valor máximo para el número de columnas está controlado por el contador `MaxMatrixCols` y el usuario puede cambiarlo usando `\setcounter` (sección 3.25.1). Por ejemplo, si el usuario necesita matrices con 15 columnas puede escribir:

```
\setcounter{MaxMatrixCols}{15}
```

antes de la aparición de la primera matriz grande o, preferiblemente, en el preámbulo del documento.

Filas de puntos. El comando `\hdotsfor{n}` del paquete `amsmath` produce una fila de puntos que se extiende por n columnas.

Ejemplo

```
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\hphantom{a_{11}} \cdots & \cdots & \cdots & \cdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix}
```

`\begin{pmatrix}`

`a_{11} & a_{12} & \cdots & a_{1n} \\`

`a_{21} & a_{22} & \cdots & a_{2n} \\`

`\hphantom{a_{11}} \cdots & \cdots & \cdots & \cdots \\`

`a_{m1} & a_{m2} & \cdots & a_{mn}`

`\end{pmatrix}`

El espaciamiento entre los puntos se puede modificar con un argumento opcional: `\hdotsfor[separación]{n}`. Esto se ilustra a continuación.

Ejemplo Aquí se usa la opción [2.5] en la fila de puntos para lograr una separación 2.5 veces mayor que la normal.

```
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\hphantom{a_{11}} \cdots & \cdots & \cdots & \cdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix}
```

`\begin{pmatrix}`

`a_{11} & a_{12} & \cdots & a_{1n} \\`

`a_{21} & a_{22} & \cdots & a_{2n} \\`

`\hphantom{a_{11}} \cdots & \cdots & \cdots & \cdots \\`

`a_{m1} & a_{m2} & \cdots & a_{mn}`

`\end{pmatrix}`

Matrices pequeñas. El paquete `amsmath` tiene un entorno especial para las matrices pequeñas, tal como $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, que aparecen en un párrafo normal. Tal entorno es `smallmatrix`. Los paréntesis exteriores se deben indicar explícitamente ya que no existen las versiones `p`, `b`, `v`, `B`, `V` para `smallmatrix`. La matriz que aparece en este párrafo fue escrita a partir de:

```
$\bigl(\begin{smallmatrix} a&b \\ c&d \end{smallmatrix}\bigr)$.
```

4.23. Matrices con el entorno array de LATEX

Como se señaló en la sección anterior, las matrices creadas con los entornos `matrix`, `pmatrix`, `bmatrix`, `vmatrix`, `Bmatrix` y `Vmatrix` tienen sus componentes centradas en sus respectivas columnas. No existe manera, usando tales entornos, de cambiar la justificación. Si queremos que las entradas de una determinada columna aparezcan cargadas a la izquierda o a la derecha debemos construir la matriz usando el entorno `array`. Dicho entorno pertenece a la colección básica de macros LATEX, no al paquete `amsmath`, y sólo se puede usar en modo matemático.

La sintaxis de `array` se asemeja a la de `matrix`, con la diferencia de que la matriz debe tener un *formato* pre-definido:

```
\begin{array}{formato}
... & ... & ...
.....
... & ... & ...
\end{array}
```

El *formato* contiene información sobre el número de columnas, con su respectiva justificación: `l` (a la izquierda), `c` (centrada) y `r` (a la derecha). Los paréntesis exteriores de la matriz se deben colocar explícitamente usando los comandos `\left` y `\right` (sección 4.20).

Ejemplo La siguiente matriz, construida con `array`, tiene formato `{rcl}`, según el cual la primera columna está justificada a la derecha, la tercera a la izquierda y la segunda está centrada.

$$\left(\begin{array}{rcl} 0 & 0 & 0 \\ -0.1 & a & 0.1 \\ -0.01 & 1+a & 0.01 \\ -0.001 & 1+a+a^2 & 0.001 \\ -0.0001 & 1+a+a^2+a^3 & 0.0001 \end{array} \right)$$

```
\[
\left(
\begin{array}{rcl}
0 & 0 & 0 \\
-0.1 & a & 0.1 \\
-0.01 & 1+a & 0.01 \\
-0.001 & 1+a+a^2 & 0.001 \\
-0.0001 & 1+a+a^2+a^3 & 0.0001
\end{array}
\right)
\]
```

4.24. Tipos de letra en modo matemático

\LaTeX utiliza letra itálica en expresiones matemáticas, excepto para los nombres de funciones y operadores básicos (secciones 4.10 y 4.18). Debemos evitar usar $\$...$$ como sustituto de $\text{\textit}{...}$ o $\{\text{\it} ... \}$ ya que el espaciamiento de los símbolos en modo matemático es diferente del que se emplea para el texto corriente.

Ejemplo Compare:

$\$$ Diferenciabilidad $\$$	<i>Diferenciabilidad</i>
$\text{\textit}{Diferenciabilidad}$	<i>Diferenciabilidad</i>

Podemos usar varios tipos de letra en modo matemático con los comandos mostrados en la Tabla 4.6.

Instrucción	Tipo de letra obtenido
$\text{\mathrm}{abc...}$	abcdefghijklmnoprstuvwxyz
$\text{\mathrm}{ABC...}$	ABCDEFGHIJKLMNPQRSTUVWXYZ
$\text{\mathit}{abc...}$	<i>abcdefghijklmnoprstuvwxyz</i>
$\text{\mathit}{ABC...}$	<i>ABCDEFGHIJKLMNPQRSTUVWXYZ</i>
$\text{\mathsf}{abc...}$	abcdefghijklmnoprstuvwxyz
$\text{\mathsf}{ABC...}$	ABCDEFGHIJKLMNPQRSTUVWXYZ
$\text{\mathtt}{abc...}$	abcdefghijklmnoprstuvwxyz
$\text{\mathtt}{ABC...}$	ABCDEFGHIJKLMNPQRSTUVWXYZ
$\text{\mathbf}{abc...}$	abcdefghijklmnoprstuvwxyz
$\text{\mathbf}{ABC...}$	ABCDEFGHIJKLMNPQRSTUVWXYZ
$\text{\mathcal}{ABC...}$	<i>ABCDEFGHIJKLMNPQRSTUVWXYZ</i>
$\text{\mathfrak}{abc...}^a$	<i>abcdefghijklmnoprstuvwxyz</i>
$\text{\mathfrak}{ABC...}^a$	<i>ABCDEFGHIJKLMNPQRSTUVWXYZ</i>
$\text{\mathbb}{ABC...}^a$	ABCDEFGHIJKLMNPQRSTUVWXYZ
$\text{\mathscr}{ABC...}^b$	<i>ABCDEFGHIJKLMNPQRSTUVWXYZ</i>

^aRequiere el paquete `amssymb`

^bRequiere el paquete `euscript` con la opción `mathscr`

TABLA 4.6. Comandos para tipos de letra en modo matemático.

Hay que tener presente que `\mathcal` y `\mathbb` requieren del paquete `amssymb` y solamente admiten letras mayúsculas como argumentos. Para usar el comando `\mathscr` hay que cargar el paquete `euscript` con la opción `mathscr`; es decir, hay que escribir en el preámbulo del documento la instrucción `\usepackage[mathscr]{euscript}`. Dicho paquete hace parte del software distribuido conjuntamente con `amsmath` y su único propósito es permitir el acceso a los símbolos de esta fuente, denominada ‘Euler Script’ o ‘Euler Caligráfica’, disponible solamente en letras mayúsculas⁴.

Ejemplo

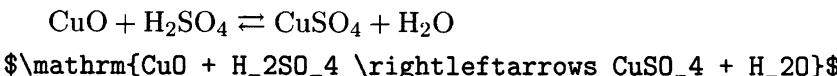
1. $\mathbf{F}(x, y, z) = yz\mathbf{i} + xz\mathbf{j} + xy\mathbf{k}$.
2. $\mathfrak{A} \models \varphi[x], \quad x \in A, \varphi \in \mathcal{K}$.
3. $(\mathfrak{a}_n, \mathfrak{b}_n) \in X \times Y$ para $\mathfrak{a}, \mathfrak{b} \in \mathcal{A}$ y para todo $n \in \mathbb{N}$.
1. $\$\\mathbf{F}(x,y,z)=yz\\mathbf{i}+xz\\mathbf{j}+xy\\mathbf{k}\$.$
2. $\$\\mathfrak{A}\\models\\varphi[x], \\quad x\\in\\mathsf{A}, \\ \\varphi\\in\\mathscr{K}\$.$
3. $\$(\\mathfrak{a}_n, \\mathfrak{b}_n)\\in\\mathsf{X}\\times\\mathsf{Y}\\$ para
$\\mathfrak{a}, \\mathfrak{b}\\in\\mathcal{A}\\$ y para todo $n\\in\\mathbb{N}\$.$

Con los comandos de la Tabla 4.6, los únicos símbolos afectados por el cambio de tipo de letra son los números y las letras del alfabeto ordinario; todo lo demás (paréntesis, símbolos de operaciones, funciones básicas, letras griegas, etc) permanece inalterado. Esto se ilustra en el siguiente ejemplo.

Ejemplo

1. $F[x \cdot 2^y] = \alpha(e^x + y \log 2)$
2. $\mathsf{F}[x \cdot 2^y] = \alpha(\mathsf{e}^x + y \log 2)$
3. $\mathit{F}[x \cdot 2^y] = \alpha(\mathit{e}^x + y \log 2)$
1. $\$F[x\\cdot 2^y]=\\alpha(e^x+y\\log 2)\$$
2. $\$\\mathsf{F}[x\\cdot 2^y]=\\alpha(\\mathsf{e}^x+y\\log 2)\$$
3. $\$\\mathit{F}[x\\cdot 2^y]=\\alpha(\\mathit{e}^x+y\\log 2)\$$

Ejemplo Los elementos y compuestos químicos se escriben en letra romana (vertical); al escribir reacciones químicas en modo matemático es entonces útil recurrir a `\mathrm`.



⁴El paquete `euscript` también ha sido distribuido como `eucal` y se puede acceder a él escribiendo `\usepackage[mathscr]{eucal}`.

4.25. Símbolos en negrilla

El comando `\mathbf{...}`, mencionado en la sección anterior, afecta solamente a las letras del alfabeto ordinario, a los números y a las letras griegas mayúsculas no inclinadas. Además, las negrillas obtenidas con `\mathbf{...}` son negrillas no-italicas. Para suplir estas limitaciones, el paquete `amsmath` tiene la instrucción `\boldsymbol{...}` con la que se obtienen los símbolos en negrilla, preservando el tipo de letra.

Ejemplo Comparación entre los comandos `\mathbf{...}` y `\boldsymbol{...}`.

1. $\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v} \cdot \mathbf{w}$
2. $\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v} \cdot \mathbf{w}$
3. $\varphi'(x) = \mathbf{F}(\varphi(x))$
4. $\varphi'(x) = \mathbf{F}(\varphi(x))$
5. $\mathbf{F}[x \cdot 2^y] = \alpha(e^x + y \log 2)$
6. $\mathbf{F}[x \cdot 2^y] = \alpha(e^x + y \log 2)$
7. $(\Psi \wedge \Phi) \rightarrow (\Psi \vee \Phi)$
8. $(\Psi \wedge \Phi) \rightarrow (\Psi \vee \Phi)$
1. $\$ \langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v} \cdot \mathbf{w} \$$
2. $\$ \langle \boldsymbol{v}, \boldsymbol{w} \rangle = \boldsymbol{v} \cdot \boldsymbol{w} \$$
3. $\$ \mathbf{\varphi}'(x) = \mathbf{F}(\varphi(x)) \$$
4. $\$ \boldsymbol{\varphi}'(x) = \boldsymbol{F}(\varphi(x)) \$$
5. $\$ \mathbf{F}[x \cdot 2^y] = \alpha(e^x + y \log 2) \$$
6. $\$ \boldsymbol{F}[x \cdot 2^y] = \alpha(e^x + y \log 2) \$$
7. $\$ \mathbf{\Psi \wedge \Phi} \rightarrow (\Psi \vee \Phi) \$$
8. $\$ \boldsymbol{\Psi \wedge \Phi} \rightarrow (\Psi \vee \Phi) \$$

Es necesario señalar que cuando las fuentes locales no incluyen los correspondientes símbolos en negrilla, el comando `\boldsymbol{...}` no tiene efecto alguno. Teniendo a nuestra disposición las fuentes estándares de `TeX`⁵ y los paquetes `amsmath`, `amsymb` y `euscript`, disponemos de la versión en negrilla de casi todos los símbolos. La siguiente tabla muestra el efecto de `\boldsymbol{...}` sobre las letras mayúsculas, en los tipos de letra romana normal, `\mathsf{...}`, `\mathcal{...}`, `\mathscr{...}` y `\mathfrak{...}`.

⁵Las fuentes estándares de `TeX` son conocidas como fuentes CM; véase al respecto el Capítulo 9.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$\boldsymbol{\$ \backslash boldsymbol\{ABCDEFHIJKLMNOPQRSTUVWXYZ\} \$}$
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$\boldsymbol{\$ \backslash boldsymbol\{\mathsf{ABCDEFGHIJKLMNPQRSTUVWXYZ}\} \$}$
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$\boldsymbol{\$ \backslash boldsymbol\{\mathcal{ABCDEFGHIJKLMNPQRSTUVWXYZ}\} \$}$
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$\boldsymbol{\$ \backslash boldsymbol\{\mathscr{ABCDEFGHIJKLMNPQRSTUVWXYZ}\} \$}$
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
$\boldsymbol{\$ \backslash boldsymbol\{\mathfrak{ABCDEFGHIJKLMNPQRSTUVWXYZ}\} \$}$

Entre los símbolos que *no* se ven afectados por `\boldsymbol` se encuentran los obtenidos con `\mathbb{...}`, como \mathbb{A} , \mathbb{B} , \mathbb{C} , ..., y los operadores grandes \sum , \prod , \bigcup , ... (sección 4.14). Cuando necesitamos imperiosamente un símbolo en negrilla que no podamos obtener con `\boldsymbol{...}` podemos recurrir a la llamada “negrilla de los pobres” `\pmb{...}` (del inglés, *poor man's bold*). El efecto de este comando es imprimir múltiples copias del mismo símbolo a distancias muy pequeñas, por lo que el resultado final no es de calidad óptima, pero es aceptable.

Djemento Con las fuentes estándares de L^AT_EX y los paquete `amsmath` y `amssymb`, el comando `\boldsymbol{...}` no produce símbolos en negrilla para los siguientes caracteres. En su lugar, podemos usar `\pmb{...}`.

$\$ \mathbb{N} \$$	\mathbb{N}	$\$ \pmb{\mathbb{N}} \$$	\mathbb{N}
$\$ \mathbb{R} \$$	\mathbb{R}	$\$ \pmb{\mathbb{R}} \$$	\mathbb{R}
$\$ \sum \$$	\sum	$\$ \pmb{\sum} \$$	\sum
$\$ \prod \$$	\prod	$\$ \pmb{\prod} \$$	\prod
$\$ \bigcup \$$	\bigcup	$\$ \pmb{\bigcup} \$$	\bigcup

L^AT_EX tiene la instrucción `\boldmath{...}`, similar a `\boldsymbol{...}`, pero (aunque parezca extraño) *no se puede usar en modo matemático*; es decir, `\boldmath{...}` no puede estar bajo el alcance de `$. . . $` ni de `\[. . . \]` ni del entorno `equation`. Si queremos usar tal instrucción en modo matemático debemos encerrarla dentro de `\boxed{...}`. Debido a esta limitación, recomendamos al lector utilizar el paquete `amsmath` y su comando `\boldsymbol{...}`.

4.26. Tamaño de los símbolos

LA**T**E**X** tiene sus propios criterios para escoger el tamaño de los símbolos en expresiones matemáticas; por ejemplo, los sub-índices y los super-índices son de menor tamaño que las bases, y las fracciones, sumas e integrales en párrafos son de menor tamaño que en despliegues. Para cambiar el tamaño de los símbolos en expresiones matemáticas debemos utilizar una de las siguientes declaraciones globales:

Declaración global	Efecto obtenido
<code>{\displaystyle ...}</code>	Tamaño en expresiones desplegadas.
<code>{\textstyle ...}</code>	Tamaño en texto normal (párrafos).
<code>{\scriptstyle ...}</code>	Tamaño de los exponentes y subíndices de primer nivel.
<code>{\scriptscriptstyle ...}</code>	Tamaño de los exponentes y subíndices de segundo nivel.

Recomendamos al lector tener precaución al usar estos comandos. **L**A**T**E**X** no sabe nada de matemáticas pero sabe más que cualquiera sobre tipografía matemática; su escogencia de los tamaños es siempre más acertada que la de los usuarios poco experimentados.

Los comandos `\scriptstyle` y `\scriptscriptstyle` son especialmente útiles cuando se necesita letra pequeña en gráficas (véanse los capítulos 11 y 12).

Ejemplo

a^{n^2}	a^{n^2}
$a^{\displaystyle n^2}$	a^{n^2}
$2^{2^{2^{2^{n}}}}$	$2^{2^{2^n}}$
$2^{\displaystyle 2^{\displaystyle 2^{\displaystyle 2^n}}}$	$2^{2^{2^n}}$

Las declaraciones globales (`\tiny`, `\small`, `\large`, etc) para el cambio del tamaño de letra (sección 3.4) también sirven para cambiar el tamaño de expresiones matemáticas. Se debe tener presente que estos comandos *no se pueden usar en modo matemático*; es decir, no pueden estar bajo el alcance de `...$`, ni de `\[... \]`, ni de entornos como `equation`. Si queremos cambiar el tamaño de un símbolo particular, usando estas declaraciones, debemos encerrarlo primero dentro de `\mbox{...}`; esto se ilustra en el segundo y tercer ejemplos de la página siguiente.

Ejemplo

<code>\tiny \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\scriptsize \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\footnotesize \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\small \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\normalsize \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\large \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\Large \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\LARGE \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\huge \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$
<code>\Huge \$f_x:A\rightarrow \mathbb{R}^2\$</code>	$f_x : A \rightarrow \mathbb{R}^2$

Ejemplo En (2) se ha aumentado el tamaño de la base e y en (4) el de A , pero no se han alterado los exponentes ni los sub-índices.

(1) <code>\$e^{\frac{2k\pi i}{n}}\$</code>	$e^{\frac{2k\pi i}{n}}$
(2) <code>\$.mbox{\Large \$e^{\frac{2k\pi i}{n}}\$}\$</code>	$e^{\frac{2k\pi i}{n}}$
(3) <code>\$A_{x_i^n}^{y_j^m}\$</code>	$A_{x_i^n}^{y_j^m}$
(4) <code>\$.mbox{\Large \$A_{x_i^n}^{y_j^m}\$}\$</code>	$A_{x_i^n}^{y_j^m}$

Ejemplo En (2), (3), (5) y (6) se ha reducido el tamaño de los subíndices —con respecto a (1) y (4)— pero las letras F y σ no se han alterado. Lo hemos hecho usando primero `\scriptscriptstyle` y luego `\tiny`; el resultado obtenido con ambos procedimientos es exactamente el mismo.

(1) <code>\$F_j\$</code>	F_j
(2) <code>\$F_{\scriptscriptstyle j}\$</code>	F_j
(3) <code>\$F_{\tiny j}\$</code>	F_j
(4) <code>\$.sigma_{ij}\$</code>	σ_{ij}
(5) <code>\$.sigma_{\scriptscriptstyle ij}\$</code>	σ_{ij}
(6) <code>\$.sigma_{\tiny ij}\$</code>	σ_{ij}

4.27. Símbolos sobre símbolos

Para colocar algo encima de algo (en modo matemático), L^AT_EX tiene el comando `\stackrel` que se usa en la forma:

```
\stackrel{algo encima de}{algo}
```

El paquete `amsmath` tiene dos macros similares: `\overset` (que se comporta exactamente como `\stackrel`) y `\underset`. Además, para colocar algo sobre o debajo de flechas, `amsmath` nos brinda los macros `\xrightarrow` y `\xleftarrow`; las flechas se extienden el espacio necesario. La sintaxis precisa de estos comandos es:

```
\overset{algo encima de}{algo}
\underset{algo debajo de}{algo}
\xrightarrow[algo debajo]{algo encima}
\xleftarrow[algo debajo]{algo encima}
```

Obsérvese que la parte `[algo debajo]` en `\xleftarrow` y `\xrightarrow` es opcional. Podemos hacer que las flechas adquieran una mayor longitud añadiendo `\lvert` o `\quad` en cualquiera de los argumentos, como se ilustra en los numerales (5), (6) y (7) del ejemplo que sigue.

Ejemplo

- (1) $\overset{a}{W}$
- (2) $\overset{a}{\underset{b}{W}}$
- (3) $\vec{x} \stackrel{\text{def}}{=} (x_1, \dots, x_n)$
- (4) $0 \rightarrow E' \xrightarrow{f} E \xrightarrow{g} E'' \rightarrow 0$
- (5) $0 \rightarrow E' \xrightarrow{f} E \xrightarrow{g} E'' \rightarrow 0$
- (6) $\cdots \rightarrow V_{i-1} \xrightarrow{d_{i-1}} V_i \xrightarrow{d_i} V_{i+1} \rightarrow \cdots$
- (7) $\mathcal{A}/R \xrightarrow[\simeq]{f \circ g \circ h} X$

- (1) `\stackrel{a}{W}` (equivalentemente, `\overset{a}{W}`)
- (2) `\overset{a}{\underset{b}{W}}`
- (3) `\vec{x} \overset{\text{def}}{=} (x_1, \dots, x_n)`
- (4) `\xrightarrow[0]{f} E' \xrightarrow[g]{g} E'' \rightarrow 0`
- (5) `\xrightarrow[0]{f} E' \xrightarrow[f]{g} E'' \rightarrow 0`
- (6) `\cdots \rightarrow V_{i-1} \xrightarrow{d_{i-1}} V_i \xrightarrow{d_i} V_{i+1} \rightarrow \cdots`
- (7) `\mathcal{A}/R \xrightarrow[\simeq]{f \circ g \circ h} X`

4.28. Definición de nuevos comandos

En la literatura matemática es muy frecuente la repetición de expresiones iguales o similares. La definición de nuevos comandos para simplificar múltiples repeticiones es prácticamente una necesidad. En la sección 3.24 ya habíamos presentado las instrucciones `\newcommand` y `\providecommand` para la definición de nuevos comandos; en la presente sección ilustraremos su uso con numerosos ejemplos.

Recuérdese que el nombre de un nuevo comando puede contener únicamente letras, mayúsculas o minúsculas, pero no dígitos ni otros símbolos. L^TE_X permite que los comandos sean usados tanto en modo normal como en modo matemático, si se añade la declaración `\ensuremath{...}` en la *definición* del nuevo comando.

Comandos sin argumentos. Estos comandos se definen en la forma

`\newcommand{\nombre}{definición}`

Podemos usar este tipo de definiciones para expresiones matemáticas fijas que aparezcan frecuentemente en un documento.

Ejemplo La flecha corta → se obtiene con `\to` pero la instrucción para la flecha larga → es exponencialmente más larga: `\longrightarrow`. Es posible definir el nuevo comando `\tto` (por ejemplo) para obtener la última flecha. Específicamente:

`\newcommand{\tto}{\longrightarrow}.`

Con esta definición, una expresión como $F : A \longrightarrow B$ se obtiene simplemente a partir de `$F:A\tto B$`.

Ejemplo Si la expresión $X^2 \times Y^2$ aparece múltiples veces en un documento, es conveniente asignarle un nombre, por ejemplo `\pr` (por producto), en la siguiente forma:

`\newcommand{\pr}{X^2\mathbin{\times} Y^2}.`

Ejemplo Los símbolos para los conjuntos \mathbb{N} , \mathbb{Z} , \mathbb{Q} , \mathbb{R} , \mathbb{C} (véase la sección 4.24) son de uso frecuente y es incómodo tener que escribir continuamente `\mathbb{N}`, `\mathbb{Z}`, etc. Podemos definir los comandos `\N` (para \mathbb{N}), `\Z` (para \mathbb{Z}), `\Q` (para \mathbb{Q}), `\R` (para \mathbb{R}), `\C` (para \mathbb{C}), en la siguiente forma:

`\newcommand{\N}{\mathbb{N}}
\newcommand{\Z}{\mathbb{Z}}
\newcommand{\Q}{\mathbb{Q}}
\newcommand{\R}{\mathbb{R}}
\newcommand{\C}{\mathbb{C}}`

La presencia de `\ensuremath` nos permite usar los nuevos comandos `\mathbb{N}`, `\mathbb{Z}`, `\mathbb{Q}`, etc, en modo matemático o en modo normal.

Comandos con argumentos obligatorios. Se pueden definir nuevos comandos con un máximo de nueve argumentos obligatorios cada uno, en la forma:

`\newcommand{\nombre}[n]{definición}`

donde n es el número de argumentos del comando `\nombre`; $1 \leq n \leq 9$. En la *definición*, los n argumentos están representados por los parámetros `#1, #2, ..., #n`.

Ejemplo Si en un documento aparecen muchas derivadas parciales como

$$\frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial g}{\partial x}, \quad \frac{\partial h}{\partial z}$$

es útil definir el comando `\parcial`, de dos argumentos: el primero para la función y el segundo para la variable. Concretamente, definimos

`\newcommand{\parcial}[2]{\frac{\partial #1}{\partial #2}}`

Con esta definición, tenemos:

$$\$ \backslash \text{parcial}\{f\}\{x\} \$ \quad \frac{\partial f}{\partial x}$$

$$\$ \backslash [\text{parcial}\{v\}\{x\}(a) = -\text{parcial}\{u\}\{y\}(a)] \$ \quad \frac{\partial v}{\partial x}(a) = -\frac{\partial u}{\partial y}(a)$$

Ejemplo Podemos definir la instrucción `\upla` de 2 argumentos para producir vectores coordenados o “uplas”. El primer argumento representa el nombre de cada una de las coordenadas y el segundo es el número de éstas. La definición concreta es:

`\newcommand{\upla}[2]{(#1_1, #1_2, \ldots, #1_{#2})}`

Podemos ahora escribir “uplas” en la forma:

$$\$ \backslash \text{upla}\{a\}\{n\} \$ \quad (a_1, a_2, \dots, a_n)$$

$$\$ \backslash \text{upla}\{b\}\{n+1\} \$ \quad (b_1, b_2, \dots, b_{n+1})$$

$$\$ \backslash \text{bigl}\| \text{upla}\{x\}\{n\} \text{bigr}\| = 1 \$ \quad \|(x_1, x_2, \dots, x_n)\| = 1$$

Ejemplo Vamos a definir un macro, con 7 argumentos, para transformaciones de Möbius:

`\newcommand{\mobius}[7]{\left(\begin{matrix} #2 & #3 & #4 \\ #5 & #6 & #7 \end{matrix}\right)}`

Obsérvese que la barra vertical `|` se obtiene con `\left|` y para completar el par es necesario `\right.`. (tal como se explicó en la sección 4.20). Con esta definición, podemos escribir:

$$\begin{aligned} \$\backslash\text{mobius}\{z\}\{a\}\{b\}\{c\}\{d\}\{e\}\{f\}\$ & \quad \left(z \left| \begin{array}{ccc} a & b & c \\ d & e & f \end{array} \right. \right) \\ \$\backslash\text{mobius}\{\bar{z}\}\{a\}\{b\}\{i\}\{-i\}\{-1\}\{2i+1\}\$ & \quad \left(\bar{z} \left| \begin{array}{ccc} a & b & i \\ -i & -1 & 2i+1 \end{array} \right. \right) \\ T(z) = \left(z \left| \begin{array}{ccc} a & b & c \\ a' & b' & c' \end{array} \right. \right) & = \left(z \left| \begin{array}{ccc} 0 & 1 & \infty \\ a' & b' & c' \end{array} \right. \right) \circ \left(z \left| \begin{array}{ccc} a & b & c \\ 0 & 1 & \infty \end{array} \right. \right) \end{aligned}$$

```
\[ T(z)=\text{mobius}\{z\}\{a\}\{b\}\{c\}\{a'\}\{b'\}\{c'\} = \text{mobius}\{z\}\{0\}\{1\}\{\infty\}
\{a'\}\{b'\}\{c'\} \circ \text{mobius}\{z\}\{a\}\{b\}\{c\}\{0\}\{1\}\{\infty\} \]
```

Comandos con un argumento opcional. LATEX permite definir comandos con un (y sólo un) argumento opcional, además de los argumentos obligatorios. Un comando de tal naturaleza se define en la forma:

`\newcommand{\nombre}[n][defecto]{definición}`

donde `n` es el número de argumentos del comando `\nombre`, $1 \leq n \leq 9$, y `defecto` es el valor que asume por defecto el único argumento opcional. El número `n` es el número *total* de argumentos, incluyendo el argumento opcional. En la `definición`, el argumento opcional está representado por el parámetro `#1`; los demás argumentos son `#2`, `#3`, ..., `#n`.

Ejemplo. En este ejemplo vamos a definir un comando similar al comando `\upla` definido arriba pero con tres argumentos, el primero de ellos opcional. El nuevo comando `\kupla` debe producir algo de la forma (a_1, \dots, a_k) . El parámetro `#2` representa el nombre de cada coordenada (a , b , c , etc), el parámetro `#3` representa el primer sub-índice y el parámetro `#1`, que es el opcional, representa el último sub-índice, el cual es k por defecto. La definición de `\kupla` es:

```
\newcommand{\kupla}[3][k]{(#2_{#3},\ldots,#2_{#1})}
```

Con esta definición podemos escribir

$$\begin{aligned} \$\backslash\text{kupla}\{a\}\{1\}\$ & \quad (a_1, \dots, a_k) \\ \$\backslash\text{kupla}\{x\}\{p\}\$ & \quad (x_p, \dots, x_k) \\ \$\backslash\text{kupla}\{m\}\{b\}\{1\}\$ & \quad (b_1, \dots, b_m) \\ \$\backslash\text{kupla}\{n\}\{b\}\{i\}\$ & \quad (b_i, \dots, b_n) \end{aligned}$$

4.29. Separación de expresiones matemáticas en el margen derecho

Con frecuencia aparecen en un párrafo fórmulas o expresiones matemáticas no desplegadas que L^AT_EX es incapaz de dividir o separar en el margen derecho. L^AT_EX invade el margen ocasionando lo que en jerga T_EX-nica se denomina un ‘overfull’. Ningún ‘overfull’ detiene el procesamiento del documento pero origina una advertencia similar a la siguiente:

```
Overfull \hbox (16.84758pt too wide) detected at line 2116
```

donde la cantidad en puntos (unidades pt) indica el exceso en el margen derecho. Tales advertencias quedan consignadas en el archivo ‘`—.log`’ generado por L^AT_EX. Al procesar un documento con la opción `draft` (Tabla 2.2), se producen “cajas negras” ■ de advertencia en los sitios neurálgicos. Cada ‘overfull’ requiere atención personal por parte del usuario.

En el caso de fórmulas o expresiones matemáticas, lo mejor es dividir la expresión en dos partes y usar `\linebreak` (Tabla 3.3) para preservar la justificación en el margen derecho. Una `$expresión$` matemática determinada se puede dividir en dos partes, por ejemplo, `$expresión$` y `$síón$`, y se puede reemplazar por `$expresión$\linebreak $síón$` en el documento fuente. Recuérdese que `\linebreak` “estira” proporcionalmente todos los caracteres y espacios en el renglón actual hasta tocar el margen derecho y comienza un nuevo renglón. Este mismo procedimiento se puede utilizar para forzar una división particular, incluso si no hay problemas de ‘overfull’.

El paquete `amsmath` dispone del comando `\nobreakdash` para evitar separaciones en el margen derecho después de un guión, en expresiones como *p*-subgrupo, *n*-lineal, *p*-ádico, *q*-binomial, etc; `\nobreakdash` se coloca antes del guión.

Ejemplo Para impedir que haya una separación entre el guión y el término adjunto escribimos, por ejemplo,

`p\nobreakdash-subgrupo` para la expresión *p*-subgrupo.

`n\nobreakdash-lineal` para la expresión *n*-lineal.

`p\nobreakdash-ádico` para la expresión *p*-ádico.

Ejemplo Si la expresión ‘*p*-subgrupo’ es de uso frecuente en un documento, es buena idea definir para ella un comando propio, por ejemplo, `\psub`:

`\newcommand{\psub}{\p\nobreakdash-sub\,-gru\,-po}`

De esta forma, no habrá una separación inmediatamente después del guión pero la división en sílabas de la palabra ‘subgrupo’ será correcta debido a la partición señalada con `sub\,-gru\,-po`. Al indicar la partición silábica evitamos, de paso, posibles casos de ‘overfull’.

4.30. Alineación y numeración de fórmulas

El entorno básico de L^AT_EX para desplegar y numerar una fórmula es

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

El entorno **equation*** de **amsmath** despliega fórmulas sin numerarlas y es equivalente a `\[... \]`. L^AT_EX posee el contador **equation** para fórmulas desplegadas, el cual se incrementa en 1 con cada aparición del entorno `\begin{equation} ... \end{equation}` (sobre el uso de contadores véase la sección 3.25). Por defecto, L^AT_EX numera las fórmulas consecutivamente en la forma (1), (2), (3), ..., en el estilo **article**, y con el número del capítulo, en la forma (5.1), (5.2), (5.3), ..., en los estilos **book** y **report**. Se puede modificar el estilo de la numeración, tal como se explica en la sección 4.31.

Con el entorno **equation** podemos usar `\label{...}`, `\ref{...}` y `\pageref{...}`, tal como lo haríamos con cualquier otro contador. En el siguiente ejemplo se usa la etiqueta `\label{derivada}`, inmediatamente después de `\begin{equation}`, y se hace luego una referencia cruzada con `\ref{derivada}`.

Ejemplo

La función f es derivable en a si

$$\lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} \quad (4.1)$$

existe. En tal caso, el límite (4.1) se denota con $f'(a)$.

La función f es derivable en a si

```
\begin{equation}\label{derivada}
\lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}
\end{equation}
```

existe. En tal caso, el límite (`\ref{derivada}`) se denota con $f'(a)$.

Los números de las fórmulas desplegadas aparecen, por defecto, en el lado derecho, pero con la opción **leqno** del paquete **amsmath** podemos hacer que tales números aparezcan a la izquierda. Así, al cargar el paquete **amsmath** en la forma

```
\usepackage[leqno]{amsmath}
```

las ecuaciones se numerarán en el lado izquierdo de las páginas. Para otras opciones de numeración de fórmulas, véase la sección 4.31.

Cuando un despliegue contiene varias fórmulas, o fórmulas muy grandes, se hace necesario dividirlas en dos o más renglones. Para manejar este tipo de situaciones, L^AT_EX solamente tiene el entorno `eqnarray` (véase 4.30.7) cuyas posibilidades son limitadas. El paquete `amsmath` posee, por el contrario, herramientas muy sofisticadas para el manejo de todo tipo de despliegues y alineaciones. Los entornos incluidos en el paquete `amsmath`, y descritos en la presente sección, son:

<code>multline</code>	<code>align</code>	<code>aligned</code>	<code>split</code>
<code>gather</code>	<code>flalign</code>	<code>gathered</code>	

Hay otro entorno, `alignat`, pero sus efectos se pueden conseguir de manera más simple con `align` y, por tal razón, no lo describiremos aquí.

4.30.1. División de fórmulas con `multline`

El entorno `multline` se usa principalmente para dividir fórmulas, sin alinearlas, en dos o más renglones, que se separan con `\backslash\backslash`. El primer renglón se obtiene cargado a la izquierda, el último a la derecha y los renglones intermedios aparecen centrados. Todo el despliegue aparece numerado en el último renglón. El entorno `multline*` no produce numeración.

Ejemplo Una fórmula dividida en dos renglones, con numeración.

$$\begin{aligned} \frac{f(a+h)-f(a)}{h} - \frac{\partial f}{\partial x}(a) = \\ \frac{u(a+h)-u(a)-d_a u(h)}{h} + i \frac{v(a+h)-v(a)-d_a v(h)}{h} \end{aligned} \quad (4.2)$$

```
\begin{multline}
\frac{f(a+h)-f(a)}{h}-\frac{\partial f}{\partial x}(a)=\\
\frac{u(a+h)-u(a)-d_a u(h)}{h}+i\frac{v(a+h)-v(a)-d_a v(h)}{h}
\end{multline}
```

Ejemplo La fórmula del ejemplo anterior, sin numeración.

$$\begin{aligned} \frac{f(a+h)-f(a)}{h} - \frac{\partial f}{\partial x}(a) = \\ \frac{u(a+h)-u(a)-d_a u(h)}{h} + i \frac{v(a+h)-v(a)-d_a v(h)}{h} \end{aligned}$$

```
\begin{multline*}
\frac{f(a+h)-f(a)}{h}-\frac{\partial f}{\partial x}(a)=\\
\frac{u(a+h)-u(a)-d_a u(h)}{h}+i\frac{v(a+h)-v(a)-d_a v(h)}{h}
\end{multline*}
```

Ejemplo La fórmula para descomposición en fracciones parciales, dividida en cuatro renglones, con numeración.

$$\begin{aligned} \frac{P(x)}{Q(x)} = & \left[\frac{A_{11}}{x - a_1} + \cdots + \frac{A_{1m_1}}{(x - a_1)^{m_1}} \right] + \cdots \\ & + \left[\frac{A_{k1}}{x - a_k} + \cdots + \frac{A_{km_k}}{(x - a_k)^{m_k}} \right] \\ & + \left[\frac{B_{11} + C_{11}}{x^2 + b_1x + c_1} + \cdots + \frac{B_{1r_1} + C_{1r_1}}{(x^2 + b_1x + c_1)^{r_1}} \right] + \cdots \\ & + \left[\frac{B_{n1} + C_{n1}}{x^2 + b_nx + c_n} + \cdots + \frac{B_{nr_1} + C_{nr_n}}{(x^2 + b_nx + c_n)^{r_n}} \right] \end{aligned} \quad (4.3)$$

```
\begin{multiline}
\frac{P(x)}{Q(x)}=\left[\frac{A_{11}}{x-a_1}+\cdots+\frac{A_{1m_1}}{(x-a_1)^{m_1}}\right]+\cdots\\
+\left[\frac{A_{k1}}{x-a_k}+\cdots+\frac{A_{km_k}}{(x-a_k)^{m_k}}\right]\\
+\left[\frac{B_{11}+C_{11}}{x^2+b_1x+c_1}+\cdots+\frac{B_{1r_1}+C_{1r_1}}{(x^2+b_1x+c_1)^{r_1}}\right]+\cdots\\
+\left[\frac{B_{n1}+C_{n1}}{x^2+b_nx+c_n}+\cdots+\frac{B_{nr_1}+C_{nr_n}}{(x^2+b_nx+c_n)^{r_n}}\right]
\end{multiline}
```

4.30.2. Alineaciones con `gather`

El entorno `gather` se asemeja a `multiline` pero cada uno de los renglones aparece centrado y numerado. Usando `\notag` podemos eliminar la numeración en renglones particulares. La versión `gather*` no produce numeración alguna.

Ejemplo Entorno `gather`, numeración automática de cada renglón.

$$A + B := \{x + y \mid x \in A, y \in B\}, \quad (4.4)$$

$$AB := \{xy \mid x \in A, y \in B\}, \quad (4.5)$$

$$-A := \{-x \mid x \in A\}, \quad (4.6)$$

$$A^{-1} := \{a^{-1} \mid a \in A, a \neq 0\} \quad (4.7)$$

```
\begin{gather}
A+B:=\{x+y \mid x\in A, y\in B\},\\
AB:=\{xy \mid x\in A, y\in B\},\\
-A:=\{-x \mid x\in A\},\\
A^{-1}:=\{a^{-1} \mid a\in A, a\neq 0\}
\end{gather}
```

Ejemplo Entorno **gather***; no se obtiene ninguna numeración.

$$\begin{aligned} A + B &:= \{x + y \mid x \in A, y \in B\}, \\ AB &:= \{xy \mid x \in A, y \in B\}, \\ -A &:= \{-x \mid x \in A\}, \\ A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\} \end{aligned}$$

```
\begin{gather*}
A+B:=\{x+y \mid x\in A, \ y\in B\}, \\
AB:=\{xy \mid x\in A, \ y\in B\}, \\
-A:=\{-x \mid x\in A\}, \\
A^{-1}:=\{a^{-1} \mid a\in A, \ a\neq 0\}
\end{gather*}
```

4.30.3. Alineaciones con align

El entorno **align** permite alinear fórmulas; con **\backslash\backslash** se separan los diferentes renglones y se coloca **&** inmediatamente antes del símbolo con respecto al cual se hace la alineación en cada renglón. Cada uno de los renglones aparece numerado pero usando **\notag** podemos eliminar la numeración en renglones particulares. La versión **align*** no produce numeración alguna.

Ejemplo

$$\begin{aligned} |z + \xi|^2 &= (z + \xi)(\overline{z + \xi}) \\ &= |z|^2 + z\bar{\xi} + \bar{z}\xi + |\xi|^2 \\ &\leq |z|^2 + 2|z||\xi| + |\xi|^2 \\ &= (|z| + |\xi|)^2 \end{aligned}$$

```
\begin{align*}
|z+\xi|^2 &= (z+\xi)(\overline{z+\xi}) \\
&= |z|^2 + z\overline{\xi} + \overline{z}\xi + |\xi|^2 \\
&\leq |z|^2 + 2|z||\xi| + |\xi|^2 \\
&= (|z| + |\xi|)^2
\end{align*}
```

Ejemplo Compárese con el último ejemplo de la página 135.

$$A + B := \{x + y \mid x \in A, y \in B\}, \tag{4.8}$$

$$AB := \{xy \mid x \in A, y \in B\}, \tag{4.9}$$

$$-A := \{-x \mid x \in A\}, \tag{4.10}$$

$$A^{-1} := \{a^{-1} \mid a \in A, a \neq 0\} \tag{4.11}$$

```
\begin{align}
A+B &:= \{x+y \mid x \in A, y \in B\}, \\
AB &:= \{xy \mid x \in A, y \in B\}, \\
-A &:= \{-x \mid x \in A\}, \\
A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\}
\end{align}
```

Ejemplo Numeración de renglones particulares. Se elimina la numeración automática en el primer y el tercer renglones con `\notag`.

$$\begin{aligned} A + B &:= \{x + y \mid x \in A, y \in B\}, \\ AB &:= \{xy \mid x \in A, y \in B\}, \end{aligned} \tag{4.12}$$

$$\begin{aligned} -A &:= \{-x \mid x \in A\}, \\ A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\} \end{aligned} \tag{4.13}$$

```
\begin{align}
A+B &:= \{x+y \mid x \in A, y \in B\}, \notag \\
AB &:= \{xy \mid x \in A, y \in B\}, \\
-A &:= \{-x \mid x \in A\}, \notag \\
A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\}
\end{align}
```

El entorno `align` también se puede usar para alinear fórmulas en dos o más columnas. Para separar las columnas se usan símbolos `&`, adicionales a los símbolos de alineación en cada columna.

Ejemplo Despliegue con tres columnas, cada una de ellas alineada por el símbolo `=`.

$$\begin{array}{lll} x = ax + b & X = uX + v & A = aA + B \\ x' = ax' + b & X' = uX' + v & A' = aA' + B' \\ y = (1-a)y & Y = (1-u)Y & B = (1-a)B \\ y' = (1-b)y' & Y' = (1-v)Y' & B' = (1-b)B' \end{array}$$

```
\begin{align*}
x &\&= ax+b & X &\&= uX+v & A &\&= aA+B \\
x' &\&= ax'+b & X' &\&= uX'+v & A' &\&= aA'+B' \\
y &\&= (1-a)y & Y &\&= (1-u)Y & B &\&= (1-a)B \\
y' &\&= (1-b)y' & Y' &\&= (1-v)Y' & B' &\&= (1-b)B'
\end{align*}
```

Ejemplo Despliegue con dos columnas, la primera alineada por el símbolo de igualdad y la segunda por la instrucción `\text{...}`.

$$\begin{array}{ll} a * (a' * b) = (a * a') * b & \text{por la ley asociativa} \\ = e * b & \text{por la definición de } a' \\ = b & \text{por ser } e \text{ elemento identidad} \end{array}$$

```
\begin{aligned}
a*(a'*b)&= (a*a')*b \& \text{por la ley asociativa}\\
&\&=e*b \& \text{por la definición de}\backslash a'\backslash \\
&\&=b \& \text{por ser \$e\$ elemento identidad}
\end{aligned}
```

Con cualquiera de los entornos de alineación se pueden insertar renglones de texto en el despliegue, utilizando `\intertext{...}`. Este comando solamente se puede usar después de `\backslash` y es especialmente útil con `align` ya que se preserva la alineación.

Ejemplo A continuación se usa `\intertext` para añadir líneas de texto en la alineación. Obsérvese que las tres igualdades aparecen alineadas por el símbolo `=`, lo cual no sucedería si simplemente se usara el entorno `equation` o `\[...]` para cada una de ellas.

Puesto que la igualdad

$$(fg)' = f'g + fg' \quad (4.14)$$

se puede escribir como

$$fg' = (fg)' - f'g,$$

se concluye entonces que

$$\int fg' = \int (fg)' - \int f'g. \quad (4.15)$$

Puesto que la igualdad

```
\begin{aligned}
&(fg)' \&= f'g + fg'\\
&\intertext{se puede escribir como} \\
&fg' \&= (fg)' - f'g, \notag\\
&\intertext{se concluye entonces que} \\
&\int fg' \&= \int (fg)' - \int f'g.
\end{aligned}
```

4.30.4. Alineaciones con `split`

El entorno `split` no es independiente y solamente se puede usar dentro de otros entornos, como `equation` o `align`. Se usa para alineaciones, en forma similar a `align`, pero todo el despliegue recibe un único número y no cada uno de los renglones, como sucede con `align`. Dicho número aparece verticalmente centrado.

No existe la versión `split*` ya que el propósito de `split` es hacer alineaciones completas que reciban un número, o alineaciones encajadas y numeradas dentro de otras alineaciones.

Ejemplo Compárese con el primer ejemplo de la página 137. La alineación recibe un único número, verticalmente centrado.

$$\begin{aligned} A + B &:= \{x + y \mid x \in A, y \in B\}, \\ AB &:= \{xy \mid x \in A, y \in B\}, \\ -A &:= \{-x \mid x \in A\}, \\ A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\} \end{aligned} \tag{4.16}$$

```
\begin{equation}
\begin{array}{l}
\begin{array}{l}
A+B &:= \{x+y \mid x \in A, y \in B\}, \\
AB &:= \{xy \mid x \in A, y \in B\}, \\
-A &:= \{-x \mid x \in A\}, \\
A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\}
\end{array}
\end{array}
\end{equation}
```

Ejemplo Compárese con el primer ejemplo de la página 134. Para hacer la alineación se usó ``, comando con el cual se crea una caja invisible de anchura suficiente para contener $f(a + h)$. El uso de `` (sección 3.15.5) es corriente en este tipo de situaciones y permite obtener el espacio en blanco adecuado.

$$\frac{f(a + h) - f(a)}{h} - \frac{\partial f}{\partial x}(a) = \frac{u(a + h) - u(a) - d_a u(h)}{h} + i \frac{v(a + h) - v(a) - d_a v(h)}{h} \tag{4.17}$$

```
\begin{equation}
\begin{array}{l}
\begin{array}{l}
&\frac{f(a+h)-f(a)}{h}-\frac{\partial f}{\partial x}(a)=\\
&\phantom{\frac{f(a+h)-f(a)}{h}}\phantom{d_a}\frac{u(a+h)-u(a)-d_a u(h)}{h}+i\frac{v(a+h)-v(a)-d_a v(h)}{h}
\end{array}
\end{array}
\end{equation}
```

Ejemplo La siguiente cadena de igualdades está alineada por =. Para las dos expresiones numeradas usamos `split` (ya que no caben en una sola línea); para los demás renglones usamos `\notag`. Obsérvese el uso de `\hspace{...}` para añadir el espacio apropiado. Se han usado las etiquetas `\label{ec:uno}` y `\label{ec:dos}` con el objeto de poder hacer referencia posterior a las expresiones correspondientes por medio de `\ref{ec:uno}` y `\ref{ec:dos}`.

El único propósito de los dos `split` usados para este despliegue es hacer que las expresiones (4.18) y (4.19) lleven sus respectivos números, adecuadamente colocados. Para obtener el despliegue sin numeración alguna, simplemente usamos `align*`, como se aprecia en el ejemplo de la página siguiente.

$$\begin{aligned}
 \int (\log x)^3 dx &= [x(\log x)^2 - 2x(\log x) + 2x] \log x \\
 &\quad - \int \frac{1}{x} [x(\log x)^2 - 2x(\log x) + 2x] dx \\
 &= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
 &\quad - \int (\log x)^2 dx + 2[x \log x - x] - 2x
 \end{aligned} \tag{4.18}$$

$$= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\ - [x(\log x)^2 - 2x(\log x) + 2x] + 2[x \log x - x] \quad (4.19) \\ - 2x$$

```

\begin{align}
\int (\log x)^3 dx &= \bigl[ x(\log x)^2 - 2x(\log x) + 2x \bigr] \\
&\quad \log x \notag \\
&\quad & \& \hspace{2.1cm} - \int \frac{1}{x} \bigl[ x(\log x)^2 - 2x(\log x) \\
&+ 2x \bigr] dx \notag \\
\begin{split}
&= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
&\quad & \& \hspace{2cm} - \int (\log x)^2 dx + 2[x \log x - x] - 2x \\
\end{split} \label{ec:uno} \\
\begin{split}
&= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
&\quad & \& \hspace{0.7cm} - \bigl[ x(\log x)^2 - 2x(\log x) + 2x \bigr] \\
&+ 2[x \log x - x] \\
&\quad & \& \hspace{0.7cm} - 2x \\
\end{split} \label{ec:dos} \\
\end{align}

```

Ejemplo Despliegue del ejemplo anterior, sin numeración alguna; no es necesario usar `split`.

$$\begin{aligned}
 \int (\log x)^3 dx &= [x(\log x)^2 - 2x(\log x) + 2x] \log x \\
 &\quad - \int \frac{1}{x} [x(\log x)^2 - 2x(\log x) + 2x] dx \\
 &= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
 &\quad - \int (\log x)^2 dx + 2[x \log x - x] - 2x \\
 &= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
 &\quad - [x(\log x)^2 - 2x(\log x) + 2x] + 2[x \log x - x] - 2x \\
 &= x(\log x)^3 - 3x(\log x)^2 + 6x \log x - 6x.
 \end{aligned}$$

```
\begin{aligned*}
\int (\log x)^3 dx &= \bigl[ x(\log x)^2 - 2x(\log x) + 2x \bigr] \\
&\log x \\
&\hspace{2.1cm} - \int \frac{1}{x} \bigl[ x(\log x)^2 - 2x(\log x) + 2x \bigr] dx \\
&\&= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
&\&\hspace{2cm} - \int (\log x)^2 dx + 2[x \log x - x] - 2x \\
&\&\&= x(\log x)^3 - 2x(\log x)^2 + 2x \log x \\
&\&\&- [x(\log x)^2 - 2x(\log x) + 2x] + 2[x \log x - x] - 2x \\
&\&\&= x(\log x)^3 - 3x(\log x)^2 + 6x \log x - 6x.
\end{aligned*}
```

4.30.5. Alineaciones con `aligned` y `gathered`

Los entornos `aligned` y `gathered` construyen bloques de alineación independientes que pueden ser colocados unos al lado de otros, o pueden ser precedidos o seguidos de otro material horizontal (como paréntesis o corchetes). Los diferentes bloques aparecen verticalmente centrados con respecto al material que los rodea, pero ambos entornos admiten los modificadores `[b]` y `[t]` para cambiar la colocación (véase el último ejemplo de la página 143).

En los bloques construidos con `aligned` hay que utilizar `&` para indicar el símbolo de alineación (como sucede con su homólogo `align`), mientras que en los bloques construidos con `gathered` sólo se requiere separar las diferentes filas con `\backslash\backslash` (como sucede con su homólogo `gather`). Los siguientes ejemplos ilustran el tipo de despliegues para los cuales se deben usar estos dos entornos.

Es necesario tener presente que tanto `aligned` como `gathered` son entornos “subsidiarios” y solamente se pueden usar dentro de despliegues de la forma `\begin{equation} ... \end{equation}`.

Ejemplo El bloque construido con `aligned` va seguido de un corchete, de tamaño adecuado, obtenido con `\right\}`; esto exige usar antes `\left.` (sección 4.20). Todo el despliegue adquiere un número, ya que está bajo el alcance del entorno `equation`.

$$\left. \begin{aligned} K^2 A''(x) - \lambda H(x)A(x) &= 0 \\ (K+1)A(\ell) &= 0 \\ A(0) &= 0 \end{aligned} \right\} \quad (4.20)$$

```

\begin{equation}
\left. \begin{aligned}
& K^2 A''(x) - \lambda H(x) A(x) = 0 \\
& (K+1)A(\text{ell}) = 0 \\
& A(0) = 0
\end{aligned} \right\}
\end{equation}

```

Ejemplo Tres bloques independientes colocados uno al lado del otro. Los dos primeros están construidos con `aligned` y el tercero es una igualdad que ocupa una sola línea. Se usó `\quad` para separar los bloques.

$$\begin{aligned} u_t - u_{xx} &= F(x, t) & u(0, t) &= 0 \\ v_t - v_{xx} &= 0 & u(\pi, t) = v(t, 1) &= 1 & u(1, t) = w(1, t) &= 0 \\ w_t - w_{xx} &= 0 \end{aligned}$$

```

\begin{equation*}
\begin{aligned}
u_t - u_{xx} &= F(x, t) \\
v_t - v_{xx} &= 0 \\
w_t - w_{xx} &= 0
\end{aligned}
\quad
\begin{aligned}
u(0, t) &= 0 \\
u(\pi, t) &= v(t, 1) = 1
\end{aligned}
\quad
\begin{aligned}
u(1, t) &= w(1, t) = 0
\end{aligned}
\end{equation*}

```

Ejemplo Compárese con el primer ejemplo de la página anterior. Se usa aquí `gathered` en lugar de `aligned` y, por lo tanto, no se requiere el símbolo de alineación `&`. Las igualdades aparecen horizontalmente centradas.

$$\left. \begin{array}{l} K^2 A''(x) - \lambda H(x) A(x) = 0 \\ (K + 1)A(\ell) = 0 \\ A(0) = 0 \end{array} \right\} \quad (4.21)$$

```
\begin{equation}
\left. \begin{array}{l}
\begin{gathered}
K^2 A''(x) - \lambda H(x) A(x) = 0 \\
(K + 1)A(\ell) = 0 \\
A(0) = 0
\end{gathered}
\right. \} \\
\end{equation}
```

Ejemplo Compárese con el segundo ejemplo de la página anterior. Se usa aquí `gathered` en lugar de `aligned` y, por lo tanto, no se requiere el símbolo de alineación `&`. Las igualdades aparecen horizontalmente centradas en sus respectivas columnas. Además, los tres bloques están alineados por la parte superior puesto que se usó el modificador `[t]` en los entornos `gathered`.

$$\begin{aligned} u_t - u_{xx} &= F(x, t) & u(0, t) &= 0 & u(1, t) &= w(1, t) = 0 \\ v_t - v_{xx} &= 0 & u(\pi, t) &= v(t, 1) & &= 1 \\ w_t - w_{xx} &= 0 & & & & \end{aligned}$$

```
\begin{equation*}
\begin{gathered}[t]
u_t - u_{xx} = F(x, t) \\
v_t - v_{xx} = 0 \\
w_t - w_{xx} = 0
\end{gathered} \quad \begin{gathered}[t]
u(0, t) = 0 \\
u(\pi, t) = v(t, 1) = 1
\end{gathered} \quad \begin{gathered}[t]
u(1, t) = w(1, t) = 0
\end{gathered}
\end{equation*}
```

4.30.6. Alineaciones con `flalign`

El último de los entornos de alineación del paquete `amsmath` es `flalign`, variación de `align`. Se usa para que las alineaciones en varias columnas se impriman de extremo a extremo de la página, distribuyendo proporcionalmente el espacio en blanco entre ellas (el prefijo `fl` proviene del inglés *flushed*).

Con `flalign`, cada una de las líneas del despliegue adquiere un número pero con `\notag` se puede eliminar la numeración de renglones particulares. La versión `\flalign*` no produce numeración alguna.

Ejemplo Compárese con el último ejemplo de la página 137.

$$\begin{array}{lll} x = ax + b & X = uX + v & A = aA + B \\ x' = ax' + b & X' = uX' + v & A' = aA' + B' \\ y = (1 - a)y & Y = (1 - u)Y & B = (1 - a)B \\ y' = (1 - b)y' & Y' = (1 - v)Y' & B' = (1 - b)B' \end{array}$$

```
\begin{flalign*}
x &= ax+b & X &= uX+v & A &= aA+B\\
x' &= ax'+b & X' &= uX'+v & A' &= aA'+B'\\
y &= (1-a)y & Y &= (1-u)Y & B &= (1-a)B\\
y' &= (1-b)y' & Y' &= (1-v)Y' & B' &= (1-b)B'
\end{flalign*}
```

Ejemplo El despliegue anterior con numeración. Aquí se añadió una columna sin texto (usando un `&` adicional antes de finalizar cada renglón) para evitar que la última columna quedara demasiado cerca de la numeración.

$$\begin{array}{lll} x = ax + b & X = uX + v & A = aA + B \end{array} \quad (4.22)$$

$$\begin{array}{lll} x' = ax' + b & X' = uX' + v & A' = aA' + B' \end{array} \quad (4.23)$$

$$\begin{array}{lll} y = (1 - a)y & Y = (1 - u)Y & B = (1 - a)B \end{array} \quad (4.24)$$

$$\begin{array}{lll} y' = (1 - b)y' & Y' = (1 - v)Y' & B' = (1 - b)B' \end{array} \quad (4.25)$$

```
\begin{flalign}
x &= ax+b & X &= uX+v & A &= aA+B & \\
x' &= ax'+b & X' &= uX'+v & A' &= aA'+B' & \\
y &= (1-a)y & Y &= (1-u)Y & B &= (1-a)B & \\
y' &= (1-b)y' & Y' &= (1-v)Y' & B' &= (1-b)B' &
\end{flalign}
```

4.30.7. Alineaciones con el entorno `eqnarray` de L^AT_EX

L^AT_EX tiene su propio entorno para alineaciones, `eqnarray` (y la versión `eqnarray*` para alineaciones no-numeradas), pero su uso es muy limitado y sus resultados, por lo general, bastante insatisfactorios. Recomendamos que el lector use siempre los entornos que hemos descrito (`multiline`, `gather`, `align`, etc) pertenecientes al paquete `amsmath`. El entorno `eqarray` es similar a `array` (sección 4.23) y trata las alineaciones como matrices, añadiendo espacio extra entre columnas, tal como se puede apreciar en el siguiente ejemplo.

Ejemplo Comparación entre `eqnarray*` y `align*`:

$$\begin{aligned} |x - y| &\leq |x - z| + |z - y| \\ &\leq r/2 + r/2 \\ &= r \end{aligned}$$

```
\begin{eqnarray*}
|x-y| &\leq |x-z|+|z-y| \\
&\leq r/2+r/2 \\
&= r
\end{eqnarray*}
```

$$\begin{aligned} |x - y| &\leq |x - z| + |z - y| \\ &\leq r/2 + r/2 \\ &= r \end{aligned}$$

```
\begin{align*}
|x-y| &\leq |x-z|+|z-y| \\
&\leq r/2+r/2 \\
&= r \\
\end{align*}
```

L^AT_EX controla la separación de columnas en los entornos `array` y `eqnarray` por medio del parámetro `\arraycolsep`. El valor que tiene por defecto este parámetro es de casi 2mm, razón por la cual se obtiene espacio excesivo alrededor de los símbolos de alineación. Podemos disminuir el valor de `\arraycolsep` usando `\setlength`; por ejemplo, con

```
\setlength{\arraycolsep}{2pt}
```

el espaciamiento es adecuado.

4.30.8. Espaciamiento vertical en alineaciones

En todos los entornos para alineaciones (`multline`, `align`, `eqnarray`, etc) se puede usar `\[\longitud]` para añadir espacio vertical adicional entre renglones (este comando fue presentado en la sección 3.9).

Ejemplo En la siguiente alineación aparecen muchas de las facetas ilustradas en ejemplos anteriores. Primero, se usa `split` dentro de `equation` para que todo el despliegue adquiera un único número, a saber, (4.26). Podemos hacer referencia a este número porque hemos creado la etiqueta `\label{propiedades}`. En segundo lugar, se han construido tres bloques diferentes con `aligned` para poder encerrarlos posteriormente con llaves exteriores `}`, del tamaño correcto. Es necesario usar `\left.` para cada uno de los tres `\right\}` utilizados. Finalmente, se usó `\[2mm]` para añadir espacio vertical adicional y separar así las tres propiedades presentadas.

$$\begin{aligned} & A \cup A = A \\ & A \cap A = A \end{aligned} \quad \left. \begin{array}{l} \text{Idempotencia} \\ \text{Commutatividad} \end{array} \right\} \quad (4.26)$$

$$\begin{aligned} & A \cup B = B \cup A \\ & A \cap B = B \cap A \end{aligned} \quad \left. \begin{array}{l} \text{Asociatividad} \end{array} \right\}$$

$$\begin{aligned} & A \cup (B \cup C) = (A \cup B) \cup C \\ & (A \cap B) \cap C = A \cap (B \cap C) \end{aligned} \quad \left. \begin{array}{l} \text{Asociatividad} \end{array} \right\}$$

```
\begin{equation}\label{propiedades}
\begin{split}
&\left.\begin{aligned}
&A \cup A = A \\
&A \cap A = A
\end{aligned}\right\} \quad \text{Idempotencia} \quad [2mm]
&\left.\begin{aligned}
&A \cup B = B \cup A \\
&A \cap B = B \cap A
\end{aligned}\right\} \quad \text{Commutatividad} \\
&\left.\begin{aligned}
&A \cup (B \cup C) = (A \cup B) \cup C \\
&(A \cap B) \cap C = A \cap (B \cap C)
\end{aligned}\right\} \quad \text{Asociatividad}
\end{split}
\end{equation}
```

4.30.9. Control sobre cambios de página en alineaciones

Las alineaciones en varios renglones, creadas con los entornos propios del paquete **amsmath** (**multline**, **align**, **gather**, etc), no son divididas en dos o más páginas. Es decir, si una determinada alineación no puede ser acomodada en el resto de la página, L^AT_EX inicia una nueva página, dejando en blanco una porción de la anterior o, peor aun, imprime sólo una parte de la alineación y omite el resto. Para prevenir estos efectos indeseables, es conveniente usar la declaración global **\allowdisplaybreaks** antes de la primera alineación del documento (o en el preámbulo). Cuando dicha declaración está vigente, L^AT_EX hace cambios de página, en medio de alineaciones, según la disponibilidad de espacio en la página en proceso.

Además, el usuario puede forzar cambios de página en alineaciones usando la instrucción **\displaybreak** inmediatamente antes del **** en el que desea terminar la página. Se puede usar **\displaybreak** en todos los entornos del paquete **amsmath** pero no en el entorno **eqnarray** de L^AT_EX.

Es necesario advertir que para ciertas alineaciones, **amsmath** encierra el contenido en cajas inseparables y, como consecuencia, los comandos **\allowdisplaybreaks** y **\displaybreak** no tienen efecto. Esto sucede en entornos como **split** y **gathered**. La filosofía de **amsmath** es: los casos problemáticos requieren atención personal por parte del usuario. La solución más simple es dividir una alineación excesivamente extensa en dos o más alineaciones.

4.31. Opciones para la numeración de fórmulas

4.31.1. Colocación y numeración de fórmulas

El paquete **amsmath** ofrece tres opciones para la colocación y numeración de fórmulas desplegadas:

- reqno** Fórmulas centradas, numeración a la derecha. Opción asumida por defecto.
- leqno** Fórmulas centradas, numeración a la izquierda.
- fleqn** Las fórmulas no aparecen centradas sino a una distancia fija (sangría) del margen izquierdo. Esta sangría es mayor que la que se usa al comienzo de párrafos.

Para acceder a la opción **fleqn**, por ejemplo, cargamos el paquete **amsmath** en la forma **\usepackage[fleqn]{amsmath}**.

4.31.2. Jerarquía de la numeración

\LaTeX numera las fórmulas consecutivamente en la forma (1), (2), (3), ..., en el estilo `article`, y con el número del capítulo, en la forma (6.1), (6.2), (6.3), ..., en el estilo `book`. Para hacer que los números de las fórmulas también incluyan el número de la sección, podemos proceder de dos formas. El primer procedimiento es redefinir el comando $\text{\LaTeX} \backslash\text{theequation}$, que controla la numeración de ecuaciones:

```
\renewcommand{\theequation}{\thesection.\arabic{equation}}
```

Esto funciona bien excepto por el hecho de que el conteo de ecuaciones, en el estilo `article`, es acumulativo y *no* se actualiza a 0 al comienzo de una nueva sección, a menos que lo hagamos nosotros mismos con `\setcounter`. Una alternativa más cómoda es usar la instrucción

```
\numberwithin{equation}{division}
```

del paquete `amsmath`, que hace que las fórmulas se enumeren con respecto a la *división* del documento señalada (véase más sobre partes o divisiones de un documento en la sección 6.1). Así por ejemplo, al escribir `\numberwithin{equation}{section}`, las fórmulas numeradas incluyen el número de la sección.

4.31.3. Numeración forzada

\LaTeX numera automáticamente las fórmulas desplegadas, pero existe la manera de forzar o modificar la numeración, incluso dentro de un entorno `*`, con la instrucción `\tag{etiqueta}`. Con `\tag*{etiqueta}`, la *etiqueta* aparece sin paréntesis exteriores. Los comandos `\tag` y `\tag*` son útiles cuando el usuario desea utilizar *tags* no-numéricos, como asteriscos, letras, palabras, etc. El contador `equation` de fórmulas *no* se incrementa.

Ejemplo En el siguiente despliegue se usan asteriscos como etiquetas. Es necesario observar que hay una diferencia entre el símbolo que se obtiene con la tecla `*` (a saber: `*`) y el que se obtiene con `\ast` (a saber: `\ast`).

$$A + B := \{x + y \mid x \in A, y \in B\}, \quad (*)$$

$$AB := \{xy \mid x \in A, y \in B\}, \quad (**)$$

```
\begin{aligned}
A+B &:= \{x+y \mid x \in A, y \in B\}, \tag{\$\\ast\$}\\
AB &:= \{xy \mid x \in A, y \in B\}, \tag{\$\\ast\\ast\$}
\end{aligned}
```

4.31.4. Numeración subordinada

El paquete **amsmath** tiene el entorno **subequations** para numeración subordinada, el cual se ilustra en el ejemplo que sigue. Las igualdades del ejemplo reciben los números (4.27a), (4.27b), (4.27c) y (4.27d) ya que la alineación está bajo el alcance del entorno **subequations**. Podemos hacer referencia a cada una de las igualdades colocando `\label{...}` antes de `\backslash`, y también podemos referirnos al listado o despliegue total colocando una etiqueta `\label{...}` inmediatamente después de `\begin{subequations}`.

Ejemplo

$$A + B := \{x + y \mid x \in A, y \in B\} \quad (4.27a)$$

$$AB := \{xy \mid x \in A, y \in B\} \quad (4.27b)$$

$$-A := \{-x \mid x \in A\} \quad (4.27c)$$

$$A^{-1} := \{a^{-1} \mid a \in A, a \neq 0\} \quad (4.27d)$$

En (4.27) aparecen las definiciones de nuevos conjuntos de números reales: (4.27a) define la suma de subconjuntos, (4.27b) el producto, (4.27c) el opuesto y (4.27d) el inverso.

```
\begin{subequations}\label{operaciones}
\begin{align}
A+B &:= \{x+y \mid x \in A, y \in B\} \label{suma}\\
AB &:= \{xy \mid x \in A, y \in B\} \label{producto}\\
-A &:= \{-x \mid x \in A\} \label{opuesto}\\
A^{-1} &:= \{a^{-1} \mid a \in A, a \neq 0\} \label{inverso}
\end{align}
\end{subequations}
```

En (`\ref{operaciones}`) aparecen las definiciones de nuevos conjuntos de números reales: (`\ref{suma}`) define la suma de subconjuntos, (`\ref{producto}`) el producto, (`\ref{opuesto}`) el opuesto y (`\ref{inverso}`) el inverso.

También podemos hacer que la numeración subordinada utilice números romanos, en vez de letras, en la forma (4.27i), (4.27ii), (4.27iii), etc. Para ello se debe tener presente que el entorno **subequations** utiliza los contadores **parentequation** y **equation** para producir los números de las fórmulas y, por lo tanto, hay que escribir la instrucción

```
\renewcommand{\theequation}{\theparentequation\roman{equation}}
```

debajo de `\begin{subequations}`. Hay que advertir que con la opción **spanish** del paquete **babel**, la numeración romana i, ii, iii, ... es reemplazada por I, II, III, ... (véase al respecto la nota TEX-nica de la página 27).

4.31.5. Referencias cruzadas

Para hacer más fácil las referencias cruzadas a los números de las fórmulas, el paquete `amsmath` tiene el macro `\eqref`. Se diferencia del usual comando `\ref` de L^AT_EX en que los paréntesis que encierran los números se obtienen automáticamente.

4.31.6. Ajustes en la posición de los números

Para fórmulas desplegadas y numeradas, el paquete `amsmath` tiene mecanismos de control que evitan que la expresión matemática haga contacto o quede demasiado cerca de su número. El comando disponible para el ajuste de la posición de los números de las fórmulas es `\raisetag{longitud}`, que se debe colocar inmediatamente antes del `\backslash` correspondiente al número de la fórmula. La `longitud` deseada puede ser positiva o negativa. Por ejemplo, `\raisetag{-3mm}` traslada el número 3 mm hacia abajo.

4.32. Teoremas y estructuras relacionadas

Los teoremas (¡y sus demostraciones!) son los ingredientes esenciales de las matemáticas pero en artículos o libros también pueden aparecer proposiciones, corolarios, lemas, definiciones, axiomas, notas, conjeturas, casos, notaciones, condiciones, conclusiones, etc. Y en la literatura no-matemática encontramos estructuras similares, tales como reglas, leyes, principios, observaciones, experimentos, hipótesis, etc. Por lo general, estas estructuras se numeran, ya sea en forma independiente o acumulativamente, y es obvio que L^AT_EX no puede proporcionar un entorno específico para cada estructura concebible. En su lugar, L^AT_EX tiene el comando genérico `\newtheorem` con el cual el usuario puede crear cualquier estructura numerada que necesite en un documento. Una vez que hayamos creado una nueva estructura, con el nombre que queramos darle, L^AT_EX nos hace el favor de llevar el conteo de sus sucesivas apariciones y nos permite hacer referencias cruzadas con los ya conocidos recursos `\label{...}`, `\ref{...}` y `\pageref{...}`.

El comando

`\newtheorem{nombre}{rótulo}`

tiene dos argumentos obligatorios: el usuario escoge el `nombre` que le quiere dar al nuevo entorno, y `rótulo` es la palabra o expresión que el usuario quiere que aparezca impresa en el documento final cuando se invoque la estructura en cuestión. Esta instrucción crea el entorno `nombre`, con su respectivo contador.

Ejemplo Vamos a definir tres estructuras típicas: `defin` (para definiciones), `teor` (para teoremas) y `corol` (para corolarios). Estos nombres los escogemos libremente. En el segundo argumento de `\newtheorem` (pero no en el primero) se pueden usar símbolos acentuados (como á, é, í, etc) si se ha cargado el paquete `inputenc` (sección 3.1).

```
\newtheorem{defin}{Definición}
\newtheorem{teor}{Teorema}
\newtheorem{corol}{Corolario}
```

Estas declaraciones pueden aparecer en cualquier parte, pero el sitio más apropiado es el preámbulo del documento, para facilitar las tareas de revisión y evitar errores de repetición. Podemos ahora escribir definiciones, teoremas y corolarios usando `\begin{...}` ... `\end{...}`, tal como se ilustra a continuación. Obsérvese que hemos forzado el cambio de tipo de letra (`\slshape`) en el entorno `teor`; por defecto, L^AT_EX utiliza letra itálica (cursiva) en los entornos creados con `\newtheorem`.

Definición 1. *Una extensión finita, normal y separable \$E\$ de un campo \$F\$ se llama una extensión de Galois de \$F\$.*

El siguiente resultado es el llamado teorema fundamental de la teoría de Galois.

Teorema 1. *Sea \$E\$ una extensión de Galois de \$F\$ y \$K\$ un campo tal que \$F \subseteq K \subseteq E\$. Entonces \$K \hookrightarrow G(E/K)\$ establece una correspondencia biyectiva entre el conjunto de los subcampos de \$E\$ que contienen a \$F\$ y los subgrupos de \$G(E/F)\$.*

Corolario 1. *Para todo subgrupo \$H\$ de \$G(E/F)\$, se tiene \$H = G(E/E_H)\$.*

```
\begin{defin}
Una extensión finita, normal y separable $E$ de un campo $F$ se
llama una extensión de Galois de $F$.
\end{defin}
El siguiente resultado es el llamado teorema fundamental de la
teoría de Galois.
\begin{teor} \slshape
Sea $E$ una extensión de Galois de $F$ y $K$ un campo tal que
$F \subsetneq K \subsetneq E$. Entonces $K \mapsto G(E/K)$ establece una
correspondencia biyectiva entre el conjunto de los subcampos de $E$ que
contienen a $F$ y los subgrupos de $G(E/F)$.
\end{teor}
\begin{corol}
Para todo subgrupo $H$ de $G(E/F)$, se tiene $H = G(E/E_H)$.
\end{corol}
```

4.32.1. Opciones de \newtheorem

El comando `\newtheorem`, cuyo uso típico se ilustró en el ejemplo de la página anterior, admite varias opciones, reseñadas a continuación.

1. Con cualquiera de las estructuras creadas con `\newtheorem` se puede usar `\label{...}` y `\ref{...}`. Por ejemplo, al Teorema 1 de la página anterior podemos adjuntarle la etiqueta `\label{galois}`:

```
\begin{teor}\label{galois}
Sea $E$ una extensión ...
\end{teor}
```

y luego podemos referirnos al *número* asignado a dicho teorema, escribiendo `\ref{galois}`.

2. Con cualquiera de las estructuras creadas con `\newtheorem` se puede usar un argumento opcional para agregar alguna descripción al rótulo de la estructura. Esto se usa principalmente para resaltar la denominación de un resultado importante o su inventor. Por ejemplo, con la estructura `teor` utilizada en la página anterior,

```
\begin{teor}[Teorema de Galois]
Sea $E$ una extensión ...
\end{teor}
```

produce:

Teorema 1 (Teorema de Galois). *Sea E una extensión ...*

3. Por defecto, cada estructura creada con `\newtheorem` se enumera independientemente de las demás. Por ejemplo, si definimos estructuras para definiciones, proposiciones, teoremas y corolarios, éstos aparecerán presentados en la forma: Proposición 1, Proposición 2, Teorema 1, Corolario 1, Definición 1, Teorema 2, Corolario 2, etc. Si queremos que las proposiciones, teoremas y corolarios comparten la secuencia de numeración (en la forma Proposición 1, Proposición 2, Teorema 3, Corolario 4, Teorema 5, Corolario 6, etc.), escribimos:

```
\newtheorem{prop}{Proposición}
\newtheorem{teor}[prop]{Teorema}
\newtheorem{corol}[prop]{Corolario}
```

El argumento opcional `[prop]` indica que los entornos `teor` y `corol` comparten la secuencia de numeración de `prop`, en lugar de utilizar un conteo independiente.

4. `\newtheorem` posee un segundo argumento opcional (escrito al final) con el cual se puede hacer que las estructuras adquieran numeración subordinada con respecto a capítulos, secciones, etc. Así por ejemplo, para que las proposiciones se enumeren con referencia al capítulo (algo como Proposición 7.1, Proposición 7.2, etc, siendo 7 el número del capítulo) escribimos la opción `[chapter]` al final:

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

El contador `prop` de proposiciones tomará el valor 0 al iniciarse un nuevo capítulo. Similarmente, utilizando el argumento `[section]`, podemos hacer que una estructura determinada se enumere con referencia a la sección vigente.

5. Las opciones mencionadas en los numerales 3 y 4 no pueden aparecer juntas en la definición de una estructura pero se pueden combinar. Por ejemplo,

```
\newtheorem{prop}{Proposición}[chapter]
\newtheorem{teor}[prop]{Teorema}
\newtheorem{corol}[prop]{Corolario}
```

hace que los entornos `teor` y `corol` compartan la secuencia de numeración de `prop`, y hagan referencia los tres (`prop`, `teor` y `corol`) al número del capítulo.

4.32.2. El comando `\newtheorem` en el paquete `amsthm`

La *American Mathematical Society* ha diseñado el paquete `amsthm` (no confundirlo con `amsmath`) para brindarle al usuario aún mayor control y flexibilidad sobre `\newtheorem`. Este paquete también define el entorno `proof` para demostraciones (véase la sección 4.32.3 en la página siguiente). Podemos cargar los paquetes `amsmath` y `amsthm` simultáneamente escribiendo, en el preámbulo del documento, `\usepackage{amsmath,amsthm}`.

El paquete `amsthm` posee el comando `\theoremstyle{...}`, con el cual se puede escoger el tipo o estilo de la estructura que se quiere definir. Hay tres estilos disponibles:

<code>plain</code>	Corresponde al estilo estándar de L ^A T _E X. Los rótulos aparecen en negrilla y el texto en letra cursiva (italica). Es el estilo asumido por defecto.
<code>definition</code>	Los rótulos aparecen en negrilla pero el texto aparece en letra normal.
<code>remark</code>	Los rótulos aparecen en letra cursiva (italica) y el texto aparece en letra normal.

Otro de los recursos útiles de **amsthm** es la posibilidad de crear estructuras no-numeradas, por medio de `\newtheorem*`. Esta versión estrella de `\newtheorem` no existe en L^AT_EX estándar.

Para crear estructuras de diferentes tipos, lo más aconsejable es dividir los comandos `\newtheorem` o `\newtheorem*` en grupos, precediendo cada grupo con el comando `\theoremstyle{...}` apropiado.

Ejemplo A continuación se definen estructuras de los tres tipos admitidos por **amsthm**. Estos comandos se deben escribir, preferiblemente, en el preámbulo del documento para facilitar futuras correcciones o modificaciones.

```
\theoremstyle{plain}
  \newtheorem{prop}{Proposición}[section]
  \newtheorem{teor}{prop}{Teorema}
  \newtheorem{corol}{prop}{Corolario}
  \newtheorem{lema}{prop}{Lema}

\theoremstyle{definition}
  \newtheorem{def}{Definición}[section]
  \newtheorem{ejem}{Ejemplo}
  \newtheorem{ejer}{Ejercicio}

\theoremstyle{remark}
  \newtheorem*{nota}{Nota}
  \newtheorem*{notac}{Notación}
```

Es corriente encontrar en la literatura que los números de los teoremas, proposiciones, definiciones, etc aparecen a la izquierda (**3.1 Teorema** en vez de **Teorema 3.1**, **5.7 Definición** en vez de **Definición 5.7**, etc). El paquete **amsthm** permite este tipo de numeración con la instrucción `\swapnumbers`. Para que el estilo sea uniforme a lo largo de todo el documento, esta declaración debe preceder a todos los comandos `\theoremstyle` y `\newtheorem`.

4.32.3. El entorno **proof** del paquete **amsthm**

El paquete **amsthm** tiene pre-definido el entorno **proof**, el cual produce la expresión *Proof.* (con su punto) y el símbolo \square para destacar el fin de la demostración, al final del entorno. Si se usa el paquete **babel** con la opción **spanish**, *Proof.* se convierte en *Demostración.* Pero incluso sin usar **babel**, podemos escoger otra denominación para ‘*Proof.*’ ya que el entorno **proof** tiene un argumento opcional para tal propósito. Por ejemplo, si queremos ‘*Prueba:*’ en lugar de ‘*Proof.*’ o ‘*Demostración.*’, iniciamos el entorno **proof** en la forma

```
\begin{proof}[Prueba:]
```

El argumento opcional de `proof` también es útil para iniciar una demostración con una denominación especial, como '*Demostración del teorema principal*', por ejemplo.

El símbolo \square puede ser reemplazado por otro cualquiera redefiniendo la instrucción `\qedsymbol`. Así por ejemplo, si el usuario prefiere el símbolo \diamondsuit (que se obtiene con `\$\\diamondsuit$`) debe escribir

```
\renewcommand{\qedsymbol}{\$\\diamondsuit$}
```

El símbolo para fin de demostración aparece colocado en el último renglón del entorno `proof`, cargado a la derecha. Si el último renglón es una fórmula desplegada, el símbolo aparece en una línea nueva, muy abajo de la fórmula, lo cual no luce bien. En tales situaciones se debe usar el comando `\qedhere` en el sitio preciso en el que finaliza la demostración, antes de `\end{proof}`⁶. Esto se ilustra en el siguiente ejemplo.

Ejemplo Supóngase que una demostración termina con la igualdad desplegada $\sec^2(x) = 1 + \tan^2(x)$. Si en el documento fuente se escribe

```
\begin{proof}[Demostración.]  
.....  
\[ \sec^2(x)=1+\tan^2(x). \]  
\end{proof}
```

obtenemos lo siguiente:

Demostración.

.....

$$\sec^2(x) = 1 + \tan^2(x).$$

□

Pero si escribimos

```
\begin{proof}[Demostración.]  
.....  
\[ \sec^2(x)=1+\tan^2(x). \qedhere \]  
\end{proof}
```

obtenemos mejores resultados:

Demostración.

.....

$$\sec^2(x) = 1 + \tan^2(x).$$

□

⁶La instrucción `\qedhere` solamente está disponible en la versión 2.0 (y posteriores) de los paquetes `amsmath` y `amsthm`.

4.33. Diagramas conmutativos con `amscd`

En las distribuciones de `amsmath` se incluye también el paquete `amscd`, al cual se accede escribiendo `\usepackage{amscd}`. Los dos paquetes se pueden cargar simultáneamente en la forma `\usepackage{amsmath,amscd}`. El paquete `amscd` permite construir diagramas conmutativos sencillos, con flechas horizontales y verticales *únicamente*. Para diagramas más complejos, que tengan flechas diagonales o flechas punteadas, es recomendable usar paquetes especializados como `pb-diagram` (descrito en la siguiente sección) o paquetes gráficos como `PSTricks` (Capítulo 11) o `PICTEX` (Capítulo 12).

Para que el paquete `amscd` se pueda usar conjuntamente con `babel`, opción `spanish`, hay que escribir `\deactivatequoting` (en el preámbulo del documento o después de `\begin{document}`) ya que `spanish` reserva las combinaciones `>>` y `<<`, utilizadas también por `amscd`.

El paquete `amscd` posee el entorno `CD`, que solamente se puede usar en modo matemático:

```
\begin{CD}
    .....
\end{CD}
```

Los diferentes nodos o vértices del diagrama se escriben como fórmulas matemáticas, en la forma usual, y los renglones o niveles se separan con `\backslash\backslash`. Los comandos para flechas se presentan en la siguiente tabla.

<code>@>></code>	Flecha horizontal que apunta hacia la derecha.
<code>@<<</code>	Flecha horizontal que apunta hacia la izquierda.
<code>@AAA</code>	Flecha vertical que apunta hacia arriba.
<code>@VVV</code>	Flecha vertical que apunta hacia abajo.
<code>@=</code>	Doble línea horizontal (sin cabezas en los extremos).
<code>@ </code>	Doble línea vertical (sin cabezas en los extremos).
<code>@.</code>	Flecha invisible.

Una etiqueta en la parte superior de una flecha horizontal se escribe entre el primero y el segundo de los símbolos `< ó >`. Para que la etiqueta aparezca debajo de la flecha, se escribe entre el segundo y el tercero de los símbolos `< ó >`.

Análogamente, una etiqueta a la izquierda de una flecha vertical se escribe entre el primero y el segundo de los símbolos `A ó V`. Para que la etiqueta aparezca a la derecha de la flecha, se escribe entre el segundo y el tercero de los símbolos `A ó V`.

Ejemplo

$$\begin{array}{ccc} F & \xrightarrow{h} & F' \\ i \uparrow & & \uparrow j \\ X & \xrightarrow{g} & Y \end{array}$$

```
\[
\begin{CD}
F @>\{h\}>> F' \\
@A\{i\}AA @AA\{j\}A \\
X @>\{g\}>> Y \\
\end{CD}
\]
```

Ejemplo

$$\begin{array}{ccc} X \times Y & \xlongequal{\quad} & X \times Y \\ \text{proyección} \downarrow & & \downarrow \text{proyección} \\ X & \longrightarrow & Y \end{array}$$

```
\[
\begin{CD}
X\times Y @= X\times Y \\
@V\{\text{proyección}\}VV \\
@V\{\text{proyección}\}V \\
X @>>> Y \\
\end{CD}
\]
```

Ejemplo En este diagrama se usan dos flechas invisibles, con `@.`, para hacer que el nodo P quede ubicado en el centro del diagrama y no en el extremo superior izquierdo.

$$\begin{array}{ccccc} & & P & & \\ & & \downarrow f & & \\ X & \xleftarrow{g} & Y & \xrightarrow{h} & Z \end{array}$$

```
\[
\begin{CD}
@. P \\
@. @V\{f\}V \\
@. X @<\{g\}< Y @>\{h\}> Z \\
\end{CD}
\]
```

Ejemplo

$$\begin{array}{ccccc} F & \xrightarrow{h} & F' & \longrightarrow & T/R \\ i \uparrow & & \downarrow j & & \\ X & \xrightarrow{g} & Y & \longrightarrow & X \otimes Y \\ i' \downarrow & & \uparrow j' & & \\ G & \xrightarrow{f} & G' & \longrightarrow & T/S \end{array}$$

```
\[
\begin{CD}
F @>\{h\}>> F' @>>> T/R \\
@A\{i\}AA @VV\{j\}V \\
X @>\{g\}>> Y @>>> X\otimes Y \\
@V\{i'\}VV @AA\{j'\}A \\
G @>\{f\}>> G' @>>> T/S \\
\end{CD}
\]
```

4.34. Diagramas conmutativos con pb-diagram



El paquete **pb-diagram** (versión 5.0) está incluido en el CD adjunto, en la carpeta `/Paquetes/pb-diagram/`. Para información sobre la instalación de paquetes nuevos, véase el Apéndice C. Una vez instalado, se accede a él escribiendo `\usepackage{pb-diagram}` en el preámbulo del documento.

El paquete **pb-diagram**⁷ posee el entorno **diagram**, que se puede usar únicamente en modo matemático, y que tiene el siguiente aspecto:

```
\begin{diagram}
  nodo flecha flecha ... nodo flecha flecha ... \\
  nodo flecha flecha ... nodo flecha flecha ... \\
  ..... \\
  nodo flecha flecha ... nodo flecha flecha ...
\end{diagram}
```

Los nodos forman una grilla y se pueden unir entre sí por cero, una, dos o más flechas. Cada **nodo** ocupa un punto de la grilla y cada **flecha** que se escriba después (pero antes del siguiente nodo) lo conectará con uno de los otros nodos. La grilla de nodos se construye por filas, las cuales se separan entre sí por `\\"`. Los nodos de la grilla que hayan de quedar vacíos en el diagrama final simplemente no se escriben. **pb-diagram** automáticamente diseña la geometría del diagrama solamente con los nodos y las flechas especificados. No tenemos nunca que medir distancias ni ángulos.

Nodos. Los nodos se presentan en la forma

`\node [k] {fórmula}`

donde **fórmula** es la expresión matemática que aparece (centrada) en el nodo correspondiente. El argumento opcional `[k]` indica a cuántas columnas de distancia se coloca el nodo, con respecto al nodo anterior, en cada fila. Por defecto, `k = 1`.

Flechas. Hay tres tipos de flechas para unir nodos:

`\arrow[extensión]{dirección, opciones}.` Flecha sin etiquetas.

`\arrow[extensión]{dirección,E,opciones}{etiqueta}.` Flecha con una etiqueta. Los posibles valores del parámetro `E` son:

⁷El paquete **pb-diagram**, escrito por Paul Burchard, tiene opciones para flechas, adicionales a las descritas en la presente sección, pero requieren la instalación previa de alguno de los paquetes XY-pic o LAMS-T_EX. Para los detalles pertinentes remitimos al lector a la documentación, contenida en el archivo `pb-manual.dvi`.

- t *etiqueta* en la parte superior de la flecha.
- b *etiqueta* en la parte inferior de la flecha.
- l *etiqueta* a la izquierda de la flecha (únicamente para flechas verticales).
- r *etiqueta* a la derecha de la flecha (únicamente para flechas verticales).

\arrow[*extensión*]{*dirección*,*E*,*opciones*}{*etiqueta1*}{*etiqueta2*}.

Flecha con dos etiquetas. Los posibles valores del parámetro *E* son:

- tb *etiqueta1* en la parte superior y *etiqueta2* en la parte inferior de la flecha.
- lr *etiqueta1* a la izquierda y *etiqueta2* a la derecha de la flecha (únicamente para flechas verticales).

Los posibles valores del parámetro *dirección* del comando \arrow son:

n	e	ne	se	nne	sse	ene	wnw
s	w	nw	sw	nnw	ssw	ese	ws

donde **n** representa ‘norte’, **s** representa ‘sur’, **e** representa ‘este’ y **w** ‘oeste’. Con las combinaciones dobles el número total de filas y columnas que se avanza en la grilla es 2, y con las triples el número total es 3.

El parámetro opcional *extensión* de \arrow es un entero ≥ 2 e indica por cuántas columnas o cuántas filas se extiende la flecha. Así por ejemplo, la flecha \arrow[2]{e} se extiende por 2 columnas y \arrow[2]{s} se extiende por 2 filas. Si se omite, el parámetro *extensión* toma el valor 1.

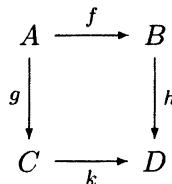
Las *opciones* para las flechas se muestran en la siguiente tabla.

..	Flecha punteada.
!	Flecha invisible.
-	Flecha sin cabeza ni cola.
<>	Flecha con cabeza en ambos extremos.
1	La etiqueta aparece a 1/4 de la distancia de la cola a la cabeza.
2	La etiqueta aparece a 2/4 de la distancia de la cola a la cabeza, es decir, en la mitad. Opción pre-determinada.
3	La etiqueta aparece a 3/4 de la distancia de la cola a la cabeza.

Si se quiere mayor flexibilidad para la posición de las etiquetas que la obtenida con los parámetros 1, 2 y 3 de la tabla anterior, se puede recurrir a la instrucción \dgARROWPARTS=*n*. Así, al declarar \dgARROWPARTS=6, antes de \begin{diagram}, los parámetros de posición permitidos son 1, 2, 3, 4 y 5. El valor *n* debe ser siempre par para permitir la colocación de las etiquetas en la mitad de las flechas. Por defecto, \dgARROWPARTS=4.

Hay que advertir que en el comando `\arrow` no puede haber espacios en blanco entre los parámetros opcionales *dirección*, *E*, *opciones*, y las comas que los separan.

Ejemplo

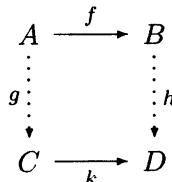


```

\[
\begin{diagram}
\node{A} \arrow{e,t}{f} \arrow{s,l}{g} \node{B} \arrow{s,r}{h} \\
\node{C} \arrow{e,b}{k} \node{D}
\end{diagram}
\]

```

Ejemplo Modificamos el diagrama del ejemplo anterior haciendo las flechas g y h punteadas.

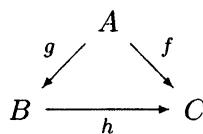


```

\[
\begin{diagram}
\node{A} \arrow{e,t}{f} \arrow{s,l,...}{g}
\node{B} \arrow{s,r,...}{h} \\
\node{C} \arrow{e,b}{k} \node{D}
\end{diagram}
\]

```

Ejemplo

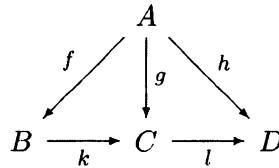


```
\[ \begin{diagram}
\node{} \node{A} \arrow{se,t}{f} \arrow{sw,t}{g} \\
\node{B} \arrow[2]{e,b}{h} \node{} \node{C}
\end{diagram} \]
```

También podemos obtener este diagrama usando el argumento opcional de `\node` para los nodos A y C , en lugar de escribir nodos vacíos:

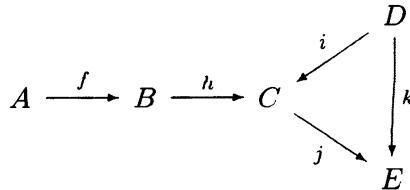
```
\[ \begin{array}{c} \begin{diagram} \node[2]{A} \arrow{se,t}{f} \arrow{sw,t}{g} \\ \node{B} \arrow[2]{e,b}{h} \node[2]{C} \end{diagram} \\ \end{array} \]
```

Ejemplo:



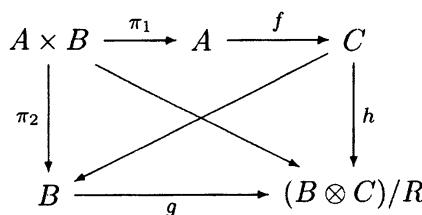
```
\[ \begin{array}{c} \begin{diagram} \node[2]{A} \arrow{sw,t}{f} \arrow{s,r}{g} \arrow{se,t}{h} \\ \node{B} \arrow{e,b}{k} \node{C} \arrow{e,b}{l} \node{D} \end{diagram} \\ \end{array} \]
```

Ejemplo:



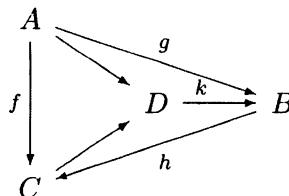
```
\[ \begin{array}{c} \begin{diagram} \node[4]{D} \arrow{sw,t}{i} \arrow[2]{s,r}{k} \\ \node{A} \arrow{e,t}{f} \node{B} \arrow{e,t}{h} \node{C} \arrow{se,b}{j} \\ \node[4]{E} \end{diagram} \\ \end{array} \]
```

Ejemplo:



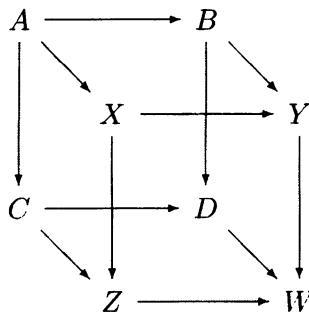
```
\[ \begin{array}{c} \begin{diagram} \node{A \times B} \arrow{e,t}{\pi_1} \arrow{s,l}{\pi_2} \\ \node{A} \arrow{ese} \node{A} \arrow{e,t}{f} \\ \node{C} \arrow{s,r}{h} \arrow{wsu} \\ \node{B} \arrow[2]{e,b}{g} \node{(B \otimes C)/R} \end{diagram} \\ \end{array} \]
```

Ejemplo En este diagrama se usa la opción de posición 1 para colocar la etiqueta k cerca del extremo izquierdo de la flecha.



```
\[
\begin{tikzpicture}
\node(A) \arrow[2]{s,1}{f} \arrow{se} \arrow{ese,t}{g} \\
\node{} \node(D) \arrow{e,t,1}{k} \node(B) \arrow{wsb,b}{h} \\
\node(C) \arrow{ne}
\end{tikzpicture}
\]
]
```

Ejemplo



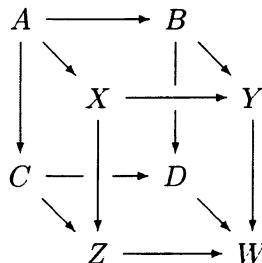
```
\[
\begin{tikzpicture}
\node(A) \arrow[2]{e} \arrow[2]{s} \arrow{se} \node(B) \\
\arrow[2]{s} \arrow{se} \\
\node(X) \arrow[2]{e} \arrow[2]{s} \node(Y) \arrow[2]{s} \\
\node(C) \arrow[2]{e} \arrow{se} \node(D) \arrow{se} \\
\node(Z) \arrow[2]{e} \node(W)
\end{tikzpicture}
\]
]
```

Para simular segmentos de flecha o flechas superpuestas, es conveniente reducir la longitud estándar de las flechas completas. El parámetro `\dgARROWLENGTH` controla la longitud de las flechas en un diagrama. Para reducir dicha longitud en un factor n utilizamos la instrucción

`\divide\dgARROWLENGTH by n`

la cual se debe escribir antes de `\begin{diagram}`. Su efecto es local: al aparecer el comando `\end{diagram}`, la longitud de las flechas retorna su valor normal. En el siguiente ejemplo se ilustra el procedimiento utilizado.

Ejemplo Este diagrama es una modificación del cubo de la página anterior. Se le ha dado un efecto tridimensional trazando las flechas BD y CD en dos tramos. Para ello, la extensión de las flechas se ha reducido a la mitad y se han utilizado nodos invisibles `\node{}`.

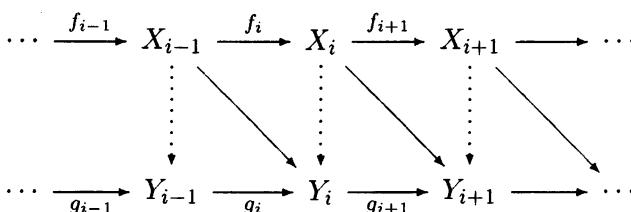


```

\[
\begin{array}{c}
\backslash [ \divide \dgARROWLENGTH by 2 \\
\begin{diagram}
\node{A} \arrow[2]{e} \arrow[2]{s} \arrow{se} \\
\node[2]{B} \arrow{s,-} \arrow{se} \\
\node[2]{X} \arrow[2]{e} \arrow[2]{s} \node{} \arrow{s} \\
\node{Y} \arrow[2]{s} \\
\node{C} \arrow{e,-} \arrow{se} \node{} \arrow{e} \\
\node{D} \arrow{se} \\
\node[2]{Z} \arrow[2]{e} \node[2]{W}
\end{diagram} \\
\end{array}
\]

```

Ejemplo En este diagrama cuatro de los nodos son `\cdots`.



```

\[
\begin{array}{ccccccc}
\cdots & \xrightarrow{f_{i-1}} & X_{i-1} & \xrightarrow{f_i} & X_i & \xrightarrow{f_{i+1}} & X_{i+1} \longrightarrow \cdots \\
& & \vdots & & \vdots & & \vdots \\
& & \searrow & & \searrow & & \searrow \\
\cdots & \xrightarrow{g_{i-1}} & Y_{i-1} & \xrightarrow{g_i} & Y_i & \xrightarrow{g_{i+1}} & Y_{i+1} \longrightarrow \cdots
\end{array}
\]

```

```

\[
\begin{array}{c}
\backslash [ \begin{diagram}
\node{\cdots} \arrow{e,t}{f_{i-1}} \node{X_{i-1}} \arrow{e,t}{f_i} \node{X_i} \arrow{e,t}{f_{i+1}} \node{X_{i+1}} \longrightarrow \cdots \\
\arrow{s,...} \arrow{se} \node{X_i} \arrow{e,t}{f_{i+1}} \\
\arrow{s,...} \arrow{se} \node{X_{i+1}} \arrow{e} \arrow{s,...} \\
\arrow{se} \node{\cdots} \\
\node{\cdots} \arrow{e,b}{g_{i-1}} \node{Y_{i-1}} \arrow{e,b}{g_i} \node{Y_i} \arrow{e,b}{g_{i+1}} \node{Y_{i+1}} \longrightarrow \cdots \\
\arrow{e,b}{g_{i+1}} \node{Y_{i+1}} \arrow{e} \\
\arrow{e}{\cdots} \\
\end{diagram} \\
\end{array}
\]

```

4.35. Los estilos **amsart** y **amsbook**

La *American Mathematical Society* ha diseñado dos estilos nuevos, **amsart** y **amsbook**, que son modificaciones de los estilos **article** y **book**, respectivamente. Fueron diseñados para los autores interesados en someter trabajos a la AMS, para su posible publicación, pero tienen características propias interesantes y por tal razón los describimos brevemente⁸. El usuario debe tener presente que los estilos **amsart** y **amsbook** están concebidos para documentos escritos en inglés. Incluso con el paquete **babel**, expresiones pre-definidas, como '*Key words and phrases*', '*Date*', '*Current address*', etc, se obtienen en inglés.

En la Tabla 4.7 se bosqueja la estructura de un documento escrito con el estilo **amsart**, que es el más útil de los dos. Se puede apreciar que en el preámbulo del documento aparecen opciones o comandos nuevos, no incluidos en el estilo **article** de L^AT_EX (sección 2.5).

Para utilizar el estilo **amsart** en un documento determinado se escribe la declaración `\documentclass{amsart}`. Al estilo **amsbook** se accede de manera similar. Ambos estilos cargan automáticamente los paquetes **amsmath** y **amsthm**, por lo que no es necesario cargarlos separadamente. Pero los paquetes **amssymb** y **amscd** sí se deben cargar explícitamente.

A parte de las opciones `[10pt]`, `[11pt]` y `[12pt]`, se ofrecen dos tamaños adicionales: `[8pt]` y `[9pt]`. Esto es útil cuando se desean documentos en letra muy pequeña.

La gama de los tamaños de letra es la siguiente (compárense con los comandos de L^AT_EX estándar, sección 3.4):

```
\tiny \tiny \small \small
      \normalsize
\large \Large \LARGE \huge \Huge
```

Aparece un tamaño más (`\Tiny`) mientras que `\small` y `\small` significan lo mismo que `\scriptsize` y `\footnotesize`, respectivamente.

Existen, además, las instrucciones `\larger` y `\smaller` para cambiar tamaños con relación al tamaño de letra normal (`\normalsize`). Estos comandos pueden tomar un argumento; `{\larger[2] ... }`, por ejemplo, produce texto de tamaño 2 veces mayor al tamaño normal y la instrucción `{\larger[1] ... }` es equivalente a `{\larger ... }`.

⁸Para información adicional, el lector puede procesar los archivos L^AT_EX **instr-I.tex** y **amsclass.dtx**, incluidos en las distribuciones de **amsmath**.

<code>\documentclass{amsart}</code>	
<code>\title[...]{...}</code>	En el argumento principal <code>{...}</code> se puede usar <code>\\\</code> para forzar separaciones de renglones. El argumento opcional <code>[...]</code> se usa para el título abreviado en los encabezados.
<code>\author[...]{...}</code>	En el argumento principal <code>{...}</code> , los nombres de dos o más autores se separan con <code>\\"</code> . También se pueden incluir en este campo las afiliaciones de los autores. El argumento opcional <code>[...]</code> se usa para los encabezados.
<code>\address{...}</code>	Aparece al final del documento. Opcional.
<code>\curraddr{...}</code>	Aparece al final del documento. Opcional.
<code>\email{...}</code>	Se pueden incluir varias direcciones, con sendos <code>\email{...}</code> . Aparecen al final del documento.
<code>\urladdr{...}</code>	Dirección URL, o página <i>Web</i> . Aparece al final del documento. Opcional.
<code>\dedicator{...}</code>	Aparece debajo de los autores, en letra cursiva (ítálica). Se puede usar <code>\\\</code> para forzar la separación de renglones. Opcional.
<code>\date{...}</code>	Aparece en la parte inferior de la primera página. Opcional.
<code>\thanks{...}</code>	Agradecimientos. Aparece en la parte inferior de la primera página. Se pueden usar varios <code>\thanks</code> en el preámbulo. Opcional.
<code>\translator{...}</code>	Traductor. Aparece al final del documento. Opcional.
<code>\keywords{...}</code>	Términos o palabras clave. Aparece en la parte inferior de la primera página. Opcional.
<code>\subjclass[2000]{...}</code>	Clasificación AMS; si no se escribe el argumento opcional <code>[2000]</code> , se citará la clasificación de 1991. Aparece en la parte inferior de la primera página. Opcional.
<code>\begin{document}</code>	
<code>\begin{abstract}</code>	Se debe escribir antes de <code>\maketitle</code> .
.....	
<code>\end{abstract}</code>	
<code>\maketitle</code>	
.....	
<code>\end{document}</code>	

TABLA 4.7. Estructura de un documento escrito con el estilo `amsart`.

El estilo **amsart** está diseñado para impresión a dos caras. En las páginas de numeración par (las páginas a mano izquierda) los encabezados contienen los nombres de los autores, determinados por `\author{...}`, y en las páginas de numeración impar (las páginas a mano derecha) los encabezados contienen el título del artículo, determinado por `\title{...}`. Esto es muy útil porque el usuario no tiene que preocuparse por escoger un formato de página con `\pagestyle` (sección 2.10).

Si el título del artículo o los nombres de los autores son muy extensos, se pueden indicar abreviaciones de ellos para los encabezados de las páginas, en la forma:

```
\title[título abreviado]{título completo}  
\author[nombres abreviados]{nombres completos}
```

Recalcamos que estos comandos opcionales para `\title` y `\author` no existen en los estilos básicos `article` y `book` de L^AT_EX.



Tablas

TEX tiene un entorno muy cómodo para la creación de tablas, el entorno **tabular**, cuya sintaxis es:

```
\begin{tabular}{formato}
... & ... & ...
... & ... & ...
:
...
\end{tabular}
```

El *formato* contiene información sobre el número de columnas, con su respectiva justificación: **l** (a la izquierda), **c** (centrada), **r** (a la derecha) y **p{...}** (columna con párrafos, véase la sección 5.2). El contenido de las diferentes columnas se separa con **&** y las filas se separan con **\\"**. El número de símbolos **&** debe ser el mismo en cada fila, incluso si una o más casillas de la tabla tienen contenido vacío. En la última fila de la tabla no es necesario usar **\\"**, aunque hacerlo no conduce a ningún mensaje de error.

Para tablas con ancho pre-determinado existe el entorno

```
\begin{tabular*}{ancho}{formato}
:
\end{tabular*}
```

siendo el *ancho* una dimensión TEX.

Los entornos **tabular** y **tabular*** crean *cajas* (en el sentido TEX-nico, sección 3.15) y tienen un parámetro que permite controlar la posición de la tabla con respecto al material vertical que la rodea. Esto último se explica en la sección 5.13.

Ejemplo La siguiente tabla tiene formato **{cccc}**, es decir, cuatro columnas centradas. Algunas de las casillas están vacías pero el número de **&** por fila debe ser siempre cuatro.

Primera	Segunda	Tercera	Cuarta
A	B		C
	D	E	F
G			H

```
\begin{tabular}{cccc}
Primera & Segunda & Tercera & Cuarta\\
A & B && C\\
& D & E & F\\
G &&& H
\end{tabular}
```

Ejemplo La siguiente tabla tiene el formato `{llc}`: las dos primeras columnas justificadas a la izquierda y la tercera centrada.

Partícula:	Descubridor:	Año del descubrimiento:
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{tabular}{llc}
Partícula: & Descubridor: & Año del descubrimiento:\\
Electrón & Joseph J. Thomson & 1897\\
Protón & James Rutherford & 1919\\
Neutrón & James Chadwick & 1932\\
Positrón & Carl D. Anderson & 1932
\end{tabular}
```

Ejemplo Tabla del ejemplo anterior centrada. Para centrar una tabla, basta usar el entorno `center` (sección 3.13), colocando sus comandos entre `\begin{center}` y `\end{center}`.

Partícula:	Descubridor:	Año del descubrimiento:
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center}
\begin{tabular}{llc}
Partícula: & Descubridor: & Año del descubrimiento:\\
Electrón & Joseph J. Thomson & 1897\\
Protón & James Rutherford & 1919\\
Neutrón & James Chadwick & 1932\\
Positrón & Carl D. Anderson & 1932
\end{tabular}
\end{center}
```

5.1. Tablas con líneas

Se usa | en el formato de la tabla cuando se desea una línea vertical en una columna determinada. Las líneas horizontales en la tabla se obtienen con \hline, que se coloca después de \\ o antes de la primera fila. Para líneas verticales dobles se usa || en el formato de la tabla, y para líneas horizontales dobles se usa \hline\hline después de \\.

Ejemplo Tabla del ejemplo anterior con líneas horizontales y verticales. El formato es {|l|l|c|}, el cual indica líneas verticales alrededor de cada una de las columnas. Las líneas horizontales se obtienen con \hline.

Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center}
\begin{tabular}{|l|l|c|}\hline
Partícula & Descubridor & Año del descubrimiento\\ \hline
Electrón & Joseph J. Thomson & 1897\\ \hline
Protón & James Rutherford & 1919\\ \hline
Neutrón & James Chadwick & 1932\\ \hline
Positrón & Carl D. Anderson & 1932\\ \hline
\end{tabular}
\end{center}
```

Ejemplo La siguiente tabla coincide con la anterior, excepto por las líneas dobles en la primera fila y en la primera columna.

Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center}
\begin{tabular}{||l||l|c||}\hline\hline
Partícula & Descubridor & Año del descubrimiento\\ \hline\hline
Electrón & Joseph J. Thomson & 1897\\ \hline
Protón & James Rutherford & 1919\\ \hline
Neutrón & James Chadwick & 1932\\ \hline
Positrón & Carl D. Anderson & 1932\\ \hline
\end{tabular}
\end{center}
```

5.2. Tablas con párrafos

Cualquier columna de una tabla puede contener un párrafo; para esto hay que indicar en el formato de la tabla el ancho de tal columna, en la forma `p{ancho}`. Este tipo de formato se usa también para diseñar tablas con columnas de ancho pre-determinado.

Ejemplo El formato de la siguiente tabla es `{|l|c|p{7cm}|}`, según el cual la primera columna está justificada a la izquierda, la segunda está centrada y la tercera es un párrafo de 7 cm de ancho. La tabla tiene además líneas horizontales y verticales. Obsérvese que el texto de la primera fila está escrito usando letra de tipo sans serif, con `\textsf{...}`.

Científico	Fecha	Logro
John Dalton	1803–08	Propone la moderna teoría atómica y revive la palabra “átomo”, acuñada por el filósofo griego Demócrito.
Jakob Berzelius	1828	Publica la primera tabla de pesos atómicos, cuyos valores coinciden con los actuales, excepto para tres elementos.
D. Mendeleiev	1869	Publica su <i>Tabla Periódica de Elementos</i> , diseñada según la noción de valencia.
Henry Moseley	1913	Añade claridad a la tabla periódica al introducir el concepto de número atómico.

```
\begin{center}
\begin{tabular}{|l|c|p{7cm}|}\hline
\textsf{Científico} & \textsf{Fecha} & \textsf{Logro}\\\hline
John Dalton & 1803--08 & Propone la moderna teoría atómica y  
revive la palabra “átomo”, acuñada por el filósofo griego  
Demócrito.\\\hline
Jakob Berzelius & 1828 & Publica la primera tabla de pesos  
atómicos, cuyos valores coinciden con los actuales, excepto  
para tres elementos.\\\hline
D. Mendeleiev & 1869 & Publica su \textit{Tabla Periódica de  
Elementos}, diseñada según la noción de valencia.\\\hline
Henry Moseley & 1913 & Añade claridad a la tabla periódica al  
introducir el concepto de número atómico.\\\hline
\end{tabular}
\end{center}
```

5.3. Tablas con filas especiales

Cuando una fila determinada de una tabla se aparta del formato declarado, hay que usar el comando `\multicolumn`, el cual tiene tres argumentos obligatorios:

```
\multicolumn{n}{justificación}{Contenido de las n columnas}
```

n es el número de columnas abarcadas y *justificación* es uno de los tres parámetros *l*, *c*, *r* (con o sin la raya vertical *|*). La instrucción `\multicolumn` también se usa para cambiar el formato (justificación, raya vertical) de una columna en una fila particular, escribiendo su contenido en la forma

```
\multicolumn{1}{justificación}{...}
```

Ejemplo El encabezado de la siguiente tabla se extiende las tres columnas de la tabla y está centrado, lo que corresponde a la instrucción `\multicolumn{3}{|c|}{...}`. La última fila de la tabla también se aparta del formato general; para ésta se usó `\multicolumn{2}{l}{...}` ya que el texto de la fila abarca 2 columnas.

PARTICULAS ATOMICAS ELEMENTALES		
Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson*	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick*	1932
Positrón	Carl D. Anderson*	1932

*Recibió el premio Nobel

```
\begin{center}
\begin{tabular}{|l|l|c|}\hline
\multicolumn{3}{|c|}{PARTICULAS ATOMICAS
ELEMENTALES}\\\hline\hline
\textsf{Partícula} & \textsf{Descubridor} & \textsf{Año del
descubrimiento}\\\hline
Electrón & Joseph J. Thomson* & 1897\\\hline
Protón & James Rutherford & 1919\\\hline
Neutrón & James Chadwick* & 1932\\\hline
Positrón & Carl D. Anderson* & 1932\\\hline
\multicolumn{2}{l}{\small *Recibió el premio Nobel}
\end{tabular}
\end{center}
```

5.4. Simplificación del formato de una tabla

LATEX admite la siguiente simplificación en el formato de una tabla:

`*{n}{columnas}`

equivale a n copias de la especificación `columnas`, siendo esta última una lista cualquiera de los parámetros `l, r, c, p{...}` y `|`.

Ejemplos

- El formato `{cccc}` se puede escribir como `*{4}{c}`.
- El formato `*{3}{|r|c|}` equivale a `{|r|c|r|c|r|c|}`.
- El formato `{r*{3}{p{4cm}}r}` equivale a `{rp{4cm}p{4cm}p{4cm}r}`.

5.5. Líneas horizontales con \cline

La instrucción `\cline{n-m}` traza una línea horizontal desde la columna n hasta la columna m ; se usa al finalizar cada fila, después de `\backslash\backslash`.

Ejemplo

Uno	Dos	Tres	Cuatro
		A C	B
	D		
	E		
1	2	3	4

```
\begin{center}
\begin{tabular}{|c|c|c|c|}\hline
Uno & Dos & Tres & Cuatro\\ \hline
&&A&B \\ \cline{3-4}
&&C& \\ \cline{1-3}
&D&& \\ \cline{2-2}
&E&& \\ \hline
1&2&3&4 \\ \hline
\end{tabular}
\end{center}
```

5.6. Líneas verticales con \vline

Para producir líneas verticales, adicionales a las especificadas con | en el formato de la tabla, se usa el comando `\vline`. La línea vertical tiene la altura total de la casilla y aparece centrada horizontalmente, aunque se puede insertar espacio horizontal, antes o después de `\vline`, con comandos como `_, \quad` o `\hspace{...}`.

Ejemplo

```
\begin{tabular}{|c|c|c|}\hline
Uno & Dos & Tres \\ \hline
\hspace{3mm}\vline A&B&\vline \cline{1-2}
C&D& \\ \hline
&&\vline
&&\vline \\ \hline
\end{tabular}
```

Uno	Dos	Tres
A	B	
C	D	

5.7. Espacio horizontal adicional

LATEX controla el espaciamiento entre las columnas de una tabla con el parámetro `\tabcolsep`; su valor por defecto es 6 pt (poco más de 2 mm). Esta distancia se añade a la izquierda y a la derecha de cada columna, incluyendo la primera y la última. Se puede aumentar o reducir el valor de este parámetro usando `\renewcommand`.

Ejemplo En la siguiente tabla se ha extendido uniformemente el espacio entre las columnas con la instrucción `\renewcommand{\tabcolsep}{0.6cm}`.

Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center} \renewcommand{\tabcolsep}{0.6cm}
\begin{tabular}{|l|l|c|}\hline
\textsf{Partícula} & \textsf{Descubridor} & \textsf{Año}\\
& & del descubrimiento\\ \hline
Electrón & Joseph J. Thomson & 1897\\
Protón & James Rutherford & 1919\\
Neutrón & James Chadwick & 1932\\
Positrón & Carl D. Anderson & 1932\\ \hline
\end{tabular} \end{center}
```

5.8. Espacio vertical adicional

LA**T**E**X** controla el espaciamiento vertical entre las filas de una tabla con el parámetro `\arraystretch`, el cual no es una dimensión sino el valor por el cual se multiplica la separación *normal* entre las filas. El usuario puede aumentar o reducir el valor de este parámetro utilizando la instrucción `\renewcommand`.

Ejemplo La siguiente tabla es una modificación de la tabla que aparece en la página 171; se ha incrementado el espacio entre renglones en un 30 % por medio de la instrucción `\renewcommand{\arraystretch}{1.3}`. Puesto que dicha instrucción aparece bajo el alcance del entorno `center`, el cambio de espaciamiento solamente afecta esta tabla particular; colocada antes de `\begin{center}`, la instrucción afectará todas las tablas sucesivas.

PARTICULAS ATOMICAS ELEMENTALES		
Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center}
\renewcommand{\arraystretch}{1.3}
\begin{tabular}{|l|l|c|}\hline
\multicolumn{3}{|c|}{\vphantom{\LARGE A} \textsf{PARTICULAS ATOMICAS ELEMENTALES}}\\ \hline\hline
\textsf{Partícula} & \textsf{Descubridor} & \\
\textsf{Año del descubrimiento}\\\hline
Electrón & Joseph J. Thomson & 1897\\\hline
Protón & James Rutherford & 1919\\\hline
Neutrón & James Chadwick & 1932\\\hline
Positrón & Carl D. Anderson & 1932\\\hline
\end{tabular}
\end{center}
```

Algunas veces se desea añadir espacio vertical adicional en determinadas filas y no en todas; por ejemplo, en el encabezado o título de la tabla. La manera más práctica de hacerlo es insertar una caja invisible de altura adecuada y ancho nulo, como la que se obtiene con la instrucción `\rule[...]{0pt}{...}` (sección 3.22), o con el comando **T**E**X** primario `\vphantom{...}`, el cual crea una caja invisible de ancho nulo (véase la

sección 3.15.5). Así, si usamos algo como `\vphantom{\LARGE Ap}` lo que obtenemos es una caja invisible, de ancho nulo, pero con altura suficiente para contener la expresión ‘Ap’, en el tamaño de letra `\LARGE`. Esto se ilustra en el ejemplo siguiente.

Ejemplo La siguiente tabla coincide con la de la página 171, excepto que se ha añadido espacio vertical a la fila que contiene el título, por medio de la caja invisible `\vphantom{\LARGE Ap}`.

PARTICULAS ATOMICAS ELEMENTALES		
Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center}
\begin{tabular}{|l|l|c|}\hline
\multicolumn{3}{|c|}{\vphantom{\LARGE Ap} PARTICULAS ATOMICAS ELEMENTALES}\hline\hline
\textsf{Particula} & \textsf{Descubridor} & \textsf{Año del descubrimiento}\hline
Electrón & Joseph J. Thomson & 1897\hline
Protón & James Rutherford & 1919\hline
Neutrón & James Chadwick & 1932\hline
Positrón & Carl D. Anderson & 1932\hline
\end{tabular}
\end{center}
```

El argumento opcional [*longitud*] de `\hline` (mencionado en la sección 3.9) también se puede usar en tablas para añadir espacio vertical entre filas. Al escribir `\hline[longitud]`, L^AT_EX añade un espacio vertical de *longitud* dada antes de comenzar la siguiente fila de la tabla.

Ejemplo En esta tabla de gentilicios se ha usado `\hline[2mm]` para añadir un espacio de 2 mm entre las filas.

Asunción	<i>asunceno</i>
Bruselas	<i>bruselense</i>
El Cairo	<i>cairota</i>
Damasco	<i>damasceno</i>
Lisboa	<i>lisboeta</i>

```
\begin{tabular}{|l|}\hline
Asunción & {\it asunceno}\hline[2mm]
Bruselas & {\it bruselense}\hline[2mm]
El Cairo & {\it cairota}\hline[2mm]
Damasco & {\it damasceno}\hline[2mm]
Lisboa & {\it lisboeta} \hline
\end{tabular}
```

5.9. Grosor de las líneas de una tabla

El grosor de las líneas horizontales y verticales de las tablas (incluyendo las líneas que se obtienen con `\hline`, `\cline` y `\vline`) se controla con el parámetro `\arrayrulewidth`. Por defecto, todas las líneas tienen un grosor de 0.4 pt. Dicho valor se puede cambiar por medio de `\renewcommand`.

Ejemplo Esta tabla es similar a la de la página 174 pero tiene líneas de grosor 1 pt; tal grosor se obtiene con `\renewcommand{\arrayrulewidth}{1pt}`. Puesto que dicha instrucción aparece bajo el alcance del entorno `center`, el cambio solamente afecta esta tabla particular; colocada antes del comando `\begin{center}`, la instrucción afectará todas las tablas sucesivas, hasta que sea contrarrestada con una instrucción del mismo tipo. En esta tabla también se ha ampliado el espaciamiento entre filas por medio de la instrucción `\renewcommand{\arraystretch}{1.4}`.

PARTICULAS ATOMICAS ELEMENTALES		
Partícula	Descubridor	Año del descubrimiento
Electrón	Joseph J. Thomson	1897
Protón	James Rutherford	1919
Neutrón	James Chadwick	1932
Positrón	Carl D. Anderson	1932

```
\begin{center}
\renewcommand{\arraystretch}{1.4}
\renewcommand{\arrayrulewidth}{1pt}
\begin{tabular}{|l|l|c|}\hline
\multicolumn{3}{|c|}{\vphantom{\LARGE A}p} PARTICULAS ATOMICAS
ELEMENTALES}\\ \hline\hline
\textsf{Particula} & \textsf{Descubridor} & \textsf{Año
del descubrimiento}\\ \hline
Electrón & Joseph J. Thomson & 1897\\ \hline
Protón & James Rutherford & 1919\\ \hline
Neutrón & James Chadwick & 1932\\ \hline
Positrón & Carl D. Anderson & 1932\\ \hline
\end{tabular}
\end{center}
```

5.10. @-expresiones

Para añadir *algo* (por ejemplo, texto o espacio horizontal) entre columnas particulares se usa el modificador `@{algo}` en el formato de la tabla. La expresión *algo* aparece insertada en todas las filas. Cuando `@{...}` aparece en medio de dos columnas particulares, se elimina el espacio adicional que L^AT_EX inserta entre éstas, según el valor de `\tabcolsep` (sección 5.7).

Ejemplo Antes de la primera columna de la siguiente tabla se ha añadido el símbolo `•` seguido de un espacio `\quad`, por medio del modificador `@{\bullet\quad}`. Entre la primera y segunda columnas y entre la segunda y tercera columnas se han añadido 2 cm de espacio horizontal con el modificador `@{\hspace{2cm}}`. Obsérvese que el número de columnas de esta tabla sigue siendo 3, con 2 símbolos `&` por fila.

• Electrón	Joseph J. Thomson	1897
• Protón	James Rutherford	1919
• Neutrón	James Chadwick	1932
• Positrón	Carl D. Anderson	1932

```
\begin{tabular}{@{$\bullet$\quad}l@{\hspace{2cm}}l@{\hspace{2cm}}c}
Electrón & Joseph J. Thomson & 1897\\
Protón & James Rutherford & 1919\\
Neutrón & James Chadwick & 1932\\
Positrón & Carl D. Anderson & 1932
\end{tabular}
```

5.11. Inserción y numeración de tablas

Al insertar una tabla (que haya sido definida con el entorno `tabular`) en un sitio determinado de un documento, es muy posible que no haya espacio suficiente para acomodarla en lo que queda de la página. En tales casos L^AT_EX deja el resto de la página en blanco y coloca la tabla en la página siguiente. El resultado es totalmente inaceptable, en la mayoría de los casos. Para obviar tales situaciones hay que insertar las tablas como *objetos flotantes* (en inglés, *floating objects* o *floats*) utilizando el entorno `table`. L^AT_EX decide cuál es el sitio más adecuado para colocar la tabla, ya sea en la página actual o en una posterior, y acomoda el texto circundante sin añadir o eliminar espacio en blanco. El manejo de objetos flotantes se explica en la sección 7.4.

El uso del entorno `table` permite también la inclusión de un rótulo o epígrafe para la tabla y la numeración automática de las tablas; esto se explica en la sección 7.5.

5.12. Tablas con el paquete `array`

El paquete `array`¹ incrementa las capacidades de edición del entorno básico `tabular` de L^AT_EX. Sus características más importantes son:

1. Además del parámetro `p{ancho}` para columnas con párrafos, el paquete `array` tiene el parámetro `m{ancho}`, con el cual el párrafo aparece verticalmente centrado respecto de las columnas vecinas (en vez de estar alineado por la parte superior). También se dispone del parámetro `b{ancho}` para que el párrafo aparezca alineado por la parte inferior.
2. Cada uno de los parámetros de columna `l`, `r`, `c`, `p{...}`, `m{...}`, `b{...}` puede ir precedido de `>{algo}`, y su efecto es insertar `algo` antes del contenido de la fila, en la columna correspondiente. La parte '`algo`' puede incluir otros comandos L^AT_EX (por ejemplo, comandos para cambio de tipo o tamaño de letra). Similarmente, cada uno de los parámetros de columna puede ir seguido de `<{algo}`.
3. El usuario puede definir nuevos tipos de columna para el formato de la tabla, por medio de

`\newcolumntype{nombre}{definición}`

El tipo de columna denominado `nombre` representa los parámetros dados en la `definición`. El `nombre` debe ser una sola letra.

4. Con el paquete `array`, las líneas muy gruesas no tocarán el texto contenido en las casillas, cosa que sí puede ocurrir en la versión estándar de `tabular`.
5. Con el paquete `array` obtenemos mejores resultados al alinear tablas con texto o material circundante. Véase al respecto la sección 5.13.

En los ejemplos que siguen ilustraremos el uso de estas facetas del paquete `array`, al cual se accede con `\usepackage{array}`.

Ejemplo La siguiente tabla se diferencia de la tabla de la sección 5.2 en que el contenido de las dos primeras columnas aparece verticalmente centrado con respecto al párrafo de la tercera columna. Esto se logra escribiendo la tercera columna con `m{7cm}` en vez de `p{7cm}`. Además, el encabezado 'Logro' de la tercera columna aparece centrado (a diferencia de lo que sucede en la citada tabla) ya que se ha usado `\multicolumn{1}{c}{...}` para dicha casilla.

¹Paquete estándar de L^AT_EX 2_E, escrito por Frank Mittelbach. No debe confundirse el paquete `array` con el entorno `array` de L^AT_EX (sección 4.23).

Científico	Fecha	Logro
John Dalton	1803-08	Propone la moderna teoría atómica y revive la palabra “átomo”, acuñada por el filósofo griego Demócrito.
Jakob Berzelius	1828	Publica la primera tabla de pesos atómicos, cuyos valores coinciden con los actuales, excepto para tres elementos.
D. Mendeleiev	1869	Publica su <i>Tabla Periódica de Elementos</i> , diseñada según la noción de valencia.
Henry Moseley	1913	Añade claridad a la tabla periódica al introducir el concepto de número atómico.

```
\begin{center}
\begin{tabular}{|l|c|m{7cm}|}\hline
\textsf{Científico} & \textsf{Fecha} &
\multicolumn{1}{c}{\textsf{Logro}} \\ \hline
John Dalton & 1803-08 & Propone la moderna ... \\ \hline
Jakob Berzelius & 1828 & Publica la primera tabla de ... \\ \hline
D. Mendeleiev & 1869 & Publica su ... \\ \hline
Henry Moseley & 1913 & Añade claridad a la tabla ... \\ \hline
\end{tabular}
\end{center}
```

Ejemplo Las dos primeras columnas de la siguiente tabla son fórmulas. Usando el entorno `tabular` estándar de LATEX, hay que escribir `$...$` en cada una de las casillas, pero con el paquete `array` sólo tenemos que añadir la declaración `>{$}c<{$}` en el formato de las dos primeras columnas. La `c` representa ‘centrado’; si quisieramos que las entradas aparecieran justificadas a la derecha escribiríamos `>{$}r<{$}`.

$\Delta > 0$	$\Delta > 0$	Mínimo local en \vec{a}
$\Delta > 0$	$\Delta < 0$	Máximo local en \vec{a}
$\Delta < 0$	—	Punto de silla en \vec{a}
$\Delta = 0$	—	Indecidable

```
\begin{center}
\begin{tabular}{|>{$}c<{$}|>{$}c<{$}|>{$}c<{$}|l|}\hline
\Delta > 0 & \Delta > 0 & Mínimo local en  $\vec{a}$  \\ \hline
\Delta > 0 & \Delta < 0 & Máximo local en  $\vec{a}$  \\ \hline
\Delta < 0 & - & Punto de silla en  $\vec{a}$  \\ \hline
\Delta = 0 & - & Indecidable \\ \hline
\end{tabular}
\end{center}
```

De hecho, si en nuestras tablas aparecen muchas columnas cuyo contenido está en modo matemático, como en la tabla anterior, podemos definir una nueva opción de columna y llamarla, por ejemplo, `M` (para distinguirla de la opción `m{...}`). Esto se hace con la instrucción `\newcolumntype`, en la forma:

```
\newcolumntype{M}{>{$}c<{$}}
```

Utilizando esta definición, la tabla del ejemplo anterior se puede obtener con el elegante formato `{|M|M|1|}`; concretamente:

```
\begin{center}
\begin{tabular}{|M|M|1|}\hline
\Delta > 0 & \Delta>0 & Mínimo local en $\vec{a}$\\\hline
\Delta > 0 & \Delta<0 & Máximo local en $\vec{a}$\\\hline
\Delta < 0 & - & Punto de silla en $\vec{a}$\\\hline
\Delta = 0 & - & Indecidible\\\hline
\end{tabular}
\end{center}
```

Ejemplo En la tabla que aparece a continuación, las entradas de la primera columna aparecen en itálicas y las de la segunda columna en versalitas. Con el entorno `tabular` estándar, la única manera de lograrlo es escribiendo la instrucción para cambio de tipo de letra en *cada una* de las casillas de la tabla (8 instrucciones en total); el uso de @-expresiones (sección 5.10) tampoco sirve en estos casos. Usando el paquete `array`, añadimos en el formato de la tabla la declaración `>{\it}` para la primera columna, y `>{\sc}` para la segunda.

Descubridores de las partículas elementales	
<i>Electrón</i>	JOSEPH J. THOMSON
<i>Protón</i>	JAMES RUTHERFORD
<i>Neutrón</i>	JAMES CHADWICK
<i>Positrón</i>	CARL D. ANDERSON

```
\begin{center}
\begin{tabular}{|>{\it}l|>{\sc}c|}\hline
\multicolumn{2}{|c|}{Descubridores de las partículas elementales}\\\hline\hline
Electrón & Joseph J. Thomson \\\hline
Protón & James Rutherford \\\hline
Neutrón & James Chadwick \\\hline
Positrón & Carl D. Anderson \\\hline
\end{tabular}
\end{center}
```

5.13. Tablas y texto circundante

\LaTeX coloca una tabla, creada con el entorno `tabular`, verticalmente centrada con respecto al material que la rodea. Pero los entornos `tabular` y `tabular*` tienen dos opciones de posición adicionales: `[t]` y `[b]`, que se usan en la forma

```
\begin{tabular}[posición]{formato}
```

Con la opción `[t]`, la tabla aparece alineada por la parte superior, con respecto al material horizontal circundante; con la opción `[b]` la alineación se hace por la parte inferior. Con `[c]`, la tabla aparece verticalmente centrada; ésta es la posición asumida por defecto.

Ejemplo El siguiente texto de entrada

```
Frase a la izquierda de la tabla \quad
\begin{tabular}{|c|c|c|}\hline
Uno & Dos & Tres \\ \hline
&&\hline
&&\hline
&&\hline
\end{tabular}
```

produce lo siguiente:

Frase a la izquierda de la tabla

Uno	Dos	Tres

Con la opción de posición `[t]`, `\begin{tabular}[t]{|c|c|c|}`, se obtiene:

Frase a la izquierda de la tabla

Uno	Dos	Tres

Con la opción de posición `[b]`, `\begin{tabular}[b]{|c|c|c|}`, se obtiene:

Uno	Dos	Tres

Frase a la izquierda de la tabla

Si se observa detenidamente el ejemplo anterior, se notará que al usar la opción de posición [t], la alineación de la tabla con respecto al texto que la precede no se hace en realidad por el primer renglón de la tabla, sino por la primera línea horizontal. Esta anomalía se presenta siempre que la tabla se inicie con \hline. Usando el paquete `array`, descrito en la sección anterior, se corrige la alineación: basta escribir `\firsthline` en vez del primer `\hline` y `\lasthline` en lugar del último `\hline`, tal como se muestra a continuación.

Frase a la izquierda de la tabla

Uno	Dos	Tres

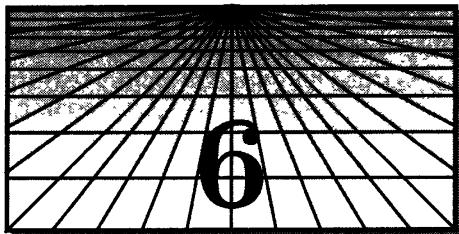
```
Frase a la izquierda de la tabla \quad
\begin{tabular}[t]{|c|c|c|}\firsthline
Uno & Dos & Tres \\ \hline
&&\hline
&&\hline
&&\hline
\end{tabular}
```

5.14. Tablas extensas, tablas a color y rotación de tablas

Tablas extensas. L^AT_EX es incapaz de dividir una tabla en dos páginas porque el entorno `tabular` produce cajas indivisibles. Para solucionar este problema se puede usar el paquete `longtable`, descrito en la sección 13.6.

Color en tablas. Hay varios paquetes diseñados para colorear filas, columnas o casillas de una tabla, entre los que se destaca `colortbl`. Puesto que dicho paquete requiere, a su vez, del paquete `color` (sección 7.2), se describirá detalladamente en la sección 13.5.

Rotación de tablas. Tablas muy anchas generalmente se rotan 90° para ser incluidas en documentos normales. Para rotar tablas, gráficas o cajas en general, L^AT_EX posee el comando `\rotatebox`. Y para manejar más eficientemente rotaciones de material muy extenso, podemos recurrir al paquete `lscape`. Ambos mecanismos de rotación hacen uso de alguno de los paquetes `graphics` o `graphicx` y por tal razón se describirán más adelante. Sobre `\rotatebox` véase la sección 7.3.3, y para una descripción del paquete `lscape` remitimos al lector a la sección 13.9.



Manejo de documentos grandes

6.1. Partes o divisiones en un documento L^AT_EX

Las unidades o divisiones permitidas en documentos L^AT_EX son:

```
\part[título abreviado]{título completo}  
\chapter[título abreviado]{título completo}  
\section[título abreviado]{título completo}  
\subsection[título abreviado]{título completo}  
\subsubsection[título abreviado]{título completo}  
\paragraph[título abreviado]{título completo}  
\ subparagraph[título abreviado]{título completo}
```

Cada una de estas unidades es una subdivisión de la unidad de nivel anterior, pero `\part` es opcional. El argumento *título abreviado* es opcional y se utiliza para los encabezados y el índice general (véase al respecto la sección 6.6). En el estilo `article` el comando `\chapter` no está definido. En los estilos `report` y `book`, los comandos `\part` producen páginas separadas con el correspondiente título y con rótulos de la forma ‘Part I’, ‘Part II’, ‘Part III’, etc (o ‘Parte’, si se usa `babel, spanish`; véase la sección 6.5).

Los títulos de las unidades creadas se obtienen siempre en letra negrilla. Las divisiones `\paragraph` y `\ subparagraph` sirven para producir títulos, del tamaño de los títulos de las subsecciones pero sin numeración. Dichos títulos son incluidos automáticamente en la tabla de contenido (sección 6.6).

En los estilos `report` y `book`, además de las subdivisiones anteriores, L^AT_EX distingue (de manera opcional) tres amplias divisiones; cada una de las cuales se inicia con un comando especial, a saber:

`\frontmatter`. Se utiliza para el prefacio, el prólogo, los agradecimientos, la tabla de contenido, los derechos de autor, etc. La característica fundamental del material que aparece bajo el alcance de `\frontmatter` es que las páginas se numeran con números romanos i, ii, iii, etc¹. El comando

¹La opción `spanish` del paquete `babel` modifica la numeración romana minúscula.

\chapter{...} crea un capítulo no numerado, el cual aparece listado en la tabla de contenido. Por ejemplo, con \chapter{Prefacio} se crea un capítulo no numerado, con el título simple de “Prefacio”.

\mainmatter. Contiene la parte central del documento; las páginas aparecen numeradas con números arábigos. Los apéndices —capítulos que aparecen bajo el comando \appendix (sección 6.4)— también se incluyen en esta parte.

\backmatter. Se utiliza para el índice alfabético, la bibliografía y material adicional como conclusiones, reconocimientos, información editorial, etc. El comando \chapter{...} crea un capítulo no numerado, el cual aparece listado en la tabla de contenido.

Las instrucciones \frontmatter, \mainmatter y \backmatter, que son enteramente opcionales, se deben escribir después de \begin{document}.

6.2. Numeración de las partes o divisiones

Cada una de las unidades o divisiones principales de un documento tiene un *nivel* pre-asignado, a saber:

Estilo	part	chapter	section	subsection	subsubsection
book o report	-1	0	1	2	3
article	0	no existe	1	2	3

Las divisiones creadas con \paragraph y \ subparagraph tienen niveles 4 y 5, respectivamente. Por defecto, únicamente las divisiones de nivel 2 o inferior se numeran automáticamente. Así, en los estilos book y report se numeran las partes, los capítulos, las secciones y las subsecciones, pero no se numeran las sub-subsecciones. En el estilo article se numeran las partes, las secciones y las subsecciones pero no las sub-subsecciones.

Aunque esta numeración es automática, el usuario puede modificarla por completo. Se puede, por ejemplo, hacer que L^AT_EX numere solamente partes y capítulos, pero no secciones ni subsecciones. O se puede forzar la numeración de todas las divisiones del documento hasta el nivel de las sub-subsecciones o, incluso, hasta los niveles 4 y 5. La manera más simple de evitar la numeración automática es usando los comandos estrella \part*, \chapter*, \section*, etc; pero con ellos, los títulos correspondientes no se incluirán en la tabla de contenido o índice general (sección 6.6) ni en los encabezados de las páginas.

Véase al respecto la nota T_EX-nica de la página 27.

El parámetro-contador `secnumdepth` controla la numeración automática; su valor por defecto es 2, lo cual significa que las partes o divisiones de nivel menor o igual a 2 se numeran automáticamente. El valor de este parámetro se puede cambiar usando `\setcounter` (sección 3.25).

Ejemplo La instrucción `\setcounter{secnumdepth}{1}`, colocada en el preámbulo de un documento escrito con el estilo `article`, hará que se numeren las partes (al usar `\part{...}`) y las secciones (al usar `\section{...}`), pero no se numerarán las subsecciones, ni las sub-subsecciones, ni las divisiones creadas con `\paragraph` o `\ subparagraph`.

6.3. El documento raíz

Al escribir un documento grande —tal como un libro con varios capítulos o un artículo con muchas secciones— lo más conveniente es sub-dividir el documento fuente en varias porciones. Trabajar con un único e inmenso documento fuente sería muy ineficiente. Con L^AT_EX podemos sub-dividir el documento fuente en la siguiente forma:

1. Un documento raíz (en inglés, *root file*) que contiene el preámbulo del documento, con los comandos `\title`, `\author`, etc, y todos los demás comandos de estructura y formato.
2. Varios documentos subsidiarios que contienen únicamente el material o texto mismo del documento. Estos documentos no tienen preámbulo; en particular, no contienen las instrucciones `\begin{document}` y `\end{document}`.

El único documento que se procesa (*typeset*) es el documento raíz. Los documentos subsidiarios se añaden al documento raíz por medio de instrucciones `\include{...}`, que se escriben después de `\begin{document}`. Para incluir, por ejemplo, el documento subsidiario `cap2.tex` en el documento raíz se escribe `\include{cap2}`, ¡sin la extensión `tex`!

Ejemplo Si un libro consta de un prefacio, una introducción, cuatro capítulos y dos apéndices, podemos sub-dividir el contenido de todo el libro en nueve documentos: un documento raíz, al que podemos llamar `libro.tex`, y los ocho documentos subsidiarios `pref.tex`, `intro.tex`, `cap1.tex`, `cap2.tex`, `cap3.tex`, `cap4.tex`, `apend1.tex` y `apend2.tex`. El documento raíz, `libro.tex`, puede entonces tener el siguiente aspecto:

```
\documentclass[12pt]{book}
:
}

\begin{document}
\frontmatter
\include{pref}
\mainmatter
\include{intro}
\include{cap1}
\include{cap2}
\include{cap3}
\include{cap4}
\appendix
\include{apend1}
\include{apend2}
\end{document}
```

Obsérvese que la extensión `.tex` no se escribe en `\include{...}`. El prefacio (`pref.tex`) está bajo el alcance de `\frontmatter`, por lo que sus páginas serán numeradas con números romanos. La numeración arábiga 1, 2, 3, ... comenzará con la introducción ya que la instrucción `\include{intro}` aparece inmediatamente después de `\mainmatter`. Los dos apéndices aparecen bajo el comando `\appendix` (sección 6.4).

Una ventaja adicional del procedimiento expuesto es que no es necesario procesar *todo* el documento cuando el autor haga modificaciones aisladas en un capítulo o sección. Esto se logra por medio de la instrucción `\includeonly{...}`. A diferencia de `\include{...}`, el comando `\includeonly{...}` debe aparecer en el preámbulo del documento raíz, es decir, antes de `\begin{document}`. Con referencia al ejemplo anterior, si se escribe

```
\includeonly{cap2,cap4,apend1}
```

en el preámbulo del documento raíz, `libro.tex`, L^AT_EX procesará únicamente los capítulos 2 y 4, y el apéndice 1, pero ¡mantendrá la numeración de las páginas, capítulos, secciones, ecuaciones, etc y todas las referencias cruzadas! porque L^AT_EX *asume* que los documentos omitidos (en este caso: `intro.tex`, `pref.tex`, `cap1.tex`, `cap3.tex` y `apend2.tex`) son exactamente los mismos que fueron incluidos, con las instrucciones `\include`, en el último procesamiento. Esto supone, por supuesto, que el documento raíz ha sido procesado con anterioridad, por lo menos una vez, con *todos* sus documentos subsidiarios invocados con `\include`.

La gran limitación de `\include` es la siguiente: cada vez que aparece el comando `\include` se inicia una nueva página. Por lo tanto, el usuario debe saber de antemano dónde hacer la partición del texto. Para un libro con varios capítulos (relativamente cortos), esto no es un problema serio porque todo capítulo comienza —siguiendo las buenas costumbres— en una página nueva, pero puede presentar dificultades en otro tipo de documentos o si los capítulos son muy extensos.

Un comando muy similar a `\include{...}` es `\input{...}`; ambos se usan en forma enteramente análoga pero se diferencian en lo siguiente:

1. Con `\input{...}` *no* se inicia una nueva página, como sucede con `\include{...}`.
2. Si un documento `doc.tex` ha sido invocado con `\include{doc}`, entonces en el documento `doc.tex` no puede haber otro comando `\include`. En otras palabras, `\include` no admite inserciones múltiples de otros comandos `\include`. Por el contrario, en un documento invocado con `\input` puede haber varios comandos `\input`, y esto puede suceder múltiples veces.
3. Con `\input{...}` *no* existe un mecanismo de exclusión similar al proporcionado por `\includeonly{...}`².

6.4. Apéndices

La instrucción `\appendix` cambia la forma en que se numeran los capítulos, secciones, subsecciones y sub-subsecciones. Así, en los estilos `report` y `book`, los capítulos (creados con `\chapter{...}`) que aparezcan debajo de `\appendix` se numeran en la forma A, B, C, etc y en la primera página se imprimen los rótulos ‘Appendix A’, ‘Appendix B’, etc (en inglés) o ‘Apéndice A’, ‘Apéndice B’, etc si se usa el paquete `babel`. Las secciones se numeran en la forma A.1, A.2, A.3, etc y las subsecciones en la forma A.1.1, A.1.2, A.1.3, ..., B.1.1, B.1.2, ..., etc.

En el estilo `article`, las secciones (creadas con `\section{...}`) que aparezcan debajo de `\appendix` se numeran en la forma A, B , C, etc (sin los rótulos ‘Apéndice’ o ‘Appendix’) y las subsecciones se numeran en la forma A.1, A.2, ..., B.1, B.2, B.3, ..., etc.

La instrucción `\appendix` no genera texto alguno ni tiene otro propósito que el explicado arriba.

²El paquete `subfiles`, descrito en la sección 13.10, es otra alternativa para el manejo de un documento raíz con documentos subsidiarios.

6.5. Redefinición de títulos o rótulos

Los títulos o rótulos como ‘Table’, ‘Figure’, ‘Index’, ‘Bibliography’, ‘Chapter’ etc son colocados automáticamente por L^AT_EX. Al usar el paquete `babel` con la opción `spanish`, estos nombres se traducen adecuadamente. Incluso sin usar el paquete `babel`, el usuario puede escoger un nombre alternativo para cualquiera de estos rótulos automáticos; en esta sección explicaremos la forma de hacerlo.

En la Tabla 6.1 se muestran los comandos o entornos cuyos títulos o rótulos son redefinibles en L^AT_EX, junto con los rótulos obtenidos por defecto y los correspondientes rótulos obtenidos al usar `babel`, opción `[spanish]`, versión posterior al año 2000. En la Tabla 6.2 aparecen los comandos que controlan los nombres de los rótulos; obsérvese que todos tienen la forma `\...name`. Para cambiar los nombres sólo tenemos que usar `\renewcommand` escribiendo

```
\renewcommand{\...name}{Nuevo nombre}
```

La parte `{Nuevo nombre}` puede contener también comandos para cambiar el tamaño o el tipo de letra.

Es buena idea colocar estas redefiniciones en el preámbulo del documento para una fácil referencia durante todo el proceso de edición. No obstante, si un paquete invocado en el preámbulo hace redefiniciones previas a los rótulos, las nuevas redefiniciones que deseé realizar el usuario deben aparecer *después* de `\begin{document}` y no en el preámbulo. Esta situación se presenta con el paquete `babel`, opción `spanish`, y se ilustra en los siguientes ejemplos.

Ejemplo Si se usa el paquete `babel` con la opción `spanish` se obtiene el rótulo ‘Cuadro’ para las tablas. Si el usuario prefiere la denominación ‘Tabla’ debe redefinir el comando `\tablename` (Tabla 6.2) así:

```
\renewcommand{\tablename}{Tabla}
```

Análogamente, con `babel` se obtiene el título ‘Índice de cuadros’ al usar el comando `\listoftables` (sección 6.7). Si se prefiere el título ‘Índice de tablas’ hay que redefinir el comando `\listtablename` (Tabla 6.2) así:

```
\renewcommand{\listtablename}{\'Indice de tablas}
```

Véase más sobre rótulos y epígrafes para tablas y gráficas en la sección 7.5.

Ejemplo Con el paquete `babel` se obtiene el título ‘Índice general’ al usar la instrucción `\tableofcontents` (sección 6.6). Si el usuario prefiere el título ‘Contenido’ debe redefinir el comando `\contentsname` (Tabla 6.2) por medio de `\renewcommand{\contentsname}{Contenido}`.

Comando o entorno	Rótulo obtenido por defecto	Rótulo obtenido con babel (spanish)
<code>abstract</code> (entorno)	Abstract	Resumen
<code>\appendix</code>	Appendix (book)	Apéndice (book)
<code>\chapter{...}</code>	Chapter	Capítulo
<code>figure</code> (entorno)	Figure	Figura
<code>\listoffigures</code>	List of Figures	Índice de figuras
<code>\listoftables</code>	List of Tables	Índice de cuadros
<code>\makeindex</code> y <code>theindex</code> (entorno)	Index	Índice (article) Índice alfabético (book)
<code>proof</code> (entorno)*	Proof	Demostración
<code>\part{...}</code>	Part	Parte
<code>table</code> (entorno)	Table	Cuadro
<code>\tableofcontents</code>	Table of Contents	Índice general
<code>thebibliography</code> (entorno)	References (article) Bibliography (book)	Referencias (article) Bibliografía (book)

* El entorno `proof` sólo está disponible en el paquete `amsthm` (véase 4.32.3)

TABLA 6.1. Comandos o entornos con rótulos redefinibles.

Comando o entorno	Comando que controla el nombre o rótulo
<code>abstract</code> (entorno)	<code>\abstractname</code>
<code>\appendix</code>	<code>\appendixname</code>
<code>\chapter{...}</code>	<code>\chaptername</code>
<code>figure</code> (entorno)	<code>\figurename</code>
<code>\listoffigures</code>	<code>\listfigurename</code>
<code>\listoftables</code>	<code>\listtablename</code>
<code>\makeindex</code>	<code>\indexname</code>
<code>proof</code> (entorno)	<code>\proofname</code>
<code>\part{...}</code>	<code>\partname</code>
<code>table</code> (entorno)	<code>\tablename</code>
<code>\tableofcontents</code>	<code>\contentsname</code>
<code>thebibliography</code> (entorno)	<code>\refname (article)</code> <code>\bibname (book)</code>
<code>theindex</code> (entorno)	<code>\indexname</code>

TABLA 6.2. Comandos que controlan los nombres de los rótulos.

6.6. Tabla de contenido o índice general

La instrucción `\tableofcontents` genera la tabla de contenido o índice general del documento, bajo el título de ‘Table of Contents’, en inglés, o ‘Índice General’, si se usa el paquete `babel` (Tabla 6.1). Al usar los estilos `report` o `book`, el sitio más apropiado para colocar esta instrucción es debajo de `\frontmatter` (sección 6.1) para que la tabla de contenido aparezca impresa al inicio del documento.

El comando `\tableofcontents` también crea un documento auxiliar, con el mismo nombre que el documento fuente, pero con extensión `.toc`. El documento ‘`—.toc`’ contiene la información que L^AT_EX utiliza para producir la tabla de contenido. Dicha información proviene de los comandos divisionales `\part`, `\chapter`, `\section`, etc. Cada vez que se añada o modifique alguno de estos comandos, el documento fuente ‘`—.tex`’ (si tiene la instrucción `\tableofcontents`) debe procesarse por lo menos *dos* veces para que la información incluida en la tabla de contenido quede actualizada.

Como se indicó en la sección 6.1, cada comando divisional o subdivisión `\unidad` admite un argumento opcional para cambiar el texto que aparece listado en el índice general y en los encabezados de las páginas:

`\unidad[título abreviado]{título completo}`

Ejemplo La instrucción

`\section[Comprobación de la teoría de Zeniev]{Experimentos de laboratorio que comprueban la teoría de Zeniev}`

crea una sección titulada ‘Experimentos de laboratorio que comprueban la teoría de Zeniev’ la cual aparecerá de forma simplificada en el índice general y en los encabezados de las páginas como: ‘Comprobación de la teoría de Zeniev’.

Los títulos obtenidos con comandos estrella como `\chapter*{...}`, `\section*{...}`, (sección 6.2) *no* admiten el argumento opcional para título abreviado y no se tienen en cuenta para la tabla de contenido (ni los encabezados de las páginas). No obstante, el usuario puede añadir al índice general los títulos que desee. Para añadir un determinado *Título* en la tabla de contenido se usa la instrucción

`\addcontentsline{toc}{Unidad divisional}{Título}`

donde la *Unidad divisional* es cualquiera de las mencionadas en la sección 6.1, es decir: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph` o `subparagraph`. Esta instrucción se escribe en el documento

fuente ‘`—.tex`’, en el sitio en el que se desea el nuevo título. Los títulos añadidos aparecerán listados en el índice general, con los números de las páginas generados automáticamente. Debemos enfatizar que el comando `\addcontentsline` no modifica la numeración ya existente de capítulos, secciones, subsecciones, figuras, etc; este comando está diseñado exclusivamente para agregar títulos al índice general, sin alterar ningún contador.

Ejemplo Si hemos creado la sección `\section*{Testimonios antiguos}`, su título no aparecerá en el índice general (debido al uso de `*`), pero podemos forzar su inclusión, junto con el número de la página en la que dicha sección se inicia, si escribimos en el documento fuente

```
\section*{Testimonios antiguos}
\addcontentsline{toc}{section}{Testimonios antiguos}
```

Un comando similar a `\addcontentsline` es:

```
\addtocontents{toc}{algo}
```

Se usa para añadir `algo` (texto, tablas, espacio vertical, etc) al índice general, pero sin el correspondiente número de página.

Ejemplo La instrucción

```
\section{Técnicas de rastreo espectral}
\addtocontents{toc}{Lectura opcional}
```

crea la sección ‘Técnicas de rastreo espectral’, la cual aparecerá listada en la forma usual en el índice general, pero con el comentario ‘Lectura opcional’ escrito en el renglón siguiente.

6.7. Índices de tablas y figuras

La instrucción `\listoftables` genera el índice de tablas del documento, bajo el título de ‘List of Tables’, en inglés, o ‘Índice de cuadros’, si se usa el paquete `babel` (Tabla 6.1). Similarmente la instrucción `\listoffigures` genera el índice de figuras o gráficas del documento, bajo el título de ‘List of Figures’, en inglés, o ‘Índice de Figuras’, en español (Tabla 6.1). Al usar los estilos `report` o `book`, el sitio más apropiado para colocar estas instrucciones es debajo de `\frontmatter` y `\tableofcontents` para que los índices de tablas y figuras aparezcan impresos inmediatamente después del índice general.

El comando `\listoftables` también crea un documento auxiliar, con el mismo nombre que el documento fuente, pero con extensión `.lot`. Análogamente, `\listoffigures` crea un documento auxiliar con extensión `.lof`.

Estos dos documentos auxiliares contienen la información que L^AT_EX utiliza para producir los índices de tablas y figuras. Dicha información proviene de los comandos `\caption{...}` utilizados en los entornos `table` o `figure` (véase la sección 7.5). Esto significa que si se no se usa el comando `\caption` al incluir en el documento fuente una determinada tabla o figura, dicha tabla o figura no aparecerá automáticamente en el índice correspondiente. Pero, de manera similar a lo que ocurre con el índice general (sección 6.6), los índices de tablas y figuras se pueden modificar. El comando

```
\addcontentsline{lot}{table}{text}
```

se usa para añadir determinado `text` en la lista de tablas y

```
\addcontentsline{lof}{figure}{text}
```

hace lo propio en la lista de figuras. El `text` añadido puede incluir tanto el número que se quiera para la tabla o figura, como su rótulo o epígrafe. Al añadir numeración a las tablas o a las figuras en los respectivos índices, hay que proceder con cierta cautela. Concretamente, hay que hacerlo en la siguiente forma:

```
\protect\numberline{numero}{rotulo}
```

Ejemplo El comando

```
\addcontentsline{lot}{table}{\protect\numberline{4.3}{Datos}}
```

escrito en el documento fuente ‘—.tex’ (si en él aparece la instrucción `\listoftables`) hace que se añada la siguiente línea en el índice de tablas:

4.3 Datos.....78

siendo 78 la página en la que dicha instrucción aparece (recuérdese que los números de las páginas son generados *automáticamente* por L^AT_EX al elaborar los índices).

Las complicaciones que presenta `\addcontentsline` se pueden evitar usando el comando `\caption` en todas las tablas y figuras del documento.

Un comando similar a `\addcontentsline` es `\addtocontents`; las instrucciones

```
\addtocontents{lot}{algo} ó \addtocontents{lof}{algo}
```

se usan para añadir `algo` (texto, espacio vertical, etc) al índice de tablas y al índice de figuras, respectivamente, pero sin el correspondiente número de página.

6.8. Elaboración manual de índices alfabéticos

En los estilos `article`, `book` y `report`, L^AT_EX posee el entorno `theindex` para la elaboración manual de un índice alfabético (que no debe confundirse con el índice general o tabla de contenido, tema de la sección 6.6). El índice alfabético contiene los términos, conceptos, personajes, etc, ordenados alfabéticamente, que el autor considere útiles como guía para la lectura o consulta del documento, principalmente un libro o monografía. El autor escribe él mismo los términos, los ordena alfabéticamente y busca los números de las páginas leyendo la versión *definitiva* del documento ‘`—.dvi`’.

Un determinado término del índice se escribe con `\item`, en la forma

<code>\item término páginas</code>

Los términos o vocablos subsidiarios que deban aparecer en un segundo nivel se escriben con `\subitem`, y los de tercer nivel con `\subsubitem`. No se permiten más subdivisiones. El comando `\indexspace` produce un espacio vertical extra entre términos y es especialmente útil para separar palabras del índice que comiencen con letras diferentes.

El índice generado por el entorno `theindex` se imprime a dos columnas, en una página nueva y bajo el título de ‘Index’, en inglés, o ‘Índice alfabético’, en español (Tabla 6.1).

Ejemplo A la izquierda: texto en el documento fuente; a la derecha: resultado obtenido. El título ‘Índice alfabético’ se obtiene, en realidad, en el tamaño de letra utilizado por L^AT_EX para los títulos de los capítulos.

```
\begin{theindex}
:
\item número 2, 5
\subitem complejo 17, 58--63
\subitem entero 9
\subitem primo 22, 27
\subsubitem irregular 29, 32, 45
\subsubitem regular 28, 32
\subitem racional 37, 41
\subitem real 3, 42, 47
\indexspace
\item orden 3, 95--103
\subitem parcial 96
\subitem total 99, 100
:
\end{theindex}
```

Índice alfabético

: número 2, 5 complejo 17, 58--63 entero 9 primo 22, 27 irregular 29, 32, 45 regular 28, 32 racional 37, 41 real 3, 42, 47 orden 3, 95--103 parcial 96 total 99, 100 :
--

6.9. Elaboración de índices alfabéticos con el programa *MakeIndex*

Elaborar manualmente el índice alfabético de un documento grande, en la forma explicada en la sección anterior, es una labor dispendiosa. El paquete *makeidx* de L^AT_EX 2 _{ε} y el programa *MakeIndex*³ ayudan a simplificar esta tarea, ordenando alfabéticamente los términos o vocablos, previamente escogidos por el autor, y generando automáticamente los números de las páginas. El usuario utiliza el comando `\index`, en el documento fuente, para señalar los términos que quiera listar en el índice. El uso de `\index` se explica en la sección 6.9.1.

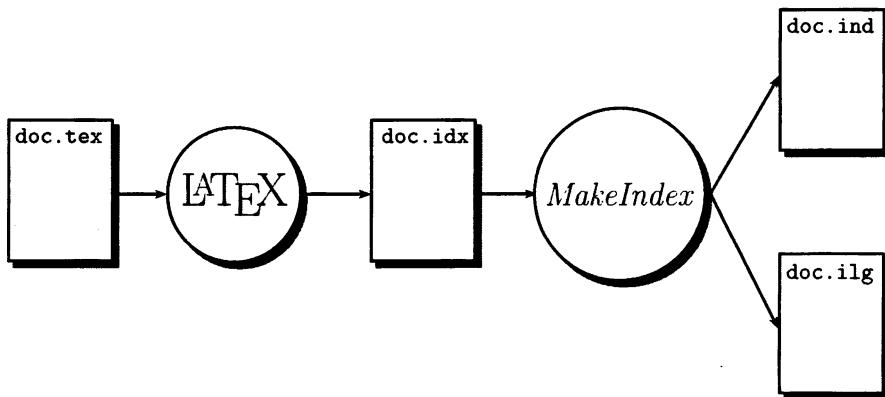


FIGURA 7.1. L^AT_EX y *MakeIndex* en acción.

Para ilustrar el empleo de *MakeIndex*, presentamos a continuación los cinco pasos que hay que seguir para generar el índice de un documento típico, *doc.tex*, una vez que se hayan escogido, con los comandos `\index`, los términos a ser listados. Los primeros cuatro pasos de este procedimiento se ilustran en la Figura 7.1.

1. Escribir las instrucciones

```
\usepackage{makeidx}
\makeindex
```

en el preámbulo del documento *doc.tex*.

³ *MakeIndex*, escrito en 1987–1993 por Pehong Chen y Nelson Beebe, se incluye en las distribuciones estandarizadas de L^AT_EX 2 _{ε} .

2. Escribir la instrucción `\printindex` en el sitio del documento fuente en el que se desee imprimir el índice. Por lo general, `\printindex` se escribe inmediatamente antes de `\end{document}`.
3. Procesar con L^AT_EX, en la forma usual, el documento `doc.tex`. Al encontrar la instrucción `\makeindex`, L^AT_EX crea el archivo auxiliar `doc.idx` que contiene toda la información de los comandos `\index`.
4. Correr el programa *MakeIndex* con el archivo `doc.idx`. Hay que tener presente que *MakeIndex* es un programa *externo*, escrito en el lenguaje de programación *C*, y su ejecución depende de la plataforma computacional utilizada. En muchos casos, se ejecuta desde la línea de comandos o *prompt*, en la forma⁴:

`makeindex doc.idx`

o simplemente

`makeindex doc`

El ejecutable puede llevar el nombre `makeindex.exe` o `makeindx.exe`, lo cual depende de la implementación local⁵. En algunas implementaciones, como *WinEdt*, la ejecución de *MakeIndex* se hace con un simple *click*, ubicando la opción *MakeIndex* en algún menú apropiado, tal como el menú de ‘Accesorios’ o el menú de ‘Herramientas’.

Una vez ejecutado, *MakeIndex* crea el archivo `doc.ind` con la información completa sobre los términos y sus páginas, tal como aparecerán en el índice. Si *MakeIndex* encuentra un error en alguna de las instrucciones `\index{...}`, la rechaza o emite una advertencia. El usuario siempre sabe cuántos comandos `\index` han sido aceptados, cuántos han sido rechazados y cuántas advertencias hay, ya que *MakeIndex* reporta algo como

```
This is makeindx, portable version 2.12 [26-May-1993].  
Scanning input file doc.idx....done  
      (51 entries accepted, 3 rejected).  
Sorting entries....done (97 comparisons).  
Generating output file doc.ind....done  
      (110 lines written, 2 warnings).  
Output written in doc.ind.  
Transcript written in doc.ilg.
```

⁴ Algunas plataformas computacionales sobre las que se ejecuta *MakeIndex* admiten un máximo de ocho símbolos para los nombres de sus archivos.

⁵ *MakeIndex* se puede ejecutar sujeto a algunas opciones; véase al respecto la sección 6.9.3, en la página 203.

Los errores y advertencias se pueden consultar en el archivo auxiliar `doc.ilg`; en él *MakeIndex* emite su “diagnóstico” sobre la posible causa de los errores e indica los números de los renglones (con referencia al archivo `doc.idx`) en los que éstos aparecen. Los errores se deben corregir (en el documento fuente `doc.tex`) y el procedimiento explicado debe repetirse hasta que el número de comandos `\index` (`entries`) rechazados sea 0.

5. Finalmente, el documento `doc.tex` debe ser procesado por L^AT_EX una vez más y el documento `doc.dvi` mostrará el índice impreso a dos columnas, en una página nueva y bajo el título de ‘Index’, en inglés, o ‘Índice alfabético’, en español (Tabla 6.1).

Para producir el índice, L^AT_EX utiliza el documento `doc.ind` creado por *MakeIndex*. Si el usuario lo examina en detalle, observará que el archivo `doc.ind` tiene la estructura del entorno `theindex` mencionado en la sección 6.8, a saber,

```
\begin{theindex}
...
\item
  \subitem
    \subsubitem
...
\item
  \subitem
    \subsubitem
...
\end{theindex}
```

con los términos y subtérminos ordenados alfabéticamente y los números de las páginas escritos en orden creciente. El orden alfabético utilizado por *MakeIndex* es el siguiente orden estándar internacional: primero los símbolos no alfabéticos, luego los dígitos, luego las letras mayúsculas y finalmente las letras minúsculas.

6.9.1. El uso del comando `\index`

El comando `\index` se utiliza para señalar, en el documento ‘`—.tex`’, los términos o vocablos que el autor quiera que aparezcan listados en el índice alfabético; `\index` se escribe inmediatamente después del término que se desea indexar. No se debe dejar ningún espacio entre el término en cuestión y el comando `\index` para evitar un inesperado cambio de página y, por consiguiente, errores de paginación en el índice.

Ejemplo Si en el documento fuente aparece el texto

Según la concepción de Darwin,\index{Darwin, Charles} la evolución\index{evolución} biológica es el resultado de la selección natural\index{selección natural}.

los términos ‘evolución’, ‘selección natural’ y ‘Darwin, Charles’ se incluirán en el índice, en la ubicación alfabética que les corresponda y con su respectivo número de página.

En la elaboración de índices alfabéticos se cometan con frecuencia muchos errores, por lo cual se hace necesario tener mecanismos de control que permitan hacer un seguimiento o rastreo de los diferentes comandos \index incluidos en el documento. L^AT_EX posee dos importantes herramientas de control; se describen en la sección 6.9.2, página 202.

A continuación detallamos otros importantes aspectos del uso de \index.

\index y \makeindex. Los comandos \index son ignorados si no aparece la instrucción \makeindex en el preámbulo del documento. El usuario puede, por lo tanto, agilizar los trabajos de edición escribiendo \makeindex únicamente cuando vaya a elaborar el índice alfabético.

Sub-términos. Para listar términos subsidiarios o subdivisiones de segundo nivel, el argumento de \index debe contener tanto el término principal como el subsidiario, separados por el símbolo !, en la forma

`\index{término!subtérmino}`

Para subdivisiones de tercer nivel se escribe

`\index{término!subtérmino!sub-subtérmino}`

L^AT_EX y *MakeIndex* sólo admiten dos niveles de subdivisión.

Ejemplo Supóngase que en el documento fuente aparece el texto

Siguiendo a Kummer, un número primo \$p\$ se dice que es regular\index{número!primo!regular} si \$p\$ no divide a \$h(p)\$.

Si lo anterior resulta incluido en la página 35 del documento final ‘**.dvi**’, entonces el índice mostrará algo como

⋮
 número
 ⋮
 primo
 regular, 35
 ⋮

Referencias múltiples. Un mismo concepto o idea puede aparecer listado en el índice bajo dos o más términos diferentes; esto se logra usando tantos comandos `\index` como se deseé. En tales situaciones, lo más aconsejable es escribir los comandos `\index` en renglones separados, terminados en %. El símbolo % impide que se agregue espacio adicional, evitando así cambios de página no deseados.

Ejemplo Supóngase que en el documento fuente aparece el texto

```
Según la concepción de Darwin, la evolución biológica
\index{evolución}%
\index{teoría!evolutiva}%
es el resultado de la selección natural.
```

Si lo anterior resulta incluido en la página 92 del documento final ‘...`.dvi`’, entonces el índice mostrará algo como

⋮
 evolución, 92
 ⋮
 teoría
 ⋮
 evolutiva, 92
 ⋮

Expresiones varias. Si en el documento fuente se escribe

```
\index{palabra1@palabra2}
```

entonces la expresión `palabra2` aparecerá listada en el índice en la posición alfabética determinada por `palabra1`. Esto es útil para incluir en el índice alfabético símbolos como π , S_n , a^{-1} , etc o cualquier otro tipo de expresión no verbal que el autor desee destacar. Esta

modalidad de `\index` también es útil para forzar un tipo de letra (italica, negrilla, etc) en términos del índice.

Ejemplo Si en el documento fuente aparece en algún lugar el comando `\index{pi@π}`, entonces en el índice se listará el símbolo π en la posición alfabética determinada por la expresión ‘pi’. Nótese que si se escribe simplemente `\index{π}`, el símbolo π aparecerá listado antes de cualquier otra palabra ya que, en el orden alfabético utilizado por *MakeIndex*, el símbolo ‘\$’ antecede a todas las letras del alfabeto.

Ejemplo Para hacer que la palabra *software* (en itálicas) aparezca en el índice, escribimos en el documento fuente:

```
\index{software@\textit{software}}
```

Rangos de páginas. Para especificar un rango de páginas se escribe `\index{...|()}` al inicio del rango, e `\index{...|)}` al final.

Ejemplo Si en el documento fuente aparece en un sitio determinado el comando `\index{número!primo|()` y varias páginas más adelante aparece la instrucción `\index{número!primo|)}`, entonces en el documento final se verá algo como

```
:  
número  
:  
primo, 26–37  
:
```

siendo 26 la primera página del rango y 37 la última (en el documento ‘`.dvi`’). Los números mismos 26 y 37 son generados por *MakeIndex* y *LATEX* al elaborar el índice.

Espacios en el argumento de `\index`. En el argumento de `\index` los espacios son tenidos en cuenta ya que *MakeIndex* trata el espacio en blanco como un símbolo. Por ejemplo, las instrucciones

```
\index{ conjunto}, \index{conjunto }, \index{conjunto}
```

producen en el índice tres términos diferentes; el primero de ellos aparece al inicio del índice porque en el orden alfabético utilizado por *MakeIndex*, el símbolo precede a todas las letras del alfabeto. El usuario debe, por lo tanto, evitar espacios no deseados en el argumento de `\index` (véase al respecto la sección 6.9.3).

Mayúsculas y minúsculas. *MakeIndex* distingue las letras mayúsculas de las minúsculas. Así, las instrucciones

```
\index{conjunto}, \index{Conjunto}, \index{CONJUNTO}
```

producen tres términos diferentes en el índice.

Números en letra cursiva o negrilla. Algunos autores destacan, ya sea usando letra cursiva o negrilla, ciertos números de páginas con el propósito de resaltar los sitios en los que, por ejemplo, se definen los términos listados. Para que el número de la página aparezca en letra cursiva (ítálica) se escribe `\index{...|textit}` y para que aparezca en negrilla se escribe `\index{...|textbf}`. Análogamente, `\index{...|textsf}` produce números en letra sans serif.

Ejemplo Si en el sitio correspondiente a la página 13 se ha escrito

```
\index{número!real|textbf}
```

y en el sitio correspondiente a la página 32 se ha escrito

```
\index{número!complejo|textbf}
```

entonces el índice mostrará algo como

```
:  
número  
:  
complejo, 32, 35  
real, 4, 13, 16–28  
:
```

Cursiva o negrilla en rangos de páginas. Los cambios de tipo de letra también se pueden usar en rangos de páginas. Para letra cursiva se usa

```
\index{...|(textit} ... \index{...|)}
```

y para negrilla

```
\index{...|(textbf} ... \index{...|)}
```

Ejemplo Si en el sitio correspondiente a la página 93 se ha escrito

`\index{compacidad| (textit)}`

y en el sitio correspondiente a la página 117 se ha escrito

`\index{compacidad|})}`

entonces el índice mostrará

:

compacidad, 93–117

:

Referencias cruzadas. L^AT_EX tiene predefinido el comando `\see` para producir en el índice la expresión ‘*see*’ (en inglés) o ‘*véase*’ (si se usa el paquete `babel`). Se emplea en la forma `\index{...|see{...}}`.

Ejemplo Si se carga el paquete `babel` con la opción `spanish`, una referencia en el índice alfabético tal como

:

continuidad, véase función continua

:

se obtiene al escribir en el documento fuente la instrucción

`\index{continuidad|see{función continua}}.`

Puesto que `\index{...|see{...}}` no genera en el índice ningún número de página, este comando se puede colocar en cualquier parte del documento fuente, después de `\begin{document}`.

Símbolos acentuados. Los símbolos acentuados como á, é, í, ó, ú, ñ, etc se pueden usar libremente en el argumento de `\index` siempre y cuando se haya cargado en el documento fuente el paquete `inputenc` (sección 3.1), pero al crear el documento ‘`—.idx`’, estos símbolos son “traducidos” por L^AT_EX en las instrucciones `\'a`, `\'e`, `\'{\i}`, etc. Desafortunadamente, *MakeIndex* considera que una expresión como `\'e` es una secuencia de tres símbolos y la ordena alfabéticamente como tal y no como é. Por lo tanto, si hay tildes, es necesario indicarle a *MakeIndex* la posición alfabética deseada. Así por ejemplo, si el vocablo ‘álgebra’ se va a incluir en el índice, debemos escribir

`\index{álgebra@\\'algebra} o \index{álgebra@\\'algebra}`

y no simplemente `\index{álgebra}` o `\index{\'algebra}`. Con estos dos últimos comandos, el vocablo ‘álgebra’ aparecerá al tope del índice, antes de cualquier palabra ya que, en el orden alfabético utilizado por *MakeIndex*, el símbolo \ antecede a cualquier letra.

Los símbolos especiales !, @, |. Puesto que los símbolos !, @ y | tienen un significado especial para *MakeIndex*, no se pueden usar libremente en el argumento de `\index`. Si el usuario necesita que alguno de estos símbolos aparezca explícitamente en un término o expresión del índice, debe escribir " antes del símbolo. Así por ejemplo, para incluir la expresión Eureka! en el índice, se debe escribir `\index{Eureka!"!}`. Otro ejemplo: para hacer que la expresión matemática $|x|$ aparezca en el índice, en la posición alfabética de la palabra ‘valor’, escribimos `\index{valor@$"|x|"|$}`.

6.9.2. Rastreo de los comandos `\index`

Para hacer un seguimiento de los comandos `\index` utilizados por el usuario, hay dos mecanismos de rastreo, siempre incluidos en las distribuciones de L^AT_EX 2_ε:

1. El paquete `showidx`. Al escribir `\usepackage{showidx}` en el preámbulo del documento, todos los términos indexados con `\index` aparecerán en los márgenes del documento ‘—.dvi’, a la altura de los sitios en los que fueron definidos.
2. El archivo `idx.tex`. Hay que ubicar primero este archivo (en el subdirectorio `/texmf/tex/latex/base` o equivalente), y procesarlo luego como un documento L^AT_EX normal. Se verá en la pantalla algo como:

```
*****
* Enter idx file's first name. *
*****
\filename=
```

El usuario debe escribir el nombre del documento ‘—.idx’, que es el mismo nombre del documento principal, sin la extensión, indicando la ruta completa (por ejemplo, C:/Documentos/Proyectos/doc). El archivo `idx.dvi` mostrará, en un formato a dos columnas, todos los términos y subtérminos que hayan indexados con `\index`, resaltando en negrilla los números de las páginas en los que fueron definidos. Esta información es la misma que aparece en el archivo `doc.idx` anteriormente generado por L^AT_EX, pero es más fácil de examinar y más útil para imprimir.

6.9.3. Opciones del programa *MakeIndex*

Hemos explicado el funcionamiento básico del programa externo *MakeIndex* (etapa 4, página 195); a continuación nos referimos a algunas opciones para su ejecución. Dado un archivo *doc.idx*, *MakeIndex* se puede ejecutar, sujeto a un cierto número de *opciones*, escribiendo

```
makeindex opciones doc.idx
```

en la línea de comandos. Entre las opciones disponibles destacamos las siguientes:

- c habilita la compresión de espacios en blanco. Como se explicó arriba (página 199), los espacios en blanco que aparezcan en el argumento de \index{...} son tenidos en cuenta. La opción -c ignora los espacios en blanco iniciales y finales, y comprime los espacios intermedios en uno solo.
- l los términos se ordenan alfabéticamente teniendo en cuenta únicamente las letras: los espacios en blanco intermedios, en expresiones de dos o más palabras, son ignorados al establecer el orden.
- r inhabilita la formación de rangos de páginas. Por defecto, *MakeIndex* forma rangos de tres o más páginas; por ejemplo, si un término particular aparece indexado en las páginas 24, 25 y 26, en el índice se obtendrá la paginación en la forma 24–26.
- p n hace que el índice se imprima a partir de una página numerada con el número n. Esta opción es útil para procesar índices por separado.

Ejemplo Al ejecutar

```
makeindex -c doc.idx
```

MakeIndex elabora el índice alfabético del documento fuente *doc.tex* comprimiendo los espacios en blanco.

Ejemplo Al ejecutar

```
makeindex -c -p 253 doc.idx
```

el índice alfabético del documento fuente *doc.tex* es elaborado por *MakeIndex* con compresión de espacios en blanco y se imprime (si en el documento aparece la instrucción \printindex) a partir de una página numerada con el número 253.

6.9.4. Índices múltiples con el paquete `multind`

 El paquete `multind` está incluido en el CD adjunto, en la carpeta `/Paquetes/multind/`. Para información sobre la instalación de paquetes nuevos, véase el Apéndice C.

Ocasionalmente son necesarios varios índices alfabéticos; por ejemplo, un índice de términos, uno de autores, uno de lugares geográficos, etc. El paquete `makeidx` de L^AT_EX 2_ε, presentado en las páginas anteriores, permite la elaboración de un único índice, titulado ‘Index’ (o ‘Índice alfabético’, si se usa `babel`). Para crear índices múltiples recomendamos usar el paquete `multind`⁶, el cual redefine los comandos `\makeindex`, `\index` y `\printindex` de L^AT_EX 2_ε, adicionándoles argumentos extras:

```
\makeindex{nombre del índice}
\index{nombre del índice}{término}
\printindex{nombre del índice}{título del índice}
```

Estos comandos extendidos permiten crear varios índices independientes; tanto el *nombre del índice* como su *título* son escogidos por el usuario.

Para usar `multind` junto con *MakeIndex* seguimos, básicamente, los cinco pasos descritos en la página 194, pero con pequeñas diferencias. En primer lugar, se debe cargar el paquete `multind` en lugar del paquete `makeidx` (los dos paquetes no pueden coexistir en un mismo documento). En segundo lugar, se debe correr *MakeIndex* de manera independiente con cada uno de los índices creados. El siguiente ejemplo ilustra el procedimiento.

Ejemplo Supóngase que queremos producir dos índices independientes, uno para conceptos y otro para autores, en el documento `doc.tex`, escrito con el estilo `book`. El primer índice lo denominamos `autores` y el segundo `conceps`. El documento `doc.tex` tiene el siguiente aspecto:

```
\documentclass[12pt]{book}
...
...
\usepackage{multind}
...
...
\makeindex{autores}
\makeindex{conceps}
\begin{document}
...
...
\printindex{autores}{\'Indice de autores}
\printindex{conceps}{\'Indice de conceptos}
\end{document}
```

⁶Escrito por F. W. Long.

El usuario escoge, con los comandos `\index`, los términos o vocablos para indexar, cuidándose de especificar el índice respectivo, en una de estas dos formas:

```
\index{autores}{término}
```

o bien

```
\index{conceps}{término}.
```

Al procesar el documento fuente `doc.tex`, LATEX crea dos archivos de índices: `autores.idx` y `conceps.idx`, y para cada uno de ellos ejecutamos `MakeIndex`:

```
makeindex autores.idx
```

e independientemente

```
makeindex conceps.idx.
```

Finalmente, el documento `doc.tex` debe ser procesado por LATEX una vez más y el documento `doc.dvi` mostrará dos índices independientes, impresos a dos columnas, con los títulos ‘Índice de autores’ e ‘Índice de conceptos’, respectivamente.

6.10. Bibliografías con el programa BIBTEX

El programa BIBTEX⁷ ayuda a producir la bibliografía en documentos LATEX. En el documento fuente ‘`—.tex`’, el usuario hace referencia a las diferentes publicaciones bibliográficas por medio del comando `\cite`, tal como se explicó en la sección 3.26. BIBTEX trabaja sobre bases de datos creadas por el usuario, organiza y ordena los items y, finalmente, LATEX los presenta en uno de los muchos estilos disponibles.

La bibliografía también puede contener publicaciones no citadas en el texto, es decir, publicaciones para las cuales no haya un `\cite`. Estas publicaciones se incluyen con el comando

```
\nocite{clave}
```

el cual se puede escribir en cualquier parte del documento, después de `\begin{document}`. Este comando le informa a BIBTEX que la publicación asociada con la `clave` dada debe aparecer en la bibliografía. Un comando relacionado es `\nocite{*}`, cuyo uso se explica más adelante.

⁷El programa BIBTEX, escrito por Oren Patashnik, se incluye en las distribuciones estandarizadas de LATEX 2ε.

Para ilustrar el empleo de BIBTEX, presentamos a continuación los seis pasos que hay que seguir para generar la bibliografía de un documento típico, `doc.tex`.

1. Crear uno o varios documentos de texto llano, con extensión `bib`, que contienen los datos bibliográficos. El formato o aspecto de los documentos ‘`—.bib`’ se explica en detalle en la sección 6.10.1 (página 207). A manera de ejemplo, suponemos que se han creado dos archivos bibliográficos: `biblio1.bib` y `biblio2.bib`.
2. En el documento fuente `doc.tex`, escribir la instrucción

```
\bibliography{archivos bib}
```

en el sitio exacto en el que se desea que se imprima la bibliografía. Los *archivos bib* se separan con comas pero no se deben escribir las extensiones ni dejar espacios entre las comas. En el caso que nos concierne, escribimos `\bibliography{biblio1,biblio2}`.

3. En el documento fuente `doc.tex`, escribir la instrucción

```
\bibliographystyle{estilo}
```

después de `\begin{document}`. Sobre los *estilos* bibliográficos disponibles, véanse las secciones 6.10.2 y 6.10.3 más adelante.

4. Procesar con LATEX el documento `doc.tex`, en la forma usual. En el archivo auxiliar `doc.aux`, LATEX escribe información sobre las referencias `\cite{...}` y `\nocite{...}` encontradas. Si en el documento fuente LATEX encuentra la instrucción `\nocite{*}`, todos y cada uno de los ítems de las bases de datos ‘`—.bib`’ serán incluidos en la bibliografía. Tal comando es también útil para procesar por separado bases de datos bibliográficas especializadas, como las que se encuentran disponibles en muchas páginas *Web*.
5. Correr el programa BIBTEX con el archivo `doc.aux`. Hay que tener presente que *MakeIndex* es un programa *externo* y su ejecución depende de la plataforma computacional utilizada. En muchos casos, se ejecuta desde la línea de comandos o *prompt*, en la forma⁸:

```
bibtex doc
```

(sin escribir la extensión `aux`). En algunas implementaciones, como WinEdt, la ejecución de BIBTEX se hace con un simple click, ubicando

⁸Algunas plataformas computacionales sobre las que se ejecuta BIBTEX admiten un máximo de ocho símbolos para los nombres de sus archivos.

la opción **BibTeX** en algún menú apropiado, tal como el menú de ‘Accesorios’ o el de ‘Herramientas’.

BIBTEX genera los archivos **doc.bbl** y **doc.blg**; en este último aparece un reporte de los errores encontrados. Los items o campos con errores sintácticos son ignorados; por tal razón, el usuario debe consultar el archivo **doc.blg** para seguir el rastro de los errores.

6. Finalmente, el documento **doc.tex** debe ser procesado por **LATEX** dos veces más y el documento **doc.dvi** mostrará la bibliografía impresa, en una página nueva, bajo los títulos de ‘Bibliography’ o ‘References’, en inglés, y ‘Bibliografía’ o ‘Referencias’, si se usa la opción **spanish** de **babel** (véase la Tabla 6.1).

Consejo práctico: Si **LATEX** o **BIBTEX** reportan el error “*I can’t find file ...*”, se recomienda crear una carpeta o subdirectorio y colocar allí todos los archivos necesarios: el documento fuente ‘**—.tex**’ que se está procesando, las bases de datos ‘**—.bib**’, los estilos bibliográficos ‘**—.bst**’ (secciones 6.10.2 y 6.10.3) y el ejecutable **bibtex.exe**.

6.10.1. Formato de un archivo ‘**—.bib**’

Los archivos con extensión **bib** son documentos de texto llano que contienen las bases de datos utilizadas por **BIBTEX** para generar la bibliografía. En el ejemplo de la página 210 aparece un archivo ‘**—.bib**’ típico que puede ser utilizado como guía para elaboración de otros similares.

Lo primero que tiene que hacer el usuario es decidir el tipo de cada publicación. A continuación aparecen los tipos de publicación permitidos por **BIBTEX**; hay suficientes tipos para acomodar cualquier publicación imaginable. Cada tipo tiene campos obligatorios y campos opcionales, usados por los estilos bibliográficos (véase 6.10.2) para producir la bibliografía final. **BIBTEX** ignora los campos no pertinentes y emite una advertencia (en el archivo ‘**—.blg**’) si falta alguno de los campos obligatorios.

ARTICLE Artículo publicado en una revista o publicación periódica.

Campos obligatorios: **author**, **title**, **journal**, **year**.

Camposopcionales: **volume**, **number**, **pages**, **month**, **note**.

BOOK Libro publicado por una editorial o institución.

Campos obligatorios: **author** o **editor**, **title**, **publisher**, **year**.

Camposopcionales: **volume** o **number**, **series**, **address**,
edition, **month**, **note**.

BOOKLET Trabajo publicado pero no patrocinado por ninguna editorial ni institución.

Campos obligatorios: **title**.

Campos opcionales: **author**, **howpublished**, **address**, **year**, **month**, **note**.

INBOOK Parte de un libro, tal como un capítulo o rango de páginas.

Campos obligatorios: **author** o **editor**, **title**, **publisher**, **chapter** y/o **pages**, **year**.

Campos opcionales: **volume** o **number**, **series**, **type**, **address**, **edition**, **month**, **note**.

INCOLLECTION Parte, con título propio, de un libro.

Campos obligatorios: **author**, **title**, **booktitle**, **publisher**, **year**.

Campos opcionales: **editor**, **volume** o **number**, **series**, **type**, **chapter**, **pages**, **address**, **edition**, **month**, **note**.

INPROCEEDINGS Artículo publicado en las memorias (*proceedings*) de un simposio o coloquio.

Campos obligatorios: **author**, **title**, **booktitle**, **year**.

Campos opcionales: **editor**, **volume** o **number**, **series**, **pages**, **address**, **edition**, **organization**, **publisher**, **month**, **note**.

MANUAL Documentación técnica.

Campos obligatorios: **title**.

Camposopcionales: **author**, **organization**, **address**, **edition**, **year**, **month**, **note**.

MASTERTHESIS Tesis a nivel de Maestría o Magister.

Campos obligatorios: **author**, **title**, **school**, **year**.

Campos opcionales: **type**, **address**, **month**, **note**.

MISC Publicación miscelánea que no corresponde a ninguno de los otros tipos.

Campos obligatorios: ninguno.

Campos opcionales: **author**, **title**, **howpublished**, **year**, **month**, **note**.

PHDTHESES Tesis a nivel de Doctorado o Ph.D.

Campos obligatorios: **author**, **title**, **school**, **year**.

Campos opcionales: **type**, **address**, **month**, **note**.

PROCEEDINGS Memorias (*proceedings*) de un simposio o coloquio.

Campos obligatorios: **title**, **year**.

Campos opcionales: **editor**, **volume** o **number**, **series**, **address**, **organization**, **publisher**, **month**, **note**.

TECHREPORT Reporte o informe técnico publicado por una institución.

Campos obligatorios: **author, title, institution, year**.

Campos opcionales: **type, number, address, month, note**.

UNPUBLISHED Trabajo que no ha sido formalmente publicado, pero con autor y título.

Campos obligatorios: **author, title, note**.

Campos opcionales: **month, year**.

Los ítems bibliográficos se escriben en el archivo ‘**—.bib**’ en la forma

@TIPO{clave, campos}

donde la **clave** de cada ítem es exactamente la que se ha usado en el documento fuente con los comandos **\cite{clave}**. Tanto los ítems de la bibliografía como los **campos** de cada ítem se pueden escribir en cualquier orden; el trabajo de **BIBTEX** es organizar y ordenar toda la información, según el estilo bibliográfico solicitado. Se debe tener presente, además, lo siguiente:

- Cada tipo debe estar precedido por el símbolo @ y en lugar de los corchetes de agrupación exteriores se pueden usar paréntesis.
- Las letras mayúsculas en los tipos de publicación son opcionales. Así, se puede escribir @BOOK o @Book o @book.
- Cada campo va seguido por el símbolo =, con espacios opcionales a su alrededor. El texto que sigue se encierra entre comillas " o entre corchetes { }.
- Los campos se separan entre sí por comas, con espacios opcionales a su alrededor.
- Dos o más autores se separan con **and**.
- En las bases de datos ‘**—.bib**’, **BIBTEX** no acepta caracteres acentuados como á, é, ñ, etc. En su lugar hay que usar los comandos **LATEX** para acentos, pero encerrando cada carácter entre corchetes. Se debe escribir, por ejemplo, G{"o}del, Col{\'o}n, c{\\'i}trico, etc.
- Algunos estilos ignoran las letras mayúsculas, excepto la primera letra de la primera palabra, en los **títulos** de las publicaciones. Para forzar una letra mayúscula hay que encerrarla entre corchetes. Se escribe, por ejemplo, {E}uropa, en vez de simplemente Europa, {N}ewton en vez de Newton, {ONU}, en vez de ONU, etc.

- El usuario puede definir sus propias abreviaciones para expresiones que se repitan múltiples veces en la bibliografía; por ejemplo, para los nombres de las revistas académicas o de las editoriales. Las abreviaciones se definen en la forma `@string{ ... = " ... "}`. La abreviación debe contener solamente letras o números y el primer símbolo debe ser una letra. Ejemplos:

```
@string{sv = "Springer Verlag"}
```

```
@string{jmb = "Journal of Molecular Biology"}
```

Es buena idea colocar todas las definiciones `@string{...}` al comienzo del archivo ‘‘.bib’’. Las abreviaciones se pueden usar luego en cualquier campo bibliográfico, sin comillas ni corchetes; por ejemplo, `JOURNAL = jmb`, `PUBLISHER = sv`.

Ejemplo A continuación se presenta el contenido de un archivo ‘‘.bib’’ típico. En las secciones 6.10.2 y 6.10.3 aparece la bibliografía obtenida al procesar esta base datos con los estilos `amsplain`, `alpha` y `achicago`.

```
@BOOK{knuth-alg,
    author = "Donald E. Knuth",
    title = "Fundamental {A}lgorithms",
    publisher = "Addison-Wesley",
    year = "1973"}
```



```
@INBOOK{apostol,
    title = "Mathematical {A}nalysis",
    author = "Tom M. Apostol",
    publisher = "Addison Wesley Publishing Co.",
    edition = "Second",
    year = 1977,
    chapter = "16"}
```



```
@PHDTHESIS{buss,
    author = "Sam Buss",
    title = "Bounded Arithmetic",
    school = "Princeton University",
    month = "July",
    year = "1986",
    note = "to be published by Bibliopolis, Inc."}
```



```
@TECHREPORT{tay-wiles,
    author = "R. Taylor and A. Wiles",
    title = "Ring theoretic properties of certain {H}ecke algebras",
    institution = "Department of Mathematics, Princeton University",
    month = "October",
```

```
year = "1994",
number = "PRIN-M94-56"}
```

```
@INCOLLECTION{perciv,
author = "Percival, Ian",
title = "Integrable and nonintegrable {H}amiltonian systems",
booktitle = "Nonlinear dynamics aspects of particle
accelerators",
pages = "12--36",
series = "Lecture Notes in Phys",
volume = "247",
publisher = "Springer Verlag",
address = "Berlin-New York",
year = "1986"}
```

```
@PROCEEDINGS{coloquio,
title = "Evolution, games and learning. {P}roceedings of the fifth
annual international conference, {L}os {A}lamos, {N.M.}",
editor = "Farmer, Doyne and Lapedes, Alan and Packard, Norman",
year = "1986",
publisher = "North-Holland Publishing Co"}
```

```
@INPROCEEDINGS{complexity,
crossref = "coloquio",
author = "B. A. Huberman and T. Hogg",
title = "Complexity and adaptation",
pages = "376--384"}
```

```
@ARTICLE{bardeen-zumino,
author = "Bardeen, William and Zumino, Bruno",
title = "Consistent and covariant anomalies in gauge and
gravitational theories",
year = "1984",
journal = "Nuclear-Phys. B",
volume = "224",
number = "2",
pages = "421--453"}
```

```
@ARTICLE{godel,
author = "Kurt G\"odel",
title = "On formally undecidable propositions of
\emph{Principia Mathematica} and related systems {I}",
journal = "Mon. f\"ur Math. und Physik",
volume = "38",
pages = "173--198",
year = "1931"}
```

6.10.2. Estilos bibliográficos

El *estilo* representa el aspecto final de la lista bibliográfica. Se escoge con la instrucción `\bibliographystyle{estilo}`. Los siguientes son los estilos básicos de L^AT_EX (no requieren instalación previa):

plain Estilo “simple”; los items se ordenan alfabéticamente por autores y se enumeran con etiquetas entre paréntesis angulares.

unsrt Los items son listados en el orden de citación, con etiquetas numéricas entre paréntesis angulares.

alpha Los items se ordenan alfabéticamente por autores, con etiquetas formadas por abreviaciones de los nombres de los autores y el año de publicación. Las etiquetas aparecen entre paréntesis angulares.

abbrv Similar al estilo **plain** pero con etiquetas formadas por abreviaciones de los nombres de los autores.

Otros estilos usualmente incluidos en las distribuciones de L^AT_EX 2_ε se presentan a continuación. Para usar el estilo **apalike**, por ejemplo, se escribe `\bibliographystyle{apalike}` después de `\begin{document}`.

acm.bst Estilo de la *Association for Computer Machinery*. Los items se ordenan alfabéticamente por autores, con etiquetas numéricas. Los nombres de los autores aparecen en versalitas.

amsplain.bst Estilo de la *American Mathematical Society*, estándar en la literatura matemática. Los items se ordenan alfabéticamente por autores, con etiquetas numéricas.

amsalpha.bst Similar al estilo **amsplain** pero con etiquetas formadas por abreviaciones de los nombres de los autores y el año de publicación.

apalike.bst Estilo similar al de la *American Psychology Association*. Los items se ordenan alfabéticamente por autores, con etiquetas formadas por los apellidos completos y el año de publicación.

siam.bst Estilo de la *Society for Industrial and Applied Mathematics*. Similar al estilo **amsplain** pero con iniciales de los nombres y los apellidos en versalitas.

 Los estilos propiamente dichos son archivos con extensión **.bst** que se colocan en el subdirectorío `/texmf/bibtex/bst` o equivalente. Hay literalmente decenas de estilos, creados por diversas instituciones académicas y editoriales comerciales. En la carpeta `/BibTeX/` del CD adjunto se incluyen más de 150 estilos bibliográficos, con numerosas ejemplos de bases de datos `'—.bib'`.

Ejemplo La base de datos de la página 210, en el estilo `amsplain`.

- [1] Tom M. Apostol, *Mathematical Analysis*, second ed., ch. 16, Addison Wesley Publishing Co., 1977.
- [2] William Bardeen and Bruno Zumino, *Consistent and covariant anomalies in gauge and gravitational theories*, Nuclear-Phys. B, **224** (1984), no. 2, 421–453.
- [3] Sam Buss, *Bounded Arithmetic*, Ph.D. thesis, Princeton University, July 1986, to be published by Bibliopolis, Inc.
- [4] Doyne Farmer, Alan Lapedes, and Norman Packard (eds.), *Evolution, games and learning. Proceedings of the fifth annual international conference, Los Alamos, N.M.*, North-Holland Publishing Co, 1986.
- [5] Kurt Gödel, *On formally undecidable propositions of Principia Mathematica and related systems I*, Mon. für Math. und Physik, **38** (1931), 173–198.
- [6] B. A. Huberman and T. Hogg, *Complexity and adaptation*, In Farmer et al. [4], pp. 376–384.
- [7] Donald E. Knuth, *Fundamental Algorithms*, Addison-Wesley, 1973.
- [8] Ian Percival, *Integrable and nonintegrable Hamiltonian systems*, Non-linear dynamics aspects of particle accelerators, Lecture Notes in Phys, vol. 247, Springer Verlag, Berlin-New York, 1986 pp. 12–36.
- [9] R. Taylor and A. Wiles, *Ring theoretic properties of certain Hecke algebras*, Tech. Report PRIN-M94-56, Department of Mathematics, Princeton University, October 1994.

Ejemplo La base de datos de la página 210, en el estilo `alpha`.

- [Apo77] Tom M. Apostol. *Mathematical Analysis*, chapter 16. Addison Wesley Publishing Co., second edition, 1977.
- [Bus86] Sam Buss. *Bounded Arithmetic*. PhD thesis, Princeton University, July 1986. To be published by Bibliopolis, Inc.
- [BZ84] William Bardeen and Bruno Zumino. Consistent and covariant anomalies in gauge and gravitational theories. *Nuclear-Phys. B*, **224**(2):421–453, 1984.
- [FLP86] Doyne Farmer, Alan Lapedes, and Norman Packard, editors. *Evolution, games and learning. Proceedings of the fifth annual international conference, Los Alamos, N.M.* North-Holland Publishing Co, 1986.

- [Göd31] Kurt Gödel. On formally undecidable propositions of *Principia Mathematica* and related systems I. *Mon. für Math. und Physik*, 38:173–198, 1931.
- [HH86] B. A. Huberman and T. Hogg. Complexity and adaptation. In Farmer et al. [Farmer et al., 1986], pages 376–384.
- [Knu73] Donald E. Knuth. *Fundamental Algorithms*. Addison-Wesley, 1973.
- [Per86] Ian Percival. Integrable and nonintegrable Hamiltonian systems. In *Nonlinear dynamics aspects of particle accelerators*, volume 247 of *Lecture Notes in Phys.*, pages 12–36. Springer Verlag, Berlin-New York, 1986.
- [TW94] R. Taylor and A. Wiles. Ring theoretic properties of certain Hecke algebras. Technical Report PRIN-M94-56, Department of Mathematics, Princeton University, October 1994.

6.10.3. Estilos bibliográficos autor-año

Con los estilos bibliográficos mencionados en la sección anterior, las citas internas y la bibliografía aparecen con etiquetas encerradas entre paréntesis angulares. Este esquema difiere del que se usa corrientemente en las ciencias naturales y en las ciencias humanas, en las que predomina el denominado “estilo autor-año”: no hay etiquetas en la lista final de publicaciones y las citas internas se hacen sin paréntesis o con paréntesis circulares; por ejemplo, (Scott, 1992) o Scott (1992).

Hay algunos paquetes especiales que permiten citas y bibliografía en el más tradicional estilo autor-año. Entre ellos mencionamos los paquetes `natbib`, `achicago` y `harvard`⁹, los cuales redefinen el comando `\cite`, o añaden versiones extendidas de él, y modifican el entorno `thebibliography` para eliminar las etiquetas.

A continuación describiremos brevemente el paquete `achicago`, inspirado en las normas del *Chicago Manual of Style*. Para acceder a este paquete se escribe `\usepackage{achicago}` en el preámbulo del documento.

Con `achicago`, el comando `\cite` de `LATEX` mantiene su funcionalidad, incluyendo las citas múltiples y las anotaciones (véase la sección 3.26), excepto que se obtienen paréntesis circulares en lugar de los angulares de `LATEX`. Se dispone, además, de nuevas versiones de `\cite` para citar los autores de las publicaciones, sin paréntesis o sin año de publicación. En el siguiente ejemplo se presentan dichos comandos.

⁹Estos paquetes también se incluyen en el CD adjunto; `natbib` es compatible con todos los estilos bibliográficos pero su descripción detallada excede los límites del presente libro.

Ejemplo El comando `\cite` y sus extensiones en el paquete `achicago`. Las claves `godel` y `tay-wiles` corresponden a la base de datos de la página 210.

<code>\cite[tay-wiles]</code>	⇒ (Taylor and Wiles 1994)
<code>\cite[page=12]{tay-wiles}</code>	⇒ (Taylor and Wiles 1994, page 12)
<code>\citeNP{tay-wiles}</code>	⇒ Taylor and Wiles 1994
<code>\citeA{tay-wiles}</code>	⇒ (Taylor and Wiles)
<code>\cite{godel,tay-wiles}</code>	⇒ (Gödel 1931; Taylor and Wiles 1994)
<code>\citeANP{tay-wiles}</code>	⇒ Taylor and Wiles

Para obtener la bibliografía final se procede siguiendo el mismo procedimiento presentado en la página 206, pero el estilo bibliográfico propio de este paquete se invoca con `\bibliographystyle{achicago}`. Con éste, la base de datos de la página 210 adquiere la siguiente presentación:

Apostol, Tom M. 1977. Chapter 16 of *Mathematical Analysis*, Second. Addison Wesley Publishing Co.

Bardeen, William, and Bruno Zumino. 1984. “Consistent and covariant anomalies in gauge and gravitational theories.” *Nuclear-Phys. B* 224 (2): 421–453.

Buss, Sam. 1986, July. “Bounded Arithmetic.” Ph.D. diss., Princeton University. to be published by Bibliopolis, Inc.

Farmer, Doyne, Alan Lapedes, and Norman Packard, eds. 1986. *Evolution, games and learning. Proceedings of the fifth annual international conference, Los Alamos, N.M.* North-Holland Publishing Co.

Gödel, Kurt. 1931. “On formally undecidable propositions of *Principia Mathematica* and related systems I.”, *Mon. für Math. und Physik* 38:173–198.

Huberman, B. A., and T. Hogg. 1986. “Complexity and adaptation.” In Farmer, Lapedes, and Packard 1986, 376–384.

Knuth, Donald E. 1973. *Fundamental Algorithms*. Addison-Wesley.

Percival, Ian. 1986. “Integrable and nonintegrable Hamiltonian systems.” In *Nonlinear dynamics aspects of particle accelerators*, Volume 247 of *Lecture Notes in Phys*, 12–36. Berlin-New York: Springer Verlag.

Taylor, R., and A. Wiles. 1994, October. “Ring theoretic properties of certain Hecke algebras.” Technical Report PRIN-M94-56, Department of Mathematics, Princeton University.



El ambiente gráfico de LATEX 2 ε

Los paquetes **color**, **graphics** y **graphicx** conforman el ambiente gráfico de LATEX 2 ε ¹. El primero permite definir y usar colores en documentos (sección 7.2) mientras que **graphics** y **graphicx** permiten rotación, reflexión y aumento a escala de objetos, así como la inclusión de gráficas elaboradas con software externo (sección 7.3). Pero estos paquetes no dependen solamente de T_EX o LATEX sino que requieren del soporte o “cooperación” de los manejadores (en inglés, *drivers*) de pantalla e impresora. Tales manejadores son los encargados de convertir el archivo ‘**—.dvi**’ en lo que finalmente vemos en la pantalla o leemos impreso en el papel. Desafortunadamente, no todos los manejadores tienen soporte completo para los paquetes del ambiente gráfico de LATEX 2 ε . Algunos visualizadores, por ejemplo, son incapaces de mostrar los efectos de rotación o aumento a escala.

Al cargar los paquetes **color**, **graphics** y **graphicx**, hay que indicar el tipo de manejador utilizado por la implementación LATEX presente en el computador local. Por ejemplo, **color** y **graphicx** se pueden cargar simultáneamente en la forma

```
\usepackage[manejador]{color,graphicx}
```

En la Tabla 7.1 aparecen algunos de los nombres permitidos para la opción **manejador**; los manejadores allí mencionados brindan soporte completo a los tres paquetes. El usuario debe chequear de antemano la instalación local y utilizar la opción adecuada. Así por ejemplo, con el manejador **dvips**, que se emplea en muchas implementaciones y en múltiples plataformas computacionales², hay que escribir, en el preámbulo del documento,

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

para cargar el paquete **graphics**. De manera análoga se accede a los paquetes **color** y **graphicx**.

¹LATEX posee además el entorno **picture** para la elaboración de gráficas sencillas; se describe en la sección 7.6.

²Para información adicional sobre **dvips**, véase la sección 7.1.

Implementación	Opción utilizada
UNIX, Linux, <i>Windows</i> , etc.	[dvips]
Y&Y	[dvipsone]
Y&Y <i>Windows</i>	[dviwindo]
PCTeX32	[pctex32] o [dvips]
OzTeX (Macintosh)	[oztex] o [dvips]
<i>Textures</i> (Macintosh)	[textures]

TABLA 7.1. Algunos manejadores con soporte completo para los paquetes `color`, `graphics` y `graphicx`.

7.1. PostScript y dvips

El lenguaje de programación PostScript, desarrollado por Adobe Systems, fue especialmente diseñado para la creación, manipulación e impresión de objetos gráficos. La mayoría de los manejadores para los paquetes `color`, `graphics` y `graphicx` de LATEX 2 ε utilizan precisamente PostScript para realizar todas sus manipulaciones gráficas. Originalmente cada manejador tenía sus propios paquetes, con su propia sintaxis, para implementar estas operaciones. LATEX 2 ε ha unificado esa situación caótica con comandos estándares de alto nivel, los cuales se describirán a lo largo del presente capítulo.

Entre los convertidores corrientemente utilizados para pasar del formato `dvi` al formato `ps` de PostScript sobresale el programa de dominio público `dvips`³, disponible para prácticamente todas las plataformas computacionales. `dvips` se puede usar desde la línea de comandos, en la forma

```
dvips doc.dvi
```

para convertir un documento dado `doc.dvi` al formato `ps`. En algunas implementaciones, como WinEdt, la ejecución de `dvips` se hace con un simple *click* sobre el botón `dvi~>ps`.

El uso de `dvips` (en general, de los traductores de `dvi` a `ps`) también permite el acceso a fuentes PostScript para reemplazar las fuentes estándares de LATEX. Este tópico se describe en el Capítulo 9. Y un fruto más de la fecunda asistencia de PostScript a LATEX es el poderoso paquete gráfico **PSTricks**, descrito detalladamente en el Capítulo 11.

Otra de las herramientas gratuitas disponibles en INTERNET es el interpretador **Ghostscript**, que permite ver e imprimir archivos `ps`, y convertirlos a otros formatos. La interfaz de **Ghostscript** es muy primitiva por

³El programa `dvips`, escrito por Tomas Rokicki, también se distribuye con un extenso manual, `dvips.dvi`, para quienes deseen conocer y utilizar todas sus facetas.

lo cual son preferibles interfaces interactivas más cómodas; entre aquéllas de dominio público destacamos:

GSview	Para <i>Windows</i> y OS/2
Ghostview	Para Unix X Windows
Mac GS Viewer	Para <i>Macintosh</i>
BMV	Para Linux

Estas interfaces requieren que Ghostscript esté previamente instalado.

 El CD adjunto incluye la versión 7.04 de Ghostscript (instalador **gs704w32.exe**) y la versión 4.3 de GSview (instalador **gsv43w32.exe**) para *Windows* y OS/2. La página Web de Ghostscript, Ghostview y GSview es:

<http://www.cs.wisc.edu/~ghost/>

de donde se pueden descargar las últimas versiones disponibles.

7.2. El paquete color

El paquete **color**, al cual se accede en la forma indicada al comienzo de este capítulo, tiene los siguientes colores pre-definidos:

black	(negro)	green	(verde)
white	(blanco)	blue	(azul oscuro)
yellow	(amarillo)	cyan	(azul claro)
red	(rojo)	magenta	(rosado)

7.2.1. Definición de nuevos colores

LATEX admite la definición de nuevos colores con el comando **\definecolor**, el cual es compatible con las más conocidas representaciones de color.

Modelo cmyk. Un color se especifica asignándole valores entre 0 y 1 a los colores básicos *cyan*, *magenta*, *yellow* y *black*. Para definir colores con este modelo se utiliza la instrucción

\definecolor{nombre}{cmyk}{n₁, n₂, n₃, n₄}

donde $0 \leq n_1, n_2, n_3, n_4 \leq 1$ y el **nombre** es escogido por el usuario.

Modelo rgb. Un color se define asignándole valores entre 0 y 1 a los colores básicos *red*, *green* y *blue*, por medio de la instrucción

\definecolor{nombre}{rgb}{n₁, n₂, n₃}

donde $0 \leq n_1, n_2, n_3 \leq 1$ y el **nombre** es escogido por el usuario.

Modelo gray. Se usa para definir tonos de gris, en la forma

```
\definecolor{nombre}{gray}{n}
```

donde $0 \leq n \leq 1$. En este modelo 0 representa el negro y 1 el blanco.

En la Tabla 7.2 ofrecemos una paleta de nuevos colores, en el modelo cmyk; el lector puede utilizarla como guía para la definición de otros matices⁴. El preámbulo del documento es el sitio más apropiado para colocar las definiciones de nuevos colores.

```
\definecolor{dorado}{cmyk}{0,0.10,0.84,0}
\definecolor{melon}{cmyk}{0,0.29,0.84,0}
\definecolor{naranja}{cmyk}{0,0.42,1,0}
\definecolor{durazno}{cmyk}{0,0.46,0.50,0}
\definecolor{fresa}{cmyk}{0,1,0.50,0}
\definecolor{ladrillo}{cmyk}{0,0.77,0.87,0}
\definecolor{violeta}{cmyk}{0.07,0.90,0,0.34}
\definecolor{purpura}{cmyk}{0.45,0.86,0,0}
\definecolor{aguamarina}{cmyk}{0.85,0,0.33,0}
\definecolor{esmeralda}{cmyk}{0.91,0,0.88,0.12}
\definecolor{pino}{cmyk}{0.92,0,0.59,0.25}
\definecolor{oliva}{cmyk}{0.64,0,0.95,0.40}
\definecolor{canela}{cmyk}{0.14,0.42,0.56,0}
\definecolor{cafe}{cmyk}{0,0.81,1,0.60}
\definecolor{marron}{cmyk}{0,0.72,1,0.45}
\definecolor{gris-claro}{cmyk}{0,0,0,0.30}
\definecolor{gris-oscuro}{cmyk}{0,0,0,0.50}
```

TABLA 7.2. Comandos para la definición de algunos colores nuevos.

7.2.2. Texto y cajas en color

Para escribir *texto* en un color previamente definido, por ejemplo, *color*, L^AT_EX tiene una declaración global:

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

y un comando con dos argumentos:

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

Ejemplo En la siguiente frase se utiliza el color gris-oscuro definido en la Tabla 7.2.

⁴La Tabla 7.2 está basada en los 68 colores internamente pre-definidos por el manejador dvips. La lista completa de dichos colores (con nombres en inglés) aparece en el archivo *dvipsnam.def*, distribuido con L^AT_EX 2_ε.

`\textcolor{gris-oscuro}{Frase tonta escrita en color gris oscuro.}`

Hay dos comandos similares a `\fbox` (sección 3.15.1). para escribir texto sobre fondos a color. La instrucción

`\colorbox{color de fondo}{texto}`

coloca el *texto* sobre una caja que tiene el *color de fondo* especificado.

La instrucción

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

coloca el *texto* sobre una caja que tiene el *color de fondo* especificado y cuyos bordes tienen el *color del borde* declarado en el primer argumento. Los parámetros `\fboxrule` y `\fboxsep` (página 55) se pueden utilizar para modificar el grosor de las líneas y la distancia entre el borde y el *texto* interior. En ambos casos, el *texto* mismo puede, a su vez, estar en un color específico, con el comando `\textcolor{...}{...}` ya mencionado.

El lector debe tener presente que el argumento *texto* en los comandos `\colorbox` y `\fcolorbox` es escrito por LATEX en modo ID (sección 2.1).

También se puede hacer que toda la página adquiera un *color de fondo* determinado, con la instrucción

`\pagecolor{color de fondo}`

Esta es una declaración global y se aplica a la página actual y a las subsiguientes hasta que sea actualizada por otra declaración `\pagecolor`.

Ejemplo

“La felicidad tiene un transfondo gris”. Mafalda.

```
\colorbox{gris-claro}{“La felicidad tiene un transfondo gris”}.
Mafalda.}
```

Ejemplo Aquí usamos algunos de los colores definidos en la Tabla 7.2.

`jEureka!`

```
\colorbox{blue}{\textcolor{white}{!`Eureka!`}}
```

`No fume`

```
\setlength{\fboxrule}{2pt}
\fcolorbox{red}{dorado}{\textcolor{red}{%
\large No fume}}}
```

 NO FUME

```
\setlength{\fboxrule}{2pt}
\setlength{\fboxsep}{6pt}
\fcolorbox{cafe}{pino}{%
\fcolorbox{cafe}{pino}{%
\textcolor{yellow}{\large \bf NO FUME}}}}
```

7.3. Los paquetes *graphics* y *graphicx*

El paquete “estándar” *graphics* y su versión “extendida” *graphicx*, que hacen parte de L^AT_EX 2 _{ϵ} , definen comandos para realizar transformaciones geométricas de objetos (aumento a escala, rotación y reflexión), así como para la inclusión en documentos L^AT_EX de gráficas elaboradas con software externo. El paquete *graphicx* es casi idéntico a *graphics*, con la adición de algunos parámetros a ciertos comandos. Ambos paquetes requieren el soporte externo de los manejadores de pantalla e impresora, como se explicó al comienzo del presente capítulo. El usuario debe chequear de antemano la instalación local para cargar los paquetes con la opción adecuada (véase la Tabla 7.1). Por ejemplo, en las implementaciones que utilicen el manejador dvips se accede a *graphicx* por medio de `\usepackage[dvips]{graphicx}`.

En las secciones 7.3.1 a 7.3.3 describimos los comandos⁵:

```
\scalebox
\resizebox
\reflectbox
\rotatebox
```

cuyo argumento principal es un *objeto*, para magnificar, reflejar o rotar, según el caso. El lector debe tener presente que L^AT_EX procesa tal *objeto* en modo ID (sección 2.1), pero se pueden usar objetos más complejos, como tablas o gráficas, encerrándolos primero en cajas (`\mbox`, `\fbox`, etc).

7.3.1. Aumento a escala de objetos

El comando

`\scalebox{factor}{objeto}`

aumenta el tamaño del *objeto* en el *factor* especificado; si éste es menor que 1, el objeto se reduce en tamaño.

Ejemplo

Fórmula de Euler: $e^{i\pi} + 1 = 0$

Fórmula de Euler: $e^{i\pi} + 1 = 0$ (tamaño normal)

Fórmula de Euler: $e^{i\pi} + 1 = 0$

```
\scalebox{0.5}{Fórmula de Euler: $e^{i\pi}+1=0$}\par
```

```
Fórmula de Euler: $e^{i\pi}+1=0$ (tamaño normal) \par\smallskip
```

```
\scalebox{1.5}{Fórmula de Euler: $e^{i\pi}+1=0$}
```

⁵ A menos que se indique explícitamente lo contrario, los comandos descritos son válidos en los dos paquetes *graphics* y *graphicx*.

El comando `\scalebox` tiene un argumento opcional que permite aumentar la escala vertical de manera independiente:

```
\scalebox{factor}[escala vertical]{objeto}
```

Ejemplo

Fórmula de Euler: $e^{i\pi} + 1 = 0$

```
\scalebox{1.2}{2}{Fórmula de Euler: $e^{i\pi}+1=0$}
```

Un comando similar a `\scalebox` es

```
\resizebox{longitud horizontal}{longitud vertical}{objeto}
```

con el cual el *objeto* adquiere las dimensiones especificadas en los argumentos *longitud horizontal* y *longitud vertical*. Puesto que estas dimensiones se escriben de manera independiente, el objeto se puede distorsionar o deformar, pero si se desea mantener la razón ancho/alto del objeto original basta usar `{!}` como uno de los argumentos.

Ejemplo Aquí el objeto se distorsiona porque el cambio en las dimensiones horizontal y vertical no es proporcional.

Fórmula de Euler: $e^{i\pi} + 1 = 0$

```
\resizebox{11cm}{0.4cm}{Fórmula de Euler: $e^{i\pi}+1=0$}
```

Ejemplo El objeto mantiene la razón ancho/alto original debido al uso de `{!}` en el segundo argumento.

Fórmula de Euler: $e^{i\pi} + 1 = 0$

```
\resizebox{8cm}{!}{Fórmula de Euler: $e^{i\pi}+1=0$}
```

Al procesar los comandos `\scalebox` y `\resizebox`, LATEX calcula y almacena cuatro dimensiones del *objeto* dado: `\width`, `\height`, `\depth` y `\totalheight` (que son las mismas mencionadas en la Tabla 3.5, página 51). Estas dimensiones, o factores de ellas, se pueden usar en los argumentos de `\scalebox` o `\resizebox`.

Ejemplo Aquí el objeto adquiere 3 veces su altura original, $3\height$, pero mantiene su anchura (`\width`).

Fórmula de Euler: $e^{i\pi} + 1 = 0$

```
\resizebox{\width}{3\height}{Fórmula de Euler: $e^{i\pi}+1=0$}
```

7.3.2. Reflexión de objetos

El comando

```
\reflectbox{objeto}
```

produce una imagen especular (“reflejada”) del *objeto* dado.

Ejemplo

`LATEX XETAI`

```
{\Large \LaTeX} \reflectbox{\Large \LaTeX}
```

7.3.3. Rotación de objetos

Cualquier *objeto* `LATEX` se puede rotar un *ángulo* determinado (entre -360 y 360 grados) usando el comando

```
\rotatebox{ángulo}{objeto}
```

La rotación se hace en el sentido anti-horario si el ángulo es positivo, y en el sentido contrario si es negativo.

Ejemplo

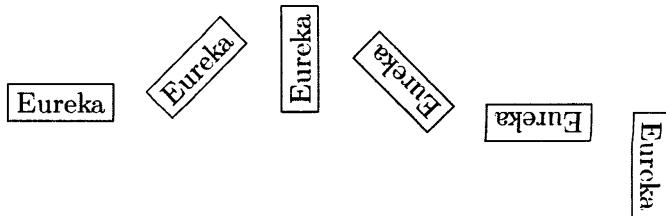
Adagio chino:

Hay libros que son como montañas: ¡no se leen sino se escalan!

Hay libros que son como montañas: `\rotatebox{20}{!`no se leen sino se escalan!`}`

Ejemplo En este ejemplo, todos los `\fbox` rotados están colocados en el mismo renglón, separados entre sí una distancia `\quad`.

Angulo: 0° 45° 90° 135° 180° 270°



```
\fbox{Eureka}\quad
\rotatebox{45}{\fbox{Eureka}}\quad
\rotatebox{90}{\fbox{Eureka}}\quad
\rotatebox{135}{\fbox{Eureka}}\quad
\rotatebox{180}{\fbox{Eureka}}\quad
\rotatebox{270}{\fbox{Eureka}}
```

El comando `\rotatebox` tiene un argumento opcional que permite escoger el *origen*, es decir, el punto con respecto al cual se hace la rotación:

```
\rotatebox[origin=...]{ángulo}{objeto}
```

Los valores permitidos para el origen de rotación aparecen en la Figura 7.1.

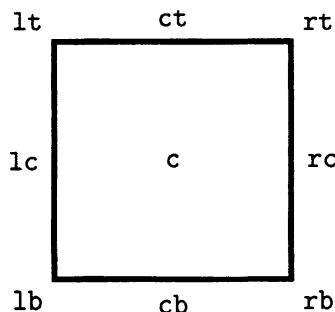
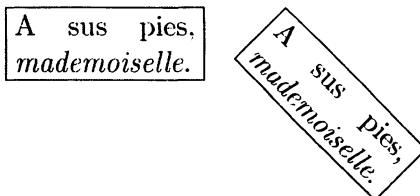


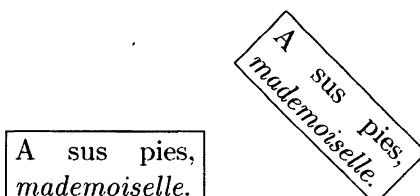
FIGURA 7.1. Puntos de rotación para la opción `origin` de `\rotatebox`.

Ejemplo El objeto

`\fbox{\parbox{2.4cm}{A sus pies, \emph{mademoiselle}.}}` se rota -45° , primero alrededor del punto `lt` y luego alrededor de `rb`.



```
\fbox{\parbox{2.4cm}{A sus pies, \emph{mademoiselle}.}}\quad
\rotatebox[origin=lt]{-45}{\fbox{\parbox{2.4cm}{A sus pies,
\emph{mademoiselle}.}}}
```



```
\fbox{\parbox{2.4cm}{A sus pies, \emph{mademoiselle}.}}\quad
\rotatebox[origin=rb]{-45}{\fbox{\parbox{2.4cm}{A sus pies,
\emph{mademoiselle}.}}}
```

Ejemplo \rotatebox puede ser útil para rotar tablas enteras, o casillas individuales, tal como se aprecia en este ejemplo.

Trabajos publicados	F. Viète	S. Stevin	J. Kepler	R. Descartes	P. Fermat	B. Pascal
Algebra	✓	✓	✓	✓		✓
Geometría	✓		✓	✓	✓	✓
Mecánica		✓		✓		
Probabilidad					✓	✓

```
\begin{center}
\newcommand{\ch}{\checkmark}
\renewcommand{\arraystretch}{1.3}
\begin{tabular}{l*{6}{c}}
\raisebox{0.5cm}{\parbox{1.6cm}{\sffamily Trabajos\publicados}} & \rotatebox{60}{F. Vi\'ete} & \rotatebox{60}{S. Stevin} & \rotatebox{60}{J. Kepler} & \rotatebox{60}{R. Descartes} & \rotatebox{60}{P. Fermat} & \rotatebox{60}{B. Pascal} \\
Algebra & \ch & \ch & \ch & \ch & \\ 
Geometría & \ch & & \ch & \ch & \ch & \\
Mecánica & & \ch & & \ch & & \\
Probabilidad & & & & \ch & & \\
\end{tabular}
\end{center}
```

Ejemplo En este ejemplo se combinan los efectos de \rotatebox y \resizebox.



```
\resizebox{2\width}{0.8\height}{\rotatebox{30}{\fbox{%
\parbox{5.5cm}{La búsqueda de la verdad es más fascinante que
su posesión. \rightline{Gotthold Lessing}}}}}}
```

7.3.4. Inclusión de gráficas externas en documentos LATEX

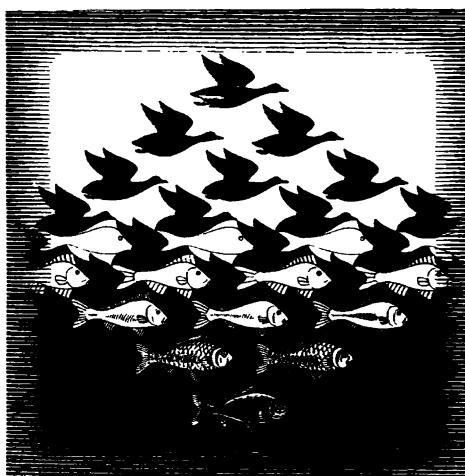
Describiremos a continuación las herramientas disponibles en el paquete **graphicx** para la inclusión en documentos LATEX de gráficas externas. El paquete **graphicx** es más apropiado que su homólogo **graphics** para esta función debido a las opciones adicionales que posee. Se supone, en lo que sigue, que **graphicx** ha sido invocado, junto con el manejador adecuado para la implementación local, en la forma indicada al comienzo de este capítulo.

Para importar una *gráfica* en un documento LATEX escribimos:

```
\includegraphics[opciones]{gráfica}
```

Las *opciones* disponibles se presentan en la Tabla 7.3. Si la *gráfica* no está ubicada en el mismo sitio que el documento fuente, se debe indicar su ruta completa, con la sintaxis del sistema operativo. Dependiendo del sistema operativo local y de los manejadores disponibles, la gráfica puede estar en cualquiera de los formatos BMP, PCX, GIF, JPG, etc, pero el formato más adecuado es EPS (Encapsulated PostScript) que incluye internamente la especificación del tamaño real de la imagen, dato requerido por LATEX para importarla⁶.

Ejemplo El archivo gráfico **Escher1.eps** está ubicado en el subdirectorio **c:/Imagenes**.



```
\begin{center}
\includegraphics{c:/Imagenes/Escher1.eps}
\end{center}
```

⁶Con el manejador **dvips**, en sistemas operativos que utilizan *pipes* (como UNIX), LATEX puede importar gráficas en muchos formatos diferentes de EPS, tales como TIF, GIF, JPG, PIC, etc.

Cuando la gráfica no está en el formato EPS, L^AT_EX es incapaz de determinar su tamaño y emite un mensaje de error como el siguiente:

! LaTeX Error: Cannot determine size of graphic in /.../...
(no BoundingBox).

En tales casos, el usuario puede convertir la imagen al formato EPS, recurriendo a uno de los muchos programas gráficos y convertidores disponibles; ésta es la alternativa más recomendable. Al respecto hay que advertir que algunos programas gráficos producen archivos EPS con cajas de delimitación (*bounding box*) incorrectas. Se recomienda editar los archivos PS o EPS con los programas GSView o Ghostview (sección 7.1) que permiten escoger la caja de delimitación deseada o determinarla automáticamente⁷. Otra alternativa es usar como guía el tamaño de la imagen en *pixels*, dato que es muy fácil de averiguar usando casi cualquier programa de manejo de gráficas. Si el tamaño de la gráfica es, por ejemplo, 250×420 *pixels*, podemos importarla con la opción `bb=0 0 250 420` (véase la Tabla 7.3).

Ejemplo La gráfica `Escher2.bmp` está en el formato BMP y tiene tamaño 1233×1441 *pixels*. Aquí la importamos usando la caja (*bounding box*) `bb=0 0 196 229`, la cual mantiene la razón alto/ancho de la imagen.



```
\begin{center}
\includegraphics[bb=0 0 196 229]{c:/Imagenes/Escher2.bmp}
\end{center}
```

⁷Este procedimiento se sigue en GSView a través de la opción ‘PS to EPS’ del menú File.

Ubicación de los archivos gráficos. LATEX busca una *gráfica*, insertada con `\includegraphics[opciones]{gráfica}`, en la misma carpeta en la que se encuentra el documento fuente. Si la *gráfica* está ubicada en otro sitio, se debe indicar la ruta completa, con la sintaxis del sistema operativo. Para evitar escribir la ruta de cada gráfica, se pueden indicar los directorios en los que se encuentran almacenadas las gráficas, por medio del comando

\graphicspath{lista de directorios}

Los nombres de los directorios se escriben usando la sintaxis del sistema operativo local y se encierran entre corchetes {}; no hay ningún separador adicional entre los directorios. Se recomienda colocar este comando en el preámbulo del documento; para importar gráficas bastará escribir los nombres de los archivos.

Ejemplo El uso de `\graphicspath` en varias plataformas.

Para Windows, UNIX o Linux:

`\graphicspath{{C:/MisArchivos/Figuras/}{C:/Imgs/FigsEPS/}}`

Para Macintosh:

`\graphicspath{{:MisArchivos:Figuras:}{:Imgs:FigsEPS:}}`

Para VMS:

`\graphicspath{{[.MisArchivos.Figuras]}{[.Imgs.FigsEPS.]}}`

Extensiones de los archivos gráficos. Para evitar escribir las extensiones de los archivos gráficos se puede usar el comando

\DeclareGraphicsExtensions{lista de extensiones}

al comienzo del documento. Las extensiones se escriben con el punto usual y se separan entre sí por comas (véase el ejemplo al final de la página). De esta forma, al importar una gráfica con `\includegraphics`, bastará escribir la raíz del nombre del archivo y LATEX intentará ubicar la gráfica adicionando las extensiones que hayan sido declaradas.

Es importante anotar que si se usa esta declaración, LATEX únicamente buscará gráficas con las extensiones indicadas, independientemente de la configuración de los manejadores de pantalla e impresora locales. Además, los manejadores deben ser compatibles con los formatos gráficos utilizados.

Ejemplo Para declarar las extensiones gráficas EPS, BMP y WMF escribimos, preferiblemente en el preámbulo del documento:

`\DeclareGraphicsExtensions{.eps,.bmp,.wmf}`

bb = <i>a b c d</i>	Caja de delimitación (<i>bounding box</i>) de la imagen, donde <i>a</i> = coordenada <i>x</i> , extremo inferior izquierdo, <i>b</i> = coordenada <i>y</i> , extremo inferior izquierdo, <i>c</i> = coordenada <i>x</i> , extremo superior derecho, <i>d</i> = coordenada <i>y</i> , extremo superior derecho. Es necesario usar esta opción para gráficas en formatos diferentes de EPS. Las unidades que L ^A T _E X asume por defecto en bb =... son los llamados <i>big points</i> (1 pulgada = 72 <i>big points</i>). También podemos usar otras unidades T _E X, como cm o mm.
scale = <i>s</i>	Factor de magnificación o reducción deseado.
angle = <i>α</i>	Angulo de rotación deseado (entre -360 y 360 grados).
origin = <i>p</i>	Se usa en conjunción con angle para escoger el punto alrededor del cual se hace la rotación; <i>p</i> puede tomar los mismos valores que el comando \rotatebox (Figura 7.1).
width = <i>w</i>	Ancho deseado para la imagen. Si se usa width pero no height , la imagen conserva la razón alto/ancho original.
height = <i>h</i>	Altura deseada para la imagen. Si se usa height pero no width , la imagen conserva la razón alto/ancho original.
totalheight = <i>h</i>	Se debe usar en lugar de height cuando la imagen se rota más de 90°, o en el sentido horario (debido a que L ^A T _E X puede “perder” la información height durante la rotación).
keepaspectratio	Mantiene la razón alto/ancho de la imagen, incluso si se usan width y height simultáneamente.
viewport = <i>a b c d</i>	Es similar a la opción bb =... pero se usa para ver sólo la porción de la gráfica encerrada en el rectángulo con vértices opuestos (<i>a, b</i>) y (<i>c, d</i>). Se debe usar junto con la opción clip para impedir que el resto de la imagen se imprima.
trim = <i>a b c d</i>	Reduce la caja de delimitación en las cantidades especificadas.
clip	Se usa en conjunción con viewport = <i>a b c d</i> para “recortar” (<i>clip</i>) de la imagen la parte delimitada por el rectángulo con vértices opuestos (<i>a, b</i>) y (<i>c, d</i>) y omitir el resto de la imagen.
draft	Se obtiene sólo una caja rectangular, del tamaño de la caja de delimitación de la imagen, y marcada con el nombre del archivo gráfico. Muy útil para agilizar el procesamiento del documento fuente en versiones preliminares.

TABLA 7.3. Opciones de `\includegraphics` (paquete `graphicx`).

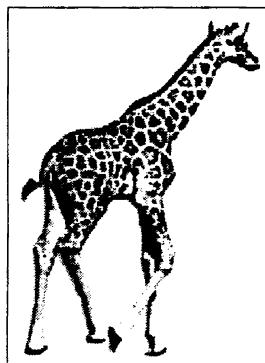
En los ejemplos que siguen ilustramos el empleo de las diferentes opciones de `\includegraphics` (Tabla 7.3) con archivos gráficos BMP y EPS.

Ejemplo En este ejemplo importamos la imagen `Escher3.bmp` utilizando la caja de delimitación (opción obligatoria) `bb=0 0 8cm 7cm`. Para no distorsionar el aspecto de la imagen original, usamos `keepaspectratio`.



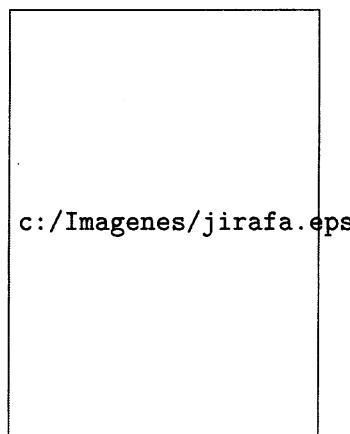
```
\includegraphics[bb=0 0 8cm 7cm,keepaspectratio]{%
c:/Imagenes/Escher3.bmp}
```

Ejemplo La imagen `jirafa.eps` es importada, sujeta a la restricción `height=4.5cm`. La razón alto/ancho del original se preserva. La imagen está centrada y encerrada en una caja `\fbox{...}`.



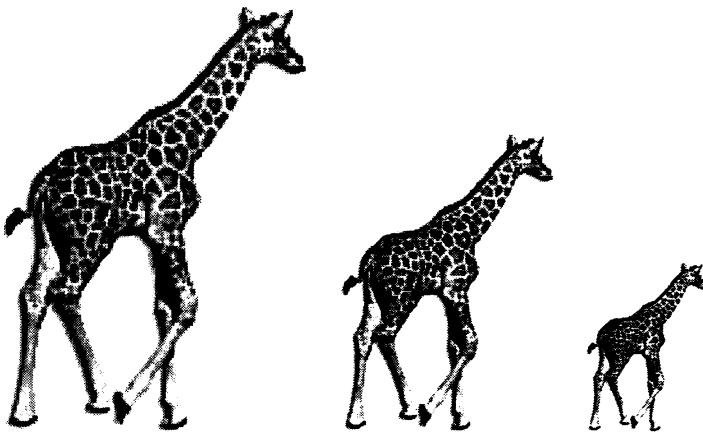
```
\begin{center}
\fbox{%
\includegraphics[height=4.5cm]{c:/Imagenes/jirafa.eps}}
\end{center}
```

Ejemplo Se usa la opción **draft** para importar la imagen **jirafa.eps**. Se obtiene un rectángulo del tamaño real de la imagen y con el nombre del archivo gráfico.



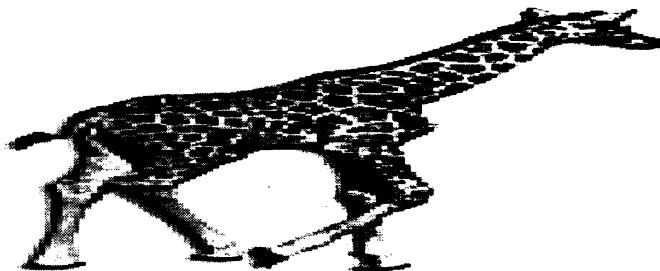
```
\begin{center}
\includegraphics[draft]{c:/Imagenes/jirafa.eps}
\end{center}
```

Ejemplo La imagen **jirafa.eps** aparece a la izquierda en su tamaño original, luego reducida al 70% (**scale=0.7**) y, a la derecha, reducida al 40% de su tamaño (**scale=0.4**).



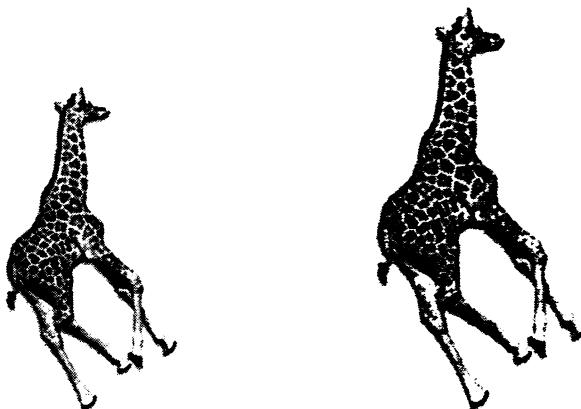
```
\begin{center}
\includegraphics{c:/Imagenes/jirafa.eps}\quad
\includegraphics[scale=0.7]{c:/Imagenes/jirafa.eps}\quad
\includegraphics[scale=0.4]{c:/Imagenes/jirafa.eps}
\end{center}
```

Ejemplo Puesto que se usan los dos parámetros `height` y `width` de manera independiente, la imagen se distorsiona. Sus dimensiones son exactamente las solicitadas: 9 cm de ancho y 3.5 cm de alto.



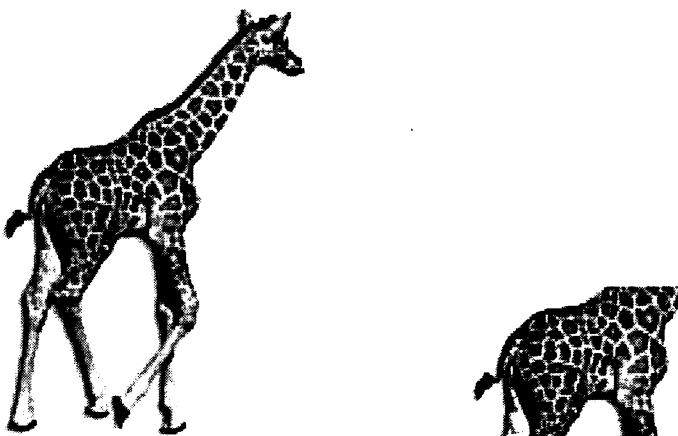
```
\begin{center}
\includegraphics[width=9cm,height=3.5cm]{c:/Imagenes/jirafa.eps}
\end{center}
```

Ejemplo Aquí se utilizan los parámetros `angle=35` y `height=4.5cm`, pero en diferente orden. Obsérvese que los resultados obtenidos no coinciden ya que LATEX implementa las opciones del comando `\includegraphics` en el estricto orden izquierda-derecha. Así, la imagen de la izquierda fue inicialmente rotada 35° y luego la caja rotada fue reducida a una altura de 4.5 cm. Por otro lado, la imagen de la derecha fue obtenida reduciendo inicialmente la altura de la gráfica original a 4.5 cm y luego rotándola 35°.



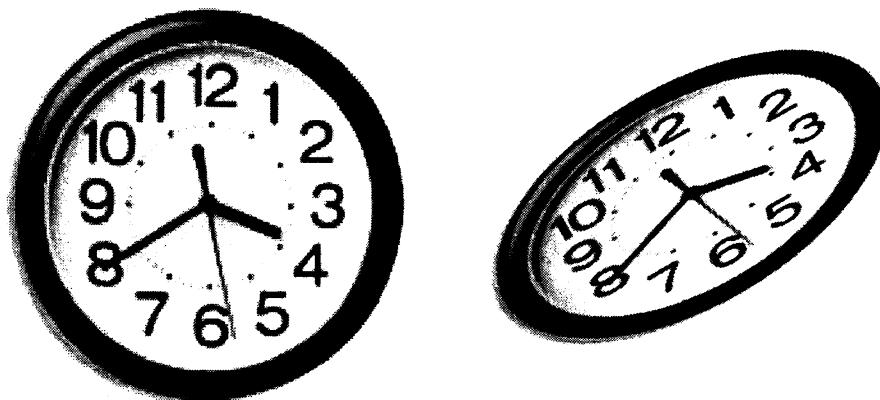
```
\begin{center}
\includegraphics[angle=35,height=4.5cm]{c:/Imagenes/jirafa.eps}
\quad
\includegraphics[height=4.5cm,angle=35]{c:/Imagenes/jirafa.eps}
\end{center}
```

Ejemplo Usando GSView, Ghostview (sección 7.1) o un programa gráfico adecuado, podemos averiguar que la caja de delimitación (*bounding box*) de la imagen `jirafa.eps` es 0 0 117 160. A la derecha se usaron las opciones `viewport=0 60 80 115` y `clip` para “recortar” una porción de la imagen total.



```
\begin{center}
\includegraphics[c:/Imagenes/jirafa.eps]\hspace{2cm}
\includegraphics[viewport=0 60 80 115,clip]{%
c:/Imagenes/jirafa.eps}
\end{center}
```

Ejemplo La imagen `reloj.eps` (gráfica izquierda) se deforma, usando `height` y `width`, y luego se gira 30 grados (gráfica derecha). Con este tipo de manipulaciones se pueden obtener efectos tridimensionales.



```
\includegraphics{moneda.eps}\quad
\includegraphics[width=6cm,height=3cm,angle=30]{reloj.eps}
```

Ejemplo Con `\includegraphics` se pueden insertar gráficas en tablas. En este ejemplo, las gráficas se reducen al 50% de su tamaño original por medio de `scale=0.5`. Se ha usado la opción `b{8cm}` del paquete `array` (sección 5.12) para alinear por la parte inferior el contenido de las filas.

Tres figuras de la mecánica cuántica	
	Max Planck (1858–1947). En 1900 formuló la hipótesis cuántica y dedujo la existencia de una constante universal, hoy llamada constante de Planck, que relaciona energía y frecuencia de radiación.
	Werner Heisenberg (1901–1976). En 1925 desarrolló una de las formulaciones básicas de la mecánica cuántica, la llamada mecánica matricial. Dos años más tarde enunció el Principio de incertidumbre.
	Erwin Schrödinger (1887–1961). En 1926 sentó las bases de la moderna mecánica cuántica ondulatoria, introduciendo la célebre ecuación de onda que lleva su nombre.

```
\begin{tabular}{|c|c|}\hline
\multicolumn{2}{|c|}{\textbf{Tres figuras de la mecánica cuántica}}\\\hline\hline
\includegraphics[scale=0.5]{c:/Imagenes/Planck.eps} & \small
\textbf{Max Planck} (1858--1947). En 1900 formuló la hipótesis ... \\
\includegraphics[scale=0.5]{c:/Imagenes/Heisenberg.eps} & \small
\textbf{Werner Heisenberg} (1901--1976). En 1925 desarrolló ... \\
\includegraphics[scale=0.5]{c:/Imagenes/Schrodinger.eps} & \small
\textbf{Erwin Schrödinger} (1887--1961). En 1926 sentó las bases
... \\ \hline
\end{tabular}
```

7.4. Gráficas y tablas como objetos flotantes

Hemos descrito el entorno `tabular` para tablas (Capítulo 5) y hemos visto cómo se pueden importar gráficas externas con `\includegraphics` (sección 7.3.4). También existen paquetes `TeX` o `LATeX` para el diseño de gráficas; entre ellos mencionamos a **PSTricks** (Capítulo 11), **PICTeX** (Capítulo 12) y el entorno `picture`, propio de `LATeX 2 ϵ` (sección 7.6). Al insertar una de estas tablas o gráficas, en un sitio determinado de un documento, es muy posible que no haya espacio suficiente para acomodarla en lo que queda de la página. En tales casos, `LATeX` deja el resto de la página en blanco, añadiendo espacio vertical entre los párrafos, y coloca el objeto en la página siguiente. El resultado es totalmente inaceptable, en la mayoría de los casos⁸. Para obviar tales situaciones hay que insertar las tablas y las gráficas como *objetos flotantes* (en inglés, *floating objects* o *floats*) utilizando uno de los entornos `table` o `figure`. `LATeX` decide cuál es el sitio más adecuado para colocar el objeto flotante, ya sea en la página actual o en una posterior, y acomoda el texto circundante sin añadir o eliminar espacio en blanco.

El uso de los entornos `table` y `figure` permite, además, la inclusión de rótulos o epígrafes para tablas y gráficas y su numeración automática; esto último se explica en la sección 7.5.

Los entornos `table` y `figure` tienen una sintaxis similar:

<code>\begin{table}[posición] ... Comandos de la tabla ... \end{table}</code>	<code>\begin{figure}[posición] ... Comandos de la gráfica ... \end{figure}</code>
---	---

Los comandos de la tabla propiamente dicha se escriben con el entorno `tabular`. Las gráficas pueden estar construidas con algún paquete gráfico, como los mencionados arriba; también pueden aparecer bajo el alcance del entorno `figure` uno o más comandos `\includegraphics`. La *posición* se especifica con uno de los siguientes cuatro parámetros:

- [t] La parte superior (en inglés, *top*) de una página.
- [b] La parte inferior (en inglés, *bottom*) de una página.
- [h] El sitio en el que aparece escrita la tabla en el documento fuente (en inglés, *here*).

⁸Cuando esto sucede, `TeX` emite una advertencia de la forma: `Underfull \vbox (badness 5040)`, la cual queda consignada en el archivo ‘`—.log`’. El número que sigue a `badness` es un estimación de la gravedad de la situación; su valor máximo es 10.000. Se considera que si `badness > 2.000`, el espacio en blanco entre los párrafos es excesivo. Ningún ‘Underfull’ detiene el procesamiento del documento, pero cada uno de ellos requiere atención personal por parte del usuario.

[p] En una página separada que contiene solamente objetos flotantes.

Podemos utilizar una combinación de estas cuatro opciones para *sugerirle* a LATEX el sitio preferido para la inserción de la tabla o gráfica. Por ejemplo, con la opción de posición [ht], LATEX tratará de colocar el objeto en el sitio exacto donde el entorno **table** o **figure** fue escrito ('here') y, de no encontrar espacio suficiente, la colocará en el tope de la página actual o de una posterior. Si el argumento de posición se omite, LATEX utiliza por defecto el orden de inserción [tbp].

El orden en el que se presentan las opciones de posición no es importante porque LATEX siempre sigue el orden h-t-b-p. Así, las opciones [tbh], [bth] y [htb] son idénticas para LATEX.

Ejemplo Las siguientes instrucciones le sugieren a LATEX insertar una gráfica, importada con \includegraphics, en el siguiente orden de preferencia: aquí (h), tope de una página (t), parte inferior de una página (b).

```
\begin{figure}[htb]
\includegraphics[...]{...}
\end{figure}
```

También existen las versiones estrella **table*** y **figure***:

<pre>\begin{table*}[posición] ... Comandos de la tabla ... \end{table*}</pre>	<pre>\begin{figure*}[posición] ... Comandos de la gráfica ... \end{figure*}</pre>
---	---

Estos entornos se aplican únicamente a páginas con el formato de dos columnas (sección 2.14) e insertan las tablas o figuras a lo ancho de toda la página, y no en columnas individuales. Para documentos en el formato de una sola columna, **table*** y **figure*** funcionan exactamente igual que **table** y **figure**, respectivamente.

7.4.1. Problemas con la inserción de objetos flotantes

Para solucionar posibles problemas relacionados con la colocación de objetos flotantes es importante entender, al menos globalmente, los mecanismos de inserción de LATEX.

- Los entornos **table** y **figure** *siempre* crean objetos flotantes y éstos se imprimen *únicamente* en los sitios permitidos por las opciones de posición especificadas por el usuario. Debemos ser lo suficientemente flexibles para permitirle a LATEX insertar las tablas y gráficas en los

sitios más apropiados. Por tal razón, se recomienda utilizar el mayor número de opciones de posición posibles; por ejemplo, [tbp], [htp]. Hay que evitar usar las opciones individuales [t], [b], [h] y [p]. De hecho, la opción [h] es tan restrictiva y problemática que en las últimas versiones de L^AT_EX se cambia automáticamente por [ht].

- Las figuras flotantes se colocan en el orden en el que aparecen en el documento fuente, es decir, L^AT_EX no coloca una determinada figura si las figuras anteriores no han sido ya colocadas. Similarmente, las tablas flotantes aparecen en el orden de definición. No obstante, las tablas y las gráficas flotantes se insertan en secuencias independientes y L^AT_EX las puede mezclar.
- Si L^AT_EX no puede colocar una determinada gráfica (tabla) flotante en alguno de los sitios solicitados, dicho objeto impide la colocación de las subsiguientes gráficas (tablas) flotantes y ninguna de ellas se imprimirá. Cuando L^AT_EX alcanza el límite máximo de 18 objetos flotantes no procesados emite el mensaje de error “**Too many unprocessed floats**”. Para que un objeto flotante sea impreso lo antes posible y no “flote demasiado” se puede usar el símbolo ! en el argumento de posición. Así por ejemplo, la opción de posición [**!ht**] no sólo le indica a L^AT_EX los sitios de inserción h y t, sino que le recomienda una pronta inserción. El uso de ! es aconsejable cuando hay muchos objetos flotantes en el documento.
- El comando \clearpage (véase la Tabla 3.4) obliga a L^AT_EX a evacuar *todos* los objetos flotantes que aún estén pendientes. Pero hay un precio que pagar: \clearpage inicia una página exactamente en el sitio en el que aparece la instrucción. Por lo tanto, al usar \clearpage debemos saber de antemano el sitio preciso en el que queremos terminar la página actual y comenzar una nueva.
- Cuando L^AT_EX coloca un objeto flotante antes de lo deseado, el usuario puede trasladarlo hacia abajo en el documento fuente o puede ensayar una variante del comando \suppressfloats. Así, la instrucción \suppressfloats[t], colocada inmediatamente antes de una tabla o gráfica flotante, impide que tal objeto flotante se imprima antes del lugar en el que aparece en el documento fuente e impide que objetos flotantes adicionales aparezcan en el *tope* de la página actual.
- La instrucción \suppressfloats[b], colocada inmediatamente antes de una tabla o gráfica flotante, impide que objetos flotantes adicionales aparezcan en la parte inferior de la página actual. Por otro

lado, el comando simple `\suppressfloats`, colocado inmediatamente antes de una tabla o gráfica flotante, impide que objetos flotantes adicionales aparezcan tanto en la parte superior como en la inferior de la página actual.

- Hay que advertir que ninguna de las versiones de `\suppressfloats`, mencionadas en los dos incisos anteriores, afecta a los objetos con opción de posición `h` o con el modificador `!`.
- Para impedir que LATEX inserte los objetos flotantes *antes* del sitio en el que aparecen en el documento fuente se puede usar el paquete `flafter` (cuyo nombre es una abreviación de *floats after*). `flafter` es un paquete estándar de LATEX 2_ε y para activarlo no hay que emitir ningún comando especial, aparte de `\usepackage{flafter}`. Una vez cargado este paquete, el usuario puede estar seguro de que todos los objetos flotantes se insertarán después del sitio en el que aparecen en el documento fuente.
- Los entornos `table` y `figure` no pueden aparecer dentro de cajas como `\fbox`, `\parbox`, `minipage`, etc.

7.4.2. Objetos flotantes horizontalmente centrados

Cuando se usan los entornos `table` o `figure` para insertar tablas o figuras, LATEX agrega un pequeño espacio vertical antes y después del objeto. Como se mencionó en la sección 3.13, LATEX también adiciona espacio vertical al encontrar el entorno `center`. De modo que si se centra una tabla con el entorno `center`, en la forma

```
\begin{center}
\begin{table}
\begin{tabular}{...}
.....
\end{tabular}
\end{table}
\end{center}
```

se añadirá doble espacio vertical: por `center` y por `table`. Para mejores resultados, se recomienda utilizar el comando `\centering` para centrar tablas y figuras:

```
\begin{table}
\centering
\begin{tabular}{...}
.....
\end{tabular}
\end{table}
```

7.5. Epígrafes y numeración de tablas y gráficas

Al usar los entornos `table` y `figure` para insertar tablas o gráficas, podemos también incluir rótulos o epígrafes (en inglés, *captions*) para ellas. La instrucción `\caption{texto}` coloca el *texto* dado como un rótulo para la tabla o gráfica y, además, incrementa en 1 el contador `table` de tablas o el contador `figure` de gráficas, según el caso (sobre el uso de contadores, véase la sección 3.25).

En el entorno `table`, el comando `\caption{texto}` genera un rótulo de la forma ‘Table *n*: *texto*’, y con el entorno `figure` se genera el rótulo ‘Figure *n*: *texto*’. Con el paquete `babel`, opción `spanish`, se obtienen las denominaciones ‘Cuadro’ y ‘Figura’, respectivamente. Si queremos rótulos diferentes (como ‘Tabla’ o ‘Gráfica’) debemos redefinir los parámetros que controlan dichos nombres: `\tablename`, para tablas, y `\figurename`, para gráficas (véase al respecto la sección 6.5).

A un `\caption{...}` cualquiera se le puede asociar (de manera opcional) una *clave*, con `\label{clave}` (sobre el uso de `\label`, véase la sección 3.25). El comando `\label` se debe colocar inmediatamente después de `\caption{...}`. Con claves así definidas, podemos luego hacer referencia a los números de las tablas o gráficas correspondientes, por medio de `\ref`, o las páginas en las que éstas aparecen, por medio de `\pageref`.

Ejemplo El epígrafe de la siguiente tabla se obtuvo con la instrucción `\caption{Datos experimentales}`; la expresión “Tabla 8:” es generada automáticamente por L^AT_EX, siendo 8 el valor actual del contador `table`.

Tabla

Tabla 8: Datos experimentales.

```
\begin{table}[htb]
\centering
\begin{tabular}{...}
:
\end{tabular}
\caption{Datos experimentales.}\label{tdatos}
\end{table}
```

A continuación resaltamos importantes hechos sobre `\caption`:

- El comando `\caption` debe estar bajo el alcance de alguno de los entornos `table` o `figure`; en caso contrario, LATEX emitirá el siguiente mensaje de error: ! LaTeX Error: `\caption outside float`.
- Si para un objeto flotante, definido con `table` o `figure`, no se escribe un `\caption`, la tabla (o figura) en cuestión no se numera y el contador `table` (o `figure`) no se altera.
- El `texto` escrito en `\caption[texto abreviado]{texto completo}` se incluye en el índice de tablas o en el índice de figuras, según corresponda, si se recurre a las instrucciones `\listoftables` o `\listoffigures`, respectivamente (véase la sección 6.7). El comando `\caption` tiene un argumento opcional para simplificar la descripción incluida en el índice. Al escribir

```
\caption[texto abreviado]{texto completo}
```

el `texto completo` aparecerá como epígrafe de la tabla o gráfica, pero en el índice respectivo se leerá el `texto abreviado`. Esto resulta útil en el caso de epígrafes extensos.

- Si el comando `\caption{...}` se escribe inmediatamente antes de `\end{table}` o `\end{figure}`, LATEX colocará el epígrafe debajo de la tabla o gráfica correspondiente, añadiendo un pequeño espacio vertical entre el objeto flotante y su epígrafe. Tal espacio vertical está controlado por el parámetro `\abovecaptionskip`, el cual asume por defecto el valor `10pt`. Por ejemplo, al escribir

```
\begin{figure}
\centering
\includegraphics{grafica.eps}
\caption{Epígrafe debajo de la gráfica.}
\end{figure}
```

se producirá algo como



Figura 12: Epígrafe debajo de la gráfica.

Pero si el comando `\caption{...}` se escribe antes del objeto flotante, por ejemplo,

```
\begin{figure}
  \centering
  \caption{Epígrafe encima de la gráfica}
  \includegraphics{grafica.eps}
\end{figure}
```

LATEX colocará el epígrafe encima de la gráfica, sin añadir espacio vertical entre los dos:

Figura 13: Epígrafe encima de la gráfica.



El espacio vertical que **LATEX** añade después de `\caption` se controla con `\belowcaptionskip`, el cual toma, por defecto, el valor `0pt`. Por consiguiente, si se desean rótulos encima de los objetos hay que redefinir tanto `\abovecaptionskip` como `\belowcaptionskip`, invirtiendo sus valores pre-definidos, lo cual se hace por medio de `\setlength`:

```
\setlength{\abovecaptionskip}{0pt}
\setlength{\belowcaptionskip}{10pt}
```

Estas dos instrucciones se escriben bajo el alcance del entorno `table` o `figure` respectivo, o en el preámbulo del documento si se desean los epígrafes encima de todas las tablas y figuras.

- Dos gráficas (o tablas) pueden compartir un mismo epígrafe si ambas están bajo el alcance del mismo entorno `figure` (o `table`). Por ejemplo, para insertar las gráficas `graf1.eps` y `graf2.eps`, bajo el alcance de un mismo epígrafe y separadas entre sí por 1 cm, escribimos:

```
\begin{figure}
  \includegraphics{graf1.eps}\hspace{1cm}
  \includegraphics{graf2.eps}
  \caption{Epígrafe común para las dos gráficas}
\end{figure}
```

Para tener aún mayor control sobre la alineación vertical de las gráficas, cada `\includegraphics` se puede escribir en una mini-página (con el entorno `minipage`, sección 3.16).

☞ El estilo de los rótulos y los epígrafes (fuentes, tamaño, formato, etc) se puede modificar con el paquete `caption2`, descrito en la sección 13.7.

7.6. El entorno `picture` de LATEX 2 ε

El entorno `picture`, propio de LATEX 2 ε , permite trazar gráficas muy simples, compuestas de texto, líneas rectas, flechas, círculos y curvas cuadráticas de Bézier. Este entorno está implementado en TEX y no requiere soporte especial de los manejadores de pantalla o impresora. Sus capacidades son muy limitadas: sólo puede trazar rectas con un número reducido de pendientes y los diámetros de los círculos no pueden exceder 40 pt (aproximadamente, 1.4 cm). Como referencia, describiremos en esta sección los comandos disponibles en el entorno `picture`, pero para los lectores interesados en ambientes gráficos para LATEX, que sean realmente flexibles y poderosos, recomendamos el uso de los paquetes **PSTricks** (descrito en el Capítulo 11) y **PICTEX** (descrito en el Capítulo 12).

Para construir una gráfica con el entorno `picture` se requiere primero escoger las unidades de medida, lo cual se hace con el comando `\setlength{\unitlength}{1mm}` establece que la unidad de medida es 1 mm. El entorno `picture` funciona mejor con unidades de medida pequeñas, como milímetros o puntos. Cambiando las unidades de medida podemos aumentar o reducir el tamaño de la gráfica pero el grosor de las líneas y el tamaño de las letras no se altera. Para aumento o reducción a escala de *todos* los elementos de una gráfica se deben usar los comandos `\scalebox` o `\resizebox` del paquete `graphicx` (véase la sección 7.3.1).

El entorno `picture` tiene el siguiente aspecto:

```
\begin{picture}(a,b)
  :
  comandos de la gráfica
  :
\end{picture}
```

LATEX reserva un rectángulo invisible con base a y altura b unidades, el cual contiene la gráfica propiamente dicha. El extremo inferior izquierdo del rectángulo tiene coordenadas $(0, 0)$ y el extremo superior derecho tiene coordenadas (a, b) . Todos los elementos de la gráfica se colocan con referencia al sistema coordenado así definido.

El comando `\setlength{\unitlength}{...}`, que establece las unidades de medida, se debe colocar *antes* de `\begin{picture}`.

LATEX puede trazar líneas de dos grosores; con `\thinlines` se obtienen líneas delgadas: —, y con `\thicklines` líneas gruesas: —. Estas declaraciones se pueden usar múltiples veces en una gráfica para cambiar

el grosor de líneas particulares. Por defecto, está vigente `\thinlines`. También se dispone del comando `\linethickness{grosor}` para cambiar el grosor de las líneas, pero tal instrucción afecta solamente a las líneas horizontales y verticales. El *grosor* se debe especificar como una dimensión TEX; por ejemplo, `\linethickness{2pt}` o `\linethickness{1mm}`.

7.6.1. Grillas con el comando `\graphpaper`

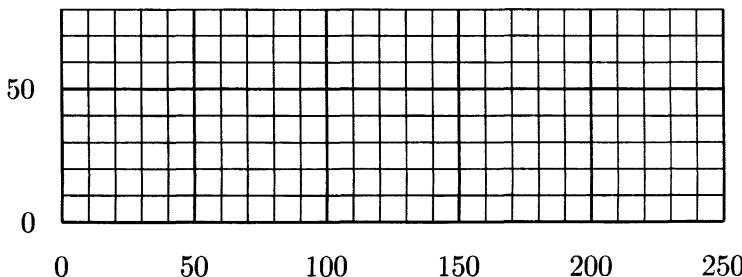
Durante la elaboración de una gráfica es útil tener una grilla de fondo que corresponda al rectángulo escogido con la instrucción `\begin{picture}`. Tal grilla se puede trazar con el comando `\graphpaper`, pero el lector debe tener presente que dicho comando hace parte del paquete `graphpap`, paquete estándar de LATEX 2 ε . Éste debe ser cargado explícitamente, en la forma usual, escribiendo `\usepackage{graphpap}`.

El comando

`\graphpaper [n] (a, b)(c, d)`

traza una grilla con extremos opuestos (a, b) y (c, d) . Los valores a , b , c y d deben ser enteros (positivos o negativos) y se refieren a las unidades definidas con `\unitlength`. Los segmentos que forman la grilla aparecen, por defecto, cada 10 unidades, pero el argumento opcional n permite cambiar tal número.

Ejemplo Con la instrucción `\begin{picture}(250,80)` se ha escogido un rectángulo de dimensiones 250×80 unidades, siendo 1pt el valor de la unidad de medida. Se ha trazado una grilla que encaja exactamente en tal rectángulo.



```
\begin{center}
\setlength{\unitlength}{1pt}
\begin{picture}(250,80)
\graphpaper(0,0)(250,80)
\end{picture}
\end{center}
```

En algunos de los ejemplos que se presentan más adelante utilizaremos, como ayuda visual para el lector, grillas de fondo de color gris, trazadas en la forma `\color{gris}\graphpaper(a,b)(c,d)`. Para esto se requiere haber cargado con anterioridad los paquetes `color` y `graphpaper`, y haber definido el color ‘gris’, lo cual hemos hecho, recurriendo al modelo `cmyk` (sección 7.2.1), en la forma:

```
\definecolor{gris}{cmyk}{0,0,0,0.5}
```

Las grillas se pueden eliminar posteriormente borrando, o comentando con el símbolo %, el comando `\graphpaper`.

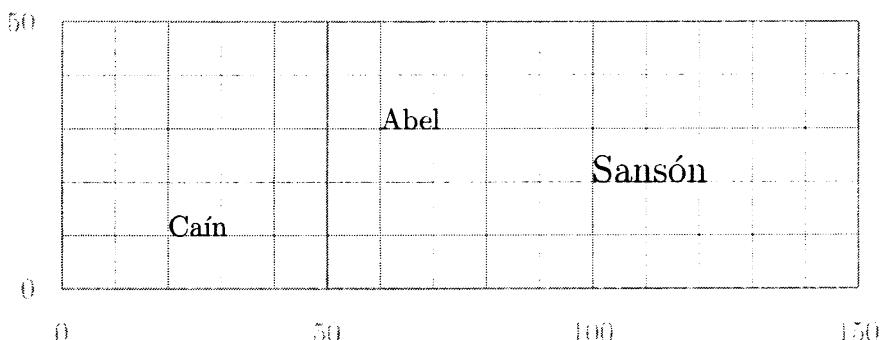
7.6.2. El comando `\put`

Todos los objetos gráficos del entorno `picture`, a excepción de `\qbezier`, se insertan en las gráficas por medio de `\put`. La instrucción

```
\put(x,y){objeto}
```

coloca el *objeto* en el punto (x, y) . Cuando el objeto en cuestión es texto corriente, la caja LATEX que lo contiene se coloca en la gráfica con su extremo inferior izquierdo en el punto (x, y) . La colocación de los demás objetos gráficos se describe en detalle más adelante.

Ejemplo



```
\begin{center}
\setlength{\unitlength}{2pt}
\begin{picture}(150,50)
\color{gris}\graphpaper(0,0)(150,50)
\put(20,10){Caín}
\put(60,30){Abel}
\put(100,20){\Large Sansón}
\end{picture}
\end{center}
```

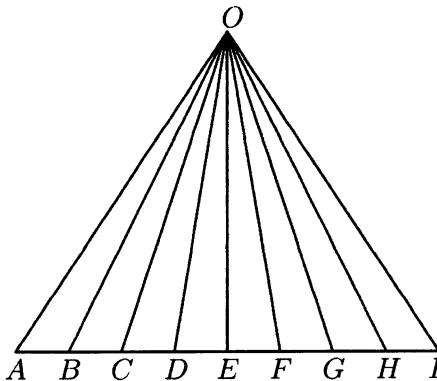
7.6.3. Líneas rectas con `\line`

El comando

```
\put(x,y){\line(a,b){ℓ}}
```

traza una línea recta desde el punto (x, y) , con pendiente b/a , cuya proyección sobre el eje horizontal tiene longitud ℓ unidades ($\ell \cdot \text{\unitlength}$). Este comando tiene las siguientes restricciones: a y b deben ser números enteros tales que $-6 \leq a, b \leq 6$, y con máximo común divisor igual a 1. La longitud ℓ debe ser > 0 . Una línea vertical ($a = 0$) se extiende ℓ unidades hacia arriba o hacia abajo, dependiendo de si b es positivo o negativo.

Ejemplo



```
\begin{center}
\setlength{\unitlength}{2pt}
\begin{picture}(100,60)\thicklines
\put(10,0){\line(2,3){40}} %segmento OA
\put(20,0){\line(1,2){30}} %segmento OB
\put(30,0){\line(1,3){20}} %segmento OC
\put(40,0){\line(1,6){10}} %segmento OD
\put(50,0){\line(0,1){60}} %segmento OE
\put(60,0){\line(-1,6){10}} %segmento OF
\put(70,0){\line(-1,3){20}} %segmento OG
\put(80,0){\line(-1,2){30}} %segmento OH
\put(90,0){\line(-2,3){40}} %segmento OI
\put(10,0){\line(1,0){80}} %segmento AI
\put(8,-5){$A\$} \put(18,-5){$B\$} \put(28,-5){$C\$}
\put(38,-5){$D\$} \put(48,-5){$E\$} \put(58,-5){$F\$}
\put(68,-5){$G\$} \put(78,-5){$H\$} \put(88,-5){$I\$}
\put(49,61){$O\$}
\end{picture}
\end{center}
```

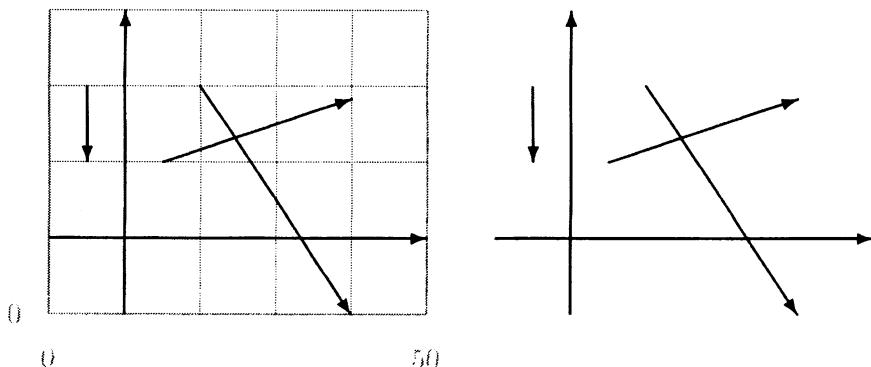
7.6.4. Flechas con \vector

El comando `\vector` es similar a `\line`. Así,

`\put(x,y){\vector(a,b){\ell}}`

traza una flecha (vector) desde (x, y) , con pendiente b/a , cuya proyección sobre el eje horizontal tiene longitud ℓ unidades. El número de pendientes permitidas es menor que para `\line`: a y b deben ser números enteros tales que $-4 \leq a, b \leq 4$, y con máximo común divisor igual a 1. La longitud ℓ debe ser > 0 . Un vector vertical ($a = 0$) se extiende ℓ unidades y apunta hacia arriba o hacia abajo, dependiendo de si b es positivo o negativo.

Ejemplo La gráfica de la izquierda coincide con la de la derecha, excepto por el uso de la grilla `\graphpaper` a la izquierda.



```
\setlength{\unitlength}{1mm}
\begin{picture}(50,40)
{\color{gris}\graphpaper(0,0)(50,40)}
\thicklines
\put(5,30){\vector(0,-1){10}}
\put(0,10){\vector(1,0){50}}
\put(10,0){\vector(0,1){40}}
\put(15,20){\vector(3,1){25}}
\put(20,30){\vector(2,-3){20}}
\end{picture}
```

7.6.5. Círculos con \circle

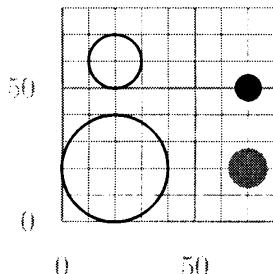
El comando

`\put(x,y){\circle{d}}`

traza un círculo de d unidades de diámetro, centrado en (x, y) . El máximo diámetro permitido por LATEX es 40 pt (1.4 cm aproximadamente). El

comando estrella `\circle*` traza círculos rellenos pero su máximo diámetro permitido es aún menor: sólo 15 pt, poco más de 5 mm.

Ejemplo



```
\setlength{\unitlength}{1pt}
\begin{picture}(80,80)
{\color{gris}\graphpaper(0,0)(80,80)}
\thicklines
\put(20,20){\circle{40}}
\put(20,60){\circle{20}}
\put(70,20){\color{gris}\circle*{15}}
\put(70,50){\circle*{10}}
\end{picture}
```

7.6.6. El comando `\oval`

El comando

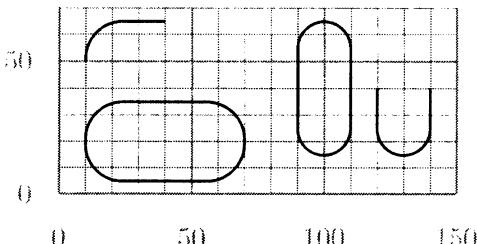
`\put(x,y){\oval(a,b)[porción]}`

traza un rectángulo con vértices redondeados, de a unidades de ancho y b unidades de altura, centrado en el punto (x, y) . El argumento opcional *porción* permite escoger sólo una parte del óvalo de acuerdo con los siguientes valores:

- t** selecciona la porción superior,
- b** selecciona la porción inferior,
- r** selecciona la parte derecha,
- l** selecciona la parte izquierda.

Estas opciones se pueden combinar; así por ejemplo, con **tr** se selecciona la parte superior derecha del óvalo.

Ejemplo



```
\setlength{\unitlength}{1pt}
\begin{picture}(150,70)
{\color{gris}%
\graphpaper(0,0)(150,70)}
\thicklines
\put(40,20){\oval(60,30)}
\put(40,50){\oval(60,30)[lt]}
\put(100,40){\oval(20,50)[b]}
\end{picture}
```

7.6.7. Cajas

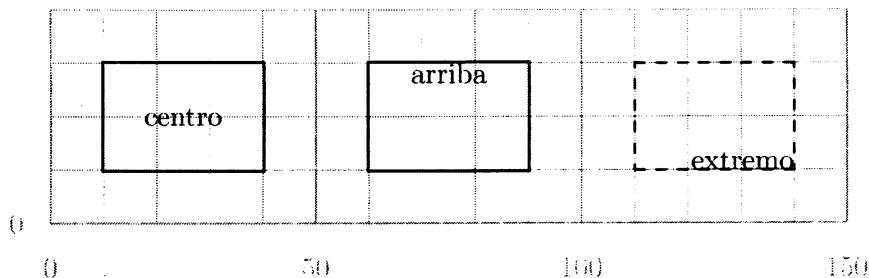
El comando

```
\put(x,y){\framebox(a,b)[justificación]{objeto}}
```

traza una caja de anchura a y altura b que encierra el *objeto* especificado. El extremo inferior izquierdo de la caja se coloca en el punto (x, y) . El parámetro opcional *justificación* controla la posición del objeto dentro de la caja; sus valores permitidos son: **c**, **t**, **b**, **l**, **r** (centro, arriba, abajo, izquierda, derecha). Se admiten combinaciones dobles como **cl**, **tr**, **bl**, etc. Por defecto, el *objeto* aparece centrado en la caja (opción **c**).

Para cajas con líneas a trozos se usa `\dashbox{r}(a,b)` en lugar de `\framebox(a,b)`. Con el argumento adicional r se especifica la longitud de cada uno de los trozos. La caja luce mejor si su altura y su anchura son múltiplos de r .

Ejemplo



```
\begin{center}
\setlength{\unitlength}{2pt}
\begin{picture}(150,40)
\color{gris}\graphpaper(0,0)(150,40)
\thicklines
\put(10,10){\framebox(30,20){centro}}
\put(60,10){\framebox(30,20)[t]{arriba}}
\put(110,10){\dashbox{2}(30,20)[br]{extremo}}
\end{picture}
\end{center}
```

Para cajas con bordes invisibles se usa la instrucción `\makebox(a,b)` en lugar de `\framebox(a,b)`. Haciendo `\makebox(0,0)`, el *objeto* aparece centrado en el punto (x, y) , lo cual es muy útil para colocar texto simple en una gráfica.

Los comandos para cajas mencionados arriba (`\framebox`, `\makebox` y `\dashbox`) requieren que la anchura y la altura de la caja sean escritos explícitamente. Existe un comando adicional, `\frame`, que traza cajas cuyas dimensiones están determinadas por su contenido. Más precisamente,

`\put(x,y){\frame{objeto}}`

coloca una caja cuyos bordes encierran el *objeto* dado, sin adicionar espacio extra a su alrededor. El extremo inferior izquierdo de la caja se coloca en el punto (x, y) .

Nótese que `\framebox` y `\makebox` tienen una sintaxis diferente dentro del entorno `picture` a la que tienen como cajas con texto normal (sección 3.15.1). Por otro lado, las líneas de las cajas creadas con `\framebox`, `\dashbox` y `\frame` están sujetas a los parámetros `\fboxrule` y `\fboxsep` (sección 3.15.3).

7.6.8. El comando `\shortstack`

El comando `\shortstack` se usa para colocar letras o palabras verticalmente apiladas. La instrucción

`\put(x,y){\shortstack[justificación]{lista}}`

coloca una caja con bordes invisibles que contiene las letras o palabras de la *lista* indicada, apiladas de arriba hacia abajo. Los elementos de la *lista* se separan entre sí por `\backslash`. El extremo inferior izquierdo de la caja aparecerá en el punto (x, y) . El parámetro opcional *justificación* controla la alineación de los ítems de la lista; sus valores permitidos son `c`, `l`, `r` (centro, izquierda, derecha), siendo `c` la justificación por defecto. Esencialmente, `\shortstack` produce una tabla con una sola columna; se puede usar `\backslash [longitud]` para añadir o eliminar espacio entre filas.

Example Frase de W. Goethe:

Se ponen	Se ponen palabras donde faltan las
palabras	i
donde	d
faltan	e
las ideas	a
	s

```
\setlength{\unitlength}{1pt}
\begin{picture}(280,60)
\put(20,0){\shortstack{Se ponen\\palabras\\donde\\faltan\\
  las ideas}}
\put(100,0){\shortstack{Se ponen palabras donde faltan las\\
  i\\d\\e\\a\\s}}
\end{picture}
```

El comando `\shortstack[justificación]{lista}` también se puede usar fuera del entorno `picture`, en párrafos normales. En tales casos, la pila de palabras y el texto circundante quedan alineados por la parte inferior. Esto se ilustra en el siguiente ejemplo.

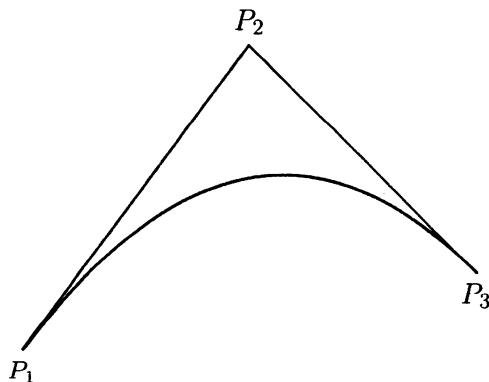
pila
de
palabras
poco

Ejemplo Frase corriente con una corriente en medio de ella.

Frase corriente con una `\shortstack{pila\\de\\palabras\\poco\\\\corriente}` en medio de ella.

7.6.9. Curvas cuadráticas de Bézier

Una curva cuadrática de Bézier con puntos de control P_1 , P_2 y P_3 es un arco que pasa por P_1 y P_3 de tal manera que el segmento de recta $\overline{P_1P_2}$ es tangente a la curva en P_1 , y el segmento de recta $\overline{P_2P_3}$ es tangente a la curva en P_3 :

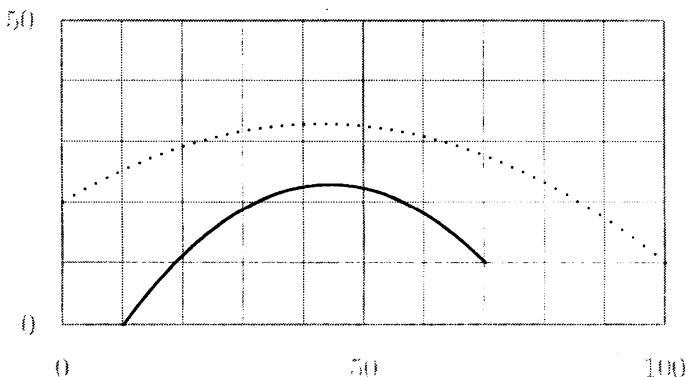


El comando

`\qbezier(a1,a2)(b1,b2)(c1,c2)`

traza la curva cuadrática de Bézier con puntos de control (a_1, a_2) , (b_1, b_2) y (c_1, c_2) . `\qbezier` tiene un parámetro opcional para escoger el número de puntos de la curva, lo cual permite trazar curvas punteadas. Así, para trazar una curva en la que se muestren n puntos uniformemente espaciados se escribe:

`\qbezier[n](a1,a2)(b1,b2)(c1,c2)`

Ejemplo

```
\begin{center}
\setlength{\unitlength}{0.8mm}
\begin{picture}(100,50)
\color{gris}\graphpaper(0,0)(100,50)
\thicklines
\qbezier(10,0)(40,40)(70,10)
\qbezier[55](0,20)(50,50)(100,10)
\end{picture}
\end{center}
```

7.6.10. El comando `\multiput`

El comando `\multiput` se usa para colocar un objeto múltiples veces en una gráfica sin tener que repetir la construcción `\put` para cada uno de ellos. Más precisamente, la instrucción

`\multiput(x,y)(Δ x, Δ y){n}{objeto}`

coloca n copias del *objeto* dado, primero en el punto (x, y) , luego en el punto $(x + \Delta x, y + \Delta y)$, a continuación en el punto $(x + 2\Delta x, y + 2\Delta y)$, y así sucesivamente.



Otras herramientas de L^AT_EX 2 _{ϵ}

8.1. El estilo `slides` para transparencias

El estilo `slides` está diseñado para la elaboración rápida de presentaciones o conferencias; se supone que el material producido se imprime o fotocopia sobre acetatos de tamaño carta¹. A este estilo se accede escribiendo

```
\documentclass{slides}
```

como primera línea del documento. El documento final se obtiene, por defecto, en hojas de tamaño carta, con márgenes generosos de 1 pulgada en los cuatro extremos, en el tamaño de letra `\LARGE` (sección 3.4) y con letra de tipo sans serif. Se pueden usar la mayoría de los comandos de edición de L^AT_EX, incluyendo tablas y fórmulas. También pueden incluirse gráficas externas, utilizando los comandos de la sección 7.3.4. Por razones tal vez obvias, los siguientes comandos *no* se pueden usar en el estilo `slides`:

- Los comandos divisionales `\chapter`, `\section`, `\subsection`, etc.
- Los entornos `table` y `figure` para la *inserción* de tablas y figuras.
- Los comandos `\pagestyle` y `\thispagestyle` (sección 2.10) para encabezados en las páginas.

No obstante, sí está permitido cambiar las dimensiones del cuerpo del documento con los parámetros `\textwidth`, `\textheight`, `\oddsidemargin`, etc. de la sección 2.13.

Con el estilo `slides` también se pueden cargar otros paquetes, tales como `graphicx`, `amsmath`, `color`, `babel`, etc, por lo cual, con un trabajo de edición relativamente fácil, cualquier documento L^AT_EX, escrito previamente con los estilos `article`, `report` o `book`, se puede transformar en una secuencia de transparencias.

¹ El estilo `slides` de L^AT_EX 2 _{ϵ} reemplaza el obsoleto paquete S_IL_ET_EX que se usaba en L^AT_EX2.09 y versiones anteriores.

El estilo `slides` posee tres entornos propios, `slide`, `overlay` y `note`, descritos a continuación. El uso de cada uno de estos entornos produce una hoja (transparencia) individual numerada, pero el usuario no está obligado a usarlos. Se puede escribir texto libremente fuera de estos entornos; la diferencia está en que el material que no sea incluido en alguno de los entornos `slide`, `overlay` o `note` aparece en hojas no numeradas.

El entorno `slide`. Su sintaxis es:

```
\begin{slide}
  texto de la transparencia
\end{slide}
```

Cada entorno `slide` da lugar a una transparencia individual; las diferentes transparencias aparecen numeradas consecutivamente en la parte inferior derecha.

Ejemplo En la Figura 8.1 de la página siguiente se reproduce una transparencia creada con el entorno `slide` (60 % del tamaño real), cuyo documento fuente completo aparece a continuación. Obsérvese que en el documento se cargan los paquetes `babel`, `inputenc` y `amsmath`.

```
\documentclass{slides}
\usepackage[spanish]{babel}
\usepackage[latin1]{inputenc}
\usepackage{amsmath}
\begin{document}
\begin{slide}
\centerline{\textbf{Principio de Inclusión y Exclusión}}
Sean $A_1, A_2, \dots, A_n$ conjuntos finitos. El número de
elementos de la unión $\bigcup_{i=1}^n A_i$ está dado por
$[ |A_1 \cup A_2 \cup \dots \cup A_n| = S_1 - S_2 + \dots +
(-1)^{n-1} S_n, ]$ donde
\begin{align*}
S_1 &= \sum_{i=1}^n |A_i|, \\
S_2 &= \sum_{1 \leq i < j \leq n} |A_i \cap A_j|, \\
&\quad && \& \hspace*{2.3cm} \vdots \\
S_k &= \sum_{1 \leq i_1 < \dots < i_k \leq n} |A_{i_1} \cap \dots \cap A_{i_k}|, \\
S_n &= |A_1 \cap \dots \cap A_n|.
\end{align*}
\end{slide}
NOTA: el número de sumandos de $S_k$, $1 \leq k \leq n$, es
exactamente $\binom{n}{k}$.
\end{document}
```

Principio de Inclusión y Exclusión

Sean A_1, A_2, \dots, A_n conjuntos finitos. El número de elementos de la unión $\bigcup_{i=1}^n A_i$ está dado por

$$|A_1 \cup A_2 \cup \dots \cup A_n| = S_1 - S_2 + \dots + (-1)^{n-1} S_n,$$

donde

$$\begin{aligned} S_1 &= \sum_{i=1}^n |A_i|, \\ S_2 &= \sum_{1 \leq i < j \leq n} |A_i \cap A_j|, \\ &\vdots \\ S_k &= \sum_{1 \leq i_1 < \dots < i_k \leq n} |A_{i_1} \cap \dots \cap A_{i_k}|, \\ S_n &= |A_1 \cap \dots \cap A_n|. \end{aligned}$$

NOTA: el número de sumandos de S_k , $1 \leq k \leq n$, es exactamente $\binom{n}{k}$.

1

FIGURA 8.1. Ejemplo de una transparencia producida con el entorno `slide`, reducida al 60% de su tamaño real.

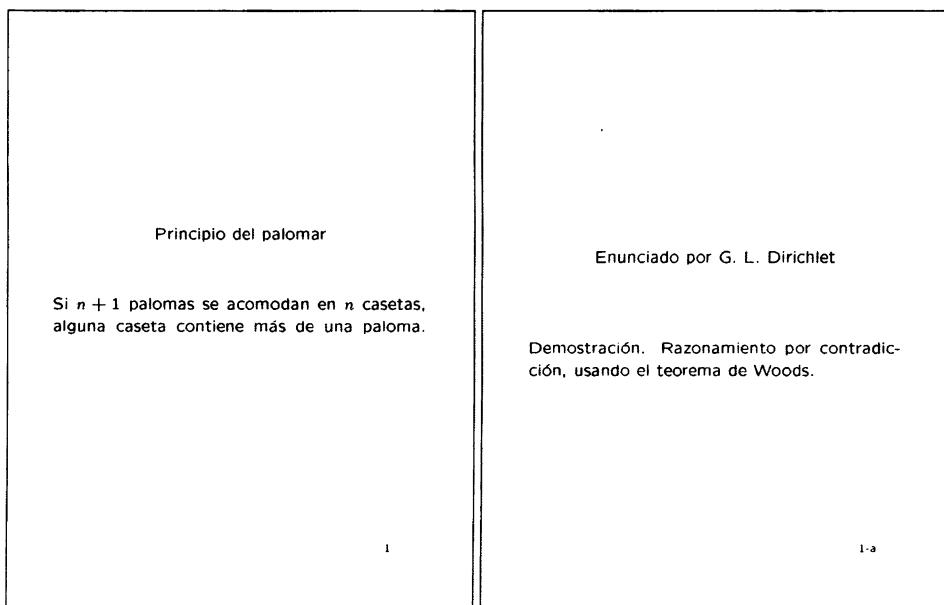
El entorno **overlay**. Para transparencias superpuestas a otras se usa:

```
\begin{overlay}
texto de la transparencia
\end{overlay}
```

Este entorno funciona exactamente como `slide`, excepto que la numeración de las transparencias creadas con `overlay` es subordinada. Así, la primera transparencia, creada con `overlay`, que sigue a la transparencia número 3 se numera 3-a, la siguiente 3-b y así sucesivamente.

Una transparencia superpuesta (creada con `overlay`) debe encajar perfectamente en la principal (creada con `slide`). Para lograrlo, lo mejor es que ambas transparencias tengan exactamente el mismo contenido, excepto que la parte visible en una de ellas sea invisible en la otra. Para hacer invisible un `texto` determinado, simplemente lo coloreamos de blanco usando la instrucción `\textcolor{white}{...}`. Esto requiere, por supuesto, acceso al paquete `color` (sección 7.2).

Ejemplo La transparencia de la izquierda fue creada con `slide` y la de la derecha con `overlay`. Con `\textcolor{white}{...}` logramos que la parte que es visible en una de ellas sea invisible en la otra; al superponerlas, encajan perfectamente. Obsérvese la numeración subordinada de la transparencia `overlay`. Las imágenes están reducidas al 30 % de su tamaño.



```
\begin{slide}
\begin{center}
Principio del palomar\\
\textcolor{white}{Enunciado por G. L. Dirichlet}
\end{center}
Si $n+1$ palomas se acomodan en $n$ casetas, alguna caseta contiene más de una paloma.

```

```
\textcolor{white}{\textbf{Demostración. Razonamiento por contradicción,}}
usando el teorema de Woods.}
\end{slide}
\begin{overlay}
\begin{center}
\textcolor{white}{\textbf{Principio del palomar}}\\
Enunciado por G. L. Dirichlet
\end{center}
\textcolor{white}{\textbf{Si } $n+1$ palomas se acomodan en $n$ casetas,}
alguna caseta contiene más de una paloma.}
\textbf{Demostración. Razonamiento por contradicción, usando el teorema de}
Woods.
\end{overlay}
```

El entorno note. Se usa para producir notas personales, como guía durante la conferencia o presentación. Su sintaxis es:

```
\begin{note}
texto de la nota
\end{note}
```

Se supone que estas notas se imprimen en papel ordinario y no en acetatos. Cada entorno **note** da lugar a una hoja individual, cuya numeración es subordinada, como en el caso de **overlay**, pero con números en lugar de letras. Así por ejemplo, la primera hoja creada con **note** que sigue a la transparencia número 3, se numera 3-1, la siguiente 3-2, y así sucesivamente.

Modalidades de impresión. El comando **\onlyslides{...}**, que se escribe en el preámbulo del documento, se usa para imprimir solamente algunas de las transparencias. Los números se separan con comas y se pueden incluir rangos de páginas. Por ejemplo, **\onlyslides{2,4,6-9,11}** solicita que se impriman solamente las transparencias 2, 4, 6 a 9 y 11, con sus respectivas transparencias superpuestas (*overlay*). De manera análoga, se utiliza **\onlynotes{...}** para imprimir sólo algunas de las notas. Sobre estos comandos hay que tener presente lo siguiente:

- Los argumentos de **\onlyslides** y **\onlynotes** no pueden ser vacíos.
- Si en el preámbulo del documento aparece **\onlyslides** pero no **\onlynotes**, entonces las notas no se imprimen, y viceversa.
- En el argumento de **\onlyslides** y **\onlynotes**, los números deben presentarse en orden ascendente, pero se pueden incluir números no existentes. Por ejemplo, con **\onlyslide{6-1000}** se imprimirán todas las transparencias (si son menos de 1000), excepto las cinco primeras.

8.2. El estilo `letter` para cartas

El estilo `letter` brinda al usuario de L^AT_EX un ambiente simple y cómodo para escribir cartas tradicionales. El formato de la carta está pre-definido y no hay que preocuparse por las márgenes ni por la colocación y la separación vertical entre los diferentes elementos de la carta. Hay que tener presente que el estilo `letter` implementa las normas epistolares anglosajonas. La estructura de un documento en el estilo `letter` se muestra a continuación; un ejemplo concreto aparece en la página 259.

```
\documentclass{letter}
\begin{document}
\address{...}
\signature{...}
\date{...}
\begin{letter}{...}
\opening{...}
```

Cuerpo de la carta

```
\closing{...}
\cc{...}
\encl{...}
\ps{...}
\end{letter}
\end{document}
```

Observaciones generales sobre el estilo `letter`:

- Con el estilo `letter` también se pueden cargar paquetes adicionales, como `babel` o `inputenc`, escribiendo `\usepackage{...}` después de `\documentclass{letter}`.
- El comando `\address{...}` se usa para la dirección del remitente (quien escribe la carta); los renglones se separan con `\backslash`. La dirección aparece impresa en la parte superior derecha de la hoja.
- El comando `\signature{...}` se usa para el nombre y la firma del remitente; se pueden escribir varios renglones, separados con `\backslash`. Aparece impreso al final de la carta, ligeramente cargado a la derecha.
- El comando `\date{...}` es opcional y se usa para escribir la fecha de la carta, la cual aparece impresa debajo de la dirección del remitente, en la parte superior derecha. Si se omite `\date`, L^AT_EX imprime la fecha vigente en el computador local (que es almacenada por L^AT_EX con el parámetro `\today`).

- La parte `\begin{letter}{...}` se usa para escribir el nombre del destinatario; se pueden escribir varios renglones, separados con `\backslash\backslash`.
- El comando `\opening{...}` se usa para la fórmula de encabezamiento de la carta (“Estimado señor”, “Apreciado amigo”, etc) y el comando `\closing{...}` para la despedida (“Hasta pronto”, “Me suscribo, atentamente”, etc).
- El comando `\cc{...}` es opcional y se usa para indicar los nombres de las personas o instituciones a las que se envían copias de la carta; los renglones se separan con `\backslash\backslash`. Da lugar al rótulo ‘cc:’; si se usa el paquete `babel`, opción `spanish`, el rótulo obtenido es ‘Copia a:’.
- El comando `\encl{...}` es opcional y se usa para indicar el material adjunto a la carta (en inglés, *enclosures*); los renglones se separan con `\backslash\backslash`. Da lugar al rótulo ‘encl:’; si se usa el paquete `babel`, opción `spanish`, el rótulo obtenido es ‘Adjunto:’.
- El comando `\ps{...}` es opcional y se usa para la posdata o *post scriptum*. Se pueden escribir varios renglones, separados con `\backslash\backslash`. No hay rótulo pre-definido para este campo; el usuario debe escribir explícitamente P.S., si así lo desea.
- En el *cuerpo de la carta* propiamente dicho se pueden usar la mayoría de los comandos de edición de LATEX, excepto instrucciones como `\chapter` o `\section` que no tendrían sentido en una carta.
- En un mismo documento LATEX se pueden escribir varias cartas con el mismo remitente (`\address` y `\signature`), utilizando tantos entornos `\begin{letter} ... \end{letter}` como se deseé. Las cartas pueden tener fechas diferentes, para lo cual hay que redefinir el comando `\today` escribiendo `\renewcommand{\today}{...}` inmediatamente antes de cada `\opening`.
- El estilo `letter` posee un comando opcional más: `\makelabels`, el cual se escribe en el preámbulo del documento y produce, para cada entorno `letter` utilizado, una página separada con el nombre del destinatario, tal cual aparece escrito en el comando `\begin{letter}{...}`.

Ejemplo En la Figura 8.2 de la página siguiente se reproduce una carta creada con el estilo `letter` (75 % del tamaño real), seguida del documento fuente completo. Obsérvese que en el documento se cargan los paquetes `babel` e `inputenc` y no se utiliza ningún comando para añadir espacios horizontales o verticales.

El Chi Lee
La Torre de Papel
Atenas

25 de Marzo de 2000

Dr. Juan Peregrino
Facultad de Ciencias Inexactas
Universidad del Cuarto Mundo
Samarkanda

Estimado señor Peregrino:

Lamento informarle que su trabajo titulado "La vuelta al mundo en 8000 páginas" no puede ser publicado debido a que los costos editoriales de un libro de 8000 páginas, como el que Ud. propone en su manuscrito, exceden los límites de inversión y riesgo que la editorial *La Torre de Papel* puede asumir en las actuales condiciones del mercado.

Nuestras consideraciones de aprecio.

El Chi Lee
Director editorial

Copia a: Gerente comercial
Oficina Jurídica

Adjunto: Cancelación del pre-contrato

P.S. El manuscrito no será devuelto

FIGURA 8.2. Carta, reducida al 75 % de su tamaño real, escrita con el estilo letter. El documento fuente aparece en la página siguiente.

```
\documentclass{letter}
\usepackage[spanish]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
\address{El Chi Lee \\ \emph{La Torre de Papel} \\ Atenas}
\signature{El Chi Lee \\ Director editorial}
\date{25 de Marzo de 2000}
\begin{letter}{Dr. Juan Peregrino \\ Facultad de Ciencias Inexactas
\\ Universidad del Cuarto Mundo \\ Samarkanda}
\opening{Estimado señor Peregrino:}
Lamento informarle que su trabajo titulado “La vuelta al mundo en
8000 páginas” no puede ser publicado debido a que los costos
editoriales de un libro de 8000 páginas, como el que Ud. propone en
su manuscrito, exceden los límites de inversión y riesgo que la
editorial \emph{La Torre de Papel} puede asumir en las actuales
condiciones del mercado.
\closing{Nuestras consideraciones de aprecio,}
\cc{Gerente comercial\\ Oficina Jurídica}
\encl{Cancelación del pre-contrato}
\ps{P.S. El manuscrito no será devuelto}
\end{letter}
\end{document}
```

8.3. El entorno `list`

Con el entorno `list` el usuario puede diseñar su propio estilo de listas si desea algo diferente de los entornos básicos de LATEX, `itemize`, `enumerate` y `description` (sección 3.19). Se usa en la forma:

```
\begin{list}{etiqueta}{parámetros}
\item Texto
\item Texto
:
\item Texto
\end{list}
```

En el primer argumento obligatorio, `{etiqueta}`, se indica la etiqueta o símbolo que aparece con cada `\item`. El argumento `{parámetros}` se usa para cambiar los valores de los parámetros mostrados en la Figura 8.3. Todos estos parámetros asumen por defecto ciertos valores, dependiendo del estilo del documento (`article`, `book`, etc) y del *nivel* de la lista (para listas encajadas en otras); si se desea mantener los valores establecidos por defecto basta escribir `{}` para este argumento.

Los parámetros de la Figura 8.3 se cambian con `\setlength`; sobre ellos precisamos lo siguiente:

- Los valores asignados a `\leftmargin`, `\rightmargin` y `\labelwidth` deben ser no-negativos. Por defecto, `\rightmargin` es 0 cm mientras que `\labelwidth` tiene la anchura natural de la caja LATEX que contiene la etiqueta respectiva.
- El parámetro `\itemindent` controla la sangría del *primer* párrafo de cada ítem y `\listparindent` la sangría de los demás párrafos. A ambos parámetros se les asigna por defecto el valor 0 cm pero pueden tomar valores tanto positivos como negativos.

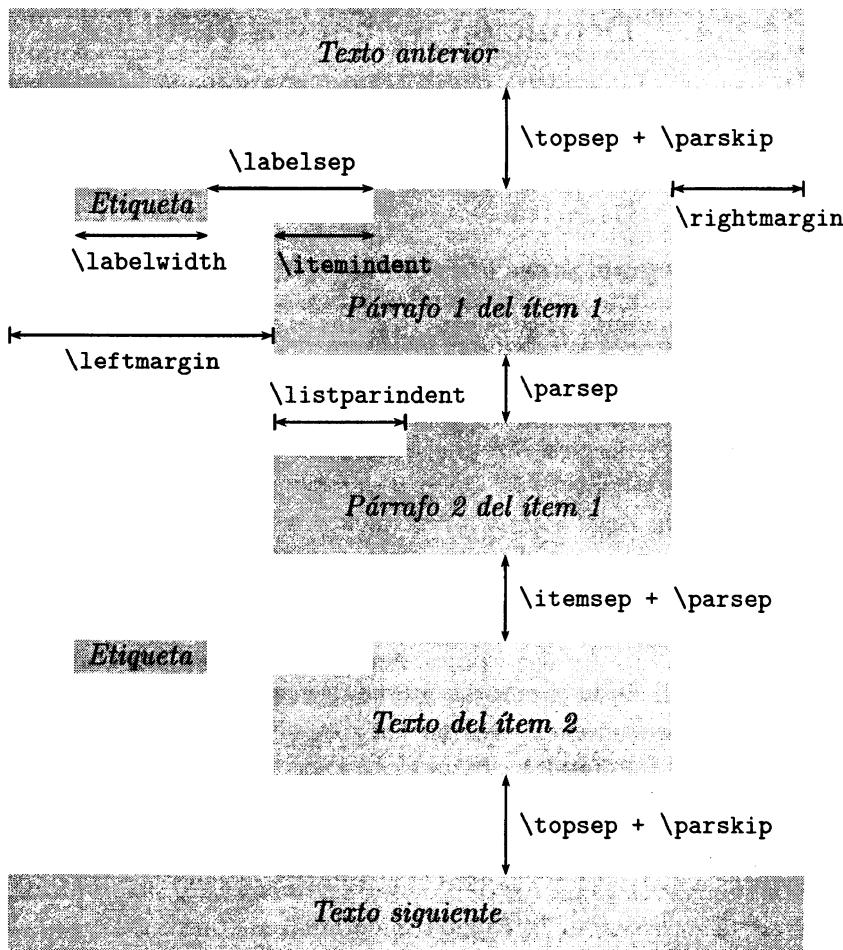


FIGURA 8.3. Parámetros del entorno *list*.

Ejemplo Con los siguientes comandos se obtiene una lista similar a la del entorno `itemize` pero con márgenes de 2cm a izquierda y a derecha (en el entorno `itemize`, `\rightmargin` es 0 cm). Cada ítem estará demarcado con la viñeta • (\$\bullet\$).

```
\begin{list}{\bullet}
\setlength{\leftmargin}{2cm}
\setlength{\rightmargin}{2cm}
\item
\item
\vdots
\item
\end{list}
```

8.4. Definición de nuevos entornos

Con la instrucción `\newenvironment` se pueden definir nuevos entornos. Al igual que sucede con `\newcommand` (sección 3.24), el nuevo entorno puede o no tener argumentos; en el último caso, uno de ellos puede ser un argumento opcional.

Entornos sin argumentos. El comando

```
\newenvironment{nombre}{parte inicial}{parte final}
```

define el entorno denominado `nombre`, para el cual LATEX reemplaza el comando `\begin{nombre}` por la *parte inicial* y `\end{nombre}` por la *parte final*. El `nombre` del nuevo entorno es asignado por el usuario usando cualquier secuencia de letras o dígitos que no comience con la expresión `end`². No debe existir previamente ningún entorno denominado `nombre` ni ningún comando `\nombre`.

Un nuevo entorno se define usualmente en términos de entornos ya existentes en LATEX.

Ejemplo Modificando el entorno `quote` (sección 3.12) podemos definir el entorno `cita` para obtener citas en letra cursiva o itálica (`\itshape`) de tamaño pequeño (`\small`):

```
\newenvironment{cita}{\begin{quote}\small\itshape}{\end{quote}}
```

Al encontrar la instrucción `\begin{cita}`, LATEX ejecuta la *parte inicial* dada en la definición del entorno `cita`, es decir:

²Obsérvese que para los nombres de entornos nuevos, LATEX permite tanto letras como dígitos, pero para comandos nuevos sólo se admiten letras.

```
\begin{quote}\small\itshape
```

Esto hace que se inicie el usual entorno `quote`, pero en letra itálica pequeña. Cuando aparece la instrucción `\end{cita}`, L^AT_EX ejecuta la *parte final* de la definición, es decir, `\end{quote}`.

Entornos con argumentos obligatorios. Un nuevo entorno definido por medio de `\newenvironment` admite argumentos o parámetros, en la forma:

```
\newenvironment{nombre}[n]{parte inicial}{parte final}
```

siendo *n* el número de argumentos del nuevo entorno denominado *nombre*. El valor máximo de *n* es 9 y los parámetros #1, #2, etc utilizados para la definición del entorno pueden aparecer únicamente en la *parte inicial*. Los *n* argumentos del entorno así definido adquieren el carácter de argumentos obligatorios.

Ejemplo Podemos definir un entorno genérico para listas, denominado *misitems*, similar a *itemize*, pero que permita escoger la viñeta para demarcar los diferentes items; se requiere entonces un argumento o parámetro:

```
\newenvironment{misitems}[1]{\begin{list}{#1}{}{\end{list}}}
```

Con esta definición, algo como

```
\begin{misitems}{$\circ$}
\item
:
\item
\end{misitems}
```

da lugar a un listado como el obtenido con *itemize*, pero con items demarcados con la viñeta *o* (\circ). El paquete *enumerate* (sección 3.19.4) define su entorno *enumerate* de forma análoga.

Entornos con un argumento opcional. Entre los argumentos de un entorno definido con `\newenvironment`, uno (y sólo uno) puede ser opcional. En tal caso, la definición del nuevo entorno se hace en la forma:

```
\newenvironment{nombre}[n][opc]{parte inicial}{parte final}
```

donde el primero de los *n* argumentos declarados es el opcional y su valor por defecto está dado por la expresión *opc*.

Redefinición de entornos existentes. Con `\renewenvironment` se puede redefinir un entorno pre-definido por L^AT_EX o por un paquete ya cargado. Su sintaxis coincide con la de `\newenvironment` pero su uso se recomienda solamente a los usuarios experimentados.

8.5. Longitudes elásticas

Algunos parámetros de TeX y LATEX utilizan *longitudes elásticas* (en inglés, *rubber lengths*). Estas longitudes se pueden expandir o contraer una determinada magnitud, a diferencia de las normales, también llamadas *longitudes rígidas*. La sintaxis de una longitud elástica es:

valor nominal plus expansión minus contracción

donde el *valor nominal*, la *expansión* y la *contracción* son longitudes normales.

Ejemplo El espacio vertical que LATEX añade antes del título de una sección, creada con el comando \section, es una longitud elástica, lo cual permite una distribución más flexible del texto en las páginas. En los estilos article, book y report, el espacio vertical en cuestión está definido como:

3.5ex plus 1ex minus 0.2ex

cuyo significado es: la distancia vertical que antecede al título de la sección es 3.5ex, pero se puede expandir hasta 4.5ex o se puede reducir a 3.3ex (recuérdese que la unidad ex es la altura de la letra x, en la fuente vigente).

8.6. Comandos internos de LATEX

En los nombres de ciertos comandos de LATEX, llamados *comandos internos*, aparece el símbolo @. Tales comandos no pueden ser utilizados en el cuerpo de un documento ni pueden ser redefinidos por aplicaciones directas de \renewcommand o \providetcommand, ya que estas instrucciones sólo admiten letras en los nombres de los comandos (sección 3.24). No obstante, se puede acceder a los comandos internos, para redefinirlos, por ejemplo, encerrando el área en la que aparece el símbolo @ entre las instrucciones \makeatletter y \makeatother. El comando \makeatletter hace que LATEX considere @ como una letra y \makeatother restaura el significado usual de @ como símbolo no-alfabético.

Ejemplo LATEX posee el comando interno \addtoreset, de dos argumentos, para hacer que un contador determinado se reinicie en 0 cuando otro contador se incrementa en 1. Podemos usar este comando para hacer que las ecuaciones se enumeren independientemente en cada sección (recuérdese que la numeración de ecuaciones es acumulativa en el estilo article, sección 4.31.2). Escribimos, preferiblemente en el preámbulo:

```
\makeatletter
@addtoreset{equation}{section}
\makeatother
```

8.7. Modificación de los comandos seccionales

El formato para los títulos de capítulos, secciones, subsecciones, etc está rígidamente predefinido por los estilos básicos de L^AT_EX, `article`, `book` y `report`. Estos estilos controlan el tipo de letra, el tamaño y la justificación, así como la colocación de los títulos con respecto al texto que los circunda. En la presente sección explicaremos cómo se pueden modificar los comandos seccionales, desde `\section` hasta `\subparagraph`, por medio del comando interno de L^AT_EX `\@startsection`.

Las redefiniciones de `\@startsection` deben estar encerradas entre los comandos `\makeatletter` y `\makeatother`, tal como se explicó en la sección 8.6, debido a la presencia del símbolo `@`. Se recomienda escribir estas redefiniciones en el preámbulo del documento.

La sintaxis del comando interno `\@startsection` es la siguiente:

```
\@startsection{secc}{nivel}{sangría}{separación anterior}%
{separación posterior}{estilo}
```

donde los seis argumentos son obligatorios y tienen el siguiente significado:

secc Es el nombre de una subdivisión existente, tal como `section` o `subsection`. No se escribe el símbolo `\` en este argumento.

nivel Se refiere al nivel de la subdivisión `secc`, de acuerdo con la jerarquía descrita en la sección 6.2. Este `nivel` se usa para la numeración automática, según el valor asignado al parámetro `secnumdepth`.

sangría Sangría de los títulos en el margen izquierdo. Puede ser una dimensión negativa, en cuyo caso, los títulos invadirán el margen.

separación anterior Longitud elástica (sección 8.5) cuyo valor absoluto es el espacio vertical añadido antes del título de cada subdivisión `secc`. Puede ser una dimensión negativa, en cuyo caso, el primer párrafo que sigue al título no lleva sangría.

separación posterior Longitud elástica; si es positiva, representa el espacio vertical añadido debajo del título de cada subdivisión `secc`. Si es negativa, el título no se despliega, es decir, el texto que sigue al título aparece en el mismo renglón que éste, a una distancia dada por la `separación posterior`.

estilo Se usa para indicar el tipo de letra de los títulos de la subdivisión `secc` (con declaraciones globales como `\bfseries`, `\itshape`, etc) y el tamaño de la letra (con declaraciones como `\large`, `\huge`, etc). En

este argumento pueden aparecer también otros comandos que modifiquen el aspecto del título, como `\centering`, para títulos centrados, y `\raggedleft` o `\raggedright`, para títulos no justificados.

Recuérdese (secciones 6.1 y 6.2) que cada comando seccional `\secc` tiene una versión estándar y una “versión estrella”. La versión estándar admite un argumento opcional:

```
\secc[título abreviado]{título completo}
```

El *título abreviado* es la simplificación del *título completo*, para ser incluida en los encabezados de las páginas y en la tabla de contenido o índice general. La versión estrella no admite el argumento opcional:

```
\secc*{título}
```

Una unidad seccional creada con `\secc*` no recibe numeración automática y su *título* no es incluido en el índice general ni en los encabezados.

Ejemplo Títulos centrados para secciones. Las instrucciones que siguen redefinen el comando `\section` de LATEX, usado en los estilos `article`, `book` y `report`. Las longitudes elásticas en los argumentos *separación anterior* y *separación posterior* son exactamente las que utiliza LATEX. El único elemento nuevo en esta definición es la instrucción `\centering`, en el argumento *estilo*, para obtener títulos centrados.

```
\makeatletter
\renewcommand{\section}{\@startsection{section}{1}{0pt}%
{-3.5ex plus -1ex minus -0.2ex}{2.3ex plus 0.2ex}%
{\centering\normalfont\Large\bfseries}}
\makeatother
```

Se recomienda escribir estas instrucciones en el preámbulo del documento.

Ejemplo Las instrucciones que siguen redefinen el comando `\subsection` de LATEX. Los títulos aparecerán no desplegados (el argumento *separación posterior* es negativo), con una sangría de 1 cm y en versalitas (`\scshape`). Las longitudes elásticas escritas en los argumentos *separación anterior* y *separación posterior* son las mismas que utiliza LATEX, excepto por el cambio de signo en la *separación posterior*.

```
\makeatletter
\renewcommand{\subsection}{\@startsection{subsection}{2}{1cm}%
{-3.25ex plus -1ex minus -0.2ex}{-1.5ex plus -0.2ex}%
{\normalfont\large\scshape}}
\makeatother
```

Hay un detalle adicional sobre el formato de los títulos: LATEX no escribe un punto después de los números de las subdivisiones de nivel ≥ 1 (secciones, subsecciones, etc); por ejemplo, 3.2, 5.1.3, etc. Podemos forzar el punto final (para obtener, por ejemplo, 3.2., 5.1.3., etc) redefiniendo el comando interno \secntformat, lo cual se puede hacer en la forma³:

```
\makeatletter
\renewcommand{\secntformat}[1]{\csname the#1\endcsname.\quad}
\makeatother
```

El comando \chapter no se controla con \startsection, por lo que modificar el formato para los títulos de los capítulos no es tan sencillo. Como alternativa, se puede usar el paquete fncychap, el cual proporciona seis formatos predefinidos para reemplazar el formato estándar de LATEX. Este paquete se describe en la sección 13.3.

8.8. Documentos LATEX por correo electrónico

Para quienes son usuarios habituales del correo electrónico, enviar un documento adjunto a un mensaje no es ningún misterio. Tratándose de documentos fuente LATEX, la tarea se simplifica porque éstos son archivos de texto llano, compactos y relativamente pequeños. Pero quien recibe el documento puede llevarse sorpresas desagradables si en él se hace uso de paquetes o archivos no existentes en el computador local.

Antes de enviar un documento LATEX por correo electrónico, es aconsejable procesarlo una vez más, colocando la instrucción \listfiles en el preámbulo. Tal instrucción hace que LATEX identifique los archivos externos utilizados durante el procesamiento del documento. El usuario puede leer la lista de tales archivos en la parte final del archivo ‘—.log’, bajo el encabezado *File List*. Allí aparecerán listados todos los archivos que hayan sido cargados con los comandos \usepackage, \input, \include, los nombres de los archivos gráficos externos importados con \includegraphics, el nombre del archivo ‘—.bbl’ con información sobre la ejecución de BIBTEX, así como los nombres de los archivos de definición de fuentes (archivos ‘—.fd’). Además, LATEX declara explícitamente cuáles de los archivos o paquetes pertenecen a su colección estándar.

Utilizando esta información, el usuario puede advertir a los destinatarios, o enviar los archivos externos necesarios.

³La opción spanish del paquete babel hace precisamente esta redefinición para añadir puntos después de los números de secciones y subsecciones.

Ejemplo A continuación aparecen algunas líneas de la lista de archivos generada por el comando \listfiles, tomadas de un documento ‘—.log’ típico.

```
*File List*
book.cls      1999/01/07 v1.4a Standard LaTeX document class
bk12.clo      1999/01/07 v1.4a Standard LaTeX file (size option)
babel.sty     1999/04/08 v3.6o The Babel package
spanish.ldf    1999/04/05 v3.4i Spanish support from babel system
inputenc.sty   1998/03/05 v0.97 Input encoding file (test version)
latin1.def    1998/03/05 v0.97 Input encoding file (test version)
color.sty     1998/05/27 v1.0g Standard LaTeX Color (DPC)
dvips.def     1998/05/03 v3.0f Driver-dependant file (DPC,SPQR)
amsmath.sty   1995/02/23 v1.2b AMS math features
graphicx.sty  1997/06/09 v1.0d Enhanced LaTeX Graphics (DPC,SPQR)
    upzd.fd    2000/01/12 PSNFSS-v8.1 font definitions for U/pzd.
    upsy.fd    2000/01/12 PSNFSS-v8.1 font definitions for U/psy.
    ulasy.fd   1998/08/17 v2.2eLaTeX symbol font definitions
    omscmr.fd  1998/03/27 v2.5g Standard LaTeX font definitions
Grafica1.eps  Graphic file (type eps)
Grafica2.bmp  Graphic file (type bmp)
```



El uso de otras fuentes en documentos L^AT_EX 2 _{ε}

Las implementaciones típicas de L^AT_EX utilizan las fuentes CM (*Computer Modern Fonts*), diseñadas por el propio Donald Knuth. El presente libro, por ejemplo, fue impreso con tales fuentes. En L^AT_EX 2 _{ε} existen comandos de alto nivel para cambiar las fuentes utilizadas por defecto; tales herramientas hacen parte del nuevo esquema de manejo de fuentes denominado NFSS (*New Font Selection Scheme*), diseñado en el período 1989–92 por el equipo L^AT_EX3. En L^AT_EX 2.09 y versiones anteriores no existían tales herramientas. Este capítulo presenta los rudimentos del esquema NFSS, con énfasis en las colecciones AE y PSNFSS de fuentes PostScript, disponibles gratuitamente e incluidas en el CD adjunto.

9.1. Atributos de las fuentes

En el esquema NFSS toda fuente en L^AT_EX 2 _{ε} tiene cinco atributos:

Codificación (encoding). Es la secuencia de los caracteres de la fuente.

Las dos codificaciones más importantes son: OT1 (codificación Knuth de 7 bits, o $2^7 = 128$ caracteres por fuente) y T1 (codificación Cork de 8 bits, o $2^8 = 256$ caracteres por fuente). La codificación OT1 ha sido usada por las implementaciones de T_EX y L^AT_EX durante la mayor parte de su existencia; las fuentes CM tienen esta codificación. La codificación T1, adoptada en la conferencia T_EX de Cork (Irlanda) de 1990, se creó principalmente para incorporar idiomas diferentes del inglés, por medio de la adición de símbolos acentuados y otros caracteres usados por los idiomas europeos. L^AT_EX 2 _{ε} es compatible con las codificaciones OT1 y T1, pero las versiones anteriores de L^AT_EX únicamente son compatibles con OT1.

En la codificación OT1, los acentos diacríticos se obtienen combinando tildes con letras. Así por ejemplo, el símbolo á es producido por LATEX (con la ayuda del paquete `inputenc`) como una combinación del acento agudo ‘́’ y la letra a. En la codificación T1, en cambio, cada letra acentuada es un símbolo individual, lo cual permite que símbolos como á, é, í, etc sean usados en el comando `\hyphenation{...}` (sección 3.1). Con las fuentes CM originales, `\hyphenation` no admite símbolos acentuados.

Existen versiones T1 de las fuentes CM, algunas de ellas comerciales; una versión de dominio público se conoce como “las fuentes EC”, abreviación de *Extended Computer Modern Fonts*¹. Los nombres de los archivos de las fuentes tradicionales CM comienzan por ‘cm’, y los de las fuentes EC comienzan por ‘ec’. En muchas implementaciones de LATEX las fuentes EC vienen ya pre-instaladas², y en tal caso se puede acceder a éstas por medio del paquete estándar `fontenc`:

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

Si no se dispone de las fuentes EC, o de otras fuentes T1, hay que limitarse a la codificación estándar OT1, para lo cual no se requiere ningún comando especial, ya que OT1 es la codificación utilizada por defecto. De todas maneras, cualquiera que sea la codificación empleada, podemos usar en el documento fuente (aunque no en el argumento de `\hyphenation{...}`) símbolos acentuados tales como á, é, ñ, ö, etc si cargamos el paquete `inputenc`, opción `[latin1]` (sección 3.1). Al usar las fuentes EC, el paquete `inputenc` ya no es necesario.

Familia (family). El esquema NFSS clasifica las fuentes en tres familias básicas: la familia serif romana —utilizada para la letra principal del documento— la familia sans serif y la familia mono-espaciada (typewriter). En la sección 9.5 se indica cómo se pueden reemplazar las familias de fuentes CM que LATEX utiliza por defecto.

Serie (series). Se refiere al “peso” de la fuente; es decir, a cuán “amplia” o “gorda” es: media (`m`), negrilla (`b`), negrilla extendida (`bx`), semi-negrilla (`sb`), condensada (`c`), ligera (`l`), etc.

Variante (shape). Se refiere al aspecto o forma de la fuente: normal (`n`), itálica (`it`), inclinada (`s1`), versalitas (`sc`), etc.

Tamaño (size). Dimensión TEX, en puntos: `10pt`, `12pt`, etc.

¹Las fuentes EC fueron diseñadas por Jörg Knappen y distribuidas en 1997.

²Tal es el caso de MiKTeX, para Windows, por ejemplo.

9.2. Los archivos de fuentes

TeX, L^AT_EX y los manejadores para visualización e impresión utilizan varios archivos de control e información, aparte de las fuentes propiamente dichas. La mayoría de los archivos relacionados con fuentes tienen el nombre básico de la fuente pero con extensiones diferentes, a saber:

- *.**tfm** (*TeX font metrics*). Contienen la información necesaria para procesar un documento: tamaño, altura, peso, inclinación y otras características de los símbolos de las fuentes. Junto con los archivos *.fd, los *.tfm son en realidad los únicos archivos consultados por L^AT_EX; las fuentes propiamente dichas y sus archivos subsidiarios son utilizados por los manejadores de pantalla o impresora.
- *.**pk** (*packed bitmap fonts*). Contienen las imágenes (*bitmaps*) de los símbolos de cada fuente; están organizados por tamaño y resolución. Algunas veces se distribuyen como archivos *.mf: en tal caso, el programa METAFONT los convierte en *.pk para los manejadores DVI.
- *.**vf** (*virtual fonts*). Las fuentes virtuales son una alternativa a los archivos *.pk. En lugar de buscar la imagen pk, el manejador consulta la definición virtual para cada símbolo, el cual puede ser luego tomado o distorsionado de una fuente real. Las letras versalitas e inclinada (*slanted*), por ejemplo, pueden ser “distorsiones” de otras fuentes, si no existen en forma “real”. Algunas fuentes PostScript existen en realidad como fuentes virtuales; tal es el caso de las fuentes AE (sección 9.3) y de las fuentes de la colección PSNFSS (sección 9.4).
- *.**fd** (*font definitions*). Contienen comandos NFSS para asociar los nombres de las fuentes externas con sus atributos. Los nombres de estos archivos constan de la codificación y la familia de la fuente, por ejemplo, **ot1cmr.fd**, **t1ptm.fd**. Cuando en el documento se solicita una determinada fuente por primera vez, L^AT_EX consulta el archivo fd correspondiente, si éste existe.
- *.**map** (*mapping files*). Estos archivos de “mapeo de fuentes” le indican al manejador los nombres de las fuentes reales para que aquél realice sobre éstas las transformaciones o re-codificaciones necesarias.
- *.**pfa**, *.**pfb** (*PostScript fonts*). Algunas fuentes PostScript vienen en forma “real”, en contraste con las versiones virtuales (*.vf) mencionadas arriba. Las fuentes *.pfa están en formato ASCII y las *.pfb en formato binario. Los archivos *.map le indican al manejador si las fuentes *.pfa ó *.pfb son necesarias.

9.3. La colección AE de fuentes PostScript

Hay varias razones por las cuales las fuentes PostScript (ya sean reales o virtuales) se consideran superiores a las fuentes en forma de *bitmaps* (archivos `*.pk` o `*.mf`). En primer lugar, son independientes de la resolución del dispositivo de salida (monitor o impresora). Por otro lado, su manipulación (aumento a escala o rotación) es más confiable y precisa. Finalmente, las fuentes PostScript son muy apropiadas para generar documentos PDF con el programa `pdflatex`; las fuentes *bitmaps*, por el contrario, producen archivos PDF de pésima calidad (la generación de archivos en formato PDF por medio de `pdflatex` es un tópico que se tratará con todo detalle en el Capítulo 10).

Hay que aclarar que las fuentes gratuitas EC, mencionadas en la sección 9.1, tienen la codificación T1 pero no son fuentes PostScript, sino *bitmaps*. Para subsanar esta deficiencia de las fuentes EC, se han diseñado fuentes PostScript virtuales, en la codificación T1, que emulan las fuentes CM. Conocidas como “fuentes AE” (abreviación de *Almost European*), estas fuentes son de gran calidad y se distribuyen gratuitamente³. Una vez instaladas (véase el recuadro gris al final de esta página), se pueden usar en documentos LATEX cargando el paquete `ae`. Unos cuantos caracteres, como Æ (\$\eth\$) y £ (\pounds), no están incluidos en la colección AE, pero hay *bitmaps* para estos símbolos aislados, a los cuales se accede cargando el paquete `aecompl`, también incluido en la distribución. Por consiguiente, para usar las fuentes AE en documentos LATEX basta añadir las instrucciones

```
\usepackage{ae}
\usepackage{aecompl}
```

en el preámbulo del documento. Puesto que las fuentes AE tienen la codificación T1, ya no es necesario el paquete `inputenc`, y en el argumento de `\hyphenation` se admiten caracteres acentuados.

La colección AE también incluye fuentes para transparencias. Estas fuentes son invocadas cuando en el documento se utiliza el estilo `slides` (sección 8.1) y se ha cargado `ae` con la opción `slides`, en la forma

```
\usepackage[slides]{ae}
```



En la carpeta `/AE/` del CD adjunto se incluye la colección de fuentes AE. Los diferentes archivos (`*.tfm`, `*.vf`, `*.fd`, `*.sty`, etc) están repartidos en subdirectorios que sugieren su ubicación final, siguiendo el esquema TDS (véase el Apéndice C). En algunas implementaciones de TeX y LATEX (como MiKTeX) estas fuentes vienen pre-instaladas.

³Las fuentes AE fueron diseñadas por Lars Engebretsen.

9.4. La colección PSNFSS de fuentes PostScript

La colección PSNFSS⁴ ofrece un variado surtido de fuentes PostScript, y paquetes subsidiarios, para reemplazar las fuentes CM en documentos LATEX. Al cargar el paquete `bookman`, por ejemplo, los tipos de letra romana (`cmr`), sans serif (`cms`) y mono-espaciada (`cmtt`) son reemplazados por la fuentes ‘Bookman’, ‘Avant Garde’ y ‘Courier’, respectivamente. A diferencia de las fuentes AE mencionadas en la sección anterior, las fuentes de PSNFSS no emulan a las CM sino que son fuentes tipográficamente diferentes.

 Todo el material de la colección PSNFSS (versión 8.2), incluyendo una completa guía de instalación (archivo `00readme.txt`) y la documentación (archivo `psnfss2e.pdf`), se encuentra en la carpeta `/PSNFSS/` del CD adjunto. En algunas implementaciones, como MiKTeX, las fuentes y paquetes de PSNFSS vienen ya instalados. En lo sucesivo, supondremos que el lector dispone de la colección PSNFSS correctamente instalada.

9.4.1. Las familias de fuentes de PSNFSS

Las familias de fuentes de la colección PSNFSS y sus atributos se muestran en la Tabla 9.1. Como se puede observar, la mayoría de estas fuentes tienen las variantes inclinada (`s1`), itálica (`it`) y versalitas (`sc`), en las series media (`m`) y negrilla (`b`). Las fuentes Palatino, New Century Schoolbook, Bookman y Times son de tipo romano; las fuentes Helvetica y Avant Garde son de tipo sans serif, y la fuente Courier es mono-espaciada (`typewriter`). La fuente Zapf Chancery tiene únicamente la variante itálica y la fuente Zapf Dingbats⁵ es una fuente especial de símbolos (véase la sección 9.4.5).

La manera más práctica de utilizar las fuentes de la Tabla 9.1, para reemplazar uniformemente las fuentes CM en un documento LATEX, es cargar alguno de los paquetes de la colección PSNFSS, descritos en la sección 9.4.2. Pero podemos apreciar el tipo de letra de una *familia* de fuentes particular por medio de la siguiente declaración global:

```
\fontfamily{familia}\selectfont
```

Escribimos determinado *texto* con las fuentes de la familia Palatino (`ppl`), por ejemplo, por medio de `{\fontfamily{ppl}\selectfont texto}`. Usando instrucciones de esta clase, se presenta a continuación, en los tipos de letra de las familias de la Tabla 9.1, excepto en la letra Courier mono-espaciada, una pequeña descripción del tradicional arte de la tipografía.

⁴Originalmente desarrollada por Sebastian Rahtz.

⁵Las fuentes Zapf Chancery y Zapf Dingbats fueron diseñadas por Hermann Zapf.

Familia	Series	Variantes	Nombre de la fuente
ptm	m, b	n, sl, it, sc	Adobe Times
ppl	m, b	n, sl, it, sc	Adobe Palatino
pnc	m, b	n, sl, it, sc	Adobe New Century Schoolbook
pbk	m, b	n, sl, it, sc	Adobe Bookman
phv	m, b, mc, bc	n, sl, sc	Adobe Helvetica
pag	m, b	n, sl, sc	Adobe Avant Garde
pcr	m, b	n, sl, sc	Adobe Courier
pzc	m	it	Zapf Chancery
pzd	m	n	Zapf Dingbats

TABLA 9.1. Familias de fuentes de la colección PSNFSS.

Para escribir porciones *aisladas* de un documento en una determinada fuente, teniendo mayor control sobre los atributos de la fuente. LATEX 2_E posee la instrucción \DeclareFixedFont, descrita en la sección 9.6.

Escrito con letra de la familia Times Roman (ptm):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

Escrito con letra de la familia Palatino (ppl):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

Escrito con letra de la familia New Century Schoolbook (pnc):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

Escrito con letra de la familia Bookman (pbk):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

Escrito con letra de la familia Avant Garde (pag):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

Escrito con letra de la familia Helvetica (phv):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

Escrito con letra de la familia Zapf Chancery (pzc):

La tipografía. Hasta hace unas pocas décadas, el método tradicional de impresión era la tipografía o *composición de caja*. Se llamaba así porque los tipos de letra se guardaban en cajas, también denominadas *chibaletes*. Para componer un texto, el *cajista* ponía las letras, una al lado de otra, en un soporte guía llamado *componedor*. Entre las palabras colocaba tacos de metal y entre los renglones intercalaba finas regletas de metal, llamadas *interlíneas*. Los espacios y los márgenes se justificaban con piezas rectangulares hasta que cada página formara un bloque compacto, llamado *rama*. Toda esta operación se conocía como la *imposición del molde*.

9.4.2. Descripción de los paquetes de PSNFSS

Los paquetes más importantes creados al instalar la colección PSNFSS aparecen en la Tabla 9.2, en la que se muestran las fuentes utilizadas para los tipos de letra romana, sans serif, typewriter y para los símbolos matemáticos (math). Todos los paquetes utilizan las fuentes CM para los símbolos matemáticos, a excepción de *mathptmx* y *mathpazo*, que los reemplazan completamente por fuentes tipográficamente semejantes a Times y Palatino, respectivamente (véanse las secciones 9.4.3 y 9.4.4).

Paquete	romana	sans serif	typewriter	math
bookman	Bookman	Avant Garde	Courier	CM
newcent	New Century Schoolbook	Avant Garde	Courier	CM
chancery	Zapf Chancery	CM	CM	CM
mathptmx	Times	CM	CM	Ptmx
mathpazo	Palatino	CM	CM	Pazo

TABLA 9.2. Principales paquetes de la colección PSNFSS.

Otro paquete importante de la colección es *pifont*, el cual proporciona

símbolos especiales y macros para listas; se describe en la sección 9.4.5. Quedan instalados, además, los paquetes **times**, **palatino** y **mathptm**, pero éstos son considerados obsoletos y sólo se incluyen por compatibilidad con documentos ya existentes.

Se accede a los paquetes de la Tabla 9.2 en la forma usual, por medio de `\usepackage{...}`. En la Figura 9.1 de la página siguiente se puede apreciar la combinación de fuentes de los paquetes **bookman** y **newcent**, y su contraste con las fuentes CM de L^AT_EX. El texto procesado es el siguiente:

```
Una frase en letra normal. Ligaduras: ff, fi, fl, ffi, ffl.\\
Acentos y puntuación: áéíóúñ ---guiones--- !'Ojo! ?'Vió?\\
"Comillas inglesas" y \guillemotleft comillas
francesas\guillemotright \\
\textit{Una frase en letra cursiva o itálica (it).}\\
\textsf{Una frase en letra sans serif (sf).}\\
\textsl{Una frase en letra inclinada (sl).}\\
\textbf{Una frase en letra negrilla (bf).}\\
\textsc{Una frase en letra versalitas (sc).}\\
\texttt{Una frase en letra mono-espaciada (tt)}.\\
```

9.4.3. El paquete **mathptmx**

Al cargar este paquete, por medio de `\usepackage{mathptmx}`, Times se convierte en la fuente principal del documento, pero se mantienen las fuentes CM para los tipos de letra sans serif y typewriter. Los símbolos matemáticos provienen de la fuente virtual 'Ptmx'⁶. Un antecesor de este paquete se distribuye como **mathptm**, considerado ahora obsoleto.

La diferencia más notoria entre los símbolos de **mathptmx** y los CM son las letras griegas. Compárese:

CM:	$\alpha \beta \gamma \delta \epsilon \eta \iota \kappa \lambda \mu \nu \psi \phi \pi \rho \sigma \tau \theta \upsilon \chi \xi \zeta \omega$
mathptmx:	$\alpha \beta \gamma \delta \varepsilon \eta \iota \kappa \lambda \mu \nu \psi \phi \pi \rho \sigma \tau \theta \upsilon \chi \xi \zeta \omega$

CM:	$\Gamma \Delta \Lambda \Psi \Phi \Pi \Sigma \Theta \Upsilon \Xi \Omega$
mathptmx:	$\Gamma \Delta \Lambda \Psi \Phi \Pi \Sigma \Theta \Upsilon \Xi \Omega$

El paquete **mathptmx** se puede cargar en la forma

```
\usepackage[slantedGreek]{mathptmx}
```

con lo cual las letras griegas mayúsculas se obtienen inclinadas:

```
 $\Gamma \Delta \Lambda \Psi \Phi \Pi \Sigma \Theta \Upsilon \Xi \Omega$ 
```

Los comandos `\upDelta` y `\upOmega` producen Δ y Ω , respectivamente, incluso si se usa la opción `[slantedGreek]`.

⁶Diseñada por Alan Jeffrey, Sebastian Rathz y Ulrik Vieth.

Tipos de letra de las fuentes CM de T_EX

Una frase en letra normal. Ligaduras: ff, fi, fl, ffi, ffl.
Acentos y puntuación: áéíóúñÑ —guiones— ¡Ojo! ¿Vió?
“Comillas inglesas” y «comillas francesas»
Una frase en letra cursiva o itálica (it).
Una frase en letra sans serif (sf).
Una frase en letra inclinada (sl).
Una frase en letra negrilla (bf).
UNA FRASE EN LETRA VERSALITAS (SC).
Una frase en letra mono-espaciada (tt).

Tipos de letra del paquete bookman

Una frase en letra normal. Ligaduras: ff, fi, fl, ffi, ffl.
Acentos y puntuación: áéíóúñÑ —guiones— ¡Ojo! ¿Vió?
“Comillas inglesas” y «comillas francesas»
Una frase en letra cursiva o itálica (it).
Una frase en letra sans serif (sf).
Una frase en letra inclinada (sl).
Una frase en letra negrilla (bf).
UNA FRASE EN LETRA VERSALITAS (SC).
Una frase en letra mono-espaciada (tt).

Tipos de letra del paquete newcent

Una frase en letra normal. Ligaduras: ff, fi, fl, ffi, ffl.
Acentos y puntuación: áéíóúñÑ —guiones— ¡Ojo! ¿Vió?
“Comillas inglesas” y «comillas francesas»
Una frase en letra cursiva o itálica (it).
Una frase en letra sans serif (sf).
Una frase en letra inclinada (sl).
Una frase en letra negrilla (bf).
UNA FRASE EN LETRA VERSALITAS (SC).
Una frase en letra mono-espaciada (tt).

FIGURA 9.1. Tipos de letra producidos por las fuentes CM de T_EX, y por las fuentes de los paquetes bookman y newcent.

9.4.4. El paquete **mathpazo**

Al cargar este paquete, por medio de `\usepackage{mathpazo}`, Palatino se convierte en la fuente principal del documento, pero se mantienen las fuentes CM para los tipos de letra sans serif y typewriter. Los símbolos matemáticos provienen de la fuente virtual ‘Pazo’⁷. Un antecesor de este paquete se distribuye como **mathpple**, pero **mathpazo** se considera superior por sus nuevas opciones y símbolos.

Las letras griegas difieren bastante de las CM; compárese:

CM:	$\alpha \beta \gamma \delta \epsilon \eta \iota \kappa \lambda \mu \nu \psi \phi \pi \rho \sigma \tau \theta \upsilon \chi \xi \zeta \omega$
mathpazo:	$\alpha \beta \gamma \delta \epsilon \eta \iota \kappa \lambda \mu \nu \psi \phi \pi \rho \sigma \tau \theta \upsilon \chi \xi \zeta \omega$

CM:	$\Gamma \Delta \Lambda \Psi \Phi \Pi \Sigma \Theta \Upsilon \Xi \Omega$
mathpazo:	$\Gamma \Delta \Lambda \Psi \Phi \Pi \Sigma \Theta \Upsilon \Xi \Omega$

El paquete **mathpazo** se puede cargar en la forma

```
\usepackage[slantedGreek]{mathpazo}
```

con lo cual las letras griegas mayúsculas se obtienen inclinadas:

```
\Gamma \Delta \Lambda \Psi \Phi \Pi \Sigma \Theta \Upsilon \Xi \Omega
```

Los comandos `\upDelta` y `\upOmega` producen Δ y Ω , respectivamente, incluso si se usa la opción `[slantedGreek]`.

Todas las letras y números bajo el alcance de `\mathbf{...}` aparecen en negrilla itálica, incluyendo las letras griegas. Por otro lado, el comando `\mathbb{...}` (sección 4.24) solamente está definido para C, I, N, Q, R y Z, para las cuales se obtienen los símbolos **C**, **I**, **N**, **Q**, **R** y **Z**, respectivamente. Si se carga el paquete **mathpazo** en la forma

```
\usepackage[noBBpl]{mathpazo}
```

los comandos `\mathbb{...}` producen los símbolos CM y no los anteriormente exhibidos.

El paquete **mathpazo** también posee el comando

```
\p{euro}
```

para el símbolo monetario europeo, el euro: €. En su versión inclinada el símbolo es €, obtenido a partir de `\textit{\p{euro}}`.

Para una comparación adicional entre los tipos de letra, en la página siguiente aparece el enunciado del Teorema de Taylor, impreso primero con las fuentes CM de TeX y luego con las fuentes de los paquetes **mathptmx** y **mathpazo**.

⁷Diseñada por Diego Puga.

Texto matemático en las fuentes CM de TEX

Teorema de Taylor. Si f es derivable en todo punto de un dominio Ω , f es analítica en Ω . Sea $a \in \Omega$ y $\overline{D}(a, r) \subseteq \Omega$, con $r > 0$. Entonces

$$f^{(n)}(a) = \frac{n!}{2i\pi} \int_{\alpha} \frac{f(\zeta)}{(\zeta - a)^{n+1}} d\zeta, \quad \alpha(t) = a + re^{2\pi it},$$

y

$$f(z) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(a)(z - a)^n, \quad z \in D(a, R),$$

donde $R = d(a, \mathbb{C} \setminus \Omega)$. Esta es la *serie de Taylor alrededor de a*.

Texto matemático en las fuentes del paquete mathptmx

Teorema de Taylor. Si f es derivable en todo punto de un dominio Ω , f es analítica en Ω . Sea $a \in \Omega$ y $\overline{D}(a, r) \subseteq \Omega$, con $r > 0$. Entonces

$$f^{(n)}(a) = \frac{n!}{2i\pi} \int_{\alpha} \frac{f(\zeta)}{(\zeta - a)^{n+1}} d\zeta, \quad \alpha(t) = a + re^{2\pi it},$$

y

$$f(z) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(a)(z - a)^n, \quad z \in D(a, R),$$

donde $R = d(a, \mathbb{C} \setminus \Omega)$. Esta es la *serie de Taylor alrededor de a*.

Texto matemático en las fuentes del paquete mathpazo

Teorema de Taylor. Si f es derivable en todo punto de un dominio Ω , f es analítica en Ω . Sea $a \in \Omega$ y $\overline{D}(a, r) \subseteq \Omega$, con $r > 0$. Entonces

$$f^{(n)}(a) = \frac{n!}{2i\pi} \int_{\alpha} \frac{f(\zeta)}{(\zeta - a)^{n+1}} d\zeta, \quad \alpha(t) = a + re^{2\pi it},$$

y

$$f(z) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(a)(z - a)^n, \quad z \in D(a, R),$$

donde $R = d(a, \mathbb{C} \setminus \Omega)$. Esta es la *serie de Taylor alrededor de a*.

FIGURA 9.2. Ejemplo de un texto matemático impreso con las fuentes CM de TEX, y con las fuentes de los paquetes mathptmx y mathpazo.

9.4.5. El paquete pifont

El paquete **pifont** también hace parte de la colección PSNFSS y proporciona los símbolos de la fuente Zapf Dingbats, mostrados en la Tabla 9.3. Para usar estos símbolos en un documento L^AT_EX hay que escribir la instrucción `\usepackage{pifont}` en el preámbulo, y recurrir al comando `\ding{código}`. Así, `\ding{48}` da lugar a , `\ding{41}` produce , `\ding{94}` produce el símbolo , etc.

32	33	34	35	36	37	38
39	40	41	42	43	44	45
46	47	48	49	50	51	52
53	54	55	56	57	58	59
60	61	62	63	64	65	66
67	68	69	70	71	72	73
74	75	76	77	78	79	80
81	82	83	84	85	86	87
88	89	90	91	92	93	94
95	96	97	98	99	100	101
102	103	104	105	106	107	108
109	110	111	112	113	114	115
116	117	118	119	120	121	122
123	124	125	126			
	161	162	163	164	165	166
167	168	169	170	171	172	173
174	175	176	177	178	179	180
181	182	183	184	185	186	187
188	189	190	191	192	193	194
195	196	197	198	199	200	201
202	203	204	205	206	207	208
209	210	211	212	213	214	215
216	217	218	219	220	221	222
223	224	225	226	227	228	229
230	231	232	233	234	235	236
237	238	239		241	242	243
244	245	246	247	248	249	250
251	252	253	254			

TABLA 9.3. Símbolos de la fuente PostScript Zapf Dingbats.

El paquete `pifont` tiene dos útiles entornos para listas. El entorno

```
\begin{dinglist}{código}
\item Texto
\item Texto
:
\item Texto
\end{dinglist}
```

es similar a `itemize` pero los items se demarcan con el símbolo de la Tabla 9.3 que tiene el *código* especificado.

Ejemplo Las instrucciones

```
\begin{dinglist}{43}
\item Primer ítem de la lista.
\item Segundo ítem de la lista.
\item Tercer ítem de la lista.
\end{dinglist}
```

dan lugar a la siguiente lista:

Primer ítem de la lista.

Segundo ítem de la lista.

Tercer ítem de la lista.

El entorno

```
\begin{dingautolist}{código}
\item Texto
\item Texto
:
\item Texto
\end{dingautolist}
```

es similar a `enumerate` pero los items se enumeran de uno en uno, a partir del símbolo que tiene el *código* especificado.

Ejemplo Las instrucciones

```
\begin{dingautolist}{202}
\item Primer ítem de la lista enumerada.
\item Segundo ítem de la lista enumerada.
\item Tercer ítem de la lista enumerada.
\end{dingautolist}
```

dan lugar a la siguiente lista:

- ❶ Primer ítem de la lista enumerada.
- ❷ Segundo ítem de la lista enumerada.
- ❸ Tercer ítem de la lista enumerada.

El comando `\dingfill{código}` se asemeja a los comandos L^AT_EX `\hfill`, `\dotfill` y `\hrulefill` (sección 3.23) y se usa para llenar espacio horizontal con copias sucesivas del símbolo que tiene el *código* especificado.

Ejemplo

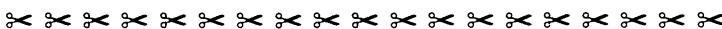
Inicio  medio  fin.

se produce a partir de

```
Inicio \dingfill{235} medio \dingfill{235} fin.
```

El comando `\dingline{código}` da lugar a un renglón compuesto por el símbolo que tiene el *código* especificado, con sangrías a izquierda y a derecha.

Ejemplo La instrucción `\dingline{34}` da lugar a lo siguiente:



9.4.6. Codificación de las fuentes de la colección PSNFSS

Las fuentes de la colección PSNFSS se distribuyen en las codificaciones OT1 y T1, siendo OT1 la codificación utilizada por defecto. Pero si se dispone de las fuentes EC (sección 9.1), es recomendable usar la codificación T1 en todo el documento, cargando el paquete `fontenc` en la forma

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

Accedemos, por ejemplo, a las fuentes del paquete `bookman`, en la codificación T1, por medio de

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

Puesto que los paquetes `bookman`, `newcent` y `chancery` no poseen símbolos matemáticos propios, son necesarios tales símbolos en la codificación T1, proporcionados por las fuentes EC. Esto quiere decir que al cargar los citados paquetes de fuentes en la codificación T1, también se requiere la presencia de las fuentes EC.

9.5. Comandos para cambiar las familias de fuentes en un documento

Como se mencionó en la sección 9.1, el esquema NFSS clasifica las fuentes en tres familias básicas: la familia serif romana, la familia sans serif y la mono-espaciada. LATEX controla estas fuentes con los comandos \rmfamily, \sffamily y \ttfamily, respectivamente (Tabla 9.4). Los valores asignados por defecto para estas familias de fuentes se pueden cambiar con \renewcommand, en el preámbulo del documento.

Familia	Se controla con	Se accede con	Por defecto
Principal (romana)	\rmfamily	\textrm{...}	cmr
Sans serif	\sffamily	\textsf{...}	cmss
Mono-espaciada (typewriter)	\ttfamily	\texttt{...}	cmtt

TABLA 9.4. Familias de fuentes según el esquema NFSS.

Ejemplo Si queremos que la letra principal del documento sea la letra sans serif de la fuente CM escribimos

```
\renewcommand{\rmfamily}{\cmss}
```

en el preámbulo del documento. El cambio de letra se realizará de manera uniforme: \textbf{...} producirá negrita sans serif, \textit{...} producirá itálica sans serif que es, en realidad, la misma letra sans serif inclinada obtenida con \textsl{...}, etc. Los títulos principales (para capítulos, secciones, subsecciones, etc) se obtienen en negrita sans serif, que es lo más apropiado desde el punto de vista tipográfico. De ser necesario, LATEX hace sustituciones de fuentes. Por ejemplo, con \textsc{...} LATEX utiliza versalitas de la familia cmr ya que no existe la variante sc en la fuente cmss.

Ejemplo Los paquetes de la colección PSNFSS hacen precisamente este tipo de redefiniciones para cambiar fuentes. Por ejemplo, en el archivo *bookman.sty*, que define el paquete *bookman*, se encuentran los siguientes comandos:

```
\renewcommand{\rmdefault}{pbk}
\renewcommand{\sfdefault}{pag}
\renewcommand{\ttdefault}{pcr}
```

Siguiendo la información de la Tabla 9.1, esto significa que la letra romana principal proviene de la fuente Bookman (familia pbk), la letra sans-serif proviene de la fuente Avant Garde (familia pag) y la letra mono-espaciada pertenece a la fuente Courier (familia pcr).

9.6. Acceso a una fuente con los comandos \newfont y \DeclareFixedFont

En la sección 3.5 se mencionó que \DeclareTextFontCommand permite definir comandos para combinaciones particulares de tamaño y tipo de letra. En esta sección presentamos dos instrucciones relacionadas que permiten acceder a una fuente, ya sea por su nombre o por sus atributos. Si se conoce el *nombre* de la fuente, se puede asignar un \comando para acceder a ésta, usando alguna de las instrucciones

```
\newfont{\comando}{nombre at tama o}
\newfont{\comando}{nombre scaled factor}
```

La fuente cuyo *nombre* se especifica debe estar f sicamente presente, como fuente real o virtual, junto con su archivo ‘—.tfm’ y demás archivos subsidiarios necesarios (v ase la secci n 9.2)⁸. No se escriben extensiones en el *nombre* de la fuente. En el primero de los dos comandos anteriores, el *tama o* se indica en puntos (unidades pt), y en el segundo, el n mero deseado como aumento a escala se multiplica por 1000 para obtener el *factor* correspondiente.

En este ejemplo consideramos la fuente de s mbolos cmsy10, cuya tabla de caracteres se exhibe en la Figura 9.3. Asignamos el comando \funo a la fuente cmsy10 en el tama o 14pt por medio de

```
\newfont{\funo}{cmsy10 at 14pt}
```

Asignamos el comando \fdos a la fuente cmsy10, aumentada 2.5 veces, por medio de

```
\newfont{\fdos}{cmsy10 scaled 2500}
```

Asignainos el comando \ftres a la fuente cmsy10, reducida al 70 % de su tama o, por medio de

```
\newfont{\ftres}{cmsy10 scaled 700}
```

El comando \newfont se usa principalmente para acceder a s mbolos particulares de una fuente, tal como se explica en la secci n 9.7.

Si se conocen todos los atributos de una fuente, seg n el esquema NFSS (secci n 9.1), tambi n se puede asignar un \comando para acceder a \'esta, usando la instrucci n

```
\DeclareFixedFont{\comando}{codif.}{fam.}{serie}{var.}{tama o}
```

⁸Examinando la lista de archivos *.tfm, se puede saber cu les fuentes se encuentran localmente instaladas.

La instrucción \comando definida con \newfont o \DeclareFixedFont es una declaración global para cambio de fuente y su alcance se delimita con corchetes exteriores: {\comando ...}, en forma similar a {\it ...}, {\sc ...}, etc.

Ejemplo Con \DeclareFixedFont se puede acceder a las fuentes de la colección PSNFSS porque conocemos todos sus atributos (Tabla 9.1). Para acceder a la fuente Zapf Chancery, por ejemplo, en el tamaño de 13 pt, podemos asignar el comando \zcal y definir:

```
\DeclareFixedFont{\zcal}{OT1}{pzc}{m}{it}{13pt}
```

El comando \zcal actúa entonces como comando de cambio de fuente y permite escribir porciones aisladas del documento en la fuente invocada. Así, al escribir

```
\begin{quote}
{\zcal Si las leyes de la mecánica son válidas en un sistema
coordenado, entonces también se cumplen en cualquier sistema
coordenado que se mueva uniformemente con relación al primero.}
\end{quote}
```

obtenemos

*Si las leyes de la mecánica son válidas en un sistema coordinado,
entonces también se cumplen en cualquier sistema coordinado que
se mueva uniformemente con relación al primero.*

9.7. Acceso a los símbolos de una fuente

El archivo nfssfont.tex, incluido en las distribuciones de LATEX 2_E, se puede utilizar para observar todos los caracteres de una fuente particular y para realizar diversos tests sobre la fuente. Al procesar nfssfont.tex como un documento LATEX normal, el programa pregunta por la fuente que se desea examinar:

Name of the font to test =

El usuario escribe la fuente, sin extensión alguna; por ejemplo, cmsy10. La fuente misma debe estar físicamente presente, ya sea como fuente real o virtual. El programa responde a continuación:

Now type a test command (\help for help):)

Se pueden realizar varios tests; la lista de ellos se obtiene escribiendo `\help`. Con `\table` se obtiene la tabla de caracteres de la fuente. Para poner a prueba otra fuente se escribe `\init` y para finalizar `\stop` o `\bye`.

Ejemplo Procesamos el archivo `nfssfont.tex` con la fuente `cmsy10` y, respondiendo de manera interactiva, obtenemos:

```
Name of the font to test = cmsy10
Now type a test command (\help for help):)
*\table
*\stop
```

El archivo `nfssfont.dvi` muestra entonces la tabla de caracteres de la fuente (Figura 9.3).

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	-	.	x	*	÷	◊	±	⋮	"0x
'01x	⊕	⊖	⊗	∅	⊙	○	◦	•	"1x
'02x	×	≡	≤	≥	≤	≥	≤	≥	"2x
'03x	~	≈	⊂	⊃	≪	≫	≺	≻	"3x
'04x	←	→	↑	↓	↔	↗	↘	≈	"4x
'05x	⇐	⇒	↑↑	↓↓	↔↔	↖↖	↙↙	×	"5x
'06x	‘	∞	€	϶	△	▽	/	,	"6x
'07x	∀	∃	¬	∅	ℝ	ℙ	⊤	⊥	"7x
'10x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"10x
'11x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"11x
'12x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"12x
'13x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"13x
'14x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"14x
'15x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"15x
'16x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"16x
'17x	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	ℳ	"17x
	"8	"9	"A	"B	"C	"D	"E	"F	

FIGURA 9.3. Archivo `nfssfont.dvi`, reducido al 75 % de su tamaño real, que muestra la tabla de caracteres de la fuente `cmsy10`.

Se accede a los símbolos de una fuente determinada, tal como `cmsy10`, con comandos específicos, similares a los exhibidos en las tablas de la sección 4.3, pero L^AT_EX posee el comando `\symbol{código}` para acceder a un símbolo particular. Su sintaxis es

`\symbol{código}`

donde el `código` es el número octal (en base 8) precedido por ', o el número hexadecimal (en base 16) precedido por ", mostrados al procesar

el archivo `nfssfont.tex`. En la tabla de caracteres de la fuente `cmsy10` (Figura 9.3), por ejemplo, el símbolo ♣ tiene código octal '174 y código hexadecimal "7C. El símbolo □ tiene código octal '164 y código hexadecimal "74. Para poder usar estos códigos y el comando `\symbol`, hay que asignar primero un `\comando` para la fuente deseada, por medio de `\newfont` o `\DeclareFixedFont`, como se explicó en la sección 9.6. Ilustramos el procedimiento en el siguiente ejemplo.

Ejemplo Vamos a asignar el comando `\fuentea` a la fuente `cmsy10`, en el tamaño 11 puntos, por medio de:

```
\newfont{\fuentea}{cmsy10 at 11pt}
```

La instrucción `\fuentea` así definida es una declaración global para cambio de fuente y su alcance se delimita con corchetes exteriores: `{\fuentea ...}`. Podemos ahora usar `\symbol` y los códigos octales⁹ mostrados en la Figura 9.3 para acceder a símbolos particulares de esta fuente: si escribimos `{\fuentea \symbol{'174}}` obtenemos ♣ y el símbolo ∃ se puede obtener a partir de `{\fuentea \symbol{'071}}`.

También podemos cargar la fuente `cmsy10` en otros tamaños, por ejemplo, al doble de su tamaño original, definiendo

```
\newfont{\fuenteb}{cmsy10 scaled 2000}.
```

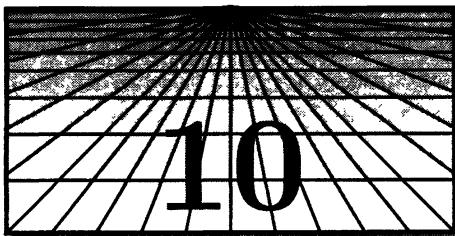
Al escribir

```
\centerline{\fuenteb \symbol{'174} \symbol{'175} %
           \symbol{'176} \symbol{'177}}
```

obtenemos:



⁹Recomendamos usar preferiblemente códigos octales porque el símbolo " usado para los códigos hexadecimales puede ocasionar conflictos con el paquete `babel`.



Documentos **L****A****T****E****X** interactivos

En la actualidad, la publicación no se limita a documentos impresos en papel. Con el auge de la red mundial Internet (también conocida como *World Wide Web* o simplemente *Web*) ha surgido la necesidad de documentos electrónicos o virtuales, caracterizados principalmente por su interactividad. El lector es guiado a través de enlaces (en inglés, *links*) que lo conducen a otros sitios del documento, a otros documentos o a otras páginas *Web*. El texto que contiene estos enlaces interactivos se conoce como *texto enriquecido* o *hipertexto*.

Texto, hipertexto, color, sonido, imágenes estáticas o en movimiento, hacen parte del cosmopolita, y a veces caótico, mundo de la Internet. Las páginas *Web* están usualmente escritas en el lenguaje HTML (*Hypertext Markup Language*), con el cual el formato no es rígido: el usuario o lector pueden alterar el color, tipo y tamaño de las fuentes, así como la amplitud horizontal de los renglones.

Para el usuario **L****A****T****E****X** hay varias maneras de “transformar” un archivo en un documento interactivo. El enfoque más directo consiste en convertir el archivo **L****A****T****E****X** al lenguaje HTML. Dos populares programas de dominio público, encaminados en tal dirección, son **L****A****T****E****X****2****H****T****M****L** y **T****E****X****4****ht**. Con ellos, la conversión a HTML no es del todo simple y automática: los símbolos, fórmulas, tablas e imágenes que no se puedan traducir directamente son convertidos en imágenes, en formato GIF, y el documento requiere, por lo general, un trabajo de edición posterior que exige proficiencia en el lenguaje HTML por parte del usuario.

Otro intento relacionado es **techexplorer**, *plug-in* de IBM para los navegadores **Netscape Navigator** e **Internet Explorer**, el cual es capaz de interpretar directamente comandos **L****A****T****E****X** y desplegarlos en la pantalla. No obstante su indudable utilidad, **techexplorer** posee serias limitaciones para el usuario **L****A****T****E****X**: por un lado sobresale su carácter eminentemente comercial, y por otro, su incompatibilidad con la filosofía de los “paquetes”, predominante en **L****A****T****E****X** 2 ε . Los comandos **\documentclass** y **\usepackage** simplemen-

te son ignorados por `techexplorer`, restringiendo el uso de este *plug-in* a documentos LATEX sencillos.

Un enfoque completamente diferente a los anteriores consiste en producir documentos PDF, notables por su rápido acceso, su seguridad, su capacidad interactiva y su portabilidad. En el presente libro sólo enfatizaremos esta manera de producir documentos LATEX interactivos; las principales herramientas disponibles para tal propósito, el paquete `hyperref` y el programa `pdflatex`, se describen en este capítulo.

10.1. Archivos en formato PDF

En 1993 Adobe Systems¹ difundió el formato PDF (*Portable Format Document*) para documentos electrónicos, el cual combina texto normal e hipertexto interactivo, junto con elementos de compresión y seguridad. A diferencia de los documentos HTML, en un documento PDF el formato de página (márgenes, fuentes, espaciamiento entre renglones, colores) es completamente rígido, lo cual permite pasar de la versión electrónica a la versión impresa preservando el aspecto visual del documento.

El formato PDF se ha convertido en uno de los preferidos para la publicación electrónica en los círculos académicos y científicos ya que es más apropiado que HTML para documentos con muchas expresiones matemáticas o documentos en los que la escogencia de símbolos y fuentes es importante. También contribuye a la popularidad de PDF el hecho de que Adobe distribuyé gratuitamente, y para todas las plataformas computacionales, el visor de archivos PDF: el programa **Acrobat Reader**.

Para el usuario LATEX, las dos maneras más convenientes de generar PDF son las siguientes:

- (1) `.tex → .dvi → .pdf` Se puede seguir este camino usando el convertidor `dvipdfm`² (ejecutable `dvipdfm.exe`), usualmente incluido en implementaciones TEX. Un archivo `doc.dvi` se convierte en `doc.pdf` al ejecutar desde la línea de comandos:

```
dvipdfm doc.dvi
```

En algunas implementaciones se dispone de un botón `dvi~pdf` para llevar a cabo esta conversión con un simple *click*.

¹El lenguaje PostScript (sección 7.1) es también una creación de Adobe Systems.

²Escrito por Mark A. Wicks.

- (2) **.tex → .pdf** Se sigue este camino usando el programa **pdftEX**, que es una variación del programa **T_EX**, especialmente diseñado para producir documentos PDF en lugar de archivos **dvi**. La versión para documentos **L_AT_EX** se denomina **pdflatEX**. Aunque se considera aún en etapa de desarrollo, **pdftEX** se distribuye con algunas implementaciones **T_EX**. En aquellas implementaciones en las que **pdflatEX** viene pre-instalado, el programa se ejecuta normalmente desde la línea de comandos, en la forma

```
pdflatex doc.tex
```

La manera concreta de invocar **pdflatEX** depende, no obstante, de la instalación y la configuración locales; en algunos casos se dispone de un botón expreso para esta operación.

pdftEX posee comandos adicionales, entre ellos, comandos para controlar los aspectos propios del formato PDF, como nivel de compresión, formato de página, tabla de contenido interactiva o panel de *Marcadores* (*Bookmarks*, en inglés), notas interactivas, etc. Algunos de tales comandos, e información adicional, se presentan en la sección 10.3.

Para producir documentos **L_AT_EX** interactivos, una herramienta indispensable es el paquete **hyperref**, el cual es compatible con los dos procedimientos mencionados arriba. Con este paquete, las referencias cruzadas producidas con **\ref**, **\pageref** y **\cite** se transforman en enlaces interactivos en el archivo **dvi** y en el archivo PDF (si el documento se convierte al formato PDF). Si el documento posee un índice alfabético, las páginas indicadas para cada ítem se pueden activar como enlaces. Además, la capacidad de navegación se incrementa con comandos para nuevos enlaces, tales como vínculos a documentos externos, a programas o a páginas *Web*. El paquete **hyperref** se describe en la sección 10.2.

Tanto **dvipdfm** como **pdftEX** poseen también sus propios comandos primarios para establecer enlaces internos y externos, pero su sintaxis es bastante incómoda. Por consiguiente, para crear un documento PDF interactivo, **doc.pdf**, a partir de un documento fuente **L_AT_EX**, **doc.tex**, recomendamos los siguientes pasos:

1. Usar el paquete **hyperref** en **doc.tex** para definir todos los enlaces y referencias cruzadas, internos y externos.
2. Procesar **doc.tex** con **pdflatEX**. Se pueden incluir en el documento, aunque no es necesario, comandos primarios de **pdftEX** para efectos específicos (véanse al respecto las secciones 10.3 y 10.4). El uso de las fuentes AE (sección 9.3) es también recomendable.

10.2. El paquete `hyperref`

En esta sección describiremos el paquete `hyperref`³ y sus opciones más importantes. Se accede a este paquete en la forma usual, escribiendo

```
\usepackage{hyperref}
```

en el preámbulo del documento. Puesto que `hyperref` redefine varios comandos importantes de LATEX, se recomienda que sea el último paquete cargado. Es también aconsejable asegurarse de que no haya archivos auxiliares del documento (tales como archivos `.aux`, `.toc`, `.lof`, `.bbl`, o `.idx`), creados antes de cargar el paquete.

`hyperref` es compatible con la mayoría de los paquetes LATEX mencionados en el presente libro; una notable excepción es el paquete gráfico **PSTricks** (Capítulo 11).

Si se procesa el documento fuente como un documento LATEX normal, se obtiene un archivo `dvi` interactivo: los sitios en los que aparecen las referencias `\ref`, `\pageref` y `\cite` se convierten en enlaces activos, así como las marcas para notas al pie de página. Esta interactividad depende, por supuesto, de que el visualizador (o más exactamente, el manejador de pantalla para archivos `dvi`) sea compatible con `hyperref`.

El documento fuente también se puede procesar con `pdflATEX`, obteniéndose así un documento PDF interactivo, en lugar de un archivo `dvi`. En tal caso, es recomendable cargar el paquete con la opción `pdftex`,

```
\usepackage[pdftex]{hyperref}
```

para que `hyperref` utilice el manejador más adecuado. Adicionalmente, si la intención es procesar el documento con `pdflATEX`, el paquete `graphicx` de LATEX 2 ϵ se debe cargar también con la opción `pdftex`, en la forma

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

Hay que advertir que el tamaño de los archivos finales PDF es varias veces mayor que el de los correspondientes archivos `dvi`, y que si se escoge un alto nivel de compresión, el tiempo de procesamiento con `pdflATEX` es sensiblemente más extenso que para documentos LATEX normales (sobre las opciones de compresión véase la sección 10.3.1).

Un archivo PDF generado con `pdflATEX` puede ser posteriormente editado con el programa **Adobe Acrobat** (editor de archivos PDF, antes conocido como **Acrobat Exchange**). No obstante, debe tenerse presente que **Adobe**

³Escrito por Sebastian Rahtz; el paquete y su documentación se incluyen en el CD adjunto. La documentación contiene una guía sucinta e incompleta: el archivo `manual.pdf`.

Acrobat, a diferencia de Acrobat Reader, es un programa comercial y no se distribuye gratuitamente. Es reconfortante saber que `pdflatex` y el paquete `hyperref` también permiten modificar directa (¡y gratuitamente!) muchas de las características del documento PDF final, como se mostrará a lo largo del presente capítulo.

10.2.1. Opciones de `hyperref`

El paquete `hyperref` tiene numerosas opciones para modificar el aspecto final del documento; las más importantes se describen a continuación. Las llamadas opciones booleanas admiten los valores `true` o `false`. Las opciones también se pueden escribir con el comando `\hypersetup`, en la forma:

`\hypersetup{opciones}`

después de cargar el paquete. En la siguiente página se presentan ejemplos que ilustran el acceso a `hyperref`, sujeto a diversas opciones. En el CD adjunto se incluye, además, un ejemplo de un documento `LATEX` modelo, en el que se han usado el paquete `hyperref` y algunos comandos primarios de `pdflatex` (véase la sección 10.4).

Opciones generales. El paquete `hyperref` preserva el formato de página declarado con `\documentclass` y `\pagestyle`, incluyendo los encabezados de las páginas. También se preservan los encabezados y pies de página generados con el paquete `fancyhdr` (sección 13.1). Se dispone, además, de las opciones de la Tabla 10.1.

<code>draft</code>	Enlaces e hipertexto desactivados; por defecto: <code>false</code> .
<code>debug</code>	Mensajes adicionales de diagnóstico en el archivo <code>.log</code> ; por defecto: <code>false</code> .
<code>a4paper</code>	Tamaño de página: <code>210mm × 297mm</code> .
<code>letterpaper</code>	Tamaño de página: <code>8.5in × 11in</code> .
<code>legalpaper</code>	Tamaño de página: <code>8.5in × 14in</code> .

TABLA 10.1. Opciones generales del paquete `hyperref`.

Opciones para enlaces. La apariencia de los enlaces internos producidos por `hyperref` depende del tipo de compilación aplicado. Si se trata de `LATEX`, el texto de los enlaces estará subrayado en el archivo `dvi`; si se aplica `pdflatex`, en cambio, el texto aparecerá dentro de un marco rojo (o verde para las referencias bibliográficas); pero si convertimos un archivo `dvi` en uno PDF a través de `dvipdfm`, el marco será azul claro (`cyan`).

El usuario puede cambiar estos estilos con la opción `colorlinks`, con la cual el texto aparece en color rosado (*magenta*); con `pdfLATEX` los enlaces pierden el marco, pero el texto es rojo (o verde para las citas bibliográficas). Aplicando `dvipdfm`, finalmente, los enlaces tienen marco azul claro, pero además el texto es púrpura.

Los colores obtenidos al usar `colorlinks` se pueden cambiar por medio de las opciones `linkcolor` y `citecolor`; para estas y otras opciones remitimos al lector a la Tabla 10.2.

En la Tabla 10.4 aparecen las opciones de `hyperref` para activar enlaces en la bibliografía y en el índice alfabético. Aparte de todas estas opciones, `hyperref` tiene varios comandos especiales para enlaces (véase la sección 10.2.2).

Opciones específicas para documentos PDF. Las opciones más importantes de `hyperref` para documentos PDF se presentan en la Tabla 10.3.

Ejemplo La instrucción

```
\usepackage[pdftex,letterpaper,bookmarksnumbered,%
           colorlinks,backref]{hyperref}
```

carga el paquete `hyperref` sujeto a algunas de las opciones listadas en las Tablas 10.1 a 10.4.

Ejemplo Es importante tener presente que, al cargar un paquete con `\usepackage`, LATEX lee las opciones indicadas, antes de trasladarlas al paquete mismo, lo cual puede generar problemas en el caso de que las opciones contengan símbolos acentuados o comandos LATEX. Por tal razón, si se usan las opciones de información para los campos de la plantilla Document Info de Acrobat Reader (Tabla 10.3), es mejor establecer las opciones de `hyperref` por medio de `\hypersetup{...}`.

A continuación se ilustra cómo se carga `hyperref` con la opción genérica `pdftex` y luego se escriben las demás opciones en el argumento de `\hypersetup`. Obsérvese que los campos de la forma `pdf...` contienen acentos y que `pdfauthor` contiene el comando LATEX `\textcopyright` para el símbolo ©.

```
\usepackage[pdftex]{hyperref}
\hypersetup{bookmarksopen,bookmarksnumbered,colorlinks,%
           linkcolor=blue,legalpaper,pagebackref,%
           pdftitle=Las asombrosas propiedades del neutrón,%
           pdfauthor=Jimmy Neutrón \textcopyright,pdfsubject=física%
           atómica, pdfkeywords=partículas atómicas; neutrones}
```

colorlinks	Opción booleana; por defecto: false . Al escribir la opción colorlinks=true (o simplemente colorlinks), los enlaces adquieren colores, en la forma explicada en la página 294.
linkcolor	Establece el color de los enlaces internos cuando se escoge la opción colorlinks . Por defecto: red (rojo).
citecolor	Color para las citas bibliográficas producidas con \cite . Por defecto: green (verde).
pagecolor	Color de los enlaces a páginas establecidos con \pageref . Por defecto: red .
filecolor	Color de los enlaces a archivos locales, establecidos con \href{file:...} , \href{run:...} y \hyperref{file:...} (sección 10.2.2). Por defecto: cyan (azul claro).
urlcolor	Color de los enlaces a direcciones URL, establecidos con \url y \href (sección 10.2.2). Por defecto: magenta .
menucolor	Color de los enlaces a las opciones de menú de Acrobat Reader producidas con \Acrobatmenu (sección 10.2.4). Por defecto: red .

TABLA 10.2. Opciones de *hyperref* para los colores de los enlaces. Se indican los valores por defecto para pdfLATEX.

bookmarks	Opción booleana; por defecto: true . Para el panel <i>Marcadores</i> o tabla de contenido (<i>Bookmarks</i>) característico de los archivos PDF. Se genera con la información de los comandos divisionales \chapter , \section , etc. Para prescindir de los <i>Marcadores</i> (<i>Bookmarks</i>) se debe cargar el paquete con la opción bookmarks=false .
bookmarksopen	Expande el árbol de <i>Marcadores</i> (<i>Bookmarks</i>). Opción booleana; por defecto: false .
bookmarksnumbered	Coloca números de capítulos y secciones en el panel de <i>Marcadores</i> . Opción booleana; por defecto: false .
pdftitle	Para el campo Title en la plantilla Document Info de Acrobat Reader.
pdfauthor	Para el campo Author en la plantilla Document Info de Acrobat Reader.
pdfsubject	Para el campo Subject en la plantilla Document Info de Acrobat Reader.
pdfkeywords	Para el campo Keywords en la plantilla Document Info de Acrobat Reader. Las palabras clave (<i>keywords</i>) se separan con punto y coma (;) o con espacios en blanco, no con comas.

TABLA 10.3. Opciones de *hyperref* para documentos PDF.

backref	En la bibliografía final se añaden enlaces a las secciones en las que aparecen los comandos <code>\cite</code> . Esta opción funciona adecuadamente sólo si se deja un renglón en blanco después del texto completo de cada <code>\bibitem</code> . Opción booleana; por defecto: <code>false</code> .
pagebackref	Funciona como <code>backref</code> pero los enlaces remiten a las páginas en los que aparecen los comandos <code>\cite</code> . Opción booleana; por defecto: <code>false</code> .
hyperindex	En el índice alfabético las páginas indicadas para cada ítem se convierten en enlaces. Se advierte que esta opción no es muy estable; el uso del modificador <code> </code> en los comandos <code>\index</code> (sección 6.9), por ejemplo, puede conducir a mensajes de error. Opción booleana; por defecto: <code>false</code> .

TABLA 10.4. Opciones de `hyperref` para otros enlaces.

10.2.2. Enlaces internos y externos

Para crear enlaces adicionales a los producidos con los comandos `\ref`, `\pageref` y `\cite` de LATEX, el paquete `hyperref` dispone de otros comandos:

\nameref Se usa como `\ref` pero crea un enlace con el *nombre* del capítulo o sección. Debe aplicarse solamente a las divisiones del documento (partes, capítulos, secciones, etc.).

\hypertarget{clave}{texto} Es análogo al comando `\label` de LATEX, pero se puede usar para una referencia cruzada a un *texto* arbitrario del documento. Sólo sirve para enlaces internos.

\hyperlink{clave}{expresión} Es análogo al comando `\ref` de LATEX; establece un enlace interno cuyo destino debe ser definido previamente con `\hypertarget`. La *expresión* aparece resaltada como hipertexto.

\href{file:archivo}{expresión} Crea un enlace a un *archivo* externo; la *expresión* aparece resaltada como hipertexto. El *archivo* es buscado en el mismo subdirectorio en el que aparece el documento fuente; si está en otra carpeta, la ruta completa debe ser indicada, usando la sintaxis propia del sistema operativo local.

\href{run:archivo}{expresión} Similar a `\href{file:...}` pero el enlace inicia el programa asociado al *archivo* especificado; éste puede ser un ejecutable u otro tipo de archivo.

\hyperdef{categoría}{clave} Análogo a \hypertarget pero para enlaces externos: define un enlace cuyo destino es la página en la que aparece la instrucción \hyperdef; *clave* es el nombre del enlace. El argumento *categoría* sirve para clasificar los destinos con nombres escogidos por el usuario, como ‘ecuación’, ‘teorema’, ‘nota’, etc.

\hyperref[file:archivo]{categoría}{clave}{expresión} Establece un enlace cuyo destino debe ser definido previamente con \hyperdef en el *archivo* especificado. Si éste se encuentra en la misma carpeta, el prefijo ‘file:’ no es necesario. La *expresión* aparece resaltada como hipertexto. Se pueden definir enlaces internos de este tipo escribiendo \hyperref{}{categoría}{clave}{expresión}.

\url{dirección URL}{ Crea un enlace; un *click* sobre la *dirección URL* inicia el navegador configurado en el sistema local, y conduce a la página Web correspondiente.

\href{dirección URL}{expresión} Similar a la instrucción anterior pero la *expresión* queda activada como enlace para iniciar el navegador.

Ejemplo La instrucción

\href{run:C:/Program Files/LView/lviewpro.exe}{EDITAR} crea el enlace ‘EDITAR’; un *click* sobre éste inicia el programa gráfico *LView Pro* cuyo ejecutable (*lviewpro.exe*) está ubicado en el subdirectorio local *C:/Program Files/LView*.

Ejemplo La instrucción \hyperdef{teor}{cayley}, que aparece en el documento *doc1.tex* (en el mismo párrafo en el que se enuncia el Teorema de Cayley), establece el destino de un enlace cuyo nombre es *cayley*. El archivo *doc1.tex* se procesa con pdf \LaTeX para producir el documento *doc1.pdf*. Por otro lado, en el documento *doc2.tex*, ubicado en la misma carpeta, aparece la instrucción

\hyperref{doc1.pdf}{teor}{cayley}{Teorema de Cayley}

Si se procesa *doc2.tex* con pdf \LaTeX , en el archivo generado, *doc2.pdf*, se encuentra el hipertexto ‘Teorema de Cayley’. Un *click* sobre éste abre el archivo *doc1.pdf*, en la página en la que aparece enunciado el teorema.

10.2.3. Inclusión de gráficas

Como se mencionó en la sección 10.2.1, para procesar un documento por medio de pdf \LaTeX , el paquete *graphicx*⁴ se debe cargar con la opción *pdftex*.

⁴Para que el paquete *graphicx* funcione adecuadamente con pdf \LaTeX , se requiere que el archivo *pdftex.def* esté presente en la carpeta */texmf/tex/latex/graphics/* o equi-

Tal opción permite el uso de los comandos propios de `graphicx`; en particular, se pueden incluir gráficas externas con el comando `\includegraphics`. Hay que advertir que `pdfLATEX` es compatible únicamente con los formatos gráficos PDF, JPEG, PNG y TIFF. Por su capacidad de compresión, es decir, por el menor tamaño de los archivos, se recomienda el formato PNG (*Portable Network Graphics*). La mayoría de los programas gráficos pueden convertir imágenes a ese formato.

Un notable recurso de `hyperref`, para tener presente, permite que las gráficas insertadas con `\includegraphics` se transformen en hiper-gráficas, es decir, en enlaces activos. Esto se logra colocando la instrucción `\includegraphics` en el argumento *expresión* de alguno de los comandos `\href`, `\hyperref` o `\hyperlink`, como se ilustra en el siguiente ejemplo.

Ejemplo El comando

```
\hyperlink{imag}{\includegraphics[scale=0.5]{%
C:/Archivos/Imagenes/graf5.png}}
```

inserta la gráfica `graf5.png`, reducida al 50 % de su tamaño original, y la activa como enlace. El destino de tal enlace debe ser establecido previamente, con la clave `imag`, usando el comando `\hypertarget`.

10.2.4. Acciones de menú en **Acrobat Reader** o **Adobe Acrobat**

El paquete `hyperref` también permite acceso a las opciones de menú de los programas **Acrobat Reader** o **Adobe Acrobat**⁵. En concreto, la instrucción

```
\Acrobatmenu{acción de menú}{expresión}
```

crea el enlace *expresión*; un *click* sobre éste ejecuta la *acción de menú* especificada. Tales acciones se deben escribir con la sintaxis de la Tabla 10.5; nótese que algunas de ellas solamente están disponibles en **Adobe Acrobat**, no en el visor **Acrobat Reader**.

Ejemplo La instrucción `\Acrobatmenu{NextPage}{Página siguiente}` crea el hipertexto ‘Página siguiente’; un *click* sobre éste conduce a la página siguiente del documento PDF. Usando comandos del paquete `color` se puede hacer que el enlace tenga apariencia de botón; por ejemplo, al escribir

```
\Acrobatmenu{NextPage}{\colorbox{yellow}{\color{red}%
Página siguiente}}
```

el enlace ‘Página siguiente’ aparece en letra roja sobre una caja de fondo amarillo.

valente. El usuario debe colocarlo manualmente en caso de no estar presente.

⁵ El programa comercial **Adobe Acrobat** (editor de archivos PDF) era distribuido en versiones anteriores con el nombre de **Acrobat Exchange**.

File	Open, Close, Scan, Save, SaveAs, Optimizer:SaveAsOpt, Print, PageSetup, Quit
File→Import	ImportImage, ImportNotes, AcroForm:ImportFDF
File→Export	ExportNotes, AcroForm:ExportFDF
File→DocumentInfo	GeneralInfo, OpenInfo, FontsInfo, SecurityInfo, Weblink:Base, AutoIndex:DocInfo
File→Preferences	GeneralPrefs, NotePrefs, FullScreenPrefs, Weblink:Prefs, AcroSearch:Preferences (Windows), AcroSearch:Prefs (Mac), Cpt:Capture Edit Undo, Cut, Copy, Paste, Clear, SelectAll, Ole:CopyFile, TouchUp:TextAttributes, TouchUp:FitTextToSelection, TouchUp>ShowLineMarkers, TouchUp>ShowCaptureSuspects, TouchUp:FindSuspect, Properties
Edit→Fields	AcroForm:Duplicate, AcroForm:TabOrder
Document	Cpt:CapturePages, AcroForm:Actions, CropPages, RotatePages, InsertPages, ExtractPages, ReplacePages, DeletePages, NewBookmark, SetBookmarkDest, CreateAllThumbs, DeleteAllThumbs
View	ActualSize, FitVisible, FitWidth, FitPage, ZoomTo, FullScreen, FirstPage, PrevPage, NextPage, LastPage, GoToPage, GoBack, GoForward, SinglePage, OneColumn, TwoColumns, ArticleThreads, PageOnly, ShowBookmarks, ShowThumbs
Tools	Hand, ZoomIn, ZoomOut, SelectText, SelectGraphics, Note, Link, Thread, AcroForm:Tool, TouchUp:TextTool, AcroMovie:MoviePlayer, Find, FindAgain, FindNextNote, CreateNotesFile
Tools→Search	AcroSrch:Query, AcroSrch:Indexes, AcroSrch:Results, AcroSrch:Assist, AcroSrch:PrevDoc, AcroSrch:PrevHit, AcroSrch:NextHit, AcroSrch:NextDoc
Window	CloseAll, ShowHideToolBar, ShowHideMenuBar, ShowHideClipboard, Cascade, TileHorizontal, TileVertical,
Help	HelpUserGuide, HelpTutorial, HelpExchange, HelpScan, HelpCapture, HelpPDFWriter, HelpDistiller, HelpSearch, HelpCatalog, HelpReader, Weblink:Home Help (Windows) About

TABLA 10.5. Acciones disponibles para el argumento *acción de menú* del comando \Acrobatmenu (sección 10.2.4).

10.2.5. Control sobre la lista de *Marcadores* (*Bookmarks*)

Al usar `hyperref`, el panel de *Marcadores* (*Bookmarks*) o tabla de contenido del documento PDF se genera automáticamente con la información de los comandos `\chapter{título}`, `\section{título}`, etc, a menos que se utilice la opción `bookmarks=false` (Tabla 10.3). Pero los comandos LATEX que aparezcan en el *título* mismo de cada división no pueden ser interpretados como código PDF. Para obviar estas situaciones, el paquete `hyperref` posee el siguiente comando de sustitución:

`\texorpdfstring{comando o texto LATEX}{texto PDF}`

El primer argumento puede contener comandos LATEX y está destinado para el cuerpo del documento; el segundo argumento contiene el texto para los *Marcadores* (*Bookmarks*).

Ejemplo La instrucción

`\section{\texorpdfstring{H$_2$O y sus propiedades}{Agua} y sus propiedades}`

crea la sección titulada ‘H₂O y sus propiedades’, la cual aparece en la lista de *Marcadores* (*Bookmarks*) como ‘Agua y sus propiedades’.

El paquete `hyperref` tiene además el comando `\pdfbookmark` para añadir un ítem a la lista de *Marcadores* (*Bookmarks*). Su sintaxis es:

`\pdfbookmark[nivel]{Texto del ítem}{clave}`

Esta instrucción añade el *Texto del ítem* a la lista de *Marcadores* (*Bookmarks*), en el sitio en el que aparece la instrucción. El *nivel* se refiere a la jerarquía de divisiones utilizada por LATEX (sección 6.2); es decir, `part` tiene nivel -1, `chapter` tiene nivel 0, `section` nivel 1, etc. La *clave*, única para cada ítem añadido, es escogida por el usuario.

Ejemplo Si la bibliografía del documento se escribe recurriendo al entorno `thebibliography`, su título (‘Bibliografía’ o ‘Referencias’) no aparece en la lista de *Marcadores* (*Bookmarks*). No obstante, si escribimos la instrucción

`\pdfbookmark[0]{Bibliografía}{bib}`

inmediatamente antes de `\begin{thebibliography}`, el ítem titulado ‘Bibliografía’ aparecerá como *marcador* en el nivel 0 (nivel de capítulos).

Los comandos `\currentpdfbookmark` y `\subpdfbookmark` tienen la misma sintaxis que `\pdfbookmark`, excepto que no poseen el argumento *nivel*. Con estos comandos el usuario puede crear *marcadores* en el mismo nivel o en el nivel inmediatamente inferior al actual, respectivamente, sin necesidad de precisar el valor mismo del nivel.

Los *marcadores* generados por el paquete `hyperref` conducen a sitios dentro del mismo documento. Para direccionar *marcadores* a documentos externos se debe usar la primitiva `\pdfoutline` de pdfTEX (sección 10.3.2).

10.3. Los programas **pdfTEX** y **pdflATEX**

Según lo sugerido al final de la sección 10.1, se recomienda generar un documento PDF procesando el documento fuente con **pdflATEX**. Sin embargo, hasta el momento no se ha mencionado ningún comando primario, propio de los programas **pdfTEX** o **pdflATEX**. En esta sección describiremos estos programas.

El propósito del proyecto **pdfTEX** (**pdflATEX**), iniciado y desarrollado principalmente por por H n Th  Th n en el a o 2000, fue crear una extensi n de **TeX** (**LATEX**) capaz de generar directamente PDF a partir de un documento fuente **TeX** (**LATEX**). El programa ha sido ensayado con todo éxito en varias plataformas computacionales y su uso se ha popularizado r pidamente. El manual oficial, incluido en la documentaci n que acompa a el programa, es el archivo **pdftex-a.pdf**; una gu a con las preguntas m s frecuentes se encuentra en el archivo **pdfTeX-FAQ.pdf**.

Algunas distribuciones de **TeX**, como **MiKTeX**, **FPTeX** y **CMACTeX**, incluyen a **pdfTEX** y **pdflATEX** como componentes pre-instalados. En caso de que **pdflATEX** no est  instalado en la implementaci n local, el usuario debe consultar los citados manuales para proceder a la instalaci n y a la generaci n del formato **pdflatex(fmt)**, necesario para procesar documentos **LATEX**. Supondremos que el lector tiene a su disposici n una versi n correctamente instalada de **pdflATEX**.

Si se usa la opci n **pdftex** para cargar alg n paquete (como **hyperref** o **graphicx**), el documento solamente se puede procesar con **pdflATEX**; al intentar procesarlo como un archivo **LATEX** normal, se recibir  el mensaje de error

Why not use pdf(e)TeX binaries?

Este mismo mensaje se recibe si el documento posee instrucciones primarias de **pdfTEX** y se intenta procesar como archivo **LATEX**. Por convenci n, las instrucciones propias de **pdfTEX** comienzan con **\pdf...**

10.3.1. Configuraci n de **pdfTEX**

Al procesar un documento, **pdflATEX** consulta el archivo de configuraci n **pdftex.cfg**, ubicado en la carpeta **/texmf/pdftex/config/** (o equivalente). Dicho archivo contiene par metros que controlan el tipo de archivo producido (DVI o PDF), el formato de p gina, el nivel de compresi n, etc. Estos par metros pueden ser modificados por medio de comandos internos, desde el archivo fuente que se est  procesando, en cuyo caso los valores contenidos en **pdftex.cfg** se sustituyen por los nuevos. Los par metros de

configuración más importantes y los correspondientes comandos internos son los siguientes:

compress_level Parámetro entero (entre 0 y 9) que determina el nivel de compresión del archivo PDF generado. Con valor 0 no hay compresión; el valor 9 representa la compresión máxima (y también el procesamiento más lento). Comando correspondiente: `\pdfcompresslevel`.

page_width Controla la anchura de la página. Comando correspondiente: `\pdfpagewidth`.

page_height Controla la altura de la página. Comando correspondiente: `\pdfpageheight`.

horigin Controla el margen izquierdo de la página. Comando correspondiente: `\pdfhorigin`.

vorigin Controla el margen superior de la página. Comando correspondiente: `\pdfvorigin`.

output_format Parámetro entero que determina el tipo de archivo producido: con valor positivo se produce un archivo PDF, con valor 0 o negativo se produce un archivo DVI. Comando correspondiente: `\pdfoutput`⁶.

decimal_digits Parámetro entero (entre 0 y 5) que controla el número de dígitos en la aritmética interna de pdfTEX. Con un valor alto se genera un archivo más preciso, pero de mayor tamaño y que requiere mayor tiempo para despliegue en pantalla e impresión. El valor óptimo recomendado es 2. Comando correspondiente: `\pdfdecimaldigits`.

pk_resolution Parámetro entero que controla la resolución de las fuentes `*.pk` (sección 9.2). Las impresoras de calidad media o superior admiten resoluciones de 600 dpi (*dots per inch*) en adelante. Comando correspondiente: `\pdfpkresolution`.

map Indica los archivos `*.map` de mapeo de fuentes (sección 9.2), que son similares a los utilizados por el manejador `dvips` (sección 7.1). Se pueden incluir varios archivos `*.map` por medio de sendos renglones `map`; en tal caso, se escribe un signo `+` antes del nombre de cada archivo añadido. Si no se especifica ningún archivo, pdfTEX utiliza por defecto

⁶El lector debe tener presente que si el archivo de configuración `pdftex.cfg` incluye un valor positivo para el parámetro `output_format` (es decir, salida PDF), el comando `\pdfoutput` ya no puede alterar el tipo de archivo producido.

`psfonts.map`, ubicado en el subdirectorio `/texmf/pdftex/config/` (o equivalente). El archivo `pdftex.cfg` puede, por ejemplo, contener los siguientes renglones para el parámetro `map`:

```
map psfonts.map  
map +archivo1.map  
map +archivo2.map
```

Ejemplo Un archivo `pdftex.cfg` típico puede tener el siguiente aspecto:

```
pk_resolution 600  
output_format 1  
compress_level 9  
decimal_digits 2  
page_width 8.5in  
page_height 11in  
horigin 1in  
vorigin 1in  
map psfonts.map  
map +misc.map
```

El usuario puede siempre consultar el archivo de configuración local y modificarlo de acuerdo con sus necesidades.

Ejemplo Independientemente de los valores establecidos en el archivo de configuración `pdftex.cfg`, los parámetros se pueden modificar desde el documento fuente. Así por ejemplo, la escribir las instrucciones

```
\pdfvorigin=1cm  
\pdfhorigin=2cm  
\pdfcompresslevel=0
```

inmediatamente después de `\begin{document}`, el margen superior de las páginas será de 1 cm y el lateral de 2 cm. Además, el archivo final no se comprimirá (agilizando así el procesamiento del documento).

10.3.2. Algunos comandos primarios de pdftEX

Destacamos a continuación algunos comandos propios de `pdftEX` que no tienen contraparte en el paquete `hyperref`. El paquete `hyperref` no es incompatible con `pdftEX`; de hecho, los dos programas se complementan bastante bien. No obstante, la mezcla de comandos puede dar lugar a resultados inesperados o insatisfactorios; esto es cierto, en especial, en el manejo de los *Marcadores (Bookmarks)*. El usuario debe proceder por ensayo y error.

El comando \pdfoutline El paquete `hyperref` no tiene mecanismos para direccionar los *Marcadores (Bookmarks)* hacia archivos externos. La primitiva `\pdfoutline` de pdfTEX sí permite ese tipo de enlaces. Concretamente, la instrucción

<code>\pdfoutline goto file {archivo PDF} page n {modo} {Texto}</code>
--

añade el *Texto* como un ítem en la lista de *Marcadores (Bookmarks)*. Un *click* sobre éste conduce al *archivo PDF* especificado, abierto en la página *n*, la cual se despliega según el *modo* indicado. En la Tabla 10.6 aparecen algunos de los parámetros que se pueden usar para el argumento *modo*. Si el *archivo PDF* de destino no está ubicado en la misma carpeta que el documento fuente, se debe indicar su ruta completa, usando para ello la sintaxis propia del sistema operativo local.

<code>/Fit</code>	Hace visible toda la página en la ventana.
<code>/FitH</code>	Hace visible la página en su amplitud horizontal.
<code>/FitV</code>	Hace visible la página en su amplitud vertical.
<code>/FitB</code>	Hace visible toda la página, omitiendo los márgenes.

TABLA 10.6. Algunos parámetros permitidos para el argumento *modo* de `\pdfoutline`.

Otra versión de `\pdfoutline` conduce a un destino identificado por medio de una clave, en vez del número de página. Más precisamente, la instrucción

<code>\pdfoutline goto file {archivo PDF} name {clave} {Texto}</code>

añade el *Texto* como un ítem en la lista de *Marcadores (Bookmarks)*. Un *click* sobre éste conduce al *archivo PDF* especificado, abierto en la página en la que aparece la *clave*, previamente definida en el archivo de destino por medio de `\pdfdest`. Este último comando tiene la siguiente sintaxis:

<code>\pdfdest name {clave} modo</code>

donde el *modo* de despliegue puede ser uno de los siguientes parámetros:

<code>fit fith fitv fitb</code>

Estos parámetros tienen un significado análogo a los de la Tabla 10.6.

Ejemplo La instrucción

<code>\pdfoutline goto file {doc2.pdf} page 5{/Fit} {Las % ecuaciones de onda}</code>

da lugar al ítem ‘Las ecuaciones de onda’ en la lista de *Marcadores (Bookmarks)*; un click sobre éste conduce a la página 5 del documento `doc2.pdf`. Se ha escogido como modo de despliegue el parámetro `/Fit`.

Ejemplo La instrucción

```
\pdfoutline goto file {doc2.pdf} name {ecsonda} {Las %
    ecuaciones de onda}
```

añade el ítem ‘Las ecuaciones de onda’ a la lista de *Marcadores (Bookmarks)*; un click sobre éste conduce al documento doc2.pdf, abierto en la página en la que ha sido definida la clave **ecsonda**, por medio del comando

```
\pdfdest name {ecsonda} fit
```

Se ha escogido como modo de despliegue el parámetro **fit**.

El comando \pdfannot Entre los elementos frecuentemente encontrados en documentos PDF están las anotaciones, ventanas desplegables que contienen información acerca de un apartado específico del documento. Estas ventanas se abren al hacer *click* sobre una pequeña papeleta o ícono de notas (Figura 10.1).

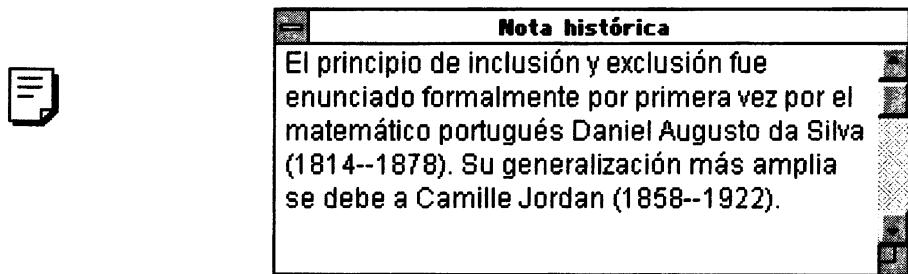


FIGURA 10.1. Papeleta o ícono de notas y su ventana abierta.

El programa pdfTEX posee el comando **\pdfannot** para incluir este tipo de notas en un documento. Se usa en la siguiente forma:

```
\pdfannot width l height h depth d {/Subtype /Text /Contents %
    (texto de la nota)}
```

donde *l* (ancho), *h* (altura) y *d* (profundidad) son las dimensiones de la ventana desplegada (no del ícono de notas). Entre los corchetes {...} que hacen parte de este comando se pueden incluir, opcionalmente, otros parámetros para modificar el aspecto de la nota; algunos de ellos son:

/C [r g b] Color de la papeleta o ícono de notas y de la barra superior de su ventana abierta, en el modelo **rgb** (sección 7.2). Por defecto, este color es amarillo.

/T (*título*) Añade un *título* a la nota. Por defecto, la ventana de notas no lleva título.

/Open true Hace que la ventana de la nota aparezca abierta. Por defecto, aparece visible únicamente la papeleta o ícono de notas.

Una limitación seria de `\pdfannot` es que no hay control sobre el tipo de letra en el *título* y en el *texto de la nota*, ni se admiten comandos TeX en ellos. En particular, si se usa el paquete `inputenc`, los símbolos acentuados á, é, í, etc no se pueden incluir en las notas, puesto que son traducidos en comandos TeX. Se recomienda, por consiguiente, usar las fuentes AE (sección 9.3), con las cuales el paquete `inputenc` no es necesario. Como se recordará, las fuentes AE son fuentes PostScript virtuales, con la codificación T1, que emulan las fuentes CM.

Ejemplo Habiendo cargado las fuentes AE, la nota de la Figura 10.1 se obtuvo a partir de lo siguiente:

```
\pdfannot width 10cm height 4 cm depth 5pt{/T (Nota
histórica) /Subtype /Text /Contents (El principio de
inclusión y exclusión fue enunciado por el . . . . . )}
```

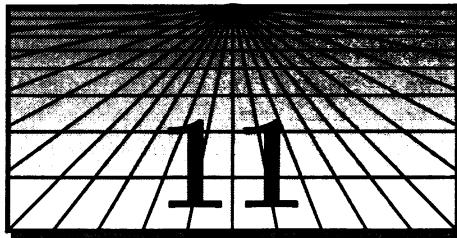
10.4. Ejemplo de un documento LATEX interactivo en formato PDF

 La versión interactiva de *El Universo LATEX* incluida en el CD adjunto fue producida con las herramientas presentadas en este capítulo. Como un ejemplo más modesto y fácil de examinar por parte del lector hemos incluido, en la carpeta `/EjemploPDF/` del CD, el documento interactivo `Ejemplo.pdf`, junto con su archivo fuente `Ejemplo.tex` y otros archivos subsidiarios. Se invita al lector a examinar con detenimiento este documento, en el cual se han utilizado muchas de las opciones y comandos de `hyperref`.

A continuación destacamos algunas facetas del documento `Ejemplo.tex`, remitiendo al lector a las secciones pertinentes del presente capítulo:

1. Se cargan los paquetes `babel`, `amsmath`, `amsthm`, `amssymb`, `graphicx`, `color`, `enumerate`, `hyperref` y las fuentes AE. Obsérvese que `inputenc` no es necesario. El archivo `Ejemplo.pdf` se obtuvo procesando el documento `Ejemplo.tex` con `pdflatex`.

2. Los títulos de las secciones aparecen en color azul, por medio de `\textcolor{blue}`. Puesto que esta instrucción de color no es admisible en los *Marcadores (Bookmarks)*, se hace necesario el uso de `\texorpdfstring` en el argumento de `\section` (véase la sección 10.2.5).
3. Para crear los botones ‘Página anterior’, ‘Página siguiente’ y ‘Pantalla completa’ se ha usado el comando `\Acrobatmenu` (sección 10.2.4).
4. Las gráficas de la página 5 se han convertido en hipergráficas escribiendo `\includegraphics` en el segundo argumento del comando `\href{file:...}{...}` (véanse las secciones 10.2.2 y 10.2.3).
5. Los archivos externos a los que conducen las hipergráficas de la página 5 contienen varios efectos como gráficas rotadas, fondos a color y enlaces a programas locales. Así, hay enlaces al programa gráfico `LViewPro` por medio del botón ‘Editar’ (tal programa no necesariamente está instalado en el computador local). Aparece también un enlace a una página *Web*. Los enlaces a programas externos se logran con el comando `\href{run:...}` y los enlaces a páginas *Web* con `\url{...}` (véase la sección 10.2.2).
6. La nota amarilla de la página 4 fue generada con `\pdfannot`, que es el único comando de `pdfTEX` usado en el documento (sección 10.3.2).
7. El ítem ‘Referencias’ en la lista de *Marcadores (Bookmarks)* se ha añadido por medio del comando `\pdfbookmark`. Este ítem no es generado automáticamente ya que no es producto de un comando `\section` (véase la sección 10.2.5).
8. En la bibliografía hay enlaces a las páginas en las que aparecen las citas bibliográficas, debido a que el paquete `hyperref` se cargó con la opción `pagebackref` (véase la Tabla 10.4).



Gráficas con el paquete **PSTricks**

11.1. El programa **PSTricks** y su instalación

 Los archivos que conforman el paquete **PSTricks** están incluidos en el CD adjunto, en la carpeta `/Paquetes/pstricks`.

PSTricks¹ es una colección de sofisticados macros **T_EX**, basados en el lenguaje de programación PostScript (sección 7.1). Puesto que PostScript fue especialmente diseñado para la creación y manipulación de objetos gráficos, su integración al ambiente **T_EX** produce una poderosa combinación. Como lo dice su creador, Timothy Van Zandt, **PSTricks** le pone la crema (PostScript) a las fresas (**T_EX**), para un irresistible festín. Existen otros intentos de integración de PostScript y **T_EX**, pero sin lugar a dudas, el más exitoso y completo es **PSTricks**.

El presente capítulo pretende ser una guía completa de **PSTricks**, con abundantes ejemplos de “la vida real”. Sólo unos pocos comandos se han omitido en nuestra presentación, ya sea por su limitada utilidad, o porque sus efectos se pueden conseguir más eficientemente con otros comandos, o porque su uso puede conducir a serios conflictos con PostScript.

La instalación del paquete depende tanto de la plataforma computacional disponible (*Windows*, Macintosh, Unix, Linux, etc) como de la implementación **T_EX** instalada en el computador local. Se debe disponer, además, de los manejadores de pantalla e impresora adecuados. El manejador más recomendable es **dvips** (sección 7.1), por su reconocida confiabilidad y porque el archivo de configuración de **PSTricks**, **pstricks.con**, viene previamente configurado para este manejador. No obstante, el paquete funciona también con otros manejadores; la documentación del programa (incluida también en el CD adjunto) contiene información detallada al respecto.

Algunos visualizadores son incapaces de mostrar correctamente las gráficas de **PSTricks**; en tales casos, lo más recomendable es convertir el archivo

¹Creado por Timothy Van Zandt, de la Universidad de Princeton, en 1993–94.

dvi al formato ps para luego ver o imprimir el archivo PostScript resultante en GSview o Ghostview (véase al respecto la sección 7.1)².

A continuación presentamos una guía de instalación manual del paquete **PSTricks**, para las implementaciones T_EX que respeten la estructura TDS (Apéndice C); esta guía puede servir para las demás implementaciones. Los archivos que conforman a **PSTricks** se deben colocar en diferentes carpetas, subcarpetas todas del directorio principal /texmf/. Se debe crear primero la subcarpeta /pstricks/ en los subdirectorios requeridos, tal como se muestra en la Tabla 11.1.

Subcarpeta /tex/generic/pstricks/	Subcarpeta /tex/latex/pstricks/	Subcarpeta /dvips/pstricks/
dvipstone.con pstricks.con textures.con vtxe.con multido.tex pst-3d.tex pst-blur.tex pst-char.tex pst-coil.tex pst-eps.tex pst-fill.tex pst-ghsb.tex pst-gr3d.tex pst-grad.tex pst-key.tex pst-lens.tex pst-node.tex pst-osci.tex pst-plot.tex pst-poly.tex pstricks.tex	multido.sty pst-3d.sty pst-all.sty pst-blur.sty pst-char.sty pst-coil.sty pst-eps.sty pst-fill.sty pst-ghsb.sty pst-gr3d.sty pst-grad.sty pst-key.sty pst-lens.sty pst-node.sty pst-osci.sty pst-plot.sty pst-poly.sty pstricks.sty pst-slpe.sty pst-text.sty pst-tree.sty pst-slpe.tex pst-text.tex pst-tree.tex	pst-blur.pro pst-coil.pro pst-dots.pro pst-ghsb.pro pst-grad.pro pst-node.pro pstricks.pro pst-slpe.pro pst-text.pro

TABLA 11.1. Colocación de los archivos de **PSTricks**. La carpetas indicadas son todas subcarpetas del directorio principal /texmf/.

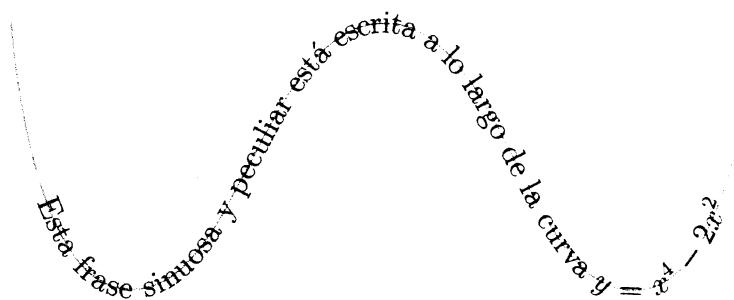
²En algunas implementaciones, como WinEdt, la conversión de DVI a PS se hace con un simple *click* sobre el botón **dvi~>ps**.

Siguiendo el esquema TDS, se recomienda colocar la documentación del programa, que incluye los manuales de referencia originales, en la carpeta `/texmf/doc/latex/pstricks/`.

Para verificar que el paquete ha sido instalado correctamente, sugerimos al lector procesar el siguiente documento fuente:

```
\documentclass{article}
\usepackage{pst-all}
\usepackage{pstcol}
\begin{document}
\begin{center}
\psset{unit=3.5cm,linecolor=lightgray}
\begin{pspicture}(-1.5,-1)(1.5,0)
\pstextpath[c]{\pscurve(-1.41,0)(-1,-1)(0,0)(1,-1)(1.41,0)}
{\color{blue}\large Esta frase sinuosa y peculiar est\'a
 escrita a lo largo de la curva $y=x^4-2x^2$}
\end{pspicture}
\end{center}
\end{document}
```

tras lo cual se debe obtener el siguiente despliegue, con el texto en color azul y la curva del fondo en color gris:



Al usar **PSTricks**, el usuario tiene que lidiar con dos monstruos invisibles: **TeX** y PostScript; los errores sintácticos cometidos pueden generar mensajes de dos tipos diferentes, pero esto no debe generar pánico.

Los comandos de **PSTricks** pueden tener muchos argumentos y parámetros y al escribirlos pueden ocupar varias líneas. Es por eso importante saber que, en los argumentos que aparecen entre corchetes `{ }`, se pueden dejar espacios en blanco después de las comas, y dos o más renglones se pueden separar libremente colocando el símbolo `%` al final de cada línea³.

³Véase al respecto la nota al final de la página 9.

Aparte de la colección básica de macros que se encuentra en el archivo **pstricks.tex**, **PSTricks** posee módulos individuales para tareas específicas (Tabla 11.2)⁴; se puede acceder a cada uno con `\usepackage{...}`, o a todos ellos simultáneamente por medio de `\usepackage{pst-all}`.

Paquete	Función	Secciones
pst-plot	Rectas, polígonos, ejes, curvas arbitrarias	11.4, 11.5, 11.16, 11.20 a 11.23
pst-grad	Sombreado gradual	11.9
pst-node	Grafos (nodos y conectores de nodos)	11.26
pst-tree	Árboles	11.27
pst-coil	Resortes y zig-zags	11.28
pst-text	Texto a lo largo de curvas	11.29
pst-char	Texto con efectos especiales	11.30

TABLA 11.2. Algunos módulos de **PSTricks**.

Si el usuario necesita cargar alguno de los paquetes **graphics** o **graphicx**, presentados en la sección 7.3, debe hacerlo después de cargar el paquete **pst-all**.

11.2. Las características básicas de una gráfica en **PSTricks**

La manera más práctica de utilizar **PSTricks** con **LATEX** para hacer una gráfica determinada es por medio del entorno **pspicture**:

```
\begin{pspicture}(x0,y0)(x1,y1)
  ... comandos de la gráfica ...
\end{pspicture}
```

LATEX reserva para la gráfica el rectángulo cuyos vértices opuestos son (x_0, y_0) y (x_1, y_1) . La gráfica misma puede exceder los bordes del rectángulo, sin originar mensaje de error alguno. Existe, además, el entorno homólogo **pspicture*** con el cual se elimina o “recorta” todo lo que se sale del rectángulo reservado. El punto (x_0, y_0) se puede omitir, en cuyo caso, **PSTricks** reserva el rectángulo con vértices opuestos $(0, 0)$ y (x_1, y_1) .

La unidad de medida básica se controla con el parámetro **unit**. Por defecto, su valor es de 1 cm pero se puede cambiar con la instrucción `\psset{ }`⁵. Por ejemplo, la instrucción `\psset{unit=0.8cm}`, colocada

⁴Hay otros módulos de tipo experimental y es de esperarse que aparezcan en el futuro otros más; el usuario debe por ello visitar con frecuencia los portales de CTAN.

⁵En general, `\psset{...}` se usa para establecer parámetros permanentes.

antes del comando `\begin{pspicture}`, establece que la unidad de medida en la gráfica es de 0.8 cm. De hecho, **PSTricks** utiliza dos parámetros para las unidades de medida: `xunit`, para las coordenadas horizontales, y `yunit` para las verticales. Por defecto, `unit = xunit = yunit = 1 cm`. Los valores de `xunit` y `yunit` se pueden cambiar independientemente. Por ejemplo, la instrucción

```
\psset{xunit=0.7cm,yunit=1.5cm}
```

colocada antes de `\begin{pspicture}`, cambia las dos unidades de manera independiente.

Como guía en el trazado de gráficas, es útil tener a disposición una grilla o cuadrícula del tamaño del rectángulo con vértices opuestos (x_0, y_0) y (x_1, y_1) . Tal grilla no está pre-definida en **PSTricks** pero podemos definirla de la siguiente manera⁶:

```
\newpsobject{grilla}{psgrid}{subgriddiv=1,griddots=10,%
gridlabels=6pt}
```

La anterior instrucción se puede colocar en el preámbulo del documento, inmediatamente antes de `\begin{document}`. El comando `\grilla` traza una cuadrícula punteada en intervalos de 1 mm y sub-dividida en unidades de 1 cm; lo usaremos en la mayoría de los ejemplos del presente capítulo como ayuda visual para el lector. Una vez elaborada la gráfica, el usuario puede borrar la instrucción `\grilla` para eliminar completamente la grilla.

Ejemplo Grilla obtenida sobre el rectángulo con vértices opuestos $(-5, 0)$ y $(5, 4)$ al usar la instrucción `\grilla` definida arriba.

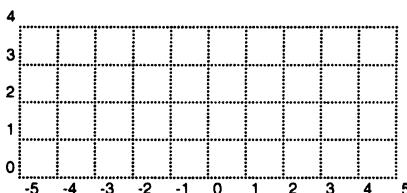


```
\begin{center}
\begin{pspicture}(-5,0)(5,4)\grilla
\end{pspicture}
\end{center}
```

⁶Los comandos generales de **PSTricks** para grillas se presentan en la sección 11.17. Sobre el uso del comando `\newpsobject`, véase la sección 11.19.

La instrucción `\psset{unit=...}` para cambio de unidad de medida se debe colocar *antes* de `\begin{pspicture}` para que la grilla obtenida con `\grilla` se reduzca proporcionalmente y sea de utilidad visual. De lo contrario, la grilla no cambiará su tamaño usual.

Ejemplo La instrucción `\begin{pspicture}(-5,0)(5,4)\grilla` del ejemplo anterior está precedida del comando `\psset{unit=0.5cm}` para cambio de unidad de medida. Obsérseve cómo la grilla se reduce también en tamaño y queda sub-dividida en unidades de 0.5 cm.



```
\begin{center}
\psset{unit=0.5cm}
\begin{pspicture}(-5,0)(5,4)\grilla
\end{pspicture}
\end{center}
```

11.3. El uso de colores en PSTRicks

Con **PSTRicks** se debe cargar el paquete `pst-col` en lugar del paquete estándar `color` de L^AT_EX, descrito en la sección 7.2. El paquete `pst-col` modifica algunas funciones internas de **PSTRicks** para que todos los comandos del paquete `color` (incluyendo la definición de nuevos colores) se puedan usar libremente. Así, para cargar todos los módulos de **PSTRicks**, con capacidad de color completa, se escribe

```
\usepackage{pst-all}
\usepackage{pstcol}
```

en el preámbulo del documento.

PSTRicks tiene pre-definidos los siguientes colores básicos:

<code>gray</code>	(gris)	<code>red</code>	(rojo)
<code>lightgray</code>	(gris claro)	<code>green</code>	(verde)
<code>darkgray</code>	(gris oscuro)	<code>blue</code>	(azul oscuro)
<code>black</code>	(negro)	<code>cyan</code>	(azul claro)
<code>white</code>	(blanco)	<code>magenta</code>	(rosado)
<code>yellow</code>	(amarillo)		

Estos colores se pueden usar libremente en todas las construcciones gráficas. Además, se pueden hacer cambios de colores, tanto en texto como en gráficas, con las declaraciones globales

`\gray ...}, {\red ...}, {\blue ...}, ..., etc.`

Ejemplo

Esta frase está escrita en color gris claro.

Esta frase está escrita en color gris normal.

```
{\lightgray \large Esta frase está escrita en color gris claro.}
{\gray \large Esta frase está escrita en color gris normal.}
```

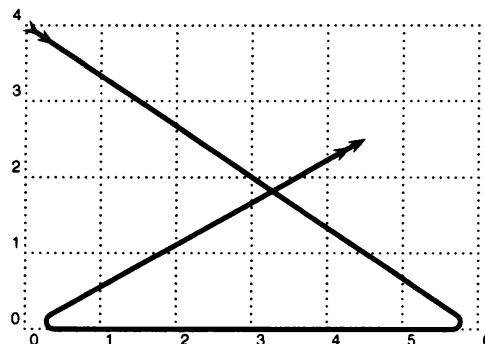
11.4. Líneas rectas

PSTricks tiene el macro `\psline` para unir con segmentos de recta los puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$; su sintaxis precisa es:

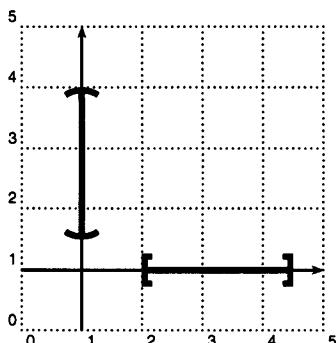
```
\psline[opciones]{flechas o extremos}(x0,y0)(x1,y1)...(xn,yn)
```

Si aparece sólo un punto en la lista, éste se une con el origen de coordenadas $(0,0)$. Las *opciones* se controlan con los parámetros `linewidth`, `linecolor`, `linearc` y `linestyle`, cuyo significado y uso aparecen en la Tabla 11.3. Las diferentes opciones utilizadas se separan con comas. Las líneas pueden terminar en flechas o en otros extremos, siendo las más importantes las que aparecen en la Tabla 11.4.

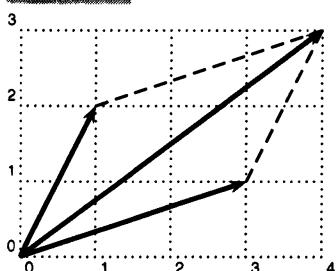
Ejemplo



```
\begin{center}
\begin{pspicture}(0,0)(6,4)\grilla
\psline[linewidth=1.5pt, linearc=0.1]{>>->}(0,4)(6,0)(0,0)(4.5,2.5)
\end{pspicture}
\end{center}
```

Ejemplo

```
\psset{unit=0.8cm}
\begin{pspicture}(0,0)(5,5) \grilla
\psline{->}(0,1)(5,1)
\psline{->}(1,0)(1,5)
\psline[linewidth=2pt]{[-]}(2,1)(4.5,1)
\psline[linewidth=2pt]{(-)}(1,1.5)(1,4)
\end{pspicture}
```

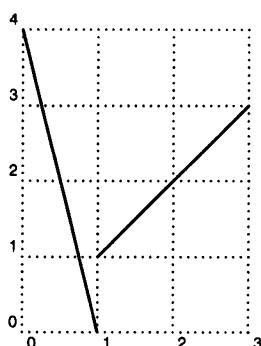
Ejemplo

```
\begin{pspicture}(0,0)(4,3)\grilla
\psline[linewidth=1.5pt]{->}(3,1)
\psline[linewidth=1.5pt]{->}(1,2)
\psline[linewidth=1.5pt]{->}(4,3)
\psline[linestyle=dashed](1,2)(4,3)
\psline[linestyle=dashed](3,1)(4,3)
\end{pspicture}
```

La versión simplificada de `\psline` es

`\qline(x1,y1)(x2,y2)`

con el cual se obtiene el segmento de recta que une los puntos (x_1, y_1) y (x_2, y_2) . Ambos puntos son obligatorios en el comando `\qline` y no se admiten opciones ni flechas en los extremos.

Ejemplo

```
\begin{pspicture}(0,0)(3,4)\grilla
\qline(0,4)(1,0)
\qline(1,1)(3,3)
\end{pspicture}
```

<code>linestyle=solid/none/dashed/dotted</code>	El valor por defecto de este parámetro para todas las líneas rectas o curvas es <code>solid</code> (línea continua o “sólida”). Con el valor <code>dashed</code> se obtiene una línea a trozos. Con el valor <code>dotted</code> , la línea es punteada.
<code>dash=n m</code>	Patrón negro-blanco en las líneas a trozos; n y m deben ser dimensiones \TeX . Valor por defecto: <code>5pt 3pt</code> .
<code>dotsep=n</code>	Separación de los puntos en las líneas punteadas; n debe ser una dimensión \TeX . Valor por defecto: <code>3pt</code> .
<code>linewidth=n</code>	Grosor de la línea; n debe ser una dimensión \TeX . Valor por defecto: <code>0.8pt</code> .
<code>linecolor=color</code>	Color de la línea; valor por defecto: <code>black</code> . Para otros colores permitidos véase la sección 11.3.
<code>linearc=n</code>	Radio del arco en los vértices de las líneas trazadas con <code>\psline</code> o <code>\pspolygon</code> ; n debe ser una dimensión no-negativa. Valor por defecto: <code>0pt</code> .
<code>doubleline=true/false</code>	Determina si las líneas trazadas son dobles o sencillas. Valor por defecto: <code>false</code> .

TABLA 11.3. Opciones para el trazado de líneas rectas o curvas.

Parámetro	Extremo	Parámetro	Extremo	Parámetro	Extremo
<code>-></code>		<code>>>-</code>		<code>-)</code>	
<code><-</code>		<code>>>-<<</code>		<code>(-</code>	
<code><-></code>		<code>-> </code>		<code>(-)</code>	
<code>->></code>		<code> <-</code>		<code>-o</code>	
<code>>>-<<</code>		<code> <-> </code>		<code>o-</code>	
<code><<-</code>		<code>- </code>		<code>o-o</code>	
<code><<->></code>		<code> -</code>		<code>-*</code>	
<code>>-</code>		<code> - </code>		<code>*-</code>	
<code>-<</code>		<code>-]</code>		<code>**-</code>	
<code>>-<</code>		<code>[-</code>		<code>o-*</code>	
<code>-<<</code>		<code>[-]</code>		<code>*-o</code>	

TABLA 11.4. Parámetros para los extremos de líneas rectas o curvas.

11.5. Polígonos

PSTricks tiene cuatro macros específicos para polígonos: las *opciones* de trazado de líneas son las mismas mostradas en la Tabla 11.3.

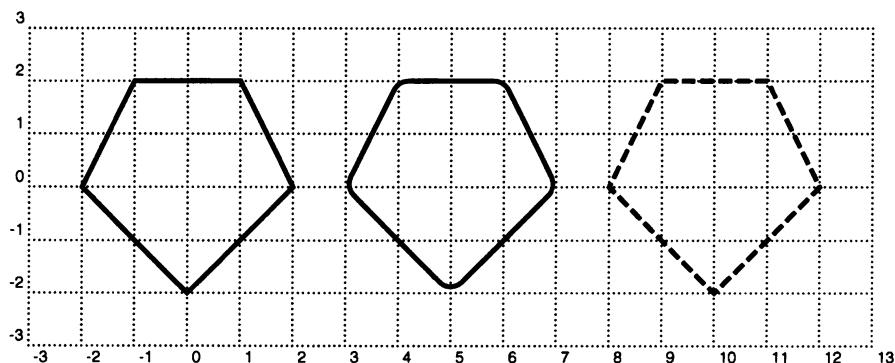
\pspolygon[*opciones*](x_0, y_0)(x_1, y_1)…(x_n, y_n). Traza un polígono uniendo los puntos $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$.

\psframe[*opciones*](x_1, y_1)(x_2, y_2). Traza un rectángulo cuyos vértices opuestos son (x_1, y_1) y (x_2, y_2) . Si se escribe solamente el punto (x_1, y_1) , se forma un rectángulo cuyos vértices opuestos son (x_1, y_1) y el origen de coordenadas $(0, 0)$. Para bordes redondeados se debe usar el parámetro **framearc** en vez de **linearc** y su valor debe ser un número entre 0 y 1.

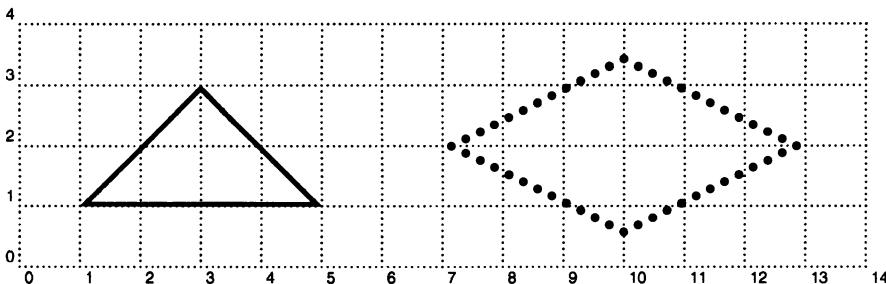
\pstriangle[*opciones*](x, y)(b, h). Dibuja un triángulo isóceles con base b , centrada en (x, y) , y altura h .

\psdiamond[*opciones*](x, y)(h, v). Dibuja un diamante (rombo) centrado en (x, y) , semidiagonal horizontal h y semidiagonal vertical v .

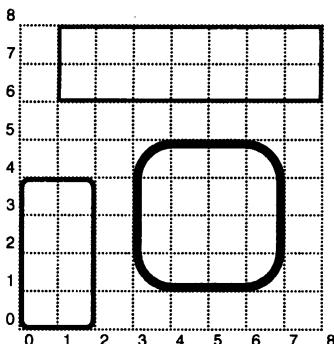
Example



```
\begin{center}
\psset{unit=0.7cm}
\begin{pspicture}(-3,-3)(13,3)\grilla
\pspolygon[linewidth=1.5pt] (0,-2)(2,0)(1,2)(-1,2)(-2,0)
\pspolygon[linewidth=1.5pt, linearc=0.3] (5,-2)(7,0)(6,2)(4,2)(3,0)
\pspolygon[linewidth=1.5pt, linestyle=dashed, dash=5pt 2pt](10,-2)%
(12,0)(11,2)(9,2)(8,0)
\end{pspicture}
\end{center}
```

Ejemplo

```
\begin{center}
\psset{unit=0.8cm}
\begin{pspicture}(0,0)(14,4)\grilla
\pstriangle[linewidth=1.5pt](3,1)(4,2)
\psdiamond[linewidth=3pt,linestyle=dotted,dotsep=3pt](10,2)(3,1.5)
\end{pspicture}
\end{center}
```

Ejemplo

```
\psset{unit=0.5cm}
\begin{pspicture}(0,0)(8,8) \grilla
\psframe[linewidth=1pt](1,6)(8,8)
\psframe[linewidth=1.5pt,framearc=0.2](2,4)
\psframe[linewidth=3pt,framearc=0.5](3,1)(7,5)
\end{pspicture}
```

11.6. Circunferencias y arcos de circunferencia

Los siguientes comandos sirven para trazar circunferencias y arcos; las *opciones* de trazado permitidas son las de la Tabla 11.3. Los ángulos se miden en grados sexagesimales, entre -360 y 360 .

\pscircle[opciones]{(x,y)}{radio}. Traza el círculo centrado en el punto (x, y) y radio dado.

\psarc[opciones]{flechas}{(x,y)}{radio}{ángulo A}{ángulo B}.

Traza un arco de la circunferencia centrada en (x, y) y radio dado, desde el ángulo A hasta el B , en la dirección anti-horaria.

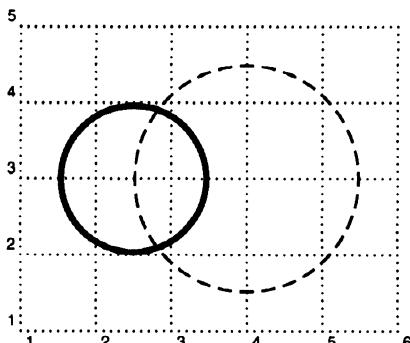
\psarcn[opciones]{flechas}(x,y){radio}{ángulo A}{ángulo B}.

Traza un arco de la circunferencia centrada en (x, y) y radio dado, desde el ángulo A hasta el B , en la dirección horaria.

\pswedge[opciones](x,y){radio}{ángulo A}{ángulo B}, Traza el sector circular (*wedge*) de la circunferencia centrada en (x, y) y radio dado, desde el ángulo A hasta el B , en la dirección anti-horaria.

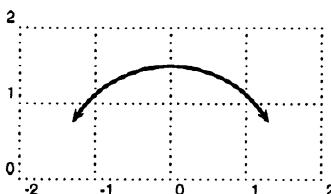
\psdisk(x,y){radio}. Es la versión simplificada de \pscircle para trazar círculos; no admite opciones.

Ejemplo



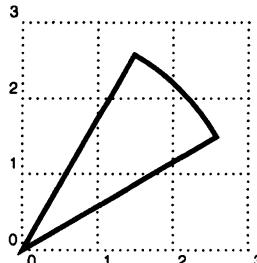
```
\begin{pspicture}(1,1)(6,5)\grilla
\pscircle[linewidth=2pt](2.5,3){1}
\pscircle[linestyle=dashed](4,3){1.5}
\end{pspicture}
```

Ejemplo

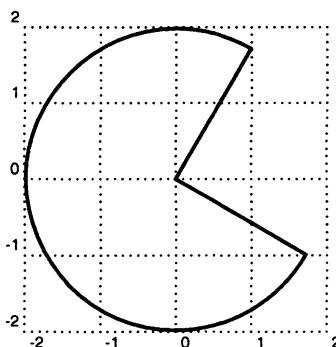


```
\begin{pspicture}(-2,0)(2,3)\grilla
\psarc[linewidth=1pt]{<->}(0,0){2}{30}{150}
\end{pspicture}
```

Ejemplo



```
\begin{center} \begin{pspicture}(0,0)(3,3)\grilla
\pswedge[linewidth=1.5pt](0,0){3}{30}{60}
\end{pspicture} \end{center}
```

Ejemplo

```
\begin{center}
\begin{pspicture}(-2,-2)(2,2)\grilla
\pswedge[linewidth=1pt](0,0){2}{60}{330}
\end{pspicture}
\end{center}
```

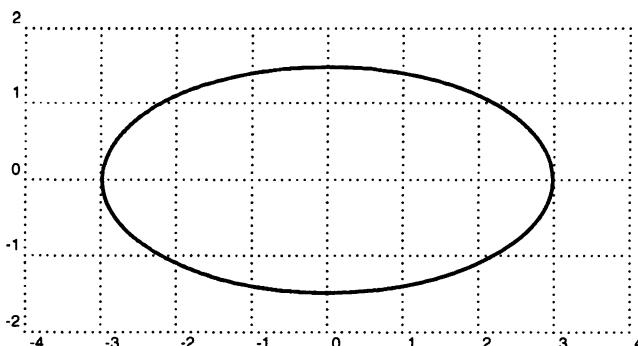
11.7. Elipses

La instrucción

\psellipse[*opciones*]{(x,y)}{(a,b)}

traza la elipse centrada en el punto (x, y) , semi-eje horizontal a y semi-eje vertical b . Las *opciones* de trazado permitidas son las de la Tabla 11.3.

Las elipses con ejes transversales se pueden obtener por rotación usando los comandos de la sección 11.14.

Ejemplo

```
\begin{center}
\begin{pspicture}(-4,-2)(4,2)\grilla
\psellipse[linewidth=1pt](0,0)(3,1.5)
\end{pspicture}
\end{center}
```

11.8. Paráboles

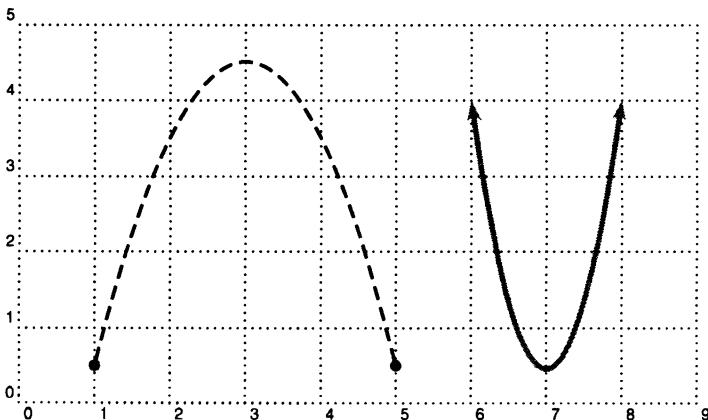
Para dibujar paráboles cuyo eje sea vertical se usa la instrucción

```
\parabola[opciones]{flechas}(x0,y0)(m1,m2)
```

la cual traza la parábola que pasa por el punto (x_0, y_0) y cuyo máximo o mínimo es (m_1, m_2) . Las *opciones* de trazado permitidas son las de la Tabla 11.3 y las *flechas* (o extremos) que se pueden usar son los mostrados en la Tabla 11.4. El parámetro *flechas* es opcional.

Las paráboles con ejes horizontales o transversales se pueden obtener por rotación usando los comandos de la sección 11.14.

Ejemplo



```
\begin{center}
\begin{pspicture}(0,0)(9,5)
\grilla
\parabola[linewidth=1pt,linestyle=dashed]{-*-*}(1,0.5)(3,4.5)
\parabola[linecolor=gray,linewidth=2pt]{<->}(6,4)(7,0.5)
\end{pspicture}
\end{center}
```

11.9. Relleno y sombreado de regiones. Parte I.

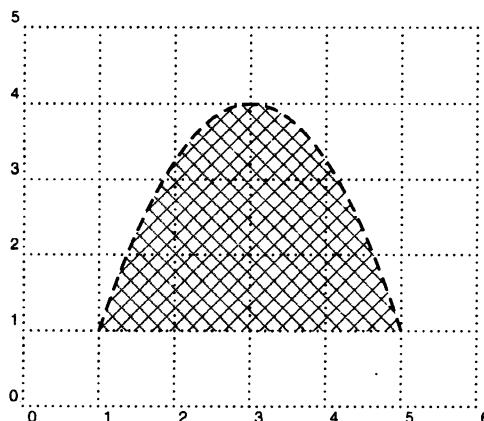
Hay tres tipos de relleno para regiones cerradas: relleno con líneas rectas (horizontales, verticales o transversales), relleno sólido con un color específico y sombreado gradual. El tipo de relleno se escoge con el parámetro *fillstyle*. La Tabla 11.5 muestra los valores permitidos por *fillstyle* y la Tabla 11.6 muestra parámetros adicionales para relleno con líneas rectas. Los parámetros que controlan el sombreado gradual aparecen en la Tabla 11.7.

none	Asumido por defecto
solid	Relleno “sólido” con el color especificado por medio del parámetro fillcolor . El color que asume por defecto fillcolor es white (blanco).
gradient	Sombreado gradual (véase también la Tabla 11.7).
vlines	 Líneas verticales. Inclinación por defecto: 45° en la dirección NO-SE.
hlines	 Líneas horizontales. Inclinación por defecto: 45° en la dirección SO-NE.
crosshatch	 Relleno cruzado con líneas transversales. Por defecto las líneas tienen inclinaciones de 45° en las direcciones NO-SE y SO-NE, respectivamente.

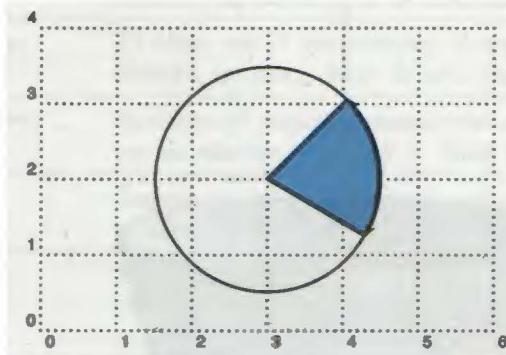
TABLA 11.5. Valores del parámetro **fillstyle** para relleno de regiones.

hatchangle	Inclinación de las líneas. Por defecto: 45°.
hatchwidth	Grosor de las líneas. Por defecto: 0.8pt.
hatchsep	Separación entre las líneas. Por defecto: 4pt.
hatchcolor	Color de las líneas. Por defecto: black .

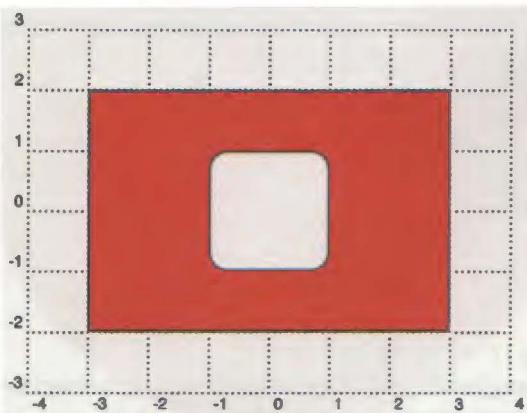
TABLA 11.6. Parámetros adicionales para relleno con líneas rectas.

Ejemplo

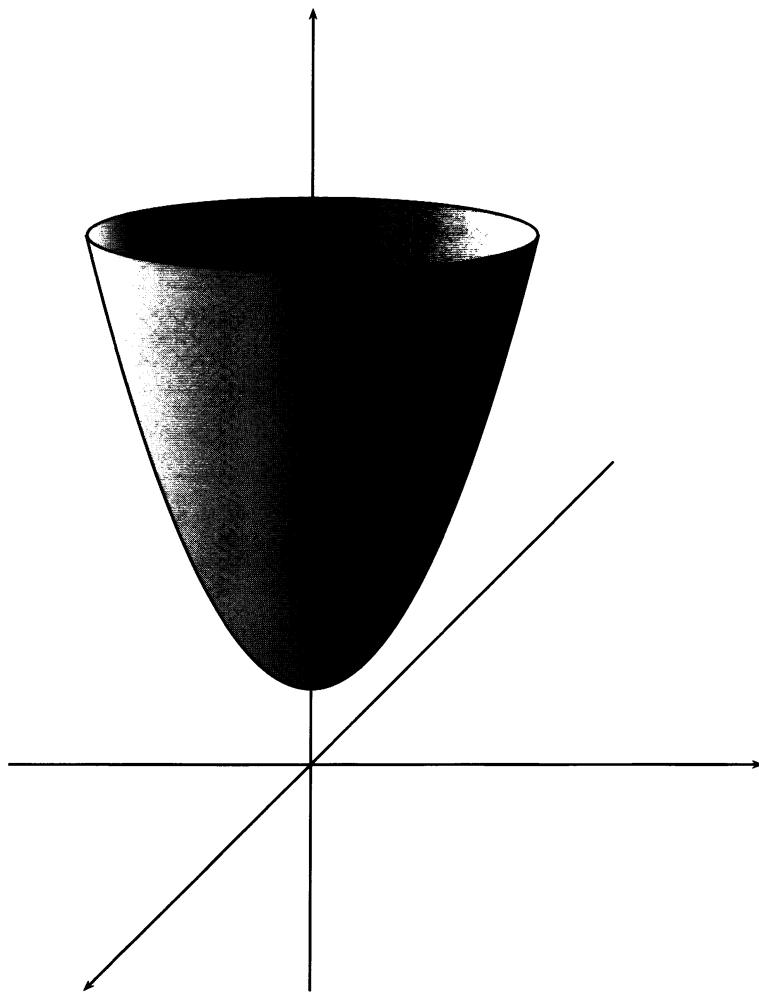
```
\begin{center}
\begin{pspicture}(0,0)(6,5)\grilla
\parabola [linestyle=dashed,fillstyle=crosshatch,
hatchcolor=gray](1,1)(3,4)
\end{pspicture}
\end{center}
```

Ejemplo

```
\begin{center}
\begin{pspicture}(0,0)(6,4)\grilla
\pscircle(3,2){1.5}
\pswedge[fillstyle=solid,fillcolor=cyan](3,2){1.5}{330}{45}
\psarc{<->}(3,2){1.5}{330}{45}
\end{pspicture}
\end{center}
```

Ejemplo

```
\begin{center}
\psset{unit=0.8cm}
\begin{pspicture}(-4,-3)(4,3)\grilla
\psframe[fillstyle=solid,fillcolor=red](-3,-2)(3,2)
\psframe[framearc=0.3,fillstyle=solid,fillcolor=white](-1,-1)(1,1)
\end{pspicture}
\end{center}
```

Ejemplo

```
\begin{center}
\begin{pspicture}(-4,-3)(6,10)
\parabola[gradangle=90,fillstyle=gradient,gradbegin=white,
    gradend=darkgray,gradmidpoint=1](-3,7)(0,1)
\psellipse[gradangle=90,fillstyle=gradient,gradbegin=white,
    gradend=darkgray,gradmidpoint=0.2](0,7)(3,0.5)
\psline{->}(-4,0)(6,0)
\psline{->}(4,4)(-3,-3)
\psline{->}(0,6.5)(0,10)
\psline(0,-3)(0,1)
\end{pspicture}
\end{center}
```

gradbegin= <i>color</i>	El color con el que se inicia el sombreado. Es, por defecto, un cierto tono de azul oscuro.
gradend= <i>color</i>	El color con el que termina el sombreado. Es, por defecto, un cierto tono de azul claro.
gradmidpoint= <i>n</i>	Posición del punto intermedio del sombreado; <i>n</i> debe ser un número entre 0 y 1. Valor por defecto: 0.9.
gradangle= <i>α</i>	Ángulo de rotación para el sombreado, medido en grados, entre -360 y 360. Valor por defecto: 0.
gradlines= <i>m</i>	El número de líneas (renglones) del sombreado. Entre mayor sea el número de líneas, más fino es el sombreado. Valor por defecto: 500.

TABLA 11.7. Opciones para el sombreado gradual de regiones, cuando se ha escogido **fillstyle=gradient**.

Ejemplo



```
\begin{center}
\begin{pspicture}(-4,-4)(4,4)
\psarc[linestyle=none,gradangle=60,fillstyle=gradient,
gradbegin=yellow,gradend=red,gradmidpoint=1,
gradlines=800](0,0){4}{0}{360}
\end{pspicture}
\end{center}
```

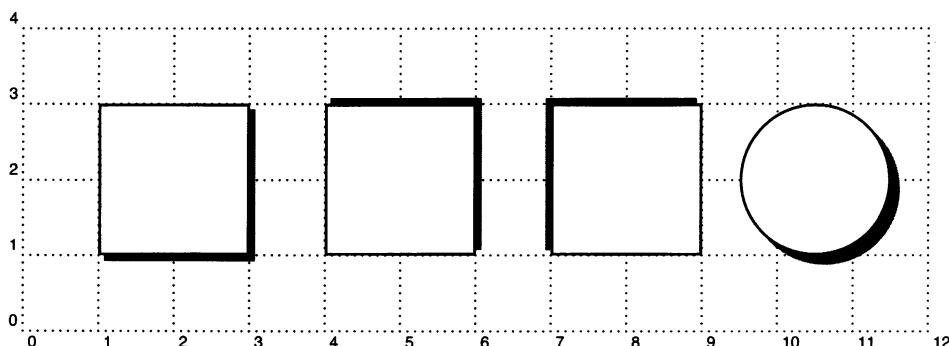
11.10. Sombras

Todas las líneas rectas y curvas admiten sombras, obtenidas con la opción **shadow=true**. En la Tabla 11.8 aparecen los parámetros que controlan el tamaño, la posición y el color de la sombra.

shadow=true/false	Determina si se traza o no una sombra a la curva. Valor por defecto: false .
shadowsize=n	Tamaño de la sombra; <i>n</i> debe ser una dimensión TeX . Valor por defecto: 3pt .
shadowangle=ángulo	Posición en la que aparece la sombra, medida en grados sexagesimales, entre -360 y 360 . Valor por defecto: -45 .
shadowcolor=color	Color de la sombra. Por defecto: darkgray .

TABLA 11.8. Parámetros para sombras de curvas.

Ejemplo



```
\begin{center}
\begin{pspicture}(12,4)\grilla
\psframe[shadow=true](1,1)(3,3)
\psframe[shadow=true,shadowangle=45](4,1)(6,3)
\psframe[shadow=true,shadowangle=135](7,1)(9,3)
\pscircle[shadow=true,shadowsize=5pt,shadowcolor=gray](10.5,2){1}
\end{pspicture}
\end{center}
```

11.11. Puntos

La instrucción

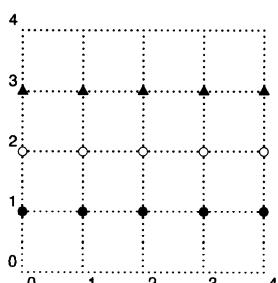
```
\psdots [dotstyle=...](x0,y0)(x1,y1) ... (xn,yn)
```

coloca un “punto” en cada (x_i, y_i) . Se puede escoger la apariencia que tiene el “punto” entre toda una gama de estilos. Esto se hace por medio del parámetro `dotstyle`, cuyos valores permitidos se encuentran en la Tabla 11.9. Por defecto, `dotstyle=*`.

Estilo	●	○	◊	◆	□	△	▲
*	● ● ● ● ● ● ● ● ● ●						
+	+	+	+	+	+	+	+
o	○ ○ ○ ○ ○ ○ ○ ○ ○ ○						
x	×	×	×	×	×	×	×
	- - - - - - - -						
asterisk	* * * * * * * * *						
diamond	◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊ ◊						
diamond*	◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆						
oplus	⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕						
otimes	⊗ ⊗ ⊗ ⊗ ⊗ ⊗ ⊗ ⊗ ⊗ ⊗						
pentagon	◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇ ◇						
pentagon*	★ ★ ★ ★ ★ ★ ★ ★ ★ ★						
square	□ □ □ □ □ □ □ □ □ □						
square*	■ ■ ■ ■ ■ ■ ■ ■ ■ ■						
triangle	△ △ △ △ △ △ △ △ △ △						
triangle*	▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲						

TABLA 11.9. Valores permitidos del parámetro `dotstyle`.

Ejemplo

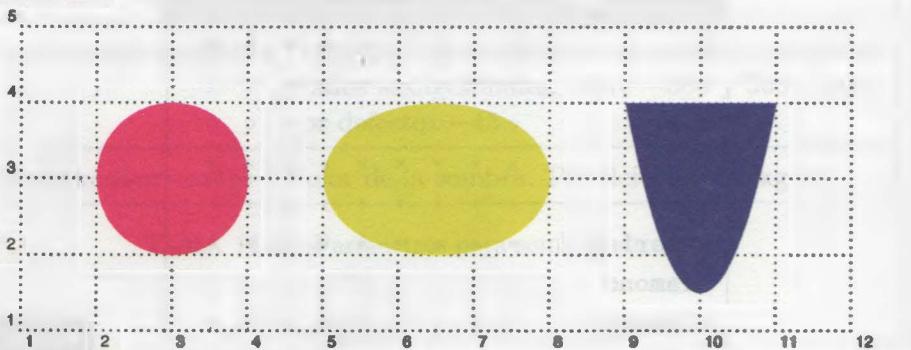


```
\psset{unit=0.8cm}
\begin{pspicture}(0,0)(4,4)\grilla
\psdots(0,1)(1,1)(2,1)(3,1)(4,1)
\psdots[dotstyle=o](0,2)(1,2)(2,2)(3,2)(4,2)
\psdots[dotstyle=triangle*](0,3)(1,3)(2,3)%
(3,3)(4,3)
\end{pspicture}
```

11.12. Los comandos “estrella”

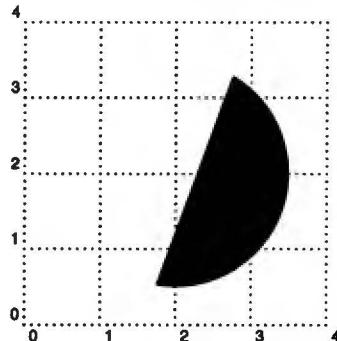
Cada uno de los comandos gráficos de **PSTricks** (excepto aquéllos que comienzan con \q) tiene su “versión estrella”. Por ejemplo, \pscircle*, \pspolygon*, \pswedge*, etc. Con ellos, el objeto gráfico correspondiente se “rellena”, con el color determinado por el parámetro `linecolor`. Automáticamente `linewidth` toma el valor 0, `fillstyle` asume el valor `solid`, `linestyle` el valor `none` y `fillcolor` utiliza el color especificado por `linecolor`.

Ejemplo



```
\begin{pspicture}(1,1)(12,5)\grilla
\pscircle*[linecolor=magenta](3,3){1}
\psellipse*[linecolor=yellow](6.5,3)(1.5,1)
\parabola*[linecolor=blue](9,4)(10,1.5)
\end{pspicture}
```

Ejemplo



```
\begin{center}
\begin{pspicture}(4,4)\grilla
\psarc*[linecolor=gray](2,2){1.5}{260}{60}
\end{pspicture}
\end{center}
```

11.13. Texto enmarcado

Los comandos de la (Tabla 11.10), para enmarcar *texto*, no requieren ni coordenadas ni el entorno `\begin{picture} ... \end{picture}`. En ese sentido son análogos al comando `\fbox{...}` de L^AT_EX (sección 3.15): el tamaño de la caja o marco depende del *texto* que encierra y es escogido automáticamente por **PSTricks**.

Comando	Tipo de marco
<code>\psframebox[opciones]{texto}</code>	Marco rectangular
<code>\psdblframebox[opciones]{texto}</code>	Doble marco rectangular
<code>\pstribobox[opciones]{texto}</code>	Marco triangular
<code>\psdiabox[opciones]{texto}</code>	Marco en forma de rombo
<code>\pscirclebox[opciones]{texto}</code>	Marco circular
<code>\psovalbox[opciones]{texto}</code>	Marco oval o elíptico

TABLA 11.10. Comandos para enmarcar texto, fórmulas y tablas.

El *texto* propiamente dicho puede ser texto normal, fórmulas o incluso tablas. Como *opciones* se pueden usar las de la Tabla 11.3 (para las líneas del marco), las de la Tabla 11.8 (para sombras), las de la Tabla 11.5 (para el relleno). Hay un parámetro adicional, `framesep=n`, que controla la distancia entre los bordes del marco y el texto que encierra (*n* debe ser una dimensión T_EX). El valor por defecto de `framesep` es 3pt.

Ejemplos

$$e^{\pi i} + 1 = 0$$

```
\psframebox{\large $e^{\pi i}+1=0$}
```

$$\boxed{e^{\pi i} + 1 = 0}$$

```
\psdblframebox[linestyle=dashed]{\large $e^{\pi i}+1=0$}
```

$$\boxed{e^{\pi i} + 1 = 0}$$

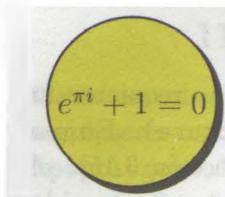
```
\psdblframebox[linestyle=dashed,framesep=8pt]{\large $e^{\pi i}+1=0$}
```

$$e^{\pi i} + 1 = 0$$

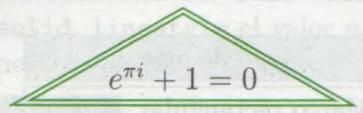
```
\psframebox[shadow=true]{\large $e^{\pi i}+1=0$}
```

$$\boxed{e^{\pi i} + 1 = 0}$$

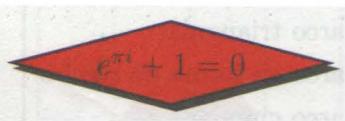
```
\psframebox[fillstyle=solid,fillcolor=black]{\Large \white $e^{\pi i}+1=0$}
```



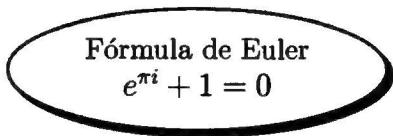
```
\pscirclebox[shadow=true,fillstyle=solid,
            fillcolor=yellow]{\large $e^{\pi i}+1=0$}
```



```
\pstribox[doubleline=true,
          linecolor=green]
          {\large $e^{\pi i}+1=0$}
```



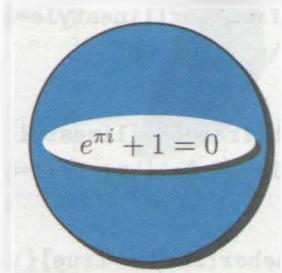
```
\psdiabox[linewidth=1pt,shadow=true,
          fillstyle=solid,fillcolor=red]
          {\large $e^{\pi i}+1=0$}
```



```
\psovalbox[shadow=true]
{\begin{tabular}{c}
Fórmula de Euler \\
\large $e^{\pi i}+1=0$ \\
\end{tabular}}
```

Los comandos de la Tabla 11.10 tienen su “versión estrella” (`\psframebox*`, `\pstribox*`, etc.) que son de particular interés: el color de los marcos y de su interior está determinado por el parámetro `fillcolor` (en lugar de `linecolor`); como resultado, el marco y su contenido ocultan lo que haya detrás. Esto se aprecia en el siguiente ejemplo; recuérdese que el valor por defecto de `fillcolor` es `white` (blanco), (véase la Tabla 11.5).

Ejemplo El marco `\psovalbox*` oculta lo que hay detrás de él.



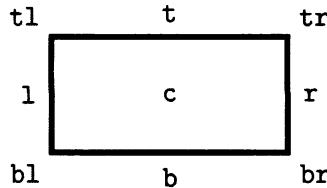
```
\begin{center}
\pscirclebox[shadow=true,fillstyle=solid,fillcolor=cyan]%
{\psovalbox*[shadow=true]{$e^{\pi i}+1=0$}} \end{center}
```

11.14. Colocación y rotación de objetos

Uno de los comandos más versátiles de **PSTricks** es `\rput`; con él se pueden colocar objetos (rotados o no) en posiciones especificadas por el usuario. La instrucción

`\rput[posición]{ángulo de rotación}(x,y){objeto}`

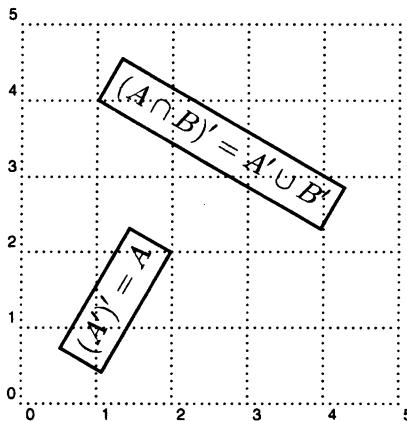
coloca un *objeto* (texto, tabla o gráfica) en el punto (x, y) . Los dos argumentos `[posición]` y `{ángulo de rotación}` son opcionales. El primero de ellos se refiere a la posición en la que se coloca la caja L^AT_EX que contiene el *objeto*, con respecto al punto (x, y) . El valor por defecto es `[c]`; es decir, el *objeto* aparece centrado en el punto (x, y) . Los demás valores de `[posición]` se pueden visualizar en el siguiente diagrama:



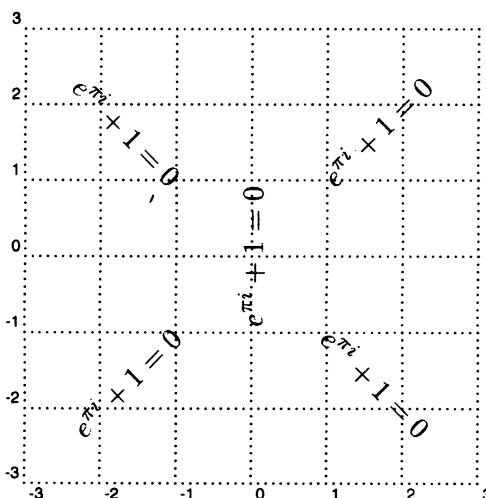
Al usar `[bl]`, por ejemplo, la caja L^AT_EX que contiene el *objeto* aparece colocada debajo y a la izquierda del punto (x, y) .

El *ángulo de rotación* se mide en grados sexagesimales, entre -360 y 360 . Hay algunas abreviaturas que se pueden usar para ángulos, a saber:

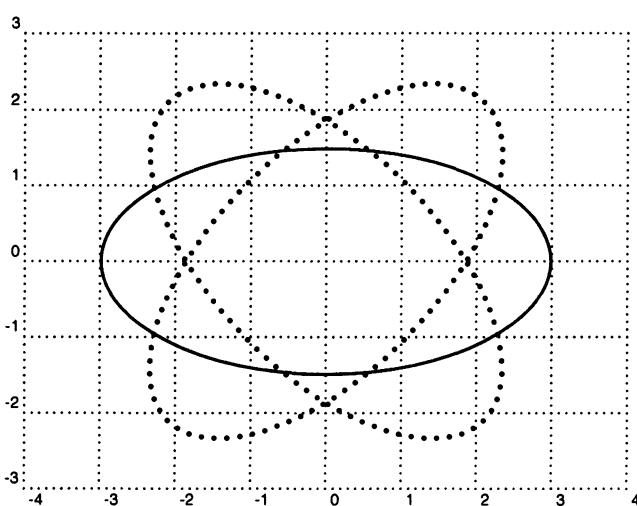
Letra	Abreviación de	Equivale a
U	Up	0
L	Left	90
D	Down	180
R	Right	270
N	North	0
W	West	-90
S	South	-180
E	East	-270

Ejemplo

```
\begin{center}
\begin{pspicture}(5,5)
\grilla
\rput[b]{-30}(1,4){\psframebox{$(A \cap B)' = A' \cup B'$}}
\rput[br]{60}(2,2){\psframebox{$(A')' = A$}}
\end{pspicture}
\end{center}
```

Ejemplo

```
\begin{center}
\begin{pspicture}(-3,-3)(3,3)
\grilla
\rput{W}{(0,0)}{$e^{\pi i}+1=0$}
\rput[1]{45}{(1,1)}{$e^{\pi i}+1=0$}
\rput[1]{-45}{(1,-1)}{$e^{-\pi i}+1=0$}
\rput[r]{-45}{(-1,1)}{$e^{-\pi i}+1=0$}
\rput[r]{45}{(-1,-1)}{$e^{\pi i}+1=0$}
\end{pspicture}
\end{center}
```

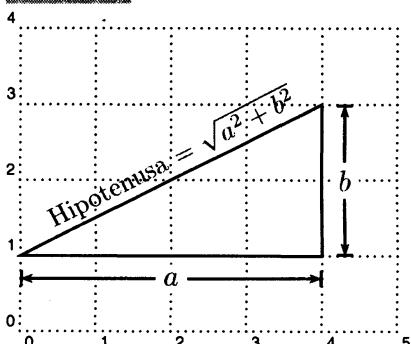
Ejemplo

```
\begin{center}
\begin{pspicture}(-4,-3)(4,3)\grilla
\psellipse(0,0)(3,1.5)
\rput{45}(0,0){%
    \psellipse[linestyle=dotted,linewidth=1.8pt](0,0)(3,1.5)}
\rput{135}(0,0){%
    \psellipse[linestyle=dotted,linewidth=1.8pt](0,0)(3,1.5)}
\end{pspicture}
\end{center}
```

El comando `\rput` tiene su “versión estrella”,

`\rput*[posición]{ángulo de rotación}{(x,y)}{objeto}`

que tiene el mismo efecto que `\rput`, excepto que la caja L^AT_EX que contiene el *objeto* en cuestión oculta todo lo que está detrás de ella. En el siguiente ejemplo, las cajas L^AT_EX con las letras *a* y *b* ocultan parte de los segmentos de recta que han sido trazados con `\psline{|<->|}`.

Ejemplo

```
\begin{pspicture}(5,4)\grilla
\pspolygon(0,1)(4,1)(4,3)
\psline{|<->|}(4.3,1)(4.3,3)
\psline{|<->|}(0,0.7)(4,0.7)
\rput*(2,0.7){$a$}
\rput*(4.3,2){$b$}
\rput{27}(2,2.3){\small{Hipotenusa
= $\sqrt{a^2+b^2}$}}
\end{pspicture}
```

11.15. Colocación de etiquetas

Aunque el comando `\rput` de la sección 11.14 se puede usar para colocar etiquetas (letras, palabras, fórmulas) en posiciones determinadas, **PSTricks** tiene el comando `\uput`, específicamente diseñado para tal propósito. Se usa en la forma

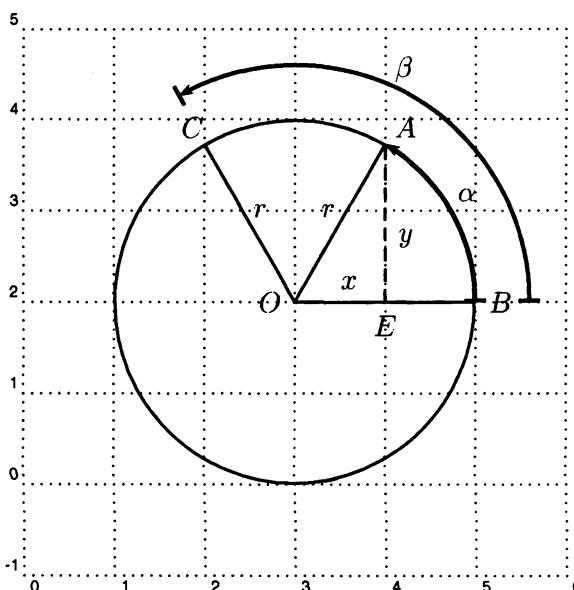
```
\uput{separación}[referencia]{ángulo de rotación}(x,y){objeto}
```

donde el argumento `{ángulo de rotación}` (opcional) tiene el mismo significado que en el comando `\rput`, y para él se pueden usar las mismas abreviaturas citadas en la sección 11.14. El argumento `{separación}` (también opcional) tiene un valor por defecto de `5pt` y constituye la principal diferencia entre `\rput` y `\uput`. Usando `\uput`, el `{objeto}` que se coloca aparece distanciado del punto (x, y) , lo que facilita la tarea de colocar etiquetas.

El argumento `[referencia]` es un ángulo, entre -360 y 360 grados, y representa la posición en la que se coloca la caja L^AT_EX que contiene el `objeto`, con respecto al punto (x, y) . Puesto que la colocación de etiquetas en gráficas generalmente se hace en ángulos de 45 grados, las siguientes abreviaciones son útiles para el argumento `[referencia]`. Enfatizamos que este argumento es *obligatorio*.

Letra	Abreviación de	Equivale a
r	right	0
u	up	90
l	left	180
d	down	270
ur	up-right	45
ul	up-left	135
dl	down-left	225
dr	down-right	315

También existe el comando homólogo `\uput*` que tiene la misma sintaxis y el mismo efecto que `\uput`, excepto que la caja L^AT_EX que contiene el `objeto` en cuestión oculta todo lo que está detrás de ella (de manera similar a lo que hace la instrucción `\rput*`, presentado en la sección 11.14).

Ejemplo

```

\begin{center}
\psset{unit=1.2cm}
\begin{pspicture}(0,-1)(6,5)
\grilla
\pscircle(3,2){2}
\pswedge(3,2){2}{0}{60}
\pswedge(3,2){2}{0}{120}
\psarc[linewidth=1.2pt]{|->}(3,2){2}{0}{60}
\psarc[linewidth=1.2pt]{|->|}(3,2){2.6}{0}{120}
\psline[linestyle=dashed](4,2)(4,3.7)
\put[1](3,2){$O$}
\put[ur](4,3.7){$A$}
\put[r](5,2){$B$}
\put[ul](2.1,3.7){$C$}
\put[d](4,2){$E$}
\put[u](3.6,2){$x$}
\put[r](4,2.7){$y$}
\put[l](3.6,3){$r$}
\put[r](2.4,3){$r$}
\put[ur](4.7,3){$\alpha$}
\put[ur](4,4.3){$\beta$}
\end{pspicture}
\end{center}

```

11.16. Ejes coordenados

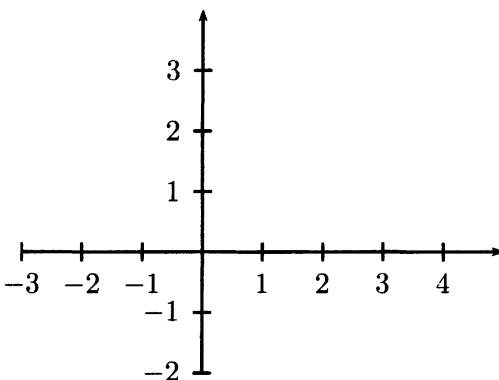
PSTricks posee un macro muy flexible para trazar ejes coordenados cartesianos. La instrucción

$\backslash\text{psaxes}[opciones]\{flechas\}(x_0,y_0)(x_1,y_1)(x_2,y_2)$

traza un par de ejes coordenados en el rectángulo cuyos vértices opuestos son (x_1, y_1) y (x_2, y_2) . El origen del sistema coordenado está localizado en (x_0, y_0) . Las diferentes opciones disponibles aparecen en la Tabla 11.11. Además, se pueden utilizar para los ejes las opciones de **linestyle** (Tabla 11.3).

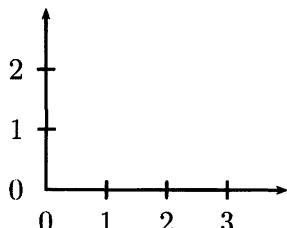
labels=all/x/y/none	Determina si las etiquetas aparecen en ambos ejes, en el eje <i>x</i> , en el eje <i>y</i> o en ninguno. Por defecto: all .
showorigin=true/false	Determina si el origen del sistema coordenado definido con \psaxes lleva o no etiquetas. Por defecto: true (hay etiquetas).
ticks=all/x/y/none	Determina si las marcas (ticks) aparecen en ambos ejes, en el eje <i>x</i> , en el eje <i>y</i> o en ninguno. Por defecto: all .
tickstyle=full/top/bottom	Determina si las marcas (ticks) se extienden por ambos lados de los ejes (full), en el mismo lado de las etiquetas (bottom) o en el lado opuesto de las etiquetas (top). Por defecto: full .
ticksize=<i>m</i>	Longitud de las marcas (ticks); <i>m</i> debe ser una dimensión T<small>E</small>X . Por defecto: 3pt .
axesstyle=axes/frame/none	Determina si se trazan dos ejes coordinados (axes), un marco rectangular (frame) o si no se trazan ejes (none). Por defecto: axes .
Ox=n	Etiqueta en el origen, eje horizontal. Por defecto: 0.
Oy=n	Etiqueta en el origen, eje vertical. Por defecto: 0.
Dx=n	Incremento entre etiquetas consecutivas en el eje horizontal. Por defecto: 1.
Dy=n	Incremento entre etiquetas consecutivas en el eje vertical. Por defecto: 1.
dx=n	Distancia entre las marcas (ticks) en el eje horizontal. Por defecto: dx=xunit .
dy=n	Distancia entre las marcas (ticks) en el eje vertical. Por defecto: dy=yunit .

TABLA 11.11. Opciones del comando **\psaxes**.

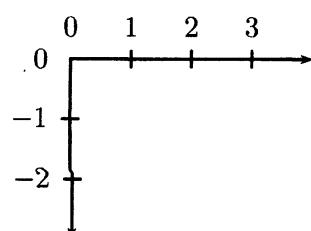
Ejemplo

```
\begin{center}
\psset{unit=0.8cm}
\begin{pspicture}(-3,-2)(5,4)
\psaxes{->}(0,0)(-3,-2)(5,4)
\end{pspicture}
\end{center}
```

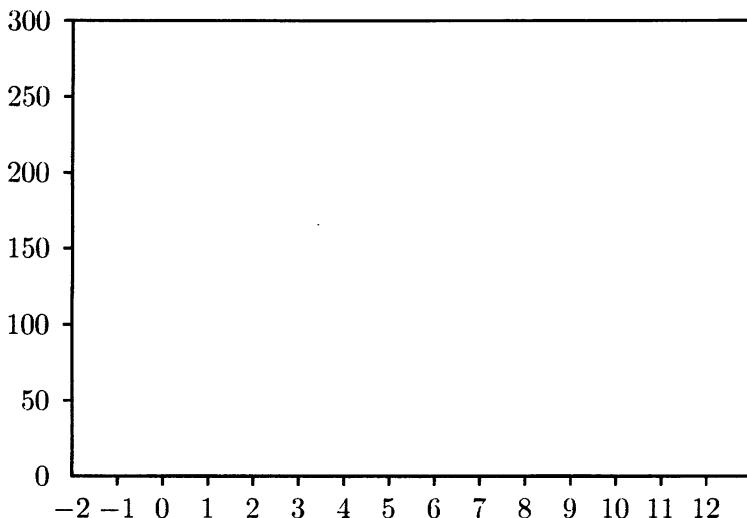
Los puntos (x_0, y_0) y (x_1, y_1) se pueden omitir en el comando `\psaxes`, en cuyo caso el sistema coordenado tiene su origen en $(0, 0)$, con x_2 unidades en el eje horizontal y y_2 unidades en el eje vertical.

Ejemplo

```
\begin{pspicture}(4,3)
\psaxes{->}(4,3)
\end{pspicture}
```

Ejemplo

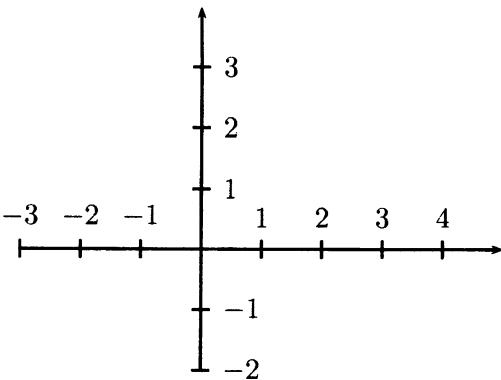
```
\begin{pspicture}(4,-3)
\psaxes{->}(4,-3)
\end{pspicture}
```

Ejemplo

```
\begin{center}
\psset{xunit=1cm,yunit=1.5cm}
\begin{pspicture}(9,4)
\psaxes[axesstyle=frame,tickstyle=bottom,0x=-2,dx=0.6cm,dy=1cm,
Dy=50](9,4)
\end{pspicture}
\end{center}
```

La posición de las etiquetas en los ejes (a la izquierda, a la derecha, arriba, abajo) se puede controlar cambiando el orden en el que se escriben los vértices opuestos (x_1, y_1) y (x_2, y_2) en el comando `\psaxes`. Esto se aprecia en el siguiente ejemplo.

Ejemplo Compárese la posición de las etiquetas en los ejes, con respecto al primer ejemplo de la página 337. Obsérvese que el parámetro de las flechas es `{<-}`, en lugar de `{->}`.



```
\psset{unit=0.8cm}
\begin{pspicture}(-3,-2)(5,4)
\psaxes{<-}(0,0)(5,4)(-3,-2)
\end{pspicture}
```

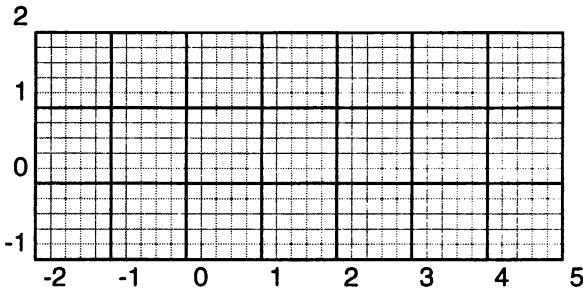
11.17. Grillas

La instrucción

\psgrid[*opciones*](x_1, y_1)(x_2, y_2)

traza una grilla con vértices opuestos (x_1, y_1) y (x_2, y_2), e intervalos numera-
dos. Las diferentes *opciones* disponibles aparecen en la Tabla 11.12. El co-
mando simple \psgrid (sin coordenadas) traza una grilla que se extiende a
lo largo y ancho del área establecida con \begin{pspicture}(a_1, b_1)(a_2, b_2).

Ejemplo

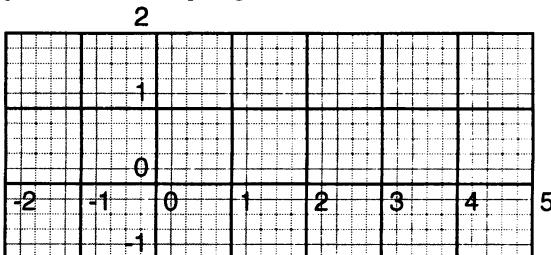


```
\begin{center}
\begin{pspicture}(-2,-1)(5,2)
\psgrid(-2,-1)(5,2)
\end{pspicture}
\end{center}
```

El comando \psgrid admite un argumento opcional, un punto (x_0, y_0), que
indica el origen de numeración de la grilla. Se usa en la forma

\psgrid[*opciones*](x_0, y_0)(x_1, y_1)(x_2, y_2)

Ejemplo Compárese con el ejemplo anterior.

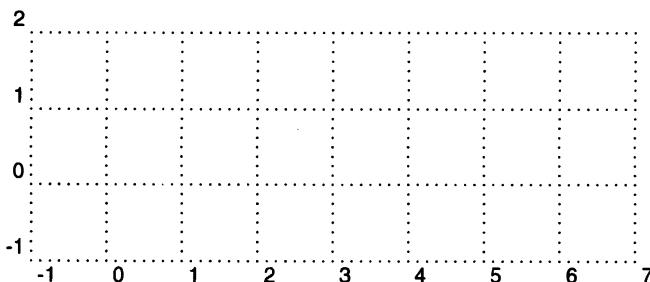


```
\begin{center}
\begin{pspicture}(-2,-1)(5,2)
\psgrid(0,0)(-2,-1)(5,2)
\end{pspicture}
\end{center}
```

gridwidth=n	Ancho de las líneas principales de la grilla. Por defecto: 0.8pt .
subgridwidth=n	Ancho de las líneas secundarias de la grilla. Por defecto: 0.4pt .
subgriddiv=n	Número de subdivisiones por unidad. Por defecto: 5 .
gridcolor=color	Color de las líneas principales de la grilla. Por defecto: black .
subgridcolor=color	Color de las líneas secundarias de la grilla. Por defecto: gray .
gridlabels=n	Tamaño de los números que demarcan la grilla. Por defecto: 10pt .
gridlabelcolor=color	Color los números que demarcan la grilla. Por defecto: black .
griddots=n	Si $n > 0$, las líneas principales de la grilla son punteadas, con n puntos por unidad. Por defecto: 0 .
subgriddots=n	Si $n > 0$, las líneas secundarias de la grilla son punteadas, con n puntos por unidad. Por defecto: 0 .

TABLA 11.12. Opciones del comando **\psgrid**.

Ejemplo A continuación aparece una grilla punteada, similar a la que se obtiene con el comando **\grilla** definido en la página 312, pero con 8 puntos por unidad.



```
\begin{center}
\begin{pspicture}(-1,-1)(7,2)
\psgrid[subgriddiv=1,griddots=8,gridlabels=8pt] (-1,-1)(7,2)
\end{pspicture}
\end{center}
```

11.18. Repeticiones

El comando `\multirput` es una extensión de `\rput` (sección 11.14). La instrucción

```
\multirput[posición]{ángulo de rotación}(x,y)( $\Delta x$ , $\Delta y$ ){n}{objeto}
```

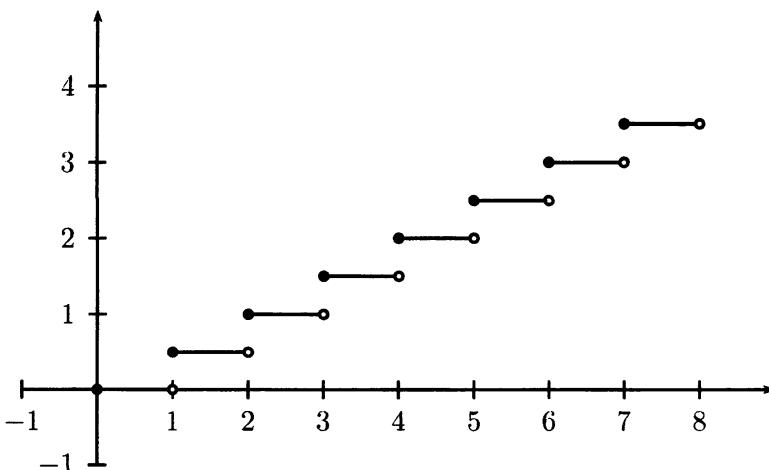
coloca *n* copias del *objeto* dado, primero en el punto (*x*, *y*), luego en el punto (*x* + Δx , *y* + Δy), a continuación en el punto (*x* + 2 Δx , *y* + 2 Δy), y así sucesivamente. Los dos argumentos [*posición*] y [*ángulo de rotación*] son opcionales y tienen el mismo significado que en el comando `\rput`. Las coordenadas que aparecen en `\multirput` deben ser cartesianas.

Cuando el *objeto* es una gráfica relativamente compleja es más apropiado usar el comando `\multipsp`, el cual no usa el argumento [*posición*]. Su sintaxis es:

```
\multipsp{ángulo de rotación}(x,y)( $\Delta x$ , $\Delta y$ ){n}{gráfica}
```

A diferencia de `\multirput`, el comando `\multipsp` admite coordenadas de cualquier tipo (véase al respecto la sección 11.25).

Ejemplo



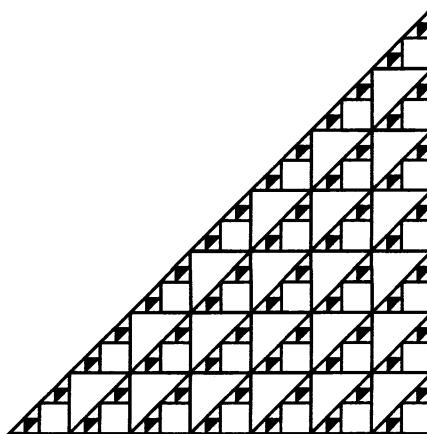
```
\begin{center}
\begin{pspicture}(-1,-1)(9,5)
\psaxes{->}(0,0)(-1,-1)(9,5)
\multipsp(0,0)(1,0.5){8}{\psline*[linewidth=1pt]{*-o}(1,0)}
\end{pspicture}
\end{center}
```

Ejemplo

La figura  está definida, bajo el nombre de `\figurin`, por medio de

```
\newcommand{\figurin}{\begin{pspicture}(0.8,0.8)
\psline*(0.2,0)(0.2,0.2)(0.4,0.2)
\psline(0.4,0)(0.4,0.4)(0.8,0.4)
\psline*(0.6,0.4)(0.6,0.6)(0.8,0.6)
\pspolygon(0,0)(0.8,0)(0.8,0.8)
\end{pspicture}}
```

El `\figurin` se coloca 7 veces en la diagonal que parte del punto $(0, 0)$, 6 veces en la diagonal que parte del punto $(0.8, 0)$, 5 veces en la diagonal que parte de $(1.6, 0)$, 4 veces en la diagonal que parte de $(2.4, 0)$, 3 veces en la diagonal que parte de $(3.2, 0)$, 2 veces en la diagonal que parte de $(4, 0)$ y una vez en el punto $(4.8, 0)$, para obtener el siguiente triángulo:



```
\begin{center}
\begin{pspicture}(6,6)\grilla
\multips(0,0)(0.8,0.8){7}{\figurin}
\multips(0.8,0)(0.8,0.8){6}{\figurin}
\multips(1.6,0)(0.8,0.8){5}{\figurin}
\multips(2.4,0)(0.8,0.8){4}{\figurin}
\multips(3.2,0)(0.8,0.8){3}{\figurin}
\multips(4,0)(0.8,0.8){2}{\figurin}
\multips(4.8,0)(0.8,0.8){1}{\figurin}
\end{pspicture}
\end{center}
```

11.19. Objetos gráficos personalizados

Los comandos de **PSTricks** para objetos gráficos con parámetros se pueden personalizar. Más precisamente, con la instrucción `\newpsobject` el usuario puede definir un comando personal `\nombre` que represente un objeto gráfico de **PSTricks**, con ciertos parámetros particulares. Esto se hace en la forma

```
\newpsobject{nombre}{objeto}{parámetros}
```

El comando `\objeto` debe existir en **PSTricks**. Nótese que en la definición de `\newpsobject` se escribe `nombre` en vez de `\nombre` y `objeto` en vez de `\objeto`, a pesar que se está definiendo el comando `\nombre` a partir del ya existente comando `\objeto`. Los parámetros se separan con comas y deben ser parámetros válidos para la instrucción `\objeto`.

Ejemplo Si en el documento aparecen muchos círculos punteados de color azul, podemos definir un comando especial, llamado `\circulo`, en la siguiente forma:

```
\newpsobject{circulo}{pscircle}{linestyle=dotted, linecolor=blue}
```

El comando `\circulo` así definido se puede ahora usar con otros parámetros (radio, centro, etc). Por ejemplo, `\circulo(3,2){2.5}` traza el círculo de radio 2.5, centrado en (3,2) y punteado en color azul. También se puede escribir algo como `\circulo[linewidth=2pt](0,2){2.5}`.

Ejemplo El lector puede ahora entender la definición de `\grilla` dada en la página 312. En la definición se usan parámetros de `\psgrid` que aparecen en la Tabla 11.12.

```
\newpsobject{grilla}{psgrid}{subgriddiv=1, griddots=10,%
gridlabels=6pt}
```

Ejemplo Si los ejes coordenados utilizados en un documento tienen siempre los mismos parámetros, se puede definir un comando genérico llamado, por ejemplo, `\ejes`.

```
\newpsobject{ejes}{psaxes}{showorigin=false, tickstyle=top,%
ticksize=2pt}
```

Con esta definición, se pueden trazar los ejes indicando solamente las flechas o extremos y las coordenadas concretas; por ejemplo,

```
\ejes{->}(0,0)(-3,-2)(5,4)
```

11.20. Trazado de curvas por interpolación I.

En esta sección presentamos los comandos básicos `\pscurve`, `\psccurve` y `\psecurve` que **PSTricks** tiene para el trazado de curvas por interpolación, a partir de un número dado de puntos. En la sección siguiente se describen otros comandos relacionados. Por defecto, la interpolación se hace de tal manera que para tres puntos consecutivos A , B y C , en la lista de puntos dados, la curva en el punto B es perpendicular a la bisectriz del ángulo $\angle ABC$.

`\pscurve[opciones]{flechas}(x_1, y_1)(x_2, y_2) \cdots (x_n, y_n)`. Traza una curva que pasa por los puntos $(x_1, y_1), \dots, (x_n, y_n)$.

`\psccurve[opciones]{flechas}(x_1, y_1)(x_2, y_2) \cdots (x_n, y_n)`. Traza una curva cerrada que pasa por los puntos $(x_1, y_1), \dots, (x_n, y_n)$.

`\psecurve[opciones]{flechas}(x_1, y_1)(x_2, y_2) \cdots (x_n, y_n)`. Traza una curva que pasa por los puntos $(x_1, y_1), \dots, (x_n, y_n)$ pero el primer y último puntos de la lista no se muestran en la curva final. La “e” que aparece en el nombre `\psecurve` proviene de “end-points”.

El parámetro `{flechas}` para símbolos terminales (que es opcional) puede tomar cualquiera de los valores de la Tabla 11.4. Las *opciones* de trazado son las usuales (Tabla 11.3), además del parámetro `curvature` que permite modificar la curvatura de la curva obtenida; se usa en la forma

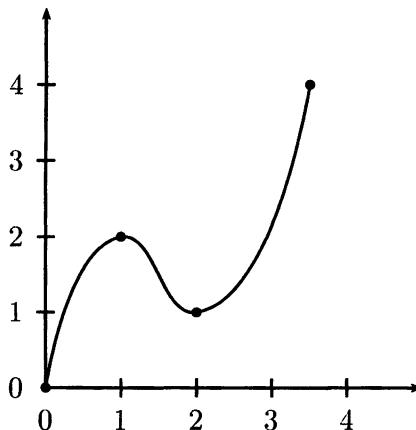
`curvature=r s t`

donde r , s y t son números reales entre -1 y 1 . Por defecto, $r = 1$, $s = 0.1$ y $t = 0$. Valores pequeños de r y s hacen que la curva sea más “apretada”; es decir, menos “redondeada”. El número t controla la pendiente de la curva en los diferentes puntos de la lista dada. Cuando $t = -1$, la interpolación se hace de tal manera que para tres puntos consecutivos A , B y C , la curva en el punto B es paralela al segmento AC .

El usuario tiene que variar los números r , s y t , por ensayo y error, hasta obtener los resultados más satisfactorios. En los dos ejemplos de la página siguiente se utilizan valores diferentes para el parámetro `curvature`, con una misma lista de puntos.

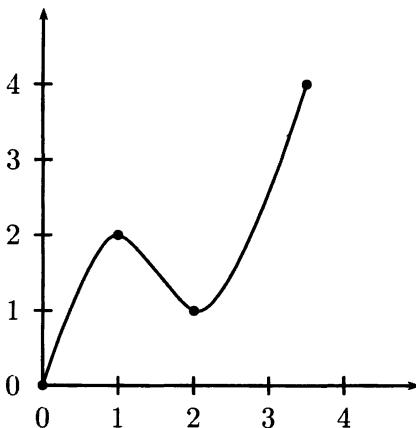
Un parámetro para el argumento `[opciones]`, que puede ser útil al trazar curvas por interpolación, es `showpoints=true`, con el cual los puntos dados $(x_1, y_1), \dots, (x_n, y_n)$ aparecen resaltados con el símbolo \bullet . El valor por defecto de `showpoints` es `false`. En lugar de \bullet se pueden escoger otros símbolos por medio del parámetro `dotstyle=\dots` (Tabla 11.9).

Ejemplo La siguiente curva ha sido obtenida con los valores por defecto del parámetro `curvature`.

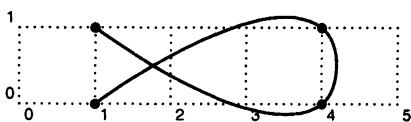


```
\begin{center}
\begin{pspicture}(5,5)
\psaxes{->}(5,5)
\pscurve[showpoints=true](0,0)(1,2)(2,1)(3.5,4)
\end{pspicture}
\end{center}
```

Ejemplo Los mismos puntos del ejemplo anterior pero con diferente curvatura.

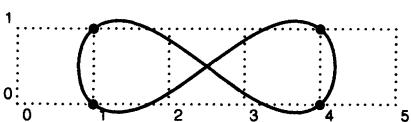


```
\begin{center}
\begin{pspicture}(5,5)
\psaxes{->}(5,5)
\pscurve[showpoints=true,curvature=0.5 0.1 1](0,0)(1,2)(2,1)(3.5,4)
\end{pspicture}
\end{center}
```

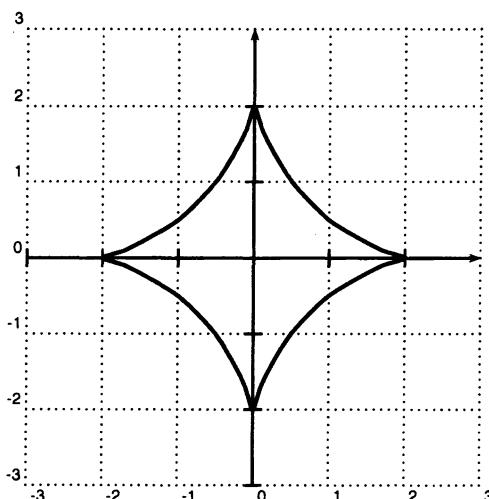
Ejemplo

```
\begin{pspicture}(5,1)\grilla
\pscurve[showpoints=true]%
(1,0)(4,1)(4,0)(1,1)
\end{pspicture}
```

Ejemplo Los mismos puntos del ejemplo anterior pero unidos en una curva cerrada, con el comando `\psccurve`.



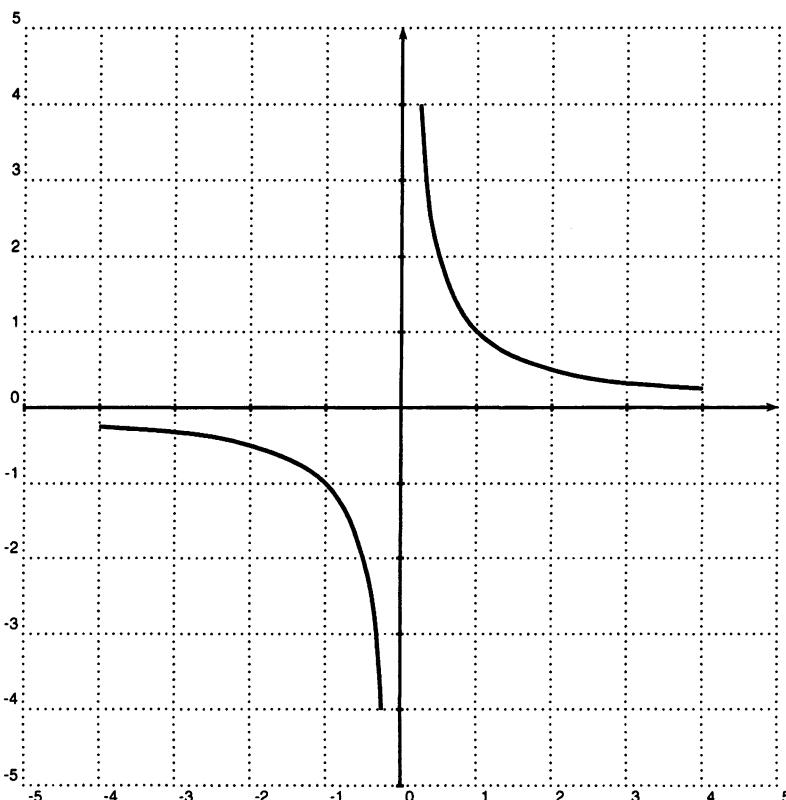
```
\begin{pspicture}(5,1)\grilla
\psccurve[showpoints=true]%
(1,0)(4,1)(4,0)(1,1)
\end{pspicture}
```

Ejemplo

```
\begin{center}
\begin{pspicture}(-3,-3)(3,3)\grilla
\psaxes[labels=none]{>}(0,0)(-3,-3)(3,3)
\psccurve[curvature=0.5 1 1, linewidth=1.2pt](2,0)(1.7,0.1)(1.5,0.2)%
(1,0.5)(0.5,1)(0.1,1.7)(0,2)(-0.1,1.7)(-0.5,1)(-1,0.5)(-1.5,0.2)%
(-1.7,0.1)(-2,0)(-1.7,-0.1)(-1.5,-0.2)(-1,-0.5)(-0.5,-1)%
(-0.1,-1.7)(0,-2)(0.1,-1.7)(0.5,-1)(1,-0.5)(1.5,-0.2)(1.7,-0.1)
\end{pspicture}
\end{center}
```

El comando `\pscurve` es de gran ayuda para trazar curvas con comportamiento asintótico, como la curva $y = 1/x$ del siguiente ejemplo. Para cada una de las ramas se dan siete puntos, y todos se utilizan en la interpolación, pero en la gráfica final sólo aparecen los cinco centrales.

Ejemplo



```
\begin{center}
\begin{pspicture}(-5,-5)(5,5)\grilla
\psaxes[labels=none,ticksize=1pt]{>}(0,0)(-5,-5)(5,5)
\pscurve[showpoints=true,dotstyle=diamond,linewidth=1.2pt]%
(0.125,8)(0.25,4)(0.5,2)(1,1)(2,0.5)(4,0.25)(8,0.125)
\pscurve[showpoints=true,dotstyle=diamond,linewidth=1.2pt]%
(-0.125,-8)(-0.25,-4)(-0.5,-2)(-1,-1)(-2,-0.5)(-4,-0.25)(-8,-0.125)
\end{pspicture}
\end{center}
```

11.21. Trazado de curvas por interpolación II.

Los comandos de esta sección tienen el propósito de trazar curvas por interpolación —como los comandos `\pscurve`, `\psccurve` y `\pseccurve` de la sección anterior— pero están concebidos para listas extensas de datos (pares de coordenadas) generadas por paquetes de computación científica como *Mathematica*, *Maple*, *Derive*, etc. Las coordenadas pueden estar encerradas entre corchetes `{ }`, en la forma `{x, y}`, o entre paréntesis `()`, en la forma `(x, y)`. La lista total de puntos se puede escribir explícitamente en el documento fuente L^AT_EX o se puede guardar en un archivo de texto independiente; en este último caso, los diferentes puntos `(x, y)` o `{x, y}` se deben separar entre sí con espacio en blanco.

Los comandos básicos⁷ son:

`\listplot[opciones]{lista de puntos}`. Traza una curva que pasa por los puntos de la lista dada.

`\savedata{\comando}[lista de puntos]`. Almacena con el nombre `\comando` la *lista de puntos* dada, para ser usada posteriormente con `\dataplot`.

`\readdata{\comando}{archivo}`. Almacena con el nombre `\comando` la lista de puntos contenida en el `archivo` indicado, para ser usada posteriormente con `\dataplot`. El `archivo` en cuestión debe ser un documento texto (preferiblemente con extensión `txt`) y debe estar en la misma carpeta o subdirectorio en el que aparece el documento fuente que se está procesando. El nombre dado al `archivo` no debe contener dígitos.

`\dataplot[opciones]{\comando}`. Traza una curva que pasa por los puntos almacenados por la instrucción `\comando`, previamente definida con `\savedata` o `\readdata`.

Como *opciones* de trazado se pueden usar las usuales de la Tabla 11.3, la opción `curvature` (sección 11.20) y las opciones del parámetro `plotstyle`. Este parámetro puede tomar los valores `dots`, `polygon`, `curve`, `ecurve`, `ccurve` y `line` (valor por defecto). Para obtener curvas “suaves” se recomienda usar `curve` (para curvas abiertas) y `ccurve` (para curvas cerradas). Al usar `plotstyle=dots` o `showpoints=true` el usuario puede escoger el tipo de puntos deseado, por medio del parámetro `dotstyle` (Tabla 11.9).

⁷Existe un comando adicional, `\fileplot[opciones]{archivo}`, pero no reconoce todos los estilos del parámetro `plotstyle` e ignora los parámetros `arrows`, `linearc` y `showpoints`; no recomendamos su uso.

Para curvas abiertas también se pueden usar las flechas o extremos de la Tabla 11.4 pero no con el usual argumento opcional `{...}` sino con el parámetro `arrows=...`.

Ejemplo Hemos usado el paquete *Mathematica* para calcular los valores de la función de Bessel de índice 2, $J_2(x)$, en el intervalo $[1.5, 35]$. Las funciones de Bessel se encuentran entre las numerosas funciones especiales implementadas en *Mathematica*. La instrucción

```
N[Table[{x,BesselJ[2,x]},{x,1.5,35,0.4}]]
```

de *Mathematica* calcula los valores de la función $J_2(x)$ desde 1.5 hasta 35 con incrementos de 0.4 y produce exactamente lo siguiente:

```
{ {1.5,-0.932194}, {1.9,-0.669879}, {2.3,-0.472617}, {2.7,-0.291887}, {3.1,-0.117535},  
{3.5,0.0453714}, {3.9,0.185763}, {4.3,0.29164}, {4.7,0.353431}, {5.1,0.366205},  
{5.5,0.330841}, {5.9,0.25417}, {6.3,0.148157}, {6.7,0.0282928}, {7.1,-0.088542},  
{7.5,-0.186414}, {7.9,-0.252526}, {8.3,-0.278936}, {8.7,-0.26356}, {9.1,-0.210322},  
{9.5,-0.128436}, {9.9,-0.0309445}, {10.3,0.0672726}, {10.7,0.151708}, {11.1,0.210337},  
{11.5,0.235309}, {11.9,0.223999}, {12.3,0.179252}, {12.7,0.108817}, {13.1,0.0240685},  
{13.5,-0.0617841}, {13.9,-0.135723}, {14.3,-0.186868}, {14.7,-0.208066},  
{15.1,-0.196877}, {15.5,-0.155834}, {15.9,-0.0919466}, {16.3,-0.0155355},  
{16.7,0.0614087}, {17.1,0.127085}, {17.5,0.171677}, {17.9,0.188807}, {18.3,0.176441},  
{18.7,0.137112}, {19.1,0.0774389}, {19.5,0.00703486}, {19.9,-0.0630267},  
{20.3,-0.121947}, {20.7,-0.160821}, {21.1,-0.174041}, {21.5,-0.160035},  
{21.9,-0.121466}, {22.3,-0.0647494}, {22.7,0.00101033}, {23.1,0.065464},  
{23.5,0.11864}, {23.9,0.152477}, {24.3,0.162044}, {24.7,0.14625}, {25.1,0.107953},  
{25.5,0.0534589}, {25.9,-0.00851053}, {26.3,-0.0682044}, {26.7,-0.116371},  
{27.1,-0.145686}, {27.5,-0.151862}, {27.9,-0.134269}, {28.3,-0.0959867},  
{28.7,-0.0432676}, {29.1,0.0154729}, {29.5,0.0709964}, {29.9,0.114689},  
{30.3,0.139893}, {30.7,0.14292}, {31.1,0.123583}, {31.5,0.08518570}, {31.9,0.033962},  
{32.3,-0.021931}, {32.7,-0.0737079}, {33.1,-0.113332}, {33.5,-0.134758},  
{33.9,-0.134855}, {34.3,-0.113858}, {34.7,-0.0752922} }
```

Esta tabla de valores podemos copiarla directamente en el documento fuente *LATEX* y almacenarla con el nombre deseado, por ejemplo, `\afuncion`:

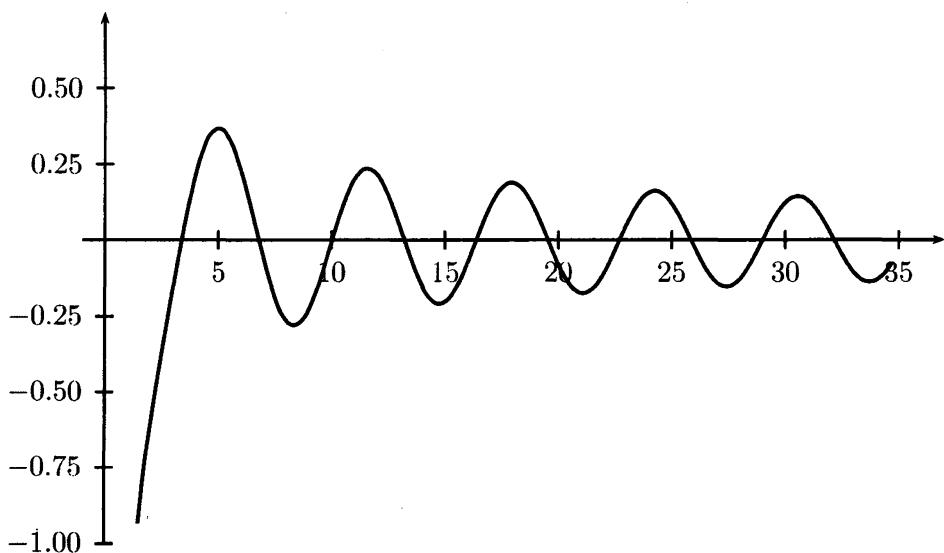
```
\savedata{\afuncion}{[1.5,-0.932194},{1.9,-0.669879}, ...  
... {33.9,-0.134855},{34.3,-0.113858},{34.7,-0.0752922}]
```

Obsérvese que, al usar `\savedata`, la lista de puntos se encierra entre paréntesis angulares `[]`, no entre corchetes `{ }`.

Alternativamente, la tabla de valores se puede almacenar en un archivo de texto independiente en el cual deben aparecer únicamente los puntos, escritos como (x, y) o como $\{x, y\}$ y separados por espacio en blanco:

```
{1.5,-0.932194} {1.9,-0.669879} {2.3,-0.472617}  
{2.7,-0.291887} {3.1,-0.117535} {3.5,0.0453714} ...
```

Para este ejemplo, guardaremos la lista de puntos en el archivo `bessel.txt` y utilizaremos el comando `\bfuncion` para acceder a los datos; este comando lo definimos así: `\readdata{\bfuncion}{bessel.txt}`.



La anterior gráfica se puede entonces obtener de dos maneras diferentes, ya sea con:

```
\begin{center}
\psset{xunit=0.3cm,yunit=4cm}
\begin{pspicture}(-4,-1)(38,0.8)
\psaxes[dx=5,Dx=5,dy=0.25,Dy=0.25]{->}(0,0)(-1,-1)(37,0.75)
\dataplot[plotstyle=curve,linewidth=1.1pt]{\afuncion}
\end{pspicture}
\end{center}
```

o a partir de:

```
\begin{center}
\psset{xunit=0.3cm,yunit=4cm}
\begin{pspicture}(-4,-1)(38,0.8)
\psaxes[dx=5,Dx=5,dy=0.25,Dy=0.25]{->}(0,0)(-1,-1)(37,0.75)
\dataplot[plotstyle=curve,linewidth=1.1pt]{\bfuncion}
\end{pspicture}
\end{center}
```

En el primer caso, hemos usado `\afuncion` (los datos son almacenados y leídos directamente en el documento fuente L^AT_EX); en el segundo caso, hemos usado `\bfuncion` (los datos son almacenados y leídos en el documento externo `bessel.txt`).

11.22. Trazado de funciones a partir de su ecuación

PSTricks puede graficar funciones de variable real con ecuación dada; la instrucción

```
\psplot[opciones]{x1}{x2}{función}
```

dibuja la *función* dada entre los valores x_1 y x_2 del dominio. Las *opciones* de trazado son las usuales (Tabla 11.3), junto con los parámetros *plotstyle* (sección 11.21) y *plotpoints*. Este último parámetro se refiere al número de puntos utilizado en el intervalo $[x_1, x_2]$ para calcular los valores de la función; por defecto *plotpoints*=50. Para obtener curvas más “suaves” se recomienda usar *plotstyle=curve* y valores grandes de *plotpoints* (por ejemplo, 100, 300 ó 500).

La fórmula que define la *función* se debe escribir en la sintaxis del lenguaje PostScript, el cual utiliza la llamada notación sufija. Dicha notación tiene la peculiaridad de que no requiere paréntesis; de hecho, el uso de paréntesis conduce a mensajes de error y la instrucción no se procesa. En la Tabla 11.13 aparece la sintaxis de las funciones pre-definidas en PostScript.

Sintaxis	Significado
<i>a neg</i>	$-a$
<i>a b add</i>	$a + b$
<i>a b sub</i>	$a - b$
<i>a b mul</i>	$a \cdot b$
<i>a b div</i>	a/b
<i>a b exp</i>	a^b
<i>a sqrt</i>	\sqrt{a}
<i>a log</i>	$\log_{10}(a)$
<i>a ln</i>	$\ln(a)$
<i>a sin</i>	$\operatorname{sen}(a)$
<i>a cos</i>	$\cos(a)$

TABLA 11.13. Sintaxis PostScript de las funciones pre-definidas.

Sólo un poco de práctica se requiere para que el usuario se familiarice con la notación sufija de PostScript; en la página siguiente presentamos algunos ejemplos.

Ejemplo

Función	Sintaxis PostScript
$x^3 + x^2$	<code>x 3 exp x 2 exp add</code>
$\frac{x - 1}{x + 1}$	<code>x 1 sub x 1 add div</code>
$\frac{1}{x\sqrt{x^2 - 1}}$	<code>1 x x 2 exp 1 sub sqrt mul div</code>
x^{2x+1}	<code>x 2 x 1 add exp mul</code>
$-\tan 3x = -\frac{\sin 3x}{\cos 3x}$	<code>3 x mul sin 3 x mul cos div neg</code>
$\log_{10} \frac{2x + 1}{2x - 1}$	<code>2 x mul 1 add 2 x mul 1 sub div log</code>

Si α , f , g son expresiones PostScript válidas de funciones, podemos usar las siguientes simplificaciones:

Expresión	Significa lo mismo que
$\alpha \text{ dup}$	$\alpha \alpha$
$\alpha \text{ dup } f \text{ exch } g$	$\alpha f \alpha g$

El término **exch** proviene de la palabra “exchange”. Estas simplificaciones son convenientes cuando la expresión α se repite, en cuyo caso, no es necesario escribirla dos veces.

Ejemplo

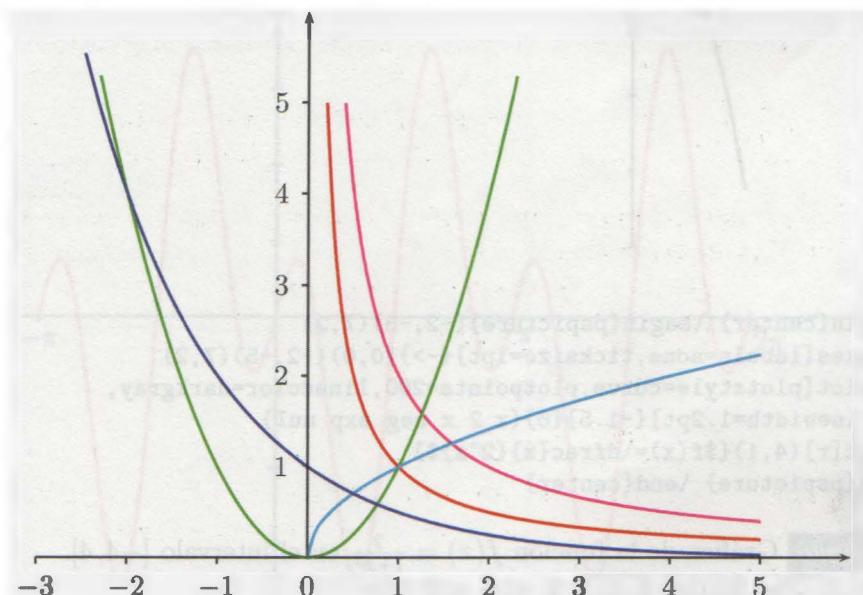
Función	Sintaxis PostScript
$(x^3 + 1) \ln(x^3 + 1)$	<code>x 3 exp 1 add dup ln mul</code>
$\log_{10}(x^2 + 1) - \ln(x^2 + 1)$	<code>x 2 exp 1 add dup log exch ln sub</code>

Al usar `\psplot` es necesario tener presente las siguientes observaciones adicionales:

- La variable dependiente debe ser siempre x .
- Las funciones seno y coseno asumen, por defecto, argumentos en grados sexagesimales; el usuario debe, por consiguiente, tener cautela al

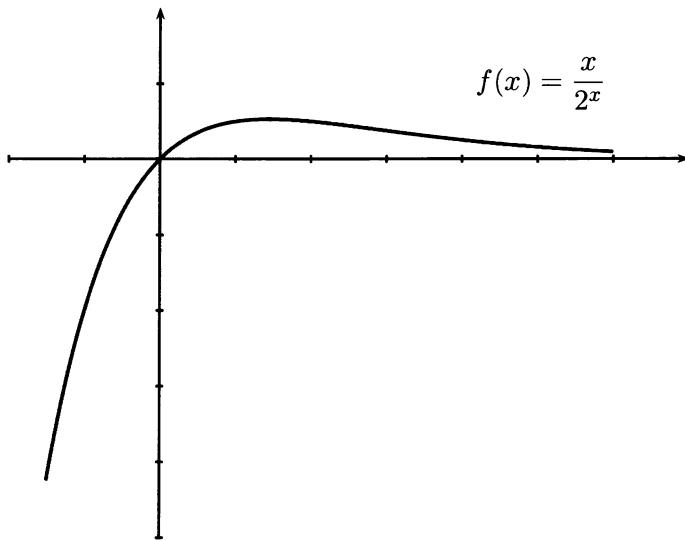
- combinar funciones trigonométricas con otras funciones de variable real.
- Ni **LATeX**, ni PostScript ni **PSTricks** están diseñados para cálculos científicos, por lo que la utilidad de `\psplot` está limitada a funciones relativamente simples (como las exhibidas en los ejemplos que siguen); para funciones extremadamente complicadas, la gráfica obtenida puede ser incompleta o incorrecta.

Ejemplo Gráfica de las funciones $y = 1/x$ (rojo), $y = 2/x$ (púrpura), $y = x^2$ (verde), $y = \sqrt{x}$ (azul claro), $y = (\frac{1}{2})^x$ (azul oscuro).



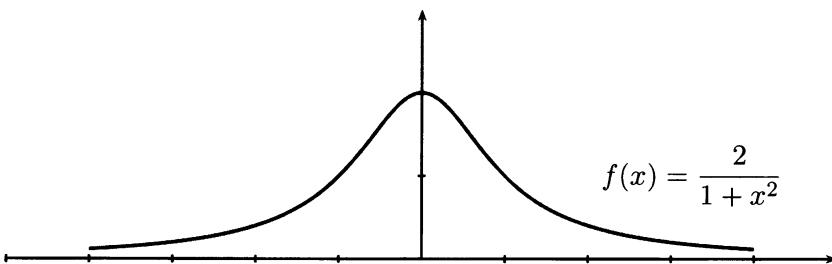
```
\begin{center}
\psset{unit=1.2cm}
\begin{pspicture}(-3,0)(6,6)
\psaxes[ticksize=2pt,tickstyle=bottom]{->}(0,0)(-3,0)(6,6)
\psplot[plotstyle=curve,linecolor=magenta,linewidth=1.1pt]%
{0.4}{5}{2 x div}
\psplot[plotstyle=curve,linecolor=red,linewidth=1.1pt]%
{0.2}{5}{1 x div}
\psplot[plotstyle=curve,linecolor=green,linewidth=1.1pt]%
{-2.3}{2.3}{x 2 exp}
\psplot[plotstyle=curve,linecolor=cyan,linewidth=1.1pt]%
{0}{5}{x sqrt}
\psplot[plotstyle=curve,linecolor=blue,linewidth=1.1pt]%
{-2.47}{5}{0.5 x exp}
\end{pspicture} \end{center}
```

Ejemplo Gráfica de la función $f(x) = x2^{-x}$ en el intervalo $[-1.5, 6]$.
 Sintaxis PostScript: `x 2 x neg exp mul`.



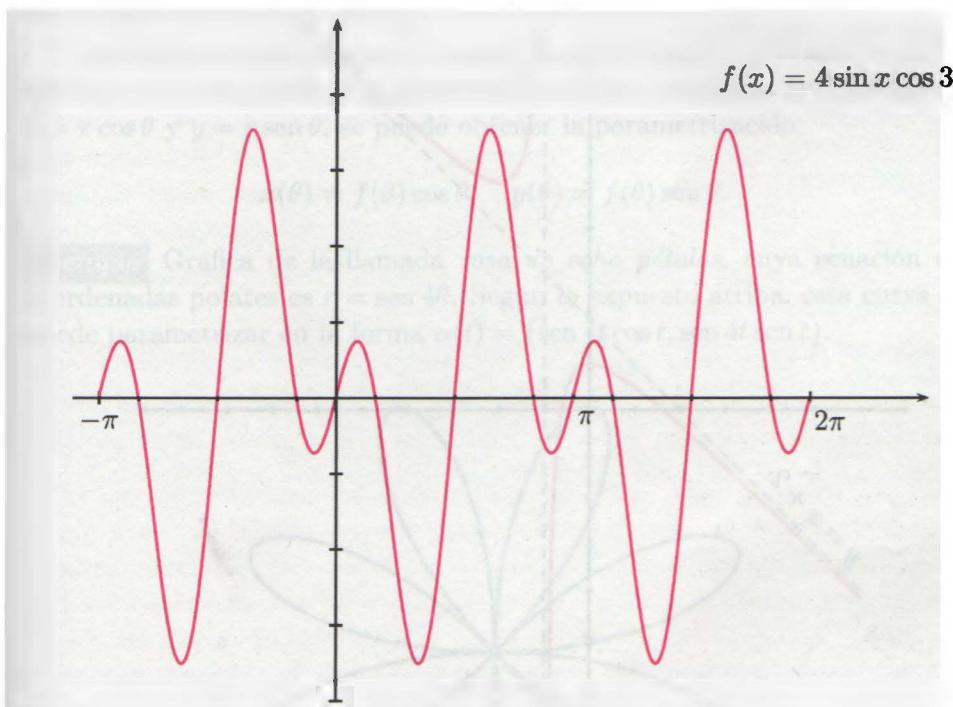
```
\begin{center} \begin{pspicture}(-2,-5)(7,2)
\psaxes[labels=none,ticksize=1pt]{->}(0,0)(-2,-5)(7,2)
\psplot[plotstyle=curve,plotpoints=200,linecolor=darkgray,
linewidth=1.2pt]{-1.5}{6}{x 2 x neg exp mul}
\uput[r](4,1){$f(x)=\frac{x}{2^x}$}
\end{pspicture} \end{center}
```

Ejemplo Gráfica de la función $f(x) = \frac{2}{1+x^2}$ en el intervalo $[-4, 4]$.
 Sintaxis PostScript: `2 1 x 2 exp add div`.



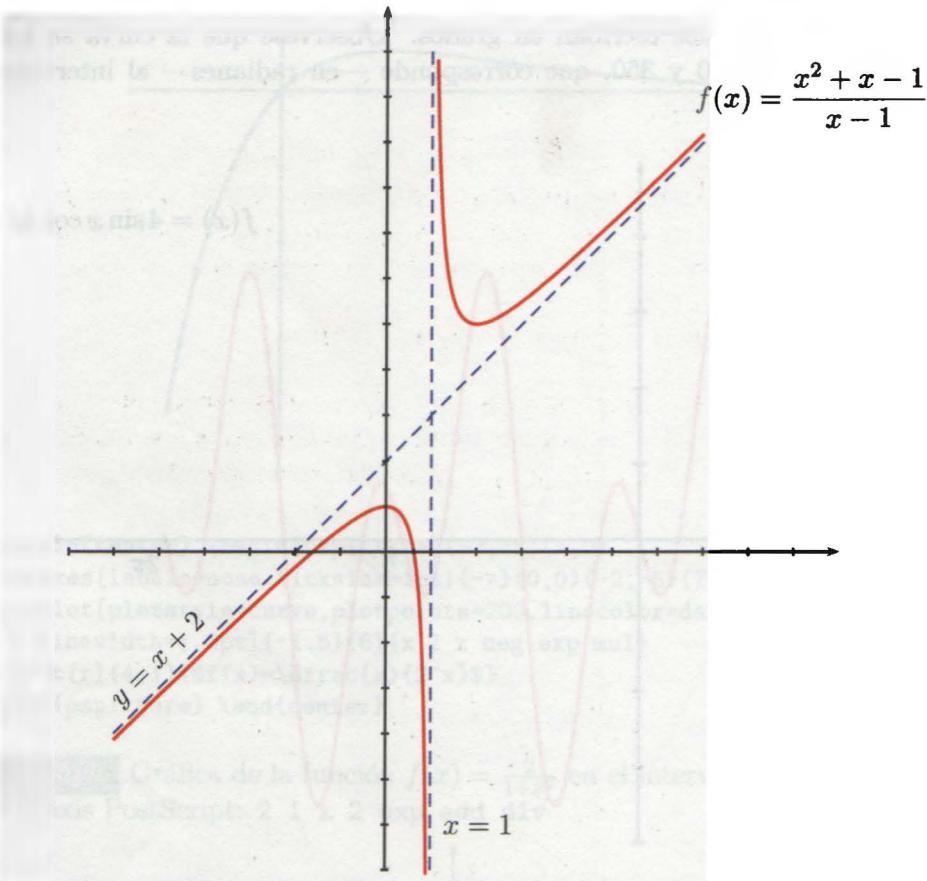
```
\psset{unit=1.1cm} \begin{pspicture}(-5,0)(5,3)
\psaxes[labels=none,ticksize=1pt]{->}(0,0)(-5,0)(5,3)
\psplot[plotstyle=curve,plotpoints=300,linecolor=darkgray,
linewidth=1.2pt]{-4}{4}{2 1 x 2 exp add div}
\uput[r](2,1){$f(x)=\frac{2}{1+x^2}$}
\end{pspicture}
```

Ejemplo Gráfica de la función $f(x) = 4 \sin x \cos 3x$, cuya sintaxis PostScript es `4 x sin mul 3 x mul cos mul`. Como se señaló arriba, las funciones trigonométricas seno y coseno asumen, por defecto, argumentos en grados sexagesimales; por esta razón, hemos tomado como `xunit` el valor $0.01745 = \pi/180$. De esta manera, el eje horizontal representa radianes aunque los valores se escriban en grados. Obsérvese que la curva se ha trazado entre -180 y 360 , que corresponde —en radianes— al intervalo $[-\pi, 2\pi]$.



```
\begin{center}
\psset{xunit=0.01745cm,yunit=1cm}
\begin{pspicture}(-200,-4)(450,5)
\psaxes[labels=none,ticks=2pt,dx=90]{>}(0,0)(-200,-4)(450,5)
\psplot[plotstyle=curve,plotpoints=500,linecolor=magenta,
        linewidth=1.1pt]{-180}{360}{4 x sin mul 3 x mul cos mul}
\put[d]{-180,0}{$-\pi$}
\put[d]{190,0}{$\pi$}
\put[d]{375,0}{$2\pi$}
\put[r]{280,4.2}{$f(x)=4\sin x \cos 3x$}
\end{pspicture}
\end{center}
```

Ejemplo Gráfica de la función $y = \frac{x^2+x-1}{x-1}$, cuya sintaxis PostScript es `x 2 exp x add 1 sub x 1 sub div`. Debido al comportamiento asintótico de la función, se han trazado las dos ramas por separado: la de la izquierda en el intervalo $[-6, 0.9]$ y la de la derecha en el intervalo $[1.13, 7]$.



```
\begin{center}\psset{unit=0.6cm}
\begin{pspicture}(-7,-7)(11,12)
\psaxes[ticksize=1pt,labels=none]{->}(0,0)(-7,-7)(10,12)
\psline[linecolor=blue,linestyle=dashed](1,-7)(1,11)
\psplot[linecolor=blue,linestyle=dashed]{-6}{7}{x 2 add}
\psplot[plotstyle=curve,plotpoints=300,linecolor=red,
linewidth=1.2pt]{-6}{0.9}{x 2 exp x add 1 sub x 1 sub div}
\psplot[plotstyle=curve,plotpoints=300,linecolor=red,
linewidth=1.2pt]{1.13}{7}{x 2 exp x add 1 sub x 1 sub div}
\rput{45}(-5,-2.4){$y=x+2$}
\uput[r](1,-6){$x=1$}
\uput[r](6.5,10){$f(x)=\frac{x^2+x-1}{x-1}$}
\end{pspicture}\end{center}
```

11.23. Curvas paramétricas

Una curva paramétricamente definida por $\alpha(t) = (x(t), y(t))$ se puede trazar, entre los valores t_0 y t_1 del parámetro t , por medio de

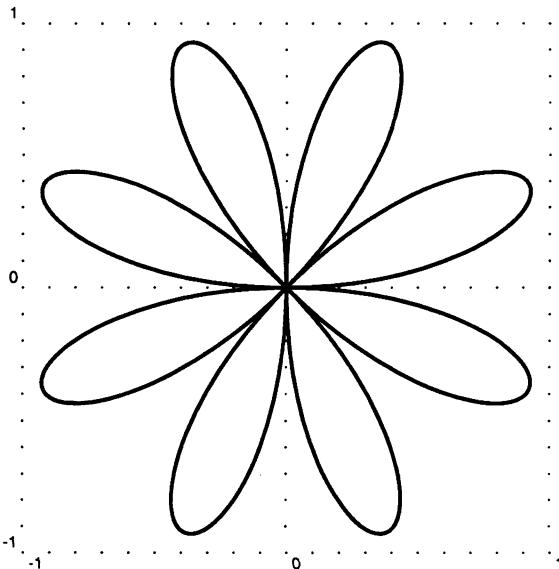
 $\text{\textbackslash parametricplot}[\text{opciones}]\{t_0\}\{t_1\}\{x(t)\ y(t)\}$

Tanto $x(t)$ como $y(t)$ deben estar escritas en sintaxis PostScript (véase la sección 11.22) y el parámetro t es obligatorio (es decir, para la parametrización no se puede usar ninguna otra letra diferente de t).

$\text{\textbackslash parametricplot}$ es muy útil para dibujar curvas de la forma $r = f(\theta)$, definidas con coordenadas polares (r, θ) ya que, utilizando las igualdades $x = r \cos \theta$ y $y = r \sen \theta$, se puede obtener la parametrización

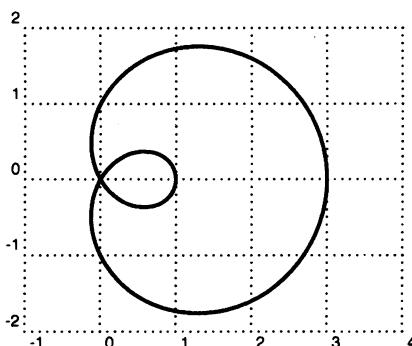
$$x(\theta) = f(\theta) \cos \theta, \quad y(\theta) = f(\theta) \sen \theta.$$

Ejemplo Gráfica de la llamada *rosa de ocho pétalos*, cuya ecuación en coordenadas polares es $r = \sen 4\theta$. Según lo expuesto arriba, esta curva se puede parametrizar en la forma $\alpha(t) = (\sen 4t \cos t, \sen 4t \sen t)$.



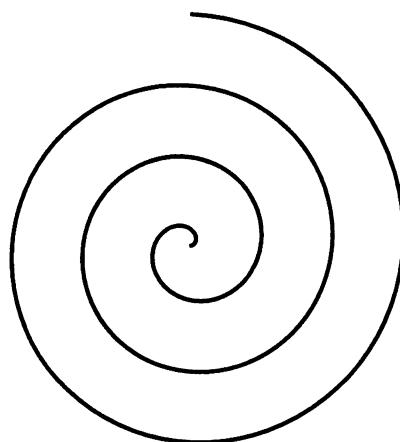
```
\begin{center}
\psset{unit=3.5cm}
\begin{pspicture}(-1,-1)(1,1)\grilla
\parametricplot[plotpoints=300,linewidth=1.1pt]{0}{360}%
{4 t mul sin t cos mul 4 t mul sin t sin mul}
\end{pspicture}
\end{center}
```

Ejemplo Gráfica del llamado *caracol de Pascal*, cuya ecuación polar es $r = 1 + 2 \cos \theta$. Parametrización: $\alpha(t) = ((1+2 \cos t) \cos t, (1+2 \cos t) \sin t)$.



```
\begin{center}
\begin{pspicture}(-1,-2)(4,2)
\grilla
\parametricplot[plotpoints=300,linewidth=1.1pt]{0}{360}%
{1 2 t cos mul add t cos mul 1 2 t cos mul add t sin mul}
\end{pspicture}\end{center}
```

Ejemplo Gráfica de la *espiral de Arquímedes*, $r = \theta$, trazada con la parametrización $\alpha(t) = (t \cos t, t \sin t)$. Puesto que, en presencia de las funciones trigonométricas seno y coseno, **PSTricks** utiliza grados sexagesimales, debemos convertir el factor t a radianes multiplicando por $\pi/180 = 0.01745$.



```
\begin{center}\psset{unit=0.15cm}
\begin{pspicture}(-20,-18)(20,22)
\parametricplot[plotpoints=500,linewidth=1.2pt]%
{0}{1170}{0.01745 t mul t cos mul 0.01745 t mul t sin mul}
\end{pspicture}\end{center}
```

11.24. Relleno y sombreado de regiones. Parte II.

En esta sección describimos las herramientas que **PSTricks** posee para el relleno o sombreado de regiones limitadas por curvas. La instrucción

```
\pscustom[opciones iniciales]{curvas \fill[opciones de relleno]}
```

rellena o sombra la región delimitada por las *curvas* dadas (rectas, curvas, polígonos, etc) utilizando las *opciones de relleno* que el usuario desee. Como *opciones iniciales* se pueden usar las usuales opciones de trazoado, aunque muchas de ellas —como **shadow**, **doubleline**, **showpoints**— serán ignoradas. En la parte '*curvas*' puede aparecer, en realidad, casi cualquier comando de **PSTricks** pero es aconsejable usar solamente los más naturales, como **\psline**, **\pscurve**, **\psecurve**, etc. El uso inadecuado de **\pscustom** puede conducir a graves mensajes de error PostScript o puede agotar rápidamente los recursos computacionales disponibles.

Para comprender el funcionamiento de **\pscustom** hay que familiarizarse con la noción de *punto actual* (en inglés, *current point*). Al inicio no hay punto actual pero al trazar una curva, el último punto trazado se convierte en el punto actual. Dicho punto actual se une al primer punto de la siguiente curva y el último punto de esta curva se convierte en el nuevo punto actual, y así sucesivamente. Esto implica, además, que el orden en el que se presentan los puntos (con **\psline**, **\pscurve**, etc) es muy importante. La región cerrada que se forma de esta manera se rellena o sombra según las opciones dadas a **\fill[...]**, que pueden ser cualquiera de las opciones mostradas en las Tablas 11.5 y 11.7; se recomienda usar la opción de relleno **fillstyle=...** en el parámetro **\fill[...]** y no en las *opciones iniciales* de **\pscustom**. Para controlar la manera cómo el punto actual se une con el primer punto de la siguiente curva tenemos a nuestra disposición el parámetro **liftpen**, el cual puede tomar los valores 0, 1 y 2 (véase la Tabla 11.14).

liftpen=0	El punto actual se une con el primer punto de la siguiente curva formando una curva continua. Es el valor asumido por defecto.
liftpen=1	El punto actual se une con el primer punto de la siguiente curva por medio de un segmento de recta.
liftpen=2	El punto actual no se utiliza; esto quiere decir que las dos curvas consecutivas no se unen y, por lo tanto, la región entre ellas no se rellena ni se sombra en su totalidad.

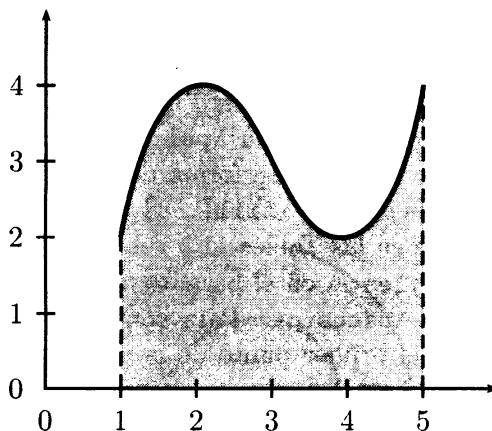
TABLA 11.14. Valores posibles del parámetro **liftpen** de **\pscustom**.

Frecuentemente queremos que las instrucciones de relleno o sombreado no afecten los bordes de la región, por lo que debemos distinguir entre la región delimitada por las curvas dadas y las curvas mismas. En tales situaciones se recomienda emplear **\pscustom** en la siguiente forma:

```
\pscustom[linestyle=none, ... ]{ ... \fill[...]}
```

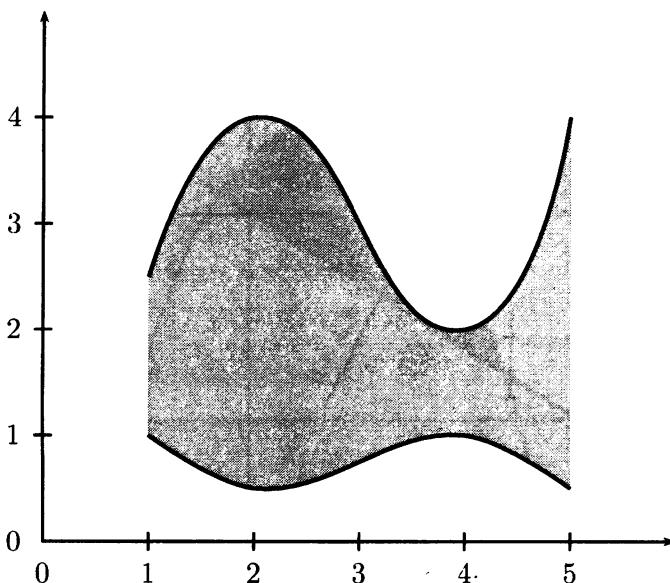
con lo cual el relleno o sombreado afecta la región deseada pero no las curvas que lo delimitan (las curvas propiamente dichas se pueden trazar independientemente, en la forma usual). **PSTricks** posee los mecanismos (con los comandos **\gsave** y **\grestore**) para trazar los bordes de la región simultáneamente con la instrucción de relleno **\pscustom** pero, en nuestra opinión, ese procedimiento es más complicado que el aquí recomendamos.

Ejemplo Las rectas verticales que delimitan la región sombreada están trazadas explícitamente con **\psline** (con la opción **linestyle=dashed**), mientras que la curva superior se trazó con **\pscurve** y grosor 1.5pt. El comando de sombreado **\pscustom** aparece con anterioridad, con la opción **linestyle=none**, y con los bordes de la región presentados en el orden: recta vertical izquierda–curva superior–recta vertical derecha.



```
\begin{center}
\begin{pspicture}(6,5)
\pscustom[linestyle=none]{\psline(1,0)(1,2)
\pscurve[liftpen=1](1,2)(2,4)(4,2)(5,4)
\psline(5,4)(5,0)
\fill[fillstyle=solid, fillcolor=lightgray]}
\psaxes{->}(6,5)
\pscurve[linewidth=1.5pt](1,2)(2,4)(4,2)(5,4)
\psline[linestyle=dashed](1,0)(1,2)
\psline[linestyle=dashed](5,4)(5,0)
\end{pspicture}
\end{center}
```

Ejemplo Al igual que en el ejemplo anterior, el comando `\pscustom` para sombrear aparece antes que las instrucciones `\pscurve` que trazan las curvas propiamente dichas. Obsérvese el orden en que se presentan los bordes de la región sombreada: primero la vertical izquierda de (1,1) a (1,2.5), luego la curva superior (con sus puntos listados de izquierda a derecha), luego la vertical derecha de (5,4) a (5,0.5) y, finalmente, la curva inferior (con sus puntos listados de derecha a izquierda).

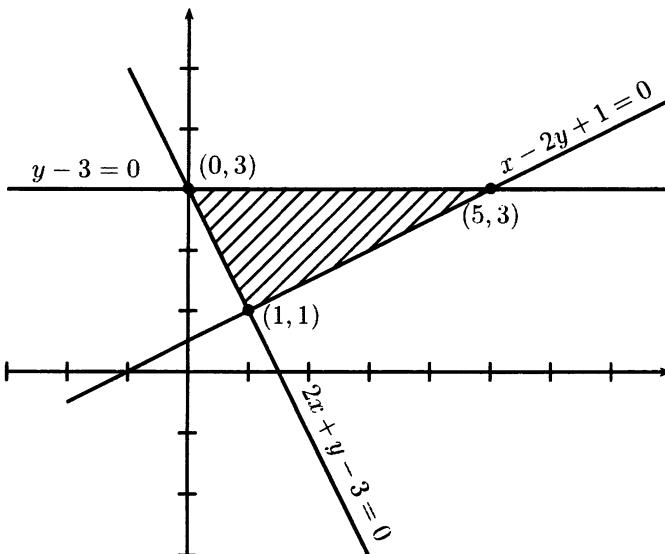


```

\begin{center}
\psset{unit=1.4cm}
\begin{pspicture}(6,5)
\psaxes{>}(6,5)
\pscustom[linestyle=none]{%
\psline(1,1)(1,2.5)
\pscurve[liftpen=1](1,2.5)(2,4)(4,2)(5,4)
\psline(5,4)(5,0.5)
\pscurve[liftpen=1](5,0.5)(4,1)(2,0.5)(1,1)
\fill[fillstyle=solid, fillcolor=lightgray]
\pscurve[linewidth=1.2pt](1,2.5)(2,4)(4,2)(5,4)% curva superior
\pscurve[linewidth=1.2pt](5,0.5)(4,1)(2,0.5)(1,1)% curva inferior
\end{pspicture}
\end{center}

```

Ejemplo Aquí se ha usado el parámetro `fillstyle=hlines` (Tabla 11.5) para llenar la región triangular definida por medio de `\pspolygon`. En este ejemplo se ilustra el uso de `\pscustom` pero cabe anotar que la región se puede llenar con `\pspolygon[fillstyle=hlines](0,3)(5,3)(1,1)`.

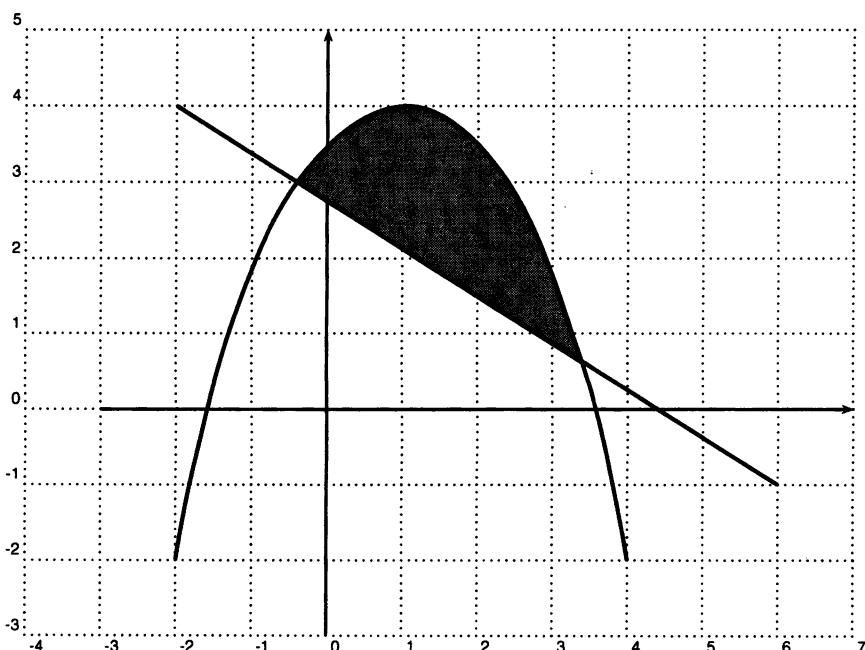


```

\begin{center}
\psset{unit=0.8cm}
\begin{pspicture}(-3,-3)(8,6)
\psaxes[labels=none]{->}(0,0)(-3,-3)(8,6)
\rput{28}(6.4,4){\small $x-2y+1=0$}
\rput{-63}(2.6,-1.5){\small $2x+y-3=0$}
\rput[1](-2.6,3.3){\small $y-3=0$}
\rput(0,3){$\bullet$}
\rput(1,1){$\bullet$}
\rput(5,3){$\bullet$}
\uput[d](5,3){\small $(5,3)$}
\uput[r](1,0.9){\small $(1,1)$}
\uput[ur](0,3){\small $(0,3)$}
\psline[linewidth=1pt](-1,5)(3,-3) % recta $2x+y-3=0$
\psline[linewidth=1pt](-3,3)(8,3) % recta $y-3=0$
\psline[linewidth=1pt](-2,-0.5)(8,4.5) % recta $x-2y+1=0$
\pscustom{\pspolygon(0,3)(5,3)(1,1)}
\fill[fillstyle=hlines]
\end{pspicture}
\end{center}

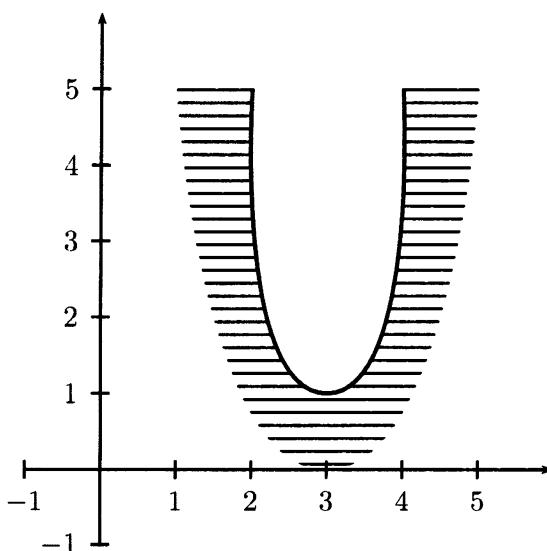
```

Ejemplo La curva (parábola) se ha trazado con `\pscurve` con los seis puntos dados y grosor de 1.2pt. En el comando de sombreado `\pscUSTOM` se utilizaron los mismos seis puntos pero con `\psecurve`, para ignorar los puntos extremos $(-2, -2)$ y $(4, -2)$. Si se usara aquí `\pscurve` en lugar de `\psecurve`, con los cuatro puntos interiores $(-0.4, 3)$, $(1, 4)$, $(2, 3.5)$ y $(3.4, 0.6)$, la interpolación realizada sería ligeramente diferente y, por lo tanto, la región sombreada no sería exactamente la región deseada (cosa que puede comprobar el lector por sí mismo).



```
\begin{center}
\begin{pspicture}(-4,-3)(7,5)\grilla
\psaxes[labels=none,ticks=none]{->}(0,0)(-3,-3)(7,5)
\pscUSTOM{%
\psecurve(-2,-2)(-0.4,3)(1,4)(2,3.5)(3.4,0.6)(4,-2)
\psline(3.4,0.6)(-0.4,3)
\fill[fillstyle=solid,fillcolor=gray]}
\pscurve[linewidth=1.2pt](-2,-2)(-0.4,3)(1,4)(2,3.5)(3.4,0.6)(4,-2)
\psline[linewidth=1.2pt](-2,4)(6,-1)
\end{pspicture}
\end{center}
```

Ejemplo Para el relleno se usó aquí la opción `fillstyle=hlines` con ángulo de rayado (`hatchangle`) igual a 0. Los bordes de la región rayada se presentaron en el siguiente orden: primero el segmento horizontal de (2,5) a (1,5), luego la curva exterior que une los puntos (1,5), (3,0) y (5,5); a continuación el segmento horizontal de (5,5) a (4,5) y, finalmente, la curva interior que une los puntos (4,5), (3,1) y (2,5). Los segmentos horizontales en la parte superior y la curva exterior son “invisibles” debido a la opción `linestyle=none` de `\pscustom`. La curva interior se trazó explícitamente por medio de `\pscurve`, con grosor 1.2pt.



```
\begin{center}
\begin{pspicture}(-1,-1)(6,6)
\psaxes{->}(0,0)(-1,-1)(6,6)
\pscustom[linestyle=none]{%
\psline(2,5)(1,5)
\pscurve[liftpen=1](1,5)(3,0)(5,5)
\psline(5,5)(4,5)
\pscurve[liftpen=1](4,5)(3,1)(2,5)
\fill[fillstyle=hlines,hatchangle=0]}
\pscurve[linewidth=1.2pt](4,5)(3,1)(2,5)
\end{pspicture}
\end{center}
```

11.25. Otras coordenadas

La instrucción `\SpecialCoor` permite utilizar otras coordenadas, dividiendo la circunferencia en el número de grados deseado. Por ejemplo, al declarar

```
\SpecialCoor
\degrees[360]
```

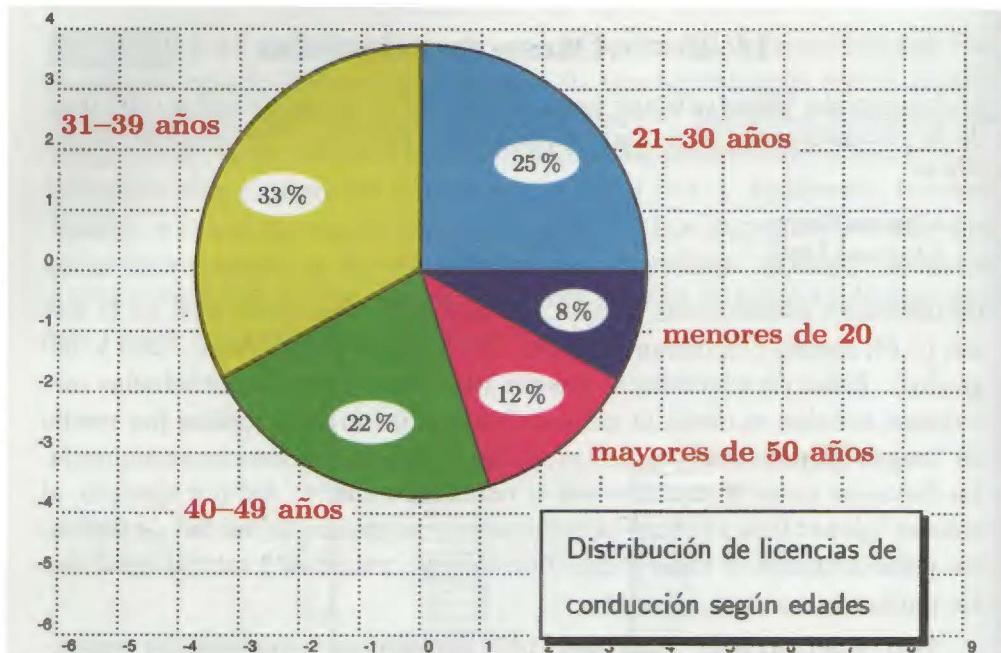
establecemos coordenadas polares, las cuales se deben presentar en la forma $(r; \theta)$, siendo r la distancia polar y θ el ángulo polar (entre -360 y 360 grados). Estas coordenadas se pueden usar junto con las coordenadas cartesianas usuales; es decir, la grilla definida al inicio de la gráfica por medio de `\begin{pspicture}(x_0,y_0)(x_1,y_1)` se utiliza en la forma acostumbrada. La distancia polar se controla con el parámetro `runit`. Así por ejemplo, al colocar `\psset{runit=2cm}` al comienzo de la gráfica, la unidad de distancia polar adquiere el valor 2 cm. Por defecto, `runit` es 1 cm (al igual que los parámetros `xunit` y `yunit`).

Con `\SpecialCoor \degrees[100]` establecemos coordenadas centesimales. Como en el caso de las coordenadas polares, las coordenadas centesimales se deben presentar en la forma $(r; \theta)$, siendo r la distancia al origen y θ el ángulo (entre -100 y 100 grados).

En una misma gráfica se pueden utilizar varios tipos de coordenadas con sendas declaraciones `\SpecialCoor` y `\degrees[...]`. También es útil saber que la instrucción `\NormalCoor` deshabilita la acción de `\SpecialCoor` y que el comando `\radians` es una abreviación de `\degrees[6.28319]`.

Ejemplo Para hacer la “torta de porcentajes” que aparece en la página siguiente hemos utilizado coordenadas centesimales. Obsérvese que:

- Las unidades cartesianas `xunit` y `yunit` tienen un valor de 0.8 cm, mientras que la unidad `runit` se establece en 3 cm.
- Los sectores circulares definidos con `\pswedge` tienen su centro en el origen de coordenadas cartesianas $(0,0)$ pero el radio y los ángulos son coordenadas centesimales (ya que los comandos `\pswedge` aparecen debajo de la instrucción `\SpecialCoor`). El radio de cada sector circular tiene longitud 1 , es decir, 3 cm.
- Para colocar la tabla con la leyenda “Distribución de licencias ...” se utilizaron coordenadas cartesianas: la tabla aparece en la posición $(2, -5)$, con el parámetro `[1]`. Para todos los demás comandos de posición, `\rput` y `\rput*`, se utilizaron coordenadas centesimales $(r; \theta)$.



```
\begin{center}
\psset{xunit=0.8cm,yunit=0.8cm}
\psset{runit=3cm}
\begin{pspicture}(-6,-6)(9,4)\grilla
\SpecialCoor
\degrees[100]
\pswedge[fillstyle=solid,fillcolor=cyan](0,0){1}{0}{25} %25%
\pswedge[fillstyle=solid,fillcolor=yellow](0,0){1}{25}{58} %33%
\pswedge[fillstyle=solid,fillcolor=green](0,0){1}{58}{80} %22%
\pswedge[fillstyle=solid,fillcolor=magenta](0,0){1}{80}{92} %12%
\pswedge[fillstyle=solid,fillcolor=blue](0,0){1}{92}{100} %8%
\rput(0.7;96){\footnotesize \psovalbox*{8\%}}
\rput(0.7;86){\footnotesize \psovalbox*{12\%}}
\rput(0.7;70){\footnotesize \psovalbox*{22\%}}
\rput(0.7;42){\footnotesize \psovalbox*{33\%}}
\rput(0.7;12){\footnotesize \psovalbox*{25\%}}
\rput*[1](1.1;9){\bf \red 21--30 años}
\rput*[r](1.1;40){\bf \red 31--39 años}
\rput*[r](1.1;70){\bf \red 40--49 años}
\rput*[1](1.1;87){\bf \red mayores de 50 años}
\rput*[1](1.1;96){\bf \red menores de 20 años}
\rput*[1](2,-5){\sf \psframebox[shadow=true]%
{\begin{tabular}{l} Distribución de licencias de\\
conducción según edades \end{tabular}}}
\end{pspicture}
\end{center}
```

11.26. Grafos: nodos y conectores de nodos

PSTricks tiene una variada gama de herramientas para crear y unir nodos con conectores (o arcos) que pueden o no tener etiquetas. Para evitar confusiones con los numerosos comandos de esta sección es necesario distinguir tres clases de comandos:

1. *Comandos para crear nodos.* Permiten definir el nombre, la forma y el contenido de los nodos (véase la sección 11.26.1).
2. *Comandos para conectar nodos.* Permiten conectar —de muchas formas— dos nodos que tienen nombres previamente asignados (véase la sección 11.26.2).
3. *Comandos para etiquetar los conectores de nodos.* Permiten colocar etiquetas a los diferentes conectores (véase la sección 11.26.3) .

11.26.1. Tipos de nodos

Los nodos más sencillos se obtienen con:

`\pnode(x,y){nombre}`. Coloca en (x,y) un nodo que no ocupa espacio y le asigna un *nombre*.

`\dotnode[opciones](x,y){nombre}`. Coloca en (x,y) un nodo en forma de punto y le asigna un *nombre*. Se pueden utilizar diferentes tipos de puntos con el parámetro `dotstyle` (Tabla 11.9, página 327).

`\cnode[opciones](x,y){radio}{nombre}`. Coloca en el punto (x,y) un nodo en forma circular, con *radio* dado, y le asigna un *nombre*. Se pueden usar las diferentes *opciones* de trazado de líneas (Tabla 11.3).

`\Cnode[opciones](x,y){nombre}`. Similar a `\cnode` pero el radio se establece de antemano con el parámetro `radius`, utilizando la instrucción `\psset{radius=...}`. Es útil cuando hay varios círculos del mismo radio en la gráfica.

`\rnode{nombre}{texto}`. Asigna un *nombre* al nodo rectangular con bordes invisibles que contiene el *texto* especificado. El *texto* que aparece escrito en el nodo puede ser texto normal, fórmulas o incluso tablas.

El *nombre* asignado a un nodo debe tener solamente letras y dígitos y debe comenzar con una letra; el uso de nombres indebidos da lugar a mensajes de error PostScript. Obsérvese que con `\rnode` no se indica el punto de

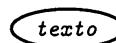
ubicación (x, y) ; para colocar este tipo de nodo en una gráfica particular se debe usar el comando de posición `\rput` (sección 11.14).

Los siguientes comandos se asemejan a `\rnodenode` pero le dan formas geométricas visibles a los nodos creados.

`\circlenode[opc.]{nombre}{texto}` Nodo circular:



`\ovalnode[opc.]{nombre}{texto}` Nodo ovalado:



`\trinode[opc.]{nombre}{texto}` Nodo triangular:



`\dianode[opc.]{nombre}{texto}` Nodo en rombo:



No existe `\framenode` ya que para nodos rectangulares se espera que el usuario utilice `\rnodenode{\psframebox{...}}`. En realidad, con `\rnodenode` se puede usar cualquiera de los comandos para enmarcar texto presentados en la sección 11.13.

11.26.2. Conectores de nodos

Todos los conectores de nodos (Tabla 11.15), a excepción de `\nccircle`, tienen la siguiente sintaxis:

`\conector[opciones]{flechas}{nombre nodo 1}{nombre nodo 2}`

Los nombres de los nodos se deben asignar previamente con los comandos de 11.26.1. Se pueden usar las usuales opciones de trazado (Tabla 11.3) y las opciones específicas para conectores de nodos (Tabla 11.16). **PSTricks** asume que el primer nodo es siempre el que aparece más a la izquierda.

El único conector de nodos que se aparta de la sintaxis anterior es:

`\nccircle[opciones]{flechas}{nombre nodo}{radio}`

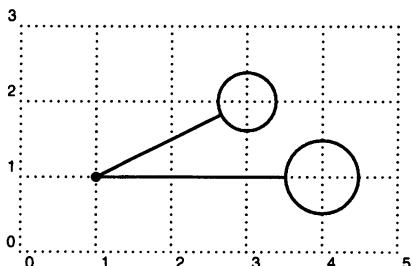
el cual conecta un nodo consigo mismo por medio de un arco circular de radio dado. El círculo se traza desde el ángulo `angleA`, en la dirección antihoraria. La separación entre el arco y el borde del nodo se puede modificar con el parámetro `nodedsepA` (Tabla 11.16).

Examinando en detalle los siguientes ejemplos, el lector se puede familiarizar con los numerosos comandos y opciones disponibles. En los ejemplos, al primer nodo (el de la izquierda) se le ha dado el nombre A y al segundo nodo (el de la derecha) B.

\ncline	Une los nodos con un segmento de recta.
\ncarc	Une los nodos con un arco. Utiliza el parámetro <code>arcangle</code> .
\nccurve	Une los nodos con una curva de Bézier (véase la sección 7.6.9).
\ncdiag	Traza un brazo desde cada nodo y los une con un segmento diagonal. Utiliza los parámetros <code>armA</code> , <code>armB</code> , <code>angleA</code> y <code>angleB</code> . Para vértices redondeados se utiliza el parámetro <code>linearc</code> .
\ncdiagg	Similar a \ncdiag pero sin el brazo para el segundo nodo.
\ncbar	Une los nodos con segmentos que se intersectan en ángulos rectos, extendiendo los brazos si es necesario. El ángulo de salida del primer nodo y el ángulo de llegada al segundo nodo coinciden y se controla con el parámetro <code>angleA</code> .
\ncangle	Traza un brazo desde cada nodo y los une con segmentos que se intersectan en ángulos rectos. Utiliza los parámetros <code>armA</code> , <code>armB</code> , <code>angleA</code> y <code>angleB</code> . Para vértices redondeados se utiliza el parámetro <code>linearc</code> .
\ncangles	Similar a \ncangle pero utiliza (por lo general) un segmento adicional para unir los brazos.
\ncloop	Diseñado para bucles (en inglés, <i>loops</i>) aunque también sirve para conectar dos nodos diferentes. En este último caso, se comporta como \ncangles pero con dos segmentos adicionales. El tamaño del brazo vertical del bucle se controla con el parámetro <code>loopsize</code> que, por defecto, es 1 cm.

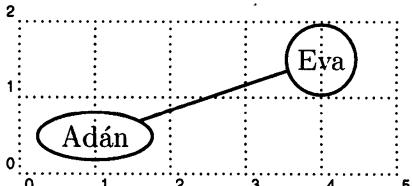
TABLA 11.15. Conectores de nodos.

Ejemplo El nodo con nombre A (creado con \dotnode) se conecta con los nodos B y C (creados con \cnode) por medio del conector \ncline.



```
\begin{pspicture}(5,3)\grilla
\dotnode(1,1){A}
\cnode(3,2){0.4cm}{B}
\cnode(4,1){0.5cm}{C}
\ncline{A}{B}
\ncline{A}{C}
\end{pspicture}
```

Ejemplo Nodos conectados con \ncline.

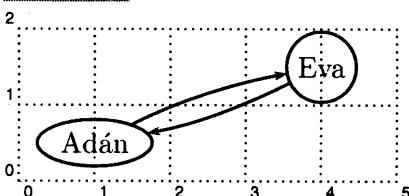


```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncline{A}{B}
\end{pspicture}
```

angleA= α	Ángulo de salida del conector desde el primer nodo. Valor por defecto: 0; α se mide grados sexagesimales.
angleB= α	Ángulo de llegada del conector al segundo nodo. Valor por defecto: 0; α se mide grados sexagesimales.
angle= α	Tiene el mismo efecto que las dos declaraciones angleA= α y angleB= α .
armA=n	Longitud del brazo de salida desde el primer nodo en ciertos conectores. Por defecto: 10pt; n debe ser una dimensión TEX .
armB=n	Longitud del brazo de llegada al segundo nodo en ciertos conectores. Por defecto: 10pt; n debe ser una dimensión TEX .
arm=n	Tiene el mismo efecto que las dos declaraciones armA=n y armB=n .
nodesepA=n	Separación entre el conector y el borde del primer nodo. Valor por defecto: 0pt; n debe ser una dimensión TEX .
nodesepB=n	Separación entre el conector y el borde del segundo nodo. Valor por defecto: 0pt; n debe ser una dimensión TEX .
nodesep=n	Tiene el mismo efecto que las dos declaraciones nodesepA=n y nodesepB=n .
arcangle= α	Ángulo entre el arco y la línea recta que une los nodos; se utiliza únicamente con \ncarc . Valor por defecto: 8; α se mide en grados sexagesimales.
loopsize=n	Longitud del brazo vertical de los bucles (<i>loops</i>) creados con \ncloop . Por defecto: 1cm; n debe ser una dimensión TEX .
offset=n	Desplazamiento (horizontal o vertical) del punto de conexión al primer nodo, n debe ser una dimensión TEX , positiva o negativa.

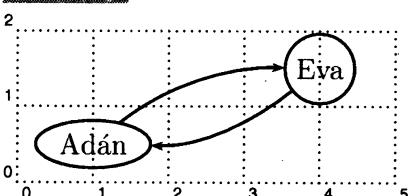
TABLA 11.16. Parámetros para conectores de nodos.

Ejemplo Nodos unidos con **\ncarc**.



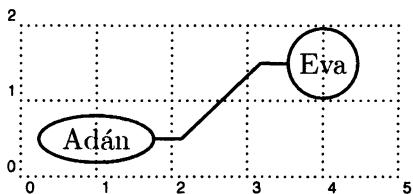
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncarc{->}{A}{B}
\ncarc{->}{B}{A}
\end{pspicture}
```

Ejemplo Nodos unidos con **\ncarc** utilizando el parámetro **arcangle=20**.



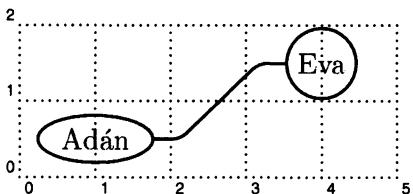
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncarc[arcangle=20]{->}{A}{B}
\ncarc[arcangle=20]{->}{B}{A}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncdiag`; ángulo de llegada al nodo B: 180.



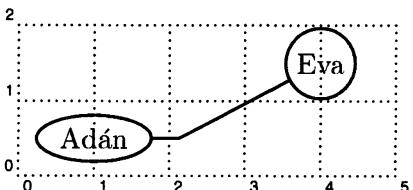
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncdiag[angleB=180]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncdiag`; el ángulo de llegada al nodo B es 180. Se usó `lineararc=0.3` para redondear los vértices del conector. Compárese con el ejemplo anterior.



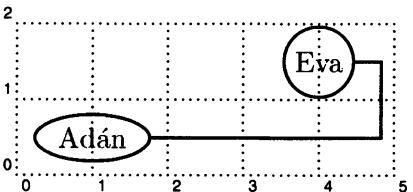
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncdiag[angleB=180,
lineararc=0.3]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncdiagg`; obsérvese que el brazo de llegada al segundo nodo se elimina.



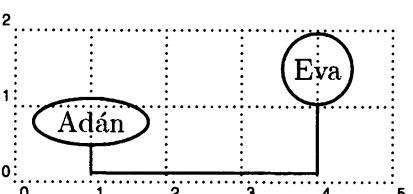
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncdiagg[angleB=180]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncbar`; el ángulo de salida del primer nodo y el ángulo de llegada al segundo nodo coinciden: 0 (valor por defecto).



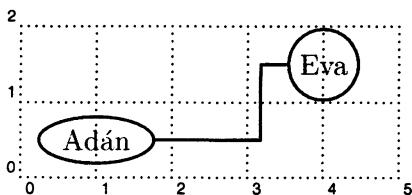
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncbar{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncbar`; el ángulo de salida del primer nodo y el ángulo de llegada al segundo nodo coinciden: -90. Compárese con el ejemplo anterior.



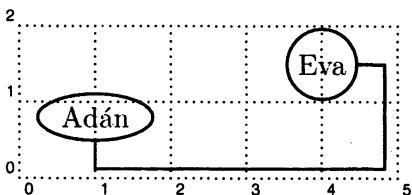
```
\begin{pspicture}(5,2)\grilla
\rput(1,0.8){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncbar[angleA=-90]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncangle`; el ángulo de llegada al segundo nodo es 180.



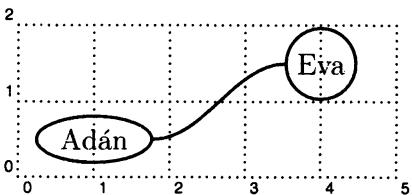
```
\begin{pspicture}(5,2)
\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncangle[angleB=180]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncangles`; el ángulo de salida del primer nodo es -90.



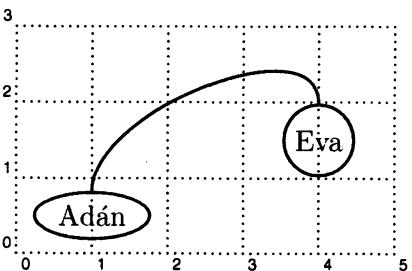
```
\begin{pspicture}(5,2)
\grilla
\rput(1,0.8){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\ncangles[angleA=-90]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\nccurve`; el ángulo de llegada al segundo nodo es 180.



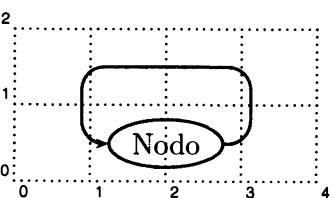
```
\begin{pspicture}(5,2)
\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\nccurve[angleB=180]{A}{B}
\end{pspicture}
```

Ejemplo Nodos unidos con `\nccurve`; tanto el ángulo de salida del primer nodo como el ángulo de llegada al segundo nodo es 90.



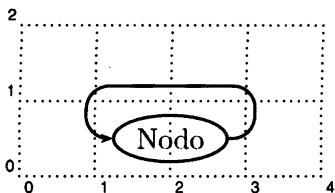
```
\begin{pspicture}(5,3)
\grilla
\rput(1,0.5){\ovalnode{A}{Ad\'an}}
\rput(4,1.5){\circlenode{B}{Eva}}
\nccurve[angleA=90,angleB=90]{A}{B}
\end{pspicture}
```

Ejemplo Bucle con ángulo de llegada 180.



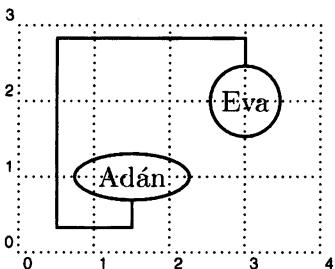
```
\begin{pspicture}(4,2)
\grilla
\rput(2,0.5){\ovalnode{A}{Nodo}}
\ncloop[angleB=180,linearc=0.3]{>}{A}{A}
\end{pspicture}
```

Ejemplo Compárese con el ejemplo anterior: el tamaño del brazo vertical del bucle, controlado por el parámetro `loopsize`, se redujo de 1 cm (valor por defecto) a 0.7 cm.



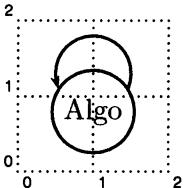
```
\begin{pspicture}(4,2)\grilla
\rput(2,0.5){\ovalnode{A}{Nodo}}
\ncloop[angleB=180,linearc=0.3,
       loopsize=0.7]{->}{A}{A}
\end{pspicture}
```

Ejemplo Nodos unidos con `\ncloop`; el ángulo de salida del primer nodo es -90 y el de llegada al segundo nodo es 90 .



```
\begin{pspicture}(4,3)\grilla
\rput(1.5,1){\ovalnode{A}{Ad\'an}}
\rput(3,2){\circlenode{B}{Eva}}
\ncloop[angleA=-90,angleB=90]{A}{B}
\end{pspicture}
```

Ejemplo Bucle (`\nccircle`) de 5 cm de radio en un nodo circular creado con `\circlenode`.



```
\begin{pspicture}(2,2)\grilla
\rput(1,0.8){\circlenode{A}{Algo}}
\nccircle[{->}]{A}{0.5cm}
\end{pspicture}
```

11.26.3. Etiquetas para los conectores de nodos

Para colocar etiquetas sobre los conectores de nodos podemos usar los comandos `\rput` (sección 11.14) y `\uput` (sección 11.15) pero **PSTricks** tiene los comandos de la Tabla 11.17, especialmente diseñados para tal propósito. Estos comandos se deben colocar inmediatamente después del conector de nodos correspondiente. Se asume que el conector va de izquierda a derecha; en caso contrario, se debe usar el comando opuesto. Como *opciones* de colocación se pueden usar los parámetros de la Tabla 11.18.

Todos los comandos de la Tabla 11.17 tienen su versión estrella: `\ncput*`, `\naput*`, etc, que tienen la misma sintaxis que la de sus homólogos y el mismo efecto, excepto que la caja L^AT_EX que contiene la *etiqueta* oculta todo lo que está detrás.

<code>\ncput[opciones]{etiqueta}</code>	Coloca la <i>etiqueta</i> sobre la línea misma del conector.
<code>\naput[opciones]{etiqueta}</code>	Coloca la <i>etiqueta</i> encima del conector.
<code>\nbput[opciones]{etiqueta}</code>	Coloca la <i>etiqueta</i> debajo del conector.
<code>\tvput[opciones]{etiqueta}</code>	En conectores verticales, coloca la <i>etiqueta</i> sobre la línea misma del conector.
<code>\tlput[opciones]{etiqueta}</code>	En conectores verticales, coloca la <i>etiqueta</i> a la izquierda del conector.
<code>\trput[opciones]{etiqueta}</code>	En conectores verticales, coloca la <i>etiqueta</i> a la derecha del conector.

TABLA 11.17. Comandos para colocar etiquetas a los conectores de nodos.

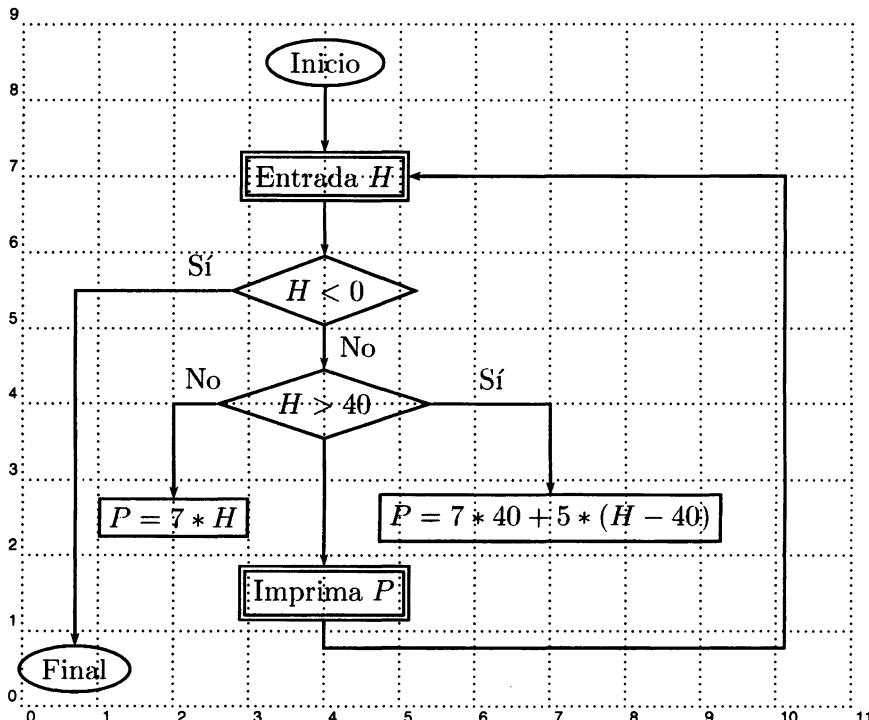
<code>labelsep=n</code>	Distancia entre la etiqueta y el conector. Por defecto: <code>5pt</code> ; <i>n</i> debe ser una dimensión \TeX .
<code>nrot=α</code>	Rotación de la etiqueta. Por defecto: 0; <i>α</i> debe ser un ángulo, entre -360 y 360 . Se pueden usar las abreviaturas U, L, D, etc. de la página 331. Una abreviatura muy útil es :U con la cual la etiqueta es paralela al conector.
<code>npos=m</code>	Sitio en el que aparece la etiqueta, de acuerdo con el número de segmentos y brazos del conector de nodos. Así, si $0 \leq \text{npos} \leq 1$, la etiqueta se coloca en el primer segmento del conector, si $1 < \text{npos} \leq 2$, se coloca en el segundo segmento, etc. Los valores que asume por defecto este parámetro, para cada conector de nodos, aparecen en la Tabla 11.19.

TABLA 11.18. Parámetros que controlan la colocación de etiquetas a los conectores de nodos.

Conector	Segmentos	Rango	Por defecto
<code>\ncline</code>	1	$0 \leq \text{npos} \leq 1$	0.5
<code>\nccurve</code>	1	$0 \leq \text{npos} \leq 1$	0.5
<code>\ncarc</code>	1	$0 \leq \text{npos} \leq 1$	0.5
<code>\ncbar</code>	3	$0 \leq \text{npos} \leq 3$	1.5
<code>\ncdiag</code>	3	$0 \leq \text{npos} \leq 3$	1.5
<code>\ncdiagg</code>	2	$0 \leq \text{npos} \leq 2$	0.5
<code>\ncangle</code>	3	$0 \leq \text{npos} \leq 3$	1.5
<code>\ncloop</code>	5	$0 \leq \text{npos} \leq 4$	2.5
<code>\nccircle</code>	1	$0 \leq \text{npos} \leq 1$	0.5

TABLA 11.19. Valores del parámetro `npos`.

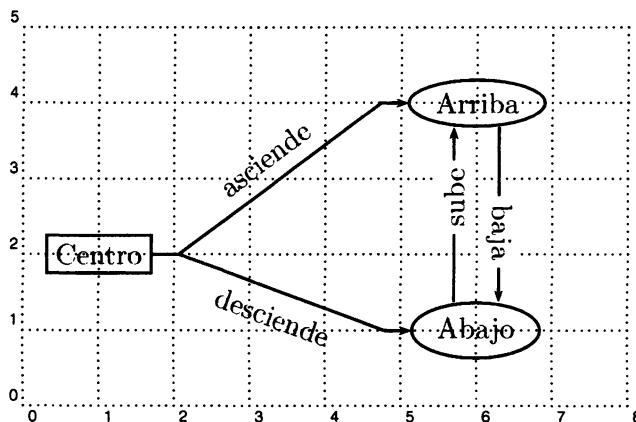
Ejemplo Para hacer el siguiente diagrama de flujo sólo se utilizaron los conectores `\ncline`, `\ncangle` y `\ncangles`.



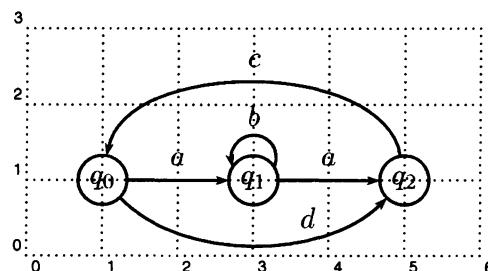
```

\begin{center}
\begin{pspicture}(11,9)\grilla
\rput(4,8.5){\ovalnode{C}{Inicio}}
\rput(0.7,0.5){\ovalnode{F}{Final}}
\rput(4,7){\rnode{E}{\psdblframebox{Entrada $H$}}}
\rput(4,5.5){\dianode{H1}{$H<0$}}
\rput(4,4){\dianode{H2}{$H>40$}}
\rput(2,2.5){\rnode{P1}{\psframebox{$P=7*H$}}}
\rput(7,2.5){\rnode{P2}{\psframebox{$P=7*40+5*(H-40)$}}}
\rput(4,1.5){\rnode{I}{\psdblframebox{Imprima $P$}}}
\ncline{->}{C}{E}
\ncline{->}{E}{H1}
\ncline{->}{H1}{H2}\trput{No}
\ncline{->}{H2}{I}
\ncangle[angleA=180,angleB=90]{->}{H2}{P1}\nbput[npos=0.3]{No}
\ncangle[angleB=90]{->}{H2}{P2}\naput[npos=0.5]{Si}
\ncangle[angleA=180,angleB=90]{->}{H1}{F}\nbput[npos=0.2]{Si}
\ncangles[angleA=-90,armB=5cm]{->}{I}{E}
\end{pspicture}
\end{center}

```

Ejemplo

```
\begin{center} \begin{pspicture}(8,5)\grilla
\rput(1,2){\rnode{A}{\psframebox{Centro}}}
\rput(6,4){\ovalnode{B}{Arriba}}
\rput(6,1){\ovalnode{C}{Abajo}}
\ncdiag[angleB=180]{->}{A}{B}\naput[nrot=:U]{asciende}
\ncdiag[angleB=180]{->}{A}{C}\nbput[nrot=:U]{desciende}
\ncline[offset=3mm]{->}{B}{C}\ncput*[nrot=-90,npos=0.6]{baja}
\ncline[offset=3mm]{->}{C}{B}\ncput*[nrot=90,npos=0.6]{sube}
\end{pspicture} \end{center}
```

Ejemplo

```
\begin{center} \begin{pspicture}(6,3)\grilla
\rput(1,1){\circlenode{Q0}{$q_0$}}
\rput(3,1){\circlenode{Q1}{$q_1$}}
\rput(5,1){\circlenode{Q2}{$q_2$}}
\ncline{->}{Q0}{Q1}\naput{$a$}
\ncline{->}{Q1}{Q2}\naput{$a$}
\nccurve[angleA=-45,angleB=-135]{->}{Q0}{Q2}
\naput[labelsep=0.1cm,npos=0.7]{$d$}
\nccurve[angleA=100,angleB=80]{->}{Q2}{Q0}\nbput{$c$}
\nccircle{->}{Q1}{0.3cm}\nbput[labelsep=0.1cm]{$b$}
\end{pspicture} \end{center}
```

11.27. Árboles

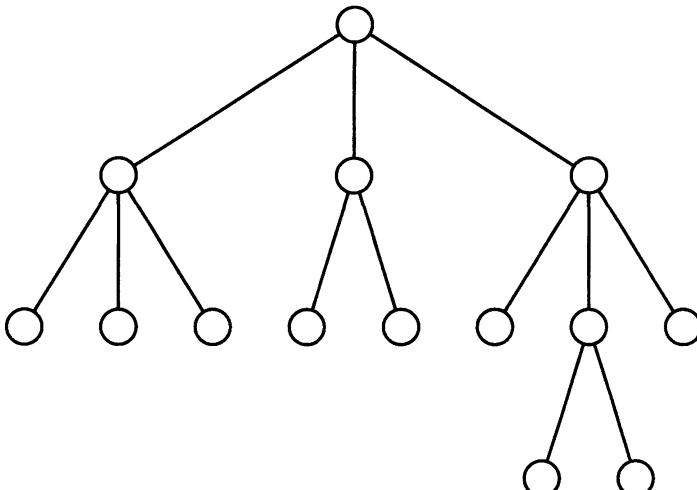
PSTricks tiene un módulo especial para el trazado de árboles. Un *árbol* (concepto matemático, no biológico) es un tipo muy particular de grafo (conjunto de nodos y aristas) en el que hay un nodo principal, llamado la raíz del árbol, con un cierto número de descendientes, cada uno de los cuales tiene, a su vez, 0, 1 o más descendientes. Estos descendientes pueden tener sus propios descendientes y así sucesivamente. La raíz de un árbol se caracteriza por la siguiente propiedad: existe una única trayectoria desde cada nodo hasta la raíz.

Un árbol se construye en la forma

```
\pstree[opciones]{raíz}{sub-árboles}
```

sin necesidad de usar `\begin{pspicture} ... \end{pspicture}`. Todos los nodos del árbol, incluyendo la raíz, se definen con los comandos de la Tabla 11.20. Se pueden usar las usuales *opciones* de trazado de líneas, junto con los parámetros específicos de la Tabla 11.21. Para añadir flechas o terminales a los conectores de nodos se usa el parámetro `arrows=...` con cualquiera de los valores de la Tabla 11.4.

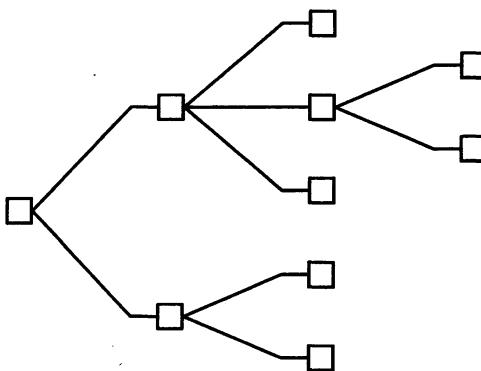
Ejemplo En el siguiente árbol todos los nodos son de la forma `\TC`, con el radio del círculo establecido por defecto (2.5mm).



```

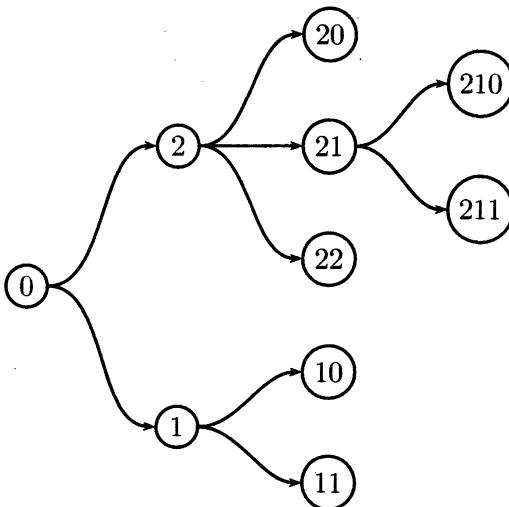
\begin{center}
\pstree{\TC}{%
  \pstree{\TC}{\TC \TC \TC}
  \pstree{\TC}{\TC \TC}
  \pstree{\TC}{\TC \pstree{\TC}{\TC \TC \TC}}
}
\end{center}
  
```

Ejemplo Árbol que crece hacia la derecha (`treemode=R`), con conectores de tipo `\ncdiag`, `armA=0pt` y `angleB=180`. Los nodos son de la forma `\Tf`.



```
\pstree[treemode=R,edge=\ncdiag,armA=0pt,angleB=180]{\Tf}{%
    \pstree{\Tf}{\Tf \pstree{\Tf}{\Tf \Tf} \Tf}
    \pstree{\Tf}{\Tf \Tf}}
```

Ejemplo Similar al ejemplo anterior pero con nodos de la forma `\Tcircle`, parámetro de conectores `edge=\nccurve` y flechas (`arrows=>`).



```
\pstree[treemode=R,edge=\nccurve,angleB=180,arrows=>]{\Tcircle{0}}{%
    \pstree{\Tcircle{2}}{\Tcircle{21}{\Tcircle{20}{}, \Tcircle{210}{}, \Tcircle{211}{}} \Tcircle{22}{}, \Tcircle{1}{\Tcircle{10}{}, \Tcircle{11}{}}}}
```

<code>\Tp[opciones]</code>	Nodo invisible.
<code>\Tdot[opciones]</code>	Nodo en forma de punto.
<code>\Tf[opciones]</code>	Nodo en forma de cuadrado.
<code>\Tfan[opciones]</code>	Nodo invisible conectado con su nodo predecesor por una figura triangular.
<code>\Tc[opciones]{radio}</code>	Nodo circular de <i>radio</i> dado.
<code>\TC[opciones]</code>	Similar a <code>\Tc</code> pero el radio se establece de antemano con el parámetro <i>radius</i> , utilizando la instrucción <code>\psset{radius=...}</code> . Es útil cuando hay varios nodos circulares del mismo radio en el árbol. Por defecto, <i>radius</i> = <i>2.5mm</i> .
<code>\Tr[opciones]{texto}</code>	Nodo rectangular con bordes invisibles que encierra el <i>texto</i> .
<code>\Tcircle[opciones]{texto}</code>	Nodo circular que encierra el <i>texto</i> .
<code>\Toval[opciones]{texto}</code>	Nodo ovalado que encierra el <i>texto</i> .
<code>\Tri[opciones]{texto}</code>	Nodo triangular que encierra el <i>texto</i> .
<code>\Tdia[opciones]{texto}</code>	Nodo en forma de rombo que encierra el <i>texto</i> .
<code>\Tn[opciones]{texto}</code>	Nodo que contiene el <i>texto</i> pero conectado a su nodo predecesor por una línea invisible. Sólo se puede usar en las hojas de un árbol (es decir, en los nodos sin descendientes).

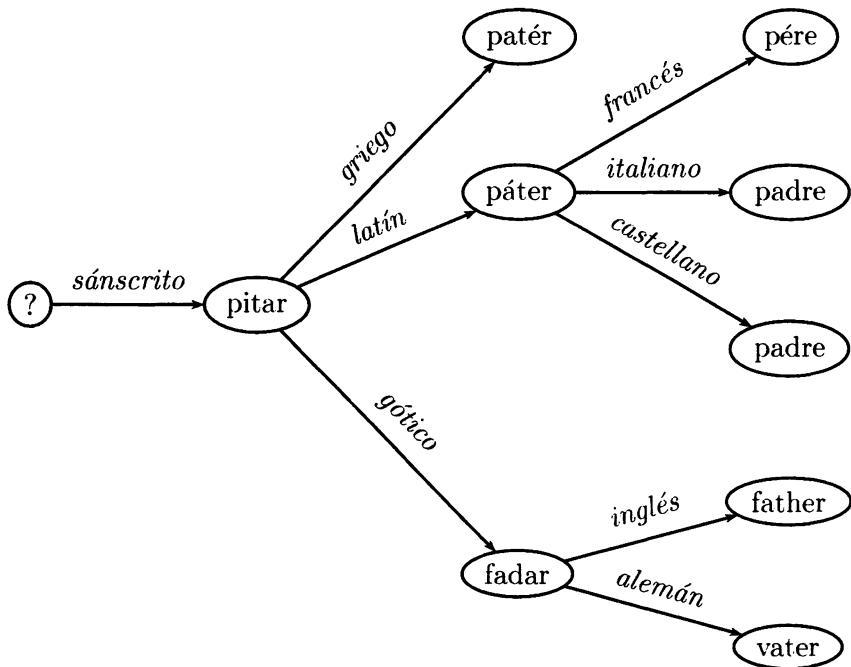
TABLA 11.20. Nodos para los árboles creados con `\pstree`.

<code>treemode=</code> U/D/R/L	Dirección de crecimiento del árbol. Por defecto: D (hacia abajo). Otros valores: U (hacia arriba), R (hacia la derecha), L (hacia la izquierda).
<code>edge=\conector</code>	Tipo de aristas o conectores de nodos usados en el árbol; se puede usar cualquiera de los conectores de nodos de la Tabla 11.15, con los parámetros de la Tabla 11.16. Por defecto: <code>edge=\ncline</code> .
<code>nodesep=n</code>	Distancia entre los nodos y las líneas conectoras. Por defecto: <code>0pt</code> ; <i>n</i> debe ser una dimensión TeX.
<code>levelsep=n</code>	Distancia entre los niveles de un árbol. Por defecto: <code>2cm</code> ; <i>n</i> debe ser una dimensión TeX. Si <i>n</i> está precedido de *, en la forma <code>levelsep=*</code> <i>n</i> , la distancia <i>n</i> es adicional al tamaño de los nodos.
<code>treesep=n</code>	Distancia entre los descendientes de un mismo nodo. Por defecto: <code>0.75cm</code> ; <i>n</i> debe ser una dimensión TeX.

TABLA 11.21. Parámetros que controlan el trazado de árboles.

Sobre los conectores de nodos podemos colocar etiquetas con los mismos comandos y parámetros mencionados en la sección anterior (Tablas 11.17 y 11.18). Para colocar una etiqueta sobre la línea que conecta el nodo N con su predecesor P, el comando de etiqueta se debe colocar inmediatamente después del nodo N, tal como se aprecia en el siguiente ejemplo.

Ejemplo Sobre los conectores de nodos se colocaron etiquetas usando el comando `\naput`. Se ha usado la opción `[nrot=:U]` para hacer que las etiquetas aparezcan paralelas a la líneas conectadoras (para más detalles sobre el parámetro `nrot` véase la Tabla 11.18).



```

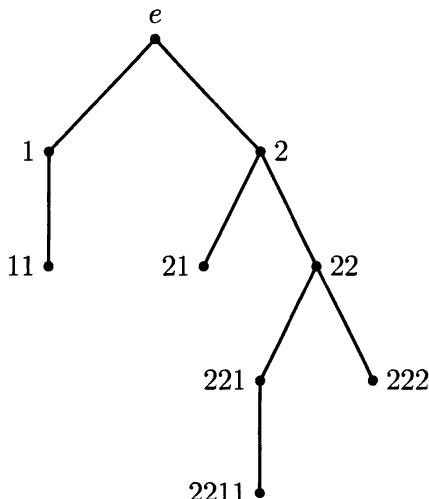
\begin{center}
\pstree*[treemode=R,arrows=->,levelsep=2cm,treesep=1.3cm]{?}{%
    \Tcircle{?}{}%
    \pstree{\Toval{pitar}}{\naput{\it sánscrito}}{%
        \Toval{patér}\naput[nrot=:U]{\it griego}%
        \pstree{\Toval{páter}}{\naput[nrot=:U]{\it latín}}{%
            \Toval{pére}\naput[nrot=:U]{\it francés}%
            \Toval{padre}\naput[nrot=:U]{\it italiano}%
            \Toval{padre}\naput[nrot=:U]{\it castellano}%
        }%
        \pstree{\Toval{fadar}}{\naput[nrot=:U]{\it góttico}}{%
            \Toval{father}\naput[nrot=:U]{\it inglés}%
            \Toval{vater}\naput[nrot=:U]{\it alemán}%
        }%
    }%
}
\end{center}

```

Los nodos de un árbol pueden tener sus propias etiquetas, que no deben confundirse con el *texto* mismo que aparece en el nodo ni con las etiquetas sobre los conectores de nodos. Las etiquetas de los nodos se escriben inmediatamente después del comando de cada nodo, en la forma `~{etiqueta}`. Hay dos parámetros que controlan la posición y la distancia de las etiquetas de los nodos:

<code>tnpos=a/b/l/r</code>	Posición de la etiqueta con relación al nodo. Por defecto: <code>b</code> (debajo). Otros valores: <code>a</code> (arriba), <code>r</code> (a la derecha), <code>l</code> (a la izquierda).
<code>tnsep=n</code>	Distancia entre el nodo y su etiqueta. Por defecto: <code>5pt</code> ; <code>n</code> debe ser una dimensión T _{EX} .

Ejemplo En este árbol todos los nodos son de la forma `\Tdot` y las etiquetas de los nodos se colocaron con `~{...}`. Debido al parámetro `tnpos=1`, las etiquetas aparecen a la izquierda, excepto en el nodo `e`, para el cual se usó `tnpos=a`, y en los nodos `2`, `22` y `222`, para los cuales se escribieron sendas declaraciones `tnpos=r`.



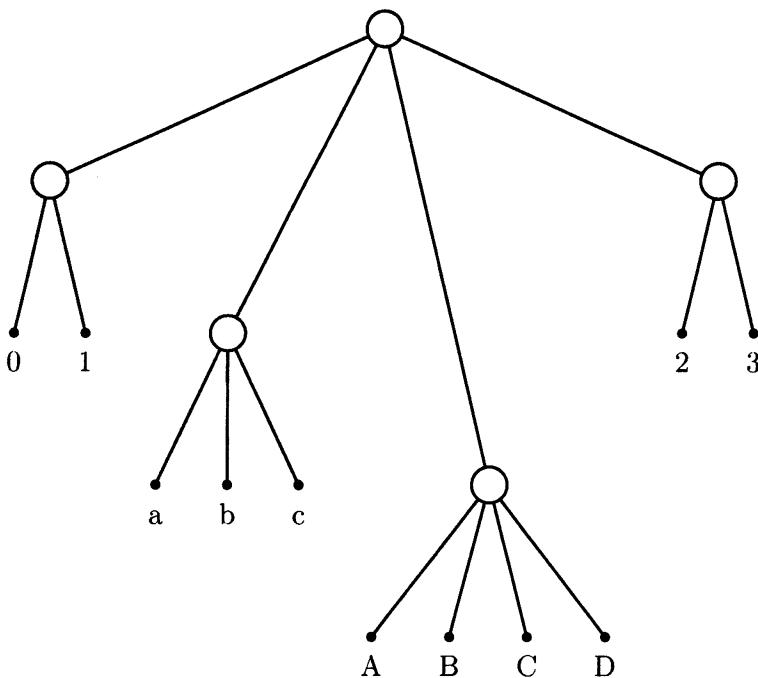
```
\begin{center}
\pstree[tnpos=l,treesep=1.5cm,levelsep=1.5cm]{%
\Tdot[tnpos=a,tnsep=2pt]~{$e$}}{%
\pstree{\Tdot~{1}}{\Tdot~{11}}
\pstree{\Tdot[tnpos=r]~{2}}{\Tdot~{21}}
\pstree{\Tdot[tnpos=r]~{22}}{%
\pstree{\Tdot~{221}}{\Tdot~{2211}}
\Tdot[tnpos=r]~{222}}}
\end{center}
```

Podemos colocar un nodo determinado un nivel más alejado del nivel que le corresponde. El comando necesario es:

`\skiplevel{nodos o sub-árboles}`

el cual salta un nivel; es decir, coloca los nodos o sub-árboles en el siguiente nivel.

Ejemplo Usando `\skiplevel{...}`, el sub-árbol cuyas hojas tienen etiquetas a, b, c se colocó un nivel más abajo. Para que el sub-árbol cuyas hojas tienen etiquetas A, B, C, D aparezca colocado dos niveles más abajo se ha utilizado `\skiplevel{\skiplevel{...}}`.



```

\begin{center}
\pstree{\TC}{\pstree{\TC}{\Tdot^0 \Tdot^1}
    \skiplevel{\pstree{\TC}{\Tdot^a \Tdot^b \Tdot^c}}
    \skiplevel{\skiplevel{\pstree{\TC}{%
        \Tdot^A \Tdot^B \Tdot^C \Tdot^D}}}
    \pstree{\TC}{\Tdot^2 \Tdot^3}}
\end{center}

```

11.28. Resortes y zigzags

El comando

```
\pszigzag[opciones]{flechas o extremos}(x1,y1)(x2,y2)
```

traza una línea en zigzag entre los puntos (x_1, y_1) y (x_2, y_2) . El comando

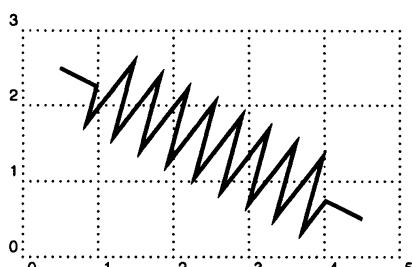
```
\pscoil[opciones]{flechas o extremos}(x1,y1)(x2,y2)
```

une los puntos (x_1, y_1) y (x_2, y_2) con una curva en forma de resorte. En ambos casos, se pueden usar las usuales *opciones* de trazado, junto con los parámetros específicos de la Tabla 11.22. Los extremos o flechas permitidos son los mismos de la Tabla 11.4.

coilwidth=<i>n</i>	Ancho del resorte o zigzag. Por defecto: 1cm; <i>n</i> debe ser una dimensión TeX.
coilheight=<i>a</i>	Determina el número de crestas de los resortes y zigzags; se debe tener $a > 0$. Entre menor sea <i>a</i> , más crestas se obtendrán. Por defecto: 1.
coilarmA=<i>n</i>	Longitud del brazo inicial del resorte o zigzag. Por defecto: 0.5cm; <i>n</i> debe ser una dimensión TeX.
coilarmB=<i>n</i>	Longitud del brazo final del resorte o zigzag. Por defecto: 0.5cm; <i>n</i> debe ser una dimensión TeX.
coilarm=<i>n</i>	Tiene el mismo efecto que las declaraciones coilarmA=<i>n</i> y coilarmB=<i>n</i> .
coilaspect=α	Determina el aspecto de los resortes (no afecta los zigzags); α es un ángulo tal que $0 \leq \alpha < 90$. Entre mayor sea α , más “apretado” será el resorte. Por defecto: 45.
coilinc=α	Determina la suavidad de la curva de los resortes (no afecta los zigzags). Entre mayor sea el ángulo α , más suave será la curva pero el procesamiento y la impresión tomarán más tiempo. Por defecto: 10.

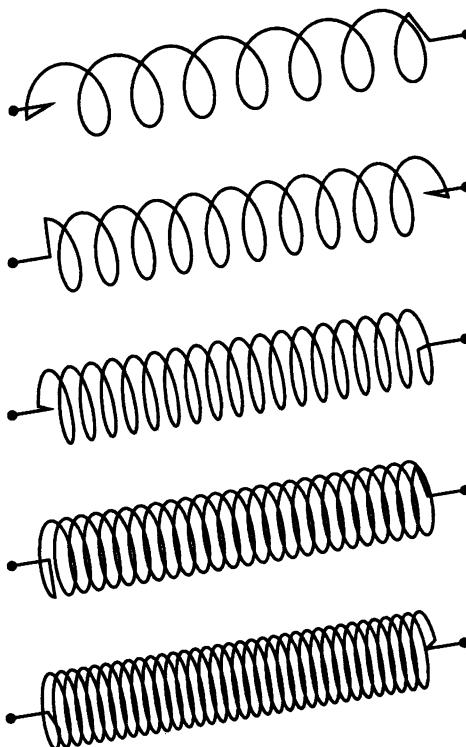
TABLA 11.22. Parámetros que controlan el trazado de resortes y zigzags.

Ejemplo



```
\begin{pspicture}(0,0)(5,3)
\grilla
\pszigzag[coilheight=0.4,
            linewidth=1.2pt](0.5,2.5)(4.5,0.5)
\end{pspicture}
```

Ejemplo El primer resorte (de arriba hacia abajo) se trazó con los parámetros establecidos por defecto, en el segundo resorte se utilizó el parámetro `coilheight=0.7` para aumentar el número de crestas, y en el tercero se aumentó todavía más por medio de `coilheight=0.4`. En el cuarto resorte también se utilizó `coilheight=0.4` pero se cambió el “aspecto” con el parámetro `coilaspect=60`. Para el último resorte se usaron los parámetros `coilheight=0.3` y `coilaspect=60`.



```
\begin{center}
\begin{pspicture}(0,0)(8,9)
\pscoil{**-*}(1,8)(7,9)
\pscoil[coilheight=0.7]{**-*}(1,6)(7,7)
\pscoil[coilheight=0.4]{**-*}(1,4)(7,5)
\pscoil[coilheight=0.4,coilaspect=60]{**-*}(1,2)(7,3)
\pscoil[coilheight=0.3,coilaspect=60]{**-*}(1,0)(7,1)
\end{pspicture}

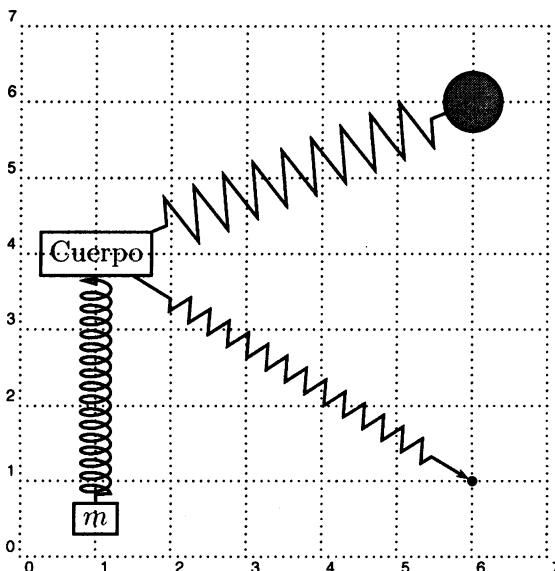
```

Los comandos `\pszigzag` y `\pscoil` tienen variantes para conectores de nodos, a saber:

```
\nczigzag[opciones]{flechas}{nombre nodo 1}{nombre nodo 2}
\nccoil[opciones]{flechas}{nombre nodo 1}{nombre nodo 2}
```

Los nombres de los nodos se deben asignar previamente con los comandos de la sección 11.26.1 (página 367).

Ejemplo



```
\begin{center}
\begin{pspicture}(0,0)(7,7)\grilla
\rput(1,4){\rnode{C}{\psframebox{Cuerpo}}}
\rput(1,0.5){\rnode{m}{\psframebox{$m$}}}
\dotnode(6,1){p}
\cnode[fillstyle=solid,fillcolor=gray](6,6){0.4cm}{P}
\nccoil[coilarm=0.1cm,coilwidth=0.4cm,coilheight=0.6]{C}{m}
\nczigzag[coilwidth=0.3cm]{->}{C}{p}
\nczigzag[coilarm=0.1cm,coilwidth=0.7cm,coilheight=0.6]{C}{P}
\end{pspicture}
\end{center}
```

11.29. Texto a lo largo de curvas

Uno de los más notables “trucos” de **PSTricks** es `\pstextpath`, con el cual se puede escribir texto a lo largo de curvas. Su sintaxis básica es:

```
\pstextpath[posición]{curva}{texto}
```

Cualquier objeto gráfico que produzca rectas o curvas se puede usar en el argumento *curva* y el argumento *texto* puede incluir fórmulas. El parámetro opcional *posición* puede tomar los siguientes valores: [1] (texto cargado hacia el inicio de la curva), [c] (texto centrado) y [r] (texto cargado hacia el final de la curva). Por defecto [1] está vigente. Cuando la longitud del *texto* excede la longitud de la *curva*, los resultados obtenidos son impredecibles.

Para que la curva misma no sea trazada, hay que colocar la instrucción `\psset{linestyle=none}` antes de `\begin{pspicture}`.

Como primer ejemplo, el lector puede examinar la frase a lo largo de la curva $y = x^4 - 2x^2$, en la página 310. La curva se trazó con el comando `\pscurve` (sección 11.20), a partir de cinco puntos.

Ejemplo Aquí se usan los comandos `\psarc` y `\psarcn` (sección 11.6) que trazan arcos de circunferencia.



```
\begin{center} \psset{linestyle=none}
\begin{pspicture}(-3,-3)(3,3)
\pstextpath[c]{\psarcn(0,0){2.5cm}{180}{0}}{%
    \LARGE \bf Asociación Mundial de}
\pstextpath[c]{\psarc(0,0){2.5cm}{180}{0}}{%
    \LARGE \bf Asociaciones Mundiales}
\end{pspicture} \end{center}
```

11.30. Texto con efectos especiales

El texto corriente se puede adornar con sombras o llenar con colores por medio de

```
\pscharpath[parámetros]{texto}
```

Este comando no requiere de `\begin{pspicture}` y por lo tanto se puede usar en un párrafo normal. Lo que se obtiene es el *contorno* de las letras del *texto*, con las modificaciones producidas por los *parámetros* indicados. Se pueden usar todos los parámetros de trazado (Tabla 11.3), de relleno (Tablas 11.5, 11.6 y 11.7) y de sombreado (Tabla 11.8), incluyendo trazado y relleno en colores. Los efectos se aprecian más en letras de gran tamaño.

Ejemplo

Arquímedes exclamó: ¡Eureka!

```
\pscharpath{\huge \bf Arquímedes exclamó:}\quad
\pscharpath[shadow=true,shadowsize=4pt,shadowcolor=lightgray,%
           fillstyle=solid,fillcolor=black]{\huge \bf !'Eureka!'}
```

Ejemplo

LATEX LATEX LATEX LATEX

```
\noindent \pscharpath[linestyle=dashed,dash=2pt 2pt]{%
  \Huge \bf \LaTeX}\quad
\pscharpath{\Huge \bf \LaTeX} \quad
\pscharpath[fillstyle=crosshatch,hatchsep=1pt,fillcolor=black]{%
  \Huge \bf \LaTeX}\quad
\pscharpath[fillstyle=solid,fillcolor=lightgray]{\Huge \bf \LaTeX}
```

Ejemplo

¡No subestime el poder de PSTRicks!

```
\pscharpath[fillstyle=solid,fillcolor=black,shadow=true,%
            shadowsize=4pt,shadowangle=45,shadowcolor=lightgray]{%
  \LARGE \bf !'No subestime el poder de PSTRicks!'}
```



Gráficas con el paquete **PICTEX**

12.1. El paquete **PICTEX** y su instalación

■ Los archivos que conforman el paquete **PICTEX** están incluidos en el CD adjunto, en la carpeta **/Paquetes/pictex/**.

El paquete **PICTEX**¹ es todo un clásico en la comunidad de **TEX**-nócratas. Debido a sus numerosos recursos gráficos y a su precisión, aún se sigue usando hoy en día, a pesar de las limitaciones que posee. Para los usuarios de **LATEX** los más serios inconvenientes son:

1. Incompatibilidad con otros paquetes. La mayoría de las incompatibilidades surgen porque **PICTEX** utiliza muchas variables de dimensión, que en **TEX** se definen con el comando **\dimen**. Al cargar otros paquetes que también utilicen variables de dimensión, se puede llegar rápidamente al límite máximo permitido por **TEX** de 256 variables **\dimen**, y el usuario recibe el siguiente mensaje de error:

! No room for a new \dimen.

No obstante, **PICTEX** se puede cargar simultáneamente con numerosos paquetes importantes sin encontrar conflicto alguno; entre ellos mencionamos: **amsmath**, **amsthm**, **amssymb**, **latexsym**, **color**, **colortbl**, **inputenc**, **graphics**, **graphicx**, **enumerate**, **array**, **makeidx**, **multind**, **caption2** y **fancyhdr**. El paquete **babel** es también compatible con los macros de **PICTEX**, a excepción del comando **\setbars** para gráficos de barras (véase la sección 12.18).

2. La sintaxis de los comandos de **PICTEX** no es siempre cómoda ni natural.

¹Creado en 1986-87 por Michael Wichura de la Universidad de Chicago.

Para trabajar con PICTEX en el ambiente LATEX se requieren tres archivos: `prepictex.tex`, `pictex.tex` y `postpictex.tex`². En las implementaciones TEX que respetan la estructura TDS (Apéndice C), estos tres archivos se deben incluir en la carpeta `/pictex/`, la cual debe ser una sub-carpeta de `/texmf/tex/latex/`.

Se accede a PICTEX escribiendo en el preámbulo del documento las siguientes instrucciones, en el orden indicado:

```
\DeclareFixedFont{\fiverm}{OT1}{cmr}{m}{n}{5pt}
\input{prepictex}
\input{pictex}
\input{postpictex}
```

`\DeclareFixedFont{\fiverm}{...}` es necesario porque PICTEX utiliza el comando `\fiverm` para acceder a la fuente `cmr` en el tamaño de 5 puntos y tal comando no está definido en LATEX.

12.2. Características de una gráfica en PICTEX

PICTEX procesa toda gráfica con respecto a un sistema coordenado bidimensional *x-y* definido por el usuario. El procedimiento más recomendable es comenzar toda gráfica especificando dos parámetros para el sistema coordenado a ser definido: las *unidades de medida* y el *área de trabajo*. Las unidades de medida se especifican con el comando

`\setcoordinatesystem units <—, —>`

Por ejemplo, si queremos unidades de un centímetro tanto en *x* como en *y*, escribimos

`\setcoordinatesystem units <1cm,1cm>`

Podemos especificar unidades diferentes para *x* y *y*, lo cual es muy conveniente para gráficas en las que alguna de las variables tiene un crecimiento o un decrecimiento muy pronunciado con respecto a la otra.

Ejemplo Los siguientes son escogencias correctas de unidades de medida:

`\setcoordinatesystem units <2cm,2cm>
\setcoordinatesystem units <0.5cm,0.5cm>
\setcoordinatesystem units <2cm,0.3cm>
\setcoordinatesystem units <1in,1cm>`

²En las primeras distribuciones de PICTEX, las raíces de los nombres de los archivos tenían un máximo de ocho letras: `prepictex.tex`, `pictex.tex` y `postpictex.tex`.

Habiendo definido las unidades de medida, especificamos el área de trabajo por medio de la instrucción

```
\setplotarea x from — to —, y from — to —
```

El usuario debe formarse de antemano una idea precisa del tamaño de la gráfica deseada, con respecto a la distancia entre márgenes (determinada por el formato del documento) y a las dimensiones del papel sobre el cual se va a imprimir. Esto no es en absoluto difícil, pero es esencial hacerlo con cuidado para resultados óptimos, ya que al especificar `\setplotarea`, PICTEX reserva en la página todo el espacio que le hemos pedido. Podríamos obtener excesivo espacio en blanco hacia arriba o hacia abajo si la parte visible de la gráfica no utiliza toda el área del sistema coordenado que ha sido definido. Así por ejemplo, al escribir

```
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -4 to 4, y from -3 to 3
```

PICTEX reserva un rectángulo de 8 cm de ancho por 6 cm de alto; este rectángulo invisible, con todo su contenido, será insertado en el papel.

Ejemplo Las siguientes declaraciones establecen 16 unidades horizontales (8 cm) y 4 unidades verticales (2 cm). El área de trabajo reservada por PICTEX es entonces un rectángulo 8 cm × 2 cm.

```
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -8 to 8, y from 0 to 4
```

Los ejes del sistema coordenado definido con `\setcoordinatesystem` son *invisibles* (existe un comando adicional, `\axis`, que los hace visibles, si se desea; véase la sección 12.11), y todos los comandos de PICTEX hacen referencia al sistema vigente.

Una gráfica desplegada y centrada se construye con la estructura:

```
\[
\begin{picture}
\setcoordinatesystem units <—, —>
\setplotarea x from — to —, y from — to —
:
comandos adicionales de la gráfica
:
\endpicture
\]
```

Para gráficas no desplegadas hay que omitir los comandos exteriores `\[` y `\]`.

12.3. Grillas

Con el comando

```
\grid {n} {m}
```

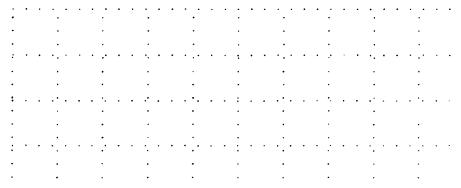
\TeX divide el ancho y la altura del área de trabajo, previamente establecidos con `\setplotarea`, en n columnas y m filas.

Ejemplo 10 cm de ancho y 2 cm de altura, establecidos con `\setplotarea`, se utilizan para trazar una grilla 8×4 (8 columnas y 4 filas).


```
\[\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -5 to 5, y from -1 to 1
\grid {8} {4}
\end{picture}\]
```

Una grilla de fondo puede ser útil como guía durante la elaboración de una gráfica. Para que sea útil, se debe definir `\grid {n} {m}` de tal manera que n sea el número de unidades horizontales y m el número de unidades verticales establecidas con `\setplotarea`. Además, el comando `\grid` puede estar precedido de la instrucción `\setdots` (descrita en la sección 12.14) para que las líneas de la grilla sean punteadas y no obstruyan la visibilidad de los demás elementos de la gráfica.

Ejemplo Grilla 10×4 , definida en concordancia con la definición del área de trabajo, `\setplotarea x from -5 to 5, y from -2 to 2`.



```
\[\begin{picture}
\setcoordinatesystem units <0.6cm,0.6cm>
\setplotarea x from -5 to 5, y from -2 to 2
\setdots
\grid {10} {4}
\end{picture}\]
```

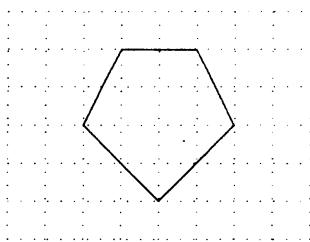
12.4. Rectas y polígonos

Se obtienen con

```
\setlinear \plot lista de puntos /
```

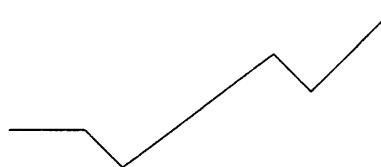
Nótese el *slash /* al final de la *lista de puntos*. Las coordenadas de los puntos se separan con uno o más espacios, y debe haber por lo menos un espacio antes del *slash* final */*.

Ejemplo Como guía para el lector, en este ejemplo usamos una grilla punteada 8×6 en el área de trabajo, en la forma explicada en la sección anterior, por medio de los comandos `\setdots \grid {8} {6}`.



```
\[\begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -4 to 4, y from -3 to 3
\setlinear \plot -2 0 -1 2 1 2 2 0 0 -2 -2 0 /
\setdots      %% Estos dos comandos se pueden
\grid {8} {6}  %% omitir para eliminar la grilla
\endpicture]
```

Ejemplo



```
\[\begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -6 to 6, y from -3 to 3
\setlinear
\plot -5 -1 -3 -1 -2 -2 2 1 3 0 5 2 /
\endpicture]
```

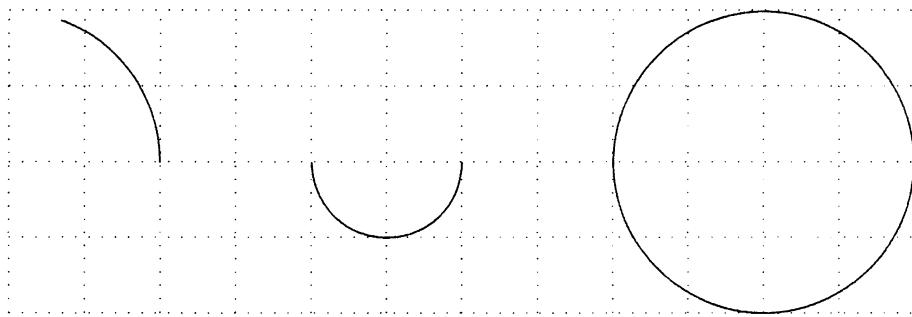
12.5. Circunferencias y arcos de circunferencia

El comando

```
\circulararc n degrees from x1 y1 center at x2 y2
```

traza un arco de n grados sexagesimales de la circunferencia centrada en (x_2, y_2) , a partir del punto (x_1, y_1) . El ángulo n puede tomar cualquier valor real entre -360 y 360 ; si es positivo, el arco es trazado en dirección contraria al giro de las manecillas del reloj, y en el sentido contrario si es negativo.

Ejemplo Como guía para el lector, utilizamos una grilla punteada 12×4 .



```
\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -6 to 6, y from -2 to 2
\circulararc 70 degrees from -4 0 center at -6 0
\circulararc -180 degrees from 0 0 center at -1 0
\circulararc 360 degrees from 2 0 center at 4 0
\setdots % Estos dos comandos se pueden
\grid {12} {4} % omitir para eliminar la grilla
\endpicture]
```

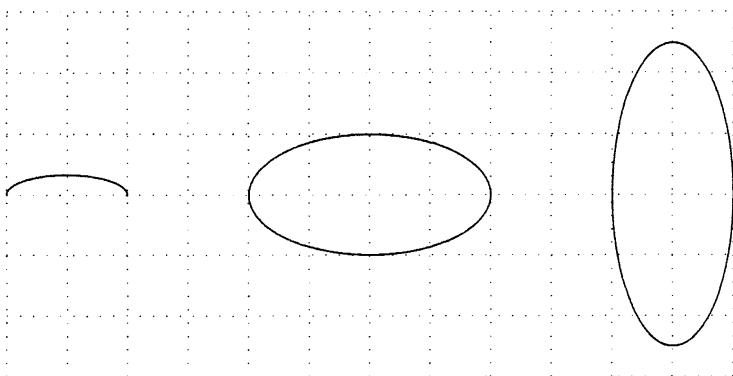
12.6. Elipses y arcos de elipse

El comando

```
\ellipticalarc axes ratio a:b n degrees from x1 y1 center at x2 y2
```

traza un arco de n grados sexagesimales de la elipse centrada en (x_2, y_2) , con semieje horizontal a y semieje vertical b , a partir del punto (x_1, y_1) . El ángulo n puede tomar cualquier valor real entre -360 y 360 ; si es positivo, el arco es trazado en dirección contraria al giro de las manecillas del reloj, y en el sentido contrario si es negativo.

Ejemplo Como guía para el lector, utilizamos una grilla punteada 12×6 .



```
\[begin{picture}
\setcoordinatesystem units <0.8cm,0.8cm>
\setplotarea x from -6 to 6, y from -3 to 3
\ellipticalarc axes ratio 3:1 180 degrees from -4 0 center at -5 0
\ellipticalarc axes ratio 2:1 360 degrees from 2 0 center at 0 0
\ellipticalarc axes ratio 1:2.5 360 degrees from 6 0 center at 5 0
\setdots          %% Estos dos comandos se pueden
\grid {12} {6}    %% omitir para eliminar la grilla
\endpicture\]
```

12.7. El comando `\put`

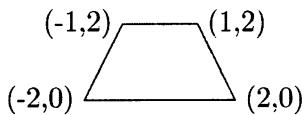
El comando

`\put {objeto} at x_1 y_1`

se usa para colocar un *objeto* en un punto (x_1, y_1) del área de trabajo. El *objeto* puede ser texto corriente, fórmulas, símbolos, o incluso otra gráfica. Este *objeto* queda, por defecto, centrado horizontal y verticalmente con respecto al punto indicado pero también existe la opción de moverlo a la izquierda, a la derecha, hacia arriba o hacia abajo con las opciones [l], [r], [t], [b], respectivamente. El resultado obtenido se resume en la siguiente tabla.

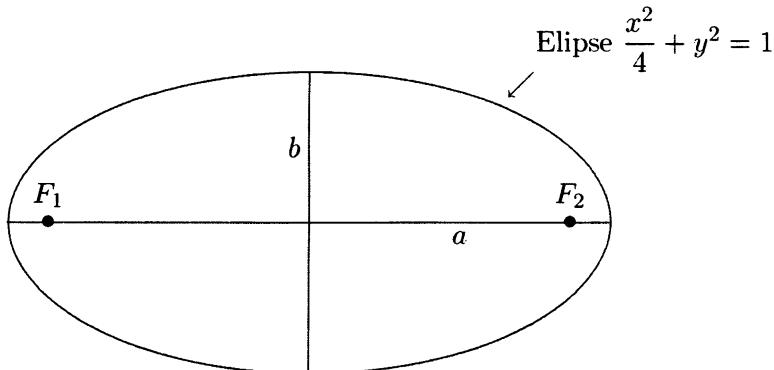
Comando	Efecto en la gráfica
<code>\put {objeto} [l] at x_1 y_1</code>	(x_1, y_1) está a la izquierda del <i>objeto</i>
<code>\put {objeto} [r] at x_1 y_1</code>	(x_1, y_1) está a la derecha del <i>objeto</i>
<code>\put {objeto} [t] at x_1 y_1</code>	(x_1, y_1) está arriba del <i>objeto</i>
<code>\put {objeto} [b] at x_1 y_1</code>	(x_1, y_1) está debajo del <i>objeto</i>

También se pueden usar las opciones [lt], [lb], [rt] y [rb] para obtener simultáneamente el efecto de dos opciones individuales.

Ejemplo

```
\[ \begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -6 to 6, y from 0 to 2
\setlinear \plot -2 0 -1 2 1 2 2 0 -2 0 /
\put {\(2,0)} [l] at 2 0
\put {\(-2,0)\} [r] at -2 0
\put {\(-1,2)\} [r] at -1 2
\put {\(1,2)} [l] at 1 2
\endpicture \]
```

Ejemplo Nótese que tanto la flecha ↗ como la frase ‘Elipse $\frac{x^2}{4} + y^2 = 1$ ’ aparecen colocadas en el mismo punto coordinado (1.5, 1) pero trasladadas con los modificadores [rt] y [lb], respectivamente.



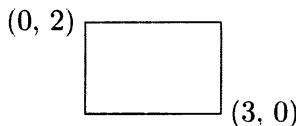
```
\[ \begin{picture}
\setcoordinatesystem units <2cm,2cm>
\setplotarea x from -3 to 3, y from -1 to 1
\ellipticalarc axes ratio 2:1 360 degrees from 2 0 center at 0 0
\setlinear \plot -2 0 2 0 /
\setlinear \plot 0 -1 0 1 /
\put {$F_1$} [b] at -1.73 0.1
\put {$F_2$} [b] at 1.73 0.1
\put {$a$} at 1 -0.1
\put {$b$} at -0.1 0.5
\put {$\bullet$} at -1.73 0
\put {$\bullet$} at 1.73 0
\put {$\swarrow$} [rt] at 1.5 1
\put {Elipse $\frac{x^2}{4} + y^2 = 1$} [lb] at 1.5 1
\endpicture \]
```

12.8. Rectángulos e histogramas

Cualquier rectángulo se puede trazar con la lista de sus cuatro vértices, utilizando el ya citado comando `\setlinear \plot`. Sin embargo, PICTEX posee una manera más fácil de dibujar rectángulos, indicando sólo dos vértices opuestos:

```
\putrectangle corners at  $x_1$   $y_1$  and  $x_2$   $y_2$ 
```

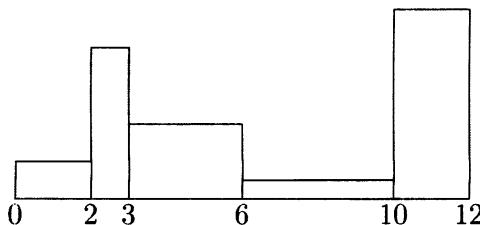
Ejemplo



```
\[\begin{picture}
\setcoordinatesystem units <0.6cm,0.6cm>
\setplotarea x from 0 to 5, y from 0 to 2
\putrectangle corners at 0 2 and 3 0
\put {\{(0, 2)\}} [r] at -0.2 2
\put {\{(3, 0)\}} [l] at 3.2 0
\endpicture\]
```

Un comando relacionado es `\sethistograms \plot ... /`, que sirve para trazar histogramas, indicando solamente el extremo inferior izquierdo del histograma y el vértice superior derecho de cada rectángulo.

Ejemplo



```
\[\begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from 0 to 12, y from -0.5 to 5
\sethistograms \plot 0 0 2 1 3 4 6 2 10 0.5 12 5 /
\put {\{0\}} at 0 -0.4
\put {\{2\}} at 2 -0.4
\put {\{3\}} at 3 -0.4
\put {\{6\}} at 6 -0.4
\put {\{10\}} at 10 -0.4
\put {\{12\}} at 12 -0.4
\endpicture\]
```

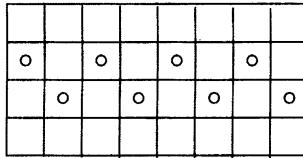
12.9. El comando \multiput

La sintaxis de \multiput es similar a la de \put. Se usa para colocar un *objeto* múltiples veces en el área de trabajo sin tener que repetir la construcción \put para cada uno de ellos. Su forma básica es:

\multiput { <i>objeto</i> } at <i>lista de puntos</i> /

El *slash /* al final de la *lista de puntos* debe estar precedido de por lo menos un espacio en blanco.

Ejemplo Para esta gráfica se han definido, utilizando \newcommand, los objetos \rectah y \rectav, los cuales son a su vez gráficas construidas con el esquema \begin{picture} ... \end{picture}. El objeto \rectah es la línea horizontal que va de (0, 0) a (8, 0) y que se repite 5 veces en la gráfica, mientras que \rectav es la línea vertical que va de (0, 0) a (0, 4), la cual se repite 9 veces.



```
\[ \begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from 0 to 8, y from 0 to 4
\newcommand{\rectah}{%
\begin{picture} \setlinear \plot 0 0 8 0 / \end{picture}}
\newcommand{\rectav}{%
\begin{picture} \setlinear \plot 0 0 0 4 / \end{picture}}
\multiput {\rectah} at 0 0 0 1 0 2 0 3 0 4 /
\multiput {\rectav} at 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0
8 0 /
\multiput {$\circ$} at 1.5 1.5 3.5 1.5 5.5 1.5 7.5 1.5 0.5 2.5
2.5 2.5 4.5 2.5 6.5 2.5 /
\endpicture]
```

El comando \multiput es susceptible de otra simplificación, que podemos ilustrar con referencia al ejemplo anterior. La instrucción

\multiput {\rectah} at 0 0 0 1 0 2 0 3 0 4 /

se puede reemplazar por \multiput {\rectah} at 0 0 *4 0 1 /. Aquí, *4 indica las 4 repeticiones de \rectah, comenzando en el punto (0,0), siguiendo en (0,1), etc, con igual espaciamiento. Similarmente, las líneas verticales se pueden trazar con \multiput {\rectav} at 0 0 *8 1 0 /, donde *8 indica las 8 repeticiones necesarias, comenzando en (0,0), siguiendo en (1,0), etc, con igual espaciamiento.

12.10. Letras y palabras apiladas

PICTEX tiene el comando `\stack` para colocar letras o palabras verticalmente apiladas; su sintaxis es:

`\stack [alineación] <separación> {lista}`

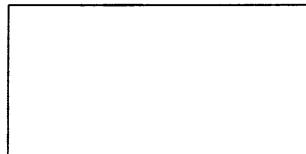
La *lista* contiene las palabras o letras que se van a apilar, de arriba hacia abajo, separadas entre sí por comas (sin espacios en blanco ni antes ni después de las comas). Si alguno de los items de la lista tiene una coma, éste se debe encerrar entre llaves `{ }`.

El parámetro *[alineación]*, que es opcional, toma los valores `[1]` o `[r]`, y se usa para indicar la alineación o justificación deseada de los items de la lista, ya sea a la izquierda o a la derecha. Por defecto, los items aparecen centrados. El parámetro *<separación>*, también opcional, se usa para indicar la separación vertical entre los items de la lista. Por defecto, PICTEX utiliza para este parámetro el valor `0.17\baselineskip`.

Una pila construida con `\stack` se coloca en una gráfica particular usando `\put` o `\multiput`.

Ejemplo

Algo a la
izquierda del
rectángulo



Algo a la
derecha
de
rectángulo

```
\[ \begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -5 to 5, y from 0 to 2
\putrectangle corners at -2 0 and 2 2
\put {\stack {Algo a la,izquierda del,rectángulo}} at -3.5 1
\put {\stack {Algo a la,d,e,r,e,c,h,a}} at 3.5 1
\end{picture}\]
```

El comando `\stack` también se puede usar en párrafos normales, sin especificar de antemano `\begin{picture} ... \end{picture}`. En tales casos, la pila o *stack* y el texto circundante quedan alineados por la parte inferior.

pila
de palabras
poco

Ejemplo Frase corriente con una corriente en medio de ella.

Frase corriente con una `\stack{pila,de palabras,poco,corriente}\` en medio de ella.

12.11. Ejes coordenados

El comando `\axis`, para trazar ejes coordenados, es uno de los más versátiles de `PTeX`; tiene cuatro formas básicas:

<code>\axis bottom opciones /</code>	Traza un eje horizontal en la parte inferior del área de trabajo.
<code>\axis top opciones /</code>	Traza un eje horizontal en la parte superior del área de trabajo.
<code>\axis left opciones /</code>	Traza un eje vertical en el extremo izquierdo del área de trabajo.
<code>\axis right opciones /</code>	Traza un eje vertical en el extremo derecho del área de trabajo.

Obsérvese que la construcción `\axis` termina con un *slash /*, precedido de un espacio en blanco. Los ejes se extienden (horizontal o verticalmente, según sea el caso) a lo largo de todo el espacio reservado con `\setplotarea`.

Se pueden colocar *ticks* o marcas sobre los ejes, escogiendo para el parámetro de *opciones* una de las siguientes modalidades:

- **ticks unlabeled from a_1 to a_2 by d**

Se trazan marcas no etiquetadas, a intervalos de d unidades, desde la coordenada a_1 hasta la coordenada a_2 .

- **ticks numbered from a_1 to a_2 by d**

Se trazan marcas numeradas, a intervalos de d unidades, desde la coordenada a_1 hasta la coordenada a_2 .

- **ticks withvalues $v_1 v_2 \dots / at a_1 a_2 \dots /$**

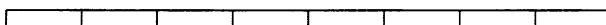
Se trazan marcas con etiquetas v_1, v_2, \dots en las coordenadas especificadas por a_1, a_2, \dots . Etiquetas vacías se señalan con {}.

- **ticks withvalues $v_1 v_2 \dots / quantity n$**

Se trazan n marcas con las etiquetas v_1, v_2, \dots , de extremo a extremo del eje y uniformemente separadas entre sí. Etiquetas vacías se señalan con {}.

Para cambiar algunos de los parámetros (longitud de las marcas, posición de los ejes, etc) con los que `PTeX` traza los ejes coordinados existen varios *modificadores*, mostrados en la Tabla 12.1. Estos modificadores son enteramente opcionales y se deben escribir en los sitios señalados.

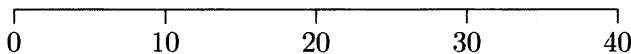
in	Hace que las marcas o <i>ticks</i> sean trazadas hacia el interior del área de trabajo; por defecto, se trazan hacia afuera (modificador out). Se coloca inmediatamente después de ticks .
short	Traza marcas cortas (por defecto se trazan con el modificador long). Se coloca inmediatamente después de ticks .
length <ℓ>	Hace que las marcas tengan longitud ℓ . Se coloca inmediatamente después de ticks .
andacross	Hace que las marcas se extiendan de extremo a extremo del área de trabajo. Se coloca inmediatamente después de ticks .
butnotacross	Tiene el efecto contrario de andacross ; \LaTeX lo utiliza por defecto.
shiftedto $x=x_1$	Colocada después de right o left , traslada un eje vertical hasta la coordenada $x = x_1$.
shiftedto $y=y_1$	Colocada después de top o bottom , traslada un eje horizontal hasta la coordenada $y = y_1$.
label {...}	Crea un rótulo o etiqueta para el eje, el cual aparece centrado horizontal o verticalmente, según sea el caso. No hay que reservar espacio adicional en \setplotarea . Se coloca inmediatamente después de \axis .

TABLA 12.1. Modificadores del comando **\axis**.**Ejemplo**

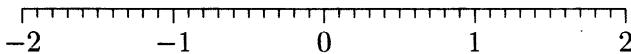
```
\[ \begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -4 to 4, y from 0 to 0.3
\axis bottom ticks unlabeled from -4 to 4 by 1 /
\end{picture}\]
```

Ejemplo

```
\[ \begin{picture}
\setcoordinatesystem units <1.5cm,1cm>
\setplotarea x from -3 to 4, y from 0 to 0.3
\axis top ticks numbered from -3 to 4 by 1 /
\end{picture}\]
```

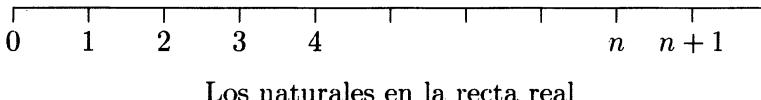
Ejemplo

```
\[
\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -4 to 4, y from 0 to 0.3
\axis bottom ticks withvalues 0 10 20 30 40 / quantity 5 /
\endpicture
\]
```

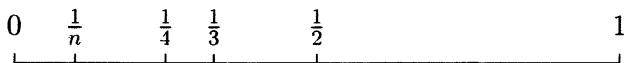
Ejemplo

```
\[
\begin{picture}
\setcoordinatesystem units <2cm,1cm>
\setplotarea x from -2 to 2, y from 0 to 0.3
\axis bottom ticks
  numbered from -2 to 2 by 1
  short unlabeled from -2 to 2 by 0.1 /
\endpicture
\]
```

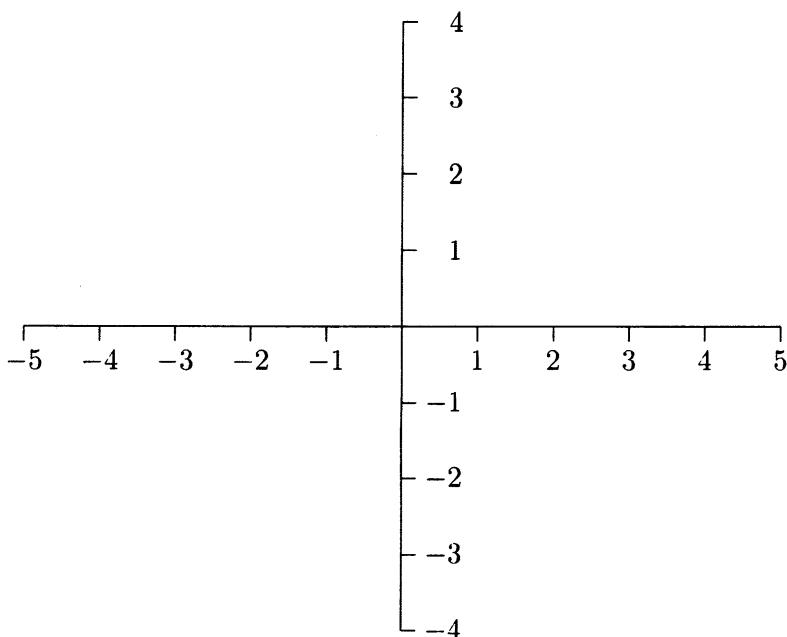
Ejemplo En este ejemplo se usa el modificador **label**.



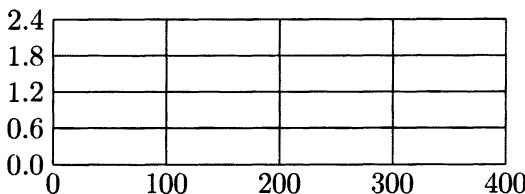
```
\[
\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from 0 to 10, y from 0 to 0.3
\axis label {Los naturales en la recta real} bottom ticks
  withvalues {0} {1} {2} {3} {4} {} {} {} {$n$} {$n+1$} {} /
  quantity 11 /
\endpicture
\]
```

Ejemplo

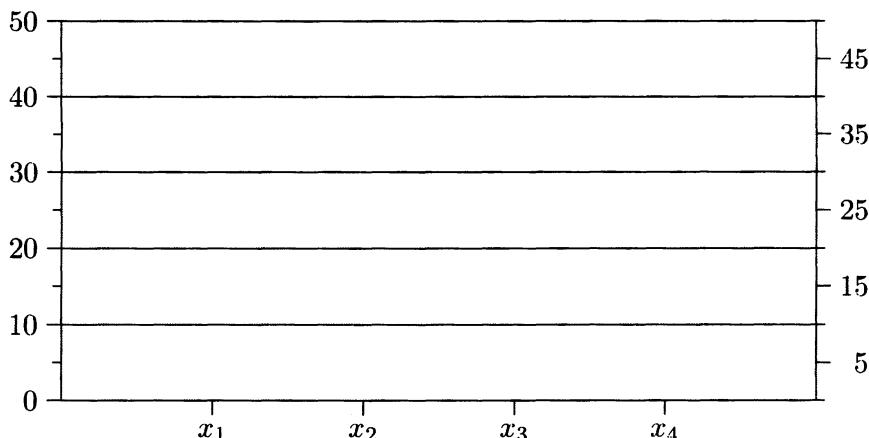
```
\[ \begin{picture}
\setcoordinatesystem units <8cm,1cm>
\setplotarea x from 0 to 1, y from 0 to 0.2
\axis top ticks short withvalues {0} {$\frac{1}{n}$} {$\frac{1}{4}$} {$\frac{1}{3}$} {$\frac{1}{2}$} {1} / at 0 0.1 0.25
0.33 0.5 1 / /
\endpicture \]
```

Ejemplo

```
\[ \begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -5 to 5, y from -4 to 4
\axis bottom shiftedto y=0 ticks
numbered from -5 to -1 by 1 numbered from 1 to 5 by 1 /
\axis right shiftedto x=0 ticks
numbered from -4 to -1 by 1 numbered from 1 to 4 by 1 /
\endpicture \]
```

Ejemplo

```
\[begin{picture}
\setcoordinatesystem units <2cm,0.8cm>
\setplotarea x from 0 to 3, y from 0 to 2.4
\axis left ticks
in and across numbered from 0 to 2.4 by 0.6 /
\axis bottom ticks
in and across withvalues {0} {100} {200} {300} {400} / quantity 5 /
\endpicture]
```

Ejemplo

```
\[begin{picture}
\setcoordinatesystem units <1cm,1mm>
\setplotarea x from 0 to 10, y from 0 to 50
\axis right ticks
numbered from 5 to 45 by 10
short unlabeled from 0 to 50 by 10 /
\axis left ticks
short unlabeled from 5 to 45 by 10
and across long numbered from 0 to 50 by 10 /
\axis bottom ticks
withvalues {$x_1$} {$x_2$} {$x_3$} {$x_4$} / at 2 4 6 8 //
\endpicture]
```

12.12. Trazado de curvas

\LaTeX tiene la capacidad de trazar curvas arbitrarias en dos dimensiones, por interpolación cuadrática a partir de una lista dada de puntos. El algoritmo que emplea \LaTeX es un tanto restrictivo y lento. El comando general es

```
\setquadratic \plot lista de puntos /
```

sujeto a las condiciones siguientes:

1. El número de puntos debe ser impar.
2. Los tres primeros puntos, P_0 , P_1 y P_2 , deben estar *aproximadamente* en línea recta y P_1 debe estar en el segundo tercio del arco entre P_0 y P_2 . Similarmente, los puntos P_2 , P_3 y P_4 , deben estar *aproximadamente* en línea recta y P_3 debe estar en el segundo tercio del arco entre P_2 y P_4 ; y así sucesivamente.

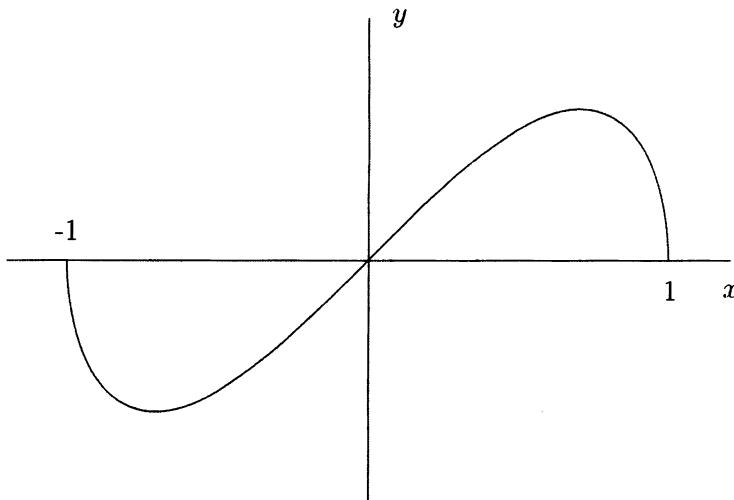
La primera condición es indispensable: al violarla, `\setquadratic` es ejecutado con $(0,0)$ como punto adicional y se emite el siguiente mensaje de error: ‘**Missing number treated as 0**’. La segunda condición es necesaria para que la interpolación realizada, y por lo tanto la curva obtenida, sean *aceptables*. \LaTeX traza una curva aun en el caso de que una o más triples de puntos no satisfagan la condición (2) pero, por lo general, los resultados serán catastróficos: una curva con “picos”, en forma de zigzag.

Para utilizar acertadamente el comando `\setquadratic`, el usuario debe hacer acopio de su paciencia y sobretodo de su Cálculo I. Es necesario hacer un bosquejo previo de la curva, detectando los puntos máximos, mínimos y los puntos de inflexión (es decir, los puntos de cambio de curvatura). Estos puntos destacados deben ser escogidos junto con puntos adicionales a su alrededor, de tal manera que se satisfagan siempre las condiciones (1) y (2). Típicamente, son necesarios dos o tres ensayos con asignaciones de puntos para un resultado satisfactorio. \LaTeX brinda una ayuda en relación con la condición (2): el archivo ‘`-.log`’ registra cuándo la condición (2) no ha sido satisfecha y nos da la razón de arcos. Un ejemplo de tal mensaje es:

```
3th point being plotted doesn't lie in the middle third of
the arc between the 2th and the 4th point.
[arc length 2 to 3]/[arc length 2 to 4]=0.28491.
```

Para que la condición (2) se satisfaga en este caso concreto, debe cumplirse que $0.333 = \frac{1}{3} \leq [\text{arc length 2 to 3}]/[\text{arc length 2 to 4}] \leq \frac{2}{3} = 0.666$. El valor 0.28491 del ejemplo indica que P_3 está muy cerca de P_2 y muy lejos de P_4 . Por lo tanto, debe hacerse una nueva escogencia para P_2 .

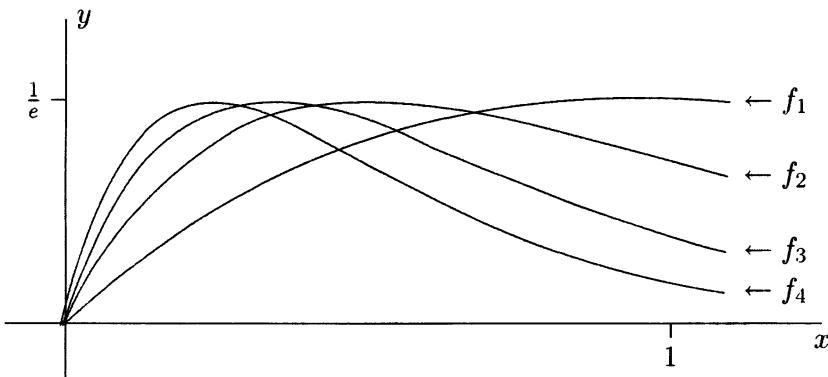
Ejemplo Consideremos la función $f(x) = x\sqrt{1 - x^2}$ en el intervalo $[-1, 1]$. Usando Cálculo I sabemos de antemano que la curva se anula en $(-1, 0)$, $(0, 0)$ y $(1, 0)$; tiene un punto máximo en $(1/\sqrt{2}, 0.5)$ y un punto mínimo en $(-1/\sqrt{2}, -0.5)$. Se han escogido estos cinco puntos destacados y puntos adicionales para un total de 21 puntos: 10 en $[-1, 0]$, 10 en $[0, 1]$ (simétricamente) y el punto $(0, 0)$. Esta escogencia de puntos resultó ser adecuada después de varios intentos y con la ayuda de los informes de compilación de los archivos ‘*.log’. Se han usado `\axis` y `\put` para los demás aditamentos de la gráfica.



Gráfica de $f(x) = x\sqrt{1 - x^2}$, $-1 \leq x \leq 1$

```
\begin{picture}
\setcoordinatesystem units <4cm,4cm>
\setplotarea x from -1.2 to 1.2, y from -0.8 to 0.8
\axis top shiftedto y=0 /
\axis right shiftedto x=0 /
\put {-1} at -1 0.1
\put {1} at 1 -0.1
\put {$x$} at 1.2 -0.1
\put {$y$} at 0.1 0.8
\setquadratic \plot -1 0 -0.99 -0.1396 -0.96 -0.2688
-0.92 -0.3605 -0.87 -0.4289 -0.7071 -0.5
-0.5 -0.4330 -0.35 -0.32786 -0.25 -0.242 -0.15 -0.1483
0 0 0.15 0.1483 0.25 0.242 0.35 0.32786 0.5 0.4330
0.7071 0.5 0.87 0.4289 0.92 0.3605 0.96 0.2688
0.99 0.1396 1 0 /
\endpicture
\centerline{Gráfica de $f(x)=x\sqrt{1-x^2}$, \quad -1\leq x\leq 1$}
```

Ejemplo Gráficas de las funciones $f_n(x) = nxe^{-nx}$, $n = 1, 2, 3, 4$. La función f_n tiene un punto máximo en $(\frac{1}{n}, \frac{1}{e})$; para cada una de las cuatro funciones consideradas hemos tomado los puntos $(0, 0)$ y $(\frac{1}{n}, \frac{1}{e})$, y puntos adicionales a su alrededor, con diferentes espaciamientos dependiendo del valor de n . Nótese que f_1 es “más lineal” que f_2 y ésta lo es más que f_3 , etc, así que se requieren más puntos al incrementar el valor de n . La flecha \leftarrow se obtiene con $\$\\gets\$$.



```
\[begin{picture}
\setcoordinatesystem units <8cm,8cm>
\setplotarea x from -0.1 to 1.25, y from -0.1 to 0.5
\axis left shiftedto x=0 ticks withvalues {$\frac{1}{e}$} / at
0.3678 / /
\axis bottom shiftedto y=0 ticks withvalues {1} / at 1 / /
\setquadratic \plot % OJO f_1
0 0 0.5 0.3032 1.1 0.3661 /
\setquadratic \plot % OJO f_2
0 0 0.11 0.17655 0.285 0.32235 0.5 0.3678
0.8 0.323 0.95 0.28418 1.1 0.2437 /
\setquadratic \plot % OJO f_3
0 0 0.06 0.15034 0.13 0.26405 0.3333 0.3678
0.61 0.29355 0.7 0.2571 0.85 0.1991 1 0.14936 1.1 0.1217 /
\setquadratic \plot % OJO f_4
0 0 0.06 0.18879 0.14 0.319877 0.25 0.3678 0.4 0.323
0.55 0.243766 0.7 0.17027 0.9 0.0983 1.1 0.054 /
\put {$\\gets f_1$} [1] at 1.12 0.3661
\put {$\\gets f_2$} [1] at 1.12 0.2437
\put {$\\gets f_3$} [1] at 1.12 0.1217
\put {$\\gets f_4$} [1] at 1.12 0.054
\put {$x$} at 1.25 -0.03
\put {$y$} at 0.03 0.5
\endpicture]
```

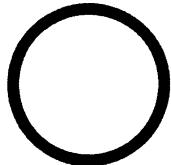
12.13. Grosor de líneas y curvas

El grosor de todas las líneas y curvas es, por defecto, 0.4 pt. Para trazar líneas con los comandos `\setlinear`, `\setquadratic`, `\circulararc` y `\ellipticalarc`, P_TEX utiliza como símbolo de trazado (*plotsymbol*) el punto · de la fuente **cmr5** (romana de 5 puntos). Para lograr líneas de mayor o menor grosor, hay que cambiar el símbolo de trazado con la instrucción:

`\setplotsymbol ({símbolo})`

Cualquier símbolo se puede utilizar con `\setplotsymbol`, tal como se ilustra en el siguiente ejemplo, en el que se usa el símbolo • (\$\bullet\$) para trazar una circunferencia.

Ejemplo



```
\begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -3 to 3, y from -2 to 2
\setplotsymbol {$\bullet$}
\circulararc 360 degrees from 2 0 center at 0 0
\end{picture}
```

Para aumentar ligeramente el grosor de las líneas, lo más práctico es aumentar el tamaño del punto de trazado con los comandos de L^AT_EX, `\large`, `\Large`, etc. A continuación se pueden observar los distintos grosores obtenidos:

Grosor por defecto	→	_____
Grosor con <code>\setplotsymbol (\{\large .\})</code>	→	_____
Grosor con <code>\setplotsymbol (\{\Large .\})</code>	→	_____
Grosor con <code>\setplotsymbol (\{\LARGE .\})</code>	→	_____
Grosor con <code>\setplotsymbol (\{\huge .\})</code>	→	_____
Grosor con <code>\setplotsymbol (\{\Huge .\})</code>	→	_____

También se pueden utilizar puntos o símbolos de otras fuentes cargadas con `\DeclareFixedFont` o `\newfont` (sección 9.6).

El grosor de las líneas (rectas) trazadas con los comandos `\axis`, `\grid`, `\putrectangle`, `\sethistograms` y `\setbars` no se puede cambiar con `\setplotsymbol`. Para dichos comandos se debe utilizar la instrucción `\linethickness=d`, siendo *d* cualquier dimensión T_EX válida. Así por ejemplo, `\linethickness=0.8pt` produce líneas de 0.8 pt de grosor.

12.14. Líneas y curvas punteadas

Las líneas y curvas trazadas con los comandos `\setlinear`, `\setquadratic`, `\circulararc`, `\ellipticalarc`, `\axis`, `\putrectangle`, `\sethistograms` pueden ser ‘punteadas’. Basta preceder estos comandos con la instrucción `\setdots < >`. A continuación, ejemplos del efecto producido.

Instrucción	Trazado obtenido
<code>\setdots</code>
<code>\setdots <8pt></code>
<code>\setdots <7pt></code>
<code>\setdots <4pt></code>
<code>\setdots <3pt></code>
<code>\setdots <2pt></code>
<code>\setdots <1pt></code>

La cantidad que aparece entre `< >` representa la distancia entre los puntos; así, con `\setdots <4pt>` los puntos están separados entre sí 4 pt. Por defecto, `\setdots` asume el valor de 5 pt; por consiguiente, `\setdots` y `\setdots <5pt>` tienen exactamente el mismo efecto. En una misma gráfica pueden aparecer varios `\setdots`, siendo el último especificado el que permanece vigente en lo sucesivo. Si se ha utilizado `\setdots` para cierta porción de la gráfica y se desea retornar al “trazado sólido”, se escribe la instrucción `\setsolid`.

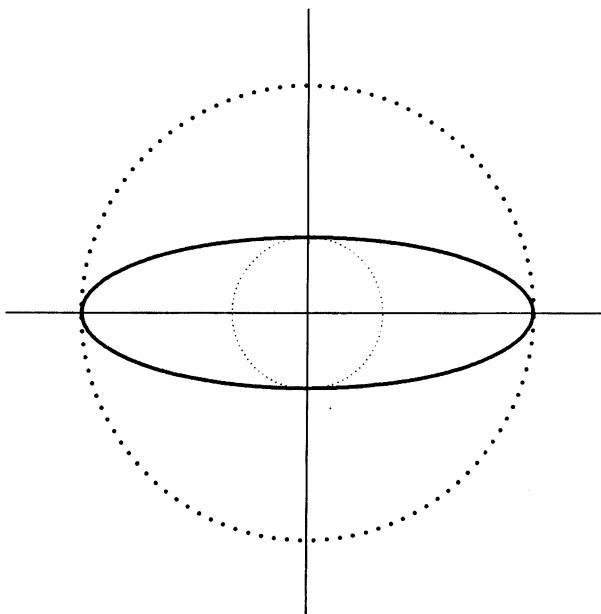
Ejemplo

```
\begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -3 to 3, y from -3 to 3
\setdots
\ellipticalarc axes ratio 3:1 360 degrees
    from 0 1 center at 0 0
\end{picture}
```



```
\begin{picture}
\setcoordinatesystem units <0.5cm,0.5cm>
\setplotarea x from -3 to 3, y from -3 to 3
\setdots <2pt>
\ellipticalarc axes ratio 3:1 360 degrees
    from 0 1 center at 0 0
\end{picture}
```

Ejemplo Los comandos `\setplotsymbol` y `\setdots` se pueden combinar para obtener trazados punteados con puntos de mayor grosor. En este ejemplo, la circunferencia exterior está punteada usando el punto `{\LARGE .}` como `\setplotsymbol`, a la distancia estándar, mientras que la circunferencia interior utiliza el punto estandar a la distancia `\setdots <2pt>`. La ellipse está dibujada con trazo sólido (`\setsolid`) de grosor `{\large .}`.



```
\[
\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -4 to 4, y from -4 to 4
\axis top shiftedto y=0 /
\axis right shiftedto x=0 /
\setdots <2pt>
\circulararc 360 degrees from 1 0 center at 0 0
\setsolid
\setplotsymbol ({\large .})
\ellipticalarc axes ratio 3:1 360 degrees from 0 1 center at 0 0
\setplotsymbol ({\LARGE .})
\setdots
\circulararc 360 degrees from 3 0 center at 0 0
\endpicture
\]
```

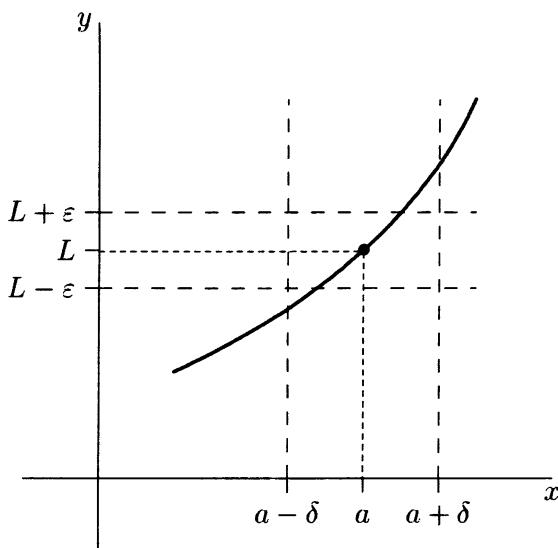
12.15. Líneas y curvas a trozos

El trazado a trozos es completamente similar al trazado punteado, excepto que requiere el uso de `\setdashes` en lugar de `\setdots`. A continuación, ejemplos del efecto producido:

Instrucción	Trazado obtenido
<code>\setdashes</code>	- - - - -
<code>\setdashes <4pt></code>	- - - -
<code>\setdashes <3pt></code>	- - -
<code>\setdashes <2pt></code>	- - - -

`\setdashes` y `\setdashes <5pt>` tienen el mismo efecto.

Ejemplo En este ejemplo utilizamos dos trazados a trozos diferentes: con `\setdashes` y `\setdashes <2pt>`. Nótese el uso de `\setsolid` para retornar al trazado sólido; si no se usa esta instrucción, la curva de la función se obtendrá a trozos, ya que aún está vigente `\setdashes`. El grosor de la curva se ha escogido con `\setplotsymbol` (`\large .`).

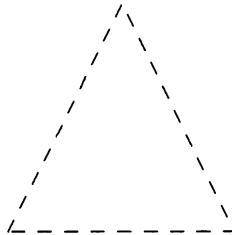


```
\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -1 to 6, y from -1 to 6
\axis bottom shiftedto y=0 ticks withvalues {$a-\delta$} {$a$}
{$a+\delta$} / at 2.5 3.5 4.5 /
\axis left shiftedto x=0 ticks withvalues {$L-\varepsilon$} {$L$}
{$L+\varepsilon$} / at 2.5 3 3.5 /
\setdashes
\setlinear \plot 0 2.5 5 2.5 /

```

```
\setlinear \plot 0 3.5 5 3.5 /
\setlinear \plot 2.5 0 2.5 5 /
\setlinear \plot 4.5 0 4.5 5 /
\setdashes <2pt>
\setlinear \plot 0 3 3.5 3 3.5 0 /
\setsolid
\setplotsymbol ({\large .})
\setquadratic \plot 1 1.4 3.5 3 5 5 /
\put {$\bullet$} at 3.5 3
\put {$x$} at 6 -0.2
\put {$y$} at -0.2 6
\endpicture]
```

La mayor limitación de `\setdots` y `\setdashes` es que los segmentos o puntos iniciales y finales de un arco de línea o una curva pueden quedar incompletos. Por ejemplo, al dibujar a trozos el triángulo cuyos vértices son (0, 0), (1.5, 3) y (3, 0), obtenemos:



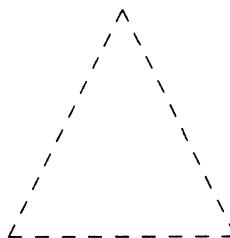
```
\[\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from 0 to 3, y from 0 to 3
\setdashes
\setlinear \plot 0 0 1.5 3 3 0 0 0 /
\endpicture]
```

Este resultado no es del todo aceptable por la apariencia de las intersecciones en los vértices. P_IC_TE_X proporciona un medio de corregir el trazado de curvas, de tal manera que todos los arcos y líneas comiencen y terminen con segmentos o puntos completos, manteniendo un espaciamiento uniforme entre ellos. Las instrucciones necesarias son:

<code>\setdotsnear <separación> for <longitud de la línea o arco></code>
<code>\setdashesnear <separación> for <longitud de la línea o arco></code>

donde la *separación* es la distancia deseada entre puntos (o segmentos). Se requiere la longitud exacta de la línea o arco. En el ejemplo anterior, las distancias entre los vértices del triángulo pueden ser calculadas fácilmente: 3 cm, 3.351 cm y 3.351 cm. Tenemos así la versión mejorada del

triángulo anterior (para cada lado del triángulo se requiere la construcción `\setdashesnear`):



```
\f[\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from 0 to 3, y from 0 to 3
\setdashesnear <5pt> for <3cm>
\setlinear \plot 0 0 3 0 /
\setdashesnear <5pt> for <3.351cm>
\setlinear \plot 0 0 1.5 3 /
\setdashesnear <5pt> for <3.351cm>
\setlinear \plot 1.5 3 3 0 /
\endpicture]
```

En el ejemplo anterior fue fácil determinar las longitudes de los lados del triángulo, necesarias para usar `\setdashesnear`, pero incluso en el caso de curvas trazadas con `\setquadratic`, \LaTeX puede encontrar la longitud deseada por medio de la instrucción

`\findlength {\setquadratic \plot ... /}`

y la almacena en el registro denominado `\totalarcLength`.

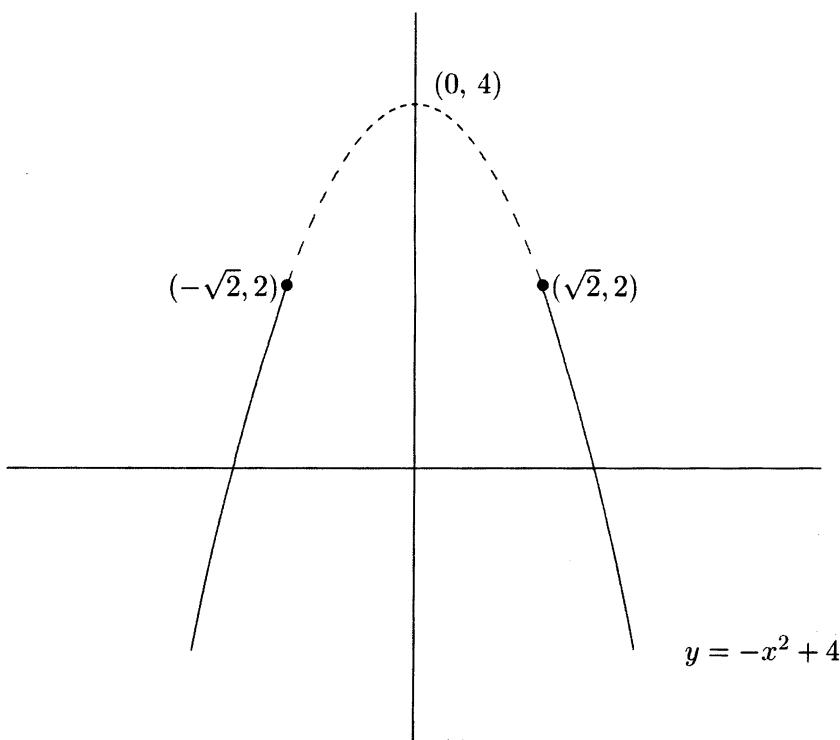
Ejemplo En la gráfica de la página siguiente, trazamos a trozos el arco que une los puntos $(-\sqrt{2}, 2)$ y $(\sqrt{2}, 2)$ de la parábola $y = -x^2 + 4$. Para dibujar este arco, consideramos los tres puntos $(-\sqrt{2}, 2)$, $(0, 4)$ y $(\sqrt{2}, 2)$, y hallamos su longitud con

`\findlength {\setquadratic \plot -1.4142 2 0 4 1.4142 2 /},`

valor que \LaTeX almacena como `\totalarcLength`. Finalmente, graficamos el arco, con un espaciamiento de 4 pt entre segmentos, por medio de

`\setdashesnear <4pt> for <\totalarcLength>.`

Se ha completado la gráfica con el trazado de los arcos sólidos de la parábola: el que une los puntos $(\sqrt{6}, -2)$, $(2, 0)$ y $(\sqrt{2}, 2)$, y el arco simétrico.



```
\[
\begin{picture}
\setcoordinatesystem units <1.2cm,1.2cm>
\setplotarea x from -4.5 to 4.5, y from -3 to 5
\axis bottom shiftedto y=0 /
\axis left shiftedto x=0 /
\findlength {\setquadratic \plot -1.4142 2 0 4 1.4142 2 /}
\setdashesnear <4pt> for <\totalarclength>
\setquadratic \plot -1.4142 2 0 4 1.4142 2 /
\setsolid
\setquadratic \plot -2.449489 -2 -2 0 -1.4142 2 /
\setquadratic \plot 1.4142 2 2 0 2.449489 -2 /
\put {$\bullet$} at 1.4142 2
\put {$\bullet$} at -1.4142 2
\put {$(-\sqrt{2}, 2)$} [r] at -1.5 2
\put {$(\sqrt{2}, 2)$} [l] at 1.5 2
\put {(0, 4)} [l] at 0.2 4.2
\put {$y=-x^2+4$} [l] at 3 -2
\endpicture
\]
```

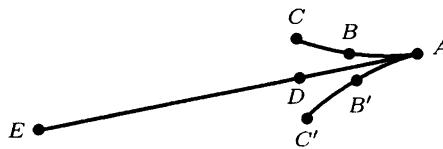
12.16. Flechas

El comando `\arrow` de `PICTEX` para trazar flechas es muy flexible; permite incluso diseñar el tamaño y la forma de la punta de la flecha.

La instrucción

```
\arrow <l> [c,d] from x0 y0 to x1 y1
```

traza una flecha de la forma



donde

$$E = (x_0, y_0)$$

$$A = (x_1, y_1)$$

ℓ es la distancia entre A y D

$c\ell$ es la distancia entre B y B'

$d\ell$ es la distancia entre C y C'

La parte `\arrow <l> [c,d]` del comando define propiamente la cabeza de la flecha. El lector puede comparar diferentes flechas, obtenidas al cambiar los parámetros ℓ , c y d :

Parámetros	Tipo de flecha producido
<code>\arrow <6pt> [0.2, 0.6]</code>	
<code>\arrow <6pt> [0.2, 0.5]</code>	
<code>\arrow <4pt> [0.2, 0.5]</code>	
<code>\arrow <7pt> [0.1, 0.3]</code>	

Para la mayoría de las necesidades basta una sola flecha que podamos rotar y extender a nuestro antojo. Para esto, es conveniente definir, por medio del comando `\newcommand` de `LATEX`, una flecha genérica, a la que

podemos denominar `\flecha`, y hacer uso de ella en todas las situaciones. Por ejemplo, la declaración

```
\newcommand{\flecha}{\arrow <6pt> [0.2, 0.6]}
```

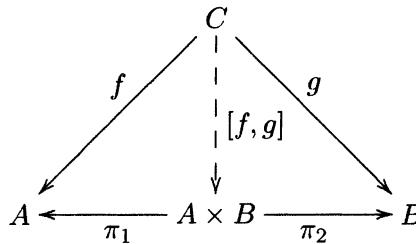
define el primer tipo de flecha del diagrama mostrado en la página anterior. Con esta definición, la instrucción

```
\flecha from  $x_0$   $y_0$  to  $x_1$   $y_1$ 
```

equivale a

```
\arrow <6pt> [0.2, 0.6] from  $x_0$   $y_0$  to  $x_1$   $y_1$ .
```

Ejemplo En este ejemplo usamos el comando `\flecha`, definido arriba, para dibujar un diagrama conmutativo.



```
\[
\begin{picture}
\setcoordinatesystem units <1.3cm,1.3cm>
\setplotarea x from -2 to 2, y from -0.2 to 2
\put {$A$} at -2 0
\put {$B$} at 2 0
\put {$C$} at 0 2
\put {$A \times B$} at 0 0
\flecha from -0.5 0 to -1.8 0
\flecha from 0.5 0 to 1.8 0
\flecha from 0.2 1.8 to 1.8 0.2
\flecha from -0.2 1.8 to -1.8 0.2
\setdashes <6pt>
\flecha from 0 1.8 to 0 0.2
\put {$[f, g]$} [1] at 0.1 0.9
\put {$\pi_1$} at -1 -0.2
\put {$\pi_2$} at 1 -0.2
\put {$g$} at 1 1.3
\put {$f$} at -1 1.3
\endpicture
\]

```

12.17. Rotaciones

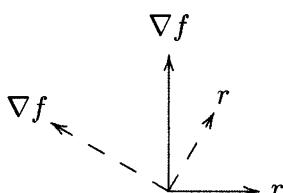
PiCTEX permite rotar algunos objetos gráficos, referidos al sistema coordenado vigente. Para rotar un determinado *objeto*, un ángulo θ alrededor de un punto (x_0, y_0) , se utiliza la instrucción

```
\startrotation by coseno de  $\theta$  seno de  $\theta$  about  $x_0$   $y_0$ 
  objeto
\stoprotation
```

donde

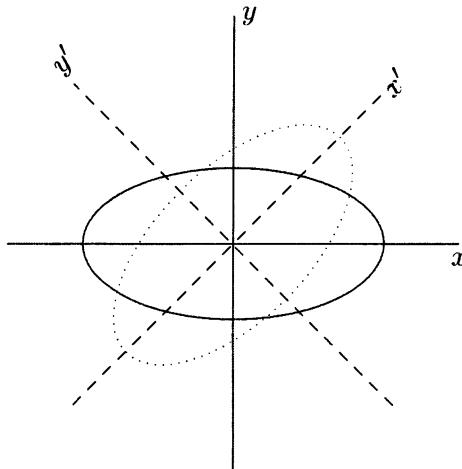
1. $-360 \leq \theta \leq 360$. Si θ es positivo, la rotación se hace en el sentido anti-horario; en sentido horario si θ es negativo.
2. Los comandos que definen el *objeto* a ser rotado y el punto (x_0, y_0) se refieren al sistema coordenado vigente.
3. Comandos que pueden ser rotados: `\setlinear`, `\setquadratic`, `\circulararc`, `\ellipticalarc`, `\arrow`. Los comandos `\put {...}` y `\multiput {...}` también pueden ser rotados pero el material que aparece en el argumento `{...}` conserva su orientación original; en particular, el texto corriente y las fórmulas no se pueden rotar.
4. Comandos que no pueden ser rotados: `\axis`, `\putrectangle`.

Ejemplo En este ejemplo se ha rotado 60° la figura consistente en los dos vectores perpendiculares denotados por ∇f y r (con trazado continuo). Obsérvese que los argumentos de `\put` (r y ∇f) se rotan con la figura básica, pero mantienen su orientación original. Se ha usado $\cos(60^\circ) = 0.5$ y $\sin(60^\circ) = 0.866025$.



```
\begin{picture}
\setcoordinatesystem units <1.2cm,1.2cm>
\setplotarea x from -1.5 to 1.5, y from
0 to 1.8
\arrow <6pt> [0.2, 0.6] from 0 0 to 1 0
\arrow <6pt> [0.2, 0.6] from 0 0 to 0 1.5
\put {$r$} at 1.2 0
\put {$\nabla f$} at 0 1.8
\startrotation by 0.5 0.866025 about 0 0
\setdashes <7pt>
\arrow <6pt> [0.2, 0.6] from 0 0 to 1 0
\arrow <6pt> [0.2, 0.6] from 0 0 to 0 1.5
\put {$r$} at 1.2 0
\put {$\nabla f$} at 0 1.8
\stoprotation
\endpicture
```

Ejemplo En este ejemplo se ha rotado la elipse junto con los ejes coordenados. Sin embargo, puesto que el comando `\axis` no es afectado por la rotación, los ejes a ser rotados deben ser definidos como líneas rectas; tal es el propósito de las instrucciones `\setlinear \plot -3 0 3 0 /` y `\setlinear \plot 0 -3 0 3 /`, las cuales están precedidas del comando `\setdashes <4pt>`. La elipse rotada se trazó con `\setdots <3pt>`. El ángulo de rotación es 45° y $\cos 45^\circ = \sin 45^\circ = 0.707106$. Puesto que `PTeX` no puede rotar texto corriente ni fórmulas, se ha usado el comando `\rotatebox` de los paquetes `graphics` y `graphicx` (sección 7.3.3) para rotar 45° las etiquetas x' y y' .



```
\[ \begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from -3 to 3, y from -3 to 3
\axis left shiftedto x=0 /
\axis bottom shiftedto y=0 /
\ellipticalarc axes ratio 2:1 360 degrees from 2 0 center at 0 0
\put {$x$} at 3 -0.2
\put {$y$} at 0.2 3
\startrotation by 0.707106 0.707106 about 0 0
\setdashes <4pt>
\setlinear \plot -3 0 3 0 /
\setlinear \plot 0 -3 0 3 /
\setdots <3pt>
\ellipticalarc axes ratio 2:1 360 degrees from 2 0 center at 0 0
\put {\rotatebox{45}{$x'$}} at 3 0
\put {\rotatebox{45}{$y'$}} at 0.1 3.3
\stoprotation
\endpicture \]
```

12.18 Gráficos de barras

La instrucción

```
\setbars breadth <a> baseline at y = y0
\plot x1 y1  x2 y2  x3 y3 ... /
```

traza un gráfico de barras verticales, cada una de anchura a y con base en $y = y_0$. Las barras están colocadas en las coordenadas horizontales x_1, x_2, x_3, \dots , y se extienden verticalmente hasta y_1, y_2, y_3, \dots . Similarmente, la instrucción

```
\setbars breadth <a> baseline at x = x0
\plot x1 y1  x2 y2  x3 y3 ... /
```

traza un gráfico de barras horizontales, cada una de anchura a y con base en $x = x_0$. Las barras están colocadas en las coordenadas verticales y_1, y_2, y_3, \dots , y se extienden horizontalmente hasta x_1, x_2, x_3, \dots . En ambos casos, el grosor de los bordes de las barras es, por defecto, 4 pt pero se pueden cambiar con \linethickness .

Opcionalmente, se pueden colocar etiquetas en los extremos iniciales (o bases) de las barras escribiendo

baselabels ([m] < x_t, y_t >)

inmediatamente antes de \plot . El modificador de posición $[m]$, que es opcional, puede asumir los mismos valores $[r], [t], [rb]$, etc del comando \put (sección 12.7). La parte $<x_t, y_t>$ se usa para trasladar la etiqueta horizontal o verticalmente, distancias x_t y y_t , respectivamente. Las etiquetas mismas se escriben entre comillas¹, en la forma "*etiqueta*", después de las respectivas coordenadas indicadas en la lista \plot . De manera análoga, se pueden colocar etiquetas en los extremos de las barras (extremos opuestos a las bases), escribiendo

endlabes ([m] < x_t, y_t >)

antes de \plot (pero después de **baselabes**, si éste último se usa).

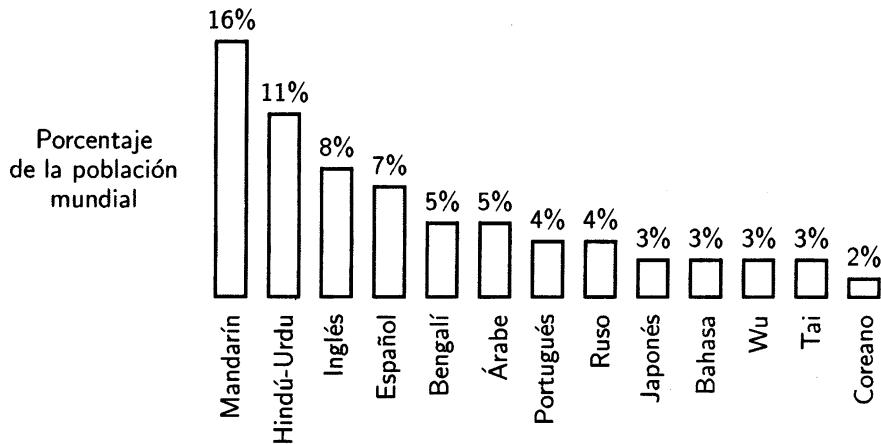
Ejemplo

F		$\begin{array}{l} \begin{picture} \\ \setcoordinatesystem units <0.5cm,0.5cm> \\ \setplotarea x from -1 to 5, y from 0 to 6 \\ \linethickness=5pt \\ \setbars breadth <0pt> baseline at x = -1 \\ baselabels (<-3mm,0pt>) \\ \plot 2 0 "A" 1 1 "B" 4 2 "C" 0.5 3 "D" \\ 1.5 4 "E" 5 5 "F" / \\ \end{picture} \end{array}$
E		
D		
C		
B		
A		

¹Hay que advertir que el uso del símbolo " ocasiona conflictos con el paquete babel.

Ejemplo Como se vió en la sección anterior, el texto corriente no se puede rotar con comandos `\text{P} \text{T} \text{E} \text{X}`, pero podemos usar la instrucción `\rotatebox` de los paquetes `graphics` o `graphicx` (sección 7.3.3) para hacer rotaciones como las que aparecen en este ejemplo.

Los idiomas más hablados del mundo



```
\small \sfamily [\begin{picture}
\setcoordinatesystem units <1.4cm,1.2cm>
\setplotarea x from -1 to 7, y from 1 to 6.4
\put {Los idiomas más hablados del mundo} [1] at 2 6
\put {\stack {Porcentaje,de la población,mundial}} at -0.8 3.6
\put {\rotatebox[origin=rt]{90}{Mandarín}} [t] at 0.5 2
\put {\rotatebox[origin=rt]{90}{Hindú-Urdu}} [t] at 1 2
\put {\rotatebox[origin=rt]{90}{Inglés}} [t] at 1.5 2
\put {\rotatebox[origin=rt]{90}{Español}} [t] at 2 2
\put {\rotatebox[origin=rt]{90}{Bengalí}} [t] at 2.5 2
\put {\rotatebox[origin=rt]{90}{\'Arabe}} [t] at 3 2
\put {\rotatebox[origin=rt]{90}{Portugués}} [t] at 3.5 2
\put {\rotatebox[origin=rt]{90}{Ruso}} [t] at 4 2
\put {\rotatebox[origin=rt]{90}{Japonés}} [t] at 4.5 2
\put {\rotatebox[origin=rt]{90}{Bahasa}} [t] at 5 2
\put {\rotatebox[origin=rt]{90}{Wu}} [t] at 5.5 2
\put {\rotatebox[origin=rt]{90}{Tai}} [t] at 6 2
\put {\rotatebox[origin=rt]{90}{Coreano}} [t] at 6.5 2
\linethickness=1pt
\setbars breadth <4mm> baseline at y = 2.2 endlabels (<0pt,3mm>
\plot 0.5 5 "16\%" 1 4.2 "11\%" 1.5 3.6 "8\%" 2 3.4 "7\%" 2.5 3 "5\%" 3 3 "5\%" 3.5 2.8 "4\%" 4 2.8 "4\%" 4.5 2.6 "3\%" 5 2.6 "3\%" 5.5 2.6 "3\%" 6 2.6 "3\%" 6.5 2.4 "2\%" /
\endpicture]
```

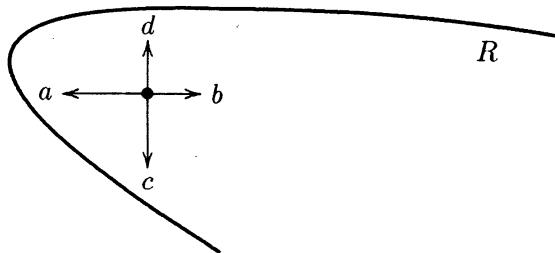
12.19. Sombreado de regiones

\TeX sombra una región R colocando un *símbolo de sombreado* (generalmente un punto) en cada vértice de una *grilla de sombreado* contenida en R . Se usa el comando `\setshadesymbol` para escoger el símbolo de sombreado y con `\setshadegrid` se define la grilla de sombreado. El sombreado propiamente dicho se realiza con `\vshade` (sombreado en modo vertical) o `\hshade` (sombreado en modo horizontal), en conjunción con los comandos `\setlinear` y `\setquadratic`.

Símbolo de sombreado. Se especifica con

`\setshadesymbol <a, b, c, d> ({símbolo})`

Este *símbolo* es típicamente un carácter individual, aunque podría ser un objeto más complejo. Por defecto, \TeX utiliza el punto \cdot de la fuente **cmr5** (romana de 5 puntos). La parte $<a, b, c, d>$ es opcional y \TeX la usa para evitar que el símbolo de sombreado sobrepase la región sombreada R . Más concretamente, \TeX no colocará el símbolo en la región R , a menos que las distancias del símbolo a la frontera de R , hacia la izquierda, la derecha, abajo y arriba, sean por lo menos a, b, c y d , respectivamente:



Si alguno de los parámetros a, b, c y d se omite, \TeX escogerá valores por defecto, según el tamaño del símbolo sombreado. Los valores a, b, c y d deben ser dimensiones \TeX , no necesariamente positivas; `0pt` se puede escribir simplemente `z`. Por ejemplo, con

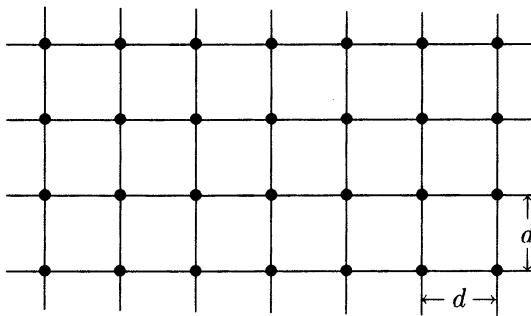
`\setshadesymbol <1pt, ,z,0.5pt> ({\small z})`

el símbolo de sombreado es el punto de tamaño `\small`, en la fuente vigente, con los siguientes parámetros: $a = 1\text{pt}$, $b = \text{valor por defecto}$, $c = 0\text{pt}$, $d = 0.5\text{pt}$.

Grilla o cuadrícula de sombreado. Se especifica con

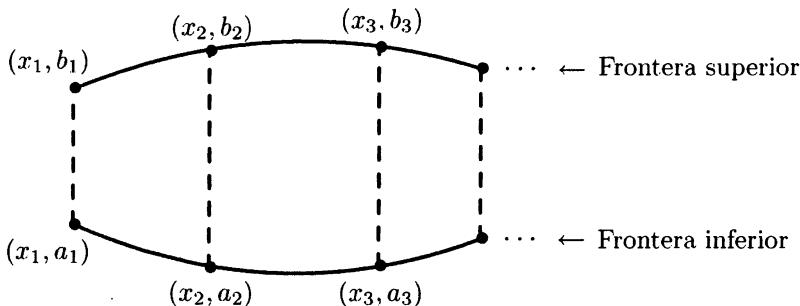
`\setshadegrid span <d>`

donde d es la distancia deseada entre los distintos puntos de la grilla:



El parámetro `\setshadegrid span <d>` es opcional; al omitirlo, **PiCTeX** utiliza una grilla con 5pt de separación entre sus vértices.

Sombreado en modo vertical. Se usa este tipo de sombreado para regiones encerradas en una franja vertical y que tengan una frontera inferior y una superior claramente delimitadas:



El sombreado de una región de este tipo se hace por medio de

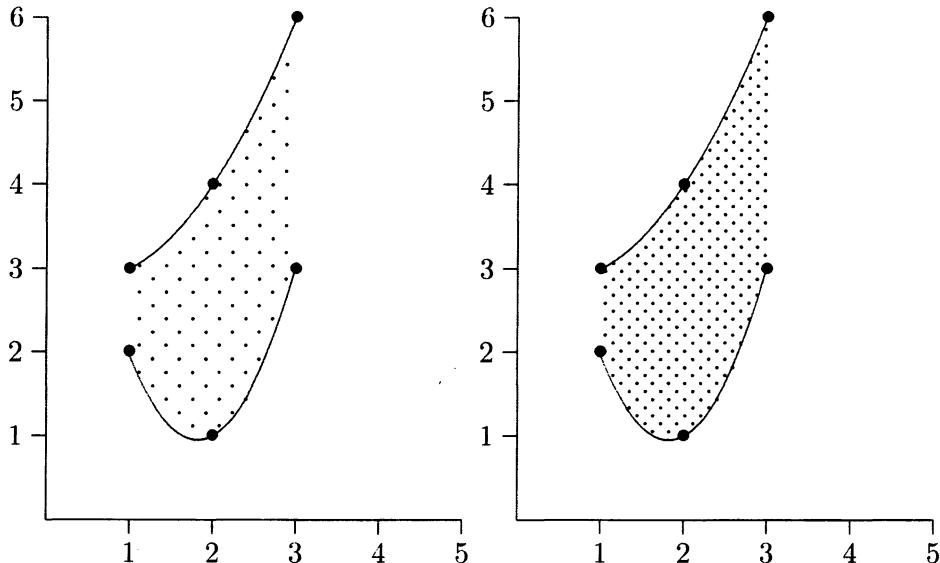
```
\setquadratic
\vshade x1 a1 b1  x2 a2 b2  x3 a3 b3  ... /
```

Hay que tener presente que, al usar `\setquadratic`, el número de triples utilizadas con `\vshade ... /` debe ser impar. Esta restricción resulta molesta y el usuario puede optar por realizar el sombreado con

```
\setlinear
\vshade x1 a1 b1  x2 a2 b2  x3 a3 b3  ... /
```

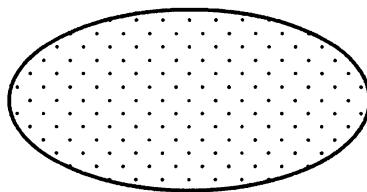
sin restricción alguna sobre el número de puntos. El uso de `\setlinear` es recomendable si las fronteras superior e inferior de la región son casi lineales o si se utilizan numerosos puntos para la interpolación `\vshade ... /`.

Ejemplo Las dos gráficas siguientes coinciden, excepto por la grilla de sombreado. A la izquierda se ha usado `\setshadegrid span <5pt>` y a la derecha `\setshadegrid span <3pt>`. En ambos casos, se ha usado el parámetro `<a, b, c, d>` con los valores `<z, z, 0.01pt, 0.01pt>`. Para sombreado vertical es usualmente conveniente escoger $a = b = z$, ya que la región está delimitada por rectas verticales a izquierda y a derecha.



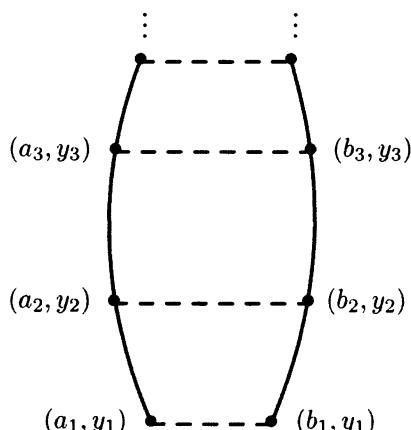
```
\begin{picture}
\setcoordinatesystem units <1.1cm,1.1cm>
\setplotarea x from 0 to 5, y from 0 to 6
\axis bottom shiftedto y=0
ticks numbered from 1 to 5 by 1 /
\axis left shiftedto x=0
ticks numbered from 1 to 6 by 1 /
\setquadratic \plot 1 3 2 4 3 6 /
\setquadratic \plot 1 2 2 1 3 3 /
\multiput {$\bullet$} at 1 3 2 4 3 6 1 2 2 1 3 3 /
\setshadesymbol <z,z,0.01pt,0.01pt> ({.})
\setshadegrid span <5pt> %% PARA LA GRAFICA DE LA IZQUIERDA
\setshadegrid span <3pt> %% PARA LA GRAFICA DE LA DERECHA
\setquadratic
\vshade 1 2 3 2 1 4 3 3 6 /
\endpicture
```

Ejemplo En este ejemplo se ha omitido el comando `\setshadegrid` para la grilla de sombreado; **PlCTEX** utiliza entonces los valores establecidos por defecto. Obsérvese que el número de triples empleadas con `\vshade ... /` es impar, requisito indispensable al usar la interpolación cuadrática `\setquadratic`.



```
\[ \begin{picture}
\setcoordinatesystem units <0.8cm,0.8cm>
\setplotarea x from -4 to 4, y from -2 to 2
\setplotsymbol({\large .})
\ellipticalarc axes ratio 2:1 360 degrees from 3 0 center at 0 0
\setshadesymbol <z,z,0.2pt,0.2pt> ({.})
\setquadratic
\vshade -3 -0.3 0.3   -2.5 -0.8 0.8   -2 -1.1 1.1   -1.5 -1.3 1.3
-0.5 -1.4 1.4   0 -1.5 1.5   0.5 -1.4 1.4
1.5 -1.3 1.3   2 -1.1 1.1   2.5 -0.8 0.8   3 -0.3 0.3 /
\endpicture \]
```

Sombreado en modo horizontal. Se usa este tipo de sombreado para regiones encerradas en una franja horizontal y que tengan fronteras a la izquierda y a la derecha claramente delimitadas:



El sombreado de una región de este tipo se hace por medio de

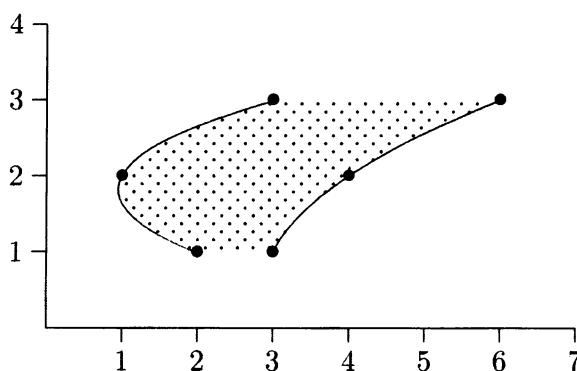
```
\setquadratic
\hshade y1 a1 b1 y2 a2 b2 y3 a3 b3 ... /
```

con la restricción de que el número de triples utilizadas para `\hshade` debe ser impar. Si las fronteras izquierda y derecha de la región son casi lineales o si se utilizan numerosos puntos para `\hshade ... /`, el sombreado también se puede realizar con

```
\setlinear
\hshade y1 a1 b1 y2 a2 b2 y3 a3 b3 ... /
```

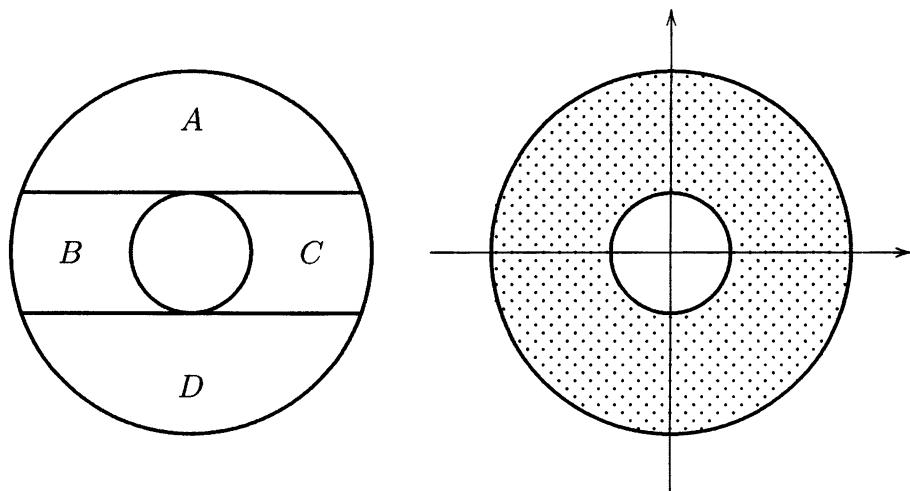
sin restricción alguna sobre el número de puntos.

Ejemplo



```
\begin{picture}
\setcoordinatesystem units <1cm,1cm>
\setplotarea x from 0 to 7, y from 0 to 4
\axis bottom shiftedto y=0
ticks numbered from 1 to 7 by 1 /
\axis left shiftedto x=0
ticks numbered from 1 to 4 by 1 /
\setquadratic \plot 3 1 4 2 6 3 /
\setquadratic \plot 2 1 1 2 3 3 /
\multiput {$\bullet$} at 3 1 4 2 6 3 2 1 1 2 3 3 /
\setshadesymbol <0.01pt,0.01pt,z,z> (\small .)
\setshadegrid span <3pt>
\setquadratic
\hshade 1 2 3 2 1 4 3 3 6 /
\endpicture]
```

Ejemplo Para sombrear la corona circular (a la derecha), se ha dividido el área total en las cuatro regiones mostradas a la izquierda: A, B, C y D. Cada una de ellas fue sombreada por separado, en modo horizontal, usando interpolación lineal \setlinear.



Corona sombreada a la derecha:

```
\begin{picture}
\setcoordinatesystem units <0.8cm,0.8cm>
\setplotarea x from -4 to 4, y from -4 to 4
\axis bottom shiftedto y=0 /
\axis left shiftedto x=0 /
\arrow <6pt> [.2,.6] from 0 3.7 to 0 4
\arrow <6pt> [.2,.6] from 3.7 0 to 4 0
\setplotsymbol({\large .})
\circulararc 360 degrees from 3 0 center at 0 0
\circulararc 360 degrees from 1 0 center at 0 0
\setshadesymbol <z,z,z,z> ({\small .})
\setshadegrid span <3pt>
\setlinear
\hshade %% SOMBREA LA REGION A
1 -2.7 2.7 2.2 -2.2 2.2 3 -0.3 0.3 /
\hshade %% SOMBREA LA REGION D
-3 -0.3 0.3 -2.2 -2.2 2.2 -1 -2.7 2.7 /
\hshade %% SOMBREA LA REGION B
-1 -2.9 -0.4 -0.3 -2.9 -1.3 0 -3 -1 0.3 -2.9 -1.3
1 -2.9 -0.4 /
\hshade %% SOMBREA LA REGION C
-1 0.4 2.9 -0.3 1.3 2.9 0 1 3 0.3 1.3 2.9 1 0.4 2.9 /
\endpicture
```



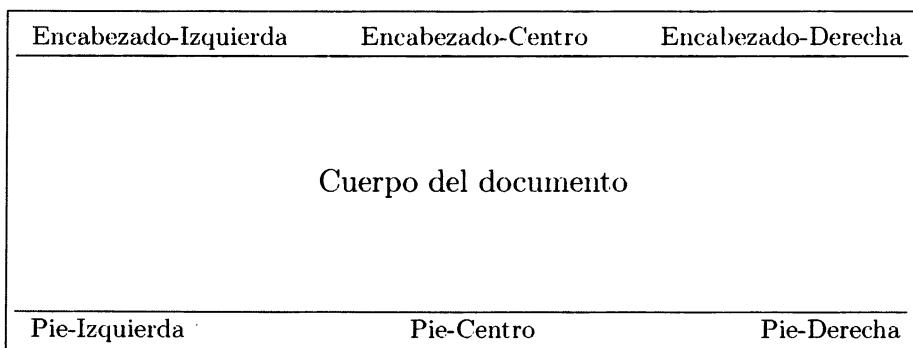
Otros paquetes importantes

En este capítulo describiremos una serie de paquetes adicionales, entre los que se encuentran algunos muy útiles y que gozan de gran popularidad. Los paquetes que no hacen parte de la colección estándar de L^AT_EX 2 _{ε} se incluyen en el CD adjunto, en las carpetas señaladas. En el apéndice C se encuentran indicaciones generales sobre instalación de paquetes.

13.1. El paquete `fancyhdr`

 Los archivos que conforman el paquete `fancyhdr` se encuentran en la carpeta `/Paquetes/fancyhdr` del CD adjunto.

Los formatos de página de L^AT_EX, `empty`, `plain`, `headings` y `myheadings` (sección 2.10) son útiles y cómodos pero son bastante rígidos. Con el paquete `fancyhdr`¹ el usuario puede controlar completamente el contenido y la apariencia de los encabezados y de los pies de página e incluir en ellos líneas decorativas. Para `fancyhdr`, el formato de una página tiene el siguiente aspecto:



¹ Escrito por Piet van Oostrum. Originalmente este paquete se denominaba `fancyheadings`; la nueva versión data de octubre de 2000 y tiene una sintaxis más sencilla y natural que la de su antecesor.

El contenido de los encabezados se controla con `\fancyhead[...]{...}` y el de los pies de página con `\fancyfoot[...]{...}`. En el argumento entre paréntesis angulares [...] se especifica para cuáles páginas y para cuáles partes de la página se aplica el contenido escrito en el argumento {...}; puede tomar combinaciones de los siguientes parámetros:

E	Página par	(Even)
O	Página impar	(Odd)
L	Izquierda	(Left)
C	Centro	(Center)
R	Derecha	(Right)

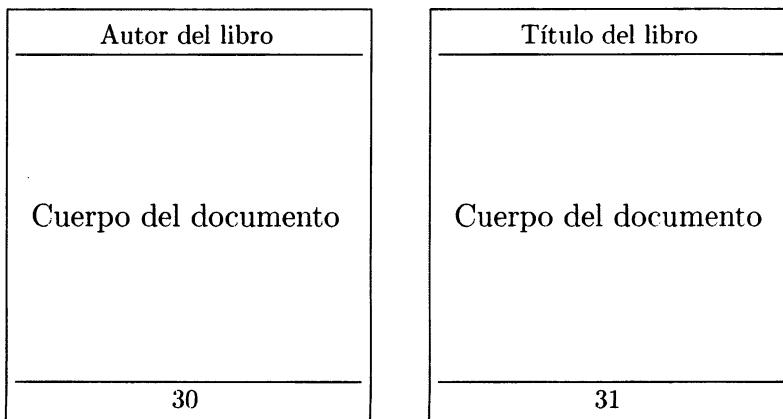
Los comandos `\headrulewidth` y `\footrulewidth` controlan el grosor de las líneas decorativas del encabezado y del pie de página, respectivamente. Sus valores se cambian con `\renewcommand`.

Para usar `fancyhdr` hay que escribir en el preámbulo del documento las instrucciones

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

y, después de `\begin{document}`, el formato deseado, con los comandos `\fancyhead` y `\fancyfoot` ya mencionados.

Ejemplo Considérese el formato de página ilustrado a continuación, para un documento escrito con el estilo `book`, opción `twoside`. A la izquierda aparece una página con numeración par, y a la derecha una que tiene numeración impar.



El anterior formato se puede obtener escribiendo las siguientes instrucciones después de `\begin{document}`:

```
\fancyhead{}% Para eliminar definiciones previas
\fancyhead[CE]{Autor del libro}
\fancyhead[CO]{Título del libro}
\fancyfoot[CE,CO]{\thepage}
\renewcommand{\headrulewidth}{0.6pt}
\renewcommand{\footrulewidth}{0.6pt}
```

El comando `LATEX \thepage` produce el número de la página actual.

Es útil saber que `LATEX` almacena el contenido de los encabezados con los comandos `\leftmark` (para las páginas a mano izquierda) y `\rightmark` (para las páginas a mano derecha). Estos contenidos se actualizan al cambiar de capítulo o de sección. Por defecto, `LATEX` escribe los encabezados en letras mayúsculas, pero si el usuario prefiere la grafía normal puede usar en el argumento de `\fancyhead` o de `\fancyfoot` la instrucción `\nouppercase` (este comando pertenece al paquete `fancyhdr`, no a `LATEX`). También se pueden incluir allí comandos para cambiar el tipo y tamaño de la letra.

Ejemplo En el presente libro hemos utilizado el paquete `fancyhdr` para los formatos de página. Hemos mantenido el contenido normal de los encabezados, cambiando solamente su presentación: centrados, en el tamaño de letra `\scriptsize` y con una línea horizontal decorativa. He aquí el formato utilizado:

```
\fancyhead{}% Para eliminar definiciones previas
\fancyhead[CE]{\scriptsize \leftmark}
\fancyhead[CO]{\scriptsize \rightmark}
\fancyhead[LE,RO]{\thepage}
\fancyfoot{}
\renewcommand{\headrulewidth}{0.9pt}
```

En los dos ejemplos anteriores, la primera página de cada capítulo no se ve afectada por el formato definido porque `LATEX` automáticamente emite el comando `\thispagestyle{plain}` cuando recibe la orden `\chapter`. Pero con `fancyhdr` se puede modificar incluso la apariencia de la primera página de cada capítulo, redefiniendo el estilo `plain`, en la siguiente forma:

```
\fancypagestyle{plain}{%
\fancyhf{}% Para "blanquear" los campos
\fancyhead[...]{...}
\fancyfoot[...]{...}
\renewcommand{\headrulewidth}{...}
\renewcommand{\footrulewidth}{...}}
```

Estas instrucciones se escriben también después de `\begin{document}` y antes del primer uso de `\chapter`.

13.2. El paquete `multicol`

`LATEX` tiene la opción `twocolumn` para escribir documentos a dos columnas (sección 2.14). Además, los comandos `\twocolumn` y `\onecolumn` permiten pasar al formato de dos columnas y al de una columna, respectivamente, tantas veces como se quiera. Pero con dichos comandos se inicia siempre una nueva página, lo que reduce su utilidad.

El paquete `multicol`², al cual se accede con `\usepackage{multicol}`, maneja con gran flexibilidad los formatos en varias columnas. Este paquete tiene definido el entorno `multicols`, que se usa en la siguiente forma:

```
\begin{multicols}{n}
  Texto
\end{multicols}
```

siendo n el número de columnas deseado, $1 \leq n \leq 10$. En el documento, incluso en una misma página, se pueden usar varios entornos `multicols`, lo que permite pasar del formato de múltiples columnas al de una columna, o viceversa, en cualquier lugar del documento. Las columnas se pueden separar con líneas verticales, si así se desea, asignándole un valor positivo al parámetro `\columnseprule`. Por ejemplo, al escribir

```
\renewcommand{\columnseprule}{0.4pt}
```

en el preámbulo o después de `\begin{document}`, las columnas aparecen separadas por líneas verticales de grosor 0.4pt.

Hay que advertir que las notas al pie de página aparecen a lo ancho de la parte inferior de la página y no debajo de columnas individuales.

Una característica notable del paquete `multicol` es que, en la última página del documento, las columnas aparecen balanceadas, es decir, con aproximadamente el mismo número de renglones en cada una. Con la opción `twocolumn` o con el comando `\twocolumn` de `LATEX`, esto no sucede. No obstante, si el usuario prefiere columnas no-balanceadas en la última página (es decir, distribución normal del texto) debe usar el entorno `multicols*` en lugar de `multicols`.

La gran limitación de `multicol` es el manejo de los objetos flotantes. Bajo el alcance del entorno `multicols`, los entornos `table` y `figure` (sección 7.4) están inhabilitados; en su lugar hay que usar las versiones estrella `table*` y `figure*` (mencionadas en la sección 7.4), con las cuales los objetos flotantes aparecerán siempre en la página siguiente, a lo ancho de la hoja, y no en columnas individuales. Bajo el alcance de `multicols` tampoco se permite el comando `\marginpar` para notas marginales.

²Paquete estándar de `LATEX 2 ϵ` , escrito por Frank Mittelbach.

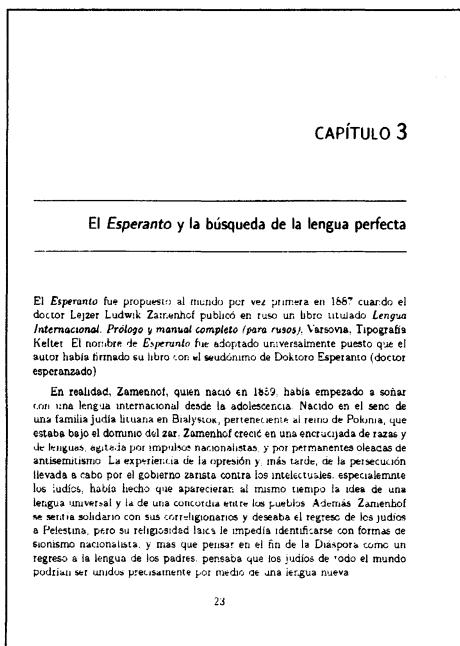
13.3. El paquete `fncychap`

 Los archivos que conforman el paquete `fncychap` se encuentran en la carpeta `/Paquetes/fncychap` del CD adjunto.

El paquete `fncychap`³ permite cambiar la presentación de la primera página de los capítulos, para documentos en los estilos `book` y `report`. Hay seis estilos predefinidos para los capítulos, a saber: `Sonny`, `Lenny`, `Glenn`, `Conny`, `Rejne` y `Bjarne` (corresponden a nombres suecos de personas). Se invoca un *estilo* determinado por medio de

```
\usepackage[estilo]{fncychap}
```

Al escribir `\usepackage[Glenn]{fncychap}`, por ejemplo, la presentación de los capítulos tendrá el estilo `Glenn`. En las Figuras 13.1 a 13.6 se puede apreciar el aspecto de los seis estilos de `fncychap`. Hay que anotar que el estilo `Bjarne` (Figura 13.6) solamente es apropiado para documentos en inglés porque los capítulos se enumeran alfabéticamente: ONE, TWO, THREE, etc. No existen comandos de alto nivel para modificar esta numeración.



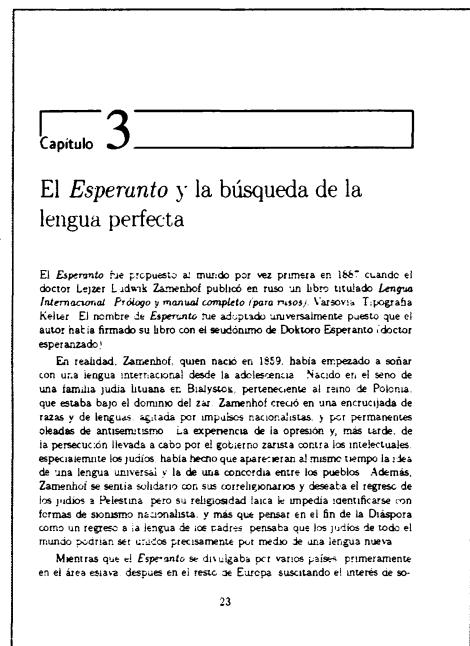
CAPÍTULO 3

El *Esperanto* y la búsqueda de la lengua perfecta

El *Esperanto* fue propuesto al mundo por vez primera en 1857 cuando el doctor Lejzer Ludwik Zamenhof publicó en ruso un libro titulado *Lengua Internacional. Prólogo y manual completo (para rusos)*, Varsavia. Tipografía Kelter. El nombre de *Esperanto* fue adoptado universalmente puesto que el autor había firmado su libro con el seudónimo de Doktoro Esperanto (doctor esperanzado).

En realidad, Zamenhof, quien nació en 1859, había empezado a soñar con una lengua internacional desde la adolescencia. Nacido en el seno de una familia judía lituana en Bielostok, perteneciente al reino de Polonia, que estaba bajo el dominio del zar, Zamenhof creció en una encrucijada de razas y de lenguas, agitada por impulsos nacionalistas y por permanentes oleadas de antisemitismo. La experiencia de la opresión y, más tarde, de la persecución llevada a cabo por el gobierno zarista contra los intelectuales, especialmente los judíos, habría hecho que aparecieran al mismo tiempo la idea de una lengua universal y la de una concordia entre los pueblos. Además, Zamenhof se sentía solidario con sus correligionarios y deseaba el regreso de los judíos a Palestina, pero su religiosidad hacia le impedía identificarse con formas de sionismo nacionalista, y más que pensar en el fin de la Diáspora como un regreso a la lengua de los padres, pensaba que los judíos de todo el mundo podrían ser unidos precisamente por medio de una lengua nueva.

FIGURA 13.1. Estilo `Sonny`



Capítulo 3

El *Esperanto* y la búsqueda de la lengua perfecta

El *Esperanto* fue propuesto al mundo por vez primera en 1857 cuando el doctor Lejzer Ludwik Zamenhof publicó en ruso un libro titulado *Lengua Internacional. Prólogo y manual completo (para rusos)*, Varsavia. Tipografía Kelter. El nombre de *Esperanto* fue adoptado universalmente puesto que el autor había firmado su libro con el seudónimo de Doktoro Esperanto - doctor esperanzado!

En realidad, Zamenhof, quien nació en 1859, había empezado a soñar con una lengua internacional desde la adolescencia. Nacido en el seno de una familia judía lituana en Bielostok, perteneciente al reino de Polonia, que estaba bajo el dominio del zar, Zamenhof creció en una encrucijada de razas y de lenguas, agitada por impulsos nacionalistas y por permanentes oleadas de antisemitismo. La experiencia de la opresión y, más tarde, de la persecución llevada a cabo por el gobierno zarista contra los intelectuales, especialmente los judíos, había hecho que aparecieran al mismo tiempo la idea de una lengua universal y la de una concordia entre los pueblos. Además, Zamenhof se sentía solidario con sus correligionarios y deseaba el regreso de los judíos a Palestina, pero su religiosidad hacia le impedía identificarse con formas de sionismo nacionalista, y más que pensar en el fin de la Diáspora como un regreso a la lengua de los padres, pensaba que los judíos de todo el mundo podrían ser unidos precisamente por medio de una lengua nueva.

Mientras que el *Esperanto* se divulgaba por varios países, principalmente en el área eslava, después en el resto de Europa, suscitando el interés de so-

FIGURA 13.2. Estilo `Lenny`

³Escrito por Ulf A. Lindgren.

CAPÍTULO 3**El Esperanto y la búsqueda de la lengua perfecta**

El Esperanto fue propuesto al mundo por vez primera en 1887 cuando el doctor Lejzer Ludwik Zamenhof publicó en ruso un libro titulado *Lengua Internacional. Prólogo y manual completo (para rusos)*. Varsavia, Tipografía Kelter. El nombre de *Esperanto* fue adoptado universalmente puesto que el autor había firmado su libro con el seudónimo de Doktoro Esperanto (doctor esperanzado).

En realidad, Zamenhof, quien nació en 1859, había empezado a soñar con una lengua internacional desde la adolescencia. Nacido en el seno de una familia judía lituana en Bielsk, perteneciente al reino de Polonia, que estaba bajo el dominio del zar, Zamenhof creció en una encrucijada de razas y de lenguas, agitado por impulsos nacionalistas, y por permanentes oleadas de antisemitismo. La experiencia de la opresión y, más tarde, de la persecución llevada a cabo por el gobierno zarista contra los intelectuales, especialmente los judíos, había hecho que aparecieran al mismo tiempo la idea de una lengua universal y la de una concordia entre los pueblos. Además, Zamenhof se sentía solidario con sus correligionarios y deseaba el regreso de los judíos a Palestina, pero su religiosidad laica le impedía identificarse con formas de sionismo nacionalista, y más que pensar en el fin de la Diáspora como un regreso a la lengua de los padres, pensaba que los judíos de todo el mundo podrían ser unidos precisamente por medio de una lengua nueva.

Mientras que el *Esperanto* se divulgaba por varios países, principalmente en el área eslava, después en el resto de Europa, suscitando el interés de

23

CAPÍTULO 3**El Esperanto y la búsqueda de la lengua perfecta**

El Esperanto fue propuesto al mundo por vez primera en 1887 cuando el doctor Lejzer Ludwik Zamenhof publicó en ruso un libro titulado *Lengua Internacional. Prólogo y manual completo (para rusos)*. Varsavia, Tipografía Kelter. El nombre de *Esperanto* fue adoptado universalmente puesto que el autor había firmado su libro con el seudónimo de Doktoro Esperanto (doctor esperanzado).

En realidad, Zamenhof, quien nació en 1859, había empezado a soñar con una lengua internacional desde la adolescencia. Nacido en el seno de una familia judía lituana en Bielsk, perteneciente al reino de Polonia, que estaba bajo el dominio del zar, Zamenhof creció en una encrucijada de razas y de lenguas, agitado por impulsos nacionalistas, y por permanentes oleadas de antisemitismo. La experiencia de la opresión y, más tarde, de la persecución llevada a cabo por el gobierno zarista contra los intelectuales, especialmente los judíos, había hecho que aparecieran al mismo tiempo la idea de una lengua universal y la de una concordia entre los pueblos. Además, Zamenhof se sentía solidario con sus correligionarios y deseaba el regreso de los judíos a Palestina, pero su religiosidad laica le impedía identificarse con formas de sionismo nacionalista, y más que pensar en el fin de la Diáspora como un regreso a la lengua de los padres, pensaba que los judíos de todo el mundo podrían ser unidos precisamente por medio de una lengua nueva.

23

FIGURA 13.3. Estilo Glenn**CAPÍTULO****3****El Esperanto y la búsqueda de la lengua perfecta**

El Esperanto fue propuesto al mundo por vez primera en 1887 cuando el doctor Lejzer Ludwik Zamenhof publicó en ruso un libro titulado *Lengua Internacional. Prólogo y manual completo (para rusos)*. Varsavia, Tipografía Kelter. El nombre de *Esperanto* fue adoptado universalmente puesto que el autor había firmado su libro con el seudónimo de Doktoro Esperanto (doctor esperanzado).

En realidad, Zamenhof, quien nació en 1859, había empezado a soñar con una lengua internacional desde la adolescencia. Nacido en el seno de una familia judía lituana en Bielsk, perteneciente al reino de Polonia, que estaba bajo el dominio del zar, Zamenhof creció en una encrucijada de razas y de lenguas, agitado por impulsos nacionalistas, y por permanentes oleadas de antisemitismo. La experiencia de la opresión y, más tarde, de la persecución llevada a cabo por el gobierno zarista contra los intelectuales, especialmente los judíos, había hecho que aparecieran al mismo tiempo la idea de una lengua universal y la de una concordia entre los pueblos. Además, Zamenhof

23

**CHAPTER
TWO****John von Neumann. the early years**

John von Neumann was born János von Neumann. He was called Jancsi as a child, a diminutive form of János; later, later he was called Johnny in the United States. His father, Max Neumann, was a top banker and he was brought up in a extended family setting in Budapest where as a child he learnt languages from the German and French governesses that were employed. Although the family were Jewish, Max Neumann did not observe the strict practices of that religion and the household seemed to mix Jewish and Christian traditions.

It is also worth explaining how Max Neumann's son acquired the "von" to become János von Neumann. In 1913 Max Neumann purchased a title but did not change his name. His son, however, used the German form von Neumann where the "von" indicates the title.

At the age of six, he was able to exchange jokes with his father in classical Greek. The Neumann family sometimes entertained guests with demonstrations of Johnny's ability to memorise phone books. A guest would select a page and column of the phone book at random. Young Johnny read the column over a few times, then handed the book back to the guest. He could answer any question put to him, "who has number such and such?" or recite names, addresses, and numbers in order. In 1911 von Neumann entered the Lutheran Gymnasium. The school had a strong academic tradition which seemed to count for more than the religious affiliation both in the Neumann's eyes and in those of the school.

His mathematics teacher quickly recognised von Neumann's genius and

13

FIGURA 13.5. Estilo Rejne**FIGURA 13.6. Estilo Bjarne**

Las primeras páginas del Índice general, el Índice alfabético y la Biблиografía también cambian su presentación, en concordancia con el estilo utilizado. El paquete `fncychap` no altera el formato de las secciones, las subsecciones ni los encabezados de las páginas, pero se puede usar, sin conflicto alguno, con el comando `\pagestyle` de L^AT_EX o con el paquete `fancyhdr` (sección 13.1). Los paquetes `babel` e `inputenc` también son compatibles con `fncychap`.

Todos los estilos utilizan las fuentes estándares, a excepción de `Lenny`, el cual recurre a las familias Times (`ptm`) y Helvetica (`phv`) de la colección PSNFSS (sección 9.4) para la expresión ‘Capítulo’ (o ‘Chapter’) y los números de los capítulos. Por consiguiente, para usar el estilo `Lenny`, la colección PSNFSS debe estar localmente instalada.

13.4. El paquete `fancybox`



Los archivos que conforman el paquete `fancybox` se encuentran en la carpeta /Paquetes/fancybox del CD adjunto.

El paquete `fancybox`⁴ define cuatro variantes del comando `\fbox` de L^AT_EX para cajas con bordes visibles (Tabla 13.1), así como nuevos entornos para enmarcar listas y material variado.

<code>\shadowbox{texto}</code>	Para cajas con sombras. El grosor de las líneas se controla con <code>\fboxrule</code> (el mismo parámetro usado por <code>\fbox</code> , sección 3.15.3). El tamaño de la sombra depende del parámetro <code>\shadowsize</code> y su valor por defecto es <code>4pt</code> ; se puede cambiar con <code>\setlength</code> .
<code>\doublebox{texto}</code>	Para cajas con doble marco exterior. El grosor del borde exterior es <code>1.5\fboxrule</code> y el del borde interior <code>0.75\fboxrule</code> .
<code>\ovalbox{texto}</code>	Para cajas con bordes redondeados. El diámetro de los arcos esquineros se puede modificar con el parámetro de un argumento <code>\cornersize{n}</code> , el cual hace que el diámetro sea <code>n</code> veces la menor de las magnitudes ancho y alto de la caja. Por defecto se establece <code>\cornersize{0.5}</code> .
<code>\Ovalbox{texto}</code>	Similar a <code>\ovalbox</code> pero con líneas de mayor grosor.

TABLA 13.1. Los comandos para cajas del paquete `fancybox`.

⁴Escrito por Timothy Van Zandt; se carga con `\usepackage{fancybox}`.

Como sucede con `\fbox`, en estos comandos el argumento *texto* es procesado por L^AT_EX en modo-ID, pero en él se pueden incluir tablas (con el entorno `tabular`) o gráficas externas (con `\includegraphics`). La distancia entre el texto y el borde la caja se controla con el parámetro `\fboxsep` mencionado en la sección 3.15.3.

Ejemplo Los cuatro tipos de cajas del paquete `fancybox`.

Caja con sombra

```
\shadowbox{Caja con sombra}
```

Caja con doble borde

```
\doublebox{Caja con doble borde}
```

Caja ovalada con líneas delgadas

```
\ovalbox{Caja ovalada con %  
líneas delgadas}
```

Caja ovalada con líneas gruesas

```
\Ovalbox{Caja ovalada con %  
líneas gruesas}
```

El paquete `fancybox` tiene predefinidos los entornos `Bcenter`, `Bflushleft`, `Bflushright`, `Bitemize`, `Benumerate` y `Bdescription` que se usan como sus homólogos `center`, `flushleft`, `flushright`, `itemize`, `enumerate` y `description` de L^AT_EX, pero permiten ser enmarcados con `\fbox` o con los comandos de la Tabla 13.1. También se dispone del entorno genérico `Blist`, con los mismos argumentos obligatorios del entorno `list` de L^AT_EX (sección 8.3).

Ejemplo Se usa el entorno `Bcenter` dentro de una caja `\shadowbox`.

“Sabio no es quien
domina muchas lenguas,
sino quien sabe guardar
silencio en todas ellas”
Jorge Luis Borges

```
\shadowbox{  
\begin{Bcenter}  
“Sabio no es quien\\  
domina muchas lenguas,\\  
sino quien sabe guardar\\  
silencio en todas ellas”\\  
Jorge Luis Borges  
\end{Bcenter}}
```

Los entornos para listas, `Bitemize`, `Benumerate`, `Bdescription` y `Blist`, admiten un argumento opcional, `[t]` (tope) o `[b]` (base), que permite alinear la caja con el material circundante. Por defecto, las cajas quedan verticalmente centradas con respecto a lo que las rodea. El argumento

opcional [t] o [b] se escribe después de \begin{B...}, como se ilustra en el siguiente ejemplo.

Ejemplo Se usa el entorno **Bitemize**, con el argumento de alineación [t], dentro de una caja \Ovalbox.

Complejo B

- Vitamina B_1 . Tiamina.
- Vitamina B_2 . Riboflavina.
- Vitamina B_3 . Niacina.
- Vitamina B_6 . Piridoxina.
- Vitamina B_9 . Ácido fólico.

```
\underline{Complejo B}\qquad \Ovalbox{
\begin{Bitemize}[t]
\item Vitamina $B\_1$. Tiamina.
\item Vitamina $B\_2$. Riboflavina.
\item Vitamina $B\_3$. Niacina.
\item Vitamina $B\_6$. Piridoxina.
\item Vitamina $B\_9$. \'Acido f\'olico.
\end{Bitemize}}
```

13.5. El paquete **colortbl**



Los archivos que conforman el paquete **colortbl** se encuentran en la carpeta /Paquetes/colortbl del CD adjunto.

Desde el punto de vista **TeX-nico**, colorear las filas, columnas o casillas de una tabla no es tan sencillo como parece. Hay varios paquetes diseñados para tal propósito entre los que se destaca **colortbl**⁵. Este paquete requiere que se carguen de antemano los paquetes **color** (sección 7.2) y **array** (sección 5.12). Los colores se pueden especificar en cualquiera de los tres modelos **cmyk**, **rgb** o **gray** presentados en la sección 7.2.

Para colorear una fila completa, el contenido de la fila va precedido de alguna de las tres instrucciones siguientes:

\rowcolor[cmyk]{n ₁ , n ₂ , n ₃ , n ₄ }	(modelo cmyk , $0 \leq n_1, n_2, n_3, n_4 \leq 1$).
\rowcolor[rgb]{n ₁ , n ₂ , n ₃ }	(modelo rgb , $0 \leq n_1, n_2, n_3 \leq 1$).
\rowcolor[gray]{n}	(modelo gray , $0 \leq n \leq 1$).

⁵Escrito por David Carlisle.

Ejemplo Las filas de la siguiente tabla se colorean usando el modelo `gray`, alternando intensidades.

Fila 1, columna 1	Fila 1, columna 2
Fila 2, columna 1	Fila 2, columna 2
Fila 3, columna 1	Fila 3, columna 2
Fila 4, columna 1	Fila 4, columna 2

```
\begin{tabular}{|l|l|}\hline
\rowcolor[gray]{0.4}
Fila 1, columna 1 & Fila 1, columna 2\\ \hline
\rowcolor[gray]{0.6}
Fila 2, columna 1 & Fila 2, columna 2\\ \hline
\rowcolor[gray]{0.4}
Fila 3, columna 1 & Fila 3, columna 2\\ \hline
\rowcolor[gray]{0.6}
Fila 4, columna 1 & Fila 4, columna 2\\ \hline
\end{tabular}
```

Para colorear una columna individual se dispone del comando

`\columncolor[modelo]{color} [exceso izq.] [exceso der.]`

donde los argumentos opcionales [*exceso izq.*] y [*exceso der.*] indican el espacio extra a ser coloreado, a izquierda y a derecha, respectivamente, de la entrada más ancha en la columna. Si se omiten los argumentos opcionales, el color abarcará la anchura natural de la columna (es decir, los argumentos asumen el valor dado por el parámetro `\tabcolsep`).

El comando `\columncolor` se debe escribir en el formato de la tabla, en la forma `>\{\columncolor[...]{...}\[...]\[...]\}`, que corresponde a la sintaxis del paquete `array` (sección 5.12).

Ejemplo Compárese el uso de `\columncolor` en estas dos tablas.

Uno	Dos
Tres	Cuatro

```
\begin{tabular}{%>{\columncolor[gray]{0.9}}c%>{\columncolor[gray]{0.7}}c%}
Uno & Dos\\
Tres & Cuatro
\end{tabular}
```

Uno	Dos
Tres	Cuatro

```
\begin{tabular}{%>{\columncolor[gray]{0.9}[1pt][1pt]}c%>{\columncolor[gray]{0.7}[2pt][2pt]}c%}
Uno & Dos\\
Tres & Cuatro
\end{tabular}
```

Ejemplo

Planeta	Volumen	Masa	Gravedad
Tierra	1	1	1
Mercurio	0.056	0.055	0.37
Venus	0.858	0.815	0.91
Marte	0.152	0.107	0.38
Júpiter	1338	318	2.54
Saturno	776	95.2	1.08
Urano	60.4	14.6	0.88
Neptuno	56.9	17.2	1.15
Plutón	0.008	0.0025	0.43

```
\begin{center}
\renewcommand{\arraystretch}{1.1}
\begin{tabular}{|>{\color[gray]{0.9}}c|>{\color[gray]{0.8}}c|>{\color[gray]{0.7}}c|>{\color[gray]{0.6}}c|} \hline
\sf Planeta & \sf Volumen & \sf Masa & \sf Gravedad \\ \hline \hline
Tierra & 1 & 1 & 1 \\
Mercurio & 0.056 & 0.055 & 0.37 \\
Venus & 0.858 & 0.815 & 0.91 \\
Marte & 0.152 & 0.107 & 0.38 \\
Júpiter & 1338 & 318 & 2.54 \\
Saturno & 776 & 95.2 & 1.08 \\
Urano & 60.4 & 14.6 & 0.88 \\
Neptuno & 56.9 & 17.2 & 1.15 \\
Plutón & 0.008 & 0.0025 & 0.43 \\
\hline
\end{tabular}
\end{center}
```

El formato de una tabla se puede simplificar con `\newcolumntype`, comando del paquete `array`. Para los cuatro tipos de columna de la tabla anterior, por ejemplo, podemos introducir nombres concretos:

```
\newcolumntype{G}{>{\color[gray]{0.9}}c}
\newcolumntype{H}{>{\color[gray]{0.8}}c}
\newcolumntype{I}{>{\color[gray]{0.7}}c}
\newcolumntype{J}{>{\color[gray]{0.6}}c}
```

con lo cual el formato de la tabla se puede presentar simplemente como `\begin{tabular}{|G|H|I|J|}`.

Para colorear casillas individuales, en vez de filas o columnas enteras, se debe usar el comando `\multicolumn` de L^AT_EX (sección 5.3), como se hace en el siguiente ejemplo.

Ejemplo Aquí se usa `\newcolumntype` para definir un nuevo tipo de columna y se colorean casillas individuales por medio de `\multicolumn{1}{c}{...}`.

A 1	A 2	A 3	A 4
B 1	B 2	B 3	B 4
C 1	C 2	C 3	C 4
D 1	D 2	D 3	D 4

```
\begin{center}
\renewcommand{\arraystretch}{1.2}
\newcolumntype{G}{>{\color{gray}{0.8}}c}
\begin{tabular}{|cccc|}\hline
\multicolumn{1}{|G}{A 1} & A 2 & \multicolumn{1}{G}{A 3} & A 4 \\
B 1 & \multicolumn{1}{G}{B 2} & B 3 & \multicolumn{1}{G}{B 4} \\
\multicolumn{1}{|G}{C 1} & C 2 & \multicolumn{1}{G}{C 3} & C 4 \\
D 1 & \multicolumn{1}{G}{D 2} & D 3 & \multicolumn{1}{G}{D 4} \\
\hline \end{tabular}
\end{center}
```

13.6. El paquete `longtable`

L^AT_EX es incapaz de dividir una tabla para acomodarla en dos páginas porque el entorno `tabular` produce cajas indivisibles. Cuando se presente este problema o cuando en el documento haya tablas muy extensas, se puede recurrir al paquete `longtable`⁶. Este paquete define un nuevo entorno, `longtable`, que se usa en la misma forma que el entorno `tabular`:

<code>\begin{longtable}{formato}</code>
.....
<code>\end{longtable}</code>

El entorno `longtable` produce tablas que pueden ser divididas en dos o más páginas por el algoritmo normal de paginación de T_EX. El nuevo entorno es compatible con `table` y usa el mismo contador que éste. Hay un detalle adicional que se debe tener en cuenta: al usar `longtable`, el documento puede requerir varias compilaciones para que las tablas obtenidas sean correctas.

⁶Paquete estándar de L^AT_EX 2_E, escrito por David Carlisle. Se carga por medio de `\usepackage{longtable}`.

13.7. El paquete `caption2`



Los archivos que conforman el paquete `caption2` se encuentran en la carpeta `/Paquetes/caption2` del CD adjunto.

Los epígrafes o rótulos para las tablas y gráficas se obtienen en L^AT_EX con el comando `\caption` (sección 7.5), pero el usuario no tiene ningún control sobre su apariencia. Para las tablas, los epígrafes aparecen por defecto en la forma:

Tabla 2.3: Texto del epígrafe.

Para las figuras se obtiene algo similar. Los dos puntos que siguen al número se imprimen automáticamente; si queremos deshacernos de ellos, o si queremos cambiar el tipo o tamaño de letra para las expresiones ‘Tabla’ o ‘Figura’, debemos recurrir al paquete `caption2`, el cual brinda todo un surtido de opciones para los rótulos. El paquete⁷ se carga escribiendo

```
\usepackage[opciones]{caption2}
```

en el preámbulo del documento. Las *opciones* disponibles aparecen en la Tabla 13.2; se escriben separadas por comas. Por ejemplo, si se carga el paquete escribiendo

```
\usepackage[centerlast,small,sf]{caption2}
```

las expresiones ‘Tabla’ y ‘Figura’ aparecerán en letra sans serif pequeña y, en el texto de los rótulos, el último renglón aparecerá centrado (`centerlast`). Los demás renglones aparecerán en la forma usual: justificados a izquierda y a derecha.

El paquete incluye varios comandos para controlar aún más la apariencia de los epígrafes, entre los que destacamos:

`\setcaptionwidth{ancho}` Hace que los epígrafes tengan la amplitud horizontal especificada por *ancho*, que debe ser una dimensión T_EX. Este comando se puede usar múltiples veces en un documento para lograr que la anchura de los epígrafes no exceda la amplitud de las tablas o gráficas (para lo cual se requiere, por supuesto, conocer o estimar la amplitud en cada caso).

`\setcaptionmargin{d}` Establece una distancia *d* entre los epígrafes y cada uno de los márgenes del documento; *d* debe ser una dimensión T_EX. Ésta es otra forma de controlar la anchura de los epígrafes.

⁷Escrito por Harald Sommerfeldt.

Estilo de los epígrafes.	<code>normal</code> <code>center</code> <code>flushleft</code> <code>flushright</code> <code>centerlast</code> <code>indent</code> <code>hang</code>	renglones justificados a izquierda y a derecha. renglones centrados. renglones justificados a la izquierda. renglones justificados a la derecha. renglones justificados a izquierda y derecha, excepto el último, que aparece centrado. sangría a partir del segundo renglón. como <code>indent</code> , pero la sangría tiene el tamaño de las expresiones ‘Tabla #’ y ‘Figura #’.
Tamaño de las expresiones ‘Tabla’ y ‘Figura’ y del texto mismo del epígrafe.	<code>scriptsize</code> <code>footnotesize</code> <code>small</code> <code>normalsize</code> <code>large</code> <code>Large</code>	
Fuente para las expresiones ‘Tabla’ y ‘Figura’. No afecta el texto mismo del epígrafe.	<code>rm</code> (letra romana normal) <code>sf</code> (letra sans serif) <code>tt</code> (letra mono-espaciada)	
Variante de la fuente para las expresiones ‘Tabla’ y ‘Figura’. No afecta el texto mismo del epígrafe.	<code>up</code> (letra vertical normal) <code>it</code> (letra cursiva) <code>sl</code> (letra inclinada) <code>sc</code> (versalitas)	
Peso de la fuente para las expresiones ‘Tabla’ y ‘Figura’. No afecta el texto mismo del epígrafe.	<code>md</code> (medio o normal; puede omitirse) <code>bf</code> (negrilla)	

TABLA 13.2. Las opciones del paquete `caption2`.

`\captionlabeldelim` Controla el símbolo que aparece después del número de la tabla o gráfica. L^AT_EX utiliza por defecto los dos puntos (:), pero este símbolo se puede cambiar con el citado parámetro. Esto se hace por medio de `\renewcommand`. Así, si se quiere solamente un punto (algo como ‘Tabla 5.1.’ o ‘Figura 3.7.’, etc), se escribe `\renewcommand{\captionlabeldelim}{.}`.

`\captionstyle{estilo}` Cambia el estilo de un epígrafe particular, con respecto al establecido en `\usepackage[...]{caption2}`. El *estilo* puede ser cualquiera de los mostrados en la Tabla 13.2.

13.8. El paquete **picinpar**

 Los archivos que conforman el paquete **picinpar** se encuentran en la carpeta **/Paquetes/picinpar** del CD adjunto.

El paquete **picinpar**⁸ permite incrustar “ventanas” en párrafos normales. Hay tres entornos disponibles: **window** y sus variantes **tabwindow** (para insertar tablas) y **figwindow** (para insertar figuras o gráficas). El entorno básico **window** se usa en la siguiente forma:

```
\begin{window}[n,posición,Texto,rótulo]
```

El significado de los argumentos es el siguiente:

- n** Número de renglones del párrafo antes de la ventana interior; $n \geq 0$.
- posición** Posición de la ventana dentro del párrafo. Sus valores permitidos son: **l** (cargada a la izquierda), **c** (centrada) y **r** (cargada a la derecha).
- Texto** Es el contenido de la ventana. Se aconseja encerrar este argumento entre corchetes `{...}`.
- rótulo** Es el epígrafe o rótulo para la ventana. Si no se desea ninguno, se escribe simplemente `{}` en este argumento.

Los entornos **tabwindow** y **figwindow** tienen la misma sintaxis de **window** pero los epígrafes o rótulos aparecen demarcados y numerados como ‘Tabla #’ y ‘Figura #’, respectivamente. Dichos entornos utilizan los contadores **table** y **figure**, pero *no* se insertan como objetos flotantes (sección 7.4). Por consiguiente, el usuario debe verificar que no haya errores en la secuencia de numeración de tablas y gráficas. De ser necesario los contadores se pueden modificar manualmente (sección 3.25.1). Alternativamente, se puede usar el entorno **window**, en vez de **tabwindow** o **figwindow**, y escribir manualmente los epígrafes en el argumento **rótulo**.

Ejemplo Se introduce una letra S grande en la parte izquierda del párrafo.

Según la teoría del *Big Bang*, el universo se habría formado hace unos quince mil millones de años, tras una gigantesca explosión. Después habrían aparecido las galaxias y sus miríadas de estrellas.

```
\begin{window}[0,1,{\bf\Huge S},{}]
\small \noindent seg\xedn la teor\xeda del \textit{Big Bang}, el ...
... habr\xian aparecido las galaxias y sus mir\xadas de estrellas.
\end{window}
```

⁸Escrito por Friedhelm Sowa, a partir de macros de Alan Hoenig.

Ejemplo En el centro del párrafo exhibido a continuación se inserta una caja `\fbox` que encierra, a su vez, un párrafo. Se ha utilizado el tipo de letra sans serif, en el tamaño `\footnotesize`. Obsérvese el uso de la declaración `{\sloppy ...}` en el párrafo exterior; este comando de L^AT_EX hace que la separación entre palabras sea más flexible que lo usual (a costa de mayor espacio en blanco entre las palabras). El uso de `\sloppy` es recomendable en estas situaciones para que la excesiva división silábica de palabras no origine particiones equivocadas o problemas de invasión de texto en los márgenes.

La diversidad de la vida en la Tierra sugiere que la vida extraterrestre debe ser también sumamente variada. A nivel molecular, puede suponerse que la vida extraterrestre se basa en la química orgánica y en el agua líquida. El carbono es un elemento atómico universo, y el agua bien común en el cosmos similar al carbono. Una química tan níaco, análogo nitro-puede sustituirla, ya una gama estrecha de temperaturas. Parece que el exobiólogo no puede hacer gala de mucha originalidad: la vida extraterrestre *debe* estar basada en el carbono y en un medio acuoso. La hipótesis del carbono se refuerza por la presencia de numerosos compuestos orgánicos en ambientes muy variados del cosmos.

ECOLOGÍA EXTRATERRESTRE
Sin proponérselo, el hombre ha enviado bacterias y microbios al espacio e incluso a otros planetas. La nueva ciencia de la exobiología, que estudia la materia viva fuera de la tierra, ha entrado ya en conflicto con aguerridos ecologistas.

```
\begin{window}[3,c,\fbox{\parbox{5.4cm}{\sf ECOLOGÍA EXTRATERRESTRE  
\footnotesize Sin proponérselo, el hombre ha enviado bacterias y  
microbios al espacio e incluso a otros planetas. La nueva ciencia  
de la exobiología, que estudia la materia viva fuera de la tierra,  
ha entrado ya en conflicto con aguerridos ecologistas.}}},{}]  
\sloppy \small La diversidad de la vida en la Tierra sugiere ...  
... compuestos orgánicos en ambientes muy variados del cosmos.  
\end{window}
```

Ejemplo En la página siguiente se usa el entorno `figwindow` para insertar una gráfica en la parte derecha de un párrafo. La gráfica propiamente dicha se invoca con `\includegraphics`. El uso de `figwindow` origina automáticamente el rótulo ‘Figura 13.7.’, en negrilla. El aspecto de estos rótulos es propio del paquete `picinpar`, el cual ignora cualquier estilo declarado por el paquete `caption2`, descrito en la sección anterior (sección 13.7). Se pueden usar todas las opciones de `\includegraphics` (sección 7.3.4) que sean necesarias para modificar la apariencia de la gráfica.

Donald Knuth, nacido en 1938 en Milwaukee (Wisconsin), es uno de los más reconocidos científicos en las ciencias de la computación. Sus contribuciones pioneras sobre compiladores, gramáticas atributivas y algoritmos lo hicieron merecedor del premio Turing en 1974, y le han valido muchas otras distinciones, entre las que se encuentran doctorados *Honoris Causa* de más de quince universidades alrededor del mundo. Su proyecto central ha sido *The Art of Computer Programming*, trabajo monumental en siete volúmenes, escrito a lo largo de casi 30 años. Knuth es más conocido en el ámbito académico por su sistema *TeX* y su programa *METAFONT* para el diseño de fuentes, que han sentado los cánones de calidad en el procesamiento digital de textos, desde su introducción a mediados de los ochenta.

En su libro *Digital Typography*, recopilación de sus trabajos en el área, publicado en 1999, afirma: “Creo que por mis venas corre tinta. Cuando comprendí el potencial de la tecnología computacional para la impresión de documentos, no pude resistir la tentación de dejar a un lado todo lo demás para dedicarme a adaptar la maestría del centenario arte de la tipografía a las posibilidades del presente.”



Figura 13.7: Donald Knuth

```
\begin{figwindow}[3,r,\includegraphics{c:/Imagenes/Knuth.eps},%
{Donald Knuth}]
\noindent \small \sloppy \textbf{Donald Knuth}, nacido en 1938 en
Milwaukee (Wisconsin), es uno de los más reconocidos científicos en
las ciencias de la computación. Sus contribuciones pioneras sobre
compiladores, gramáticas atributivas y algoritmos lo hicieron
merecedor del premio Turing en 1974, y le han valido muchas otras
distinciones, entre las que se encuentran doctorados \textit{Honoris
Causa} de más de quince universidades alrededor del mundo. Su
proyecto central ha sido \emph{The Art of Computer Programming},
trabajo ... escrito a lo largo de casi 30 años. Knuth es más conocido
en el ámbito académico por su sistema {\TeX} y su programa ...
... introducción a mediados de los ochenta.\par
En su libro \emph{Digital Typography}, recopilación de sus trabajos
en el área, publicado en 1999, afirma: “Creo que por mis venas
corre tinta. Cuando comprendí el potencial de la tecnología
computacional para la impresión de documentos, no pude resistir la
tentación de dejar a un lado todo lo demás para dedicarme a adaptar
la maestría del centenario arte de la tipografía a las posibilidades
del presente.”}
\end{figwindow}
```

13.9. El paquete `lscape`

El paquete `lscape`⁹ es útil para incluir material en posición de paisaje o *landscape* (rotado 90°) en un documento con orientación vertical normal. Para usarlo, es necesario cargar alguno de los paquetes `graphics` o `graphicx` (sección 7.3). El paquete `lscape` tiene definido el entorno `landscape`, que se usa en la forma

```
\begin{landscape}
    material
\end{landscape}
```

El *material* bajo el alcance de este entorno, que puede incluir cualquier combinación de texto, tablas y gráficas, es rotado 90°. El contenido del entorno `landscape` se puede extender por varias páginas. Los encabezados y pies de página no son rotados, es decir, mantienen la orientación y presentación de páginas normales.

El entorno `landscape` es compatible con el paquete `longtable` (sección 13.6) y es capaz de generar páginas consecutivas con tablas divididas en partes y rotadas.

Hay que tener presente la siguiente gran limitación del paquete `lscape`: con el entorno `landscape` se inicia siempre una nueva página, lo que puede dar lugar a páginas parcialmente vacías.

13.10. El paquete `subfiles`



Los archivos que conforman el paquete `subfiles` se encuentran en la carpeta `/Paquetes/subfiles` del CD adjunto.

El paquete `subfiles`¹⁰ proporciona una alternativa para el manejo de un documento raíz, o documento principal, con documentos subsidiarios. A diferencia de los mecanismos mencionados en la sección 6.3, con el paquete `subfiles` los archivos subsidiarios se pueden procesar independientemente, en cuyo caso adquieren el preámbulo del archivo principal.

En el archivo principal se carga el paquete `subfiles`, y los archivos subsidiarios se invocan con `\subfile{...}`, tal como se exhibe en el siguiente diagrama.

⁹Paquete estándar de L^AT_EX 2_ε, escrito por David Carlisle. Se carga por medio de `\usepackage{lscape}`.

¹⁰Escrito por Federico García De Castro.

```
\documentclass[...]{...}

preámbulo  =>  {   :
                  \usepackage{subfiles}
                  :
                  \begin{document}
                  :
                  \subfile{archivo1}
                  \subfile{archivo2}
                  :
                  \end{document}
```

Cada uno de los archivos subsidiarios, `archivo1.tex`, `archivo2.tex`, etc, tiene la siguiente estructura:

```
\documentclass[archivo principal]{subfiles}
\begin{document}
:
\end{document}
```

donde sólo es necesario indicar la raíz del nombre del *archivo principal*, si su extensión es `.tex`. Si éste se encuentra en otra carpeta, se debe escribir la ruta completa, usando la sintaxis del sistema operativo local.

El comando `\subfile` se asemeja más a `\input` que a `\include` (sección 6.3): con `\subfile` no se inicia una nueva página, en los documentos subsidiarios se admiten inserciones múltiples de otros comandos `\subfile`, pero no hay un mecanismo de exclusión similar `\includeonly`.

Si un documento subsidiario se procesa por sí solo, adquiere el preámbulo del *archivo principal*, incluyendo su `\documentclass`. Hay que advertir que, al procesar independientemente un documento subsidiario, todo lo que aparezca después de `\end{document}` en el archivo principal también es tenido en cuenta.

Apéndice A

A.1 Mensajes de error de L^AT_EX

Cuando L^AT_EX encuentra un error sintáctico, detiene el procesamiento del documento fuente y emite un mensaje de error. La siguiente es la lista alfabética de los mensajes de error más frecuentes generados por L^AT_EX, con una explicación sucinta sobre las causas que los generan. El programa núcleo T_EX también genera sus propios mensajes de error (véase la sección A.2).

- **Bad math environment delimiter.** Falta o sobra alguno de los símbolos de delimitación del modo matemático \$, \[o \].
- **\begin{...} on input line ... ended by \end{...}.** Hay un comando \end que no corresponde a ningún \begin. Es posible que haya algún error tipográfico en el nombre del entorno invocado o algún \end{...} faltante.
- **Can be used only in preamble.** L^AT_EX ha encontrado uno de los comandos del preámbulo del documento *después* de la instrucción \begin{document}.
- **Command ... already defined.** Se ha intentado definir un comando o entorno *ya existente*.
- **Command ... invalid in math mode.** El comando indicado no se puede usar en modo matemático.
- **Environment ... undefined.** Algun comando \begin está tratando de invocar un entorno que no está definido. Es probable que haya un error tipográfico en el nombre del entorno o se haya olvidado cargar el paquete que define el entorno.
- **Illegal character in array arg.** Hay un carácter no permitido en los argumentos de un entorno **array** o **tabular**, o en el segundo argumento de un **multicolumn**.

- **Lonely \item—perhaps a missing list environment.** Hay un comando `\item` que no pertenece a ninguno de los entornos `itemize`, `enemorate`, `description` o `list`.
- **Missing \begin{document}.** Falta el comando `\begin{document}`, sin el cual L^AT_EX no sabe dónde termina el preámbulo y dónde comienza el cuerpo del documento.
- **Missing p-arg in array arg.** La sintaxis de un argumento `p{...}` en alguno de los entornos `array` o `tabular` es incorrecta.
- **No \title given.** Hay un comando `\maketitle` pero no se ha definido `\title`.
- **Option clash for package ...** Se ha cargado un mismo paquete dos veces, con opciones diferentes. Es posible que el paquete esté siendo cargado automáticamente por otro paquete.
- **Something's wrong--perhaps a missing \item.** L^AT_EX percibe que hay algo mal, pero no puede determinar qué es. Este mensaje puede ser generado por muchas causas, incluyendo la omisión de un comando `\item`, o de algún argumento en el entorno `thebibliography`.
- **There's no line here to end.** Hay un comando `\newline` o `\\"` no permitido en el interior de algún comando o entorno.
- **Unknown option ... for ...** Algun comando `\usepackage` o el comando `\documentclass` tiene especificada una opción ilegal.
- **\verb illegal in command argument.** El comando `\verb` no puede aparecer en el argumento de otro comando.

A.2 Mensajes de error de T_EX

La siguiente es la lista de los mensajes de error más frecuentes generados por T_EX, con una explicación sucinta sobre las causas que los generan.

- **! Double subscript.** Hay dos subíndices consecutivos en una fórmula matemática, como en `x_{i}_{j}`. Para producir $x_{i,j}$ se debe escribir `$x_{i,j}$`.
- **! Double superscript.** Hay dos superíndices consecutivos en una fórmula matemática, como en `x^{a}{b}`. Para producir x^{a^b} se debe escribir `x^{a^b}`.

- ! Extra alignment tab has been changed to \cr. Hay demasiados &'s en una sola fila de un entorno **array** o **tabular**.
- ! Extra }, or forgotten \$. Error muy común; se ha olvidado abrir o cerrar un corchete o un signo \$.
- ! Illegal parameter number in definition of Se ha usado incorrectamente un carácter # en la definición de un nuevo comando o entorno.
- ! Misplaced alignment tab character &. El carácter especial & sólo se puede usar para separar las columnas en los entornos **array** o **tabular**.
- ! Missing number, treated as zero. T_EX estaba esperando un número o una longitud como argumento de algún comando, y encuentra algo diferente. Causas frecuentes de este mensaje son la omisión de un argumento o la omisión de las unidades de medida.
- ! Missing { inserted. o ! Missing } inserted. Error muy frecuente; falta o sobra un corchete.
- ! Missing \$ inserted. Es posible que falte un símbolo \$, o bien que T_EX haya encontrado, en modo normal, un comando que sólo puede ser usado en modo matemático.
- ! Paragraph ended before ... was complete. Error muy común; se ha dejado una línea en blanco (o un comando \par) dentro de un argumento que no puede contenerla. Otras causas posibles: se ha olvidado un } derecho o un \end{...}.
- ! TeX capacity exceeded, sorry [...]. T_EX ha agotado su memoria. La causa más probable para este tenebroso mensaje es algún error sintáctico en el documento fuente que hace que T_EX ingrese en un bucle interminable o lea texto indefinidamente, como argumento de un comando, hasta agotar la memoria disponible.
- ! Undefined control sequence. Error muy común; T_EX ha encontrado un comando que no está definido. Lo más probable es que haya un error tipográfico por parte del usuario.
- ! Use of ... doesn't match its definition. Uso incorrecto de un comando o entorno.

Apéndice B

Configuración de L^AT_EX para el uso de babel

Para usar el paquete **babel** en un documento L^AT_EX no basta invocarlo con `\usepackage{babel}`. Se requiere también que los archivos de patrones de partición silábica de los idiomas usados estén presentes en el sistema, tal como se indicó en la sección 2.9. Finalmente, hay que reconfigurar el formato de procesamiento de L^AT_EX, si la implementación local no lo hace automáticamente. Para esta última etapa se procede de la siguiente manera:

1. Ubicar el archivo `language.dat` en `/texmf/tex/generic/config` (o carpeta equivalente). Se trata de un documento de texto llano con un contenido similar a:

```
% File      : language.dat
% Purpose   : specify which hyphenation patterns to load
%             while running iniTeX
=USenglish
american ushyphen.tex
english ukhyphen.tex
```

2. Utilizando un editor de texto, añadir la línea correspondiente a los nuevos idiomas; por ejemplo,

```
spanish eshyph.tex
french frhyph.tex
```

3. Correr INITEX con el formato L^AT_EX. La manera precisa de hacerlo depende de la implementación T_EX local; el usuario debe consultar la documentación pertinente.

Para comprobar que el procedimiento se realizó correctamente, se puede leer el contenido de un archivo ‘`—.log`’ particular, correspondiente a un documento en el que se invoque **babel** con los nuevos idiomas. En la parte inicial del archivo ‘`—.log`’ se debe leer algo como:

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

Apéndice C

La estructura de directorios TDS y la instalación de paquetes L^AT_EX

Puesto que T_EX, L^AT_EX y los paquetes de macros que se ejecutan en su ambiente son de dominio público, ha habido desde un comienzo mucha flexibilidad y cierta anarquía en las implementaciones, tanto gratuitas como comerciales, de estos programas. Con el propósito de evitar el caos y crear cierta uniformidad para todas las plataformas computacionales, la asociación TUG (*T_EX Users Group*) propuso en 1994 una estructura de directorios, conocida como TDS o *T_EX Directory Structure*. Según el esquema TDS, el directorio principal de toda implementación de T_EX y L^AT_EX debe ser `/texmf/`, abreviación de ‘T_EX y METAFONT’, los programas básicos diseñados por Donald Knuth.

Para el usuario L^AT_EX, el aspecto esencial sobre la estructura TDS es la ubicación de los archivos de paquetes nuevos. Vamos a explicar a continuación el procedimiento de instalación manual de un paquete L^AT_EX típico denominado **paquete**, aplicable a las implementaciones que respeten la estructura de directorios TDS. Esto puede servir de guía para otro tipo de implementaciones¹. En primer lugar, es necesario crear tres carpetas diferentes, con el nombre **paquete**, en cada uno de los directorios `/texmf/doc/latex/`, `/texmf/source/latex/` y `/texmf/tex/latex/`, tal como se ilustra en la Figura C.1 de la página siguiente.

En caso de que el paquete posea un archivo de instalación (`paquete.ins`) y un archivo fuente documentado (`paquete.dtx`), éstos se deben colocar en el subdirectorio `/texmf/source/latex/paquete/`. Luego se procesa el archivo `paquete.ins`, ya sea como documento T_EX o como documento L^AT_EX, utilizando para ello la interfaz propia de la implementación T_EX local (en general, se hace desde la línea de comandos, en la forma `tex paquete.ins`

¹ Algunas implementaciones tienen una interfaz propia para la instalación de paquetes. Con MiK_TeX, por ejemplo, se puede realizar la instalación automática de paquetes a través de la plantilla **Packages** de MiK_TeX Options o recurriendo al Package Manager. La instalación se hace por conexión en línea o desde un depósito local.

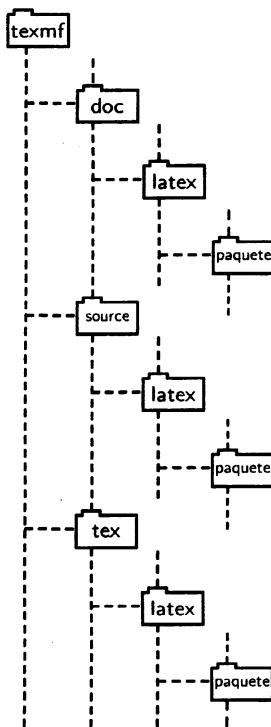


FIGURA C.1. Estructura de directorios, según el esquema TDS, para la instalación de paquetes LATEX.

o `latex paquete.ins`). Este procedimiento genera el archivo de macros `paquete.sty` que constituye el paquete propiamente dicho; LATEX consulta precisamente el archivo `paquete.sty` cuando encuentra la instrucción `\usepackage{paquete}`. Algunos paquetes se distribuyen sin archivo de instalación `ins`, en cuyo caso, sólo es necesario colocar directamente el archivo `sty` en la carpeta `/texmf/tex/latex/paquete/`. Ciertos paquetes generan también un archivo `paquete.cls`, que representa un estilo o clase de documento (similar a los estilos `article`, `book`, etc). Tanto los archivos `sty` como los `cls` se deben trasladar a `/texmf/tex/latex/paquete/`.

El archivo `paquete.dtx`, si hace parte de la distribución, se procesa luego como un documento LATEX normal para generar el archivo `paquete.dvi`, que contiene la documentación propia del paquete. Sucede en ocasiones que la documentación es un archivo `txt` de texto llano o un archivo en formatos PS o PDF, o incluso un archivo LATEX estándar. En cualquier caso, se recomienda colocar estos documentos en `/texmf/doc/latex/paquete/`.

Bibliografía

A la siguiente lista de publicaciones se debe añadir la documentación de los diferentes paquetes descritos a lo largo del presente libro. La documentación, si hace parte de la distribución de un paquete, se incluye en la carpeta respectiva del directorio **/Paquetes/** del CD adjunto.

- [1] Paul W. Abrahams, *TeX for the impatient*, Addison-Wesley, 1990.
- [2] Adobe Systems, *PostScript Language Reference Manual*, Addison-Wesley, 1985.
- [3] American Mathematical Society, *AMS- \LaTeX Version 1.2. User's Guide*, incluido en las distribuciones de $\text{\LaTeX}\ 2\varepsilon$, 1995.
- [4] Rodrigo De Castro K., *Gráficas en \TeX* , Lect. Mat. 15 (1994), 95–137.
- [5] Michel Goosens, Frank Mittelbach & Alexander Samarin, *The \LaTeX Companion*, Addison-Wesley, 1994.
- [6] Michel Goosens, Sebastian Rahtz & Frank Mittelbach, *The \LaTeX Graphics Companion*, Addison-Wesley, 1994.
- [7] George Grätzer, *Math into \LaTeX . An Introduction to \LaTeX and AMS- \LaTeX* , Birkhäuser, 1996.
- [8] Jane Hahn, *\LaTeX for Everyone*, second edition, Personal \TeX Inc, 1991.
- [9] Donald E. Knuth *The $\text{\TeX}book$* , Addison-Wesley, 1986.
- [10] Helmut Kopka & Patrick W. Daly, *A Guide to \LaTeX . Document Preparation for Beginners and Advanced Users*, third edition, Addison-Wesley, 1999.
- [11] Leslie Lamport, *\LaTeX . A Document Preparation System*, second edition, Addison-Wesley, 1994.
- [12] $\text{\LaTeX}3$ Project Team, *$\text{\LaTeX}\ 2\varepsilon$ for authors*, contenido del archivo `usrguide.tex`, incluido en las distribuciones de $\text{\LaTeX}\ 2\varepsilon$, 1999.

- [13] Keith Reckdahl, *Using imported graphics in L^AT_EX 2_&*, CTAN, 1997.
- [14] Michael J. Wichura, *The P_IC_TE_X Manual*, T_EXniques, Publications for the T_EX Community, no. 6, 1986.
- [15] Timothy van Zandt, **PSTricks: PostScript macros for generic T_EX**, *Users Guide*, CTAN, 1993.

Índice alfabético

- \#, 42
\\$, 42
\%, 42
\&, 42
\,, 43, 111
\:, 111
\;, 111
_, 6, 43
\!, 111
\|, 115
_, 42
\[\dots], 44, 146, 175
@, 264, 265
\@startsection, 265–267
\@seccntformat, 267
- abrv** (estilo bibliográfico), 212
abovecaptionskip, 240, 241
abstract (entorno), 16–18, 165, 189
abstractname, 189
acentos, 33
 - en modo matemático, 109**achicago** (paquete), 214, 215
acm (estilo bibliográfico), 212
Acrobat Reader, 290, 298, 299
Acrobatmenu, 298, 299
activeacute (opción), 34
\Acute, 109
\acute{e}, 109
\addcontentsline, 190–192
\address, 165, 257, 258
\addtocontents, 191, 192
\addtocounter, 78
Adobe Acrobat, 298, 299
ae (paquete), 272
AE (fuentes), 37, 271, 272, 291, 306
aecompl (paquete), 272
ajustes en
 - posición de números, 150
radicales, 100
@-expresiones en tablas, 177
align (entorno), 134, 136–138
align* (entorno), 136, 145
alignat (entorno), 134
aligned (entorno), 134, 141–143
alineación de fórmulas, 133–147
 - cambios de página en, 147
 - con **aligned**, 141–143
 - con **align**, 136–138
 - con **eqnarray**, 145
 - con **flalign**, 144
 - con **gathered**, 141–143
 - con **gather**, 135
 - con **multiline**, 134
 - con **split**, 139–141
allowdisplaybreaks, 147
almacenamiento de cajas, 73, 74
Alph (tipo de numeración), 27
\Alph, 79
alph (tipo de numeración), 27
\alph, 79
alpha (estilo bibliográfico), 212, 213
American Mathematical Society, 2, 10, 19, 153, 164
amsalpha (estilo bibliográfico), 212
amsart (estilo), 39, 164–166
amsbook (estilo), 39, 164–166
amscd (paquete), 156, 157
AMS Fonts, 87
amsfonts (paquete), 87
\AMS-\LaTeX, 2, 87
amsmath (paquete), 19, 87–166
amsplain (estilo bibliográfico), 212, 213
amssymb (paquete), 87–94
\AMS-\TeX, 2, 87
amsthm (paquete), 153–155

- \and, 14
apalike (estilo bibliográfico), 212
 apéndices, 184, 186, 187, 189
 \appendix, 184, 186, 187, 189
 \appendixname, 189
 arabic (tipo de numeración), 27
 \arabic, 78
 árboles (**PSTricks**), 377–382
 \arccos, 101
 archivos
 con extensión **aux**, 4, 5
 con extensión **bbl**, 207, 267
 con extensión **bib**, 206, 207, 209,
 210
 con extensión **blg**, 207
 con extensión **bst**, 212
 con extensión **dtx**, 449, 450
 con extensión **dvi**, 4, 5
 con extensión **fd**, 267, 271
 con extensión **idx**, 194–196
 con extensión **ilg**, 194–196
 con extensión **ind**, 194–196
 con extensión **ins**, 449, 450
 con extensión **ldf**, 21
 con extensión **lof**, 191
 con extensión **log**, 4, 5
 con extensión **lot**, 191
 con extensión **map**, 271, 302
 con extensión **mf**, 271, 272
 con extensión **pdf**, 272, 290, 291,
 294, 295, 305, 306
 con extensión **pfa**, 271
 con extensión **pfb**, 271
 con extensión **pk**, 271, 272
 con extensión **sty**, 18, 449, 450
 con extensión **tex**, 4, 5
 con extensión **tfm**, 271, 285
 con extensión **vf**, 271
 PDF, 3, 272, 290, 291, 294, 295,
 305, 306
 \arcsen, 113
 \arcsin, 101
 \arctan, 101
 \arctg, 113
 \arg, 101
 argumentos
 obligatorios, 8, 9
 opcionales, 8, 9
 array (paquete), 178–180, 182, 434–
 436
 array (entorno), 121, 145
 \arraycolsep, 145
 \arrayrulewidth, 176
 \arraystretch, 174
 \arrow (**PiCTeX**), 414
 \arrow (pb-diagram), 158
 arrows (**PSTricks**), 349, 377
 article (estilo), 10, 13, 14, 22, 23, 28,
 183
 ASCII, 4
 aumento a escala de objetos, 221, 222,
 225
 \author, 13, 14, 16, 17, 165, 166, 185
 Avant Garde (fuente), 274–276
 axesstyle (**PSTricks**), 336
 \axis (**PiCTeX**), 399–403
 babel (paquete), 19–21, 27, 34–36, 58,
 62, 85, 97, 113, 114, 154, 188,
 189, 257, 258
 configuración de **LATeX** para, 448
 conflictos con, 388, 418
 \backmatter, 184
 backslash, 5
 \Bar, 109
 \bar, 109
 \baselinestretch, 45
 Bcenter (entorno), 433
 Bdescription (entorno), 433
 \belowcaptionskip, 241
 Benumerate (entorno), 433
 Bézier
 curvas de, 250, 251
 \bf, 39
 Bflushleft (entorno), 433
 Bflushright (entorno), 433
 \bfseries, 39
 \bibitem, 81–85
 bibliografía, 80, 189
 con **BIBTeX**, 205–211
 \bibliography, 206
 \bibliographystyle, 206, 212–214

- \bibname, 81, 189
BIBT_EX, 205–211
\Big, 115
\big, 115
\bigcap, 107
\bigcup, 107
\Bigg, 115
\bigg, 115
\Biggl, 115, 116
\biggl, 115, 116
\Biggr, 115, 116
\biggr, 115, 116
\Bigl, 115, 116
\bigl, 115, 116
\bigoplus, 107
\bigotimes, 107
\Bigr, 115, 116
\bigr, 115, 116
\bigsqcup, 107
\biguplus, 107
\bigvee, 107
\bigwedge, 107
\binom, 99
Bitemize (entorno), 433, 434
bitmaps (fuentes), 271, 272
Bmatrix (entorno), 119
bmatrix (entorno), 119
\bmod, 114
\boldmath, 125
\boldsymbol, 124, 125
book (estilo), 10, 17, 23, 28, 183
bookman (paquete de fuentes), 276, 278
Bookman (fuente), 274–276
Bookmarks (archivos PDF), 295, 300, 304, 305
bounding box, 227
\boxed, 108
cajas, 50–57, 73, 74, 167
 almacenamiento de, 73, 74
 con líneas, 55
 con párrafos, 52
 con una sola línea de texto, 51
 de anchura nula, 56, 57
 en color, 219, 220
 en entorno **picture**, 248, 249
 invisibles, 56, 57
cambios de página, 46
 en alineaciones, 147
\caption, 192, 239–241, 438
caption2 (paquete), 241, 438, 439
\captionlabeldelim, 439
\captionstyle, 439
caracol de Pascal, 358
caracteres especiales, 42
cartas (estilo letter), 257–260
\cases, 118
casos, 118
\cc, 257, 258
CD (entorno), 156, 157
\cdots, 98
center (entorno), 49, 238, 433
\centering, 49, 238
\centerline, 49
CervanT_EX, 3, 21
\cfrac, 99
chancery (paquete de fuentes), 276
chapter (contador), 75
\chapter, 17, 25, 183, 184, 189
\chapter*, 17, 24, 184
\chaptername, 189
\Check, 109
\check, 109
\circle (entorno **picture**), 246
\circle* (entorno **picture**), 246
\circlenode (**PSTricks**), 368
\circulararc (**PCTEX**), 393
círculos
 en entorno **picture**, 246
citas, 47
\cite, 80–85, 205, 206, 209
 en paquete **achicago**, 214, 215
claves, 75, 84, 85
\cleardoublepage, 46
\clearpage, 46, 237, 238
\cline, 172
\closing, 257, 258
cm (unidad), 26
CM (fuentes), 37, 269, 278, 280
cmr, 284
cmss, 284

- cmtt**, 284
cmyk (modelo de color), 218, 219, 434
\Cnode (**PSTricks**), 367
\cnode (**PSTricks**), 367
codificación
 OT1, 36, 269, 270, 283
 T1, 269, 270, 283
 Cork, 269
 de fuentes, 36, 269, 270, 283
 Knuth, 269
coeficientes binomiales, 99
colección
 AE, 37, 271, 272, 291, 306
 EC, 270, 272, 283
 PSNFSS, 37, 271, 273–284
color (paquete), 19, 182, 216–220, 244, 313, 434
\color, 219
\colorbox, 220
colores
 en **PSTricks**, 313
 modelo **cmyk**, 218, 219, 434
 modelo **gray**, 219, 434, 435
 modelo **rgb**, 218, 434
 paleta de, 219
colortbl (paquete), 182, 434
\columncolor, 435–437
\columnseprule, 429
coma decimal, 97
comandos
 con argumentos, 8
 con argumentos obligatorios, 130
 con un argumento opcional, 131
 definición de nuevos, 72, 73, 129–131
 divisionales, 265–267
 estrella, 24
 en **PSTricks**, 328
 internos de L^AT_EX, 264, 265
 seccionales, 265–267
 simples, 7, 73
 sin argumentos, 129
comillas, 35
 francesas, 35
 inglesas, 35
comment (entorno), 68
Computer Modern Fonts, 37, 269, 278, 280
configuración de L^AT_EX para **babel**, 448
contadores, 27, 75–80
 creados por el usuario, 79
 modificación de, 77
 valor actual de, 77
\contentsname, 189
control sobre cambios de página, 46
\coprod, 107
\copyright, 5, 42
correo electrónico, 267, 268
\cos, 101
\cosec, 113
\cosh, 101
\cot, 101
\cotg, 113
\coth, 101
Courier (fuente), 274, 276
\csc, 101
CTAN, 3
\curraddr, 165
\currentpdfbookmark, 300
curvas cuadráticas de Bézier, 250, 251
curvas paramétricas (**PSTricks**), 357
curvature (**PSTricks**), 344

\dag, 42
dash (**PSTricks**), 316
\dashbox (entorno **picture**), 248, 249
\dataplot (**PSTricks**), 348
\date, 13, 14, 16, 17, 165, 257
\dbinom, 99
\ddag, 42
\dddot, 110
\ddot, 110
\ddot, 109
\decimalpoint, 97
declaraciones globales, 8
\DeclareFixedFont, 285, 286
\DeclareGraphicsExtensions, 228
\DeclareMathOperator, 113
\DeclareMathOperator*, 113
\DeclareTextFontCommand, 41, 285
\dedicatory, 165
\definecolor, 218, 219

- definición
 de colores, 218, 219
 de comandos, 72, 73, 129–131
 de entornos, 262, 263
definition (para `\theoremstyle`),
 153
`\deg`, 101
`\degrees` (**PSTricks**), 365
`\depth`, 51
`description` (entorno), 63, 260, 433
`\det`, 112
`\frac`, 98
`\dgARROWLENGTH` (`pb-diagram`), 162
`\dgARROWPARTS` (`pb-diagram`), 159
`diagram` (entorno), 158–163
 diagramas commutativos
 con paquete `amscd`, 156, 157
 con paquete `pb-diagram`, 158–163
`\dianode` (**PSTricks**), 368
`\dim`, 101
`\dimen`, 388
`\ding`, 281
`dingautolist` (entorno), 282
`\dingfill`, 283
`\dingline`, 283
`dinglist` (entorno), 282
`\displaybreak`, 147
`displaymath` (entorno), 86
`\displaystyle`, 126
 división de fórmulas, 133
 divisiones de un documento, 183
 documentación de paquetes, 449, 450
`\documentclass`, 10, 12, 13, 17, 28,
 32, 164, 252
 documento
 cuerpo del, 10
 fuente, 4
 raíz, 185–187, 443, 444
`\dot`, 109
`\dotfill`, 69–71, 283
`\dotnode` (**PSTricks**), 367
`\dotsb`, 98
`\dotsc`, 98
`\dotsep` (**PSTricks**), 316
`\dotsm`, 98
`\dotso`, 98
`\dotstyle` (**PSTricks**), 327
`\doubleline` (**PSTricks**), 316
`\doublebox`, 432, 433
`\draft` (opción), 12
`\dvipdfm`, 290, 291, 293, 294
`\dvips`, 216, 217, 221, 226, 308
 EC (fuentes), 270, 272, 283
 ejes coordenados
 con `PiCTeX`, 399–403
 con **PSTricks**, 336–338
`\ellipticalarc` (`PiCTeX`), 393, 394
`\em` (unidad), 26
`\em`, 39
`\email`, 165
`\emph`, 38
`\empty` (formato de página), 22, 426
 encabezados, 21–25, 426–428
 demasiado largos, 24
 títulos para los, 183, 190
 vacíos, 25
`\encl`, 257
 enlaces, 293, 295–297
`\enlargethispage`, 47
`\enlargethispage*`, 47
`\ensuremath`, 129, 130
 entornos, 9
 con argumentos obligatorios, 263
 con un argumento opcional, 263
 definición de nuevos, 262, 263
`\enumerate` (paquete), 65, 66
`\enumerate` (entorno), 62–65, 260, 433
`\enumi` (contador), 75
`\enumii` (contador), 75
`\enumiii` (contador), 75
`\enumiv` (contador), 75
 epígrafes, 177, 239–241, 438, 439
`\EPS` (formato gráfico), 226, 227
`\eqnarray` (entorno), 134, 145, 147
`\eqnarray*` (entorno), 145
`\eqref`, 150
`\equation` (contador), 75, 149
`\equation` (entorno), 86, 133
`\equation*` (entorno), 133
 errores, 7, 445–447
 espaciamiento francés, 36

- espacio
 en blanco, 6
 horizontal, 43
 en tablas, 173
 vertical, 6, 44
 en alineaciones, 146
 en tablas, 174, 175
 espiral de Arquímedes, 358
 esquema de fuentes NFSS, 269–271,
 284
 esquinas, 89
 estilo
 amsart, 164–166
 amsbook, 164–166
 artículo (article), 10, 13, 22, 23,
 28, 183
 carta (letter), 10, 257–260
 libro (book), 10, 17, 23, 28, 183
 reporte (report), 10, 18, 22, 23,
 28, 183
 transparencias (slides), 10, 252–
 256
 estilos
 bibliográficos, 212–215
 autor-año, 214
 de capítulos, 430–432
 de documentos, 10
 de numeración para contadores,
 78
 estructura de archivos TDS, 449, 450
eucal (paquete), 123
euro (símbolo), 279
euscript (paquete), 122, 123
\evensidemargin, 29, 31
ex (unidad), 26
\exp, 101
 extensiones de archivos gráficos, 228
- familia
 mono-espaciada, 270, 284
 romana, 270, 284
 sans serif, 270, 284
 typewriter, 270, 284
 familias de fuentes de PSNFSS, 273,
 274
fancybox (paquete), 432–434
- \fancyfoot*, 427, 428
fancyhdr (paquete), 25, 426–428
\fancyhead, 427, 428
\fancyhf, 428
\fancypagesstyle, 428
\fbox, 51, 54, 55, 432
\fboxrule, 55, 220, 249
\fboxsep, 55, 220, 249, 433
\fcolorbox, 220
figure (contador), 75
figure (entorno), 46, 189, 235–241
*figure** (entorno), 236
\figurename, 189, 239
figwindow (entorno), 440, 442
fillstyle (**PSTricks**), 321–325
\findlength (**PlCEX**), 412
\firsthline, 182
\fiverm, 389
\flafter (paquete), 238
\flalign (entorno), 134, 144
*\flalign** (entorno), 144
 flechas, 90
 con **PlCEX**, 414, 415
 en entorno *picture*, 246
 negadas, 92
\fleqn (opción), 147
\flushbottom, 46
\flushleft (entorno), 50, 433
\flushright (entorno), 50, 433
\fncychap (paquete), 430–432
\fnsymbol, 58
\fontenc (paquete), 270, 283
\fontfamily, 273
\footnote (contador), 58, 75
\footnote, 58, 59
\footnotemark, 60
\footnotesize, 39, 164
\footnotetext, 60
\footrulewidth, 427, 428
\footskip, 29, 32
 formato de
 página, 21–25, 28–32
 títulos seccionales, 265–267
 un archivo ‘—.bib’, 207–211
 una página aislada, 24
 una tabla, 167

- simplificación, 172
fórmulas
 desplegadas, 86
 en cajas, 108, 126, 127
 no numeradas, 136
\frac, 98
fracciones, 98
 continuas, 99
\frame (entorno *picture*), 249
\framebox, 51, 55, 57
\framebox (entorno *picture*), 248, 249
framesep (**PSTricks**), 329
\frenchspacing, 36
\frontmatter, 183, 184, 186
fuentes
bitmaps, 271, 272
 AE, 271, 272, 291, 306
 atributos, 269–271
 cambio de, 284
 CM, 37, 269, 278, 280
 codificación de, 269, 270, 283
 EC, 270, 272, 283
 esquema NFSS, 269–271, 284
 familias de, 270
 por defecto, 284
 PostScript, 269, 271–284
 PSNFSS, 271, 273–284
 series de, 270
 tamaños de, 270
 variantes de, 270
 virtuales, 271
funciones matemáticas, 101
gather (entorno), 134, 135
gather* (entorno), 135, 136
gathered (entorno), 134, 141–143
\gcd, 112
Ghostscript, 217, 218
Ghostview, 218, 227
gradangle (**PSTricks**), 324
gradbegin (**PSTricks**), 324
gradend (**PSTricks**), 324
gradient (**PSTricks**), 322
gradlines (**PSTricks**), 324
gradmidpoint (**PSTricks**), 324
gráficas
 centradas, 238
 con **PlCTEX**, 388–425
 con **PSTricks**, 308–387
 con el entorno *picture*, 242–251
 en documentos **LATEX**, 226–234
 en párrafos, 440–442
 en tablas, 234
 numeradas, 239–241
 grafos (**PSTricks**), 367–376
graphics (paquete), 216–218, 221, 311, 417
\graphicspath, 228
graphicx (paquete), 19, 216–218, 221–234, 292, 311, 417
graphpap (paquete), 243, 244
\graphpaper, 243, 244
\Grave, 109
\grave, 109
gray (modelo de color), 219, 434, 435
\grid (**PlCTEX**), 391
grillas
 con **PlCTEX**, 391
 con **PSTricks**, 339, 340
 en entorno *picture*, 243, 244
grosor de líneas
 en **PlCTEX**, 407
 en **PSTricks**, 316
 en entorno *picture*, 242, 243
 en tablas, 176
GSview, 218, 227
guillemets, 35
\guillemotleft, 35
\guillemotright, 35
guiones, 35, 42
harvard (paquete), 214
\hat, 109
\hat, 109
hatchangle (**PSTricks**), 322
hatchcolor (**PSTricks**), 322
hatchsep (**PSTricks**), 322
hatchwidth (**PSTricks**), 322
\hdotsfor, 120
\headheight, 29, 31, 32

- headings** (formato de página), 22, 23,
 426
\headrulewidth, 427, 428
\headsep, 29, 31
\height, 51
Helvetica (fuente), 274, 276
\hfill, 43, 69–71, 283
hipertexto, 289
histogramas (PICTEX), 396
\hline, 169, 182
\hoffset, 29, 30
\hom, 101
\phantom, 56
\href, 296–298
\hrulefill, 69, 70, 283
\hshade (PICTEX), 420–425
\hspace, 43
\hspace*, 43, 71
HTML, 289
\Huge, 39, 164
\huge, 39, 164
\hyperdef, 297
\hyperlink, 296, 298
hyperref (paquete), 291–300, 306
\hyperref, 297, 298
\hypersetup, 293, 294
\hypertarget, 296
\hyphenation, 36, 270, 272
- \idotsint**, 105, 106
idx.tex (archivo), 202
\iiiint, 105, 106
\iiint, 105, 106
\iint, 105, 106
impresión en
 dos caras, 12
 una cara, 12
in (unidad), 26
\include, 185–187, 444
\includegraphics, 226–234, 236, 240,
 241, 267, 297, 298
\includeonly, 186, 187, 444
inclusión de gráficas, 226–234, 297,
 298
\index, 194–205
 espacios en el argumento de, 199
 expresiones varias en, 198
 mayúsculas y minúsculas en, 200
 rangos de páginas en, 199, 200
 rastreo de, 202
 referencias múltiples en, 198
 símbolos acentuados en, 201
 símbolos especiales en, 202
 subtérminos en, 197
\indexname, 189
\indexspace, 193
índice
 alfabético, 189, 193–205
 con *MakeIndex*, 194–205
 de cuadros, 189, 191
 de figuras, 189, 191
 de tablas, 189, 191
 general, 189, 190
índices múltiples, 204
\inf, 112
INITeX, 21, 448
\injlim, 112
\input, 187, 267, 444
\inputenc (paquete), 19, 34, 85, 201,
 270, 272
inserción
 de gráficas, 235–238
 de tablas, 177, 235–238
instalación de paquetes, 449, 450
\int, 105, 106
integrales, 105, 106
\intertext, 138
\intlimits (opción), 106
\it, 39
\item, 62, 193, 196
\itemindent, 261
\itemize (entorno), 62, 260, 433
\itemsep, 261
\itshape, 39
- \ker**, 101
\keywords, 165
Knuth, Donald, 1, 37, 269, 442, 449
\label, 75–77, 84, 133, 150, 152, 239
\labels (PSTTRicks), 336
\labelsep, 261

- \labelwidth, 261
lablst.tex (archivo), 85
Lamport, Leslie, 1
landscape (entorno), 443
landscape (opción), 12
\angle, 115
language.dat (archivo), 448
\language, 21
\LARGE, 39, 164
\Large, 39, 164
\large, 39, 164
\larger, 164
\lasthline, 182
\LaTeX, 5
LATEX 2.09, 2
LATEX 2ϵ , 3
LATEX2HTML, 289
LATEX 3, 3
latexsym (paquete), 88–90, 93
latin1 (opción), 34
latin2 (opción), 34
latin3 (opción), 34
layout (paquete), 28
\lceil, 115
\ldots, 98
\left, 115, 117
\leftarrowfill, 69, 70
\leftline, 50
\leftmargin, 261
\leftmark, 428
\leftroot, 100
leqno (opción), 133, 147
letra
 - caligráfica, 122
 - cursiva, 37
 - gótica, 122
 - inclinada, 37
 - ítálica, 37
 - mono-espaciada (*typewriter*), 37
 - negrilla, 37
 - en modo matemático, 122, 124
 - romana normal, 37
 - sans serif, 37
 - versalitas, 37**letras**
 - apiladas**con** **PlCoEX**, 398
en entorno **picture**, 249
en párrafos, 250
griegas, 94, 277, 279
hebreas, 94
letter (estilo), 10, 257–260
letter (entorno), 257–260
\lfloor, 115
\lg, 101
liftpen (**PSTTricks**), 359
ligaduras, 35, 42
\lim, 112, 113
\liminf, 112, 113
\limits, 103, 104, 108, 112
\limsup, 112, 113
\line (entorno **picture**), 245
linearc (**PSTTricks**), 316
líneas, 69
 - a trozos
 - en **PlCoEX**, 410
 - en **PSTTricks**, 316
 - punteadas
 - en **PlCoEX**, 408
 - en **PSTTricks**, 316**\linebreak**, 45, 132
linecolor (**PSTTricks**), 316
linestyle (**PSTTricks**), 316
\linethickness, 243
linewidth (**PSTTricks**), 316
list (entorno), 260–262
listas
 - con **description**, 63
 - con **enumerate**, 62
 - con **itemize**, 62
 - con incisos o items, 62–66
 - con paquete **enumerate**, 65, 66
 - descriptivas, 63
 - encajadas en listas, 64
 - viñetas en, 62**\listfigurename**, 189
\listfiles, 267, 268
\listoffigures, 189, 191, 240
\listoftables, 189, 191, 240
\listparindent, 261
\listplot (**PSTTricks**), 348
\listtablename, 189

- \ln, 101
 \log, 101
 longitudes, 26
 elásticas, 26, 264
 rígidas, 26, 264
 longtable (paquete), 182, 437, 443
 longtable (entorno), 437
 lspace (paquete), 182, 443
- \mainmatter, 184, 186
 \makeatletter, 27, 264–267
 \makeatother, 27, 264–267
 \makebox, 51, 56, 57
 \makebox (entorno picture), 248, 249
 makeidx (paquete), 194–205
MakeIndex, 194–205
 \makeindex, 189, 194, 197, 204
 \makelabels, 258
 \maketitle, 13, 14, 16, 17, 24, 165
 manejadores (*drivers*), 216, 217
 Marcadores (archivos PDF), 295, 300,
 304, 305
 \marginpar, 61
 \marginparsep, 61
 \marginparwidth, 29, 32, 61
 \marginparsep, 29, 32
 \markboth, 23, 25
 \markright, 23, 25
 math (entorno), 86
 \mathbb, 122–125
 \mathbf, 122, 124
 \mathcal, 122–125
 \mathfrak, 122, 124, 125
 \mathit, 122
 mathpazo (paquete de fuentes), 276,
 279, 280
 mathptm (paquete obsoleto de fuen-
 tes), 277
 mathptmx (paquete de fuentes), 276,
 277, 280
 \mathrm, 122
 \mathscr (opción), 122, 123
 \mathscr, 122–125
 \mathsf, 122
 \mathtt, 122
 matrices, 119–121
- con el entorno array, 121
 filas de puntos en, 120
 máximo número de columnas en,
 120
 pequeñas, 120
 matrix (entorno), 119
 \max, 112, 113
 MaxMatrixCols (contador), 120
 \mbox, 51, 57, 108, 126, 127
 \medskip, 45
 mensajes de error, 7, 445–447
 de L^AT_EX, 445, 446
 de TeX, 446, 447
 METAFONT, 271, 442
 micro-espacios, 43, 111
 MiK^TeX, 21, 87, 270, 272, 273, 449
 \min, 112, 113
 mini-páginas, 57, 60
 minipage (entorno), 57, 60
 mm (unidad), 26
 \mod, 114
 modelos de color, 218, 219, 434
 modo
 de párrafo, 5
 enfático, 38
 ID, 6
 matemático, 6, 86, 87
 normal, 5
 mpfootnote (contador), 75
 multicol (paquete), 32, 429
 multicols (entorno), 429
 multicols* (entorno), 429
 \multicolumn, 171
 multind (paquete), 204
 \multips (PSTricks), 341
 \multiput
 en PjCT_EX, 397
 en entorno picture, 251
 \multirput (PSTricks), 341
 multiline (entorno), 134, 135
 multiline* (entorno), 134
 myheadings (formato de página), 22,
 23, 426
- \nameref, 296
 \naput (PSTricks), 374

- \natbib** (paquete), 214
\nbput (**PSTricks**), 374
\ncangle (**PSTricks**), 369, 374
\ncangles (**PSTricks**), 369
\ncarc (**PSTricks**), 369, 374
\ncbar (**PSTricks**), 369, 374
\nccircle (**PSTricks**), 368, 374
\nccoil (**PSTricks**), 385
\nccurve (**PSTricks**), 369, 374
\ncdiag (**PSTricks**), 369, 374
\ncdiagg (**PSTricks**), 369, 374
\ncline (**PSTricks**), 369, 374
\ncloop (**PSTricks**), 369, 374
\ncput (**PSTricks**), 374
\nczigzag (**PSTricks**), 385
negrilla, 37, 122, 124, 125
 de los pobres (**\pmb**), 125
New Century Schoolbook (fuente), 274–
 276
newcent (paquete de fuentes), 276, 278
\newcolumntype, 178, 180, 436
\newcommand, 72, 73, 129–131
\newcounter, 79
\newenvironment, 262, 263
\newfont, 285, 287, 288
\newline, 45
\newpage, 46
\newpsobject (**PSTricks**), 312, 343
\newsavebox, 74
\newtheorem, 150–155
 en el paquete **amsthm**, 153
 opciones de, 152
\newtheorem*, 154
NFSS (esquema de fuentes), 269–271,
 284
nfssfont.tex (archivo), 286, 287
\nobreakdash, 132
\nocite, 205, 206
\nocite*, 205, 206
\node (**pb-diagram**), 158
nodos y conectores (**PSTricks**), 367–
 376
\noindent, 44
\nointlimits (opción), 106
\nolimits, 103, 104, 108, 112
\nolinebreak, 45
\nonfrenchspacing, 36
\nopagebreak, 46, 47
\normalmarginpar, 61
\normalsize, 39, 164
\nosumlimits (opción), 104, 108
\not, 92
\notag, 135–137, 144
notas
 al pie de página, 58–60
 en mini-páginas, 60
 no permitidas, 59
 marginales, 61
note (entorno), 253, 256
notitlepage (opción), 12
\nouppercase, 428
\numberwithin, 148
numeración
 Alph, 27
 Roman, 27
 alph, 27
 arabic, 27
 roman, 27
 automática, 75
 de fórmulas, 133–150
 a la derecha, 147
 a la izquierda, 147
 opciones para, 147–150
 de gráficas, 239–241
 de notas al pie de página, 58, 59
 de páginas, 27
 de partes de un documento, 184
 de tablas, 177, 239–241
 forzada de fórmulas, 148
 subordinada de fórmulas, 149
 número de columnas, 12, 429
objetos flotantes, 177, 235–238
\oddsidemargin, 29, 31
\oint, 105, 106
\onecolumn (opción), 12
\onecolumn, 32, 429
\oneside (opción), 12, 18, 61
\onlynotes, 256
\onlyslides, 256
opciones de
 MakeIndex, 203

- \documentclass, 12
 \includegraphics, 229
 un comando, 8
 openany (opción), 12, 17, 18
 \opening, 257, 258
 openright (opción), 12
 operadores, 101, 112
 binarios, 88, 89
 con acentos, 114
 con límites inferiores, 112
 grandes, 107
 pre-definidos, 101, 112
 ordinales, 36
 orientación del papel, 12
 \oval (entorno *picture*), 247
 \Ovalbox, 432–434
 \ovalbox, 432, 433
 \ovalnode (**PSTricks**), 368
 \overbrace, 117
 Overfull (advertencia), 132
 overlay (entorno), 253–256
 \overleftarrow, 110
 \overleftrightarrow, 110
 \overline, 110
 \overrightarrow, 110
 \overset, 128
- \P, 42
 page (contador), 75
 \pagebreak, 46
 \pagecolor, 220
 \pagenumbers, 27
 \pageref, 75–77, 84, 133, 150, 239
 \pagestyle, 21, 23–25, 28, 427
 páginas
 con dos columnas, 32, 429
 con varias columnas, 429
 más largas, 46
 no numeradas, 24
 palabras
 apiladas
 con **PTEX**, 398
 en entorno *picture*, 249
 en párrafos, 250
 unión de, 35
- palatino (paquete obsoleto de fuentes), 277
 Palatino (fuente), 274, 276
 paquetes, 3, 18, 449, 450
 estándares de **LATEX 2 ϵ** , 18
 instalación de, 449, 450
 \par, 44, 45
 \parabola (**PSTricks**), 321
 paragraph (contador), 75
 \paragraph, 183, 184
 \parametricplot (**PSTricks**), 357, 358
 parámetros de longitud, 26
 \parbox, 52, 54, 57
 parentequation (contador), 149
 paréntesis, 115–117
 \parindent, 44
 \parskip, 45
 part (contador), 75
 \part, 183, 184, 189
 \part*, 184
 partes de un documento, 183
 partición silábica, 20, 35, 36
 \partname, 189
 pb-diagram (paquete), 158–163
 pc (unidad), 26
 \pdfannot, 305, 306
 \pdfbookmark, 300
 \pdfcompresslevel, 302
 \pdfdecimaldigits, 302
 \pdfdest, 304, 305
 \pdforigin, 302
 pdf**LATEX**, 3, 272, 290, 291, 301–306
 \pdfoutline, 304, 305
 \pdfoutput, 302
 \pdfpageheight, 302
 \pdfpagewidth, 302
 \pdfpkresolution, 302
 pdf**TEX**, 3, 291, 301–306
 pdftex (opción), 292
 pdftex.cfg (archivo), 301–303
 \pdfvorigin, 302
 \phantom, 56
 picinpar (paquete), 440–442
PTEX, 388–425
 picture (entorno), 242–251

- pie de página, 22, 426–428
pifont (paquete de fuentes), 281–283
plain (estilo bibliográfico), 212
plain (formato de página), 22, 426
plain (para `\theoremstyle`), 153
plotstyle (**PSTricks**), 348
pmatrix (entorno), 119
`\pmb`, 125
`\pmod`, 114
PNG (formato gráfico), 298
\pnode (**PSTricks**), 367
`\pod`, 114
poor man's bold, 125
portrait (opción), 12
PostScript, 217, 308
 - fuentes, 272–284
 - sintaxis de funciones, 351–358`\pounds`, 42
`\ppleuro`, 279
`\Pr`, 112
preámbulo, 13, 28
`\printindex`, 195, 204
`\prod`, 107
`\projlim`, 112
proof (entorno), 154, 155, 189
`\proofname`, 189
`\protect`, 192
`\providemode`, 72
`\ps`, 257, 258
`\psarc (PSTricks), 318
\psarcn (PSTricks), 319
\psaxes (PSTricks), 336
\psccurve (PSTricks), 344, 348
\pscharpath (PSTricks), 387
\pscircle (PSTricks), 318
\pscirclebox (PSTricks), 329
\pscoil (PSTricks), 383
\pscurve (PSTricks), 344, 348
\pscustom (PSTricks), 359
\psdblframebox (PSTricks), 329
\psdiabox (PSTricks), 329
\psdiamond (PSTricks), 317
\psdisk (PSTricks), 319
\psdots (PSTricks), 327
\psecurve (PSTricks), 344, 348
\psellipse (PSTricks), 320`
- `\psframe (PSTricks), 317
\psframebox (PSTricks), 329
\psgrid (PSTricks), 339, 340
\psline (PSTricks), 314, 315
PSNFSS (fuentes), 37, 271, 273–284
\psovalbox (PSTricks), 329
\pspicture (entorno PSTricks), 311
\psplot (PSTricks), 351–356
\pspolygon (PSTricks), 317
\psset (PSTricks), 311
pst-all (paquete), 311, 313
pst-char (paquete), 311
pst-coil (paquete), 311
pst-grad (paquete), 311
pst-node (paquete), 311
pst-plot (paquete), 311
pst-text (paquete), 311
pst-tree (paquete), 311
pstcol (paquete), 313
\pstextpath (PSTricks), 386
\pstree (PSTricks), 377–382
\psttriangle (PSTricks), 317
\pstribox (PSTricks), 329
PSTricks, 308–387
\pswedge (PSTricks), 319
\pszigzag (PSTricks), 383
pt (unidad), 26
punto decimal, 97
puntos, 34

 - en PSTricks, 327
 - en modo matemático, 98
 - suspensivos, 34, 98puntuación, 34–36
\put
 - en PGF, 394, 395
 - en entorno picture, 244\putrectangle (PGF), 396`
- `\qbezier`, 250, 251
`\qedhere`, 155
`\quedsymbol`, 155
`\qline (PSTricks), 315
\quad, 43
\quad, 43
quotation (entorno), 47, 48
quote (entorno), 47, 48`

- \raggedbottom, 46
 \raggedleft, 50
 \raggedright, 50
 raíces, 100
 \raisebox, 55
 \raisetag, 150
 \rangle, 115
 rastreo
 de claves, 84, 85
 de \index, 202
 \rceil, 115
 \readdata (PSTricks), 348
 rectángulos (PiCTEX), 396
 redefinición
 de comandos, 72
 de entornos, 263
 de rótulos, 188
 \ref, 75–77, 84, 133, 150, 152, 239
 referencias
 bibliográficas, 80–83
 cruzadas, 75–80, 84, 150
 \reflectbox, 221, 223
 reflexión de objetos, 223
 \refname, 81, 189
 relaciones
 binarias, 88, 89, 91
 con módulo, 114
 de congruencia, 114
 negadas, 92
 relleno, 69–71
 relleno de regiones (PSTricks), 321–325, 359–364
 remark (para \theoremstyle), 153
 \ renewcommand, 72
 \ renewenvironment, 263
 report (estilo), 10, 18, 22, 23, 28, 183
 reqno (opción), 147
 \ resizebox, 221, 222, 225, 242
 resortes (PSTricks), 383–385
 \ reversemarginpar, 61
 \rfloor, 115
 \rgb (modelo de color), 219, 434
 \right, 115, 117
 \rightarrowfill, 69, 70
 \rightline, 50
 \rightmargin, 261
 \rightmark, 428
 \rm, 39
 \rmdefault, 284
 \rmfamily, 39, 284
 \rnode (PSTricks), 367
 Roman (tipo de numeración), 27
 \Roman, 79
 roman (tipo de numeración), 27
 \roman, 78
 rosa de ocho pétalos, 357
 rotación de objetos, 182, 223–225, 443
 en PiCTEX, 416, 417
 en PSTricks, 331–333
 \rotatebox, 182, 221, 223–225, 417
 rótulos para tablas o gráficas, 239–241, 438, 439
 \rowcolor, 435
 \rput (PSTricks), 331
 \rput* (PSTricks), 333
 \rule, 69
 \S, 5, 42
 sangrías, 44
 \savebox, 74
 \savedata (PSTricks), 348
 \sbox, 74
 \sc, 39
 \scalebox, 221, 222, 242
 \scriptscriptstyle, 126
 \scriptsize, 39, 164
 \scriptstyle, 126
 \scshape, 39
 \sec, 101
 secnumdepth (contador), 184, 185
 section (contador), 75
 \section, 13, 16, 17, 23, 25, 183, 184
 \section*, 16, 24, 184
 \see, 201
 \selectfont, 273
 \selectlanguage, 20
 \sen, 113
 \senh, 113
 separación
 de columnas, 145
 de expresiones matemáticas, 132
 de palabras, 20, 35, 36

- \setbars (**PICTEX**), 418, 419
 \setcaptionmargin, 438
 \setcaptionwidth, 438
 \setcoordinatesystem (**PICTEX**), 389,
 390
 \setcounter, 77
 \setdashes (**PICTEX**), 410, 411
 \setdashesnear (**PICTEX**), 411, 412
 \setdots (**PICTEX**), 408
 \setdotsnear (**PICTEX**), 411
 \sethistograms (**PICTEX**), 396
 \setlength, 26, 30
 \setlinear (**PICTEX**), 392
 \setplotarea (**PICTEX**), 390
 \setplotsymbol (**PICTEX**), 407
 \setquadratic (**PICTEX**), 404–406
 \setshadegrid (**PICTEX**), 420
 \setshadesymbol (**PICTEX**), 420
 \setsolid (**PICTEX**), 408
 \sf, 39
 \sfdefault, 284
 \sffamily, 39, 284
 shadow (**PSTricks**), 326
 shadowangle (**PSTricks**), 326
 \shadowbox, 432, 433
 shadowcolor (**PSTricks**), 326
 shadowsize (**PSTricks**), 326
 shareware, 3
 \shortstack (entorno *picture*), 249,
 250
 showidx (paquete), 202
 showkeys (paquete), 84
 showorigin (**PSTricks**), 336
 showpoints (**PSTricks**), 344, 348
 siam (estilo bibliográfico), 212
 \sideset, 104, 108
 \signature, 257, 258
 sílabas, 35, 36
 símbolo
 \$, 6, 86
 %, 9, 310
 &, 167
 @, 264, 265
 símbolos
 ASCII, 4
 de admiración, 33, 42
 de agrupación, 115–117
 de interrogación, 33, 42
 de una fuente, 286–288
 especiales, 7
 especiales para *MakeIndex*, 202
 matemáticos, 88–94
 en negrilla, 122, 124, 125
 sobre símbolos, 128
 varios, 93
 \sin, 101
 \sinh, 101
 syntax PostScript, 351–358
 \skiplevel (**PSTricks**), 382
 \sl, 39
 slide (entorno), 253–256
 slides (estilo), 10, 252–256
 slides (opción), 272
 \slshape, 39
 \SMALL, 164
 \Small, 164
 \small, 39, 164
 \smaller, 164
 \smallint, 105
 smallmatrix (entorno), 120
 \smallskip, 45
 \smash, 100
 sombras (**PSTricks**), 326
 sombreado de regiones
 con **PICTEX**, 420–425
 con **PSTricks**, 321–325, 359–364
 sombreado gradual (**PSTricks**), 321,
 322, 324, 325
 spanish (opción del paquete *babel*),
 21, 27, 34–36, 58, 62, 85, 97,
 113, 114, 154, 188, 189, 258
 spanish.1df (archivo), 21
 \SpecialCoor (**PSTricks**), 365, 366
 split (entorno), 134, 139–141
 \spextext, 36
 \sqrt, 100
 \stack (**PICTEX**), 398
 \stackrel, 128
 \startrotation (**PICTEX**), 416, 417
 \stepcounter, 78, 80
 \stoprotation (**PICTEX**), 416, 417
 \strut, 56, 57

- subarray** (entorno), 104, 108
subequations (entorno), 149
\subfile, 443, 444
subfiles (paquete), 443, 444
 subíndices, 97
\subitem, 193, 196
\subjclass, 165
\subparagraph, 183, 184
subparagraph (contador), 75
\subpdfbookmark, 300
subsection (contador), 75
\subsection, 13, 16, 25, 183, 184
\subsection*, 16, 24
\substack, 103, 104, 108
\subsubitem, 193, 196
subsubsection (contador), 75
\subsubsection, 183, 184
\sum, 103, 107
 sumas, 103
 sumatorias, 103
\sup, 112
 superíndices, 97
\suppressfloats, 237, 238
 sustitución de fuentes, 38
\symbol, 287, 288
- \tabcolsep**, 173, 177
 tabla de contenido, 190
 tablas, 167–182
 @-expresiones en, 177
 a color, 182, 434–437
 centradas, 238
 con el paquete *array*, 178–180
 con filas especiales, 171
 con líneas, 169, 172, 173
 con párrafos, 170
 en párrafos, 440
 extensas, 182, 437
 grosor de líneas en, 176
 líneas horizontales en, 169, 172
 líneas verticales en, 173
 numeradas, 239–241
 rotadas, 182, 443
 simplificación del formato de, 172
 texto alrededor de, 181
- table** (contador), 75
- table** (entorno), 46, 177, 189, 235–241
table* (entorno), 236
\tablename, 189, 239
\tableofcontents, 189, 190
tabular (entorno), 167–182
tabular* (entorno), 167
tabwindow (entorno), 440
\tag, 148
\tag*, 148
 tamaño
 de la letra, 12, 39–41, 126, 164
 de los símbolos, 126
 del papel, 12
\tan, 101
\tanh, 101
\tbinom, 99
\TC (PSTricks), 379
\Tc (PSTricks), 379
\Tcircle (PSTricks), 379
\Tdial (PSTricks), 379
\Tdot (PSTricks), 379
 TDS (estructura de archivos), 449, 450
techexplorer, 289
 teoremas, 150
\TeX, 5
\TeX, 1
TeX font metrics, 271
\TeX4ht, 289
\texorpdfstring, 300
\text, 102
\textasciicircum, 42
\textasciitilde, 42
\textbackslash, 42
\textbar, 42
\textbf, 37
\textbullet, 42
\textcircled, 42
\textcolor, 219, 220
\textemdash, 42
\textendash, 42
\textexclamdown, 42
\textheight, 29, 30
\textit, 37
 texto

- a lo largo de curvas (**PSTricks**), 386
 cargado a la derecha, 50
 cargado a la izquierda, 50
 centrado, 49
 con efectos especiales (**PSTricks**), 387
 en color, 219, 220
 en expresiones matemáticas, 102
 enmarcado (**PSTricks**), 329
 enriquecido, 289
 subrayado, 47
`\textquestiondown`, 42
`\textquotedblleft`, 42
`\textquotedblright`, 42
`\textquoteright`, 42
`\textquoteright`, 42
`\textregistered`, 42
`\texttt`, 37, 284
`\textsc`, 37
`\textsf`, 37, 284
`\textsl`, 37
`\textstyle`, 126
`\texttrademark`, 42
`\texttt`, 37, 284
`\textwidth`, 29, 30
`\Tf` (**PSTricks**), 379
`\Tfan` (**PSTricks**), 379
`\tfrac`, 98
`\tg`, 113
`\tgh`, 113
`\thanks`, 16, 165
`\thebibliography` (entorno), 80–83, 189
`\thechapter`, 79
`\thecontador`, 77
`\theequation`, 148, 149
`\thefootnote`, 58
`\theindex` (entorno), 189, 193, 196
`\theoremstyle`, 153, 154
`\thepage`, 79, 428
`\theparentequation`, 149
`\thesection`, 79, 148
`\thicklines`, 242
`\thinlines`, 242, 243
`\thispagestyle`, 24, 428
- `\text` (**PSTricks**), 336
`\textsize` (**PSTricks**), 336
`\textstyle` (**PSTricks**), 336
`\Tilde`, 109
`\tilde`, 109
 tildes, 33
 en modo matemático, 109
`\times` (paquete obsoleto de fuentes), 277
`Times` (fuente), 274, 276
`\tiny`, 164
`\tiny`, 39, 164
 tipos de
 comandos, 7
 documentos, 10
 fuentes, 37
 impresión, 12
 letra, 37, 40, 41
 en modo matemático, 122, 277, 279, 280
 nodos (**PSTricks**), 367
 numeración, 27
 para contadores, 78
`\title`, 13, 14, 16, 17, 165, 166, 185
`\titlepage` (opción), 12
 títulos abreviados, 183, 190
`\tput` (**PSTricks**), 374
`\Tn` (**PSTricks**), 379
`\today`, 257
`\topmargin`, 29, 31
`\totalarclength` (**PCTEX**), 412
`\totalheight`, 51
`\Toval` (**PSTricks**), 379
`\Tp` (**PSTricks**), 379
`\Tr` (**PSTricks**), 379
 transformaciones de Möbius, 130, 131
`\translator`, 165
 traslación vertical de cajas, 55
 trazado de curvas
 con **PCTEX**, 404–406
 con **PSTricks**, 344–358
 en entorno `picture`, 250, 251
 paramétricas (**PSTricks**), 357, 358
`\trinode` (**PSTricks**), 368
`\trput` (**PSTricks**), 374
`\tt`, 39

- \ttdefault, 284
 \ttfamily, 39, 284
 \Ttri (**PSTricks**), 379
 TUG (*TEX Users Group*), 449
 \tvput (**PSTricks**), 374
 twocolumn (opción), 12, 32, 429
 \twocolumn, 32, 429
 twoide (opción), 12, 46, 61
 \unaccentedoperators, 113, 114
 \underbrace, 117
 Underfull, 235
 \underleftarrow, 110
 \underleftrightarrow, 110
 \underline, 47, 110
 \underrightarrow, 110
 \underset, 128
 unidades de medida, 26
 unión de palabras, 35
 \unitlength, 242, 245
 unsrt (estilo bibliográfico), 212
 \uproot, 100
 \uput (**PSTricks**), 334
 \uput* (**PSTricks**), 334
 \url, 297
 \usebox, 74
 \usepackage, 3, 18, 267
- \varinjlim, 112
 \varliminf, 112
 \varlimsup, 112
 \varprojlim, 112
 \Vec, 109
 \vec, 109
 \vector (entorno **picture**), 246
 ventanas en párrafos, 440–442
 \verb, 67
 \verb*, 67
 verbatim (paquete), 67, 68
 verbatim (entorno), 67, 68
 verbatim* (entorno), 67, 68
 \verbatiminput, 68
 versalitas, 37
 \Vert, 115
 \vert, 115
 \vfill, 71
 viñetas, 62
 \vline, 173
 \matrix (entorno), 119
 \matrix (entorno), 119
 \voffset, 29, 30
 volados, 36
 \vphantom, 56, 174, 175
 \vshade (**PICTEX**), 420–425
 \vspace, 44, 45
 \vspace*, 45
 \widehat, 110
 \widetilde, 110
 \width, 51
 window (entorno), 440, 441
 WinEdt, 195, 206, 217
 \xleftarrow, 128
 \xrightarrow, 128
 Zapf Chancery (fuente), 274, 276, 286
 Zapf Dingbats (fuente), 274
 zigzags (**PSTricks**), 383–385

NOTA:

Este trabajo se realizó con la intención principal de preservar y conservar el contenido de la obra por medio de su digitalización. No se efectuó ningún tipo de modificación directa sobre su contenido ni sobre su estructura original.

No obstante, también se intenta acercar el material a una mayor cantidad de estudiantes que de otra forma no podrían hacerlo.

Es conveniente aclarar que aunque en ningún momento se consultó con el autor del libro para la elaboración de esta versión, no se está recibiendo, ni se recibirá, ningún tipo de lucro por la misma.

Escaneo, compilación y optimización por:

Rey Ernesto Martínez

Bogotá, Colombia

Marzo de 2013

“Porque... compartir conocimiento no es delito”

Para producir libros, artículos y otros documentos tenidos de calidad artística, Ud. debe ingresar a *El Universo L^T_EX*. El presente libro está concebido como guía de aprendizaje y manual de referencia para L^T_EX 2.1, y suple las necesidades tanto de principiantes como de T_EX-nócratas consumados.

Después del éxito de la primera edición, en esta muy esperada segunda edición se incluye material adicional y un CD con útil y variado software.

- Descripción completa de las herramientas de L^T_EX 2.1 y de numerosos paquetes que se ejecutan en su ambiente. Con más de 450 ejemplos, e ilustraciones a todo color.
- Manejo de textos matemáticos con el paquete *amsmath* y diagramas commutativos con el paquete *pb-diagram*.
- Elaboración de gráficas con los paquetes *PSTricks* y *PC^L_EX*.
- Creación de documentos L^T_EX interactivos con el programa *pdfL^T_EX* y el paquete *hyperref*.
- Elaboración de índices alfabéticos con el programa *MakeIndex* y de bibliografías con el programa *B^L_EX*.
- Uso de fuentes alternas PostScript.



Incluye una espléndida versión interactiva de *El Universo L^T_EX* y abundante software de dominio público.

ISBN 958-701-050-4

9 789587 010502