



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ACATLAN

CREANDO PRESENTACIONES CON LA CLASE BEAMER DE LATEX

TESIS

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA:

LÓPEZ GARCÍA, HORTENCIA ADORACIÓN

ASESOR: LÓPEZ GARCÍA, FRANCISCO MARCOS

MÉXICO, D. F.

2007



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Creando presentaciones con la clase Beamer de L^AT_EX

Hortencia Adoración López García

12 de Noviembre de 2007

Con todo mi amor y cariño dedico este trabajo a mi madre por estar conmigo en todo momento y ser siempre mi abrigo en los momentos difíciles.

A mis hermanos Marcos, Sagrario, Alejandra y Luis por confiar y apoyarme para concluir mi carrera. A ellos gracias por su infinito amor y por estar siempre pendientes de mi.

A mis sinodales por haber revisado este trabajo y contribuir a mejorarlo con sus consejos y sugerencias.

A la Universidad Nacional Autónoma de México por acogerme en su seno y permitirme realizar mis sueños brindándome la oportunidad de estudiar la carrera de Matemáticas Aplicadas y Computación con lo cual podré ejercer como profesional dando mi mayor esfuerzo y poniendo en alto su nombre.

Índice general

1.	Introducción	1
I.	Empezando	5
2.	Instalación de programas y paquetes	6
2.1.	Programas necesarios	6
2.2.	Instalación de paquetes	6
	MiKTeX	6
2.3.	Instalación en un árbol de texmf	6
2.4.	Actualizando la instalación	7
2.5.	Comprobando la instalación	7
2.6.	Compatibilidad con otros paquetes y clases	7
2.7.	Versiones y dependencias	10
3.	Guía para crear una presentación sencilla	11
3.1.	Plantilla-solución	11
3.2.	Estructurando un marco	17
3.3.	Cambiando la manera en que se ven las cosas: los temas	17
3.4.	Cambiando la manera en que se ven las cosas: colores y fuentes	17
4.	Flujo de trabajo para crear una presentación en Beamer	18
4.1.	Paso Uno: Organización de archivos	18
4.2.	Paso Dos: Estructurando la presentación	18
4.3.	Paso Tres: Creando un archivo PDF o un archivo PostScript	19
	Creando un archivo PDF	19
	Creando un archivo PostScript	19
	Formas de mejorar la velocidad de compilación	20
4.4.	Paso Cuatro: Creando marcos	20
4.5.	Paso Cinco: Probando la presentación	20
4.6.	Paso Seis: Creando un folleto	21
	Creando el folleto	21
	Impresión del folleto	21
5.	Directrices para crear presentaciones	22
5.1.	Estructurando una presentación	22
	Conocer las restricciones de tiempo	22
	Estructura global	22
	Estructura del marco	24
	Elementos interactivos	26
5.2.	Usando gráficos	27
5.3.	Usando animaciones y transiciones	27
5.4.	Elegiendo temas apropiados	27
5.5.	Elegiendo colores apropiados	28
5.6.	Elegir fuentes y atributos de fuentes apropiados	28
	Tamaño de la fuente	28

	Familias de fuente	30
	Forma de la fuente: Italic y Small Capital	31
	Peso de la fuente	32
6.	Plantillas solución	33
II. Construyendo una Presentación		35
7.	Creando marcos	36
7.1.	El ambiente frame.	36
7.2.	Componentes de un marco	41
	El encabezado y el pie de página	42
	Las barras laterales	45
	Las barras de navegación	47
	Los símbolos de navegación	50
	El logotipo	52
	El título del marco	52
	El fondo	54
7.3.	Medidas de los márgenes	56
7.4.	Restringiendo las diapositivas de un marco	56
8.	Creando Cubiertas	58
8.1.	El comando pause	58
8.2.	Concepto general de especificación de cubierta	59
8.3.	Comandos con especificación de cubierta	60
8.4.	Ambientes con especificación de cubierta	63
8.5.	Cambio dinámico de texto o imágenes	64
8.6.	Especificaciones de cubierta avanzadas	66
	Fabricación de comandos y ambientes conscientes de especificación de cubierta	66
	Especificaciones de modo	68
	Especificaciones de acción	69
	Especificaciones incrementales	70
9.	Estructurando una Presentación: La Estructura Global Estática	73
9.1.	Añadiendo una página de título	73
9.2.	Agregando secciones y subsecciones	75
9.3.	Agregando partes	77
9.4.	Dividiendo un curso en conferencias	79
9.5.	Agregando una tabla de contenidos	80
9.6.	Agregando una bibliografía	83
9.7.	Agregando un apéndice	85
10.	Estructurando una Presentación: La Estructura Global Interactiva	86
10.1.	Agregando hiperligas y botones	86
10.2.	Repetir un marco en un punto posterior.	89
10.3.	Añadiendo una ampliación anticipada	90
11.	Estructurando una presentación: Estructura Local	92
11.1.	Listados, enumeraciones y descripciones	92
11.2.	Resaltamiento	97
11.3.	Ambientes de bloque	99
11.4.	Ambientes tipo teorema	101
11.5.	Texto enmarcado	105
11.6.	Figuras y tablas	108
11.7.	Dividiendo un marco en múltiples columnas.	109
11.8.	Colocando texto y gráficos absolutamente	110
11.9.	Texto verbatim y fragile	110
11.10.	Abstract	111
11.11.	Verse, quotation, quote.	111
11.12.	Pies de página	113

12.	Gráficos	114
12.1.	Incluyendo gráficos en archivos externos versus gráficos en línea.	114
12.2.	Incluyendo archivos de gráficos con extensión <code>.eps</code> o <code>.ps</code>	115
12.3.	Incluyendo archivos gráficos con extensión <code>.pdf</code> , <code>.jpg</code> , <code>.jpeg</code> o <code>.png</code>	115
12.4.	Incluyendo archivos gráficos con extensión <code>.mps</code>	115
12.5.	Incluir archivos gráficos con extensión <code>.mmp</code>	116
13.	Animaciones, Sonidos y Transiciones de Diapositiva	117
13.1.	Animaciones	117
	Incluyendo archivos de animación externos	117
	Animaciones creadas al mostrar diapositivas en sucesión rápida	120
	Incluyendo animaciones externas que residen en múltiples archivos de imagen	122
13.2.	Sonidos	123
13.3.	Transiciones de diapositiva	125
III.Cambiando la Apariencia de las Cosas		129
14.	Temas	130
14.1.	Cinco sabores de temas	130
14.2.	Temas de presentación sin barras de navegación	131
14.3.	Temas de presentación con barra de navegación tipo árbol.	135
14.4.	Temas de presentación con barra lateral con la tabla de contenido	137
14.5.	Temas de presentación con un mini marco de navegación	139
14.6.	Temas de presentación con tablas de sección y subsección	142
14.7.	Temas de presentación incluidas para tener compatibilidad	144
15.	Temas Internos, Temas Externos y Plantillas	145
15.1.	Temas internos	145
15.2.	Temas externos	148
15.3.	Cambiando las plantillas para elementos de una presentación	153
	Resumen del administrador de plantillas de Beamer	153
	Usando las plantillas de beamer	155
	Estableciendo plantillas de Beamer	156
16.	Colores	160
16.1.	Temas de color	160
	Temas de color predefinidos y de propósito especial	160
	Temas de color completos	162
	Temas de color internos	166
	Temas de color externos	167
16.2.	Cambiando los colores de elementos de una presentación	169
	Perspectiva general del administrador de color de Beamer	169
	Usando colores de Beamer	170
	Estableciendo colores de Beamer	171
16.3.	El color del texto matemático	173
16.4.	Las paletas de color	173
16.5.	Miscelánea de colores	174
16.6.	Efectos de transparencia	175
17.	Fuentes	178
17.1.	Temas de Fuente	178
17.2.	Haciendo cambio de fuente sin usar temas de fuente.	181
	Elegiendo un tamaño de fuente para el texto normal.	181
	Elegiendo una familia de fuente.	182
	Elegiendo una codificación de fuente.	182
17.3.	Cambiando las fuentes de los elementos de una presentación.	183
	Resumen del administrador de fuentes de Beamer	183
	Usando fuentes de Beamer	183
	Estableciendo fuentes de Beamer	184

IV.Creando material de apoyo.	187
18. Añadiendo notas para el expositor	188
18.1. Especificando el contenido de la nota.	188
18.2. Especificando contenido para múltiples notas.	190
18.3. Especificando qué notas y marcos se van a mostrar.	190
19. Creando transparencias.	192
20. Creando folletos y Notas de la Conferencia	194
20.1. Creando folletos usando el modo handout	194
20.2. Creando folletos usando el modo article.	194
Iniciando el modo article	194
Flujo de trabajo	197
Incluyendo diapositivas de la versión de presentación en la versión de artículo. . .	197
20.3. Detalles sobre los Modos	198
21. Tomando ventaja de pantallas múltiples	202
21.1. Mostrando notas en la segunda pantalla	202
21.2. Mostrando material en un segundo modo en la segunda pantalla.	202
21.3. Mostrando la diapositiva anterior en la segunda pantalla.	204
V. Los Cómo	205
22. Cómo descubrir cosas por partes	206
22.1. Descubriendo una enumeración por partes	206
22.2. Resaltando el objeto actual en una enumeración	206
22.3. Cambiando el símbolo que aparece en una enumeración.	207
22.4. Descubriendo por partes las fórmulas etiquetadas	208
22.5. Descubriendo una tabla por renglones	208
22.6. Descubriendo una tabla por columnas	209
23. Conclusiones	209

1. Introducción

El uso de \LaTeX se ha extendido rápidamente en un amplio sector del ámbito científico, convirtiéndose en la referencia obligada para la comunicación y participación en congresos internacionales, además es el lenguaje requerido por las revistas internacionales científicas para la publicación de artículos académicos.

\LaTeX es un procesador de textos basado en un lenguaje de marcado. Un *lenguaje de marcado* o *lenguaje de marcas* es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

Un ejemplo de cómo funciona el lenguaje de marcado puede observarse en el dictado de viva voz de un documento a una persona que lo transcribe a máquina:

Ponga estilo de carta, ponga comillas, ponga mayúsculas, Estimado Juan, ponga dos puntos, aparte, sangría, ponga primera letra mayúscula, te escribo esta carta, ponga negrillas, de forma muy urgente, cierre negrilla, ya que no me has enviado... etc.

Históricamente, el proceso de marcado se usaba y se usa en la industria editorial y de la comunicación, así como entre autores, editores e impresores. Actualmente el lenguaje de marcado más extendido es el HTML, fundamento de la red mundial (World Wide Web).

Específicamente, \LaTeX está formado por un conjunto de macros de \TeX escritos inicialmente por Leslie Lamport en 1984 (conocido por ser el desarrollador inicial del sistema de formateo de textos \LaTeX y BibTeX), con la intención de facilitar el uso del lenguaje de composición tipográfica creado por Donald Knuth. **\LaTeX es utilizado para la elaboración de artículos académicos, tesis, libros técnicos, dado que la calidad tipográfica de los documentos realizados con \LaTeX es comparable a la de una editorial científica de primera línea.** Un aspecto muy importante es que, a diferencia del conjunto de programas Office de Microsoft, \LaTeX es software libre bajo la Licencia Pública del Proyecto \LaTeX (LPPL) aunque no es una licencia pública general (GPL) ya que no contiene todos los términos necesarios para ser considerada así.

\LaTeX presupone una filosofía de trabajo diferente a la de los procesadores de texto habituales (conocidos como WYSIWYG, es decir, “lo que ves es lo que obtienes”) y se basa en comandos. Tradicionalmente, este aspecto se ha considerado una desventaja (probablemente la única). Sin embargo, \LaTeX a diferencia de los procesadores de texto de tipo WYSIWYG, permite a quien escribe un documento centrarse exclusivamente en el contenido, sin tener que preocuparse de los detalles del formato. Además tiene capacidades gráficas para representar ecuaciones, fórmulas complicadas, notación científica e incluso musical, permite estructurar fácilmente el documento (con capítulos, secciones, notas, bibliografía, índices analíticos, etc.), lo cual lo hace extremadamente confortable y eficiente para artículos académicos y libros técnicos.

Con \LaTeX , la elaboración del documento requiere normalmente dos etapas: en la primera hay que crear mediante cualquier editor de texto plano un fichero fuente que, con las órdenes y comandos adecuados, contenga el texto que queremos imprimir. La segunda consiste en procesar este fichero; el procesador de textos interpretará las órdenes escritas en él y compilará el documento, dejándolo preparado para que pueda ser enviado a la salida correspondiente, ya sea la pantalla o la impresora. Ahora bien, si se quiere añadir o cambiar algo en el documento, se deberá hacer los cambios en el fichero fuente y procesarlo de nuevo. Esta idea, que puede parecer poco práctica a priori, resultará conocida a los que estén familiarizados con el proceso de compilación que se realiza con los lenguajes de programación de alto nivel (C, C++, etc.), ya que es completamente análogo.

El modo en que \LaTeX interpreta la *forma* que debe tener el documento es mediante etiquetas. Por ejemplo, “\documentclass{article}” le dice a \LaTeX que el documento que va a procesar es un artículo. Puede resultar extraño que hoy en día se siga usando algo que no es WYSIWYG, pero las características de \LaTeX siguen siendo muchas y muy variadas. También hay varias herramientas (aplicaciones) que

ayudan a una persona a escribir estos documentos de una manera más visual (LyX, TeXmacs y otros). A estas herramientas se les podría llamar WYSIWYM (“lo que ves es lo que quieres decir”).

Una de las ventajas de \LaTeX es que la salida que ofrece es siempre la misma, con independencia del dispositivo (impresora, pantalla, etc.) o el sistema (Windows, MacOS, Unix, GNU/Linux, etc.) y puede ser exportado muy fácilmente a partir de una misma fuente a numerosos formatos: Postscript, PDF, SGML, HTML, RTF, etc. Existen distribuciones de \LaTeX para todos los sistemas operativos más extendidos, que incluyen todo lo necesario para trabajar. Hay, por ejemplo, programas para Windows como TeXnicCenter, para Linux como Kile, o para MacOS como TeXShop, todos con licencia GPL. Existe además un editor multiplataforma (MacOS, Win y Unix) llamado Texmaker, que también tiene licencia GPL.

En particular, BEAMER es una clase (paquete) de \LaTeX que sirve para crear presentaciones (**con un enfoque científico**) que usan un proyector, y se apoya con elementos multimedia como imágenes, audio y video, pero también se puede usar para crear acetatos. Funciona con pdf \LaTeX , dvips y LyX. BEAMER fue creado por Till Tantau principalmente en su tiempo libre, aunque muchas personas le han ayudado a mejorarlo con sugerencias, correcciones, parches, nuevos temas, etc. De hecho, la mayoría del desarrollo se inició por peticiones específicas e informes de errores. Sin esta retroalimentación, BEAMER todavía sería una pequeña colección privada de macros que harían más fácil el uso de la clase seminar de \LaTeX .

El propósito de este trabajo es presentar el manual de BEAMER en idioma español con el fin de facilitar el uso del mismo a la comunidad científica de habla hispana con conocimiento en el lenguaje de programación \LaTeX , y lo más importante: se realizó un tutorial que enseña el uso de la clase BEAMER de \LaTeX .

Primero se procedió con la traducción y actualización de dicho manual sin perjudicar la esencia del mismo, después se estudió su funcionamiento, se revisaron y compilaron los ejemplos existentes, para finalmente realizar el tutorial. El proyecto siguió los siguientes pasos:

- exportación del manual a un traductor básico.
- introducción de lo obtenido en el paso anterior a un procesador de texto plano.
- revisión por capítulo de la coherencia de la información mostrada.
- corrección manual de los párrafos que así lo requirieran, cuidando y respetando código donde lo había.
- uso de \LaTeX para darle formato y estilo a la tesis, haciendo la correspondiente compilación, obteniendo el resultado final en formato PDF.
- se generaron alrededor de 130 archivos en formato PDF que muestran todas las plantillas disponibles en BEAMER.
- se realizó el tutorial que contiene las indicaciones necesarias para preparar una presentación en BEAMER.

Se anexa un CD que contiene el software libre necesario para el funcionamiento de BEAMER, además contiene el archivo fuente en \LaTeX de la presente tesis, los 130 archivos en formato PDF que muestran todas las plantillas disponibles, el tutorial y algunos ejemplos reales de presentaciones hechas por investigadores de la UNAM en Congresos Internacionales, así como presentaciones hechas por alumnos de la UNAM para obtener el grado de Doctor.

La lista de características que posee BEAMER es bastante larga (desafortunadamente, así de grande podría llegar a ser la lista de problemas). Las más importantes son:

- Se puede usar con pdf \LaTeX y latex+dvips.
- Los comandos estándar de \LaTeX funcionan. La instrucción `\tableofcontents` todavía crea una tabla de contenidos, `\section` aún crea una estructura e `\itemize` crea una lista, etc.

- Se pueden crear fácilmente cubiertas y efectos dinámicos.
- Permite cambiar la apariencia de una presentación para satisfacer ciertos propósitos.
- Los temas se diseñan para ser útiles en la práctica, no para dar espectáculo.
- La distribución, los colores y las fuentes usadas en una presentación se pueden cambiar globalmente de una manera fácil, pero todavía se tiene el control sobre el detalle más mínimo.
- Existe un paquete especial en BEAMER que permite usar directamente el código fuente de una presentación en otras clases de L^AT_EX como `article` o `book`. Esto facilita la creación de presentaciones al margen de las notas de conferencia o notas de conferencia al margen de presentaciones.
- Típicamente el resultado final es un archivo en PDF. Existen aplicaciones para visualizar este formato en cualquier plataforma. Al llevar la presentación a una conferencia en algún dispositivo, uno no se tiene que preocupar sobre la versión del programa que servirá para mostrar la presentación, ésta se verá exactamente como en nuestra computadora.

Una presentación en BEAMER se crea como cualquier otro documento en L^AT_EX: tiene un preámbulo y un cuerpo, el cuerpo contiene secciones y subsecciones, las diferentes diapositivas (llamados marcos en BEAMER) se ponen en ambientes, se estructuran usando los ambientes `\itemize` y `\enumerate`, y así sucesivamente. La desventaja obvia es que es necesario saber L^AT_EX para usar BEAMER. De ser así, la ventaja es que se pueden crear presentaciones y no sólo escribir artículos.

La primera parte está organizada como sigue: la Sección 2 trata sobre la instalación del paquete. Si se es nuevo en BEAMER, es recomendable leer la guía didáctica en la Sección 3. Al estar ante la creación de la primera presentación real usando BEAMER, se debe leer la Sección 4 donde se discuten los detalles técnicos de un posible flujo de trabajo. En la Sección 5 se dan algunas directrices sobre qué hacer y qué no hacer. Finalmente, se debe hojear la Sección 6 donde se encontrarán unas plantillas-solución para crear charlas.

La segunda parte entra en los detalles de los comandos definidos en BEAMER, pero también se abordan otros asuntos técnicos que tienen que ver con la creación (cómo incluir gráficos o animaciones).

La tercera parte explica cómo se puede cambiar fácilmente la apariencia de una presentación, ya sea usando temas, especificando colores o fuentes para elementos específicos de una presentación (como la fuente usada para los números en una enumeración).

La última parte contiene “los cómo”, que son explicaciones de cómo conseguir hacer ciertas cosas usando BEAMER.

En este trabajo se incluyen las descripciones de todos los comandos “públicos”, ambientes y conceptos definidos por la clase BEAMER. Los siguientes ejemplos muestran cómo se documentan las cosas. Como regla general, el texto en rojo está *definido*, el texto en verde es *opcional* y el texto en azul indica en que modo se aplica el párrafo.

`\alguncomandobeamer` [*argumentos opcionales*] {*primer argumento*} {*segundo argumento*}

Aquí se encontrará la explicación de lo que hace el comando `\alguncomandobeamer`. El argumento en verde es opcional. En este caso el comando toma dos parámetros.

`\begin{algunambientebeamer}` [*argumentos opcionales*] {*primer argumento*}
<Contenido del ambiente>
`\end{algunambientebeamer}`

Aquí se encontrará la explicación del efecto del ambiente `algunambientebeamer`.

Plantilla-/Color-/Fuente-Beamer algún elemento beamer

Aquí se encontrará una explicación de la plantilla, color y/o fuente de algún elemento beamer. Un “elemento-BEAMER” es un concepto que se explica con más detalle en la Sección 15. Burdamente, un *elemento* es una parte de la presentación que potencialmente está escrita de alguna manera especial. Ejemplos de elementos son los encabezados de marcos, el nombre del autor o el símbolo de pie de página. Para la mayoría de los elementos existe una plantilla-BEAMER, un color-BEAMER y una fuente-BEAMER.

Para cada elemento se indica si existe una plantilla-BEAMER, un color-BEAMER y/o una fuente-BEAMER. Típicamente los tres aspectos existen y se emplean juntos cuando se edita un elemento, es decir, cuando se inserta una plantilla se instalan antes el color-BEAMER y la fuente-BEAMER correspondientes. Sin embargo, en ocasiones las plantillas no tienen un color o fuente asociada con ellas (como las plantillas padre). También existen colores-BEAMER y fuentes-BEAMER que no tienen una plantilla subyacente.

En la Sección 15.3 se explica como usar y cambiar las plantillas. Aquí está la esencia: para cambiar una plantilla, se puede escribir

```
\setbeamertemplate{algún elemento beamer}{nuestra definición para esta plantilla}
```

Desgraciadamente no es fácil proponer una buena definición para algunas plantillas. Afortunadamente a menudo hay *opciones predefinidas* para una plantilla. Por ejemplo, éstas se indican así:

- `[square]` causa el uso de un cuadrado pequeño para hacer la plantilla.
- `[circle]{<radio>}` causa el uso de un círculo con radio dado para hacer la plantilla.

Se puede instalar tal opción predeterminada así:

```
\setbeamertemplate{algún elemento beamer}[circle]{3pt}  
% Ahora se usan círculos
```

Los colores-BEAMER se explican en la Sección 16. Aquí está la esencia: para cambiar el color del primer plano a rojo usamos

```
\setbeamercolor{algún elemento beamer}{fg=red}
```

Para cambiar el fondo a negro, usamos:

```
\setbeamercolor{algún elemento beamer}{bg=black}
```

También se pueden cambiar usando `fg=red`, `bg=black` juntos. El fondo no siempre será “respetado” dado que es difícil mostrar correctamente un fondo coloreado y la plantilla debe hacer un esfuerzo extra (mientras que el color del primer plano normalmente se usa de manera automática).

Las fuentes-BEAMER se explican en la Sección 17. Aquí esta la esencia: para cambiar el tamaño de la fuente a grande usamos:

```
\setbeamerfont{algún elemento beamer}{size=\large}
```

Además del tamaño, se pueden hacer cosas como `series=\bfseries` para fijar las series en negritas, `shape=\itshape` para cambiar la forma, `family=\sffamily` para cambiar la familia y se pueden usar juntas. Hay que agregar un asterisco al comando para “reiniciar” primero la fuente.

PRESENTACIÓN Como ahora, a veces se encontrará la palabra PRESENTACIÓN en azul al lado de algún párrafo. Esto significa que el párrafo sólo aplica cuando “se va a editar una presentación usando L^AT_EX o pdfL^AT_EX”.

ARTÍCULO Por el contrario, un párrafo con la palabra ARTÍCULO al lado describe algún comportamiento que es especial para el modo article. Este modo especial se usa para crear notas de la conferencia al margen de una presentación (los dos pueden coexistir en el mismo archivo).

Empezando

Esta parte nos ayuda a empezar. Inicia con una explicación de cómo instalar la clase, además se da una explicación de aspectos especiales que hay que considerar al usar otros paquetes.

En seguida, se da una guía didáctica corta que explica la mayoría de las características que se necesitan en una presentación típica.

Después de la guía didáctica se encontrará un “posible flujo de trabajo” para crear una presentación. Seguirlo puede evitar problemas posteriores. Esta parte además incluye una sección de directrices. Seguir estas pautas puede ayudar a crear buenas presentaciones (sin garantía). Esta sección permanece lo más general como es posible; la mayoría de lo que se dice se aplica a las presentaciones en general, independientemente de si han sido creadas usando BEAMER o no.

Al final de esta parte se encontrará un resumen de las plantillas-solución que vienen con BEAMER. Se pueden usar las plantillas-solución para iniciar la creación de una presentación.

2. Instalación de programas y paquetes

Hay diferentes maneras de instalar la clase BEAMER dependiendo de nuestros recursos y necesidades. Antes de instalar la clase BEAMER, es necesario instalar algunos programas y paquetes.

2.1. Programas necesarios

Si se va a trabajar con el Sistema Operativo *Windows XP* se debe proceder como sigue:

1.- Se debe instalar primero el programa de distribución libre *Adobe Reader*, el cual se puede descargar de:

<http://www.adobe.com/es/>

2.- En seguida se debe instalar el programa de distribución libre *Ghostscript* de:

<http://easynews.dl.sourceforge.net/sourceforge/ghostscript/gs854w32-gpl.exe>
<http://www.ghostscript.com/awki>

3.- A continuación se debe instalar el programa de distribución libre *Ghostview* de:

<http://www.cs.wisc.edu/~ghost/gsview/get48.htm>

4.- Luego se debe instalar el programa de distribución libre *MikTeX* de:

<http://www.miktex.org/>
<http://superb-east.dl.sourceforge.net/sourceforge/miktex/setup-2.5.2449.exe>

En este caso es recomendable hacer la instalación completa, ya que ésta incluye todos los paquetes de \LaTeX posibles (incluyendo *beamer*, *pgf* y *xcolor*).

5.- Por último se instala el programa *WinEdt* de:

<http://www.winedt.org/>

2.2. Instalación de paquetes

MiKTeX

Si cuando se instaló MiKTeX no se dió la opción de instalación completa entonces los paquetes necesarios se tienen que instalar “a mano”, para ello se usa el mago de actualización (MiKTeX Update Wizard) de MiKTeX para instalar las versiones más recientes de los paquetes *beamer*, *pgf* y *xcolor* de \LaTeX .

2.3. Instalación en un árbol de texmf

La manera correcta de instalar BEAMER es colocarlo en el llamado árbol *texmf*. Para ello hay que obtener la versión más reciente del archivo fuente (con extensión *.tar.gz* o *.zip*) del paquete BEAMER de <http://sourceforge.net/projects/latex-beamer/>

Se necesitan los paquetes *PGF* y *xcolor* que se encuentran en el mismo lugar (aunque las versiones en CTAN podrían ser más nuevas).

En cualquier caso, los paquetes contienen grupos de archivos, que hay que colocarlos en un árbol *texmf* apropiado.

Cuando se le pide a \TeX que use una cierta clase o paquete, éste normalmente busca los archivos necesarios en los llamados árboles *texmf*. Estos árboles simplemente son grandes directorios que contienen estos archivos. Por defecto, \TeX busca archivos en tres árboles *texmf* diferentes:

- El árbol raíz *texmf*, que generalmente se localiza en `/usr/share/texmf/`, `c:\texmf\` o en `c:\Program Files\TeXLive\texmf\`.

- El árbol local `texmf`, que normalmente se localiza en `/usr/local/share/texmf/`, `c:\localtexmf\` o en `c:\Program Files\TeXLive\texmf-local\`.
- El árbol personal `texmf`, que se suele localizar en el directorio `home~/texmf/` o `~/Library/texmf/`.

Los paquetes se deben instalar en el árbol local o en el árbol personal, dependiendo si se tiene acceso de escritura al árbol local. La instalación en el árbol raíz puede causar problemas, debido a que una actualización de la instalación total de $\text{T}_{\text{E}}\text{X}$ reemplazará todo este árbol.

Dentro del directorio `texmf` ya elegido, se crean los sub-sub-sub-directorios (folders)

- `texmf/tex/latex/beamer`,
- `texmf/tex/latex/pgf`, y
- `texmf/tex/latex/xcolor`

y se ponen todos los archivos en los correspondientes directorios.

Finalmente, se debe reconstruir la base de datos del sistema de archivos de $\text{T}_{\text{E}}\text{X}$. Para hacer esto en $\text{M}_{\text{I}}\text{K}_{\text{T}}\text{E}_{\text{X}}$, se pulsa la opción *Refresh Now* del menú *MiKTeX Options* que está en el folder $\text{M}_{\text{I}}\text{K}_{\text{T}}\text{E}_{\text{X}}$. En Linux esto se hace ejecutando el comando `texhash` o `mktexlsr` (son lo mismo).

2.4. Actualizando la instalación

Para actualizar la instalación de una versión anterior, simplemente se reemplaza lo que está en los directorios como en `texmf/tex/latex/beamer` por los archivos de la nueva versión. La manera más fácil de hacer esto es eliminar primero la versión antigua y luego se procede como antes. Algunas veces hay cambios en la sintaxis de ciertos comandos de versión en versión. Si las cosas que antes funcionaban ya no lo hacen, se debe echar un vistazo a las notas de la nueva versión y a la bitácora de cambios.

2.5. Comprobando la instalación

Para probar la instalación se copia el archivo `beamerexample1.tex` que se localiza en el directorio `\MiKTeX2.5\doc\latex\beamer\examples` a algún lugar donde normalmente creamos presentaciones. Entonces se ejecuta varias veces el comando `pdflatex` sobre el archivo (en WinEdt se pulsa el ícono $\text{P}_{\text{D}}\text{F}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ que se encuentra en la parte superior en la barra de herramientas) y checamos si el archivo resultante `beamerexample1.pdf` se ve bien.

2.6. Compatibilidad con otros paquetes y clases

Al usar ciertos paquetes o clases junto con la clase `beamer` se pueden necesitar opciones extras o se debe tener precaución.

```
\usepackage{AlDraTex}
```

Los gráficos que se crean usando `AlDraTex` se deben tratar como texto literal (`verbatim`). La razón es que `DraTex` intenta arreglar catcodes y espacios de manera muy parecida a como lo hace el ambiente `verbatim`. Así que para insertar un dibujo se agrega la opción `fragile` al marco o se usa el comando `\defverbatim` para crear una caja que contendrá al dibujo.

```
\usepackage{alltt}
```

El texto en un ambiente `alltt` se debe tratar como texto literal (`verbatim`). Así que se agrega la opción `fragile` a los marcos conteniendo este ambiente o se usa `\defverbatim`.

```
\usepackage{amsthm}
```

Este paquete se carga automáticamente dado que `BEAMER` lo usa para editar teoremas. Si no desea que se cargue (lo cual puede ser necesario sobre todo en el modo `article`, si el paquete es incompatible

con la clase del documento) se puede usar la opción de clase `noamsthm`. Ver la Sección 11.4 para más detalles.

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

Al usar el estilo `french` ciertas características que desentonan con la funcionalidad de la clase `BEAMER` serán desactivadas. Por ejemplo, las enumeraciones todavía se producen según lo dicta el tema y no como lo hace el estilo `french`.

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

PRESEN-
TACIÓN Al usar el estilo `spanish` ciertas características que desentonan con la funcionalidad de la clase `BEAMER` serán desactivadas. En particular, el comportamiento especial de los paréntesis angulares `<` y `>` deja de funcionar.

ARTÍCULO Para activar los caracteres `<` y `>` en este modo, se debe pasar la opción `activeospeccharacters` al paquete `beamerbasearticle`. Esto conducirá a problemas con las especificaciones de cubierta.

```
\usepackage{color}
```

PRESEN-
TACIÓN El paquete `color` se carga automáticamente con la clase `beamer.cls`. Esto imposibilita el paso de opciones a `color` en el preámbulo del documento de la manera normal. Para pasar una *<lista de opciones>* a `color`, se puede usar la siguiente opción de la clase:

```
\documentclass[color=<lista de opciones>]{beamer}
```

Causa que se pase al paquete `color` la *<lista de opciones>*. Si la *<lista de opciones>* contiene más de una opción éstas se deben encerrar entre llaves.

ARTÍCULO El paquete `color` no se carga automáticamente si `beamerarticle` se carga con la opción `noxcolor`.

```
\usepackage{colortbl}
```

PRESEN-
TACIÓN Con las versiones más recientes de `xcolor.sty` es necesario pasar la opción `table` a `xcolor.sty` si se desea usar `colortbl`. Ver las notas de `xcolor` sobre cómo hacer esto.

```
\usepackage{CJK}
```

PRESEN-
TACIÓN Se debe usar la opción de clase `CJK` para utilizar fuentes asiáticas.

```
\usepackage{deluxetable}
```

PRESEN-
TACIÓN Los servicios de generación de títulos de `deluxetable` se desactivan. En cambio se usa la plantilla `caption`.

```
\usepackage{DraTex}
```

Ver `AlDraTex`.

```
\usepackage{enumerate}
```

ARTÍCULO Este paquete se carga automáticamente en los modos de `presentation`. Si se desea usar sus características en el modo `article` se tiene que cargar el paquete “a mano”.

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

Usar esta opción sólo con fuentes que tienen fuentes outline disponibles en la codificación `T1` como `Times` o las fuentes `lmodern`. En una instalación normal las fuentes estándar `Computer Modern` (las fuentes que diseñó originalmente Donald Knuth y que se usan por defecto) *no* están disponibles en la codificación `T1`. Usar esta opción con estas últimas resultará en una pobre representación de la presentación al usar un visor de aplicaciones pdf como `Acrobat Reader` o `xpdf`. Para usar las fuentes `Computer Modern` con la codificación `T1` hay que usar el paquete `lmodern`. Ver Sección 17.2.3.

```
\usepackage{fourier}
```

El paquete cambia a una codificación `T1` pero no redefine todas las fuentes, de tal manera que se usan por defecto las fuentes outline (fuentes no construidas en mapas de bits). Por ejemplo, no se reemplazan el texto `sans-serif` y el texto `typewriter`. Para usar fuentes outline para éstas, escribir `\usepackage{lmodern}`

antes de incluir el paquete `fourier`.

```
\usepackage{hyperref}
```

PRESEN-
TACIÓN El paquete `hyperref` se carga automáticamente con la clase `beamer.cls` y se establecen ciertas opciones. Para anular opciones o pasar opciones adicionales a `hyperref` se puede usar la siguiente opción de la clase:

```
\documentclass[hyperref=(lista de opciones)]{beamer}
```

Causa que se pase *(lista de opciones)* al paquete `hyperref`.

Ejemplo: `\documentclass[hyperref={bookmarks=false}]{beamer}`

Otra alternativa es usar el comando `\hypersetup`.

ARTÍCULO En este modo no se le incluye automáticamente, si se le quiere usar se debe incluir a mano.

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

PRESEN-
TACIÓN Al usar Unicode se puede usar una de las siguientes opciones de la clase:

```
\documentclass[ucs]{beamer}
```

Carga el paquete `ucs` y pasa las opciones de Unicode correctas a `hyperref`. También precarga las páginas cero y uno de código Unicode.

```
\documentclass[utf8]{beamer}
```

Lo mismo que la opción `ucs`, pero también pone la codificación de entrada a `utf8`. También se podría usar la opción `ucs` y poner en el preámbulo `\usepackage[utf8]{inputenc}`.

Si se usa un carácter Unicode fuera de las primeras dos páginas de código (que incluye al alfabeto latino y al alfabeto latino extendido) en un encabezado de sección o de subsección, se tiene que usar el comando `\PreloadUnicodePage {<página de código>}` para darle oportunidad a `ucs` de precargar estas páginas de código. Se sabrá que un carácter no ha sido precargado, si se obtiene un mensaje como “Por favor insertar en el preámbulo”. La página de código de un carácter esta dada por el número Unicode del carácter dividido por 256.

```
\usepackage{listings}
```

PRESEN-
TACIÓN Los ambientes `lstlisting` se deben tratar de la misma manera que los ambientes `verbatim`. Para usar `\defverbatim` conteniendo un `lstlisting` coloreado, se debe dar la opción `colored` de `\defverbatim`.

```
\usepackage{msc}
```

PRESEN-
TACIÓN Dado que este paquete internamente usa `pstricks`, todo lo que aplica para `pstricks` aplica a `msc`.

```
\usepackage{musixtex}
```

Al usar `MusiXTEX` para editar partituras musicales, el documento se debe compilar con `pdfelatex` o `elatex` en lugar de `pdflatex` o `latex`.

Dentro de un ambiente `music`, el comando `\pause` se redefine para coincidir con la definición de `MusiXTEX` (una pausa durante la cuarta parte de un todo). Se puede usar el comando `\beamerpause` para crear cubiertas en este ambiente.

```
\usepackage{pdfpages}
```

Algunos comandos como `\includepdf` sólo funcionan *fuera* de los marcos cuando producen páginas “por sí mismos”. Cuando se usa tal comando también se puede poner

```
\setbeamercolor{background canvas}{bg=}
```

ya que el fondo (incluso uno blanco) se imprimirá sobre la imagen que se intenta incluir.

Ejemplo:

```
\begin{document}
```

```
\begin{frame}
```

```
\titlepage
```

```

\end{frame}
{
\setbeamercolor{background canvas}{bg=}
\includepdf{algunasimagenespdf.pdf}
}
\begin{frame}
Un marco normal.
\end{frame}
\end{document}

```

```
\usepackage{<professional font page>}
```

PRESEN- Si se usa un paquete de fuentes profesional, la redefinición interna de BEAMER de cómo se editan las
TACIÓN variables puede interferir con el modo superior de edición del paquete de fuentes. En este caso se debe usar la opción de clase `professionalfont` para suprimir cualquier substitución de fuentes. Ver Sección 17.2.2 para detalles.

```
\usepackage{pstricks}
```

Se debe agregar la opción `xcolor=pst` para advertir a `xcolor` del hecho que se está usando `pstricks`.

```
\usepackage{textpos}
```

PRESEN- BEAMER instala automáticamente un fondo blanco detrás de todo, a menos que se instale una plantilla
TACIÓN de fondo diferente. Debido a esto, se debe usar la opción `overlay` al usar `textpos` de tal manera que se pondrán “cajas” *antes* de todo. Alternativamente, se puede instalar una plantilla de fondo vacía, pero en ciertas situaciones esto puede producir un despliegue incorrecto en versiones antiguas de Acrobat Reader.

```
\usepackage{ucs}
```

Ver `\usepackage[utf8]{inputenc}`.

```
\usepackage{xcolor}
```

PRESEN- Los paquetes `xcolor` y `color` se cargan automáticamente con la clase `beamer.cls`.
TACIÓN

```
\documentclass[xcolor=(lista de opciones)]{beamer}
```

Causa que se pase al paquete `xcolor` la *(lista de opciones)*.

Al usar BEAMER con el paquete `pstricks`, hay que pasar la opción `xcolor=pst` a BEAMER (y por lo tanto a `xcolor`).

ARTÍCULO Si `beamerarticle` se carga con la opción `noxcolor` el paquete `color` no se carga automáticamente.

2.7. Versiones y dependencias

Esta documentación es parte de la versión 3.06 de la clase BEAMER. Para que funcione BEAMER se necesita una versión reciente de varios paquetes estándar, así como los siguientes paquetes especiales (versiones posteriores deben funcionar, pero no necesariamente):

- `pgf.sty` version 1.00,
- `xcolor.sty` version 2.00.

Se necesita además la versión 0.14 de `pdflatex` (o más recientes), versiones anteriores no funcionan.

3. Guía para crear una presentación sencilla

Esta sección presenta una guía corta que se enfoca en algunas características básicas de BEAMER. Los detalles se explicarán posteriormente.

3.1. Plantilla-solución

Supongamos que se quiere usar BEAMER para crear la presentación de una conferencia de 20-25 minutos, incluidas las preguntas. Lo primero que se hace es buscar una plantilla-solución para esa presentación. Vemos que el archivo

```
beamer/solutions/conference-talks/conference-ornate-20min.en.tex
```

parece apropiado. Es recomendable crear un subdirectorio `presentación` en el directorio que contiene el artículo a exponer, luego se copia la plantilla-solución a este subdirectorio, renombrándolo como `principal.tex`.

En BEAMER una presentación consiste de una serie de marcos. Cada marco a su vez puede consistir de varias diapositivas (si hay más de una, se les llama cubiertas). Normalmente, todo lo que está entre `\begin{frame}` y `\end{frame}` se pone en una sola diapositiva y no se realiza ningún cambio de página.

A continuación reproducimos el archivo antes mencionado con comentarios y ejemplos adicionales para explicar el funcionamiento de algunos comandos. Es recomendable que después de cada cambio, se compile dos veces el archivo usando `pdflatex` (si se usa WinEdt) o `latex` seguido de `dvips` y `ps2pdf` (si se usa Linux), para enseguida revisar los cambios efectuados usando Acrobat Reader. Ver la Sección 4.3.3 para agilizar la recompilación.

```
\documentclass{beamer}
\mode<presentation>
{
\usetheme{Warsaw}
% o se pueden usar los temas AnnArbor, JuanLesPins, PaloAlto, Rochester, etc. Experimentar.
\setbeamercovered{transparent}
% o cualquier otra cosa (posiblemente sólo hay que borrarla)
}

\usepackage[english]{babel}
% o spanish por ejemplo
\usepackage[latin1]{inputenc}
% o cualquier otra cosa
\usepackage{times}
\usepackage[T1]{fontenc}
% o cualquier otra cosa. Note que la codificación y la fuente deben coincidir. Si T1
% no se ve bien, intentar borrar la línea con fontenc.

% La primera cosa lógica que aparece es \title. Naturalmente, se reemplaza por el título de la plática
% (o del artículo). Este comando toma un argumento opcional “corto” encerrado entre corchetes, que
% se muestra en lugares donde hay poco espacio.
\title[T\’{i}tulo corto de la pl\’{a}tica] % (opcional, usarlo sólo con títulos largos)
{T\’{i}tulo de la Pl\’{a}tica}
\subtitle {Incluirlo si se tiene un subt\’{i}tulo}
% De la misma manera se precisan los campos \author y \date
\author[Autor, Otro] % (opcional, usar sólo si hay varios autores)
{F.~Autor\inst{1} \and S.~Otro\inst{2}}
% - Poner los nombres en el mismo orden con el que aparecen en el artículo.
```

```

% - Usar el comando \inst{?} sólo si los autores tienen diferente afiliación.
% - Si hay un solo autor hay que quitar el comando \and
\institute[Universidad de Algunaparte y Otrositio] % (opcional, pero necesario)
{
\inst{1}\\
\texttt{euclides@alexandria.edu}\\
Departamento de Ciencia Computacional\\
Universidad de Algunaparte
\and
\inst{2}\\
\texttt{pitagoras@matem.edu}\\
Departamento de Filosof\`{i}a Te\`{o}rica\\
Universidad de Otrositio}
% - Usar el comando \inst sólo si hay varias afiliaciones.
\date[CPF 2003] % (opcional, debe ser abreviación del nombre de la conferencia)
{Conferencia sobre Presentaciones Fabulosas, 2003}
% - Usar el nombre de la conferencia o su abreviación.
\subject{Ciencia Computacional Te\`{o}rica}
% Esto sólo se inserta en el catálogo de información del archivo PDF. Se puede quitar.

% Si se tiene un archivo “nombre-logo-universidad.xxx”, donde xxx es un formato gráfico que pueda
% ser procesado por latex o pdflatex, resp., entonces se puede añadir un logo como sigue:
% \pgfdeclareimage[height=0.5cm]{logo-universidad}{nombre-logo-universidad}
% \logo{\pgfuseimage{logo-universidad}}

% El siguiente comando inserta un marco antes del inicio de cada subsección, el cual contiene la tabla
% de contenidos y además resalta el título de la subsección. Si no es agradable, simplemente eliminarlo
% y desaparece el marco (recomendable si se va a hacer una presentación sencilla):
\AtBeginSubsection
{
\begin{frame}(beamer)
\frametitle{Esbozo}
\tableofcontents[currentsection,currentsubsection]
\end{frame}
}

% Si se desea dejar al descubierto todas las cosas paso-a-paso, descomentar el siguiente comando:
%\beamerdefaultoverlayspecification{<+>}
\begin{document}
\begin{frame}
\titlepage
\end{frame}
% Genera el primer marco, el cual contiene los datos previos (título de la conferencia, autor, etc.)
% Lo siguiente genera una tabla de contenidos, que contiene el título de las secciones y subsecciones
% (en caso de que los haya).
\begin{frame}
\frametitle{Esbozo}
\tableofcontents%[pausesections]
% se podría usar la opción pausesections, que descubre (el título de) una sección a la vez
\end{frame}

\section{Motivaci\`{o}n}
\subsection{El Problema B\`{a}sico que Estudiamos}
% Estos comandos van fuera de los marcos, crean entradas en la tabla de contenidos así como un

```

% marco al inicio de cada subsección (ver el comando `\AtBeginSubsection` mencionado antes).
 % Además toman argumentos cortos opcionales (como en el comando `\title`) que se utilizan sólo si
 % hay necesidad. Los argumentos opcionales en BEAMER funcionan así, al contrario de la manera usual
 % en que L^AT_EX trata un argumento opcional para las secciones (el argumento opcional dicta lo que se
 % muestra en la tabla de contenidos y el argumento principal dicta lo que se muestra en las otras
 % partes).

```
\begin{frame}
\frametitle{Hacer T\`{i}tulos Informativos. Usar Letras May\`{u}sculas.}
% Crea el título del marco
\framesubtitle{Los Subt\`{i}tulos son Opcionales.} % Crea el subtítulo del marco
```

% CREANDO CUBIERTAS SIMPLES

% El siguiente ejemplo muestra dos líneas, una después de la otra (es decir, después de pulsar alguna
 % tecla durante el despliegue del archivo en PDF). Esto se logra usando el ambiente *itemize* y el
 % comando `\pause`

```
\begin{itemize}
\item Usar \texttt{itemize} mucho.
\pause
\item Usar oraciones muy cortas o frases cortas.
\end{itemize}
\end{frame}
```

 % `\pause` es útil para enfocar la atención del público en el objeto sobre el que se está hablando
 % actualmente. No abusar en su uso, el mecanismo de destapar es apreciable sólo en circunstancias
 % especiales.

% USANDO ESPECIFICACIONES DE CUBIERTAS

```
\begin{frame}
\frametitle{Hacer los T\`{i}tulos Informativos.} % Crea el título del marco
% No hay subtítulo
Se pueden crear cubiertas \dots
\begin{itemize}
\item usando el comando \texttt{pause} :
\begin{itemize}
\item Primer objeto. % Toda la información previa es visible en el marco, primer cubierta.
\pause % causa que el texto de abajo aparezca atenuado en color gris muy claro
\item Segundo objeto. % Esto se ve al presionar por primera vez alguna tecla, segunda cubierta.
\end{itemize}
\item usando especificaciones de cubiertas: % Se ve al presionar por primera vez alguna tecla
\begin{itemize}
\item<3-> Primer objeto. % Esto se ve al presionar por segunda vez alguna tecla,
% se interpreta como: mostrar a partir de la tercera cubierta
\item<4-> Segundo objeto. % Esto se ve al presionar por tercera vez alguna tecla
\end{itemize}
\item
usando el comando general \texttt{uncover} % Esto se ve al presionar por primera vez alguna tecla
\begin{itemize}
\uncover<5->{\item Primer objeto.} % Esto se ve al presionar por cuarta vez alguna tecla,
% se interpreta como: descubrir a partir de la quinta cubierta
\uncover<6->{\item Segundo objeto.} % Esto se ve al presionar por quinta vez alguna tecla
\end{itemize}
\end{itemize}
\end{frame}
```

% Las especificaciones de una cubierta se encierran entre `< y >`, en general, pueden ser listas de núme-

```

% ros o rangos de números donde el inicio o el final de un rango se pueden dejar abiertos. Por ejemplo
% -3,5-6,8- significa “mostrar su argumento en todas las diapositivas excepto en la 4 y 7”, en las otras
% diapositivas (4 y 7), el argumento está oculto (aunque ocupa espacio). BEAMER calcula
% automáticamente el número de diapositivas necesarias para cada marco.
\subsection{Trabajo Previo}
\begin{frame}[t]
% La opción [t] causa el desplazamiento hacia arriba del texto contenido en este marco.
\frametitle{¿Qué son los números primos?}
\begin{definition}
Un \alert{número primo} es aquel que tiene exactamente 2 divisores.
% el comando \alert edita su argumento en color rojo brillante para resaltarlo del texto restante
\end{definition} % se crea un rectángulo con colores vistosos
\pause % El texto siguiente aparece atenuado, se despliega al presionar alguna tecla
% Lo que muestra que este comando trasciende ambientes
\begin{example}
\begin{itemize}
\item 2 es primo (dos divisores: 1 y 2).
\item 4 no es primo (\alert{tres} divisores: 1, 2 y 4).
\end{itemize}
\end{example} % se crea un rectángulo de color verde que contiene el ambiente itemize.
\end{frame}
% Los ambientes theorem, lemma, proof, corollary, también están predefinidos en BEAMER. Como en
% amsmath, ellos toman argumentos opcionales encerrados entre corchetes. BEAMER carga
% automáticamente amsmath.
\begin{frame}
\frametitle{Hacer los T\`{i}tulos Informativos.}
\end{frame}
\section{Nuestros Resultados/Contribuci\`{o}n}
\subsection{Resultados Principales}

% Para crear un contraste visual más fuerte, se puede usar el ambiente columns dentro de un marco,
% el comando \column crea una nueva columna.
\begin{frame}
\frametitle{¿Qu\`{e} hay por hacer?}
\begin{columns}[t] % la opción [t] hace que las columnas se alineen por arriba
\column{.5\textwidth}
\begin{block}{Problemas resueltos} % El ambiente block funciona como el ambiente definition,
¿Cuántos primos hay? % pero permite dar un título arbitrario.
\end{block}
\pause % la otra columna sólo se “destapa” en la segunda diapositiva del marco.
\column{.5\textwidth}
\begin{block}{Preguntas abiertas}
¿Todo número par es la suma de 2 n\`{u}meros primos?
\cite{Goldbach1742} % Hace una cita a un trabajo listado en la bibliografía
\end{block}
\end{columns}
\end{frame}
\begin{frame}
\frametitle{Hacer los T\`{i}tulos Informativos.}
\end{frame}
\begin{frame}
\frametitle{Hacer los T\`{i}tulos Informativos.}
\end{frame}

```

```

\subsection{Ideas Básicas del Programa}

% USANDO TEXTO LITERAL. Para poder insertar símbolos reservados en LATEX
% usualmente se usa el ambiente verbatim y a veces el ambiente lstlisting para editar listados,
% ambos están disponibles en BEAMER pero hay que añadir la opción fragile en el marco:
\begin{frame}[fragile]
\frametitle{Un Algoritmo para Encontrar Números Primos.}
\begin{verbatim}
int main (void)
{
std::vector<bool> esprimo (100, true);
for (int i = 2; i < 100; i++)
if (esprimo[i])
{
std::cout << i << " ";
for (int j = i; j < 100; esprimo[j] = false, j+=i);
}
}
return 0;
}
\end{verbatim}
\begin{uncoverenv}<2>
Notar el uso de \verb|std:|.
\end{uncoverenv}
\end{frame}

% Si se desea destapar parte del algoritmo paso a paso y hacer énfasis en (partes de) líneas, se puede
% usar el paquete alltt, en casos simples es más útil el ambiente semiverbatim:
\begin{frame}[fragile]
\frametitle{Un algoritmo para encontrar números primos.}

% El siguiente ambiente funciona como verbatim sólo que \, {, y }, retienen
% su significado. Si se necesitan se editan usando \\, \{, \}.
\begin{semiverbatim}
\uncover<1->{\alert<0>{int main (void)}}
\uncover<1->{\alert<0>{\{\}}}
\uncover<1->{\alert<1>{\alert<4>{std::vector<bool> esprimo(100, true);}}}
\uncover<1->{\alert<1>{ for (int i = 2; i < 100; i++)}}
\uncover<2->{\alert<2>{ if (esprimo[i])}}
\uncover<2->{\alert<0>{\{\}}}
\uncover<3->{\alert<3>{\alert<4>{std::cout << i << " ";}}}
\uncover<3->{\alert<3>{ for (int j = i; j < 100;)}}
\uncover<3->{\alert<3>{ esprimo[j] = false, j+=i);}}}
\uncover<2->{\alert<0>{\{\}}}
\uncover<1->{\alert<0>{ return 0;}}}
\uncover<1->{\alert<0>{\{\}}}
\end{semiverbatim}
\visible<4->{Notar el uso of \alert{\texttt{std:}}.}
% este comando hace casi lo mismo que \uncover.
\end{frame}

\section*{Resumen}
\begin{frame}
\frametitle<presentation>{Resumen}
% Mantener el resumen *muy corto*.
\begin{itemize}

```

```

\item El \alert{primer mensaje principal} de tu pl\`{a}tica en una o dos l\`{i}neas
\item El \alert{segundo mensaje principal} de tu pl\`{a}tica en una o dos l\`{i}neas
\item Quizás un \alert{tercer mensaje}, pero no m\`{a}s.
\end{itemize}
% La siguiente vista es opcional.
\vskip0pt plus.5fill
\begin{itemize}
\item Vista
\begin{itemize}
\item Algo que no se ha resuelto.
\item Algo m\`{a}s que no se ha resuelto.
\end{itemize}
\end{itemize}
\end{frame}

% Lo que sigue es opcional y generalmente no se necesita.
\appendix
\section<presentation>*\{appendixname}
\subsection<presentation>*\{Para lectura posterior}
% Agregando Referencias
\begin{frame}[allowframebreaks]
\frametitle<presentation>\{Para lectura posterior}
\begin{thebibliography}{10}
\beamertemplatebookbibitems
% Empezar con libros.
\bibitem{Goldbach1742}[Goldbach, 1742] % Genera una entrada en la bibliografía
Christian Goldbach.% En el archivo PDF se ve además la imagen de un libro
\newblock {\em Carta a Euler}.
\newblock Alguna editorial, 1990.
\beamertemplatearticlebibitems
% A continuación artículos interesantes. Mantener una lista corta.
\bibitem{Someone2000} % Genera una entrada en la bibliografía
A.~Alguen.% En el archivo PDF se ve además la imagen de una hoja
\newblock Sobre esto y aquello.
\newblock {\em Journal de Esto y Aquello}, 2(1):50--100,
2000.
\end{thebibliography}
\end{frame}
\end{document}

```

Generado el archivo PDF, se puede ver que al pulsar sobre una sección o subsección en la barra de navegación situada en la parte superior del marco, saltamos a tal lugar. Se nota que los símbolos pequeños en la parte inferior derecha parecen estar disponibles. Al pulsar un rato sobre ellos, se encuentra que al hacerlo nuevamente sobre las flechas izquierda o derecha de un símbolo se produce un salto hacia delante o hacia atrás de una diapositiva/marco/subsección/sección. También se puede saltar al inicio o final de un marco/subsección/sección.

El comando `\item` no es el único que acepta especificaciones de cubierta. Otro comando que las acepta es `\uncover`, sólo que muestra su argumento en las diapositivas especificadas en la especificación de cubierta y en las demás diapositivas se oculta su argumento (aunque todavía ocupa espacio). El comando `\only` funciona de manera similar: en las diapositivas no especificadas el comando `\only` simplemente “desecha su argumento” y no ocupa espacio alguno. El usar este comando nos lleva a tener diferentes alturas de texto en las distintas diapositivas. Si se centra el texto verticalmente, esto originará que el texto se “tambalee”, es preferible usar el comando `\uncover`. Sin embargo, a veces se quiere que “realmente

desaparezcan” cosas en algunas diapositivas entonces `\only` es útil.

También se puede usar la opción de clase `t` (`top`) que causa que el texto en todos los marcos sea “empujado” verticalmente a la parte superior. Entonces la diferencia en las alturas del texto no causa ningún temblor. También se puede hacer esto para un sólo marco si se da el argumento opcional `[t]` al ambiente `frame`. Si se da la opción de clase `t`, se puede centrar verticalmente un sólo marco usando la opción `[c]` para tal marco.

Resulta que ciertos ambientes también aceptan especificaciones de cubierta. Si se da tal especificación, el ambiente entero sólo se muestra en las diapositivas indicadas.

3.2. Estructurando un marco

Se puede usar el ambiente `block` para crear un nuevo ambiente con un título arbitrario. Además se pueden definir ambientes tipo-teorema poniendo lo siguiente en el preámbulo:

```
\newtheorem{problemasresueltos}[theorem]{Problemas Resueltos}
\newtheorem{problemasabiertos}[theorem]{Problemas Abiertos}
```

El argumento opcional `[theorem]` asegura que estos ambientes se numeren de la misma forma que todos los demás. Sin embargo, dado que de cualquier manera estos números no se muestran no importa realmente si se da, pero es una buena práctica y quizás se podría necesitar después.

3.3. Cambiando la manera en que se ven las cosas: los temas

Se puede cambiar el tema Warsaw por el tema Frankfurt, pero el contraste luz-sombra es demasiado fuerte. Para suavizar el resultado se puede agregar:

```
\usecolortheme{seahorse}
\usecolortheme{rose}
```

Si se cree que la fuente que se usó para los títulos no es lo suficientemente clásica se puede agregar:

```
\usefonttheme[onlylarge]{structuresmallcaps serif}
```

Si se nota que las fuentes pequeñas en las barras de la navegación son difíciles de leer por ser tan delgadas, se puede agregar:

```
\usefonttheme[onlysmall]{structurebold}
```

3.4. Cambiando la manera en que se ven las cosas: colores y fuentes

Si se decide que la fuente para el título (únicamente) tiene que ser itálica cursiva se agrega:

```
\setbeamerfont{title}{shape=\itshape,family=\rmfamily}
```

Se puede notar que la fuente todavía es bastante grande, la razón de esto es que las llamadas a `\setbeamerfont` se acumulan y el tamaño ya se estableció como `\large` por algún tema-fuente. Al usar la versión con asterisco de `\setbeamerfont` se “reestablece” la fuente.

Si se decide cambiar el color del título a rojo, añadiendo quizás un poquito de negro, se usa el comando:

```
\setbeamerfont{title}{fg=red!80!black}
```

Al probar el siguiente comando, se encuentra que especificar un color de fondo tiene cierto efecto:

```
\setbeamerfont{title}{fg=red!80!black,bg=red!20!white}
```

4. Flujo de trabajo para crear una presentación en Beamer

En esta sección se presenta un flujo de trabajo para crear una presentación en BEAMER. Se abordan preguntas técnicas tales como qué programas llamar con qué parámetros.

4.1. Paso Uno: Organización de archivos

PRESEN-
TACIÓN Es recomendable crear una carpeta para cada presentación ya que \TeX crea tantos archivos extra que pueden crear confusión. El nombre ideal de la carpeta debería empezar con la fecha de su conferencia en formato ISO (por ejemplo 2003-12-25 para una conferencia en Navidad), seguido por algún texto que vaya de acuerdo a lo que se va a charlar, este formato permite organizar las carpetas en un directorio. Si se usa un directorio extra para cada presentación, se puede nombrar al archivo principal `principal.tex`.

Para crear un primer archivo `principal.tex` para la conferencia, se copia un archivo existente en el directorio `beamer/solutions` y se adapta a las necesidades. Más adelante se encuentra una lista de posibles soluciones BEAMER que contienen plantillas en \TeX para la presentación de archivos.

Si se desea que la conferencia resida en un mismo archivo como algo distinto, por ejemplo la versión artículo del texto, es aconsejable organizar un esquema del archivo más detallado. Ver Sección 20.2.2 para más detalles.

4.2. Paso Dos: Estructurando la presentación

El siguiente paso es completar el archivo de la presentación con `\section` y `\subsection` para crear un bosquejo preliminar. Se encontrarán algunas sugerencias de cómo crear un buen bosquejo en la Sección 5.1.

Los comandos `\section` y `\subsection` se ponen en el archivo `principal.tex` (más o menos vacío). No crear ningún marco hasta tener una primera versión del funcionamiento de una posible tabla de contenidos. El archivo podría verse así:

```
\documentclass{beamer}
% Este es el archivo principal.tex

\usetheme{Berlin}

\title{Ejemplo de una Presentaci\'}{o}n Creada con el Paquete Beamer}
\author{Till Tantau}
\date{\today}

\begin{document}

\begin{frame}
\titlepage
\end{frame}

\section*{Esbozo}
\begin{frame}
\tableofcontents
\end{frame}

\section{Introducci\'}{o}n}
\subsection{Perspectiva General de la clase Beamer}
\subsection{Perspectiva General de Clases Similares}

\section{Uso}
\subsection{...}
```

```

\subsection{...}

\section{Ejemplos}
\subsection{...}
\subsection{...}

\begin{frame}
\end{frame}% para imponer entradas en la tabla de contenidos

\end{document}

```

El marco vacío al final (que se debe borrar después) asegura que realmente las secciones y subsecciones sean parte de la tabla de contenidos. Este marco es necesario después de un comando `\section` o `\subsection`, si estos comandos se encuentran en la última página de un documento entonces no tienen efecto alguno.

4.3. Paso Tres: Creando un archivo PDF o un archivo PostScript

PRESENCIA Una vez que se termina una primera versión de la estructura, se debe crear un primer archivo PDF o archivo PostScript de la conferencia (aún vacía) para asegurar que todo está trabajando apropiadamente. Este archivo contendrá sólo la página título y la tabla de contenidos.

Creando un archivo PDF

PRESENCIA Para crear una versión PDF de este archivo, se ejecuta por lo menos dos veces el programa `pdflatex` sobre `principal.tex` (en WinEdt sólo se pulsa sobre el ícono PDF \LaTeX). Se necesita correrlo dos veces para que \TeX pueda crear la tabla de contenidos (incluso puede ser necesario ejecutarlo frecuentemente dado que se crean toda clase de archivos auxiliares). En el siguiente ejemplo, el signo mayor que (`>`) es el prompt (MSDOS o Linux)

```

> pdflatex principal.tex
... muchas salidas...
> pdflatex principal.tex
... muchas salidas...

```

Luego se puede usar un programa como Acrobat Reader (Windows) o `xpdf` (Linux) para ver la presentación resultante.

```

> acroread principal.pdf (sólo Linux)

```

En WinEdt sólo se pulsa el ícono que representa el program Acrobat Reader.

Creando un archivo PostScript

PRESENCIA Para crear una versión PostScript, primero se debe averiguar que el paquete `HYPERRREF` (el cual se carga automáticamente con la clase `BEAMER`) use la opción `dvips` o alguna opción compatible, ver la documentación del paquete `HYPERRREF` para detalles. Si éste es el caso se depende del contenido del archivo local `hyperref.cfg`. Se puede imponer el uso de esta opción pasando `dvips` o una opción compatible a la clase `BEAMER` (escribir `\documentclass[dvips]{beamer}`), que a su vez la pasará al paquete `HYPERRREF`.

Entonces se puede ejecutar dos veces `latex`, seguido de `dvips` (*Linux*).

```

> latex principal.tex
... muchas salidas...
> latex principal.tex
... muchas salidas...

```

```
> dvips -P pdf principal.dvi
```

La opción (`-P pdf`) le dice a `dvips` que use fuentes outline Tipo 1 en lugar de las fuentes bitmap Tipo 3 usuales. Se puede omitir esta opción si hay algún problema.

Se puede convertir un archivo PostScript a un archivo pdf usando (MSDOS, Linux)

```
> ps2pdf principal.ps principal.pdf
```

En WinEdt se pulsa dos veces el ícono \LaTeX y luego el ícono `dvi2ps`.

Formas de mejorar la velocidad de compilación

Mientras se trabaja en la presentación, a veces puede ser útil para \TeX que los archivos `.tex` sean “rápidos” y que la presentación contenga sólo la información más importante. Esto es útil sobre todo si se tiene una máquina lenta. En este caso, se pueden hacer varias cosas para acelerar la compilación. Primero, se puede usar la opción de clase `draft`.

`\documentclass[draft]{beamer}` Causa que los encabezados de página, los pies de página y las barras laterales se reemplacen por rectángulos grises (aunque todavía se calculan sus medidas). Otros paquetes, incluyendo `pgf` y `hyperref`, también se “aceleran” cuando se usa esta opción.

Segundo, se puede usar el siguiente comando:

```
\includeonlyframes{<lista de etiquetas de marcos>}
```

Este comando se comporta casi como el comando `\includeonly`: sólo se incluyen los marcos mencionados en la lista. Los otros marcos se suprimen. No obstante, todavía se ejecutan los comandos `\section` y `\subsection`, para que se tengan las barras de navegación correctas. Al etiquetar el marco actual como, digamos, `actual` y luego decir `\includeonlyframes{actual}`, se puede trabajar rápidamente en un solo marco.

La `<lista de etiquetas de marcos>` es una lista separada por comas (sin espacios) de los nombres de marcos que han sido etiquetados. Para etiquetar un marco, se debe pasar la opción `label=<nombre>` al comando `\frame` o al ambiente `frame`.

Ejemplo:

```
\includeonlyframes{ejemplo1,ejemplo3}
\frame[label=ejemplo1]
{Se incluir\'}{a} este marco.}
\frame[label=ejemplo2]
{No se incluir\'}{a} este marco.}
\frame{No se incluir\'}{a} este marco.}
\againframe{ejemplo1}% Será incluido
```

4.4. Paso Cuatro: Creando marcos

Una vez que la tabla de contenidos es satisfactoria, se empiezan a crear marcos para la presentación agregando ambientes `frame`.

4.5. Paso Cinco: Probando la presentación

Siempre pruebe la presentación. Para esto, se debe vocalizar o subvocalizar la conferencia en un ambiente tranquilo. Generalmente esto mostrará si la conferencia es demasiado larga. Se deben quitar partes de la presentación de tal manera que quepa en el tiempo asignado. *No* se debe intentar hablar rápidamente para terminar la conferencia en el tiempo asignado ya que se perderá la atención del público.

No intentar crear la presentación “perfecta” inmediatamente. Más bien, se prueba y reevalúa la conferencia y se modifica como sea necesario.

4.6. Paso Seis: Creando un folleto

Creando el folleto

Una vez que la conferencia está preparada, si parece apropiado se puede crear un folleto. Para esto, se puede usar la opción de clase `handout`. Generalmente se pueden poner varias diapositivas del folleto en una página.

También se puede crear una versión artículo de la conferencia. Una “versión artículo” de la presentación es un texto normal editado en \TeX usando, por ejemplo, la clase de documento `article` o quizás `llncs` o una clase de documento similar. La clase `BEAMER` ofrece medios para hacer que esta versión coexista en el mismo archivo con la versión de la presentación, compartiendo código. También se pueden incluir diapositivas de la presentación como figuras en la versión artículo.

Impresión del folleto

La manera más fácil de imprimir una presentación es usar Acrobat Reader con la opción “expand small pages to paper size” en el cuadro de diálogo que despliega la impresora. Esto es necesario, porque las diapositivas son sólo de 128 mm. por 96 mm.

Para la versión en PostScript y para imprimir diapositivas múltiples en una sola página este método simple no funciona. En tales casos se puede usar el paquete `pgfpages`, que funciona directamente tanto con `pdflatex` y con `latex` más `dvips`. Notar sin embargo *que este paquete destruye hiperligas*. Esto es debido a fallas fundamentales en la especificación PDF y probablemente no cambien.

El paquete `pgfpages` puede hacer toda clase de trucos con páginas. El más importante para imprimir diapositivas en `BEAMER` es el siguiente comando:

```
\usepackage{pgfpages}
\pgfpagelayout{resize}[a4paper,border shrink=5mm,landscape]
```

Esto significa “Reescalar todas las páginas para papel A4 en modo landscape, no importando su tamaño original, encogiéndolo las páginas en 5mm, para que haya borde alrededor de todo”. Naturalmente, en lugar de papel A4 se puede usar `letterpaper` (tamaño carta) o cualquiera de los otros tamaños estándar de papel. Para más opciones y detalles ver la documentación de `pgfpages`.

La segunda cosa que se puede hacer es poner varias diapositivas en una sola página como sigue:

```
\usepackage{pgfpages}
\pgfpagelayout{2 on 1}[a4paper,border shrink=5mm]
```

Esto significa “Poner dos páginas en una sola y entonces reescalar todo para que encaje en papel A4”. Notar que esta vez no se necesitó el modo `landscape` dado que la página resultante, después de todo, no está en modo `landscape`. En lugar de 2 en 1 también se puede usar 4 en 1, pero entonces una vez más con `landscape` y también 8 en 1 e incluso 16 en 1 para conseguir una apreciación global (aunque ilegible).

Si se ponen varias diapositivas en una página y si éstas normalmente tienen fondo blanco, es útil escribir lo siguiente en el preámbulo:

```
\mode{handout}{\setbeamercolor{background canvas}{bg=black!5}}
```

Esto origina que las diapositivas de la versión folleto tengan un fondo gris muy luminoso. Esto ayuda a distinguir el borde de las diapositivas si se ponen varias en una página.

5. Directrices para crear presentaciones

En esta sección se esbozan las directrices que se deberían seguir para crear presentaciones. Estas directrices resultan de la experiencia, del sentido común o de las recomendaciones de otras personas o libros. Estas reglas no se proponen como mandamientos que si no se siguen, entonces todo resultará un desastre.

La regla central de la tipografía también se aplica para crear presentaciones: *Cada regla se puede romper, pero ninguna se puede ignorar.*

5.1. Estructurando una presentación

Conocer las restricciones de tiempo

Cuando se crea una presentación, lo primero que nos debe preocupar es la cantidad de tiempo disponible para la presentación, dependiendo de la ocasión puede ser de dos minutos a dos horas.

- Una regla simple acerca del número de marcos es que se debe tener a lo más un marco por minuto.
- En la mayoría de las veces se dispondrá de menos tiempo del que nos gustaría para la presentación.
- *En una presentación no exprimir más de lo que nos permite el tiempo.* No importa cuan importantes parezcan algunos detalles es preferible omitirlos, comunicar el mensaje principal es mejor que no comunicar ni el mensaje principal ni los detalles. En muchas situaciones, una estimación rápida del tiempo necesario para nuestra presentación mostrará que no se pueden mencionar ciertos detalles. Sabiendo esto, se pueden ahorrar horas de trabajo en preparar diapositivas que de cualquier otra forma se tendrían que quitar después.

Estructura global

Para crear la “estructura global” de una presentación, con las restricciones de tiempo en mente, se procede como sigue:

- Hacer un inventario mental de las cosas acerca de las que se puede hablar razonablemente dentro del tiempo disponible.
- Clasificar el inventario en secciones y subsecciones.
- Para pláticas muy largas (como conferencias de 90 minutos), también se puede dividir la conferencia en partes independientes (como un “Repaso de la Parte Previa” y una “Parte Principal”) usando el comando `\part`. Notar que cada parte tiene su propia tabla de contenidos.
- No se debe temer cambiar la estructura después, mientras se trabaja en la conferencia.

Partes, Secciones y Subsecciones

- Usar entre dos y cuatro secciones por parte.

Incluso cuatro secciones es demasiado a menos que sigan un diseño fácil. Cinco o más son difíciles de recordar. Después de todo, cuando se presente la tabla de contenidos, el público realmente no podrá todavía comprender la importancia y relevancia de las diferentes secciones y se habrá olvidado de ellas cuando las hayamos alcanzado.

- Idealmente, una tabla de contenidos se debe explicar por sí misma. En particular, debe ser comprensible incluso *antes* de que alguien haya oído la conferencia.
- Que los títulos de sección y subsección se expliquen por sí mismos.
- Las secciones y subsecciones deben seguir un patrón lógico.

- Empezar con una explicación acerca de lo que trata la conferencia. (No asumir que todos saben acerca de la materia. La *Ley del Público Ignorante* dice: Alguien importante en el público siempre sabe menos de lo que uno piensa que todo el mundo debe saber, aun cuando uno tenga en cuenta la Ley del Público Ignorante.)
- Luego explicar lo que uno o alguien más ha averiguado acerca de la materia.
- Siempre concluir la conferencia de una manera simple y corta, con un resumen que reitere el mensaje principal. Las personas prestan más atención al principio y al final de las conferencias. El resumen es una “segunda oportunidad” para comunicar un mensaje.
- Se puede agregar un apéndice usando el comando `\appendix`. En esta parte se debe poner todo sobre lo que no se piensa hablar realmente, pero que podría ser conveniente para cuando se hagan las preguntas.
- No usar subsubsecciones, son malignas.

Dar un extracto (abstract). En los artículos científicos, el abstract da un pequeño resumen del artículo entero en aproximadamente 100 palabras. Este resumen tiene la intención de ayudar a los lectores a estimar si deben leer el artículo completo o no.

- Dado que es improbable que, en una presentación, el público huya después de la primera diapositiva no es necesario presentar un abstract.
- Sin embargo, si se puede dar un enunciado preciso acerca de la charla, se podría incluir un abstract.
- Si se incluye un abstract, asegurarse que *no* sea un texto largo sino sólo un mensaje muy corto.
- *Nunca* en la vida reutilizar el abstract del artículo para una presentación, *excepto* si el abstract es “Probamos que $P = NP$ ” o “Probamos que $P \neq NP$ ”

Numeración de Teoremas y Definiciones. Una manera común de estructurar globalmente los artículos y libros (matemáticos) es usar definiciones y teoremas numerados consecutivamente. Desafortunadamente, para una presentación la situación es un poco más complicada y se debe evitar numerar los teoremas. El público no tiene oportunidad de recordar estos números. *Nunca* decir cosas como “ahora, por el Teorema 2.5 que demostré antes, tenemos ...”. Sería mucho mejor referirse al Teorema de Kummer en lugar del Teorema 2.5. Si el Teorema 2.5 es algún teorema oscuro que no tiene su propio nombre (distinto a Teorema de Kummer o Teorema Principal o Segundo Teorema Principal o Lema Importante) entonces el público se habrá olvidado de él cuando se le refiera.

La única situación en que tiene sentido numerar los teoremas de una presentación, es cuando en una conferencia los estudiantes pueden leer paralelamente notas de la misma, donde los teoremas se numeran exactamente de la misma manera.

Si se numeran teoremas y definiciones, numerar todo consecutivamente. Así, si hay un teorema, un lema y una definición, se tendría Teorema 1, Lema 2 y Definición 3. Algunos prefieren numerar las tres cosas con un 1. El problema es que encontrar algo se vuelve una empresa casi imposible ya que el Teorema 2 podría venir después de la Definición 10 o al revés. Los artículos y, lo que es peor, los libros que tienen un Teorema 1 y una Definición 1 son un dolor de cabeza.

Bibliografías. También se puede presentar una bibliografía al final de la conferencia para que el público pueda ver que es posible una “lectura adicional”. Al agregar una bibliografía a una presentación, debemos tener presente lo siguiente:

- Es mala idea mostrar una bibliografía extensa en una presentación. Presentar pocas referencias. (Naturalmente, esto sólo se aplica a la conferencia misma, no al posible folleto.)

- Si se presentan más referencias de las que caben en una sola diapositiva se puede estar casi seguro que ninguna de ellas será recordada.
- Presentar referencias sólo si les piensa como “lectura adicional”. No presentar una lista de todas las cosas que se usaron (como sucede en un artículo).
- No se debe presentar una lista extensa de todos nuestros grandes artículos, *excepto* si se está dando una conferencia para conseguir trabajo.
- Usar el comando `\cite` puede ser confuso ya que el público tiene poca oportunidad de recordar las citas. Si se citan las referencias, siempre citar con el nombre completo del autor y año como “[López, 2003]” en lugar de “[2,4]” o “[Lo01,HLo02].”
- Si se quiere ser modesto, se puede abreviar nuestro nombre al auto citarnos como en “[Nickelsen y L.,2003]” o “[Nickelsen y L., 2003]”. Sin embargo, esto puede ser confuso para el público ya que frecuentemente no es inmediatamente claro quién exactamente podría ser “L”. Se recomienda usar el nombre completo.

Estructura del marco

Al igual que en la presentación entera, cada marco también se debe estructurar. Un marco que está lleno de bastante texto es muy difícil de seguir. Es nuestro trabajo estructurar el contenido de cada marco de tal manera que, con suerte, el público inmediatamente verá cuál información es importante, cuál es sólo un detalle, cómo se relaciona la información presentada, etc.

Título del marco

- Poner un título en cada marco. El título explica el contenido del marco a las personas que no siguieron todos los detalles en la diapositiva.
- A menos que se haya entendido la diapositiva completa, el título realmente debe *explicar* cosas no sólo dar un resumen oculto que no pueda entenderse. Por ejemplo, un título como “Copao” tendrá a todos confundidos sobre lo que podría tener esa diapositiva. Títulos como “Revisión de la Definición de Conjuntos Parcialmente Ordenados (Copao)” o “Un Orden Parcial sobre las Columnas de la Matriz de Genotipo” son *mucho* más informativos.
- Idealmente, los títulos en marcos consecutivos deben por sí mismos “contar una historia”.
- En español *siempre* se debe escribir la primera letra de todas las palabras en un título de marco con mayúsculas, excepto palabras como “a” o “el”, o usar *siempre* las letras minúsculas normales. *No* mezclar ambas reglas. Lo mismo aplica para los títulos de bloque. Por ejemplo, no usar títulos como “Una Revisión breve de máquinas de Turing”. Usar “Una Revisión Breve de Máquinas de Turing.” o “Una revisión breve de máquinas de Turing.” (Turing se escribe correctamente con letra mayúscula ya que es un nombre).
- En español el título de todo el documento se debe escribir en mayúsculas.

¿Qué tanto se puede poner en un marco?

- Un marco con poco texto es mejor que uno con demasiado texto. Un marco ordinario debe tener entre 20 y 40 palabras. El máximo debe ser aproximadamente de 80 palabras.
- No asumir que todos en el público son unos expertos en la materia. Aun cuando las personas que nos escuchan son expertos, puede que hayan escuchado ya hace varios años las cosas que consideramos obvias. Siempre se debe tener tiempo para un recordatorio rápido de lo que es exactamente una “clase de complejidad semántica” o un “orden parcial ω -completo”.

- Nunca poner en una diapositiva algo que no se va a explicar durante la conferencia, incluso si se trata de impresionar a alguien con lo complicada que es nuestra área. Sin embargo, se pueden explicar cosas que no están en una diapositiva.
- Manténgalo simple. Típicamente, el público verá una diapositiva durante menos de 50 segundos. No tendrán tiempo para dar rodeos a través de frases largas o fórmulas complicadas.
- Lance Forthnow afirmó: Los usuarios de PowerPoint dan las mejores conferencias. La justificación: dado que PowerPoint es tan malo para editar símbolos matemáticos, usan menos matemáticas, haciendo sus conferencias fáciles de entender. Hay algo de verdad en esto. La gran capacidad de \TeX para editar símbolos matemáticos puede hacernos caer en el uso de muchas fórmulas, más de lo que es necesario y saludable. Por ejemplo, en lugar de escribir “Dado que $|\{x \in 0, 1^* | x \sqsubseteq y\}| < 1$, tenemos ...” usar “Dado que y tiene sólo una cantidad finita de prefijos, tenemos ...”. Nos sorprenderá cuánto texto matemático puede ser reformulado en lenguaje llano o que simplemente se puede omitir. Naturalmente, si el argumento matemático es sobre lo que realmente se está hablando, como en una conferencia de matemáticas se puede hacer uso de la capacidad de edición matemática de \TeX .

Estructurando un marco

- Usar los ambientes de bloque `block`, `theorem`, `proof`, `example`, etc.
- Preferir enumeraciones y ambientes `\itemize` por encima del lenguaje llano.
- Usar `description` cuando se definan varias cosas.
- No usar más de dos niveles de “subitemizing”. BEAMER soporta tres niveles, pero no se debe usar aquel tercer nivel. En general, no se debería usar ni siquiera el segundo nivel. En cambio, usar buenos gráficos.
- No crear listados `itemize` o `enumerate` interminables.
- No descubra el listado por partes, sino línea por línea.
- El énfasis es una parte importante para crear estructura. Usar `\alert` para resaltar cosas importantes. Puede ser una sola palabra o una frase completa. Sin embargo, no hacer uso excesivo del resaltamiento ya que anulará el efecto.
- Usar columnas.
- *Nunca* usar notas a pie de página, rompen inútilmente el flujo de la lectura. O lo que se dice en ella es importante y se debe poner en texto normal; o no lo es y debe omitirse (*sobre todo* en una presentación).
- Usar `quote` o `quotation` para editar citas.
- No usar la opción `allowframebreaks` excepto para bibliografías extensas.
- No usar bibliografías extensas.

Escribiendo el Texto

- Usar frases cortas.
- Prefiera frases sobre oraciones completas. Por ejemplo, en lugar de “La figura de la izquierda muestra una máquina de Turing, la figura de la derecha muestra un autómata finito.” intentar “Izquierda: Una máquina de Turing. Derecha: Un autómata finito”. Aun mejor, convertir esto en un `itemize` o una descripción.
- Acentuar correctamente: no puntuar después de frases, puntuar completamente en y después de oraciones completas.

- *Nunca* usar un tamaño de fuente más pequeño para “meter más en un marco”. *Nunca en la vida* usar la opción *maligna shrink*.
- No unir palabras con guión. Si es completamente necesario, unir las palabras con guión “a mano” usando el comando `\-`.
- Hacer cambio de línea “a mano” usando el comando `\\`. No confiar en el cambio de línea automático. Interrumpa donde haya una pausa lógica. Por ejemplo, buenas rupturas en “el alfabeto grabado es más grande que el alfabeto de la entrada” son: antes de “es” y antes del segundo “el”. Malas rupturas serían: antes de cualquier “alfabeto” y antes de “más grande.”
- El texto y los números en las figuras deben tener el *mismo* tamaño que en el texto normal. Números ilegibles en los ejes normalmente arruinan una gráfica y su mensaje.

Elementos interactivos

Durante una presentación nos gustaría presentar las diapositivas en una forma absolutamente lineal, probablemente apretando la tecla *AvPág* una vez por cada diapositiva. Sin embargo, hay diferentes razones que nos podrían desviar de este orden lineal:

- La presentación puede tener “diferentes niveles de detalle” que pueden o no, ser evitados o extendidos dependiendo de la reacción del público.
- Se hacen preguntas y se deben mostrar diapositivas suplementarias.
- Se presenta un gráfico complicado y algunas de sus partes se tienen que poner en un plano general para explicar detalles.
- Se hacen preguntas de una diapositiva anterior que nos obliga a encontrarla y saltar a ella.

Realmente uno no se puede preparar contra esta última clase de preguntas. En este caso, se pueden usar las barras y símbolos de navegación para encontrar la diapositiva en la que se está interesado. Acerca de los primeros tres tipos de desviaciones, hay muchas cosas que se pueden hacer para preparar “desvíos planeados” o “atajos planeados”.

- Se pueden agregar “botones de salto”. Cuando se presiona tal botón se evita una parte bien definida de la conferencia. El botón de salto tiene dos ventajas sobre la opción de presionar repetidamente la tecla *Avpág*: primero, se posiciona inmediatamente en la ubicación correcta y segundo, la etiqueta del botón puede dar al público una noción visual de lo que se evitará exactamente. Por ejemplo, al presionar un botón de salto etiquetado con “Saltar Prueba” nadie empezará a adivinar sobre lo que se ha perdido.
- Se puede agregar un apéndice a la conferencia. El apéndice se mantiene “perfectamente separado” de la conferencia principal. Una vez que se “entra” a la parte del apéndice (probablemente por medio de un hipersalto), la estructura del apéndice se hace visible. Aquí se pueden poner todos los marcos que no se piensan mostrar durante el curso normal de la conferencia, pero que nos gustaría tener a la mano en caso de que hagan preguntas.
- Para crear desvíos se pueden agregar “botones ir a” y “botones de retorno”. Al presionar un botón *ir a* se saltará a cierta parte de la presentación donde se pueden mostrar más detalles. En esta parte hay un botón de *retorno* presente en cada diapositiva que nos regresará al lugar donde fue presionado el botón *ir a*.
- En BEAMER, se puede usar el comando `\againframe` para “continuar” marcos que se empezaron previamente en alguna parte, pero donde se han suprimido ciertos detalles. Se puede usar el comando `\againframe` en un punto muy avanzado, por ejemplo sólo en el apéndice para mostrar allí diapositivas adicionales.
- En BEAMER, se puede usar el comando `\framezoom` para crear ligas a partes complicadas de una diapositiva que fueron puestas en planos generales (zoom).

5.2. Usando gráficos

Los gráficos a menudo comunican conceptos o ideas de una manera más eficaz que un texto: Una imagen puede decir más que mil palabras (aunque a veces una palabra puede decir más que mil imágenes).

- Siempre que sea posible, poner por lo menos un gráfico en cada diapositiva. Las visualizaciones ayudan enormemente al público.
- Generalmente poner los gráficos a la izquierda del texto (usar el ambiente `columns`). En nuestra cultura que lee de izquierda a derecha, se mira primero hacia la izquierda.
- Los gráficos deben tener los mismos parámetros tipográficos que el texto: usar las mismas fuentes (el mismo tamaño además) en los gráficos como en el texto principal. Un punto pequeño en el gráfico debe tener exactamente el mismo tamaño que un punto pequeño en el texto. El ancho de línea debe ser igual al ancho de golpe usado para crear los glyphs de las fuentes. Por ejemplo, la fuente Modern Computer de 11 puntos no-negrita tiene un ancho de golpe de 0.4 puntos.
- Mientras los gráficos en mapa de bits, como las fotografías, pueden ser mucho más coloridos que el resto del texto, los gráficos vectoriales deben seguir la misma “lógica de color” del texto principal (como negro=líneas normales, rojo=partes resaltadas, verde=ejemplos, azul=estructura).
- Como en el texto, se debe explicar todo lo que se muestra en un gráfico. Detalles sin explicar crean confusión en el público ya que sienten que se han perdido algo importante. Tener cuidado al importar gráficos de un artículo o de alguna otra fuente, normalmente tienen más detalles de los que uno podrá explicar y se deben simplificar radicalmente.
- A veces la complejidad de un gráfico es intencional y se desea pasar mucho tiempo explicándolo a gran detalle. En este caso, a menudo nos encontraremos con el problema de que los detalles finos del gráfico son difíciles de percibir por el público y se debe usar el comando `\framezoom` para crear acercamientos planeados a partes interesantes del gráfico.

5.3. Usando animaciones y transiciones

- Usar animaciones para explicar la dinámica de sistemas, algoritmos, etc.
- *No* usar animaciones sólo para llamar la atención del público, frecuentemente los distrae del tema principal de la diapositiva. No importa cuan ingenioso parezca un teorema que rota y se desplaza y no importa que uno sienta que el público necesita algo de acción para tenerlo feliz, la mayoría de las personas en el público sentirán que se están burlando de ellos.
- *No* usar efectos especiales que distraigan, como “disolver” diapositivas, a menos que se tenga una muy buena razón para hacerlo. Si se usan, hacerlo poco. *Pueden* ser útiles en algunas situaciones: por ejemplo, se podría mostrar a un joven en una diapositiva, luego disolver esta diapositiva en otra que muestra a un hombre maduro. En este caso, la disolución le da al público el efecto visual que aquel joven “se convierte lentamente” en hombre.

5.4. Eligiendo temas apropiados

BEAMER viene con un buen número de temas diferentes. Al elegir un tema se debe tener presente lo siguiente:

- Temas diferentes son apropiados para ocasiones diferentes. No atarse a un tema favorito, se debe elegir el tema según la ocasión.
- Es más probable que una conferencia extensa requiera de ayuda para navegar que una corta. Cuando se da una conferencia de 90 minutos para estudiantes, se debe escoger un tema que siempre muestre una barra lateral con el tópico actual resaltado para que todos sepan exactamente cuál es el “estado” actual de la conferencia; cuando se pronuncia un discurso introductorio de diez minutos, es probable que parezca tonta una tabla de contenidos.

- Un tema que muestra el nombre del autor y su afiliación, es apropiado en situaciones donde el público probablemente no lo conoce (como en una conferencia). Si todos lo conocen, tener el nombre en cada una de las diapositivas simplemente es vanidad.
- Primero escoger un tema para la presentación que tenga un esquema apropiado para la conferencia.
- Luego se pueden cambiar los colores instalando un tema-color diferente. Esto puede cambiar drásticamente la apariencia de la presentación. Un tema “colorido” como **Berkeley** se verá menos llamativo si se usan los temas-color *seahorse* y *lily*.
- También se pueden cambiar las fuentes instalando un tema-fuente diferente.

5.5. Eligiendo colores apropiados

- Usar pocos colores. Los temas preparados ya son bastante coloridos (azul=estructura, rojo=alerta, verde=ejemplo). Se debe tener una *muy* buena razón para agregar más colores a cosas como código, texto matemático, etc.
- Tener cuidado al usar colores brillantes sobre fondo blanco, *sobre todo* al usar el color verde. Lo que se ve bien en el monitor puede que se vea mal durante una presentación debido a las diferentes clases de monitores, proyectores y copiadoras a color. Agregar mucho color negro sobre los colores puros al usar fondos luminosos.
- Maximizar el contraste. El texto normal debe ser negro sobre fondo blanco o al menos algo oscuro sobre algo muy luminoso. *Nunca* hacer cosas como “texto verde brillante sobre fondo verde no tan brillante.”
- El sombreado del fondo disminuye la legibilidad sin aumentar el contenido de la información. No agregar sombreado al fondo simplemente porque “de alguna manera se ve mejor”.
- El video inverso (texto brillante sobre fondo oscuro) puede ser un problema en ambientes luminosos ya que sólo un pequeño porcentaje del área de la presentación es iluminada por el proyector. Además, es más difícil reproducir el video inverso en copias impresas y transparencias.

5.6. Elegir fuentes y atributos de fuentes apropiados

Literalmente texto y fuentes nos rodean constantemente. Pensar en la última vez que no había texto a nuestro alrededor en un radio de diez metros. ¡Probablemente esto nunca ha pasado en la vida! (incluso en un traje de baño hay mucho texto). La historia de las fuentes es casi tan extensa como la historia de la civilización. En estos días hay diez millones de fuentes disponibles, algunas de las cuales son producto de cientos de años de optimización.

Elegir las fuentes apropiadas para una presentación no es trivial y una mala elección simplemente “hará que se vea mal”, o lo que es peor, hará que el público tenga problemas para leer las diapositivas. Esta guía no puede reemplazar ningún buen libro en tipografía, pero en esta sección se encuentran varias indicaciones que deben ayudar a organizar las fuentes para que nuestra presentación se vea bien. Una fuente tiene varios atributos como peso, familia o tamaño y estos tienen un impacto sobre la utilidad de la fuente en presentaciones. A continuación se describen estos atributos y se da un bosquejo de las ventajas y desventajas de las diferentes opciones.

Tamaño de la fuente

Quizás el atributo más evidente de una fuente es su tamaño. Las fuentes tradicionalmente se miden en “puntos”. ¿Cuanto mide un punto? Depende a quien se le pregunte. \TeX considera que un punto es la 72.27 parte de una pulgada (2.54 cm.). Por otro lado, PostScript y Adobe consideran que un punto es la 72ava parte de una pulgada (\TeX lo llama punto grande). Éstas son las diferencias entre el punto americano y el europeo. Una vez establecida la medida de un punto, afirmar que un texto es de “11pt”

significa que “la altura” de las letras de la fuente es de 11puntos. Sin embargo, esta “altura” proviene de la época en que las letras se imprimían con moldes de plomo y se refiere al tamaño vertical de las letras de plomo. Así que no necesita tener correlación alguna con la altura real de, digamos, la letra x o incluso la letra M. La letra x de 11pt Times de Adobe tendrá una altura que es diferente de la altura de la letra x de 11pt Times de UTC y la letra x de 11pt Helvética de Adobe tendrá también otra altura.

Resumiendo, el tamaño de la fuente poco tiene que ver con el tamaño real de las letras. En estos días se conviene que 10pt o 11pt es el tamaño en que una fuente debe ser impresa para su “lectura normal”. Las fuentes se diseñan para que se puedan leer con estos tamaños.

En una presentación las medidas clásicas de las fuentes evidentemente pierden su significado. Nadie podría leer un texto proyectado que tuviera realmente 11pt. En cambio, las letras proyectadas deben tener varios centímetros de altura. Así, no tiene sentido especificar de la manera usual el “tamaño de la fuente” para la presentación. En cambio, se debe pensar en el número de líneas que cabrían en la diapositiva si se le llenara completamente de texto línea por línea (nunca ponerlo en práctica). Dependiendo de cuan lejano este alejado el público de la proyección y de cuan grande es la proyección, deberían caber entre 10 y 20 líneas por diapositiva. A menos líneas, más legible será el texto.

En BEAMER, los tamaños predefinidos de las fuentes se eligen de tal manera que es difícil que quepa “demasiado” en una diapositiva. También, asegura que las diapositivas sean legibles incluso bajo malas condiciones como un cuarto grande y una pequeña área de proyección. No obstante, se pueden agrandar o encoger las fuentes si se piensa que será lo más apropiado para el ambiente de la presentación.

Una vez que se establece el tamaño del texto normal, los otros tamaños normalmente se definen en función de él. Por esta razón, L^AT_EX tiene comandos como `\large` o `\small`. El tamaño real que eligen estos comandos depende del tamaño del texto normal.

En una presentación se querrá usar una fuente muy pequeña para el texto en el título, pie de página o las barras laterales, ya que éste no es vital y lo lee el público sólo cuando quiere. Naturalmente, el texto aún debe ser lo suficientemente grande para que se pueda leer sin binoculares. Sin embargo, en un ambiente de presentación normal el público podrá incluso leer texto `\tiny` cuando sea necesario.

Sin embargo, el uso de fuentes pequeñas puede ser complicado. Muchas fuentes de PostScript apenas se reducen cuando se usan tamaños pequeños. Cuando se usa una fuente a menos de su tamaño normal, los caracteres en realidad serán impresos usando una “pluma” un poco más gruesa en vez de sólo reducir las cosas. Por esta razón, las fuentes maestras múltiples de alta calidad o las fuentes Computer Modern utilizan diversas fuentes para los caracteres pequeños y para los caracteres normales. Sin embargo, cuando se usan las fuentes Helvética normal o Times los caracteres apenas se reducen. Un problema similar surge cuando se usa una fuente brillante sobre un fondo oscuro. Incluso cuando se imprime sobre papel en alta resolución, el texto brillante-sobre-oscuro tiende a ser “desbordado” por el fondo oscuro. Cuando se representa el texto brillante-sobre-oscuro en una presentación este efecto puede ser peor, haciendo el texto casi imposible de leer. Se puede contrarrestar ambos efectos negativos usando la versión en negritas para el texto pequeño.

Por otro lado, se puede usar texto más grande para los títulos; sin embargo, esto no siempre tiene el efecto deseado. No porque el título del marco esté impreso en letras grandes nos garantiza que se leerá primero. De hecho, al echar un vistazo a la portada de nuestra revista favorita, notamos que el nombre de la revista fue editada con la fuente más grande, pero nuestra atención no siempre se dirige primero a los temas anunciados en la portada. Igualmente, en una tabla de contenidos es probable que nos enfoquemos primero en las entradas, no en las palabras “Tabla de Contenido”. Probablemente, no se descubriría un error de ortografía allí. En esencia, texto grande en el título de una página señala “insignificante dado que yo se qué esperar”. Para que, en lugar de usar un título del marco muy grande, también considere usar un título de marco de tamaño normal que se edite en negritas o en cursiva.

Familias de fuente

Otra propiedad central de cualquier fuente es su familia. Ejemplos de familias de fuente son Times, Helvética o Futura. Como el nombre lo sugiere, muchas fuentes diferentes pueden pertenecer a la misma familia. Por ejemplo, Times viene en diferentes tamaños, existe su versión en negritas, una versión en cursivas y así sucesivamente. Para evitar confusiones a las familias de fuente tipo Times a menudo se les llama las “fuentes Times”.

Hay dos grandes clases de familias de fuente: fuentes serif y fuentes sans-serif. Una fuente sans-serif es una fuente en la cual las letras no tienen serifs (sans es una palabra francesa que significa “sin”). Serifs son los pequeños ganchos al final de las líneas que constituyen una letra. La fuente que se está leyendo actualmente es una fuente serif. Para comparar, este texto está en una fuente sans-serif. Las fuentes sans-serif son (consideradas generalmente) más fáciles de leer cuando se les usa en una presentación. En una representación de baja resolución, los serifs disminuyen la legibilidad de una fuente. Sin embargo, en proyectores de alta resolución el texto serif es tan legible como el texto sans-serif. Una presentación editada en fuente serif crea una impresión más conservadora, que podría ser exactamente lo que se desea crear.

Probablemente se tendrán preinstaladas en el sistema diferentes familias de fuente. La fuente predefinida usada por $\text{T}_{\text{E}}\text{X}$ (y $\text{B}_{\text{E}}\text{A}^{\text{M}}\text{E}^{\text{R}}$) es la fuente Computer Modern. Es la familia de fuente original diseñada por Donald Knuth para el programa $\text{T}_{\text{E}}\text{X}$. Es una fuente madura que viene con casi todo lo que se podría desear para: alfabetos matemáticos extensos, versiones outline de PostScript, mayúsculas pequeñas (small caps), números de estilo antiguo, letras grandes y pequeñas especialmente diseñadas, etcétera.

Sin embargo, hay razones para usar familias de fuente distintas a Computer Modern:

- Las fuentes Computer Modern son un poco aburridas si se les ha visto demasiado. Usar otra fuente (pero no Times!) puede dar un aspecto fresco.
- Otras fuentes, sobre todo Times y Helvética, se representan mejor ya que parecen tener mejor trazado interior.
- La versión sans-serif de Computer Modern no está tan bien diseñada como la versión serif. De hecho, la versión sans-serif es, en esencia, la versión serif con parámetros de diseño diferentes, no un diseño independiente.
- Computer Modern necesita mucho más espacio que fuentes más económicas como Times (esto explica por qué Times es tan popular entre las personas que necesitan condensar su artículo en unas cuantas páginas). Es justo decir que la fuente Times fue diseñada específicamente para ser económica (la compañía que publica el periódico The Times necesitaba una fuente robusta pero económica en espacio).

Una pequeña selección de fuentes alternas a Computer Modern:

- Helvética se usa muy a menudo. Sin embargo, Helvética también tiende a verse aburrida (ya que la vemos por todas partes) y tiene una altura-x muy grande (la altura de la letra x comparada con la de la letra M). Una altura-x grande normalmente se considera buena para idiomas que rara vez usan letras mayúsculas (como el inglés) y no tan buena para idiomas que usan mucho letras mayúsculas (como el alemán). Una advertencia: la altura-x de la fuente Helvética es tan diferente de la altura-x de la fuente Times que al mezclar ambas en una misma línea provoca que el texto se vea extraño. Sin embargo, los paquetes para cargar Times y Helvética proveen opciones para arreglar esto.
- Futura es una fuente bonita que va bien con las presentaciones. Sus letras gruesas la hacen robusta contra la re-escala, inversión de colores y bajo contraste. Desafortunadamente, aunque es muy probable que de alguna forma esté instalada en alguna parte del sistema, hacer que $\text{T}_{\text{E}}\text{X}$ funcione con ella es un proceso complicado.
- Times es otra alternativa a Computer Modern. Su principal desventaja es que es una fuente serif que requiere un proyector de alta resolución. También se usa muy a menudo y es conocida por todos.

Las familias que no se deberían usar para texto normal son:

- Todas las fuentes monoespacio (como Courier).
- Fuentes script (letra manuscrita). El ancho de golpe de la letra es demasiado pequeña para una presentación.
- Fuentes serif más delicadas como Stempel e incluso quizás Garamond (aunque Garamond realmente es una fuente bonita para libros).
- Fuentes góticas. Sólo una pequeña fracción del público podrá leerlas fluidamente.

Hay una fuente popular que es un poco especial: Comic Sans de Microsoft. Por un lado, hay un website presionando para prohibir el uso de esta fuente. De hecho, el problema principal con la fuente es que no es muy legible y se ve terrible en la edición matemática. Por otro lado, esta fuente crea la impresión de una diapositiva “escrita a mano”, lo que le da a la presentación una apariencia natural. Hay que pensarlo dos veces antes de usar esta fuente, pero no hay que intimidarse.

Una de las reglas más importantes de la tipografía es que se deben usar en un texto tan pocas fuentes como sea posible. En particular, la sabiduría tipográfica dicta que no se deben usar más de dos familias diferentes en una página. Sin embargo, cuando se edita texto matemático, a menudo es necesario y útil usar fuentes de diferentes familias. Por ejemplo, era práctica común usar letras góticas para denotar vectores. También, los códigos de programas a menudo se editan con fuentes monoespacio. Si el público está acostumbrado a una cierta familia de fuentes para cierto tipo de texto, usar tal familia, sin tener en cuenta lo que dice la sabiduría tipográfica. Una práctica común en tipografía es usar una fuente sans serif para títulos y fuente serif para texto normal (revise su revista favorita). También se pueden usar dos fuentes sans serif diferentes o dos fuentes serif diferentes, pero entonces uno se tiene que asegurar que las fuentes luzcan lo “suficientemente diferentes”. Si parecen sólo ligeramente diferentes, la página se verá de algún modo “extraña”, pero el público no podrá distinguir por qué. Por ejemplo, no mezclar Arial y Helvética (son casi idénticas) o Computer Modern y Baskerville (son bastante similares). Una combinación de Gills Sans y Helvética es peligrosa pero quizás posible. Una combinación como Futura y Optima es correcta, por lo menos respecto a que las fuentes son muy diferentes.

Forma de la fuente: *Italic* y *Small Capital*

L^AT_EX presenta el concepto de *forma* de una fuente. Las únicas realmente importantes son *italics* (cursivas) y *small capitals* (versalitas). Una fuente *italic* es aquella en la que el texto se inclina ligeramente a la derecha *como ésta*. Cosas que se deben saber acerca de fuentes cursivas:

- Normalmente se usan cursivas en las novelas para expresar énfasis. Sin embargo, sobre todo con fuentes sans-serif, las cursivas no son lo “suficientemente fuertes” y se pierde el énfasis en una presentación. Es mejor usar un color diferente o texto en negritas para crear énfasis en las presentaciones.
- Si se mira atentamente, el texto *italic* no sólo está inclinado sino que realmente se usan letras diferentes (por ejemplo a y *a*). Sin embargo, esto es cierto sólo para texto serif no para texto sans-serif. El texto que sólo se inclina sin usar caracteres diferentes se le llama “inclinado” (slanted) en lugar de “cursivo”. A veces se le llama “oblicuo” al texto inclinado, pero a veces también se usa para texto en cursivas, así que es mejor evitar tal término. Los tipógrafos desaprueban el uso de texto serif inclinado y la consideran “tipografía barata de computadora”. Sin embargo, entre las personas que usan texto inclinado en sus libros se incluye a Donald Knuth. Si se quiere usar una fuente serif en una presentación, se debe poner también en cursivas y no inclinado.
- Los diferentes caracteres usados por serif *italics* han cambiado muy poco de las letras manuscritas originales que están basadas en el texto serif normal. Por esta razón, las *italics* serif crean la impresión de texto manuscrito, que puede ser conveniente para darle a una presentación un “toque personal”. Sin embargo, es más difícil de leer que el texto normal, así que no se debe usar más que para el texto de una línea de largo.

La segunda forma de fuente apoyada por T_EX son letras versalitas (small caps). Usarlas puede crear una conservadora e incluso una impresión formal, sólo unas palabras de precaución:

- Versalitas es diferente a texto todo en mayúsculas. Un texto en versalitas deja las letras mayúsculas normales inalteradas y usa versiones más pequeñas de letras mayúsculas para editar letras minúsculas normales. Así la palabra “German” se edita como GERMAN usando versalitas pero como GERMÁN usando letras mayúsculas.
- La fuente versalitas viene como “falsa” versalitas o como “auténtica” versalitas. La versalitas falsa se crea simplemente reduciendo las letras mayúsculas normales. Esto hace que las letras se vean demasiado delgadas. La versalitas real son versiones más pequeñas de letras mayúsculas especialmente diseñadas que tienen el mismo ancho de golpe que el texto normal.
- Las fuentes Computer Modern y la versión experta de fuentes PostScript vienen con versalitas auténtica (aunque por alguna razón inexplicable las versalitas de Computer Modern son un punto más grandes, pero el público lo perdonará ya que de todos modos no lo notará). Las fuentes “simples” de PostScript como Times o la Helvética fuera-de-la-caja sólo vienen con versalitas falsa.
- El texto editado en versalitas es más difícil de leer que el texto normal. La razón es que leemos viendo “la forma” de las palabras. Por ejemplo, la palabra “pala” se reconoce como una letra descendente, una letra normal, una letra ascendente y una letra normal. Uno tiene cierta dificultad para observar una falta de ortografía como en “pafa” y en “pela”. Las versalitas destruyen la forma de las palabras ya que pala, pafa y pela tienen la misma forma, por lo tanto hacen más difícil reconocer alguna falta. El público leerá el texto en versalitas más despacio que el texto normal. Por cierto, para descargo de responsabilidad legal se escribe a menudo en letras mayúsculas: no para hacerlas parecer más importante sino para hacerlas más difícil de leer.

Peso de la fuente

El “peso” de una fuente se refiere al grosor de las letras. Normalmente, las fuentes vienen como regular o en negritas. Con frecuencia existen versiones de seminegritas, ultranegritas (o negro), delgada o ultradelgada (tipo cabello). En tipografía se desaprueba el uso de una fuente en negritas para crear énfasis sobre todo dentro del texto normal, (palabras en negritas en medio de un texto normal se les llama “suciedad”). Para presentaciones esta regla en realidad no se aplica. En una diapositiva de una presentación normalmente hay poco texto y existen numerosos elementos que intentan llamar la atención del espectador. Usar las cursivas tradicionales para crear énfasis a menudo se pasa por alto. Así, usar texto en negritas parece una buena alternativa en una presentación. Sin embargo, una alternativa aún mejor es usar un color luminoso como el rojo para llamar la atención. Como se señaló antes, se debe usar texto en negritas para texto pequeño a menos que se use una fuente especialmente robusta como Futura.

6. Plantillas solución

En los subdirectorios del directorio `beamer/solutions` se encuentran varias *plantillas-solución* en diferentes idiomas. Una plantilla-solución es un texto escrito en $\text{T}_{\text{E}}\text{X}$ que “resuelve” un problema específico. Tal problema podría ser: “crear una charla de 20 minutos para una conferencia” o “crear una diapositiva que presente al próximo conferencista” o “crear una tabla que se descubra por partes”. Para tal problema, una plantilla-solución consiste de una mezcla de una plantilla y un ejemplo que se puede usar para resolver este problema particular. Basta copiar el archivo de la plantilla-solución (o partes de él) y ajustarlo libremente a nuestras necesidades.

La recolección de plantillas-solución de BEAMER apenas ha empezado y actualmente hay muy pocas de ellas. Se espera que en el futuro puedan estar disponibles más soluciones y se anima a los usuarios de la clase BEAMER a que desarrollen y envíen soluciones. Se anima también a los usuarios a que ayuden a traducir las plantillas soluciones a otros idiomas distintos al inglés y alemán.

La siguiente lista de plantillas-solución esta ordenada considerando la longitud de las conferencias para las que proporcionan una plantilla. Como siempre, las soluciones se pueden encontrar en el directorio `\MiKTeX2.5\tex\latex\beamer\solutions`.

Plantilla Solución `short-talks/speaker_introduction-ornate-2min`

- Presentar a otro conferencista.
- Duración de la plática de aproximadamente 2 minutos.
- Estilo adornado.

PRESEN-
TACIÓN Versión $\text{T}_{\text{E}}\text{X}$ disponible en los idiomas alemán e inglés.

Plantilla Solución `generic-talks/generic-ornate-15min-45min`

- Plantilla solución genérica para conferencias de cualquier asunto.
- Duración de la plática de entre 15 minutos y 45 minutos.
- Estilo adornado.

PRESEN-
TACIÓN Versión $\text{T}_{\text{E}}\text{X}$ disponible en los idiomas alemán e inglés.

Plantilla Solución `conference-talks/conference-ornate-20min`

- Plática para una conferencia o coloquio.
- Duración de aproximadamente 20 minutos.
- Estilo adornado.

PRESEN-
TACIÓN Versión $\text{T}_{\text{E}}\text{X}$ disponible en los idiomas alemán e inglés.

Construyendo una Presentación

Esta parte contiene una explicación de todos los comandos que se usan para crear presentaciones. Empieza con una sección que trata los comandos y ambientes usados para crear *marcos* y enseguida se explica la creación de cubiertas.

Las siguientes tres secciones involucran los comandos y métodos para *estructurar* una presentación: se trata la estructura *global estática*, la estructura *global interactiva* y la estructura *local*.

Las secciones posteriores tratan los gráficos y animaciones. La mayoría de este material se aplica a otros paquetes no sólo a BEAMER.

7. Creando marcos

7.1. El ambiente frame.

Una presentación consiste de una serie de marcos. Cada marco consiste de una serie de diapositivas. Un marco se crea usando el comando `\frame` o el ambiente `frame` (hacen lo mismo) y toma un parámetro: el contenido del marco. Todo texto que no se etiqueta con especificaciones de cubierta se muestra en todas las diapositivas del marco. Si un marco contiene comandos que tienen una especificación de cubierta entonces contendrá múltiples diapositivas, en caso contrario, sólo una.

```
\begin{frame}<especificación de cubierta>[<especificación de cubierta predefinida>][<opciones>]
{<título>}{<subtítulo>}
<Contenido del ambiente>
\end{frame}
```

La *<especificación de cubierta>* determina cuáles diapositivas de un marco se van a mostrar. Si se omite, el número de diapositivas se calcula automáticamente. El *<Contenido del ambiente>* puede ser texto normal en L^AT_EX, pero no puede contener el comando `\verb` o ambientes `verbatim` o cualquier otro ambiente que cambie el código de caracteres, a menos que se agregue la opción `fragile` al marco.

La opción *<título>* se detecta por medio de una llave izquierda, es decir, si la primera cosa en el marco es una llave izquierda se asume que lo que sigue es el título del marco. Igualmente, el *<subtítulo>* opcional se detecta por una llave izquierda después de *<título>*. También se puede dar el título y subtítulo usando los comandos `\frametitle` y `\framesubtitle`.

El comando `\frame` del L^AT_EX normal está disponible *dentro* de los marcos con su significado usual. Ambos tipos de marcos interior y exterior están disponibles como `\framelatex`.

Ejemplo:

```
\begin{frame}{T\'}{i}tulo}
  Alg\'}{u}n contenido.
\end{frame}
% Mismo efecto
\begin{frame}
  \frametitle{T\'}{i}tulo}
  Alg\'}{u}n contenido.
\end{frame}
```

Normalmente, el *<Contenido del ambiente>* completo se pone en una diapositiva. Si el texto no cabe en una diapositiva se apretará tanto como sea posible, se emitirá una advertencia y el texto simplemente se extenderá desagradablemente sobre la parte inferior. Se puede usar la opción `allowframebreaks` para provocar que el texto en el marco se divida en varias diapositivas, aunque no se pueden usar cubiertas.

La *<especificación de cubierta predefinida>* es un argumento opcional que se “detecta” según la siguiente regla: si el primer argumento opcional entre corchetes inicia con un `<`, entonces este argumento es una *<especificación de cubierta predefinida>*, en caso contrario es un argumento de *<opciones>* normal. Así `\begin{frame}[<+>][plain]` sería legal, pero también `\begin{frame}[plain]`.

El efecto de la *<especificación de cubierta predefinida>* es el siguiente: Cada comando o ambiente *dentro del marco* que acepta una especificación de acción, (esto incluye los comandos `\item`, el ambiente `actionenv`, `\action` y todo los ambientes block ver Sección 8.6.3) y que no es seguido por una especificación de cubierta obtiene la *<especificación de cubierta predefinida>* como su especificación. Al proporcionar una especificación incremental como `<+>`, esencialmente se causará que todos los bloques y que todas las enumeraciones se destapen por partes (los bloques internamente emplean especificaciones de acción).

Ejemplo: En este marco, el teorema se muestra desde la primera diapositiva, la palabra proof en un rectángulo a partir de la segunda, la tercera diapositiva contiene el primer y tercer objeto de la lista y la última contiene el segundo objeto. En total, este marco contendrá cuatro diapositivas.

```
\begin{frame}[<+>]
  \begin{theorem}
    $ A = B$ .
  \end{theorem}
  \begin{proof}
    \begin{itemize}
      \item Claramente, $ A = C$ .
      \item Como se mostr\ 'o antes, $ C = B$ .
      \item<3-> As\ 'i $ A = B$ .
    \end{itemize}
  \end{proof}
\end{frame}
```

Se pueden dar las siguientes *opciones*:

- **allowdisplaybreaks**=*(rompimiento deseable)* Causa que se emita el comando de AMSTEX `\allowdisplaybreaks[(rompimiento deseable)]` para el marco actual. El *(rompimiento deseable)* puede ser un valor entre 0 (que significa que nunca se pueden romper las fórmulas) y 4 (el valor por defecto, que significa que las fórmulas se pueden romper en cualquier parte sin penalización). La opción es sólo una conveniencia y tiene sentido sólo con la opción **allowsframebreaks**.
- **allowframebreaks**=*(fracción)*. Cuando se da esta opción el marco se rompe automáticamente en varios marcos si su texto no cabe en una sola diapositiva. Específicamente:
 1. No se permiten cubiertas.
 2. Cualquier nota para el marco que se crea usando el comando `\note` se insertará después de la primera página del marco.
 3. Cualquier pie de página para el marco se insertará en la última página del marco.
 4. Si hay un título de marco, cada una de las páginas tendrá este título con una nota especial agregada, indicando a qué página del marco corresponde. Por defecto, esta nota especial es un número romano. Sin embargo esto se puede cambiar si se usa la plantilla siguiente.

Plantilla-/Color-/Fuente-Beamer `frametitle continuation`

El texto de esta plantilla se inserta al final de cada título de un marco considerando además la opción (fracción) de **allowframebreaks**.

Las siguientes opciones de plantilla están predefinidas:

- **[default]** Instala un número romano como en la plantilla original. El número indica la página actual del marco.
- **[roman]** Alias para el valor por defecto.
- **[from second]** [*(texto)*] Instala una plantilla que inserta *(texto)* a partir de la segunda página de un marco en adelante. Por defecto, el texto que se inserta es `\insertcontinuationtext` que por defecto es *(cont.)*.

Están disponibles las siguientes inserciones:

- `\insertcontinuationcount` Inserta la página actual del marco en números árabes.
- `\insertcontinuationcountroman` Inserta la página actual del marco en números romanos.
- `\insertcontinuationtext` Sólo inserta el texto *(cont.)*.

Si un marco se divide en varias páginas, el material en cada página (excepto la última) llena por defecto sólo el 95 %. Así, se dejará algún espacio en la parte inferior y/o superior dependiendo de la opción de colocación vertical para el marco. Esto rinde un mejor resultado visual que un llenado del 100 %, que típicamente parece atestado de cosas. Se puede cambiar este porcentaje usando el argumento opcional *<fracción>*, donde 1 significa 100 % y 0.5 significa 50 %. Este porcentaje incluye el título del marco. Así para dividir un marco “aproximadamente por la mitad”, se debe dar 0.6 como *<fracción>*.

La mayoría de los detalles finos de T_EX normal para romper páginas también se aplican a esta opción. Por ejemplo, cuando queramos romper las ecuaciones automáticamente debemos usar el comando `\allowdisplaybreaks`. Al insertar los comandos `\break`, `\nobreak` y `\penalty` se controla el lugar donde debe ocurrir la ruptura. Los comandos `\pagebreak` y `\nopagebreak` también funcionan e incluyen sus opciones.

Dado que típicamente no se quiere que las divisiones de páginas de un marco también se apliquen al modo `article`, se puede agregar una especificación de modo como *<presentation>* para hacer que estos comandos sólo se apliquen en los modos de presentación. El comando `\framebreak` es una abreviatura de `\pagebreak<presentation>` y `\noframebreak` lo es de `\nopagebreak<presentation>`.

Usar esta opción es *maligno*. En una buena presentación se prepara cada diapositiva cuidadosamente y se piensa dos veces qué cosa poner en una diapositiva en lugar de en alguna otra. Al usar la opción `allowframebreaks` se invita a la creación de presentaciones horribles, interminables, que parecen más un “artículo proyectado en la pared” que una presentación.

No obstante, la opción tiene sus usos, puede ser conveniente para la división automática de bibliografías o ecuaciones largas.

Ejemplo:

```
\begin{frame}[allowframebreaks]{Referencias}
  \begin{thebibliography}{XX}
    \bibitem...
    ...
    \bibitem...
  \end{thebibliography}
\end{frame}
```

Ejemplo:

```
\begin{frame}[allowframebreaks,allowdisplaybreaks]{Una ecuación larga}
  \begin{align}
    \zeta(2) &= 1 + 1/4 + 1/9 + \cdots \\
    &= \dots \\
    &= \pi^2/6.
  \end{align}
\end{frame}
```

- **b,c,t** Causa que el marco se alinee verticalmente al fondo(b)/centro(c)/parte superior(t). Esto anula la política global de colocación que se gobierna por las opciones de clase `t` y `c`.
- `fragile=singleslide` le dice a BEAMER que el contenido del marco es “frágil”. Esto significa que el marco contiene texto que no se “interpreta como de costumbre”. Por ejemplo, esto se aplica a texto en ambiente `verbatim`, el cual se interpreta algo diferente del texto normal.

Si un marco contiene texto frágil se usan mecanismos internos distintos al editar el marco, para asegurar que en su interior los códigos de caracteres se puedan restablecer. El precio de cambiar a otro mecanismo interno es que no se pueden usar cubiertas o un archivo externo que necesita ser escrito y luego leído de vuelta (lo que no siempre es deseable).

En detalle, pasa lo siguiente cuando se da esta opción para (pdf)L^AT_EX normal: El contenido del marco se examina y luego se escribe en un archivo especial llamado $\langle nombre \rangle.vrb$ o, si se ha asignado una etiqueta al marco, $\langle nombre \rangle.\langle número de marco actual \rangle.vrb$. Luego, el marco se inicia de nuevo y el contenido de este archivo se lee de vuelta. Dado que ahora los códigos de caracteres se pueden modificar, esto permite el uso simultáneo de texto verbatim y cubiertas.

Para determinar la parte final del marco se usa la siguiente regla: La primera aparición de una sola línea conteniendo exactamente $\backslash end \{ \langle nombre del ambiente frame \rangle \}$ finaliza el marco.

El $\langle nombre del ambiente \rangle$ normalmente es `frame`, pero se puede cambiar usando la opción `environment`. Se necesita esta regla especial dado que cuando se recoge el contenido del marco, éste no se interpreta.

También se puede agregar la información opcional `=singleslide`, que le dice a BEAMER que el marco contiene sólo una diapositiva. En este caso el contenido del marco no se escribe en un archivo especial, sino que se interpreta directamente.

- `environment= $\langle nombre del ambiente marco \rangle$` . Esta opción es útil sólo junto con la opción `frágil` (pero no se usa para `fragile=singleslide`, sólo para `frágil`). El $\langle nombre del ambiente marco \rangle$ se usa para determinar el fin de la búsqueda al recoger el contenido del marco.

Normalmente, el marco acaba cuando se alcanza una línea que tiene $\backslash end \{ frame \}$. Sin embargo, si se usa $\backslash begin \{ frame \}$ dentro de otro ambiente se necesita usar esta opción:

Ejemplo:

```
\newenvironment{diapositiva}[1]
  {\begin{frame}[fragile,environment=diapositiva]
   \frametitle{#1}}
  {\end{frame}}
\begin{diapositiva}{Mi t' {i}tulo}
  Texto.
\end{diapositiva}
```

Si no se especifica la opción `environment=slide` en el ejemplo anterior, T_EX “perdería” el fin de la diapositiva, dado que no interpreta el texto al reunir el contenido del marco.

- `label= $\langle nombre \rangle$` Causa que el contenido del marco se almacene bajo el nombre $\langle nombre \rangle$ para su reanudación posterior usando el comando $\backslash againframe$. Además, en cada diapositiva del marco se crea una etiqueta con el nombre $\langle nombre \rangle \langle número de diapositiva \rangle$. Además en la *primera* diapositiva se crea una etiqueta con el nombre $\langle nombre \rangle$ (así que las etiquetas $\langle nombre \rangle$ y $\langle nombre \rangle \langle 1 \rangle$ apuntan a la misma diapositiva). Notar que en general las etiquetas, y éstas en particular, se pueden usar como blancos para hiperligas. Se puede usar esta opción junto con `fragile`.
- `plain` Causa que se supriman los encabezados, pie de páginas y barras laterales. Es útil para crear marcos sencillos con encabezados y pie de páginas diferentes o para crear un marco que muestra una imagen grande que lo llena completamente.

Ejemplo: Un marco con un gráfico que llena todo el marco:

```
\begin{frame}[plain]
  \begin{centering}%
    \pgfimage[height=\paperheight]{algúnarchivodeimagengrande}%
  \par%
```

```

\end{centering}%
\end{frame}
Ejemplo: Una página de título en que el encabezado y el pie de página se reemplazan por dos
gráficos.
\setbeamertemplate{title page}
{
\pgfdeclareimage[height=5cm]{superior}{logosuperior}
\pgfdeclareimage[height=5cm]{inferior}{logoinferior}
\pgfuseimage{logosuperior}
\vskip0pt plus 1filll
\begin{centering}
{\usebeamerfont{title}\usebeamercolor[fg]{title}\inserttitle\\}
\insertdate
\end{centering}
\vskip0pt plus 1filll
\pgfuseimage{logoinferior}
}
\begin{frame}[plain]
\titlepage
\end{frame}

```

- **shrink**=*⟨porcentaje de encogimiento mínimo⟩*. Causa que el texto en el marco se encoja si es demasiado grande para caber en él. Normalmente BEAMER edita primero el marco entero, luego mira el tamaño vertical del texto en el marco (excluyendo el título del marco), si éste es más grande que la altura de texto disponible menos la altura del título de marco, BEAMER calcula un factor de encogimiento y reduce el texto del marco por este factor de tal manera que el texto ahora cabe completamente. Usar esta opción causará automáticamente el uso de la opción **squeeze**.

Dado que el encogimiento tiene lugar sólo después de que todo ha sido editado, el texto encogido no llenará completamente el marco horizontalmente. Por esta razón, se puede especificar un *⟨porcentaje de encogimiento mínimo⟩*. Si se especifica este porcentaje, el marco se encogerá *por lo menos* en este porcentaje. En cuanto BEAMER sabe esto, puede aumentar proporcionalmente la anchura horizontal de manera que el texto encogido una vez más llene el marco entero. No obstante, si el porcentaje no es suficiente el texto se encogerá como sea necesario y se emitirá un mensaje de advertencia.

La mejor manera de usar esta opción es identificar marcos que están demasiado llenos, en los que absolutamente todo su texto tiene que caber en un solo marco. Entonces iniciamos especificando **shrink=5**, luego **shrink=10** y así sucesivamente, hasta que no se emita ninguna advertencia (o simplemente se ignora la advertencia cuando la cosas se ven de manera satisfactoria).

Usar esta opción es *muy malo*. Producirá cambios en el tamaño de las fuentes de diapositiva en diapositiva, lo que es una pesadilla tipográfica. *Siempre* se puede evitar su uso reestructurando y simplificando marcos, lo que producirá una mejor presentación.

Ejemplo:

```

\begin{frame}[shrink=5]
Una diapositiva maligna, sin terminar, que es 5\% más grande.
\end{frame}

```

- **squeeze** Causa que todos los espacios verticales en el texto se reduzcan tanto como sea posible. Comúnmente, esto sólo causa que el espacio vertical en enumeraciones o listados se reduzca a cero. Usar esta opción no es bueno pero tampoco malo.

ARTÍCULO En modo **article**, el ambiente **frame** no crea ninguna referencia visual al marco original (no se muestra ningún marco). Más bien, el texto del marco se inserta en el texto normal. Para cambiar esto, se pueden modificar las plantillas **begin frame** y **end frame**. Para suprimir un marco en este modo, por ejemplo, se puede especificar **<presentation>** como especificación de cubierta.

Plantilla-Beamer **frame begin**

En modo **article** (y sólo allí), el texto de esta plantilla se inserta al principio de cada marco. Se puede usar, digamos, para comenzar un ambiente **minipage** al principio de un marco o para insertar una barra horizontal o cualquier otra cosa.

Plantilla-Beamer **frame end**

En modo **article**, el texto de esta plantilla se inserta al final de cada marco.

Se *puede* usar el ambiente **frame** dentro de otros ambientes así

```
\newenvironment{diapositiva}{\begin{frame}}{\end{frame}}
```

o así

```
\newenvironment{mimarco}[1]
  {\begin{frame}[fragile,environment=mimarco]\frametitle{#1}}
  {\end{frame}}
```

Sin embargo, las mecánicas reales son algo sensibles dado que la “recolección” del contenido del marco no es fácil, así que no se debe intentar algo demasiado extravagante. Como regla general, el principio del ambiente puede ser todo lo bonito que se quiera, pero para finalizar debe acabar con **\end{frame}** y no debe contener ningún **\end{xxx}**. Si se intenta hacer algo realmente complejo es probable que se fracase. Si se necesita algún **\end{xxx}** allí, se define un nuevo comando que contenga estas cosas como en el ejemplo siguiente:

```
\newenvironment{milistado}
  {\begin{frame}\startitemizeframe}
  {\stopitemizeframe\end{frame}}
\newcommand\startitemizeframe{\begin{bfseries}\begin{itemize}}
\newcommand\stopitemizeframe{\end{itemize}\end{bfseries}}
\begin{milistado}
\item Primera cosa
\end{milistado}
```

7.2. Componentes de un marco

Cada marco consiste de varios componentes:

1. un encabezado y un pie de página,
2. una barra lateral izquierda y otra derecha,
3. barras de navegación,
4. símbolos de navegación,
5. un logotipo,
6. un título del marco,
7. un fondo y

8. algún contenido del marco.

Un marco no necesariamente tiene todos estos componentes. Normalmente, los primeros tres se establecen automáticamente por el tema que se está usando.

El encabezado y el pie de página

El encabezado de un marco es el área correspondiente a su parte superior. Si no está vacío, debe mostrar cierta información que ayude al público a orientarse durante la charla. Asimismo, el pie de página es el área en la parte inferior del marco. BEAMER no usa los mecanismos estándar de L^AT_EX para editar el encabezado y el pie de página. En cambio, para editarlos, se usan las plantillas `headline` y `footline`.

El tamaño del encabezado y del pie de página se determina como sigue: su ancho es el mismo del papel, su altura se determina provisionalmente al editar el encabezado y el pie de página justo después del comando `\begin{document}`. La cabeza del encabezado y del pie de página se “congela” en ese punto y se usa a lo largo de todo el documento, incluso si varía posteriormente la altura del encabezado y del pie de página (que no deberían). La apariencia del encabezado y del pie de página se determina por las siguientes plantillas:

Plantilla-/Color-/Fuente-Beamer `headline`

Esta plantilla se usa para editar el encabezado. El `color-` y la fuente-BEAMER `headline` se instalan al principio. El fondo del `color-BEAMER` no se usa por defecto, es decir, no se dibuja ningún rectángulo de fondo detrás del encabezado y del pie de página (esto puede cambiar en el futuro con la introducción de un lienzo `headline` y `footline`).

La anchura del encabezado es la del papel. La altura se determina automáticamente como ya se describió. El encabezado se edita en modo vertical con el salto de renglón inhabilitado, además el salto de párrafo se establece en cero. Dentro de esta plantilla el significado del comando `\\` cambia, aquí sólo inserta una coma.

Ejemplo:

```
\setbeamertemplate{headline}
{
  \%
  \begin{beamercolorbox}{section in head/foot}
  \vskip2pt\insertnavigation{\paperwidth}\vskip2pt
  \end{beamercolorbox}%
}
```

Las siguientes opciones de la plantilla están predefinidas:

- [`default`] El valor por defecto es un encabezado vacío. Para tener el encabezado predefinido de versiones anteriores de la clase BEAMER, usar el tema `compatibility`.
- [`infolines theme`] Esta opción está disponible (y se usa) si se carga el tema exterior `infolines`. El encabezado muestra la sección y subsección actual.
- [`miniframes theme`] Esta opción está disponible si se carga el tema exterior `miniframes`. El encabezado muestra las secciones con pequeños mini marcos (pulsables) que están por debajo de ellas.
- [`sidebar theme`] Esta opción está disponible si se carga el tema exterior `sidebar` y si la altura del encabezado (y de la opción del tema `sidebar`) no es cero. En este caso, el encabezado es una barra vacía del color de fondo `frametitle` con el logotipo a la izquierda o derecha de esta barra.
- [`smoothtree theme`] Esta opción está disponible si se carga el tema exterior `smoothtree`. Se muestra un árbol de navegación (atenuado) en el encabezado.
- [`smoothbars theme`] Esta opción está disponible si se carga el tema exterior `smoothbars`. Se muestra una versión “atenuada” del encabezado `miniframes`.

- `[tree]` Esta opción está disponible si se carga el tema exterior `tree`. Se muestra en el encabezado un árbol navegacional.
- `[split theme]` Esta opción está disponible si se carga el tema exterior `split`. El encabezado se divide en una parte izquierda que muestra las secciones y una derecha que muestra las subsecciones.
- `[text line]` $\langle texto \rangle$ El encabezado se edita más o menos como si fuera una línea de texto normal con el $\langle texto \rangle$ como contenido. Los márgenes derecho e izquierdo se establecen como si fueran los márgenes de un texto normal. El $\langle texto \rangle$ se edita dentro de un `\hbox`, mientras que el encabezado se edita normalmente en modo vertical.

Dentro de la plantilla se pueden usar numerosas inserciones:

- `\insertnavigation` $\langle ancho \rangle$ Inserta en una plantilla una barra de navegación horizontal del $\langle ancho \rangle$ dado. La barra lista las secciones y debajo de ellas un mini marco para cada marco en esa sección.
- `\insertpagenumber` Inserta en una plantilla el número de la página actual.
- `\insertsection` Inserta en una plantilla la sección actual.
- `\insertsectionnavigation` $\langle ancho \rangle$ Inserta una barra de navegación vertical que contiene todas las secciones, con la sección actual destacada de las demás.
- `\insertsectionnavigationhorizontal` $\langle ancho \rangle$ $\langle inserción izquierda \rangle$ $\langle inserción derecha \rangle$

Inserta una barra de navegación horizontal que contiene todas las secciones, con la sección actual destacada. La $\langle inserción izquierda \rangle$ se inserta a la izquierda de las secciones, la $\langle inserción derecha \rangle$ a la derecha. Al insertar un triple llenado (un `filll`) se puede empujar la barra hacia la izquierda o la derecha.

Ejemplo: `\insertsectionnavigationhorizontal{.5\textwidth}{\hskip0pt plus1filll}`

- `\insertshortauthor` $[\langle opciones \rangle]$ Inserta la versión corta de los datos del autor en una plantilla. El texto se imprime en una línea larga, se suprimen los cambios de línea introducidos por el comando `\.`. Se pueden dar las siguientes opciones:
 - `width` $=\langle ancho \rangle$ Causa que el texto se ponga en una mini página multilínea del tamaño dado. Por defecto, los cambios de línea todavía se suprimen.
 - `center` Centra el texto dentro de la mini página creada con la opción previa, en vez de alinearla a la izquierda.
 - `respectlinebreaks` Causa que se respeten los cambios de línea introducidos por el comando `\.`.

Ejemplo: `\insertauthor[width={3cm},center,respectlinebreaks]`

Los siguientes comandos también aceptan las opciones `width`, `center` y `respectlinebreaks`.

- `\insertshortdate` $[\langle opciones \rangle]$ Inserta en una plantilla la versión corta de la fecha.
- `\insertshortinstitute` $[\langle opciones \rangle]$ Inserta en una plantilla la versión corta del instituto.
- `\insertshortpart` $[\langle opciones \rangle]$ Inserta en una plantilla la versión corta del nombre de la parte.
- `\insertshorttitle` $[\langle opciones \rangle]$ Inserta en una plantilla la versión corta del título del documento.
- `\insertshortsubtitle` $[\langle opciones \rangle]$ Inserta la versión corta del subtítulo del documento.
- `\insertsubsection` Inserta en una plantilla la subsección actual.

- `\insertsubsubsection` Inserta en una plantilla la sub-subsección actual.
- `\insertsubsectionnavigation{⟨ancho⟩}` Inserta una barra de navegación vertical que contiene todas las subsecciones de la sección actual, con la subsección actual destacada.
- `\insertsubsectionnavigationhorizontal{⟨ancho⟩}{⟨inserción izquierda⟩}{⟨inserción derecha⟩}`
Ver `\insertsectionnavigationhorizontal`.
- `\insertverticalnavigation{⟨ancho⟩}` Inserta en una plantilla una barra de navegación vertical del `⟨ancho⟩` proporcionado. La barra muestra una tabla de contenidos pequeña. Las líneas individuales se editan usando las plantillas `section in head/foot` y `subsection in head/foot`.
- `\insertframenumber` Inserta en una plantilla el número del marco actual (no de la diapositiva).
- `\inserttotalframenumber` Inserta en una plantilla el número total de marcos (no de las diapositivas). El número es correcto sólo en la segunda compilación en \LaTeX del documento.
- `\insertframestartpage` Inserta el número de la primera página del marco actual.
- `\insertframeendpage` Inserta el número de la última página del marco actual.
- `\insertsubsectionstartpage` Inserta el número de la primera página de la subsección actual.
- `\insertsubsectionendpage` Inserta el número de la última página de la subsección actual.
- `\insertsectionstartpage` Inserta el número de la primera página de la sección actual.
- `\insertsectionendpage` Inserta el número de la última página de la sección actual.
- `\insertpartstartpage` Inserta el número de la primera página de la parte actual.
- `\insertpartendpage` Inserta el número de la última página de la parte actual.
- `\insertpresentationstartpage` Inserta el número de la primera página de la presentación.
- `\insertpresentationendpage` Inserta el número de la última página de la presentación (excluyendo el apéndice).
- `\insertappendixstartpage` Inserta el número de la primera página del apéndice. Si no hay apéndice, este número es la última página del documento.
- `\insertappendixendpage` Inserta el número de la última página del apéndice. Si no hay apéndice, este número es la última página del documento.
- `\insertdocumentstartpage` Inserta un 1.
- `\insertdocumentendpage` Inserta el número de la última página del documento (incluyendo el apéndice).

Plantilla-/Color-/Fuente-Beamer `footline`

Esta plantilla se comporta exactamente de la misma manera que `headline`. Se puede notar que `BEAMER` comúnmente agrega un espacio de 4 puntos entre el fondo del texto del marco y la parte superior del pie de página, lo que a veces es bastante molesto.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` El valor por defecto es un pie de página vacío. Por defecto, los símbolos de navegación no son parte del pie de página. Más bien, son parte de una barra lateral derecha (invisible).
- `[infolines theme]` Esta opción está disponible (y se usa) si se carga el tema exterior `infolines`. El pie de página muestra cosas como el nombre del autor y el título de la charla.

- `[miniframes theme]` Esta opción esta disponible si se carga el tema exterior `miniframes`. Dependiendo de las opciones exactas que se usen al cargar el tema `miniframes`, se pueden mostrar diferentes cosas en el pie de página.
- `[page number]` Muestra el número de la página actual en el pie de página.
- `[frame number]` Muestra el número del marco actual en el pie de página.
- `[split]` Esta opción está disponible si se carga el tema exterior `split`. El pie de página (como en el encabezado) se divide en una parte izquierda que muestra el nombre del autor y una derecha que muestra el título de la charla.
- `[text line]` $\{\langle texto \rangle\}$ El pie de página se edita más o menos como si fuera una línea de texto normal con el $\langle texto \rangle$ como contenido. Los márgenes izquierdo y derecho se establecen como si fueran los márgenes de un texto normal. El $\langle texto \rangle$ se edita dentro de un `\hbox`, mientras que el encabezado normalmente se edita en modo vertical. Usar el comando `\strut` en alguna parte de tal línea podría ser una idea buena.

Se pueden usar las mismas inserciones que para los encabezados.

Color-, Fuente-Beamer `page number in head/foot`

Este color- y fuente-BEAMER se usan para editar el número de la página o del marco en el pie de página.

Las barras laterales

Las barras laterales son áreas verticales que van del extremo inferior del encabezado a la parte superior del pie de página. Allí puede haber una barra lateral a la izquierda y otra a la derecha (o incluso ambas). Las barras laterales pueden mostrar una tabla de contenidos, pero también se podrían agregar por razones completamente estéticas.

Cuando se instala una plantilla de barra lateral, se debe especificar explícitamente el tamaño horizontal de la barra lateral usando el comando `\setbeamerwidth` con la opción `sidebar left width` o `sidebar right width`. El tamaño vertical se determina automáticamente. Cada barra lateral tiene su propio lienzo de fondo que se puede establecer usando las plantillas de lienzo para barra lateral.

Agregar una barra lateral de cierto tamaño, digamos 1cm, hará 1cm más estrecho al texto principal. La distancia entre el lado interno de una barra lateral y el lado exterior del texto, que también se especifica con el comando `\setbeamerwidth` con la opción `text margin left` y su contraparte para el margen derecho, no cambia cuando se instala una barra lateral.

Internamente, las barras laterales se editan al mostrarse como parte del encabezado. La clase BEAMER guarda la pista de seis dimensiones, tres para cada lado: las variables `\beamer@leftsidebar` y `\beamer@rightsidebar` almacenan las medidas (horizontales) de las barras laterales, las variables `\beamer@leftmargin` y `\beamer@rightmargin` almacenan la distancia entre la barra lateral y el texto y los macros `\Gm@lmargin` y `\Gm@rmargin` guardan la distancia del borde del papel al borde del texto. Así que la suma de `\beamer@leftsidebar` y `\beamer@leftmargin` es exactamente `\Gm@lmargin`. Así, si se desea poner algún texto justo al lado de la barra lateral izquierda, para conseguirlo se podría escribir `\hskip-\beamer@leftmargin`.

Plantilla-, Color-, Fuente-Beamer `sidebar left`

Color/fuente padre: `sidebar`

Esta plantilla se usa al editar la barra lateral izquierda. Como se expresó antes, el tamaño de la barra lateral izquierda se fija usando el comando `\setbeamerwidth{sidebar width left=2cm}`

Si las barras laterales son demasiado grandes BEAMER no las sujeta automáticamente. Cuando se edita la barra lateral se pone dentro de un `\vbox`. Actualmente uno debe establecer cosas como `\hsize o`

`\parskip` por iniciativa propia.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` Instala una plantilla vacía.
- `[sidebar theme]` Esta opción está disponible si se carga el tema exterior `sidebar` con la opción `left`. En este caso, esta opción se selecciona automáticamente. Muestra una mini tabla de contenidos en la barra lateral.

Plantilla-/Color-/Fuente-Beamer `sidebar right`

Color/font parents: `sidebar`

Funciona de la misma forma que la plantilla para la barra lateral izquierda. Las siguientes opciones de la plantilla están predefinidas:

- `[default]` En este caso la barra lateral derecha tiene ancho cero. No obstante, muestra símbolos de navegación y, si se instaló, un logotipo al fondo de la barra lateral, destacándose a la izquierda del texto.
- `[sidebar theme]` Esta opción está disponible si se carga el tema exterior `sidebar` con la opción `left`. En este caso, esta opción se selecciona automáticamente. Muestra una mini tabla de contenidos en la barra lateral.

Plantilla-Beamer `sidebar canvas left`.

Como el lienzo de fondo global, este lienzo se dibuja detrás del texto real de la barra lateral. Esta plantilla normalmente debe insertar un rectángulo del tamaño de la barra lateral, aunque una altura demasiado grande no conducirá a un error o advertencia. Cuando se llama a esta plantilla se instala el color-BEAMER `sidebar left`.

Las siguientes opciones de la plantilla están predefinidas:

- Por defecto, como lienzo de la barra lateral se usa un rectángulo grande coloreado con `sidebar.bg`. Sin embargo, si el fondo de `sidebar` está vacío, no se dibuja nada y el lienzo es “transparente”.
- `[vertical shading]` [*opciones de color*] instala un fondo sombreado verticalmente. Se pueden dar las siguientes *opciones de color*:
 - `top=<color>` especifica el color en la parte superior de la barra lateral. Por defecto, se usa 25% del primer plano del color-BEAMER `palette primary`.
 - `bottom=<color>` especifica el color en la parte inferior de la barra lateral (más precisamente, a una distancia de la altura de la página por debajo de la parte superior de la barra lateral). Por defecto, se usa el fondo de `normal text` al momento de la invocación de este comando.
 - `middle=<color>` especifica el color para la parte media de la barra lateral. Si se da esta opción, el sombreado cambia del color de la parte inferior a este color y luego al color de la parte superior.
 - `midpoint=<factor>` especifica en que punto de la página se usará el color medio. Un factor de 0 es la parte inferior de la página, un factor de 1 es la superior. El valor por defecto es 0.5 que es al medio.

Notamos que aquí se deben dar colores de L^AT_EX “reales”. A menudo esto hace necesaria la invocación del comando `\usebeamercolor` antes de que se pueda usar esta opción. El ancho de la barra lateral se debe establecer antes de usar esta opción.

Ejemplo: Un elegante, pero no muy útil sombreado:

```
{\usebeamercolor{palette primary}}
\setbeamertemplate{sidebar canvas}[vertical shading]
[top=palette primary.bg,middle=white,bottom=palette primary.bg]
```

- `[horizontal shading]` [*opciones de color*] instala un fondo sombreado horizontalmente. Se pueden dar las siguientes *opciones de color*:
 - `left=<color>` especifica el color a la izquierda de la barra lateral.
 - `right=<color>` especifica el color a la derecha de la barra lateral.
 - `middle=<color>` especifica el color a la mitad de la barra lateral.
 - `midpoint=<factor>` especifica en que punto de la barra lateral se usará el color medio. Un factor de 0 es la izquierda de la barra lateral, un factor de 1 la derecha. El valor por defecto es 0.5 que es al medio.

Ejemplo: Agrega dos “pilares”

```
\setbeamerwidth{sidebar width left=0.5cm,sidebar width right=0.5cm}
{\usebeamercolor{sidebar}}
\setbeamertemplate{sidebar canvas left}[horizontal shading]
[left=white,middle=sidebar.bg,right=white]
\setbeamertemplate{sidebar canvas right}[horizontal shading]
[left=white,middle=sidebar.bg,right=white]
```

Plantilla-Beamer `sidebar canvas right`

Funciona exactamente como en el caso del lado izquierdo.

Las barras de navegación

Muchos temas instalan un encabezado o una barra lateral que muestra una *barra de navegación*. Aunque estas barras de navegación ocupan poco espacio, a menudo son útiles por dos razones:

- Proporcionan al público una retroalimentación visual de cuánto se ha cubierto de la charla y lo que todavía está por venir.
- Se puede pulsar sobre todas las partes de la barra de navegación. Directamente se “saltará” a la parte sobre la que se pulsó. Esto es particularmente útil para saltar ciertas partes de la charla, además durante una “sesión de preguntas” se puede volver a un marco particular sobre el que alguien ha preguntado algo.

Algunas barras de navegación se pueden “comprimir” usando la siguiente opción:

```
\documentclass[compress]{beamer}
```

Intenta que todas las barras de navegación sean tan pequeñas como sea posible. Por ejemplo, todas las pequeñas representaciones de marco en la barra de navegación para una sola sección se muestran una junto a la otra. Normalmente, las representaciones para subsecciones diferentes se muestran en líneas diferentes. Además, las representaciones de navegación de sección y subsección se comprimen en una sola línea.

Algunos temas usan el comando `\insertnavigation` para insertar una barra de navegación en el encabezado. Dentro de esta barra se muestran iconos pequeños (llamados “mini marcos”) que representan los marcos de una presentación. Cuando se pulsa sobre tal icono pasa lo siguiente:

- Si se pulsa sobre (el icono de) cualquier otro marco distinto del actual, la presentación saltará a la primera diapositiva del marco sobre la que se pulsó.
- Si se pulsa sobre el marco actual y si no se está en la última diapositiva de éste, se saltará a la última diapositiva del marco.
- Si se pulsa sobre el marco actual y se está en la última diapositiva, se saltará a la primera diapositiva del marco.

Por las reglas anteriores se puede:

- Saltar al principio de un marco desde cualquier otra parte pulsando una vez sobre su icono.

- Saltar al final de un marco desde cualquier otra parte pulsando dos veces sobre su icono.
- Brincar el resto del marco actual pulsando una vez sobre su icono.

Plantilla-Beamer Padre `mini frames`

Esta plantilla padre tiene los productos `mini frame` y `mini frame in current section`.

Ejemplo: `\setbeamertemplate{mini frames}[box]`

Las siguientes opciones de plantilla están predefinidas:

- `[default]` muestra círculos pequeños como mini marcos.
- `[box]` muestra rectángulos pequeños como mini marcos.
- `[tick]` muestra barras verticales pequeñas como mini marcos.

Plantilla-/Color-/Fuente-Beamer `mini frame`

Esta plantilla se usa para representar el mini marco del marco actual en una barra de navegación. El ancho de la plantilla se ignora. En cambio, cuando se muestran múltiples mini marcos, su posición se calcula basada en los tamaños-BEAMER `mini frame size` y `mini frame offset`. Ver el comando `\setbeamersize` para una descripción de cómo cambiarlos.

Plantilla-Beamer `mini frame in current subsection`

Esta plantilla se usa para representar cada mini marco de los marcos en la subsección actual que no están en el marco actual. El color/fuente-BEAMER `mini frame` se instala antes del uso de esta plantilla.

Plantilla-Beamer `mini frame in other subsection`

Esta plantilla se usa para representar los mini marcos (del marco) de subsecciones distintas a la actual.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` [`<porcentaje>`] Por defecto, esta plantilla muestra `mini frame in current subsection`, excepto que el color se cambia primero a `fg!<porcentaje>!bg`. El `<porcentaje>` predefinido es 50%.

Ejemplo: Para tener una representación extremadamente “sombreada” de los marcos fuera de la subsección actual, se puede usar lo siguiente:

`\setbeamertemplate{mini frame in other subsection}[default][20]`

Ejemplo: Para representar todos los mini marcos distintos al actual, usar `\setbeamertemplate{mini frame in other subsection}[default][100]`

Algunos temas presentan las secciones y/o subsecciones en las barras de navegación. Al pulsar sobre una sección o subsección en la barra de navegación, se saltará a esa sección. Pulsar sobre una sección es particularmente útil si la sección empieza con un `\tableofcontents[currentsection]`, dado que se puede usar para saltar a diferentes subsecciones.

Plantilla-/Color-/Fuente-Beamer `section in head/foot`

Esta plantilla se usa para representar el acceso a una sección que se encuentra en el encabezado o en el pie de página. El fondo del color-BEAMER se usa típicamente como el fondo del “área” total donde se muestran en el encabezado los accesos de sección. Normalmente uno no puede usar esta plantilla dado que la inserción `\insertsectionhead` se establece correctamente sólo cuando se está editando una lista de secciones en el encabezado.

La plantilla predefinida sólo inserta el nombre de la sección. Las siguientes inserciones son útiles:

- `\insertsectionhead` Inserta el nombre de la sección que va aparecer en la barra de navegación.

- `\insertsectionheadnumber` Inserta el número de la sección que va aparecer en la barra de navegación.
- `\insertpartheadnumber` Inserta el número de la parte de la sección o subsección actual que va aparecer en la barra de navegación.

Plantilla-Beamer `section in head/foot shaded`

Esta plantilla se usa en lugar de `section in head/foot` para editar secciones que actualmente están sombreadas. Tal sombreado se aplica normalmente a todas las secciones excepto a la actual. Notar que la plantilla *no* tiene color y fuente propia. Cuando se invoca esta plantilla, se establecerá la fuente y color-BEAMER de `section in head/foot`. Entonces, al inicio de la plantilla, típicamente se cambiaría el color actual o se iniciaría un ambiente `colormixin`.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` [`<porcentaje>`] La plantilla predefinida cambia el color actual a `fg!<porcentaje>!bg`. Esto causa que el color actual se vuelva “desteñido” o “sombreado”. El porcentaje predefinido es 50.

Ejemplo: Se puede usar lo siguiente para hacer que los accesos sombreados sean muy “luminosos”:

```
\setbeamertemplate{section in head/foot shaded}[default][20]
```

Plantilla-/Color-/Fuente-Beamer `section in sidebar`

Esta plantilla se usa para representar el acceso a una sección que se encuentra en la barra lateral, como parte de una mini tabla de contenidos que se muestra allí. El fondo del color-BEAMER se usa como fondo para la entrada. Exactamente como en `section in head/foot`, normalmente uno no puede usar esta plantilla y se debe usar también `\insertsectionhead` para insertar el nombre de la sección.

Por única vez, no se instala ningún valor predefinido para esta plantilla.

Las siguientes opciones de la plantilla están predefinidas:

- `[sidebar theme]` Esta plantilla, que sólo está disponible si se carga el tema exterior `sidebar`, inserta una barra que muestra el nombre de la sección con el primer plano y fondo usando el color-BEAMER. El ancho de la barra es el mismo que el de toda la barra lateral.

Se pueden usar las mismas inserciones que en `section in head/foot`.

Plantilla-/Color-Beamer `section in sidebar shaded`

Esta plantilla se usa en lugar de `section in sidebar` para editar secciones que actualmente están sombreadas. Tal sombreado se aplica normalmente a todas las secciones excepto a la actual. Al contrario que en `section in head/foot shaded`, esta plantilla *tiene* su propio color-BEAMER.

La siguiente opción de la plantilla está predefinida:

- `[sidebar theme]` Hace lo mismo que la versión no sombreada, sólo que usa un color-BEAMER diferente.

Plantilla-/Color-/Fuente-Beamer `subsection in head/foot`

Esta plantilla funciona como `section in head/foot`, sólo que para subsecciones.

- `\insertsubsectionhead` funciona como `\insertsectionhead`.
- `\insertsubsectionheadnumber` funciona como `\insertsectionheadnumber`.

Plantilla-Beamer `subsection in head/foot shaded`

Esta plantilla funciona como `section in head/foot shaded`, sólo que para las subsecciones.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` [*porcentaje*] funciona como la opción correspondiente para las secciones.

Ejemplo: `\setbeamertemplate{section in head/foot shaded}[default][20]`
`\setbeamertemplate{subsection in head/foot shaded}[default][20]`

Plantilla-/Color-/Fuente-Beamer `subsection in sidebar`

Esta plantilla funciona como `section in sidebar`, sólo que para subsecciones.

Plantilla-Beamer `subsection in sidebar shaded`

Esta plantilla funciona como `section in sidebar shaded`, sólo que para subsecciones.

Plantilla-/Color-/Fuente-Beamer `subsubsection in head/foot`

Esta plantilla funciona como `section in head/foot`, sólo que para subsubsecciones. Actualmente los temas predefinidos no lo usan.

- `\insertsubsubsectionhead` funciona como `\insertsectionhead`.
- `\insertsubsubsectionheadnumber` funciona como `\insertsectionheadnumber`.

Plantilla-Beamer `subsubsection in head/foot shaded`

Esta plantilla funciona como `section in head/foot shaded`, sólo que para subsubsecciones.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` [*porcentaje*] funciona como la opción correspondiente para las secciones.

Plantilla-/Color-/Fuente-Beamer `subsubsection in sidebar`

Esta plantilla funciona como `section in sidebar`, sólo que para subsubsecciones.

Plantilla-Beamer `subsubsection in sidebar shaded`

Se comporta exactamente como `section in sidebar shaded`, sólo que para subsubsecciones.

Al presionar sobre el título del documento en una barra de navegación (no todos los temas lo muestran), se saltará a la primera diapositiva de la presentación (normalmente la página del título) *excepto* si ya se está en la primera diapositiva. En la primera diapositiva, al pulsar sobre el título del documento se saltará al final de la presentación, si lo hay. Así, se puede saltar al final con sólo pulsar *dos* veces sobre el título del documento en una barra de navegación.

Los símbolos de navegación

Los símbolos de navegación son pequeños iconos que por defecto se muestran en cada diapositiva. Se muestran los símbolos siguientes:

1. Un icono de diapositiva que se representa como un solo rectángulo. A la izquierda y derecha de este símbolo, se muestra una flecha izquierda y otra derecha respectivamente.
2. Un icono de marco que se representa como tres iconos de diapositiva “apiladas una sobre la otra”. Como antes, este símbolo está flanqueado por flechas.
3. Un icono de subsección que se representa como una entrada de subsección (resaltada) en una tabla de contenidos. Este símbolo está flanqueado por flechas.
4. Un icono de sección que se representa como una entrada de sección (resaltada junto con todas las subsecciones) en una tabla de contenidos. Este símbolo está flanqueado por flechas.
5. Un icono de presentación que se representa como una tabla de contenidos completamente resaltada.

6. Un icono de apéndice que se representa como una tabla de contenidos completamente resaltada que consiste sólo de una sección. (Este icono se muestra sólo si hay un apéndice.)
7. Iconos de hacia atrás y hacia adelante, representados como flechas circulares.
8. Un icono de “buscar” o “encontrar”, representado como la lupa de un detective.

Presionar sobre la flecha izquierda al lado de un icono siempre saltará a (la última diapositiva) la anterior diapositiva, marco, subsección o sección según sea el caso. Presionar sobre la flecha derecha al lado de un icono siempre saltará a (la primera diapositiva) la siguiente diapositiva, marco, subsección o sección según sea el caso.

Pulsar *sobre* cualquiera de estos iconos tiene un efecto diferente:

1. Si se tiene el apoyo de una aplicación visor (como xpdf o Acrobat Reader), al presionar sobre el icono de diapositiva aparece una ventana que permite acceder el número de la diapositiva a la que se desea saltar.
2. Al pulsar sobre el lado izquierdo de un icono de marco se saltará a la primera diapositiva del marco, al pulsar sobre el derecho se saltará a la última diapositiva del marco (útil para saltar cubiertas).
3. Al pulsar sobre el lado izquierdo de un icono de subsección se saltará a la primera diapositiva de la subsección, al pulsar sobre el derecho se saltará a la última diapositiva de la subsección.
4. Al pulsar sobre el lado izquierdo de un icono de sección se saltará a la primera diapositiva de la sección, al pulsar sobre el derecho se saltará a la última diapositiva de la sección.
5. Al pulsar sobre el lado izquierdo del icono de presentación se saltará a la primera diapositiva, al pulsar sobre el derecho se saltará a la última diapositiva de la presentación. Sin embargo, esto *no* incluye el apéndice.
6. Al pulsar sobre el lado izquierdo del icono de apéndice se saltará a la primera diapositiva del apéndice, al pulsar sobre el derecho se saltará a la última diapositiva del apéndice.
7. Si se tiene el apoyo de una aplicación visor, al pulsar sobre los símbolos de hacia atrás o hacia adelante se saltará a las diapositivas previamente visitadas.
8. Si se tiene el apoyo de una aplicación visor, al pulsar sobre el icono de búsqueda aparece una ventana que permite acceder la cadena a buscar. Si se encuentra, la aplicación visor saltará a esa cadena.

Se puede reducir el número de iconos que se muestran o su distribución ajustando la plantilla `navigation symbols`.

Plantilla-/Color-/Fuente-Beamer `navigation symbols`

Esta plantilla se invoca en “modo-tres-asteriscos”, por temas en el lugar donde se deben mostrar los símbolos de navegación. El “modo-tres-asteriscos” significa que se usa el comando `\usebeamertemplate***`.

Notamos que aunque parece que los símbolos son parte del pie de página, a menudo forman parte de una invisible barra lateral derecha.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` Organiza horizontalmente los símbolos de navegación.
- `[horizontal]` Es un alias para el valor por defecto.
- `[vertical]` Organiza verticalmente los símbolos de navegación.
- `[only frame symbol]` Muestra sólo el símbolo de navegación para navegar entre marcos.

Ejemplo: El comando siguiente suprime todos los símbolos de navegación:

```
\setbeamertemplate{navigation symbols}{}
```

Dentro de esta plantilla, las siguientes inserciones son útiles:

- `\insertslidenavigationssymbol` Inserta los símbolos de navegación entre diapositivas, es decir, el símbolo de diapositiva (un rectángulo) flanqueado de flechas que sirven de hiperligas.
- `\insertframenavigationssymbol` Insertar el símbolo de navegación de marco.
- `\insertsubsectionnavigationssymbol` Insertar el símbolo de navegación de subsección.
- `\insertsectionnavigationssymbol` Insertar el símbolo de navegación de sección.
- `\insertdocnavigationssymbol` Insertar el símbolo de navegación de presentación y (si es necesario) el símbolo de navegación de apéndice.
- `\insertbackfindforwardnavigationssymbol` Inserta los símbolos de hacia atrás, búsqueda y hacia adelante.

El logotipo

Para instalar un logotipo, se usa el siguiente comando:

```
\logo{<texto logotipo>}
```

El *<texto logotipo>* generalmente es un comando para incluir una gráfica, pero puede ser cualquier texto. La posición donde se inserta el logotipo se determina por el tema actual, actualmente ésta no se puede especificar directamente.

Ejemplo:

```
\pgfdeclareimage[height=0.5cm]{logotipo}{tu-logotipo}  
\logo{\pgfuseimage{logotipo}}
```

Ejemplo:

```
\logo{\includegraphics[height=0.5cm]{logotipo.pdf}}
```

Actualmente, el efecto de este comando es sólo establecer la plantilla `logo`. Sin embargo, en el futuro se podría implementar un efecto mas sofisticado.

ARTÍCULO Este comando no tiene efecto.

Plantilla-/Color-/Fuente-Beamer logo

Esta plantilla se usa para hacer el logotipo.

El siguiente comando se puede usar para insertar un logotipo en alguna parte:

- `\insertlogo` Inserta el logotipo en la posición actual. Tiene el mismo efecto que `\usebeamertemplate*{logo}`.

El título del marco

El título del marco se muestra de manera prominente en la parte superior del marco y se especifica con el siguiente comando:

```
\frametitle<<especificación de cubierta>>[<Título Corto del Marco>]{<Texto del Título de Marco>}
```

Si el título es una frase propia se debe finalizar el *<Texto del Título de Marco>* con un punto. En caso contrario, no debe haber un punto final. El *<Título Corto del Marco>* normalmente no se muestra, pero

está disponible vía el comando `\insertshortframetitle`. La *especificación de cubierta* es principalmente útil para suprimir el título del marco en el modo `article`.

Ejemplo:

```
\begin{frame}
  \frametitle{Un Título de Marco es Importante.}
  \framesubtitle{Los subtítulos no lo son tanto.}
  <Contenido del marco.>
\end{frame}
```

Si en el marco actual se está usando la opción `\allowframebreaks`, se agrega automáticamente un texto de continuación (como “(cont.)” o algo similar dependiendo de la plantilla `frametitle continuation`) al final del *Texto del Título de Marco*, separados por un espacio.

PRESENCIA Cuando se encuentra el comando `\frametitle`, el título del marco no se edita inmediatamente. Más bien, su argumento se almacena internamente y el título del marco sólo se edita cuando se ha leído el marco completo. Esto da acceso al *Texto del Título de Marco* y al *Texto del Título del Submarco* que se introduce usando el comando `\framesubtitle`.

ARTÍCULO En este modo, este comando por defecto crea un párrafo nuevo titulado *Texto del Título de Marco*. El usar la *especificación de cubierta* facilita de vez en cuando la eliminación del título del marco. Si se desea suprimir de manera general *todos* los títulos de marco en este modo, poner `\setbeamertemplate<article>{frametitle}{}`.

Plantilla-/Color-/Fuente-Beamer `frametitle`

Color/fuente padres: `titlelike`

Cuando se va a editar el título y subtítulo del marco, esta plantilla se invoca con el color- y fuente-BEAMER `frametitle`. Esta plantilla *no* se invoca cuando se llaman los comandos `\frametitle` o `\framesubtitle`. Más bien, se invoca cuando se ha leído completamente el marco entero. Hasta entonces, los textos del título y subtítulo de marco se almacenan en un lugar especial. De esta manera, ambas inserciones se establecen correctamente cuando se invoca la plantilla. Entonces la caja- \TeX resultante se vuelve a poner mágicamente en la parte superior del marco.

Las siguientes opciones de plantilla están predefinidas:

- **[default]** [*alineación*] El marco se edita usando el color-BEAMER `frametitle` y la fuente-BEAMER `frametitle`. El subtítulo se pone debajo usando el color y fuente `framesubtitle`.

Si el color `frametitle` tiene un fondo, detrás del título se pone una barra de fondo cubriendo el ancho del marco entero. Se ignora el color de fondo del subtítulo. La *alineación* se pasa al ambiente `beamercolorbox`. En particular, existen opciones útiles como `left`, `center` y `right`. Como caso especial, la opción `right` causa que el borde izquierdo del título del marco sea algo más grande de lo normal, para que el título del marco esté más al medio.

- **[shadow theme]** Esta opción está disponible si se carga el tema exterior `shadow`. Dibuja el título del marco en la parte superior de un sombreado horizontal entre los colores de fondo `frametitle` y `frametitle right`. Si existe, se pone un subtítulo en esta barra. Debajo de la barra se dibuja una “sombra”.
- **[sidebar theme]** Esta opción está disponible si se carga el tema exterior `sidebar` y si la altura del encabezado no se pone a 0pt (que se puede hacer usando una opción del tema `sidebar`).

Con esta opción, el título del marco se pone dentro de una área rectangular que es parte del encabezado (se usa algún “espacio negativo” para subir el título del marco a esta área). No se usa el fondo del color `frametitle`, en este caso, es trabajo de la plantilla `headline`.

- [`smoothbars theme`] Esta opción está disponible si se carga el tema exterior `smoothbars`. Edita el título del marco sobre una barra coloreada con el color de fondo de `frametitle`. La parte superior e inferior de la barra se mezclan suavemente sobre los fondos de arriba y debajo.
- [`smoothtree theme`] Como el tema de `smoothbars`, sólo que para el tema `smoothtree`.

Los siguientes comandos son útiles para esta plantilla:

- `\insertframetitle` proporciona el título del marco.
- `\insertframesubtitle` proporciona el subtítulo del marco.

`\framesubtitle`<<especificación de cubierta>{(texto subtítulo del marco)}

Si existe, se mostrará debajo del título principal un subtítulo con una fuente más pequeña. Como el comando `\frametitle`, este comando se puede dar en cualquier parte del marco, dado que el título del marco realmente se edita sólo cuando todo lo demás ya ha sido editado.

Ejemplo:

```
\begin{frame}
  \frametitle<presentation>{El Título del Marco Tiene que Estar en Mayúsculas.}
  \framesubtitle{Los subtítulos pueden estar en minúsculas si son oraciones enteras.}
  <Contenido del marco>
\end{frame}
```

ARTÍCULO En este modo, por defecto, no se muestra el subtítulo.

Color/-Fuente Beamer `framesubtitle`

Color/fuente padre: `frametitle`

Este elemento proporciona un color y una fuente para el subtítulo, pero sin plantilla. Es trabajo de la plantilla `frametitle` editar el subtítulo.

Por defecto, todo el material de una diapositiva se centra verticalmente. Esto se puede cambiar usando la siguiente clase de opciones:

```
\documentclass[t]{beamer}
```

Coloca el texto de las diapositivas en la parte superior (vertical) de las diapositivas, corresponde a un “jalón” vertical. Para anularlo se puede usar la opción `c` (center) o `b` (below) en los marcos individuales.

```
\documentclass[c]{beamer}
```

Coloca el texto de las diapositivas en el centro (vertical) de las diapositivas. Éste es el valor por defecto, para anularlo se puede usar la opción `t` (top) o `b` (below) en los marcos individuales.

El fondo

Cada marco tiene un *fondo*, que –como sugiere el nombre– esta “detrás de todo”. El fondo es un objeto sorprendentemente complejo: en BEAMER, consiste de un *lienzo de fondo* y el *fondo principal*. El lienzo de fondo se puede imaginar como una área grande sobre la que se pinta todo (el fondo principal y todo lo demás). Por defecto, este lienzo es un rectángulo grande que llena el marco entero cuyo color es el fondo del color-BEAMER `background canvas`. Dado que este color se hereda de `normal text`, se puede cambiar el color del lienzo cambiando el color de fondo del texto normal.

Ejemplo: El siguiente comando cambia el color del fondo a un rojo luminoso.

```
\setbeamercolor{normal text}{bg=red!20}
```

El lienzo no necesita ser monocromático. En cambio, se puede instalar un sombreado o incluso se puede hacer transparente. Hacerlo transparente es una buena idea si se desea incluir las diapositivas en

algún otro documento.

Ejemplo: El siguiente comando hace transparente el fondo del lienzo:

```
\setbeamercolor{background canvas}{bg=}
```

Plantilla-/Color-/Fuente-Beamer `background canvas`

Color padre: `normal text`

Esta plantilla se inserta “detrás de todo”. Típicamente debe tener algunos comandos T_EX que producen un rectángulo de altura `\paperheight` y ancho `\paperwidth`.

Las siguientes opciones de plantilla están predefinidas:

- **[default]** Instala un rectángulo grande con el color del fondo. Si el fondo está vacío, el lienzo es “transparente”. Dado que `background canvas` se hereda de `normal text`, se puede cambiar el fondo del color-BEAMER `normal text` para cambiar el color predefinido del lienzo. Sin embargo, para hacer el lienzo transparente, sólo se deja vacío el fondo del lienzo lo que deja el fondo del texto `normal` en blanco.
- **[vertical shading]** [*opciones de color*] Instala un fondo sombreado verticalmente. *Hay que tener cuidado: ¡A menudo los sombreados del fondo distraen!*. Se pueden dar las siguientes *opciones de color*:
 - `top=<color>` especifica el color en la parte superior de la página. Por defecto, se usa el 25 % del primer plano del color-BEAMER `palette primary`.
 - `bottom=<color>` especifica el color en la parte inferior de la página. Por defecto, al momento de la invocación de este comando se usa el fondo de `normal text`.
 - `middle=<color>` especifica el color para la parte media de la página. Así, si se da esta opción, el sombreado cambia del color de la parte inferior a este color y luego al color de la parte superior.
 - `midpoint=<factor>` especifica en que punto de la página se va a usar el color de enmedio. Un factor de 0 es la parte inferior de la página, un factor de 1 es la superior. El valor por defecto es 0.5 que es por la mitad.

El fondo principal se dibuja en la parte superior del lienzo de fondo. Se puede usar para agregar, digamos, una reja a cada marco o una imagen grande de fondo o cualquier otra cosa.

Plantilla-/Color-/Fuente-Beamer `background`

Color padre: `background canvas`

Esta plantilla se inserta “detrás de todo, pero encima del lienzo de fondo”. Se puede usar para poner imágenes, rejas o cualquier otra cosa que no necesariamente llena todo el fondo. Cuando se edita esta plantilla, se establece el color- y fuente-BEAMER `background`.

Las siguientes opciones de la plantilla están predefinidas:

- **[default]** Está vacía.
- **[grid]** [*opciones de reja*] coloca una reja en el fondo. Se tienen las siguientes *opciones de reja*:
 - `step=<dimensión>` especifica la distancia entre las líneas de la reja. El valor por defecto es 0.5 cm.
 - `color=<color>` especifica el color de las líneas de la reja. El valor por defecto es 10 % del primer plano.

7.3. Medidas de los márgenes

El “tamaño del papel” en una presentación BEAMER se fija en 128 mm por 96 mm. El aspecto de la proporción de este tamaño es de 4:3, que es exactamente lo que la mayoría de beamers ofrecen en estos días. Es trabajo del programa de presentación (como `acroread`, `xpdf`, `Acrobat Reader`) desplegar las diapositivas en toda la pantalla. La ventaja principal de usar un “tamaño de papel” pequeño es que se pueden usar todas las fuentes normales en su tamaño natural. En particular, insertar un gráfico con etiquetas de 11pt resultará en etiquetas de medidas razonables durante la presentación.

Nos debemos abstener de cambiar el “tamaño del papel”. Sin embargo, se puede cambiar el tamaño de los márgenes izquierdo y derecho, que tienen un valor predefinido de 1 cm. Para hacerlo se debe usar el siguiente comando:

```
\setbeamersize{<opciones>}
```

Se pueden dar las siguientes *<opciones>*:

- `text margin left=<dimensión TEX>` fija un nuevo margen izquierdo. Esto excluye la barra lateral izquierda. Así, es la distancia entre la orilla derecha de la barra lateral izquierda y la orilla izquierda del texto.
- `text margin right=<dimensión TEX>` fija un nuevo margen derecho.
- `sidebar width left=<dimensión TEX>` fija el tamaño de la barra lateral izquierda. Actualmente, se debe dar este comando *antes* de instalar un sombreado para el lienzo de la barra lateral.
- `sidebar width right=<dimensión TEX>` fija el tamaño de la barra lateral derecha.
- `description width=<dimensión TEX>` fija el ancho predefinido de las etiquetas de descripción.
- `description width of=<texto>` fija el ancho predefinido de las etiquetas de descripción al ancho del *<texto>*.
- `mini frame size=<dimensión TEX>` fija el tamaño de los minimarcos en una barra de navegación. Cuando se muestran dos iconos de mini marcos uno junto al otro, sus puntos extremos izquierdos están separados por *<dimensión T_EX>*.
- `mini frame offset=<dimensión TEX>` fija un desplazamiento vertical adicional que se agrega al tamaño del mini marco al colocar mini marcos verticalmente.

ARTÍCULO Este comando no tiene efecto en este modo.

7.4. Restringiendo las diapositivas de un marco

El número de diapositivas en un marco se calcula automáticamente. Si el número más grande mencionado en cualquier especificación de cubierta dentro del marco es 4, sólo se introducen cuatro diapositivas (a pesar del hecho que una especificación como `<4->` podría sugerir que son posibles más de cuatro diapositivas). También se puede especificar “a mano” el número de diapositivas en el marco. Para hacerlo, se pasa una especificación de cubierta con el comando `\frame`. El marco contendrá sólo las diapositivas especificadas en su argumento. Consideremos el ejemplo siguiente:

```
\begin{frame}<1-2,4->
  \’{E}sta es la diapositiva n\’{u}mero \only<1>{1}\only<2>{2}\only<3>{3}%
  \only<4>{4}
\end{frame}
```

Este comando crea un marco que contiene cuatro diapositivas. La primera contiene el texto “Ésta es la diapositiva número 1”, la segunda “Ésta es la diapositiva número 2”, la tercera “Ésta es la diapositiva número 3” y la cuarta “Ésta es la diapositiva número 4”.

Una especificación útil es `<0>` que causa que el marco no tenga ninguna diapositiva. Por ejemplo, el comando `\begin{frame}<handout:0>` causa que se suprima el marco en la versión *handout*, pero se muestra de manera normal en todas las otras versiones. Otra especificación útil es `<beamer>` que causa que se muestre el marco de manera normal en el modo *beamer*, pero se suprime en todas las otras versiones.

8. Creando Cubiertas

8.1. El comando `pause`

El comando `pause` ofrece una manera fácil, pero no muy flexible de crear marcos que se descubren por partes. Si se pone `\pause` en alguna parte de un marco, en la primera diapositiva sólo se muestra el texto en el marco hasta el primer comando `\pause`. En la segunda diapositiva, se presenta todo hasta el segundo `\pause`, y así sucesivamente. Es decir, el efecto de `\pause` dura hasta el próximo `\pause`, `\onslide`, o el fin del marco. También se puede usar `\pause` dentro de ambientes, su efecto persiste más allá del ambiente. Sin embargo, abusar del comando `\pause` dentro de un ambiente anidado puede no producir el resultado deseado.

Al usar especificaciones de cubierta se puede lograr un mayor control sobre lo que se muestra en cada diapositiva. Sin embargo, para muchos casos simples el comando `\pause` es suficiente:

```
\begin{frame}
  \begin{itemize}
    \item Se muestra desde la primera diapositiva.
  \pause
    \item Se muestra desde la segunda diapositiva.
    \begin{itemize}
      \item Se muestra desde la segunda diapositiva.
    \pause
      \item Se muestra desde la tercera diapositiva.
    \end{itemize}
    \item Se muestra desde la tercera diapositiva.
  \pause
    \item Se muestra desde la cuarta diapositiva.
  \end{itemize}
  Se muestra desde la cuarta diapositiva.
  \begin{itemize}
    \onslide
    \item Se muestra desde la primera diapositiva.
  \pause
    \item Se muestra desde la quinta diapositiva.
  \end{itemize}
\end{frame}
```

`\pause[⟨número⟩]`

Este comando causa que sólo se muestre el texto que lo sigue a partir de la siguiente diapositiva en adelante, o, si se da la opción `⟨número⟩`, desde la diapositiva con número `⟨número⟩`. Si se da la opción `⟨número⟩`, entonces se asigna este número al contador `beamerpauses`. Este comando usa internamente el comando `\onslide`. Este comando *no* funciona dentro de los ambientes de `amsmath`, como por ejemplo en `align`, dado que hacen cosas muy malas.

Ejemplo:

```
\begin{frame}
  \begin{itemize}
    \item
      A
    \pause
    \item
      B
    \pause
    \item
```

```

C
\end{itemize}
\end{frame}

```

ARTÍCULO Se ignora este comando.

Para “desenpausar” algún texto, es decir, para suspender temporalmente el efecto de pause, se usa el comando `\onslide`.

8.2. Concepto general de especificación de cubierta

El método seguido por la mayoría de las clases para la presentación de diapositivas es algo similar al comando `\pause`. Estos comandos obtienen como entrada un cierto número de diapositiva y afectan de cierta manera el texto que lo sigue. Por ejemplo, el comando de Prosper `\FromSlide{2}` causa que todo el texto que sigue después de este comando sólo se muestre de la segunda diapositiva en adelante.

La clase BEAMER usa un método diferente (aunque el comando `\onslide<2->` tiene el mismo efecto que `\FromSlide{2}`, sólo que `\onslide` trasciende ambientes; del mismo modo, a `\pause` le corresponde internamente un comando con especificaciones de cubierta apropiadas). La idea es agregar *especificaciones de cubierta* a ciertos comandos. Estas especificaciones siempre se dan entre paréntesis angulares y siguen al comando “lo más pronto posible”, aunque en ciertos casos BEAMER también permite que se den un poco después las especificaciones de cubierta. En el caso más simple, la especificación contiene sólo un número. Un comando seguido de una especificación de cubierta, sólo tendrá “efecto” en las diapositivas mencionadas en la especificación. Lo que significa exactamente “tener efecto” depende del comando. Consideremos el siguiente ejemplo.

```

\begin{frame}
\textbf{Esta l\'}{i}nea est\'}{a} en negritas en las tres diapositivas.}
\textbf<2>{Esta l\'}{i}nea est\'}{a} en negritas s\'}{o}lo en la segunda diapositiva.}
\textbf<3>{Esta l\'}{i}nea est\'}{a} en negritas s\'}{o}lo en la tercera diapositiva.}
\end{frame}

```

Para el comando `\textbf`, la especificación de cubierta causa que el texto se ponga en negritas sólo en las diapositivas especificadas. En todas las demás, el texto se pone en la fuente normal. Como segundo ejemplo, consideremos el siguiente marco:

```

\begin{frame}
\only<1>{Esta l\'}{i}nea se inserta s\'}{o}lo en la diapositiva 1.}
\only<2>{Esta l\'}{i}nea se inserta s\'}{o}lo en la diapositiva 2.}
\end{frame}

```

El comando `\only`, introducido por BEAMER, normalmente sólo inserta su parámetro en el marco actual. Sin embargo, si está presente una especificación cubierta, “desecha” su parámetro en las diapositivas que no se mencionan.

Se pueden escribir especificaciones de cubierta sólo detrás de ciertos comandos, no en todo comando. En la próxima sección se explica qué comandos se pueden usar y los efectos que tendrán. Sin embargo, es bastante fácil redefinir un comando existente de tal manera que se vuelva consciente de “especificación de cubierta”.

La sintaxis de las especificaciones de cubierta (básicas) es la siguiente: son listas separadas por comas o rangos de diapositivas. Los rangos se especifican así: 2–5 que significa: de la diapositiva dos hasta la cinco. Se puede omitir el principio o final de un rango: por ejemplo 3– significa “diapositivas tres, cuatro,

cinco, y así sucesivamente” y -5 significa lo mismo que 1-5. Un ejemplo complicado es -3,6-8,10,12-15 que selecciona las diapositivas 1 2, 3, 6, 7, 8, 10, 12,13,14 y 15.

8.3. Comandos con especificación de cubierta

En los siguientes comandos, agregar una especificación de cubierta causa simplemente que se ignore el comando en las diapositivas que no están incluidas en la especificación: `\textbf`, `\textit`, `\textsl`, `\textrm`, `\textsf`, `\color`, `\alert`, `\structure`. Si un comando toma varios argumentos, como `\color`, la especificación debe seguir directamente al comando como en el siguiente ejemplo (pero hay excepciones a esta regla):

```
\begin{frame}
\color<2-3>[rgb]{1,0,0} Este texto es rojo en las diapositivas 2 y 3, en las dem\'}{a}s
es negro.
\end{frame}
```

Para los siguientes comandos, el efecto de una especificación de cubierta es especial:

```
\onslide<modificador><especificación de cubierta>{<texto>}
```

El comportamiento de este comando depende de si se da o no el argumento opcional `<texto>` (notamos que se da el argumento opcional `<texto>` entre llaves normales, no entre corchetes). Si está presente, el `<modificador>` puede ser un `+` o un `*`.

Si no se da `<texto>`, pasa lo siguiente: todo el texto que sigue después de este comando sólo se mostrará (descubrirá) en las diapositivas especificadas. En las diapositivas no especificadas, el texto todavía ocupa espacio. Si no se especifica ninguna diapositiva, siempre se muestra el texto que sigue. No se necesita llamar a este comando en el mismo grupo T_EX su efecto trasciende grupos de bloque. Sin embargo, este comando tiene un efecto *diferente* dentro de un ambiente `overprint`.

Si el `<modificador>` es `+`, el texto oculto no será tratado como cubierto sino como invisible. La diferencia es la misma que la que hay entre `\uncover` y `\visible`. Si ningún argumento `<texto>` está presente no se puede dar el modificador `*`.

Ejemplo:

```
\begin{frame}
  Se muestra en la primera diapositiva.
  \onslide<2-3> Se muestra s\'}{o}lo en la segunda y tercera diapositiva.
  \begin{itemize}
  \item Todav\'}{i}a se muestra en la segunda y tercera diapositiva.
  \onslide+<4->
  \item Se muestra desde la diapositiva 4 en adelante.
  \end{itemize}
  Se muestra desde la diapositiva 4 en adelante.
  \onslide Se muestra en todas las diapositivas.
\end{frame}
```

Si está presente el argumento `<texto>`, `\onslide` (sin `<modificador>`) se vuelve `\uncover`, `\onslide+` se vuelve `\visible` y `\onslide*` se vuelve `\only`.

Ejemplo:

```
\begin{frame}
  \onslide<1>{Mismo efecto que el siguiente comando.}
  \uncover<1>{Mismo efecto que el comando previo.}
  \onslide+<2>{Mismo efecto que el siguiente comando.}
  \visible<2>{Mismo efecto que el comando previo.}
```

```

\onslide*<3>{Mismo efecto que el siguiente comando.}
\only<3>{Mismo efecto que el comando previo.}
\end{frame}

```

`\only`<*especificación de cubierta*>{<texto>}<*especificación de cubierta*>

Si está presente cualquier *especificación de cubierta* (aunque sólo puede haber una), el *texto* se inserta sólo en las diapositivas especificadas. Para las otras diapositivas, el texto simplemente se “desecha”. En particular, no ocupa espacio.

Ejemplo: `\only<3->{Texto que se inserta de la diapositiva 3 en adelante.}`

Dado que también se puede dar la especificación de cubierta después del texto, a menudo se usa `\only` para hacer de una manera simple que otros comandos sean conscientes-de-especificación-de-cubierta:

Ejemplo:

```

\newcommand{\miazul}{\only{\color{blue}}}
\begin{frame}
\miazul<2> Este texto es azul s\’o}lo en la diapositiva 2.
\end{frame}

```

`\uncover`<*especificación de cubierta*>{<texto>}

Si está presente la *especificación de cubierta*, el *texto* se muestra (“descubre”) sólo en las diapositivas especificadas. En las otras diapositivas, el texto todavía ocupa espacio y todavía se edita, pero no se muestra o sólo se muestra como si fuera transparente.

Ejemplo: `\uncover<3->{Este texto se muestra de la diapositiva 3 en adelante.}`

ARTÍCULO Este comando tiene el mismo efecto que `\only`.

`\visible`<*especificación de cubierta*>{<texto>}

Este comando hace casi lo mismo que `\uncover`. La única diferencia es que si el texto no se muestra, nunca se muestra de una manera transparente, sino que no se muestra en absoluto. Así que para este comando los ajustes para transparencias no tienen efecto.

Ejemplo: `\visible<2->{El texto se muestra desde la diapositiva 2 en adelante.}`

ARTÍCULO Este comando tiene el mismo efecto que `\only`.

`\invisible`<*especificación de cubierta*>{<texto>}

Este comando es lo opuesto de `\visible`.

Ejemplo: `\invisible<-2>{El texto se muestra de la diapositiva 3 en adelante.}`

`\alt`<*especificación de cubierta*>{<texto predefinido>}{<texto alternativo>}<*especificación de cubierta*>

Sólo se puede dar una *especificación de cubierta*. Se muestra el texto predefinido en las diapositivas especificadas, en cualquier otro caso, el texto alternativo. Siempre debe estar presente la especificación.

Ejemplo: `\alt<2>{En la diapositiva 2}{No en la diapositiva 2.}`

Una vez más, es útil dar la especificación de cubierta al final cuando se usa dentro de otros comandos.

Ejemplo: Aquí está la definición de `\uncover`:

```

\newcommand{\uncover}{\alt{\@firstofone}{\makeinvisible}}

```

`\temporal`<especificación de cubierta>{(texto diapositivas previas)}{(texto predefinido)}
{(texto diap. posteriores)}

Este comando alterna entre tres textos diferentes, dependiendo de si la diapositiva actual está antes que las diapositivas especificadas, es una de las diapositivas especificadas, o viene después de ellas. Si la <especificación de cubierta> no es un intervalo (es decir, si tiene un “agujero”), el “agujero” se considera como parte previa a las diapositivas.

Ejemplo:

```
\temporal<3-4>{Se muestra en 1, 2}{Se muestra en 3, 4}{Se muestra en 5, 6, 7, ...}
\temporal<3,5>{Se muestra en 1, 2, 4}{Se muestra en 3, 5}{Se muestra en 6, 7, 8,...}
```

Como una posible aplicación del comando `\temporal` considerar el siguiente ejemplo:

```
\def\micoloreado<#1>{%
\temporal<#1>{\color{red!50}}{\color{black}}{\color{black!50}}}
\begin{frame}
\begin{itemize}
\micoloreado<1> \item Primer objeto.
\micoloreado<2> \item Segundo objeto.
\micoloreado<3> \item Tercer objeto.
\micoloreado<4> \item Cuarto objeto.
\end{itemize}
\end{frame}
```

`\item`<especificación de alerta>[<etiqueta del objeto>]<especificación de alerta>

Sólo se puede dar una <especificación de alerta>.

PRESEN-
TACIÓN

Ejemplo:

```
\begin{frame}
\begin{itemize}
\item<1-> Primer punto, se muestra en todas las diapositivas.
\item<2-> Segundo punto, se muestra de la diapositiva 2 en adelante.
\item<2-> Tercer punto, tambi\ 'e)n se muestra de la diapositiva 2 en adelante.
\item<3-> Cuarto punto, se muestra en la diapositiva 3.
\end{itemize}
\end{frame}
```

```
\begin{frame}
\begin{enumerate}
\item<3-| alert@3>[0.] Se muestra un cero-\ 'e)simo punto al final.
\item<1-| alert@1> El primero y principal punto.
\item<2-| alert@2> El segundo punto.
\end{enumerate}
\end{frame}
```

ARTÍCULO Actualmente se ignora completamente la <especificación de acción>.

El comando relacionado `\bibitem` también tiene conciencia de especificación de cubierta de la misma manera que `\item`.

`\label`<especificación de cubierta>{(nombre de la etiqueta)}

Si está presente la <especificación de cubierta>, la etiqueta sólo se inserta en la diapositiva especificada. Insertar una etiqueta en más de una diapositiva causará una advertencia de “etiquetado múltiple”. Sin embargo, si no existe especificación de cubierta, la especificación se pone automáticamente en “1” y así la etiqueta sólo se inserta en la primera diapositiva. Éste es típicamente el comportamiento deseado dado

que realmente no importa en que diapositiva se inserta la etiqueta, *excepto* si se usa un comando `\only` y *excepto* si se desea usar la etiqueta como objetivo de un hipersalto. Entonces se necesita especificar una diapositiva.

Se pueden usar etiquetas como objetivos de hipersaltos. Una manera conveniente de etiquetar un marco es usar la opción `label=<nombre>` del ambiente `frame...` Sin embargo, esto causará que el marco entero se mantenga en la memoria hasta el final de la compilación, lo que plantea un problema.

Ejemplo:

```
\begin{frame}
  \begin{align}
    a &= b + c \label{primero} \\ % no se necesita especificación
    c &= d + e \label{segundo} \\ % no se necesita especificación
  \end{align}
  Bla bla, \uncover<2>{más bla bla.}
  \only<3>{Ahora se necesita una especificaci\'}{o}n.\label<3>{miqueta}}
\end{frame}
```

8.4. Ambientes con especificación de cubierta

También se puede equipar a los ambientes con especificaciones de cubierta. Para la mayoría de los ambientes predefinidos, agregar una especificación de cubierta causa que el ambiente entero sólo se descubra en las diapositivas especificadas. Esto es útil para mostrar cosas sucesivamente, como en el siguiente ejemplo:

```
\begin{frame}
  \frametitle{Un Teorema sobre Conjuntos Infinitos}

  \begin{theorem}<1-> Existe un conjunto infinito.
  \end{theorem}

  \begin{proof}<3-> Esto se sigue del axioma del infinito.
  \end{proof}

  \begin{example}<2-> El conjunto de  $n$ '{u}meros naturales es infinito.
  \end{example}
\end{frame}
```

Así, la primera diapositiva contiene sólo el teorema, la segunda agrega un ejemplo y en la tercera se muestra la prueba.

Para cada uno de los comandos básicos `\only`, `\alt`, `\visible`, `\uncover` y `\invisible` existe una “versión de ambiente” `onlyenv`, `altenv`, `visibleenv`, `uncoverenv` y `invisibleenv`. Salvo `altenv` y `onlyenv`, estos ambientes hacen lo mismo que sus comandos respectivos.

```
\begin{onlyenv}<especificación de cubierta>
<Contenido del ambiente>
\end{onlyenv}
```

Si se da la *<especificación de cubierta>*, el contenido del ambiente se inserta en el texto sólo en las diapositivas especificadas. La diferencia con `\only` es que el texto en realidad se edita dentro de una caja que luego se desecha, mientras que `\only` desecha inmediatamente su contenido. Si el texto no es “editable”, `onlyenv` puede producir un error donde `\only` no lo haría.

Ejemplo:

```

\begin{frame}
  Esta l\,'{i}nea siempre se muestra.
  \begin{onlyenv}<2>
    Esta l\,'{i}nea se inserta en la diapositiva 2.
  \end{onlyenv}
\end{frame}

```

```

\begin{altenv}<<especificación de cubierta>>{\langle texto inicial \rangle}{\langle texto final \rangle}{\langle texto inicial alternativo \rangle}
{\langle texto final alternativo \rangle}<<especificación de cubierta>>
{\langle Contenido del ambiente \rangle}
\end{altenv}

```

Sólo se puede dar una *especificación de cubierta*. En las diapositivas especificadas, se inserta el *texto inicial* al principio del ambiente y el *texto final* al final. En todas las otras diapositivas, se usará el *texto inicial alternativo* y el *texto final alternativo*.

Ejemplo:

```

\begin{frame}
  Esta
  \begin{altenv}<2>{\langle \rangle}{\langle [] \rangle}
    palabra
  \end{altenv}
  est\,'{a} entre par\,'{e}ntesis en la diapositiva 2 y entre corchetes en la primera.
\end{frame}

```

8.5. Cambio dinámico de texto o imágenes

Es posible que algunas veces se quiera que alguna parte de un marco cambie dinámicamente de diapositiva en diapositiva. En cada diapositiva del marco, se debe mostrar algo diferente dentro de esta área. Se puede lograr el efecto de cambiar de manera dinámica el texto dando una lista de comandos `\only` como sigue:

```

\only<1>{\langle Texto inicial. \rangle}
\only<2>{\langle Reemplazado por esto en la segunda diapositiva. \rangle}
\only<3>{\langle Reemplazado por esto en la tercera diapositiva. \rangle}

```

El problema con este método es que puede conducir a pequeñas, pero molestas diferencias en las alturas de las líneas, que pueden causar que el marco entero “tiemble” de diapositiva en diapositiva. Este problema se vuelve mucho más severo si el texto de reemplazo tiene varias líneas de largo. Para resolver este problema, se pueden usar dos ambientes: `overlayarea` y `overprint`. El primero es más flexible pero menos amigable para el usuario.

```

\begin{overlayarea}{\langle ancho del área \rangle}{\langle altura del área \rangle}
{\langle contenido del ambiente \rangle}
\end{overlayarea}

```

Todo dentro del ambiente se pondrá en un área rectangular del tamaño especificado. El área tendrá el mismo tamaño en todas las diapositivas de un marco, sin tener en cuenta su contenido real.

Ejemplo:

```

\begin{overlayarea}{\textwidth}{3cm}
  \only<1>{\langle Algo de texto para la primera diapositiva. \\\Posiblemente varias l\,'{i}neas
de largo. \rangle}
  \only<2>{\langle Reemplazo de la segunda diapositiva. \rangle}

```

```

\end{overlayarea}

\begin{overprint}[\langle ancho del área \rangle]
\langle contenido del ambiente \rangle
\end{overprint}

```

El $\langle ancho del área \rangle$ predefine el ancho del texto. Dentro del ambiente, se usan comandos `\onslide` para especificar cosas distintas que se deben mostrar en diferentes diapositivas para este ambiente. Los comandos `\onslide` se usan como los comandos `\item`. Todo dentro del ambiente se pondrá en un área rectangular del ancho especificado. La altura y profundidad del área se eligen suficientemente grandes para acomodar el contenido más grande del área. Las especificaciones de cubierta de los comandos `\onslide` deben ser conjuntos disjuntos.

Éste puede ser un problema para los folletos, dado que, allí, todas las especificaciones de cubierta tienen como valor predefinido 1. Si se usa la opción *handout*, se pueden desactivar todos menos un `\onslide` poniendo los demás en 0.

Ejemplo:

```

\begin{overprint}
  \onslide<1|handout:1>
    Algo de texto para la primera diapositiva.\\
    Posiblemente varias líneas de largo.
  \onslide<2|handout:0>
    Reemplazo de la segunda diapositiva. Suprimido para la opción \o}nhandout.
\end{overprint}

```

Una necesidad similar de cambios dinámicos surge cuando se tiene, digamos, una serie de imágenes llamadas `primero.pdf`, `segundo.pdf`, y `tercero.pdf` que muestran diferentes fases de algún proceso. Para hacer un marco que muestre estas imágenes en diferentes diapositivas, se podría usar el siguiente código:

```

\begin{frame}
  \frametitle{Las Tres Fases del Proceso}
  \includegraphics<1>{primero.pdf}
  \includegraphics<2>{segundo.pdf}
  \includegraphics<3>{tercero.pdf}
\end{frame}

```

El código anterior usa el hecho de que BEAMER hace al comando `\includegraphics` consciente de especificación de cubiertas. Funciona muy bien, pero sólo si cada archivo `.pdf` contiene el gráfico completo a mostrar. Sin embargo, algunos programas, como `xfig`, a veces producen series de gráficos en los que cada archivo contiene solamente los elementos gráficos *adicionales* a mostrarse en la próxima diapositiva. En este caso, el primer gráfico no se debe mostrar sólo en la diapositiva 1, sino de la diapositiva 1 en adelante, y así sucesivamente. Mientras que esto se logra fácilmente cambiando la especificación de cubierta `<1>` a `<1->`, los gráficos también se deben mostrar *uno encima del otro*. Una manera de lograr esto es usar el comando `\llap` de T_EX así:

```

\begin{frame}
  \frametitle{Las Tres Fases del Proceso}
  \includegraphics<1->{primero.pdf}%
  \llap{\includegraphics<2->{segundo.pdf}}%
  \llap{\includegraphics<3->{tercero.pdf}}
\end{frame}

```

O así:


```

\begin{frame}
  \frametitle{Las Tres Fases del Proceso}
  \includegraphics{primero.pdf} %
  \pause %
  \llap{\includegraphics{segundo.pdf}} %
  \pause %
  \llap{\includegraphics{tercero.pdf}}
\end{frame}

```

Una manera más conveniente es usar el comando `\multiinclude`.

8.6. Especificaciones de cubierta avanzadas

Fabricación de comandos y ambientes conscientes de especificación de cubierta

Esta sección explica cómo definir nuevos comandos que son conscientes de especificaciones de cubierta. También, explica cómo establecer correctamente los contadores que se deben incrementar de marco en marco (como la numeración de ecuaciones), pero no de diapositiva en diapositiva.

BEAMER extiende la sintaxis del comando estándar de L^AT_EX `\newcommand`:

```

\newcommand<>{<nombre del comando>}[<número de argumentos>][<valor opcional predefinido>]{<texto>}

```

Declara el nuevo comando llamado `<nombre del comando>`. El `<texto>` debe contener el cuerpo de este comando y puede contener apariciones de parámetros como `#<número>`. Aquí `<número>` puede estar entre 1 y `<número de argumentos>+1`. El argumento adicional permitido es la especificación de cubierta.

Cuando se usa `<nombre del comando>`, éste examina tantos argumentos como `<número de argumentos>`. Mientras los examina busca una especificación de cubierta, la cual se puede dar entre cualesquiera dos argumentos, antes del primer argumento o después del último argumento. Si encuentra una especificación de cubierta como `<3>`, llamará al `<texto>` con los argumentos del 1 al `<número de argumentos>` puestos como argumentos normales y el argumento número `<número de argumentos>+1` se pone como `<3>` (incluyendo los paréntesis angulares). Si no se encuentra una especificación de cubierta, el argumento extra es vacío. Si se proporciona un `<valor opcional predefinido>`, el primer argumento de `<nombre del comando>` es opcional. Si no se especifica un argumento opcional entre corchetes, se usa el `<valor opcional predefinido>`.

Ejemplo: El siguiente comando edita su argumento en rojo en las diapositivas especificadas:

```

\newcommand<>{\hacerrojo}[1]{\color#2{red}#1}

```

Ejemplo: Aquí esta la definición del comando `\emph` de BEAMER:

```

\newcommand<>{\emph}[1]{\only#2{\itshape}#1}

```

Ejemplo: Esta es la definición del comando `\transdissolve` de BEAMER (el comando `\beamer@dotrans` principalmente pasa su argumento a `hyperref`):

```

\newcommand<>{\transdissolve}[1][\only#2{\beamer@dotrans[#1]{Dissolve}}]

```

```

\renewcommand<>{<nombre de comando existente>}[<número de argumentos>][<valor opcional predefinido>]{<texto>}

```

Redeclara un comando que ya existe de la misma manera que `\newcommand<>`. Dentro del `<texto>`, todavía se puede acceder a definiciones originales usando el comando `\beameroriginal`, por ejemplo, en BEAMER se usa el siguiente comando para hacer consciente de especificación de cubierta a `\hyperlink`:

```

\renewcommand<>{\hyperlink}[2]{\only#3{\beameroriginal{\hyperlink}{#1}{#2}}}

```

```

\newenvironment<>{<nombre del ambiente>}[<número de argumentos>][<valor opcional predefinido>]{<texto inicial >}{<texto final>}

```

Declara un nuevo ambiente que es consciente de especificación de cubierta. Si se encuentra este ambiente, se usa el mismo algoritmo que para `\newcommand<>` para analizar sintácticamente los argumentos y la especificación de cubierta. Notamos que, como siempre, el *<texto final>* no puede contener ningún argumento como #1. En particular, no se tiene acceso a la especificación de cubierta. En este caso, normalmente es una buena idea usar el ambiente `altenv` en el *<texto inicial>*.

Ejemplo: Declarar un bloque de acción propio:

```
\newenvironment<>{mibloquenegritas}[1]{%
  \begin{actionenv}#2%
    \textbf{#1}
    \par}
{\par%
\end{actionenv}}

\begin{frame}
  \begin{mibloquenegritas}<2>
    {Este teorema se muestra en la segunda diapositiva.}
  \end{mibloquenegritas}
\end{frame}
```

Ejemplo: El texto en el siguiente ambiente normalmente esta en negritas y en las diapositivas no especificadas en itálicas:

```
\newenvironment<>{negritaonormal}
{\begin{altenv}#1
  {\begin{bfseries}}{\end{bfseries}}
  {}{}}
{\end{altenv}}
```

Incidentalmente, dado que `altenv` también acepta su argumento al final, se podría lograr el mismo efecto usando sólo

```
\newenvironment{negritaonormal}
{\begin{altenv}
  {\begin{bfseries}}{\end{bfseries}}
  {}{}}
{\end{altenv}}
```

`\renewenvironment<>`{<nombre de ambiente existente>}[<número de argumentos>][<valor opcional pre-definido>]{< texto inicial>}{<texto final>}

Redefine un ambiente existente. El ambiente original todavía está disponible bajo el nombre *original* *<nombre de ambiente existente>*.

Ejemplo:

```
\renewenvironment<>{verse}
{\begin{actionenv}#1\begin{originalverse}}
{\end{originalverse}\end{actionenv}}
```

Los dos comandos siguientes se pueden usar para asegurar que un cierto contador se restablece automáticamente en las subsiguientes diapositivas de un marco. Esto es necesario por ejemplo para el conteo de ecuaciones. Se podría querer que este conteo se incremente de marco en marco, pero ciertamente no de cubierta de diapositiva en cubierta de diapositiva. Para el conteo de ecuaciones y de pies de página (no deberíamos usar pies de página), ya han sido invocados estos comandos.

`\resetcounteronoverlays`{<nombre del contador>}

Después de que se invoca este comando, el valor del contador especificado será el mismo en todas las diapositivas de cada marco.

Ejemplo: `\resetcounteronoverlays{equation}`

`\resetcountonoverlays{<nombre de registro de conteo>}`

Lo mismo que `\resetcounteronoverlays`, sólo que este comando se debe usar con conteos que se han creado usando el comando primitivo de T_EX `\newcount` en lugar de `\definecounter` de L^AT_EX.

Ejemplo:

```
\newcount\micuenta
\resetcountonoverlays{micuenta}
```

Especificaciones de modo

Esta sección sólo es importante si se usa el mecanismo de modo BEAMER para crear diferentes versiones de una presentación.

En ciertos casos se quisiera tener distintas especificaciones de cubierta para aplicar a un comando en los diferentes modos. Por ejemplo, se podría desear que cierto texto aparezca durante la presentación sólo de la tercera diapositiva en adelante, pero en un folleto para el público no debe haber tercera diapositiva y el texto debe aparecer ya solo en la segunda diapositiva. En este caso se podría escribir:

```
\only<3|handout:2>{Algún texto}
```

La barra vertical, a la que debe seguir un espacio, separa las dos diferentes especificaciones `3` y `handout:2`. Al escribir un nombre de modo antes de dos puntos, se indica que la siguiente especificación sólo se aplica a ese modo. Si no se da ningún modo, como en `3`, se agrega automáticamente el modo `beamer`. Por esta razón, si se escribe `\only <3>{Texto}` y se está en modo `handout`, el texto se mostrará en todas las diapositivas dado que no se especifica restricción alguna para los folletos y dado que `3` es lo mismo que `beamer:3`.

También es posible dar una especificación de cubierta que contiene sólo un nombre de modo (o varios, separados por barras verticales):

```
\only<article>{Este texto se muestra sólo en el modo article.}
```

A una especificación de cubierta que no contiene ningún número de diapositiva se le llama una *especificación de modo* (pura).

Si se da una especificación de modo, se suprimen automáticamente todos los modos que no se mencionan. Así `<beamer:1->` significa “en todas las diapositivas en modo `beamer` y también en todas las diapositivas de los otros modos, dado que nada especial se especifica para ellos”, mientras que `<beamer>` significa “en todas las diapositivas en modo `beamer` y en ninguna otra diapositiva de los demás modos”.

Las especificaciones de modo también se pueden usar fuera de los marcos como en los siguientes ejemplos:

```
\section<presentation>{Esta sección sólo existe en los modos de presentación}
\section<article>{Esta sección sólo existe en el modo article}
```

Se pueden mezclar especificaciones de modo puras y especificaciones de cubierta, aunque esto puede causar confusión:

```
\only<article|beamer:1>{Adivinanza}
```

Esto causa que se inserte el texto `Adivinanza` en el modo `article` y también en la primera diapositiva de un marco en el modo `beamer`, y nada en absoluto en los modos `handout` o `trans`. (Intentar averiguar cómo `<beamer | beamer:1>` difiere de `<beamer>` y de `<beamer:1>`.)

Como si todo esto no fuera ya lo suficientemente complicado, hay otro modo que se comporta de manera especial: el modo `second`. Para este modo se aplica una regla especial: una especificación de cubierta para el modo `beamer` también aplica al modo `second` (pero no al revés). Así, si se está en el modo `second`, la especificación `<second:2>` significa “en la diapositiva 2” y `<beamer:2>` también significa “en la diapositiva 2”. Para conseguir que una diapositiva se edite en modo `beamer`, pero no en el modo `second`, se puede usar `<second:0>`.

Especificaciones de acción

Esta sección introduce un concepto bastante avanzado. Algunos comandos conscientes de especificación de cubierta no sólo manejan especificaciones de cubierta normales, sino también las llamadas *especificaciones de acción*. En una especificación de acción, la lista de números o rango de diapositivas se prefija por `<acción>@`, donde `<acción>` es el nombre de una cierta acción que se lleva a cabo en las diapositivas indicadas:

```
\item<3-| alert@3> Se muestra a partir de la diapositiva 3, resaltada en rojo en la 3.
```

En el ejemplo anterior, el comando `\item`, que permite especificar acciones, destapa su texto de la diapositiva tres en adelante, además resalta tal texto exactamente en la diapositiva 3.

No todos los comandos pueden tomar una especificación de acción. Actualmente, sólo las manejan: `\item` (aunque no en el modo `article`, por el momento), `\action`, el ambiente `actionenv` y los ambientes de bloque (como `block` o `theorem`).

Por defecto, las siguientes acciones están disponibles:

- `alert` destaca en rojo el objeto o el bloque.
- `uncover` destapa el objeto o bloque (si no se especifica ninguna acción, es el valor por defecto).
- `only` causa que el objeto o bloque entero se inserte sólo en las diapositivas especificadas.
- `visible` causa que el texto sólo sea visible en las diapositivas especificadas (la diferencia entre `uncover` y `visible` es la misma que hay entre `\uncover` y `\visible`).
- `invisible` causa que el texto se vuelva invisible en las diapositivas especificadas.

El resto de esta sección explica cómo se agregan acciones propias y cómo fabricar comandos conscientes de especificación de acción. Se pueden agregar fácilmente acciones propias: una especificación de acción como `<acción>@<números de diapositivas>` simplemente inserta un ambiente llamado `<acción>env` alrededor de `\item` o inserta el parámetro `<números de diapositivas>` como especificación de cubierta para `\action`. Así, para definir un nuevo ambiente consciente de especificación de acción llamado `<mi nombre de acción>env`, se puede añadir una acción propia:

```
\newenvironment{checarenv}{\only{\setbeamertemplate{itemize item}{X}}}{}
```

Se puede escribir entonces

```
\item<beamer:checar@2> Texto.
```

Esto cambia el símbolo de listado a una X antes de `Texto.`, en la diapositiva 2 en modo `beamer`. La definición de `checarenv` usa el hecho de que `\only` puede aceptar también una especificación de cubierta después de su argumento.

Todo mecanismo de acción se basa en el siguiente ambiente:

```
\begin{actionenv}<especificación de acción>
<contenido de ambiente>
\end{actionenv}
```

Para el modo actual este ambiente extrae todas las acciones de la `<especificación de acción>`. Para cada acción de la forma `<acción>@<números de diapositivas>`, inserta el texto siguiente:

`\begin{<acción>env}<número de diapositiva>` al inicio del ambiente y el texto `\end{<acción>env}` al final. Si hay varias especificaciones de acción se abren varios ambientes (que se cierran en el orden apropiado). Una *<especificación de cubierta>* sin una acción se promueve a `uncover@<especificación de cubierta>`.

Si la llamada *especificación de cubierta predefinida* no es vacía, se usará en caso de que no se de ninguna *<especificación de acción>*. La especificación de cubierta predefinida normalmente es vacía, pero se puede establecer proporcionando un argumento opcional adicional al comando `\frame` o a los ambientes `itemize`, `enumerate` o `description`. Usar el comando `\beamerdefaultoverlayspecification` es otra forma de establecer la especificación de acción predefinida.

Ejemplo:

```
\begin{frame}
  \begin{actionenv}<2-| alert@3-4,6>
    Este texto se muestra de la misma manera que el texto que sigue.
  \end{actionenv}
  \begin{uncoverenv}<2->
    \begin{alertenv}<3-4,6>
      Este texto se muestra de la misma manera que el texto anterior.
    \end{alertenv}
  \end{uncoverenv}
\end{frame}
```

`\action<especificación de acción>{<texto>}`

Tiene el mismo efecto que cuando se pone *<texto>* en un `actionenv`.

Ejemplo: `\action<alert@2>{También se pudo haber usado \alert<2>{}}.`

`\beamerdefaultoverlayspecification{<especificación de cubierta predefinida>}`

Localmente asigna a la especificación de cubierta predefinida el valor dado. Esta especificación de cubierta se usa en cada ambiente `actionenv` y en cada `\item` que no tiene su propia especificación de cubierta. El principal uso de este comando es para instalar una especificación de cubierta incremental como `<+>` o `<+>| alert@+>`.

Normalmente, la especificación de cubierta predefinida se instala automáticamente por medio de los argumentos opcionales de `\frame`, `frame`, `itemize`, `enumerate` y `description`. Sólo se tiene que usar este comando si se desea hacer cosas divertidas.

Si se da fuera de cualquier marco, este comando establece la especificación de cubierta predefinida para todos los marcos siguientes para los que no se cancelo la especificación de cubierta predefinida.

Ejemplo: `\beamerdefaultoverlayspecification{<+>}`

Ejemplo: `\beamerdefaultoverlayspecification{}` limpia la especificación de cubierta predefinida. (En realidad instala la especificación de cubierta predefinida `<*>` que significa “siempre”, pero la manera “portable” de limpiar la especificación de cubierta predefinida es la primera.)

Especificaciones incrementales

Esta sección es para las personas que ya han usado muchas especificaciones de cubierta y se han cansado de escribir una y otra vez cosas como `<1->`, `<2->` y así sucesivamente.

A menudo se desean especificaciones de cubierta que siguen un modelo similar al siguiente:

```
\begin{itemize}
\item<1-> Manzana
\item<2-> Durazno
```

```
\item<3-> Ciruela
\end{itemize}
```

El problema empieza si se decide insertar una nueva fruta, digamos, al principio. En este caso, se tendría que ajustar todas las especificaciones de cubierta. También se tendrían que actualizar las especificaciones de cubierta si se desea añadir un comando `\pause` antes de `itemize`.

BEAMER ofrece una sintaxis especial más “robusta” para crear listas como la anterior. Se le puede reemplazar por la siguiente lista de *especificaciones de cubierta incrementales*:

```
\begin{itemize}
\item<+-> Manzana
\item<+-> Durazno
\item<+-> Ciruela
\end{itemize}
```

El símbolo `+-` se puede usar en cualquier especificación de cubierta, en cualquier lugar donde normalmente se usaría un número. Si se encuentra un símbolo `+-` se reemplaza por el valor actual del contador `beamerpauses` de L^AT_EX, que al inicio del marco es 1. Luego el contador se incrementa en 1, aunque sólo se incrementa una vez para cada especificación de cubierta, aun cuando la especificación contenga múltiples símbolos `+-` (se reemplaza por el mismo número).

En el ejemplo anterior, la primera especificación se reemplaza por `<1->`, la segunda por `<2->` y así sucesivamente. Ahora se pueden insertar nuevas entradas de una manera fácil, sin tener que cambiar nada. También se podría escribir:

```
\begin{itemize}
\item<+-| alert@+> Manzana
\item<+-| alert@+> Durazno
\item<+-| alert@+> Ciruela
\end{itemize}
```

Esto destacará el objeto actual cuando se le descubra. Por ejemplo, la primera especificación `<+-| alert@+>` se reemplaza por `<1-| alert@1>`, la segunda por `<2-| alert@2>` y así sucesivamente. Dado que el ambiente `itemize` también permite especificar una especificación de cubierta predefinida, el ejemplo anterior se puede escribir como sigue:

```
\begin{itemize}[<+-| alert@+>]
\item Manzana
\item Durazno
\item Ciruela
\end{itemize}
```

El comando `\pause` también actualiza el contador `beamerpauses`. Se puede cambiar este contador usando los comandos de L^AT_EX `\setcounter` o `\addtocounter`.

Cualquier ocurrencia de un símbolo `+-` puede ser seguido por un *desplazamiento* entre paréntesis. Este desplazamiento se agrega al valor de `beamerpauses`. Así, si `beamerpauses` es 2, entonces `<+(1)->` se expande a `<3->` y `<+(-1)->` se extiende a `<1-2>`.

Hay otro símbolo especial que se puede usar en una especificación de cubierta y se comporta de manera semejante al símbolo `+-`: un punto. Cuando se escribe `<.->`, pasa una cosa similar como en `<+->` *sólo que no se incrementa el contador `beamerpauses` y sólo se consigue que el valor de `beamerpauses` disminuya en 1*. Así un punto, posiblemente seguido de un desplazamiento, sólo expande el valor actual del contador `beamerpauses` menos uno. Esta notación del punto puede ser útil en un caso como el siguiente:

```
\begin{itemize}[<+->]
\item Manzana
```

```
\item<.-> Durazno  
\item Ciruela  
\end{itemize}
```

Dado que no se actualiza el contador, el segundo objeto se muestra al mismo tiempo que el primero.

En el ejemplo siguiente, cada vez que se descubre un objeto, se destaca el texto indicado. Cuando se descubre el siguiente objeto finaliza el realce.

```
\begin{itemize}[<+>]  
\item Esto es \alert<.>{importante}.  
\item Queremos \alert<.>{destacar} esto y \alert<.>{esto}.  
\item Cual es la \alert<.>{matriz}?  
\end{itemize}
```

9. Estructurando una Presentación: La Estructura Global Estática

Esta sección lista los comandos que se usan para estructurar “globalmente” una presentación tales como `\section` o `\part`. Éstos crean una estructura estática, lo que significa que la presentación resultante normalmente muestra una diapositiva después de la otra, en el orden que aparecen. Para la estructura *interactiva* es necesario que interactuemos con el programa de presentación, pulsando sobre hiperligas para avanzar en la presentación.

9.1. Añadiendo una página de título

Se puede usar el comando `\titlepage` para insertar una página de título en un marco. Por defecto, en ésta se coloca lo siguiente: el título del documento, los nombres del autor(es), su afiliación, un gráfico y una fecha.

`\titlepage`

Inserta el texto de una página de título en el marco actual.

Ejemplo: `\frame{\titlepage}`

Ejemplo: `\frame[plain]{\titlepage}` para una página de título que llena el marco entero.

Plantilla-/Color-/Fuente-Beamer title page

Se invoca esta plantilla cuando se usa el comando `\titlepage`.

Las siguientes opciones de plantilla están predefinidas:

- `[default]` [*alineación*] La página de título se edita mostrando el título, seguido por el nombre del autor, su afiliación, la fecha y un gráfico. Si no se proporcionan algunos datos, éstos no se muestran. Salvo por el gráfico, si los colores-BEAMER `title`, `author`, `institute` o `date` están definidos, entonces se usan como colores del texto de las respectivas entradas. Si se les define un color de fondo, detrás de ellos se dibuja una barra coloreada del color correspondiente, abarcando el ancho del texto. Para estas entradas se usan las correspondientes fuentes-BEAMER.

La opción *alineación* se pasa al comando `beamercolorbox` y se puede usar, por ejemplo, para jalar la página de título a la izquierda poniendo la opción `left`. Los siguientes comandos son útiles para esta plantilla:

- `\insertauthor` Inserta una versión del nombre del autor, útil para la página de título.
- `\insertdate` Inserta la fecha.
- `\insertinstitute` Inserta el instituto.
- `\inserttitle` Inserta una versión del título del documento, útil para la página de título.
- `\insertsubtitle` Inserta una versión del subtítulo del documento, útil para la página de título.
- `\inserttitlegraphic` Inserta un gráfico en una plantilla.

Para tener compatibilidad con otras clases, en el modo `article` se proporciona el siguiente comando:

`\maketitle`

PRESEN- Si se usa dentro de un marco tiene el mismo efecto que `\titlepage`. Si se usa fuera de uno, tiene el **TACIÓN** mismo efecto que `\frame{\titlepage}`; es decir, si es necesario se agrega un marco.

Antes de invocar el comando `\titlepage` se deben especificar todos los elementos que se desean mostrar. Esto se hace usando los siguientes comandos:

`\title[⟨título corto⟩]{⟨título⟩}` El `⟨título corto⟩` se usa en encabezados y pies de página. Dentro del `⟨título⟩` se pueden insertar cambios de línea con el comando doble-backslash (`\\`).

Ejemplos:

```
\title{La Clase Beamer}
\title[Versión corta]{Un Título Muy Largo \\ Varias líneas más}
```

ARTÍCULO En este modo se ignora la forma corta.

`\subtitle[⟨subtítulo corto⟩]{⟨subtítulo⟩}` Por defecto no se usa el `⟨subtítulo corto⟩`, pero está disponible vía la inserción de `\insertshortsubtitle`. El subtítulo se muestra debajo del título con una fuente más pequeña.

Ejemplo:

```
\title{La clase Beamer}
\subtitle{Una introducción con muchos ejemplos.}
```

ARTÍCULO Este comando causa que se añada al título un cambio de línea y un comando `\normalsize` antes del subtítulo. Esto puede o no ser algo que nos guste.

`\author[⟨nombre(s) de autor(es) corto⟩]{⟨nombre(s) de autor(es)⟩}`

Los nombres se separan usando el comando `\and`. Si los autores tienen diferentes afiliaciones, éstas se ponen como sufijos usando el comando `\inst` con diferentes parámetros.

Ejemplo: `\author[Hemaspaandra et al.]{L. Hemaspaandra\inst{1} \and T. Tantau\inst{2}}`

ARTÍCULO En este modo se ignora la forma corta.

`\institute[⟨nombre de instituto corto⟩]{⟨instituto⟩}`

Si hay más de un instituto, se deben separar usando el comando `\and` y se ponen como sufijos usando el comando `\inst` con diferentes parámetros.

Ejemplo:

```
\institute[Universities of Rochester and Berlin]{
  \inst{1}Department of Computer Science\\
  University of Rochester
  \and
  \inst{2}Fakultät für Elektrotechnik und Informatik\\
  Technical University of Berlin}
```

ARTÍCULO En este modo se ignora la forma corta. La forma larga también se ignora, excepto si la clase del documento (como `llncs`) la define.

`\date[⟨fecha corto⟩]{⟨fecha⟩}`

Ejemplo: `\date{\today}` o `\date[UMALCA 2003]{Conferencia UMALCA, 2003}`.

ARTÍCULO En este modo se ignora la forma corta.

`\titlegraphic{⟨texto⟩}`

El `⟨texto⟩` se muestra como gráfico. Típicamente se usa un ambiente de imagen como `⟨texto⟩`.

Ejemplo: `\titlegraphic{\pgfuseimage{gráfico}}`

ARTÍCULO En este modo se ignora el comando.

`\subject{⟨texto⟩}`

Introduce el *⟨texto⟩* como información-acerca-del-tema en la parte de información del documento en PDF. Actualmente no tiene otro efecto.

`\keywords{⟨texto⟩}`

Introduce el *⟨texto⟩* como palabras-clave en la parte de información del documento en PDF. Actualmente no tiene otro efecto.

Por defecto, los comandos `\title` y `\author` también insertan sus argumentos en los campos de información del documento PDF resultante. Esto puede causar problemas si se usan cosas complicadas como cajas en los argumentos de estos comandos. En este caso, se puede anular la generación automática de estas entradas usando la siguiente opción de clase:

`\documentclass[usepdftitle=false]{beamer}`

Suprime la generación automática de las entradas de título y de autor en la información del documento en PDF.

9.2. Agregando secciones y subsecciones

Se puede estructurar el texto usando los comandos `\section` y `\subsection`. A diferencia del \LaTeX normal, estos comandos no crean un título en la posición donde se les usa. Más bien, agregan una entrada en la tabla de contenidos y en la barra de la navegación. Para crear un cambio de línea en la tabla de contenidos (aunque no es una buena idea), se puede usar el comando `\breakhere`. El comando `\\` no funciona.

`\section<⟨especificación de modo⟩>[⟨nombre de sección corto⟩]{⟨nombre de sección⟩}`

Se inicia una sección. No se crea ningún encabezado. El *⟨nombre de sección⟩* se muestra en la tabla de contenidos y en la barra de navegación, excepto si se especifica un *⟨nombre de sección corto⟩*. En este caso se usa el *⟨nombre de sección corto⟩* en la barra de navegación. Si se da una *⟨especificación de modo⟩*, el comando sólo tiene efecto para los modos indicados.

Ejemplo: `\section[Resumen]{Resumen de los Resultados Principales}`

ARTÍCULO En este modo la *⟨especificación de modo⟩* permite proporcionar un comando alternativo de sección. Por ejemplo, esto es necesario si el *⟨nombre de sección corto⟩* es impropio para la tabla de contenidos:

Ejemplo:

```
\section<presentation>[Resultados]{Resultados sobre los Problemas Principales}
\section<article>{Resultados sobre los Problemas Principales}
```

Plantilla-/Color-/Fuente-Beamer `section in toc`

Esta plantilla se usa cuando se va a editar una entrada de sección. Para opciones admisibles ver la plantilla padre `table of contents`.

Los siguientes comandos son útiles para esta plantilla:

- `\inserttocsection` Inserta la versión de tabla de contenidos del nombre de la sección actual.
- `\inserttocsectionnumber` Inserta el número de la sección actual (en la tabla de contenidos).

Plantilla-/Color-/Fuente-Beamer `section in toc shaded`

Se usa en lugar de la anterior si la sección se debe mostrar de manera sombreada, debido a que no es la sección actual. Para opciones admisibles ver la plantilla padre `table of contents`.

`\section<⟨especificación de modo⟩>*<⟨nombre de sección⟩>`

Inicia una sección sin una entrada en la tabla de contenidos. No se crea ningún encabezado, pero el *<nombre de sección>* se muestra en la barra de la navegación.

Ejemplo: `\section*{Esbozo}`
Ejemplo: `\section<beamer>*{Esbozo}`

`\subsection<especificación de modo>[<nombre corto de subsección>]{<nombre de subsección>}`

Este comando funciona de la misma manera que el comando `\section`.

Ejemplo: `\subsection[Aplicaciones]{Aplicaciones a la Reducción de la Polución}`

Plantilla-/Color-/Fuente-Beamer `subsection in toc`

Como `section in toc`, sólo que para subsecciones.

Además de las inserciones para la plantilla `section in toc`, los siguientes comandos están disponibles para esta plantilla:

- `\inserttocsubsection` inserta la versión de tabla de contenidos del nombre de la subsección actual.
- `\inserttocsubsectionnumber` Inserta el número de la subsección actual (en la tabla de contenidos).

Plantilla-/Color-/Fuente-Beamer `subsection in toc shaded`

Como `section in toc shaded`, sólo que para subsecciones.

`\subsection<especificación de modo>*<nombre de subsección>`

Inicia una subsección sin una entrada en la tabla de contenidos. No se crea ningún encabezado, pero el *<nombre de subsección>* se muestra en la barra de navegación, excepto si el *<nombre de subsección>* está vacío. En este caso, no se crea una entrada ni en la tabla de contenidos ni en la barra de navegación, pero en la barra de navegación se muestra todo marco en esta subsección “vacía”.

Ejemplo:

```
\section*{Resumen}
\frame{Este marco no se muestra en la barra de navegaci\'}{o}n}
\subsection*{}
\frame{\'}{E}ste se muestra en la barra de navegaci\'}{o}n pero no se muestra una
entrada de subsecci\'}{o}n}
\subsection*{Una subsecci\'}{o}n}
\frame{Marco normal, se muestra en la barra de navegaci\'}{o}n. El nombre de la
subsecci\'}{o}n tambi\'}{e}n se muestra en la barra de navegaci\'}{o}n pero no en la tabla
de contenidos}
```

`\subsubsection<especificación de modo>[<nombre corto de subsubsección>]{<nombre de subsubsección>}`

Este comando funciona como `\subsection`. Sin embargo, las subsubsecciones se soportan menos bien que las subsecciones. Por ejemplo, en la tabla de contenidos las subsubsecciones siempre se muestran con los mismos parámetros de sombreado/ocultamiento que la subsección. Se *desaprueba* fuertemente el uso de subsubsecciones en una presentación.

Ejemplo: `\subsubsection[Aplicaciones]{Aplicaciones a la Reducción de la Polución}`

Plantilla-/Color-/Fuente-Beamer `subsubsection in toc`

Como `subsection in toc`, sólo que para subsubsecciones.

Además de las inserciones para la plantilla `subsection in toc`, los siguientes comandos están disponibles para esta plantilla:

- `\inserttocsubsubsection` inserta la versión de tabla de contenidos del nombre de la subsección actual.
- `\inserttocsubsubsectionnumber` Inserta el número de la subsubsección actual (en la tabla de contenidos).

Plantilla-/Color-/Fuente-Beamer `subsubsection in toc shaded`

Como `subsection in toc shaded`, sólo que para subsubsecciones.

`\subsubsection<{especificación de modo}>*\{<nombre de subsubsección>\}`

Inicia una subsección sin una entrada en la tabla de contenidos. No se crea ningún encabezado, pero el `<nombre de subsubsección>` se muestra en una posible barra lateral.

A menudo, se quiere mostrar cierto tipo de marco directamente después del inicio de una sección o subsección. Por ejemplo, se puede querer que cada subsección inicie con un marco que muestra la tabla de contenidos con la subsección actual resaltada. Para facilitar esto, se pueden usar los siguientes comandos.

`\AtBeginSection[<texto especial asterisco>]\{<texto>\}`

El `<texto>` dado se inserta al inicio de cada sección. Si se da el parámetro `<texto especial asterisco>`, éste en cambio se usará para las secciones con `*`. Las diferentes llamadas de este comando no “sumará” los textos dados (como lo hace el comando `\AtBeginDocument`), sino que borrará cualquier texto anterior.

Ejemplo:

```
\AtBeginSection[] % No hace nada para \section*
{
  \begin{frame}<beamer>
    \frametitle{Esbozo}
    \tableofcontents[currentsection]
  \end{frame}
}
```

ARTÍCULO En este modo no tiene efecto.

`\AtBeginSubsection[<texto especial asterisco>]\{<texto>\}`

El `<texto>` dado se inserta al inicio de cada subsección. Si se da el parámetro `<texto especial asterisco>`, éste en cambio se usará para las subsubsecciones con `*`. Las diferentes llamadas de este comando no “sumará” los textos dados (como lo hace el comando `\AtBeginDocument`), sino que borrará cualquier texto anterior.

Ejemplo:

```
\AtBeginSubsection[] % No hace nada para \subsection*
{
  \begin{frame}<beamer>
    \frametitle{Bosquejo}
    \tableofcontents[currentsection,currentsubsection]
  \end{frame}
}
```

`\AtBeginSubsubsection[<texto especial asterisco>]\{<texto>\}`

Como `\AtBeginSubsection`, sólo que para subsubsecciones.

9.3. Agregando partes

Si se da una plática larga (como una conferencia), ésta se puede dividir en varias partes. Cada parte actúa como una pequeña “plática por derecho propio” con su propia tabla de contenidos, su propia barra

de navegación, etcétera. Dentro de una parte, las secciones y subsecciones de las otras partes no se muestran en absoluto.

Para crear una parte nueva, se usa el comando `\part` y todas sus secciones, subsecciones, etc. serán “locales” a ella. Como los comandos `\section` y `\subsection`, el comando `\part` no causa que se produzca ningún marco o texto especial. Sin embargo, a menudo es aconsejable usar el comando `\partpage` para insertar algún texto en un marco que “anuncia” el inicio de una parte nueva.

`\part`<especificación de modo>[<nombre corto de parte>]{<nombre de parte>}

Inicia una parte. El <nombre de parte> se muestra cuando se usa el comando `\partpage`. Por defecto, el <nombre corto de parte> no se muestra en ningún lugar, pero es accesible vía el comando `\insertshortpart`.

Ejemplo:

```
\begin{document}
  \frame{\titlepage}

  \section*{Bosquejos}
  \subsection{Parte I: Repaso de la Conferencia Previa}
  \frame{
    \frametitle{Bosquejo de la Parte I}
    \tableofcontents[part=1]}
  \subsection{Parte II: Conferencia de hoy}
  \frame{
    \frametitle{Bosquejo de la Parte II}
    \tableofcontents[part=2]}

  \part{Repaso de la Conferencia Previa}
  \frame{\partpage}
  \section[Conferencia Previa]{Resumen de la Conferencia Previa }
  \subsection{T\’{o}picos}
  \frame{...}
  \subsection{Objetivos de Aprendizaje}
  \frame{...}

  \part{Conferencia de hoy}
  \frame{\partpage}
  \section{T\’{o}pico A}
  \frame{\tableofcontents[currentsection]}
  \subsection{Foo}
  \frame{...}
  \section{T\’{o}pico B}
  \frame{\tableofcontents[currentsection]}
  \subsection{bar}
  \frame{...}
\end{document}
```

`\partpage`

Funciona como `\titlepage` sólo que “anuncia” la parte actual, no la presentación actual.

Ejemplo: `\frame{\partpage}`

Plantilla-/Color-/Fuente-Beamer part page

Se invoca esta plantilla cuando se usa el comando `\partpage`.

Las siguientes opciones de la plantilla están predefinidas

- `[default]` [`\alineación`] la página de parte se edita mostrando el número de la parte actual y debajo el título de la parte actual. Se usa el color- y fuente-BEAMER `part page`, incluyendo el color de fondo `part page`. Como en la plantilla `title page`, se pasa la opción `\alineación` al comando `beamercolorbox`.

Los siguientes comandos son útiles para esta plantilla:

- `\insertpart` Inserta el nombre de la parte actual.
- `\insertpartnumber` Inserta en una plantilla el número de la parte actual usando un número árabe.
- `\insertpartromannumber` Inserta en una plantilla el número de la parte actual con un número romano.

`\AtBeginPart` $\langle texto \rangle$

El $\langle texto \rangle$ dado se inserta al principio de cada parte.

Ejemplo:

```
\AtBeginPart{\frame{\partpage}}
```

9.4. Dividiendo un curso en conferencias

Al usar BEAMER con el modo `article`, se puede querer que las notas de las conferencias de un curso entero residan en un solo archivo. En este caso, sólo unos marcos son realmente parte de alguna conferencia particular.

El comando `\lecture` facilita la selección de un cierto conjunto de marcos en un archivo para luego ser presentados. Este comando toma (entre otras cosas) un nombre de etiqueta. Si se pone `\includeonlylecture` con este nombre de etiqueta, entonces se muestran sólo los marcos que siguen al correspondiente comando `\lecture`. Se suprimen los marcos que siguen a otros comandos `\lecture`. Por defecto, el comando `\lecture` no tiene otro efecto. No crea ningún marco o introduce entradas en la tabla de contenidos. Sin embargo, se puede usar `\AtBeginLecture` para tener inserciones de BEAMER, digamos, una página de título al principio de cada conferencia.

`\lecture` [`\nombre corto de conferencia`] $\langle nombre de conferencia \rangle$ $\langle etiqueta de conferencia \rangle$

Inicia una conferencia. El $\langle nombre de conferencia \rangle$ y el $\langle nombre corto de conferencia \rangle$ están disponibles vía los comandos `\insertlecture` y `\insertshortlecture` respectivamente.

Ejemplo:

```
\begin{document}
\lecture{Espacios Vectoriales}{Semana 1}

\section{Introducci\'}{o}n
...

\section{Resumen}

\lecture{Producto Escalar}{Semana 2}

\section{Introducci\'}{o}n
...
\section{Resumen}
```

```
\end{document}
```

ARTÍCULO En este modo no tiene efecto.

```
\includeonlylecture<etiqueta de conferencia>
```

Causa que se supriman todos los comandos `\frame`, `frame`, `\section`, `\subsection` y `\part` que siguen al comando `\lecture`, excepto si la etiqueta de la conferencia coincide con la *<etiqueta de conferencia>*. Siempre se incluyen los marcos que preceden a los comandos `\lecture`. Este comando se debe incluir en el preámbulo.

Ejemplo:

```
\includeonlylecture{semana 1}
```

ARTÍCULO En este modo no tiene efecto.

```
\AtBeginLecture{<texto>}
```

El *<texto>* se inserta al principio de cada conferencia.

Ejemplo:

```
\AtBeginLecture{\frame{\Large Conferencia de hoy: \insertlecture}}
```

ARTÍCULO En este modo no tiene efecto.

9.5. Agregando una tabla de contenidos

El comando `\tableofcontents` crea una tabla de contenidos. Al contrario del comando `table of contents` de L^AT_EX normal, este comando toma un parámetro opcional entre corchetes que se usa para crear ciertos efectos especiales.

```
\tableofcontents[<lista de opciones separadas por comas>]
```

Inserta una tabla de contenidos en el marco actual.

Ejemplo:

```
\section*{Esbozo}
\frame{\tableofcontents}

\section{Introducci\'}{o}n}
\frame{\tableofcontents[currentsection]}
\subsection{¿Porque?}
\frame{...}
\subsection{¿Donde?}
\frame{...}

\section{Resultados}
\frame{\tableofcontents[currentsection]}
\subsection{Debido}
\frame{...}
\subsection{Aqu\'}{i}}
\frame{...}
```

Se pueden dar las siguientes opciones:

- `currentsection` causa que todas las secciones se muestren de manera semi transparente, excepto la actual. También, todas las subsecciones se muestran de manera semi transparente excepto las de la sección actual. Este comando es una forma corta para especificar las siguientes opciones:

`sectionstyle=show/shaded,subsectionstyle=show/show/shaded`

- **currentsubsection** causa que todas las subsecciones se muestren de manera semi transparente, excepto la actual. Este comando es una forma corta para especificar la opción:
`subsectionstyle=show/shaded`.
- **firstsection**=*<número de sección>* especifica qué sección debe numerarse como sección “1”. Es útil si se tiene una primera sección (como una sección general) que no recibirá un número. Por defecto no se muestran números de sección, para mostrarlos se deben instalar diferentes plantillas de tabla de contenidos.
- **hideallsubsections** causa que se oculten todas las subsecciones. Es una forma corta para especificar la opción `subsectionstyle=hide`.
- **hideothersubsections** causa que se oculten las subsecciones de secciones excepto las de la actual. Es una forma corta para especificar la opción `subsectionstyle=show/show/hide`.
- **part**=*<número de parte>* causa que se muestre la tabla de contenidos de la parte *<número de parte>*, en lugar de la tabla de contenidos de la parte actual (qué es el valor por defecto). Esta opción se puede combinar con otras, aunque obviamente combinarla con la opción **current** no tiene ningún sentido.
- **pausesections** causa que se emita un comando `\pause` antes de cada sección. Esto es útil si se desea mostrar de manera incremental la tabla de contenidos.
- **pausesubsections** causa que se emita un comando `\pause` antes de cada subsección.
- **sections**=*<especificación de cubierta>* en este caso se causa que se muestren sólo las secciones mencionadas en la *<especificación de cubierta>*. Por ejemplo, `sections={<2-4 | handout:0>}` causa que se muestren sólo la segunda, tercera y cuarta sección en la versión normal, no se muestra nada en la versión folleto y se muestra todo en las demás versiones. Por conveniencia, si se omiten los paréntesis angulares, se asume que la especificación se aplica a todas las versiones. Así `sections={2-4}` causa que se muestren las secciones dos, tres y cuatro en todas las versiones.
- **sectionstyle**=*<estilo para la sección actual>/<estilo para otras secciones>* especifica cómo se deben desplegar las secciones. Los estilos permitidos son **show**, **shaded** y **hide**. Normalmente el primero muestra el título de la sección, el segundo lo muestra de una manera semi transparente y el tercero lo suprime completamente. También se puede omitir el segundo estilo en cuyo caso se usa el primero para todas las secciones (esto no es muy útil).
- **subsectionstyle**=*<estilo para la subsección actual>/<estilo para otras subsecciones en la sección actual>/<estilo para subsecciones en otras secciones>*
Especifica cómo se deben desplegar las subsecciones. Se pueden dar los mismos estilos que la opción **sectionstyle**. Se puede omitir el último estilo en cuyo caso lo sustituye el segundo y se pueden omitir los últimos dos en cuyo caso el primero se aplica a todos.

Ejemplo: `subsectionstyle=shaded` causa que se oscurezcan todas las subsecciones.

Ejemplo: `subsectionstyle=hide` causa que se oculten todas las subsecciones.

Ejemplo: `subsectionstyle=show/shaded` causa que se muestren de una manera semi transparente todas las subsecciones excepto la subsección actual en la sección actual.

Ejemplo: `subsectionstyle=show/show/hide` causa que se supriman todas las subsecciones fuera de la sección actual.

Ejemplo: `subsectionstyle=show/shaded/hide` causa que se supriman todas las subsecciones fuera de la sección actual y sólo resalta la subsección actual en la sección actual.

Estos ejemplos son útiles si no se desea mostrar demasiados detalles al presentar el esbozo de la plática.

ARTÍCULO Se ignoran las opciones.

Plantilla-Beamer Padre `sections/subsections in toc`

Es una plantilla padre cuyos productos son `section in toc` y `subsection in toc`. Esto significa que si se usa el comando `\setbeamertemplate` sobre esta plantilla, éste a su vez es llamado por ambos productos (con los mismos argumentos).

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` En el entorno predefinido, las secciones y subsecciones se editan usando las fuentes y colores `section in toc` y `subsection in toc`, aunque se ignoran los colores de fondo. Las subsecciones están indentadas.
- `[sections numbered]` Similar al entorno predefinido pero se muestran también los números de sección. Las subsecciones no se numeran.
- `[subsections numbered]` Esta vez se numeran las subsecciones pero no las secciones. Sin embargo, dado que las subsecciones están “completamente numeradas” como en “1.2” o “3.2”, si cada sección tiene por lo menos una subsección realmente no se extrañará la sección numerada.
- `[circle]` Dibuja círculos pequeños antes de las secciones. Dentro de los círculos se muestra el número de la sección. Para editar los círculos se usa la fuente y color-BEAMER `section number projected`. Es decir, el círculo consigue el color de fondo y el texto dentro del círculo consigue el color del primer plano.
- `[square]` Similar a la opción `circle`. Muestra pequeños cuadrados no numerados delante de las subsecciones.
- `[ball]` Como `square` sólo que se usan bolas en lugar de cuadrados.
- `[ball unnumbered]` Similar a `ball` sólo que no se usa la numeración. Esta opción hace que la tabla de contenidos se vea como un `itemize`.

Si lo anterior no se ajusta a lo que se desea, se tiene que cambiar directamente las plantillas `section in toc` y `subsection in toc`.

Plantilla-Beamer Padre `sections/subsections in toc shaded`

Son plantillas padre cuyos productos son `section in toc shaded` y `subsection in toc shaded`. Se usan para representar las entradas de sección y subsección cuando están sombreadas actualmente, como en el caso de todas las subsecciones no actuales en `\tableofcontents[currentsubsection]`.

Las siguientes opciones de la plantilla están predefinidas:

- `[default][<opacidad>]` En el entorno predefinido, las plantillas `section in toc shaded` y `subsection in toc shaded` sólo muestran las versiones no oscurecidas de estas plantillas, pero opacas sólo al `<opacidad>%`. El valor por defecto es 20%.

Ejemplo: `\setbeamertemplate{table of contents shaded}[default][50]` hace entradas transparentes al 50%

9.6. Agregando una bibliografía

En una presentación en BEAMER se puede usar el ambiente `bibliography` y el comando `\cite` de \LaTeX . Típicamente se tienen que editar los objetos de la bibliografía “a mano”. No obstante, se *puede* usar `bibtex` para crear una “primera aproximación” de la bibliografía. Copie en su presentación el contenido del archivo `main.bbl`. Si no se está familiarizado con `bibtex`, se puede consultar su documentación. Es una herramienta poderosa para crear citas de alta calidad. Usando `bibtex` o el editor, las referencias bibliográficas se ponen en el ambiente `thebibliography`. Este ambiente (estándar de \LaTeX) toma un parámetro que debe ser la etiqueta `\bibitem` más larga en la lista de entradas bibliográficas siguiente.

```
\begin{thebibliography} {\langle texto de la etiqueta mas larga \rangle}
\langle contenido del ambiente \rangle
\end{thebibliography}
```

Inserta una bibliografía en el marco actual. El `\langle texto de la etiqueta mas larga \rangle` se usa para determinar la sangría de la lista. Sin embargo, varias opciones predefinidas para la edición de la bibliografía ignoran este parámetro dado que reemplazan las referencias por un símbolo.

Dentro del ambiente, se usa el comando `\bibitem` (de \LaTeX normal) para cada objeto de referencia. Dentro de cada objeto, se use el comando `\newblock` (de \LaTeX normal) para separar los nombres de los autores, el título, la referencia del libro o revista o cualquier nota. Cada uno de estos comandos puede introducir una nueva línea o color u otro formato, distinta a la indicada por la plantilla para bibliografías.

El ambiente se debe poner dentro de un marco. Si la bibliografía no cabe en un marco, se debe dividir (se crea un marco nuevo y un segundo ambiente `thebibliography`) o se usa la opción `allowframebreaks`. Aun mejor, se debe revisar si es buena idea presentar tantas referencias.

Ejemplo:

```
\begin{frame}
\frametitle{Para lectura posterior}

\begin{thebibliography}{Dijkstra, 1982}
\bibitem[Solomaa, 1973]{Solomaa1973}
A.~Salomaa.
\newblock {\em Lenguajes formales}.
\newblock Academic Press, 1973.

\bibitem[Dijkstra, 1982]{Dijkstra1982}
E.~Dijkstra.
\newblock Ordenaci\`{o}n suave, una alternativa para ordenar in situ.
\newblock {\em Ciencia de Programaci\`{o}n computacional}, 1(3):223--233, 1982.
\end{thebibliography}
\end{frame}
```

Cuatro plantillas gobiernan la apariencia de los campos `author`, `title`, `journal` y `note text`. La plantilla de autor se inserta antes del primer bloque de la entrada (el primer bloque es todo el texto antes de la primera ocurrencia de un comando `\newblock`). La plantilla título se inserta antes del segundo bloque (el texto entre la primera y segunda ocurrencia de `\newblock`). Funciona de manera similar para los campos revista y nota.

Las plantillas se insertan *antes* de los bloques y no se tiene acceso a ellos vía comandos de inserción. Además, antes de los bloques se instalan el color- y fuente-BEAMER correspondientes.

Plantilla-/Color-/Fuente-Beamer `bibliography entry author`

Esta plantilla se inserta antes del campo `author` de una entrada de bibliografía. También se instalan el color y la fuente. Notamos que el efecto de esta plantilla persistirá hasta el final del objeto (item) de bibliografía o hasta que una de las siguientes plantillas deshaga el efecto. Por defecto esta plantilla no hace nada. El color predefinido es el color `estructure`.

Plantilla-/Color-/Fuente-Beamer `bibliography entry title`

Esta plantilla se inserta antes del campo `title` de una entrada de bibliografía (más precisamente, se inserta después de la primera ocurrencia del comando `\newblock`). Por defecto esta plantilla inicia un nuevo párrafo causando un cambio de línea. El color predefinido es el color `normal text`.

Plantilla-/Color-/Fuente-Beamer `bibliography entry journal`

Esta plantilla se inserta antes del campo `journal` de una entrada de bibliografía (el segundo `\newblock`). Por defecto esta plantilla inicia un nuevo párrafo. El color predefinido es una versión ligeramente transparente del color `estructure`.

Plantilla-/Color-/Fuente-Beamer `bibliography entry note`

Esta plantilla se inserta antes de cualquier nota de texto al final de una entrada de bibliografía (se inserta antes del tercer `\newblock`). Por defecto esta plantilla inicia un nuevo párrafo. El color predefinido es igual al del campo `journal`.

`\bibitem`<especificación de cubierta>[<texto de la cita>]{<nombre de la etiqueta>}

El <texto de la cita> se inserta cuando se cita el objeto usando `\cite{<nombre de la etiqueta>}` en el texto de la presentación principal. Para una presentación BEAMER, normalmente este debe ser tan grande como sea posible.

Hay que usar comandos `\newblock` para separar los nombres de los autores, el título, la referencia del libro y/o revista y cualquiera nota. Si la <especificación de cubierta> está presente, la entrada sólo se mostrará en las diapositivas especificadas.

Ejemplo:

```
\bibitem[Dijkstra, 1982]{Dijkstra1982}
  E.~Dijkstra.
  \newblock Ordenaci\`{o}n suave, una alternativa para ordenar en situ.
  \newblock {\em La ciencia de programar}, 1(3):223--233, 1982.
```

Plantilla-/Color-/Fuente-Beamer `bibliography item`

Color/fuente padre: `item`

Esta plantilla se usa para representar un objeto (item) de la bibliografía. A diferencia del L^AT_EX normal, la plantilla predefinida para la bibliografía no repite el texto de la cita (como “[Dijkstra, 1982]”) antes de cada objeto en la bibliografía. En cambio, se dibuja un pequeño símbolo de artículo. La razón es que el público no es capaz de recordar hasta el final de la charla cualquier abreviatura del texto de la cita.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` Como referencia dibuja un pequeño icono de artículo. Usar esto para artículos de revistas, partes de libros o informes técnicos.
- `[article]` Alias para el valor por defecto.
- `[book]` Como referencia dibuja un pequeño icono de libro. Use esto para libros.
- `[triangle]` Como referencia dibuja un triángulo. Ésto esta más acorde con la representación de objetos en una lista.
- `[text]` Como texto de referencia usa el texto de cita (como “[Dijkstra, 1982]”). Si se usa esto hay que estar seguros de lo que se está haciendo.

La siguiente inserción es útil para la plantilla:

- `\insertbiblabel` Inserta la actual etiqueta de cita.

9.7. Agregando un apéndice

El comando `\appendix` agrega un apéndice a la conferencia. Quizás aquí se pondrán marcos y subsecciones que no se piensan mostrar durante la presentación, pero qué podrían ser útiles al contestar una pregunta. El comando `\appendix` esencialmente sólo inicia una nueva parte llamada `\appendixname`. Sin embargo, también establece ciertas hiperligas. Como otras partes, el apéndice se mantiene separado de la conferencia real.

`\appendix`<<especificación de modo>>

Inicia el apéndice en el modo indicado. Todos los marcos y todos los comandos `\subsection` y `\section` usados después de este comando no se muestran como parte de la barra de navegación normal.

Ejemplo:

```
\begin{document}
\frame{\titlepage}
\section*{Bosquejo}
\frame{\tableofcontents}
\section{Texto principal}
\frame{Alg\,'{u}n texto}
\section*{Resumen}
\frame{Texto resumen}

\appendix
\section{\appendixname}
\frame{\tableofcontents}
\subsection{Material adicional}
\frame{Detalles}
\frame{Texto omitido en la pl\,'{a}tica principal.}
\subsection{Incluso material adicional}
\frame{M\,'{a}s detalles}
\end{document}
```

10. Estructurando una Presentación: La Estructura Global Interactiva

10.1. Agregando hiperligas y botones

Para crear en la estructura de la conferencia saltos no lineales anticipados, se pueden agregar hiperligas a la presentación. Una hiperliga es un texto (normalmente representado como un botón) que, cuando se le pulsa, salta la presentación a alguna otra diapositiva. La creación de tal botón es un proceso de tres pasos:

1. Se especifica un objetivo usando el comando `\hypertarget` o (más fácil) el comando `\label`. En algunos casos este paso se puede saltar, ver más adelante.
2. Se representa el botón usando `\beamerbutton` o un comando similar. Esto *representará* al botón, pero al pulsarlo todavía no tendrá efecto alguno.
3. Se pone el botón dentro de un comando `\hyperlink`. Al pulsarlo se saltará al objetivo de la liga.

`\hypertarget`<{*especificación de cubierta*}>{<nombre del objetivo>}{<texto>}

Si está presente la *especificación de cubierta*, el *texto* sólo se muestra en la diapositiva especificada al hacer hipersaltos a *nombre del objetivo*. En todas las demás diapositivas el texto se muestra de manera normal. Hacemos notar que se *debe* agregar una especificación de cubierta al comando `\hypertarget` siempre que se usa en marcos que tienen múltiples diapositivas (de otra manera `pdflatex` se queja legítimamente porque se define el mismo objetivo en diferentes diapositivas).

Ejemplo:

```
\begin{frame}
  \begin{itemize}
    \item<1-> Primer objeto.
    \item<2-> Segundo objeto.
    \item<3-> Tercer objeto.
  \end{itemize}
  \hyperlink{saltoalasegunda}{\beamerbutton{Salto a la segunda diapositiva}}
  \hypertarget<2>{saltoalasegunda}{}
\end{frame}
```

ARTÍCULO En este modo se debe agregar en el preámbulo `\usepackage{hyperref}` para usar este comando.

Como efecto lateral el comando `\label` crea un hiperobjetivo y como efecto lateral la opción `label=<nombre>` del comando `\frame` crea una etiqueta llamada *nombre*< número de diapositiva > para cada diapositiva del marco. Así el ejemplo anterior se puede escribir así:

```
\begin{frame}[label=tresobjetos]
  \begin{itemize}
    \item<1-> Primer objeto.
    \item<2-> Segundo objeto.
    \item<3-> Tercer objeto.
  \end{itemize}
  \hyperlink{tresobjetos<2>}{\beamerbutton{Salto a la segunda diapositiva}}
\end{frame}
```

Los siguientes comandos se pueden usar para especificar de manera abstracta para que se va a usar un botón.

`\beamerbutton`{<texto del botón>}

Dibuja un botón con el *texto del botón* dado.

Example: `\hyperlink{algunlado}{\beamerbutton{Ir a algun lado}}`

ARTÍCULO En este modo, este comando (y los siguientes) sólo insertan sus argumentos.

Plantilla-/Color-/Fuente-Beamer `button`

Cuando se llama el comando `\beamerbutton`, esta plantilla se usa para representar el botón. Dentro de la plantilla se puede usar el comando `\insertbuttontext` para insertar el argumento que se pasó a `\beamerbutton`.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` Edita al botón con esquinas redondeadas. Se usa el primer plano y el fondo del color-BEAMER `button`, además de la fuente-BEAMER `button`. El borde del botón obtiene su primer plano del color-BEAMER `button border`.

Las siguientes inserciones son útiles para este elemento:

- `\insertbuttontext` inserta el texto del botón actual. Dentro de los botones “ir a”, este texto se prefiere con la inserción `\insertgotosymbol`. Lo mismo aplica para los botones de saltar y retornar.
- `\insertgotosymbol` Este texto se inserta al principio del botón “ir a”. Redefinir este comando para cambiar el símbolo.

Ejemplo: `\renewcommand{\insertgotosymbol}{\algúncomandoflecha}`

- `\insertskipsymbol` Este texto se inserta al principio de los botones de salto.
- `\insertreturnsymbol` Este texto se inserta al principio de los botones de retorno.

Beamer-Color `button border`

El primer plano de este color se usa para representar el borde de los botones.

`\beamergetobutton`{*texto del botón*}

Dibuja un botón con el *texto del botón* dado. Antes del texto se inserta un pequeño símbolo (normalmente una cabeza de flecha apuntando hacia la derecha) que indica que al pulsar ese botón se saltará a otra “área” de la presentación.

Ejemplo: `\hyperlink{desvío}{\beamergetobutton{Ir al desvío}}`

`\beamerskipbutton`{*texto del botón*}

El símbolo que se dibuja para este botón normalmente es una doble cabeza de flecha apuntando hacia la derecha. Se debe usar este botón si se quiere (al presionarlo) saltar una parte bien definida de la conferencia.

Ejemplo:

```
\frame{
  \begin{theorem}
  ...
  \end{theorem}
  \begin{overprint}
  \onslide<1>
  \hfill\hyperlinkframestartnext{\beamerskipbutton{Saltar prueba}}
  \onslide<2>
  \begin{proof}
  ...
  \end{proof}
  \end{overprint}
}
```

`\beamerreturnbutton`{*texto del botón*}

El símbolo que se dibuja para este botón normalmente es una cabeza de flecha apuntando hacia la izquierda. Se debe usar este botón si se quiere (al presionarlo) regresar al lugar previo al desvío.

Ejemplo:

```
\frame<1>[label=miteorema]
{
  \begin{theorem}
  ...
  \end{theorem}
  \begin{overprint}
  \onslide<1>
  \hfill\hyperlink{miteorema<2>}{\beamergotobutton{Ir a los detalles de la prueba}}
  \onslide<2>
  \begin{proof}
  ...
  \end{proof}
  \hfill\hyperlink{miteorema<1>}{\beamerreturnbutton{Volver}}
  \end{overprint}
}
\appendix
\againframe<2>{miteorema}
```

Para hacer que un botón sea “pulsable” se debe poner dentro de un comando como `\hyperlink`. El comando `\hyperlink` es un comando estandar del paquete `hyperref`. La clase BEAMER define un grupo entero de comandos de hiperliga que también se pueden usar.

`\hyperlink`<*especificación de cubierta*>{*nombre del objetivo*}{*texto de la liga*}<*especificación de cubierta*>

Sólo se puede dar una *especificación de cubierta*. El *texto de la liga* se edita como siempre. Si se pulsa sobre cualquier parte de este texto, se saltará a la diapositiva referida en el comando `\hypertarget` con parámetro *nombre del objetivo*. Si está presente una *especificación de cubierta*, la hiperliga (incluyendo el *texto de la liga*) se suprime completamente en las diapositivas no especificadas.

Los siguientes comandos tienen un objetivo predefinido, además se comportan exactamente como `\hyperlink`. En particular, todos aceptan una especificación de cubierta y también la aceptan al final, no sólo al principio.

`\hyperlinkslideprev`<*especificación de cubierta*>{*texto de la liga*}

Al pulsar sobre el texto se salta una diapositiva hacia atrás.

`\hyperlinkslidenext`<*especificación de cubierta*>{*texto de la liga*}

Al pulsar sobre el texto se salta una diapositiva hacia adelante.

`\hyperlinkframestart`<*especificación de cubierta*>{*texto de la liga*}

Al pulsar sobre el texto se salta a la primera diapositiva del marco actual.

`\hyperlinkframeend`<*especificación de cubierta*>{*texto de la liga*}

Al pulsar sobre el texto se salta a la última diapositiva del marco actual.

`\hyperlinkframestartnext`<*especificación de cubierta*>{*texto de la liga*}

Al pulsar sobre el texto se salta a la primera diapositiva del próximo marco.

`\hyperlinkframeendprev`<*especificación de cubierta*>{*texto de la liga*}

Al pulsar sobre el texto se salta a la primera diapositiva del marco anterior.

Los cuatro comandos anteriores también existen al reemplazar “frame” por “subsection” o por “section”.

`\hyperlinkpresentationstart`<⟨especificación de cubierta⟩>{⟨texto de la liga⟩}

Al pulsar sobre el texto se salta a la primera diapositiva de la presentación.

`\hyperlinkpresentationend`<⟨especificación de cubierta⟩>{⟨texto de la liga⟩}

Al pulsar sobre el texto se salta a la última diapositiva de la presentación. Esto *excluye* el apéndice.

`\hyperlinkappendixstart`<⟨especificación de cubierta⟩>{⟨texto de la liga⟩}

Al pulsar sobre el texto se salta a la primera diapositiva del apéndice. Si no hay apéndice, saltará a la última diapositiva del documento.

`\hyperlinkappendixend`<⟨especificación de cubierta⟩>{⟨texto de la liga⟩}

Al pulsar sobre el texto se salta a la última diapositiva del apéndice.

`\hyperlinkdocumentstart`<⟨especificación de cubierta⟩>{⟨texto de la liga⟩}

Al pulsar sobre el texto se salta a la primera diapositiva de la presentación.

`\hyperlinkdocumentend`<⟨especificación de cubierta⟩>{⟨texto de la liga⟩}

Al pulsar sobre el texto se salta a la última diapositiva de la presentación o si está presente un apéndice, a la última diapositiva del apéndice.

10.2. Repetir un marco en un punto posterior.

En ocasiones queremos mostrar en nuestra conferencia algunas diapositivas de un marco, pero además podemos querer mostrar algunas diapositivas “suplementarias” del marco sólo en el apéndice. En este caso es útil el comando `\againframe`.

`\againframe`<⟨especificación de cubierta⟩>[⟨especificación de cubierta predefinida⟩][⟨opciones⟩]{⟨nombre⟩}

**PRESEN-
TACIÓN** Reanuda un marco que se creó antes usando `\frame` con la opción `label=⟨nombre⟩`. Se debió usar esta opción, el poner “a mano” una etiqueta dentro de un marco no es suficiente. Se puede usar este comando para “continuar” un marco que fue interrumpido por otro marco. El efecto de este comando es llamar al comando `\frame` con la ⟨especificación de cubierta⟩ dada, la ⟨especificación de cubierta predefinida⟩ (si está presente) y las ⟨opciones⟩ (si están presentes) junto con el contenido del marco original.

Ejemplo:

```
\frame<1-2>[label=mimarco]
{
  \begin{itemize}
    \item<alert@1> Primer objeto
    \item<alert@2> Segundo objeto
    \item<alert@3> Tercer objeto
  \end{itemize}
}
\frame
{
  Algunas cosas que ahondan sobre el segundo objeto.
}
\againframe<3>{mimarco}
```


El efecto del código anterior es crear cuatro diapositivas. En las dos primeras, los puntos 1 y 2 están resaltados. La tercera diapositiva contiene el texto “Algunas cosas que ahondan sobre el segundo objeto”. La cuarta diapositiva es idéntica a las dos primeras, sólo que el tercer punto ahora está resaltado.

Ejemplo:

```
\frame<1>[label=Cantor]
{
  \frametitle{Teorema Principal}

  \begin{theorem}
     $\alpha < 2^\alpha$  para todos los ordinales  $\sim \alpha$  .
  \end{theorem}

  \begin{overprint}
    \onslide<1>
      \hyperlink{Cantor<2>}{\beamergotobutton{Detalles de la Prueba}}

    \onslide<2->
      % Esto se muestra en el apéndice, donde se reanuda el marco.
      \begin{proof}
        Como lo probamos Cantor, ...
      \end{proof}

      \hfill\hyperlink{Cantor<1>}{\beamerreturnbutton{Volver}}
    \end{overprint}
  }

  ...
  \appendix

  \againframe<2>{Cantor}
```

En este ejemplo, los detalles de la prueba se aplazan hasta una diapositiva en el apéndice. Se establecen hiperligas así que uno puede saltar a la prueba y luego volver.

ARTÍCULO Este comando se ignora en este modo.

10.3. Añadiendo una ampliación anticipada

Es necesario anticipar una ampliación, cuando se tiene un gráfico muy complicado al que no se está dispuesto a simplificar dado que, de hecho, todos los detalles complejos merecen una explicación. En este caso usamos el comando `\framezoom`, que permite especificar que al pulsar sobre una cierta área de un marco, ésta debe ser ampliada a un plano general. Entonces se pueden explicar los detalles. Al pulsar sobre el gráfico ampliado se vuelve al gráfico original.

`\framezoom` \langle *especificación de cubierta del botón* \rangle \langle *especificación de cubierta de la ampliación* \rangle \langle *opciones* \rangle \langle *superior_izq x* \rangle \langle *superior_izq y* \rangle \langle *ancho del área a ampliar* \rangle \langle *altura del área a ampliar* \rangle

Este comando se debe dar en algún lugar al principio de un marco. Si se da, pasarán dos cosas diferentes dependiendo de si la *especificación de cubierta del botón* aplica a la diapositiva actual del marco o si aplica a la *especificación de cubierta de la ampliación*. Estas especificaciones de cubiertas no se deben traslapar.

Si aplica la *especificación de cubierta del botón* dentro del marco se crea un área pulsable. El tamaño de esta área está dada por *ancho del área a ampliar* y *altura del área a ampliar*, que deben ser dos

dimensiones normales de $\text{T}_{\text{E}}\text{X}$ (como 1cm o 20pt). La esquina superior izquierda de esta área esta dada por $\langle superior_izq\ x \rangle$ y $\langle superior_izq\ y \rangle$, que también deben ser dimensiones de $\text{T}_{\text{E}}\text{X}$. Son medidas *relativas al lugar donde iría el primer texto normal de un marco*. Así, $(0\text{pt},0\text{pt})$ es la ubicación del principio del texto normal (que excluye el título de página y también el título del marco).

Por defecto el botón es pulsable pero no se indica de alguna manera especial. Se puede dibujar un borde alrededor del botón usando la siguiente *opción*:

- **border**= $\langle ancho\ en\ pixeles \rangle$ dibuja un borde alrededor del área de botón especificada. El valor por defecto de la anchura es de 1 pixel. El color de este botón es `linkbordercolor` de `hyperref`. Por defecto BEAMER fija este color en un 50% de gris. Para cambiarlo se puede usar el comando `\hypersetup{linkbordercolor={\rojo} \langle verde \rangle \langle azul \rangle}`, donde $\langle rojo \rangle$, $\langle verde \rangle$ y $\langle azul \rangle$ son valores entre 0 y 1.

Cuando se pulsa el botón creado de este modo, la aplicación visor hipersalta al primer marco especificado en $\langle especificación\ de\ cubierta\ de\ la\ ampliación \rangle$. Para las diapositivas a las que aplica esta especificación de cubierta pasa lo siguiente:

Exactamente la misma área especificada como antes, se “cambia a un plano general” para llenar el área total correspondiente al texto normal del marco. Todo lo demás, incluidas las barras laterales, los encabezados y pies de página e incluso el título del marco, mantiene su tamaño normal. La ampliación se ejecuta de tal manera que se muestra completamente el área total especificada. La proporción del aspecto debe ser correcto; sí la proporción del aspecto de esta área y la proporción del aspecto del área de texto disponible no coinciden entonces el área ampliada posiblemente mostrará algo más que sólo el área especificada. Detrás del área de texto entero (qué contiene el área ampliada) se pone un gran botón invisible de “Volver”. Así que al pulsar cualquier parte del área de texto se volverá a la imagen original (sin ampliar).

En un mismo marco se pueden especificar varias áreas a ampliar. En este caso se deben especificar distintas $\langle especificación\ de\ cubierta\ de\ la\ ampliación \rangle$ pero se puede especificar la misma $\langle especificación\ de\ cubierta\ del\ botón \rangle$. No se pueden anidar ampliaciones en el sentido de que no se puede tener un botón de ampliación sobre una diapositiva que está en alguna $\langle especificación\ de\ cubierta\ de\ la\ ampliación \rangle$. Sin embargo, se pueden traslapar e incluso anidar la $\langle especificación\ de\ cubierta\ del\ botón \rangle$. Cuando se pulsa sobre un área que pertenece a varios botones, el que se dió al último “ganará” (de ahí que debe ser el más pequeño).

Si no se desea mostrar el título del marco en una diapositiva ampliada, se puede agregar una especificación de cubierta al comando `\frametitle` que simplemente suprima el título de la diapositiva. Al usar la opción `plain` se puede tener la diapositiva ampliada llenando la página entera.

Ejemplo: Un caso simple

```
\begin{frame}
  \frametitle{Un gráfico complicado}
  \framezoom<1><2>(0cm,0cm)(2cm,1.5cm)
  \framezoom<1><3>(1cm,3cm)(2cm,1.5cm)
  \framezoom<1><4>(3cm,2cm)(3cm,2cm)
  \pgfimage[height=8cm]{nombearchivográficomcomplicado}
\end{frame}
```

Ejemplo: Un caso más complicado en el que las partes ampliadas llenan completamente los marcos.

```
\begin{frame}<1>[label=zooms]
  \frametitle<1>{Un gráfico complicado}
  \framezoom<1><2>[border](0cm,0cm)(2cm,1.5cm)
  \framezoom<1><3>[border](1cm,3cm)(2cm,1.5cm)
  \framezoom<1><4>[border](3cm,2cm)(3cm,2cm)
  \pgfimage[height=8cm]{nombearchivográficomcomplicado}
\end{frame}
\againframe<2->[plain]{zooms}
```

11. Estructurando una presentación: Estructura Local

L^AT_EX proporciona varios comandos para estructurar texto “localmente”, por ejemplo, vía el ambiente `itemize`. Estos ambientes también están disponibles en la clase BEAMER, aunque su apariencia ha cambiado ligeramente. Además, la clase BEAMER también define nuevos comandos y ambientes que pueden ayudar a estructurar el texto.

11.1. Listados, enumeraciones y descripciones

Hay tres ambientes predefinidos para crear listas: `enumerate`, `itemize` y `description`. Los dos primeros soportan tres niveles de anidación, pero al anidar hasta un tercer nivel se crean diapositivas totalmente ilegibles.

El comando `\item` es consciente de especificación de cubierta. Si se proporciona una especificación de cubierta, el objeto sólo se muestra en las diapositivas especificadas. Si el comando `\item` va a tomar un argumento opcional y una especificación de cubierta, la especificación de cubierta puede ir primero como en `\item<1>[Cat]` o al final como en `\item[Cat]<1>`.

```
\begin{frame}
  Hay tres puntos importantes:
  \begin{enumerate}
    \item<1-> Un Primero,
    \item<2-> Un Segundo con un mont'\{o\}n de subpuntos,
      \begin{itemize}
        \item Primer subpunto. (¡S'\{o\}lo se muestra a partir de la segunda diapositiva!).
        \item<3-> Se agrega un segundo subpunto en la tercera diapositiva.
        \item<4-> Se agrega un tercer subpunto en la cuarta diapositiva.
      \end{itemize}
    \item<5-> y un Tercero.
  \end{enumerate}
\end{frame}
```

```
\begin{itemize}[<<especificación de cubierta predefinida>>]
  <Contenido del ambiente>
\end{itemize}
```

Despliega una lista de objetos que no tienen un orden especial. Dentro del ambiente se usa un comando `\item` por cada tópico. Si se da el parámetro opcional *<especificación de cubierta predefinida>*, en cada ocurrencia de un comando `\item` que no tiene una especificación de cubierta ligada a él, se usa la *<especificación de cubierta predefinida>*. Al establecer que esta especificación sea una especificación de cubierta incremental, se puede implementar por ejemplo que los objetos se descubran por partes. Los subambientes heredan la *<especificación de cubierta predefinida>* Naturalmente, en un subambiente se puede restablecer localmente poniéndolo como `<1->`.

Ejemplo:

```
\begin{itemize}
  \item Esto es importante.
  \item Esto tambi'\{e\}n es importante.
\end{itemize}
```

Ejemplo:

```
\begin{itemize}[<+>]
  \item Esto se muestra a partir de la primera diapositiva.
  \item Esto se muestra a partir de la segunda diapositiva.
  \item Esto se muestra a partir de la tercera diapositiva.
```

```

\item<1-> Esto se muestra a partir de la primera diapositiva.
\item Esto se muestra a partir de la cuarta diapositiva.
\end{itemize}

```

Ejemplo:

```

\begin{itemize}[<+| alert@>]
\item Esto se muestra a partir de la primera diapositiva
y resaltada en la primera diapositiva.
\item Esto se muestra a partir de la segunda diapositiva
y resaltada en la segunda diapositiva.
\item Esto se muestra a partir de la tercera diapositiva
y resaltada en la tercera diapositiva.
\end{itemize}

```

Ejemplo:

```

\newenvironment{milistadoporpartes}{\begin{itemize}[<+| alert@>]}\end{itemize}}

```

La apariencia de una lista `itemize` está controlada por varias plantillas. La primera plantilla se refiere a la manera en que se edita el pequeño marcador que introduce cada objeto:

Plantilla-Beamer Padre `itemize items`

Esta es una plantilla padre, cuyos productos son `itemize item`, `itemize subitem` e `itemize subsubitem`. Esto significa que si se usa en esta plantilla el comando `\setbeamertemplate`, el comando es llamado (con los mismos argumentos) por todos los productos.

Las siguientes opciones de la plantilla están predefinidas:

- [**default**] El marcador de objeto predefinido es un triángulo pequeño coloreado con el primer plano del color-BEAMER `itemize item` (o para subitems, `itemize subitem` etc.). Notamos que bajo ciertas circunstancias estos colores cambian automáticamente, como cuando están dentro de un bloque ejemplo o dentro de un ambiente `alertenv`.
- [**triangle**] Alias para el valor por defecto.
- [**circle**] Usa círculos (o puntos) pequeños como marcadores de objeto.
- [**square**] Usa cuadrados pequeños como marcadores de objeto.
- [**ball**] Usa bolas pequeñas como marcadores de objeto.

Plantilla-/Color-/Fuente-Beamer `itemize item`

Color/fuente padre: `item`

Esta plantilla controla la manera cómo se edita el marcador delante de un objeto en el primer nivel de anidación. Checar la plantilla `itemize items` para ver las *<opciones>* que se pueden dar.

Cuando se inserta la plantilla, se instala la fuente- y el color-BEAMER `itemize item`. Normalmente la fuente es ignorada por la plantilla cuando se dibuja algún símbolo especial, pero la fuente puede ser importante si se da un argumento opcional al comando `\item` como en `\item[Primero]`. La fuente y color de esta plantilla se heredan de la fuente y color `item`, que se explica al final de esta sección.

Plantilla-/Color-/Fuente-Beamer `itemize subitem`

Color/fuente padre: `subitem`

Como `itemize item` sólo que para los objetos en un segundo nivel de anidación. Cualquier objeto de un listado dentro de una enumeración se considera un objeto en un segundo nivel.

Plantilla-/Color-/Fuente-Beamer `itemize subsubitem`

Color/fuente padre: `subsubitem`

Como `itemize item` sólo que para los objetos en un tercer nivel de anidación.

```
\begin{enumerate}[<especificación de cubierta predefinida>][<mini plantilla>]
<Contenido del ambiente>
\end{enumerate}
```

Se usa para desplegar una lista de elementos que están ordenados. Dentro del ambiente se usa el comando `\item` para cada tópico. Por defecto antes de cada objeto se imprimen de manera creciente números arábigos seguidos por un punto (como en “1.” y “2.”). Esto se puede cambiar especificando una plantilla diferente. El primer argumento opcional *<especificación de cubierta predefinida>* tiene el mismo efecto que en el ambiente `itemize`. Se detecta por el signo de apertura `<` en la *<especificación de cubierta predefinida>*. Así, si hay sólo un argumento opcional y éste no empieza con `<` entonces se considera que es una *<mini plantilla>*.

La sintaxis de la *<mini plantilla>* es igual que la sintaxis de las mini plantillas en el paquete `enumerate` (no se necesita incluir el paquete `enumerate`, se hace automáticamente). Por así decirlo, el texto de la *<mini plantilla>* se imprime antes de cada objeto pero cualquier ocurrencia de un 1 en la mini plantilla se reemplaza por el número del objeto actual, cualquier ocurrencia de la letra A se reemplaza por la i-ésima letra del alfabeto (en mayúsculas) para el i-ésimo objeto, y las letras a, i y I se reemplazan por las correspondientes letras minúsculas, letras minúsculas romanas y letras mayúsculas romanas respectivamente. Así la mini plantilla (i) producirá los objetos (i), (ii), (iii), y así sucesivamente. La mini plantilla A.) producirá los objetos A.), B.), C.) y así sucesivamente. Para más detalles de las posibles mini plantillas, ver la documentación del paquete `enumerate`. Notamos que también hay una plantilla que controla la apariencia de la mini plantilla.

Ejemplo:

```
\begin{enumerate}
\item Esto es importante.
\item Esto tambi\`{e}n es importante.
\end{enumerate}
```

```
\begin{enumerate}[(i)]
\item Primer punto romano.
\item Segundo punto romano.
\end{enumerate}
```

```
\begin{enumerate}[<+>][(i)]
\item Primer punto romano.
\item Segundo punto romano, descubierto en la segunda diapositiva.
\end{enumerate}
```

ARTÍCULO Para usar la *<mini plantilla>* se tiene que incluir el paquete `enumerate`.

Plantilla-Beamer Padre `enumerate items`

Similar a `itemize items`, esta es una plantilla padre cuyos productos son `enumerate items`, `enumerate subitem`, `enumerate subsubitem` y `enumerate mini template`. Estas plantillas controlan la manera en que se edita el texto (número) de una enumeración.

Las siguientes opciones de plantilla están predefinidas:

- `[default]` El marcador de enumeración predefinido usa los esquemas 1., 2. para el primer nivel, 1.1, 1.2 para el segundo nivel y 1.1.1, 1.1.2 para el tercer nivel.

- `[circle]` Coloca los números dentro de círculos pequeños. Los colores se toman de `item projected` o `subitem projected` o `subsubitem projected`.
- `[square]` Coloca los números sobre cuadrados pequeños.
- `[ball]` “Proyecta” los números sobre bolas pequeñas.

Plantilla-/Color-/Fuente-Beamer `enumerate item`

Esta plantilla controla la manera cómo se edita el número delante de un objeto en el primer nivel de anidación de enumeraciones. Así una enumeración dentro de un listado está en un primer nivel de enumeración (pero usa el segundo nivel `itemize/enumerate body`). Cuando se inserta la plantilla, se instala la fuente y el color-BEAMER `enumerate item`.

El siguiente comando es útil para esta plantilla:

- `\insertenumlabel` Inserta el número (arábigo) actual de la enumeración de más alto nivel. Esta inserción también está disponible en las siguientes dos plantillas.

Plantilla-/Color-/Fuente-Beamer `enumerate subitem`

Como `enumerate item`, sólo que para objetos de segundo nivel de anidación.

- `\insertsubenumlabel` inserta el número (arábigo) actual de la enumeración de segundo nivel.

Ejemplo:

```
\setbeamertemplate{enumerate subitem}{\insertenumlabel-\insertsubenumlabel}
```

Plantilla-/Color-/Fuente-Beamer `enumerate subsubitem`

Como `enumerate item`, sólo que para los objetos de tercer nivel de anidación.

- `\insertsubsubenumlabel` inserta el número (arábigo) actual de la enumeración de tercer nivel.

Plantilla-/Color-/Fuente-Beamer `enumerate mini template`

Esta plantilla se usa para editar el número que surge de una mini plantilla.

- `\insertenumlabel` inserta el número actual representado según la mini plantilla. Por ejemplo, si la `\mini plantilla` es (i) y se usa `\insertenumlabel` en el cuarto objeto se producirá (iv).

Las siguientes plantillas controlan la manera en que se edita el *cuerpo* de un listado o una enumeración.

Plantilla-Beamer `itemize/enumerate body begin`

Esta plantilla se inserta al principio de un primer nivel de anidación en un ambiente `itemize` o `enumerate`. Antes de insertar la plantilla se usa la fuente y color-BEAMER `itemize/enumerate body`.

Plantilla-Beamer `itemize/enumerate body end`

Esta plantilla se inserta al final de un primer nivel en un ambiente `itemize` o `enumerate`.

Existen las correspondientes plantillas `itemize/enumerate subbody begin(end)` para el segundo y tercer nivel del ambiente `itemize` o `enumerate`.

Plantilla-Beamer Padre `items`

Es una plantilla padre de `itemize items` y `enumerate items`.

Ejemplo: `\setbeamertemplate{items}[circle]` causa que todos los objetos dentro de un ambiente `itemize` o `enumerate` se conviertan en círculos (del tamaño, color y fuente apropiados).

```
\begin{description}[<(especificación de cubierta predefinida)>][<texto largo>]
<Contenido del ambiente>
```

```
\end{description}
```

Como `itemize` pero se usa para desplegar una lista que explica o define etiquetas. El ancho del *(texto largo)* se usa para establecer la indentación. Normalmente uno elige la etiqueta mas ancha en la descripción y se copia aquí. Si no se da este argumento, se usa el ancho predefinido, el cual se puede cambiar usando `\setbeamersize` con el argumento `description width=<ancho>`.

Como para `enumerate`, la *(especificación de cubierta predefinida)* se detecta por el símbolo de apertura `<`. El efecto es el mismo que en `enumerate` e `itemize`.

Ejemplo:

```
\begin{description}
\item[Leon] Rey de la sabana.
\item[Tigre] Rey de la jungla.
\end{description}
```

Ejemplo:

```
\begin{description}[<+>][etiqueta más larga]
\item[corto] Algo de texto.
\item [etiqueta m\ '{a}s larga] Algo de texto.
\item[etiqueta larga] Algo de texto.
\end{description}
```

Plantilla-/Color-/Fuente-Beamer `description item`

Esta plantilla se usa para editar los objetos de una descripción. Cuando se llama esta plantilla, se instalan la fuente y el color-BEAMER `description item`.

Las siguientes opciones de las plantillas están predefinidas:

- `[default]` Por defecto el texto del objeto de una descripción simplemente se inserta sin ninguna modificación.

La principal inserción, útil dentro de esta plantilla es:

- `\insertdescriptionitem` inserta el texto del actual objeto de la descripción.

Para simplificar el cambio de color o de fuente de los objetos, los diferentes tipos de objetos heredan su forma o sólo usan los siguientes colores y fuentes-BEAMER generales:

Color/Fuente-Beamer `item`

Color padre: `local structure`

Fuente padre: `structure`

Este color/fuente sirve como padre para los objetos individuales en los ambientes `itemize` y `enumerate`, pero también para los objetos en la tabla de contenidos. Dado que su color padre es `structure local`, un cambio de aquel causará el correspondiente cambio de color de los objetos.

Color/Fuente-Beamer `item projected`

Color/fuente padre: `item`

Son una “versión” especial del color y fuente `item`, que se deben usar por plantillas que representan objetos con texto (como en una enumeración) y que “proyectan” este texto sobre algo como una bola o un cuadrado o cualquier otra cosa. Mientras el color de la plantilla normal `item` tiene un fondo transparente, la plantilla `item projected` tiene un fondo coloreado y un primer plano blanco.

Color/Fuente-Beamer `subitem`

Color/fuente padre: `item`

Lo mismo que `item` pero para subobjetos, esto es, para objetos en el segundo nivel de indentación.

Color/Fuente-Beamer `subitem projected`

Color/fuente padre: `item projected`

Lo mismo que `item projected` pero para subobjetos, esto es, para objetos en el segundo nivel de indentación.

Color/Fuente-Beamer `subsubitem`

Color/fuente padre: `subitem`

Color/Fuente-Beamer `subsubitem projected`

Color/fuente padre: `subitem projected`

11.2. Resaltamiento

Para resaltar textos la clase BEAMER predefine comandos y ambientes. Al usar estos comandos se facilita el cambio de apariencia de un documento con sólo cambiar el tema.

`\structure` \langle *especificación de cubierta* \rangle $\{$ *texto* $\}$

El texto dado se marca como parte de la estructura, es decir, se supone que ayuda al público a ver la estructura de la presentación. Si está presente la \langle *especificación de cubierta* \rangle el comando sólo tiene efecto en las diapositivas indicadas. Internamente, este comando sólo pone el texto dentro de un ambiente `structureenv`.

Ejemplo: `\structure{Encabezado de Párrafo.}`

ARTÍCULO El texto de la estructura se edita en negritas. Esto se puede cambiar modificando las plantillas.

Color/Fuente-Beamer `structure`

Este color/fuente se usa cuando se edita texto estructurado, pero también se usa ampliamente como base para muchos otros colores incluyendo los encabezados de bloques, botones de objetos y títulos. En la mayoría de los temas de color, los colores para los elementos de navegación en el encabezado o el pie de página se derivan del color del primer plano de `structure`. Al cambiar el color de la estructura se puede cambiar fácilmente el “color básico” de la presentación, fuera del color de texto normal. Ver el color relacionado `structure local` y la fuente relacionada `tiny structure`.

Dentro del comando `\structure` se ignora el color de fondo, pero esto no necesariamente se cumple para elementos que heredan su color de `structure`. No existe la plantilla `structure`, se debe usar `structure begin` y `structure end`.

Color-Beamer `local structure`

Este color se debe usar para editar elementos estructurales que cambian su color de acuerdo al “ambiente local”. Por ejemplo, el color de un “botón” de objeto en un ambiente `itemize` cambia según las circunstancias. Si se usa dentro de un bloque ejemplo debe tener el color `example text`; si actualmente esta “resaltado” debe tener el color `alerted text`. Este color establecerá a través de ciertos ambientes que tenga el color que se debe usar para editar cosas como botones de objeto. Dado que el color que se usa para objetos, `item`, hereda por defecto este color, los objetos automáticamente cambian su color según la situación actual.

Si se escribe un ambiente propio en el cual los botones de objeto son elementos estructurales similares que deben tener un color diferente, se debe cambiar el color `local structure` dentro de estos ambientes.

Fuente-Beamer `tiny structure`

Esta fuente especial se usa para texto estructural “diminuto”. Básicamente, esta fuente se debe usar siempre que un elemento estructural usa una fuente diminuta. La idea es que, a menudo, la versión tiny de la fuente `structure` no es conveniente. Por ejemplo, a menudo es necesario usar su versión en negritas. También, uno a veces quiere tener texto estructural con serif smallcaps pero todavía retener el texto estructural diminuto con sans-serif.

```
\begin{structureenv}<<especificación de cubierta>>
<contenido del ambiente>
\end{structureenv}
```

Versión ambiente del comando `\structure`.

Plantilla-Beamer `structure begin`

Este texto se inserta al principio de un ambiente `structureenv`.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]`

ARTÍCULO El texto se edita en negritas.

Plantilla-Beamer `structure end`

Este texto se inserta al final de un ambiente `structureenv`.

```
\alert<<especificación de cubierta>>{<texto resaltado>}
```

Normalmente se resalta el texto dado coloreándolo en rojo. Si está presente la *<especificación de cubierta>*, el comando sólo tiene efecto en las diapositivas indicadas. Internamente, este comando pone el *texto resaltado* dentro de un ambiente `alertenv`.

Ejemplo: Esto es `\alert{importante}`.

ARTÍCULO El texto resaltado se edita como texto inclinado (`emphasize`). Esto se puede cambiar modificando las plantillas.

Color/Fuente-Beamer `alerted text`

Este color/fuente se usa cuando se edita texto resaltado. Actualmente se ignora el fondo. No existe plantilla `alerted text`, más bien existen las plantillas `alerted text begin` y `alerted text end` que se insertan antes y después del texto resaltado.

```
\begin{alertenv}<<especificación de cubierta>>
<Contenido del ambiente>
\end{alertenv}
```

Versión ambiente del comando `\alert`.

Plantilla-Beamer `alerted text begin`

Este texto se inserta al principio de un ambiente `alertenv`.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]`

PRESEN- TACIÓN Esta plantilla cambia el color local `structure` a `alerted text`. Esto causa que cosas como botones u objetos sean coloreados del mismo color que el texto resaltado, lo que a menudo es agradable visualmente. Ver también el comando `\structure`.

ARTÍCULO El texto se edita inclinado (`emphasize`)

Plantilla-Beamer `alerted text end`

Este texto se inserta al final de un ambiente `alertenv`.

11.3. Ambientes de bloque

La clase BEAMER predefine un ambiente para editar un “bloque” de texto que tiene un encabezado. La apariencia de bloques se puede cambiar fácilmente usando la siguiente plantilla:

Plantilla-Beamer Padre `blocks`

Al cambiar esta plantilla padre, cambian las plantillas de bloques normales, bloques resaltados y bloques ejemplo.

Example: `\setbeamertemplate{blocks}[default]`

Example: `\setbeamertemplate{blocks}[rounded][shadow=true]`

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` El entorno predefinido edita el título del bloque sobre su propia línea. Si se especifica un fondo para el `block title` o para el `block body`, este color de fondo se usa como fondo del título o del cuerpo respectivamente. Para los bloques resaltados y de ejemplo, se usan en cambio los colores y fuentes-BEAMER correspondientes.
- `[rounded]` [`shadow=true`] Hace los bloques “redondeados”. Esto significa que se “redondean” las esquinas del fondo del bloque. Si se da la opción `shadow=true`, se dibuja una “sombra” detrás del bloque.

```
\begin{block}<especificación de acción>{<título del bloque>}<especificación de acción>  
<Contenido del ambiente>  
\end{block}
```

Sólo se puede dar una *<especificación de acción>*. Inserta un bloque, como una definición o un teorema, con el *<título del bloque>*. Si está presente la *<especificación de acción>*, las acciones dadas se asumen en las diapositivas indicadas. En el ejemplo, la definición se muestra sólo a partir de la diapositiva 3.

Ejemplo:

```
\begin{block}<3->{Definición}  
  Un \alert{conjunto} consiste de elementos.  
\end{block}
```

ARTÍCULO El nombre del bloque se edita en negritas.

Plantilla-Beamer `block begin`

Esta plantilla se inserta al principio de un bloque antes del *<Contenido del ambiente>*. Dentro de esta plantilla se puede acceder al título del bloque vía la siguiente inserción:

- `\insertblocktitle` Inserta en la plantilla el *<título del bloque>*.

Cuando empieza la plantilla no se instala ningún color o fuente especial (por razones algo complicadas). Así, la plantilla misma debe instalar los colores y las fuentes adecuadas para el título y el cuerpo.

Plantilla-Beamer `block end`

Esta plantilla se inserta al final de un bloque.

Color/Fuente-Beamer `block title`

Este color/fuente-BEAMER se debe usar para editar el título del bloque. Dado que el color ni la fuente se crean automáticamente, la plantilla misma `block begin` lo debe hacer.

La plantilla de bloque predefinida (así como la de la versión redondeada) respeta el fondo de este color.

Color/Fuente-Beamer `block body`

Este color/fuente-BEAMER se debe usar para editar el cuerpo del bloque, es decir el *Contenido del ambiente*. Como para `block title`, el color y la fuente se deben establecer por medio de la plantilla `block begin`.

```
\begin{alertblock}<(especificación de acción)>{(título del bloque)}<(especificación de acción)>
<Contenido del ambiente>
\end{alertblock}
```

Inserta un bloque cuyo título está resaltado. Se comporta como el ambiente `block`.

Ejemplo:

```
\begin{alertblock}{Teorema Erróneo}
  $1=2$.
\end{alertblock}
```

ARTÍCULO El nombre del bloque se edita en negritas inclinadas (`emphasize`).

Plantilla-Beamer `block alerted begin`

Aplica lo mismo que para bloques normales.

Plantilla-Beamer `block alerted end`

Aplica lo mismo que para bloques normales.

Color/Fuente-Beamer `block title alerted`

Aplica lo mismo que para bloques normales

Color/Fuente-Beamer `block body alerted`

Aplica lo mismo que para bloques normales.

```
\begin{exampleblock}<(especificación de acción)>{(título del bloque)}<(especificación de cubierta)>
<Contenido del ambiente>
\end{exampleblock}
```

Inserta un bloque que se supone representa un ejemplo. Se comporta como el ambiente `block`.

En el siguiente ejemplo el bloque se suprime completamente en la primera diapositiva (incluso no ocupa espacio alguno).

```
\begin{exampleblock}{Ejemplo}<only@2->
  El conjunto  $\{1,2,3,5\}$  tiene cuatro elementos.
\end{exampleblock}
```

ARTÍCULO El nombre del bloque se edita en itálicas.

Plantilla-Beamer `block example begin`

Aplica lo mismo que para bloques normales.

Plantilla-Beamer `block example end`

Aplica lo mismo que para bloques normales.

Color/Fuente-Beamer `block title example`

Aplica lo mismo que para bloques normales.

Color/Fuente-Beamer `block body example`

Aplica lo mismo que para bloques normales.

11.4. Ambientes tipo teorema

La clase BEAMER predefine varios ambientes, como `theorem` o `definition` o `proof`, que se pueden usar para editar cosas como teoremas, definiciones o pruebas. La lista completa es la siguiente: `theorem`, `corollary`, `definition`, `definitions`, `fact`, `example` y `examples`.

Aquí esta un ejemplo típico de como usarlos:

```
\begin{frame}
  \frametitle{Un Teorema sobre Conjuntos Infinitos}

  \begin{theorem}<1->
    Existe un conjunto infinito.
  \end{theorem}

  \begin{proof}<2->
    Esto se sigue del Axioma del Infinito.
  \end{proof}

  \begin{example}<3->[N]u}meros Naturales]
    El conjunto de n)u}meros naturales es infinito.
  \end{example}
\end{frame}

\begin{theorem}<especificación de acción>[<texto adicional>]<especificación de acción>
<Contenido del ambiente>
\end{theorem}
```

Inserta un teorema. Sólo se puede dar una *<especificación de acción>*. Si está presente, el *<texto adicional>* se muestra después de la palabra “Theorem” entre paréntesis (aunque la plantilla puede cambiar esto). Las plantillas `theorem begin` y `theorem end` controlan la apariencia del teorema. Cada teorema se pone en un ambiente `block`, así que también aplican las plantillas para bloques.

El estilo `theorem` (un concepto de `amsthm`) que se usa para este ambiente es `plain`. En este estilo, el cuerpo de un teorema se debe editar en itálicas. Además, la cabeza del teorema se debe editar en negritas pero la plantilla normalmente anula esto.

Si se da la opción `envcountsect`, ya sea como opción de clase en uno de los modos `presentation` o como una opción para el paquete `beamerarticle` en modo `article`, entonces la numeración de los teoremas es local a cada sección, con el número de sección prefijando el número del teorema; en caso contrario, se numeran consecutivamente a lo largo de la presentación o artículo. Se recomienda usar esta opción en modo `article`.

Por defecto, en los modos de `presentación` no se muestra ninguna numeración de teoremas.

Ejemplo:

```
\begin{theorem}[Kummer, 1992]
\end{theorem}

\begin{theorem}<2->[Tantau, 2002]
  $$
\end{theorem}
```

Los ambientes `corollary`, `fact` y `lemma` funcionan exactamente de la misma manera.

```
\documentclass[envcountsect]{beamer}
```

Causa que los teoremas, definiciones, etc. se numeren localmente a cada sección. Por lo tanto, el primer teorema de la segunda sección sería el Teorema 2.1 (asumiendo que en la sección no hay antes definiciones, lemas o corolarios).

```
\begin{definition}<especificación de acción>[<texto adicional>]<especificación de acción>
<contenido del ambiente>
\end{definition}
```

Funciona como el ambiente `theorem` excepto que se usa el estilo de teorema `definition`. En este estilo el cuerpo de una definición se edita con una fuente vertical.

El ambiente `definitions` se comporta exactamente de la misma manera.

```
\begin{example}<especificación de acción>[<texto adicional>]<especificación de acción>
<contenido del ambiente>
\end{example}
```

Funciona como el ambiente `theorem` excepto que se usa el estilo de teorema `example`. Al usar este estilo de teorema, un efecto lateral es que el `<contenido del ambiente>` se pone en un `exampleblock` en lugar de un `block`.

El ambiente `examples` se comporta exactamente de la misma manera.

**PRESEN-
TACIÓN** La plantilla predefinida para la edición de teoremas suprime la numeración de teoremas, aun cuando este número está “disponible” para edición (el cual por defecto está en todos los ambientes predefinidos; pero si definimos un ambiente propio que usa `\newtheorem*`, ningún número estará disponible).

ARTÍCULO En este modo los teoremas se numeran automáticamente. Indicando la opción de clase `envcountsect`, los teoremas se numeran localmente a cada sección; usualmente es una idea buena, excepto para artículos muy cortos.

```
\begin{proof}<especificación de acción>[<nombre de la prueba>]<especificación de acción>
<contenido del ambiente>
\end{proof}
```

Edita una prueba. Si se da `<nombre de la prueba>` opcional, ésta reemplaza completamente la palabra “Proof”. Esto es diferente a la edición de los teoremas normales donde el argumento opcional se muestra entre corchetes.

Al final del teorema se muestra un símbolo `\qed`. Normalmente al final de un ambiente `proof` se inserta automáticamente el símbolo `QED` dentro de una fea línea vacía. Para evitar esto se puede poner `\qedhere` en la última línea antes de finalizar el ambiente (exactamente como sucede en `amsthm`). El símbolo predefinido `\qed` es un cuadrado abierto. Para suprimir completamente el símbolo hay que escribir `\def\qedsymbol{}` en el preámbulo. Para conseguir un rectángulo cerrado hay que indicar

```
\setbeamertemplate{qed symbol}{\vrule width1.5ex height1.5ex depth0pt}
```

Si se usa `babel` y un idioma diferente al inglés, el texto “Proof” se reemplaza por cualquier cosa que sea apropiada en el idioma seleccionado.

Ejemplo:

```
\begin{proof}<2->[Bosquejo de la prueba]
Supongamos ...
```

```
\end{proof}
```

Plantilla-/Color-/Fuente-Beamer `qed symbol`

El símbolo se muestra al final de cada prueba.

Se pueden definir nuevos ambientes usando el siguiente comando:

```
\newtheorem*{\langle nombre del ambiente \rangle}[\langle numerado igual que \rangle]{\langle texto cabeza \rangle}[\langle número dentro de \rangle]
```

Este comando se usa exactamente de la misma manera que en el paquete `amsthm` (de hecho, es el comando de ese paquete). La única diferencia es que, en BEAMER, los ambientes declarados que usan este comando son conscientes de especificación de cubierta y que se editan de acuerdo a las plantillas de BEAMER.

ARTÍCULO En este modo los ambientes declarados que usan este comando también son conscientes de especificación de cubierta.

Example: `\newtheorem{observation}[theorem]{Observación}`

También se puede usar el comando `\newtheoremstyle` de `amsthm` para definir nuevos estilos de teorema. Notamos que la plantilla predefinida para teoremas ignora cualquier fuente del encabezado establecida pero respeta la fuente del cuerpo establecida.

Si se desea definir de manera diferente los ambientes como `theorem` (por ejemplo, tenerlos numerados dentro de cada subsección), se puede usar la siguiente opción de clase para desactivar la definición de los ambientes predefinidos:

```
\documentclass[notheorems]{beamer}
```

Desactiva la definición de bloques predefinidos como `theorem`, pero aún carga `amsthm` y hace los teoremas conscientes de especificación de cubierta. Esta opción también está disponible como opción de clase para `beamerarticle` y tiene el mismo efecto.

ARTÍCULO En este modo el paquete `amsthm` a veces choca con la clase del documento. En este caso se puede usar la siguiente opción, que una vez más está disponible como una opción de clase para BEAMER y como opción de paquete para `beamerarticle`, para desactivar totalmente la carga de `amsthm`.

```
\documentclass[noamsthm]{beamer}
```

No carga `amsthm` ni tampoco `amsmath`. Ambientes como `theorem` o `proof` no estarán disponibles.

Plantilla-Beamer Padre `theorems`

Esta plantilla es padre de `theorem begin` y `theorem end` (ver la primera parte para una discusión detallada de cómo se establecen las plantillas de teorema).

Example: `\setbeamertemplate{theorems}[numbered]`

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` Por defecto los teoremas se editan como sigue: se respeta la especificación de la fuente para el cuerpo y se ignora la especificación de la fuente para el encabezado. No se imprime ningún número de teorema.
- `[normal font]` Como en `default`, excepto que se ignoran todas las especificaciones de la fuente para el cuerpo. Las fuentes que se usan normalmente son las que se usan para los bloques.
- `[numbered]` Esta opción es semejante a la predefinida excepto que se imprime el número del teorema para ambientes que están numerados.

- **[ams style]** Esto provoca que se pongan los teoremas en un `block` o `exampleblock` para que se editen de otro modo, como se hace normalmente en `amsthm`. Así, las fuentes del encabezado y del cuerpo dependen del ajuste del teorema a editar, los teoremas se numeran.

Plantilla-Beamer `theorem begin`

Siempre que se va a editar un ambiente que se declaró usando el comando `\newtheorem`, al principio se inserta esta plantilla y al final la plantilla `theorem end`. Si hay una especificación de cubierta cuando se usa un ambiente como `theorem`, esta especificación de cubierta seguirá a la *plantilla de principio de bloque* en su invocación. Incluso esto es cierto si había un argumento opcional para el ambiente `theorem`. Este argumento opcional está disponible vía la inserción `\inserttheoremaddition`.

En esta plantilla están disponibles numerosas inserciones.

Antes de que inicie la plantilla, la fuente se establece como la fuente del cuerpo prescrito por el ambiente a ser editado.

Ejemplo: Lo siguiente edita teoremas como `amsthm`:

```
\setbeamertemplate{theorem begin}
{%
  \begin{\inserttheoremblockenv}
  {%
    \inserttheoremheadfont
    \inserttheoremname
    \inserttheoremnumber
    \ifx\inserttheoremaddition\@empty\else\ (\inserttheoremaddition)\fi%
    \inserttheorempunctuation
  }%
}
\setbeamertemplate{theorem end}{\end{\inserttheoremblockenv}}
```

En el siguiente ejemplo se suprimen o ignoran todas las fuentes “sugeridas” para el ambiente y además se suprime el número del teorema.

```
\setbeamertemplate{theorem begin}
{%
  \normalfont% ignora la fuente del cuerpo
  \begin{\inserttheoremblockenv}
  {%
    \inserttheoremname
    \ifx\inserttheoremaddition\@empty\else\ (\inserttheoremaddition)\fi%
  }%
}
\setbeamertemplate{theorem end}{\end{\inserttheoremblockenv}}
```

Dentro de esta plantilla están disponibles las siguientes inserciones:

- `\inserttheoremblockenv` Esto normalmente se extenderá como un `block`, pero si se edita un teorema que tiene estilo de teorema `example` se extenderá como un `exampleblock`. Así que se puede usar esta inserción para decidir qué ambiente se debe usar al editar el teorema.
- `\inserttheoremheadfont` Esto se extenderá como una fuente cambiando el comando que reemplaza la fuente que se usa en el encabezado del teorema. Al no insertarlo se puede ignorar la fuente del encabezado.
- `\inserttheoremname` Este se extenderá como el nombre del ambiente a editar (como “Theorem” o “Corollary”).

- `\inserttheoremnumber` Si el teorema actual no tiene un número, esto se extenderá como el número del teorema actual precedido por un espacio o nada.
- `\inserttheoremaddition` Si no había ningún argumento opcional, esto se extenderá como el argumento opcional dado al ambiente o estará vacío.
- `\inserttheorempunctuation` Esto se extenderá como el caracter de puntuación para el ambiente actual. Normalmente es un punto.

Plantilla-Beamer `theorem end`

Se inserta al final de un teorema.

11.5. Texto enmarcado

Para dibujar un marco (un rectángulo) alrededor de algún texto, se puede usar el comando estándar de L^AT_EX `\fbox` y también `\frame` (dentro de un marco BEAMER, el comando `\frame` cambia su significado al del comando normal de L^AT_EX `\frame`). El paquete `fancybox` ofrece más tipos de marcos mediante los comandos: `\shadowbox`, `\doublebox`, `\ovalbox` y `\Ovalbox`.

La clase BEAMER también define dos ambientes para crear cajas coloreadas.

```
\begin{beamercolorbox} [opciones] {color beamer}
<contenido del ambiente>
\end{beamercolorbox}
```

Este ambiente se puede usar para editar convenientemente algún texto que usa algún color-BEAMER. Los dos siguientes bloques de comandos básicamente hacen lo mismo:

```
\begin{beamercolorbox}{color beamer}
  Texto
\end{beamercolorbox}
{
  \usebeamercolor{color beamer}
  \colorbox{bg}{
    \color{fg}
    Texto
  }
}
```

En otras palabras, el ambiente instala el `<color beamer>` y usa su fondo para el fondo de la caja y su primer plano para el texto dentro de la caja. En realidad se pueden dar numerosas opciones para especificar a gran detalle la manera de representar la caja.

Si el color de fondo de `<color beamer>` está vacío, detrás del texto no se dibuja ningún fondo, es decir, el fondo es “transparente”.

En los temas internos y externos predefinidos este comando se usa extensivamente para editar los encabezados y pies de página. Realmente no está destinado para usarse en marcos normales (por ejemplo, no está disponible dentro del modo `article`). Se debe preferir usar elementos estructurados como bloques o teoremas que automáticamente insertan tantas cajas coloreadas como sean necesarias.

Ejemplo: El siguiente ejemplo se podría usar para editar un encabezado con dos líneas, la primera muestra el título del documento, la segunda muestra el nombre del autor:

```
\begin{beamercolorbox}[ht=2.5ex,dp=1ex,center]{title in head/foot}
  \usebeamerfont{title in head/foot}
  \insertshorttitle
\end{beamercolorbox} %
```



```
\begin{beamercolorbox}[ht=2.5ex,dp=1ex,center]{author in head/foot}
  \usebeamerfont{author in head/foot}
  \insertshortauthor
\end{beamercolorbox}
```

Ejemplo: Editando un postit:

```
\setbeamercolor{postit}{fg=black,bg=yellow}
\begin{beamercolorbox}[sep=1em,wd=5cm]{postit}
  ;Colócame en algún lugar!
\end{beamercolorbox}
```

Se pueden dar las siguientes opciones:

- **wd**= $\langle ancho \rangle$ establece el ancho de la caja. Este comando tiene dos efectos: primero, `\hspace` de \TeX se establece como $\langle ancho \rangle$; segundo, después de que se edita la caja su ancho se establece como $\langle ancho \rangle$ (realmente no importa lo que resulte). Dado que al establecer `\hspace` no cambian automáticamente las dimensiones del ancho de línea de \LaTeX , al jugar con el ancho se debe considerar usar una mini página dentro de este ambiente.

Si el ancho es más grande que el ancho del área de texto normal (que indica el valor de `\textwidth`), el ancho de la caja resultante se restablece como `\textwidth`, pero se insertan inteligentes saltos negativos en los extremos izquierdo y derecho de la caja. El efecto neto de esto es que, para una caja, se puede usar un ancho más grande que el ancho del texto y se puede insertar directamente la caja resultante en el texto normal sin tener molestas advertencias y tener la caja colocada sensatamente.

- **dp**= $\langle profundidad \rangle$ establece la profundidad de la caja anulando su profundidad real. La caja primero se edita de manera normal y después se cambia su profundidad. Esta opción es útil para crear cajas que tienen un tamaño garantizado.

Si no se da la opción, la caja tiene su profundidad “natural” que es resultado de la edición. Por ejemplo, una caja que contiene sólo la letra “a” tendrá diferente profundidad que una caja que contiene sólo la letra “g”.

- **ht**= $\langle altura \rangle$ establece la altura de la caja, primordialmente la altura real. Note que la “altura” no incluye la profundidad. Si se quiere una caja de una línea que siempre tenga el mismo tamaño, normalmente es una buena opción establecer la altura en 2.25ex y la profundidad en 1ex.
- **left** causa que el texto dentro de la caja se alinee a la izquierda, con un borde derecho radicalmente irregular. Esto está predefinido. Para tener un mejor borde derecho irregular usaremos la opción `rightskip`.
- **right** causa que el texto se alinee a la derecha con un borde izquierdo irregular.
- **center** centra el texto dentro de la caja.
- **leftskip**= $\langle salto izquierdo \rangle$ instala $\langle salto izquierdo \rangle$ dentro de la caja como left skip. El left skip de \TeX es un pegamento que se inserta en el extremo izquierdo de cada línea.
- **rightskip**= $\langle salto derecho \rangle$ instala $\langle salto derecho \rangle$ como right skip. Para conseguir un buen borde derecho irregular, intentemos con `\rightskip=0pt más 4em`.
- **sep**= $\langle dimensión \rangle$ establece el tamaño del espacio extra alrededor del texto. Este espacio se agrega “dentro de la caja”, lo que significa que si se especifica un **sep** de 1cm y se inserta de manera normal la caja en la lista vertical, entonces el borde izquierdo de la caja se alinearán con el borde izquierdo del texto de la diapositiva, mientras que el borde izquierdo del texto dentro de la caja estará un 1cm a la derecha del borde izquierdo de la caja. Igualmente, el texto dentro de la caja se detendrá 1cm antes del borde derecho del área de texto normal.

- `colsep=<dimensión>` establece el “espacio de separación de color” extra alrededor del texto. Este espacio se comporta de la misma manera que el espacio agregado por `sep`, sólo que éste sólo se inserta si la caja tiene fondo coloreado, es decir, si el fondo del `<color beamer>` no está vacío. Este comando se puede usar junto con `sep`, los efectos se acumulan.
- `colsep*=<dimensión>` establece un espacio de separación de color extra alrededor del texto que está *horizontalmente fuera de la caja*. Esto significa que si la caja tiene un fondo, este fondo sobresale a la izquierda y derecha del texto tanto como `<dimensión>`, pero este fondo sobresaliente no se toma en consideración para propósitos de edición en T_EX.

Un ejemplo típico del uso de esta opción surge cuando se inserta una caja con un fondo coloreado en medio del área de texto normal. En este caso, si se establece el color de fondo, nos gustaría que se dibujara un fondo detrás del texto *y* nos gustaría tener un cierto espacio extra alrededor de este texto (el fondo no se debe detener inmediatamente en los bordes del texto, es tonto) *y* nos gustaría que el texto normal siempre esté en la misma posición horizontal, independientemente de si está presente o no un fondo. En este caso es una buena opción usar `colsep*=4pt`.

- `shadow=<true o false>` dibuja una sombra detrás de la caja. Actualmente, esta opción sólo tiene efecto si se usa junto con la opción `rounded`, pero se puede cambiar esto.
- `rounded=<true o false>` si hay un fondo instalado causa que se redondeen los bordes de la caja. Internamente este comando llama a `beamerboxesrounded`.
- `ignorebg` causa que se ignore el color de fondo del `<color beamer>`, esto es, se le trata como si se hubiera puesto “transparente” o “vacío”.
- `vmode` causa que cuando se inicien las cajas, T_EX este en modo vertical. Normalmente, al inicio de la caja T_EX está en modo horizontal (al principio de la caja se inserta automáticamente `\leavevmode`, a menos que se de esta opción). Sólo T_EXpertos necesitan esta opción, si se usa, es porque probablemente sabemos lo que estamos haciendo.

```
\begin{beamerboxesrounded}[<opciones>]{<encabezado>}
<contenido del ambiente>
\end{beamerboxesrounded}
```

El texto dentro del ambiente está enmarcado por un área rectangular con esquinas redondas. Para el área rectangular grande, se usa el color-BEAMER especificado con la opción `lower`, su fondo se usa para el fondo, su primer plano para el primer plano. Si el encabezado no está vacío, se le dibuja en la parte superior de la caja usando el color-BEAMER especificado con la opción `upper`, éste se usa tanto para el primer plano como para el fondo. Se pueden dar las opciones siguientes:

- `lower=<color beamer>` establece el color-BEAMER que se va a usar para la parte inferior (principal) de la caja. Su fondo se usa para el fondo, su primer plano para el primer plano de la parte principal de la caja. Si cualquiera de los dos está vacío, se usa el fondo o el primer plano actual. La caja nunca será transparente.
- `upper=<color beamer>` establece el color que se va a usar para la parte superior `<encabezado>` de la caja. Sólo se usa si el encabezado no está vacío.
- `width=<dimensión>` causa que el ancho del texto dentro de la caja sea de la `<dimensión>` especificada. Por defecto se usa `\textwidth`. Notamos que la caja sobresaldrá 4pt a la izquierda y a la derecha del texto.
- `shadow=<true o false>` . Si se pone `true` se dibuja una sombra.

Si no se da el encabezado, se suprime completamente esta parte.

Ejemplo:

```
\begin{beamerboxesrounded}[upper=block head,lower=block body,shadow=true]{Teorema}
  $A=B$.
\end{beamerboxesrounded}
```

ARTÍCULO Este ambiente no está disponible en modo `article`.

11.6. Figuras y tablas

Se pueden usar los ambientes `figure` y `table` de L^AT_EX estándar de la misma manera que las solemos usar. Sin embargo, se ignora cualquier especificación de colocación. Las figuras y tablas se insertan inmediatamente donde inicia el ambiente. Si hay demasiadas como para que quepan en el marco, se les puede dividir manualmente en marcos adicionales o se usa la opción `allowframebreaks`.

Ejemplo:

```
\begin{frame}
  \begin{figure}
    \pgfuseimage{mifigura}
    \caption{Este subt\,'{i}tulo se coloca debajo de la figura.}
  \end{figure}

  \begin{figure}
    \caption{Este subt\,'{i}tulo se coloca debajo de la figura.}
    \pgfuseimage{miotrafigura}
  \end{figure}
\end{frame}
```

Plantilla-/Color-/Fuente-Beamer `caption`

Esta plantilla se usa para hacer el subtítulo.

Las siguientes opciones de la plantilla están predefinidas:

- `[default]` edita el nombre del subtítulo (una palabra como “Figure” o “Abbildung” o “Tabla” dependiendo de si se edita una figura o tabla y dependiendo del idioma actualmente instalado) antes del texto del subtítulo. No se imprime ningún número ya que tienen poco sentido en una presentación normal.
- `[numbered]` agrega al subtítulo el número de la figura o tabla. Esta opción se usa sólo si el público tiene las notas de la conferencia o un folleto impreso siguiendo la misma numeración.
- `[caption name own line]` Como lo sugiere el nombre, esta opción pone el nombre del subtítulo (como “Figure”) en su propia línea.

Dentro de la plantilla se pueden usar las siguientes inserciones:

- `\insertcaption` Inserta el texto del subtítulo actual.
- `\insertcaptionname` Inserta el nombre del subtítulo actual. Esta palabra es “Table” o “Figure” o, si se usa el paquete `babel`, alguna traducción de eso.
- `\insertcaptionnumber` Inserta el número de la figura o tabla actual.

Color-/Fuente-Beamer `caption name`

Este color- y esta fuente-BEAMER se usan para editar el nombre del subtítulo. La plantilla `caption` debe “usarlos” directamente, no se instalan automáticamente por el comando `\insertcaptionname`.

11.7. Dividiendo un marco en múltiples columnas.

La clase BEAMER ofrece varios comandos y ambientes para dividir (quizás sólo parte de) un marco en múltiples columnas. Estos comandos no tienen nada que ver con los comandos de L^AT_EX para crear columnas. Las columnas son especialmente útiles para colocar un gráfico al lado de una descripción o explicación.

El principal ambiente para crear columnas se llama `columns`. Dentro de este ambiente se pueden colocar varios ambientes `column`, cada uno de los cuales crea una nueva columna, o usar el comando `\column` para crear nuevas columnas.

```
\begin{columns}[\langleopciones\rangle]
\langlecontenido del ambiente\rangle
\end{columns}
```

Un área de múltiples columnas. Dentro del ambiente se deben colocar sólo ambientes `column` o comandos `\column`. Se pueden dar las siguientes opciones:

- `b` causa que se alineen verticalmente las líneas inferiores de las columnas.
- `c` causa que las columnas se centren verticalmente, relativamente entre sí. Opción predefinida a menos que se use la opción global `t`.
- `onlytextwidth` es lo mismo que `totalwidth=\textwidth`.
- `t` causa que se alineen verticalmente las líneas superiores de las columnas. Opción predefinida si se usa la opción global `t`.
- `T` es similar a la opción `t`, pero `T` alinea la parte superior de las líneas superiores mientras que `t` alinea las llamadas líneas de base de las líneas superiores. Si parece que pasan cosas extrañas con la opción `t` (por ejemplo, si de repente una gráfica “cae” en lugar de “subir”) intente usar en su lugar esta opción.
- `totalwidth=\langleancho\rangle` causa que las columnas no ocupen el ancho total de la página sino en total sólo el `\langleancho\rangle`.

Ejemplo:

```
\begin{columns}[t]
  \begin{column}{5cm}
    Dos \\ 1\'{i}neas.
  \end{column}
  \begin{column}{5cm}
    Una 1\'{i}nea (pero alineada).
  \end{column}
\end{columns}
```

Ejemplo:

```
\begin{columns}[t]
  \column{5cm}
    Dos \\ 1\'{i}neas.

  \column[T]{5cm}
    \includegraphics[height=3cm]{migrafico.jpg}
\end{columns}
```

ARTÍCULO Este ambiente se ignora en este modo.

Para crear una columna, se puede usar el ambiente `column` o el comando `\column`.

```
\begin{column}[\langle colocación \rangle]{\langle ancho de la columna \rangle}
\langle contenido del ambiente \rangle
\end{column}
```

Creará una sola columna de $\langle ancho de la columna \rangle$. La colocación vertical de los ambientes cercadores `columns` se puede anular al especificar una $\langle colocación \rangle$ (`t` y `T` para los dos modos de parte superior, `c` para centrado y `b` para el fondo).

El siguiente código tiene el mismo efecto que los ejemplos anteriores:

```
\begin{columns}
  \begin{column}[t]{5cm}
    Dos \\l\ 'i}neas.
  \end{column}
  \begin{column}[t]{5cm}
    Una l\ 'i}nea (pero alineada).
  \end{column}
\end{columns}
```

ARTÍCULO Este comando se ignora en este modo.

```
\column[\langle colocación \rangle]{\langle ancho de la columna \rangle}
```

Inicia una sola columna. Los parámetros y las opciones son iguales que para el ambiente `column`. La columna automáticamente finaliza con la próxima ocurrencia de `\column` o de un ambiente `column` o del final del ambiente actual `column`.

Ejemplo:

```
\begin{columns}
  \column[t]{5cm}
    Dos \\l\ 'i}neas.
  \column[t]{5cm}
    Una l\ 'i}nea (pero alineada).
\end{columns}
```

ARTÍCULO Este comando se ignora en el modo `article`.

11.8. Colocando texto y gráficos absolutamente

Normalmente, BEAMER usa el mecanismo de edición normal de \TeX para colocar texto y gráficos en la página. En cambio, en ciertas situaciones se quiere que cierto texto o gráfico aparezca en una posición de la página que se especifica *absolutamente*. Esto significa que la posición se especifica relativa a la esquina izquierda superior de la diapositiva.

El paquete `textpos` provee varios comandos para colocar texto absolutamente y funciona junto con BEAMER. Al usar este paquete, normalmente se tiene que especificar las opciones `overlay` y quizás `absolute`. Para detalles de cómo usar el paquete ver su documentación.

11.9. Texto verbatim y fragile

Si se quiere usar un ambiente `{verbatim}` en un marco, se tiene que agregar la opción `[fragile]` al ambiente `{frame}`. En este caso, realmente se tiene que usar el ambiente `{frame}` (no el comando `\frame`) y el `\end{frame}` debe estar solitario, en una sola línea. El usar esta opción causará que se escriba el contenido del marco en un archivo externo y que se vuelva a leer.

También se debe usar la opción `[fragile]` para marcos que incluyen cualquier texto “frágil”, que es cualquier texto que no se “interpreta de la manera que normalmente se interpreta el texto en \TeX ”. Por ejemplo, se debe usar esta opción si se usa un paquete que (localmente) redefine el significado de, digamos, el carácter `&`

Dentro de ambientes `{verbatim}` obviamente no se pueden usar comandos como `\alert<2>` para resaltar parte de código ya que el texto está escrito, pues bueno, en verbatim. Existen varios buenos paquetes como `alltt` o `listings` que permiten sortear este problema. Para los casos simples se puede usar el siguiente ambiente definido en BEAMER:

```
\begin{semiverbatim}
<contenido del ambiente>
\end{semiverbatim}
```

El texto dentro de este ambiente se edita como texto verbatim. Sin embargo, los caracteres `\`, `{` y `}` mantienen su significado. Así, se pueden expresar cosas como

```
\alert<1->{\std::cout <<"AT&T le gusta 100% la presentación";}
```

Para editar los tres caracteres `\`, `{` y `}` se pueden usar los comandos `\\` (que se redefine dentro de este ambiente-como sea, no se necesita), `\{`, y `\}`. Así para conseguir editar “`\alert<1>{X}`” se puede escribir `\\alert<1>{\X\}`.

11.10. Abstract

En BEAMER el ambiente `abstract` es consciente de especificación de cubierta:

```
\begin{abstract}<especificación de acción>
<contenido del ambiente>
\end{abstract}
```

Se puede usar este ambiente para editar un abstract.

Color/Fuente-Beamer `abstract`

Este color y esta fuente-BEAMER se usan para editar el extracto (`abstract`). Si se establece un color de fondo, éste se usa por defecto como fondo para el extracto entero.

Plantilla-/Color-/Fuente-Beamer `abstract title`

Color: `titlelike`

Esta plantilla se usa para el título. Por defecto se inserta la palabra `\abstractname` centrada. Se ignora el color de fondo.

Plantilla-Beamer `abstract begin`

Esta plantilla se inserta muy al principio del extracto, antes de insertar el título del extracto y el contenido del ambiente.

Plantilla-Beamer `abstract end`

Esta plantilla se inserta al final del extracto, después del contenido del ambiente.

11.11. Verse, quotation, quote.

\LaTeX define tres ambientes para editar citas y versos: `verse`, `quotation` y `quote`. Estos ambientes también están disponibles en la clase BEAMER, en donde son conscientes de especificación de cubierta. Si se da una especificación de cubierta, el verso o la cita sólo se muestra en las diapositivas indicadas y se cubre de otro modo. La diferencia entre `quotation` y `quote` es que el primero tiene sangría de párrafo

mientras que el segundo no lo tiene.

Se les puede cambiar su fuente y su color cambiando los colores y las fuentes BEAMER listados abajo. A diferencia de los ambientes de L^AT_EX estándar, por defecto el tema fuente edita un verso en una fuente itálica serif, quotations y quotes se editan usando una fuente itálica (si es serif o sans-serif depende de la fuente del documento estándar).

```
\begin{verse}<<especificación de acción>>
<contenido del ambiente>
\end{verse}
```

Se puede usar estos ambientes para editar un verso.

Color/Fuente-Beamer verse

Este color y esta fuente-BEAMER se usan para editar el verso (verse). Si se establece un color de fondo, éste se usa por defecto como fondo para el verso entero. La fuente predefinida es Italic serif.

Plantilla-Beamer verse begin

Esta plantilla se inserta al principio del verso.

Plantilla-Beamer verse end

Esta plantilla se inserta al final del verso.

```
\begin{quotation}<<especificación de acción>>
<contenido del ambiente>
\end{quotation}
```

Usar este ambiente para editar citas en múltiples párrafos. Pensarlo de nuevo antes de presentar citas en múltiples párrafos.

Color/Fuente-Beamer quotation

Este color y esta fuente-BEAMER se usan para editar la cita.

Plantilla-Beamer quotation begin

Esta plantilla se inserta al principio de la cita.

Plantilla-Beamer quotation end

Esta plantilla se inserta al final de la cita.

```
\begin{quote}<<especificación de acción>>
<contenido del ambiente>
\end{quote}
```

Usar este ambiente para editar una cita de un solo párrafo.

Color/Fuente-Beamer quote

Este color y esta fuente-BEAMER se usan para editar la cita.

Plantilla-Beamer quote begin

Esta plantilla se inserta al principio de la cita.

Plantilla-Beamer quote end

Esta plantilla se inserta al final de la cita.

11.12. Pies de página

Primero unas palabras de advertencia: Usar notas a pie de página normalmente no es una buena idea. Interrumpen el flujo de la lectura. Se puede usar el comando `\footnote`. En BEAMER puede tomar una opción adicional para colocar las notas a pie de página en el fondo del marco en lugar del fondo de la mini página actual.

`\footnote`<especificación de cubierta>[opciones]{texto}

Inserta un pie de página en el marco actual. Los pies de página siempre se muestran al fondo del marco actual; nunca se “mueven” a otros marcos. Como de costumbre, se puede dar un número como *opciones* que causa que el pie de página use ese número. La clase BEAMER agrega una opción adicional:

- `frame` origina que el pie de página se muestre en el fondo del marco. Éste es normalmente el comportamiento por defecto, pero hace la diferencia en mini páginas y ciertos bloques. En una mini página, el pie de página normalmente se muestra como parte de la mini página más que como parte del marco.

Si se da una especificación de cubierta, se causa que el texto del pie de página sólo se muestre en las diapositivas indicadas. En todas las diapositivas se muestra el símbolo de pie de página en el texto.

Ejemplo: `\footnote{Sobre una maquina rápida.}`

Ejemplo: `\footnote[frame,2]{No probado.}`

Ejemplo: `\footnote<.->{Der Spiegel, 4/04, S.\ 90.}`

Plantilla-/Color-/Fuente-Beamer `footnote`

Esta plantilla se usa para hacer el pie de página. Dentro de esta plantilla se pueden usar las siguientes dos inserciones:

- `\insertfootnotetext` Inserta el texto del pie de página actual.
- `\insertfootnotemark` Inserta una marca al pie de página actual (como un número elevado). Esta marca se calcula automáticamente.

Color/Fuente-Beamer `footnote mark`

Este color/fuente se usa cuando se fabrica la marca del pie de página, tanto en el texto como al principio de el propio pie de página.

12. Gráficos

En lo que sigue se discuten las ventajas y desventajas de las diferentes maneras posibles de crear gráficos para una presentación BEAMER. Mucha de la información presentada no es especial a BEAMER sino que también aplica para cualquier otra clase de documento.

12.1. Incluyendo gráficos en archivos externos versus gráficos en línea.

Hay dos maneras principales de incluir gráficos en documentos de \TeX : el gráfico reside en un archivo externo que es *incluido* o el gráfico está *en línea* lo que significa que los archivos en \TeX contienen un grupo de comandos para “dibujar una línea roja de aquí para allá”.

Para crear un gráfico se puede usar un programa externo como `xfig` o el Gimp. Estos programas tienen una opción para exportar archivos gráficos en un formato que se puede insertar en la presentación.

La ventaja principal es:

- Se puede usar un programa poderoso para crear un gráfico de alta calidad.

Las desventajas principales son:

- Nos tenemos que ocupar de muchos archivos. Normalmente hay al menos dos por cada presentación: el archivo de datos gráficos del programa y el archivo gráfico exportado en un formato que pueda ser leído por \TeX .
- El cambiar el gráfico usando el programa, no cambia automáticamente el gráfico en la presentación. Más bien, se debe reexportar el gráfico y repetir la compilación en \LaTeX .
- Puede ser difícil conseguir el ancho de línea, fuentes y tamaño de fuente correctos.
- A menudo es difícil o imposible crear fórmulas como parte de gráficos.

Para insertar gráficos se pueden usar todos los comandos estándar de \LaTeX como `\includegraphics` (se debe incluir el paquete `graphics` o `graphicx`). También el paquete `pgf` ofrece comandos para incluir gráficos. En la mayoría de las situaciones cualquiera funciona bien, se puede elegir el que nos guste. Como `\pgfdeclareimage`, `\includegraphics` también incluye una imagen a la vez en un archivo `.pdf`, aun si se usa varias veces (como hecho práctico acerca de esto, el paquete `graphics` es incluso un poco más inteligente que `pgf`). Sin embargo, actualmente sólo `pgf` ofrece la habilidad para incluir imágenes que son en parte transparentes.

Al final de esta sección se encontrarán notas sobre cómo incluir formatos gráficos específicos como `.eps` o `.jpg`.

En BEAMER los comandos `\includegraphics`, `\pgfuseimage` y `\pgfimage` son conscientes de especificación de cubierta. El comando no tiene efecto si no aplica la cubierta indicada. Esto es útil para crear una simple animación donde cada imagen de la animación reside en un archivo diferente:

```
\begin{frame}
  \includegraphics<1>[height=2cm]{paso1.pdf} %
  \includegraphics<2>[height=2cm]{paso2.pdf} %
  \includegraphics<3>[height=2cm]{paso3.pdf} %
\end{frame}
```

Una manera diferente de crear gráficos es insertar directamente comandos de dibujo gráfico en el archivo \LaTeX . Hay numerosos paquetes que ayudan a hacer esto y que tienen varios grados de sofisticación. Los gráficos en línea no sufren ninguna de las desventajas mencionadas antes para la inclusión de archivos gráficos externos, pero la principal desventaja es que a menudo es difícil usar estos paquetes. En cierto modo, los gráficos se “programan”, lo que requiere de un poco de práctica. Al elegir un paquete gráfico se deben tener presente algunas cosas:

- Muchos paquetes producen gráficos de mala calidad. Esto es especialmente cierto en el caso del ambiente `picture` de \LaTeX estándar.
- Paquetes poderosos que producen gráficos de alta calidad a menudo no funcionan junto con `pdflatex`.
- El paquete más poderoso y fácil de usar, a saber `pstricks`, no funciona con `pdflatex` y este es un problema fundamental. Debido a las diferencias fundamentales entre PDF y PostScript, no es posible escribir “`pdflatex` back-end para `pstricks`.”

Una solución al problema anterior (aunque no necesariamente la mejor) es usar el paquete PGF. Produce gráficos de alta calidad y funciona con `pdflatex`, aunque también con `latex` normal. No es tan poderoso como `pstricks` (como se señaló antes, esto es debido a razones fundamentales) y no es tan fácil de usar; para la mayoría de los casos debe ser suficiente.

12.2. Incluyendo archivos de gráficos con extensión `.eps` o `.ps`

Se pueden incluir gráficos en archivos externos con extensión `.eps` (Encapsulated PostScript) o `.ps` (PostScript) si se usa `latex` y `dvips`, pero *no* si se usa `pdflatex`. Esto es cierto tanto para el paquete `graphics` como para `pgf`. Al usar `pgf` *no* se agrega la extensión `.eps`. Al usar `graphics` se debe agregar.

Si se tiene un gráfico `.eps` y se quiere usar `pdflatex`, se puede usar el programa `ps2pdf` para convertir el gráfico a un archivo `.pdf`. Observe sin embargo, que a menudo es mejor idea generar el `.pdf` si el programa que produjo el `.eps` soporta esta opción.

12.3. Incluyendo archivos gráficos con extensión `.pdf`, `.jpg`, `.jpeg` o `.png`

Los cuatro formatos `.pdf`, `.jpg`, `.jpeg`, y `.png` sólo pueden ser incluidos por `pdflatex`. Como antes, no hay que agregar estas extensiones al usar `pgf` pero sí al usar `graphics`. Si el archivo gráfico tiene cualquiera de estos formatos y se desea/debe usar `latex` y `dvips` primero se tiene que convertir el gráfico a `.eps`.

12.4. Incluyendo archivos gráficos con extensión `.mps`

Un archivo gráfico con extensión `.mps` (MetaPost PostScript) es un tipo especial de archivo Encapsulated PostScript. El programa MetaPost crea archivos con este formato. Como se sabe, \TeX es un programa que convierte simple texto llano en documentos bellamente editados. El programa MetaPost es similar sólo que éste convierte simple texto llano en gráficos bonitos.

El programa MetaPost convierte un archivo de texto llano con extensión `.mp` en un archivo `.mps` (aunque por alguna razón inexplicable no se agrega la extensión). El archivo `.mp` debe contener texto escrito en el lenguaje de programación MetaFont. Puesto que realmente los archivos `.mps` también son archivos `.eps`, para incluirlos se puede usar el comando `\includegraphics`.

Sin embargo, como ventaja especial *tal* archivo se puede incluir al usar `pdflatex`. Normalmente `pdflatex` no maneja archivos `.eps` pero los archivos `.mps` producidos por MetaPost tienen tal estructura simple y especial que lo hace posible. El paquete `graphics` implementa algunos filtros para convertir tal salida PostScript a PDF de manera dinámica. Para que funcione, el archivo debe terminar en `.mps` en lugar de `.eps`. Se puede usar el siguiente comando para hacer que el paquete `graphics` simplemente *asuma* la extensión `.mps` para cualquier archivo sobre el que no sabe nada (como los archivos con extensión `.1`, que le encanta producir a MetaPost):

```
\DeclareGraphicsRule{*}{mps}{*}{}{}
```

Actualmente esta característica especial sólo funciona con el paquete `graphics` no con `pgf`.

12.5. Incluir archivos gráficos con extensión .mmp

El formato `.mmp` (Multi-MetaPost) realmente no es un formato que pueda ser directamente incluido en un archivo `TEX`. Más bien, como un archivo `.mp`, primero se tiene que convertir usando el programa MetaPost. La diferencia crucial entre `.mp` y `.mmp` es que en este último formato múltiples gráficos pueden residir en un solo archivo `.mmp` (en realidad, también en un solo archivo `.mp`, pero por convención tales archivos se les llama `.mmp`). Al correr MetaPost sobre un archivo `.mmp`, no se crea sólo un archivo PostScript encapsulado sino varios con extensiones `.0`, `.1`, `.2`, etcétera. La idea es que `.0` pueda contener un gráfico principal y los siguientes dibujos puedan contener material de cubierta que se debe agregar incrementalmente a este gráfico.

Para incluir la serie de archivos resultantes, se puede usar el comando `\multiinclude` del paquete `mpmulti` o `xmpmulti`. El funcionamiento de este programa se explica en la Sección 13.1.3.

13. Animaciones, Sonidos y Transiciones de Diapositiva

13.1. Animaciones

Incluyendo archivos de animación externos

Si se ha creado una animación usando algún programa externo (como un renderer), se pueden usar las capacidades del programa de presentación (como Acrobat Reader) para mostrar la animación. Desgraciadamente, actualmente no existe una manera portátil de hacerlo e incluso Acrobat Reader no mantiene esta característica en todas las plataformas. Para incluir una animación en una presentación, se puede usar, por ejemplo, el paquete `multimedia.sty` que es parte de la clase BEAMER. Este paquete se tiene que incluir explícitamente. A pesar de ser distribuido como parte de la distribución BEAMER, este paquete es absolutamente autosuficiente y se puede usar independientemente.

```
\usepackage{multimedia}
```

Un paquete autosuficiente que implementa varios comandos para incluir animación externa y archivos de sonido en un documento PDF. El paquete se puede usar tanto con `dvips` más `ps2pdf` como con `pdflatex`, aunque el soporte de sonido especial sólo está disponible en `pdflatex`. Cuando se incluye este paquete también se debe incluir el paquete `hyperref`. Dado que `hyperref` típicamente se incluye al final del preámbulo, `multimedia` no incluirá `hyperref` por sí mismo. Sin embargo, `multimedia` se puede incluir tanto antes como después de `hyperref`. Dado que BEAMER incluye automáticamente `hyperref`, no se necesita tener cuidado de esto al crear una presentación con BEAMER.

Para incluir una animación en un archivo PDF, se puede usar el comando `\movie`. Dependiendo de las opciones usadas, este comando o arregla el archivo PDF de tal manera que la aplicación visor misma (como Acrobat Reader) intente reproducir la película o que se llame un programa externo. La última propuesta, aunque mucho menos flexible, se debe tomar en cuenta si la aplicación visor es incapaz de desplegar la película por sí misma.

```
\movie[<opciones>]{<texto del póster>}{<nombre de archivo película>}
```

Este comando inserta la película con el *<nombre de archivo película>* en el archivo PDF. El archivo de película debe residir en algún lugar donde la aplicación visor pueda encontrarlo, que suele ser el directorio en el que reside el archivo PDF final. El archivo de la película *no* se encaja en el archivo PDF, en el sentido que los datos reales de la película sean parte del archivo `main.pdf`. El archivo de película se debe copiar y pasar junto con el archivo PDF. (No obstante, a menudo se dice que la película esta “encajada” en el documento, pero eso sólo significa que al ver el documento se puede pulsar sobre la película e iniciar su reproducción).

La película usa un área rectangular cuyo tamaño se determina por las opciones `width=` y `height=` o por el tamaño de *<texto del póster>*. El *<texto del póster>* puede ser cualquier texto `TEX`, por ejemplo, puede ser un comando `\pgfuseimage` o un comando `\includegraphics` o un ambiente `pgfpicture` o simplemente texto. El *<texto del póster>* se edita en una caja, ésta a su vez se inserta en el área de texto normal y el rectángulo de película se pone exactamente sobre tal caja. Así, si el *<texto del póster>* es una imagen de la película, esta imagen se muestra hasta que inicia la película, cuando se reemplazará exactamente por la propia película. Sin embargo, también existe una manera diferente (a veces mejor) de crear una imagen de póster, a saber usando la opción `poster`.

La proporción del aspecto de la película *no* se corrige automáticamente si la dimensión de la caja del *<texto del póster>* no tiene la misma proporción. La mayoría de las películas tienen una proporción de aspecto de 4:3 o 16:9.

A pesar del nombre, una película puede consistir sólo de sonido sin imágenes. En este caso, el *<texto del póster>* puede ser un símbolo que representa el sonido. El comando `\sound` también incluye sonidos en un archivo PDF.

A menos que se den más opciones, la película sólo inicia cuando el usuario pulsa sobre ella. Que la aplicación visor pueda realmente desplegar la película depende de la aplicación y la versión. Por ejemplo, Acrobat Reader hasta la versión 5 parece que no es capaz de desplegar cualquier película o sonido en Linux. Por otro lado, Acrobat Reader versión 6 en MacOS es capaz de desplegar cualquier cosa que despliegue QuickTime, que es casi todo. El encaje de películas en un documento PDF se proporciona por el PDF normal y no es una peculiaridad de Acrobat Reader. En particular, se podría esperar que en el futuro otros visores como `xpdf` soporten el encaje de películas.

Ejemplo: `\movie{\pgfuseimage{miimagendeposter}}{mipelicula.avi}`

Ejemplo: `\movie[width=3cm,height=2cm,poster]{}{mipelicula.mpg}`

Si la aplicación visor no es capaz de reproducir la película (pero alguna aplicación externa si), se debe usar la opción `externalviewer` que le pide a la aplicación visor que lance una aplicación para mostrar la película en lugar de desplegarlo él mismo. Dado que esta aplicación se inicia en una nueva ventana, esto no es tan bonito como desplegar la película directamente por medio del visor (a menos que se usen trucos diabólicos para suprimir el marco de la aplicación visor). La aplicación que se elige queda a discreción de la aplicación visor, que intenta hacer su elección según la extensión del *<nombre de archivo película>* y de acuerdo a alguna tabla que envía tipos de extensiones a la aplicación visor. Cómo se puede modificar esta tabla depende de la aplicación visor y se debe consultar su manual.

Se pueden dar las siguientes *<opciones>*:

- **autostart** Causa la reproducción inmediata de la película cuando se muestra la página. De esta manera se puede iniciar a lo más una película. La aplicación visor típicamente sólo puede mostrar a lo más una película al mismo tiempo. Cuando ya no se muestra la página, la película se para inmediatamente. Éste puede ser un problema si se usa el comando `\movie` para incluir un sonido que se debe reproducir después de cerrar la página. En este caso, se debe usar el comando `\sound`.
- **borderwidth**=*<dimensión T_EX>* Causa que se dibuje un borde de *<dimensión T_EX>* alrededor de la película. Parece que algunas versiones de Acrobat Reader tienen un error y no lo despliegan si es más pequeño que 0.5bp (aproximadamente 0.51pt).
- **depth**=*<dimensión T_EX>* Anula la profundidad del *<texto del póster>* y lo establece como la dimensión dada.
- **duration**=*<tiempo>*s Especifica el tiempo de reproducción de la película. El *<tiempo>* puede ser un valor fraccional y debe ir seguido por la letra **s**. Junto con la opción **start**, se puede “recortar” una parte de la película a presentar.
- **externalviewer** Como se explicó anteriormente, esto causa que se lance una aplicación externa para desplegar la película en una ventana separada. La mayoría de las opciones, como **duration** o **loop**, no tienen efecto ya que no se pasan a la aplicación visor.
- **height**=*<dimensión T_EX>* Anula la altura de la caja del *<texto del póster>* y la pone como la *<dimensión T_EX>* dada.
- **label**=*<etiqueta de la película>* Asigna una etiqueta a la película de tal manera que ésta puede ser aludida posteriormente por medio del comando `\hyperlinkmovie`, el cual se puede usar para detener la película o para mostrar una parte diferente de ella. La *<etiqueta de la película>* no es una etiqueta normal. No debe ser demasiado elegante dado que se inserta literalmente en el código PDF. En particular no debe contener paréntesis derechos.
- **loop** Causa que empiece de nuevo la película cuando ha llegado a su final. Normalmente, la película sólo se detiene hasta el final.
- **once** Causa que la película se detenga sólo hasta el final. Este es el valor predefinido.
- **palindrome** Causa que la película se reproduzca hacía atrás cuando ha llegado a su final y que se reproduzca una vez más hacía adelante cuando ha llegado a su inicio y así sucesivamente.

- **poster** Le pide a la aplicación visor que muestre la primera imagen de la película cuando ésta no se esta reproduciendo. Normalmente no se muestra nada cuando la película no se está reproduciendo (y así, se muestra la caja conteniendo el *<texto del póster>*). Esta opción no es tan útil para una película que no tiene ninguna imagen (sino sonido) o para películas con una primera imagen sin información.
- **repeat** es lo mismo que loop.
- **showcontrols**=*<true o false>* Causa que se despliegue una barra de mando debajo de la película que se está reproduciendo. En lugar de **showcontrols=true** también se puede poner sólo **showcontrols**. Por defecto no se muestra la barra de mando.
- **start**=*<tiempo>*s Causa que se salten los primeros *<tiempo>* segundos de la película. Por ejemplo **start=10s, duration=5s** muestra los segundos del 10 al 15 de la película cuando se le reproduce.
- **width**=*<dimensión TEX>* funciona como la opción **height** sólo que para el ancho de la caja del póster.

Ejemplo: lo siguiente crea un “sonido de fondo” para la diapositiva.

```
\movie[autostart]{}{prueba.wav}
```

Ejemplo: una película con dos botones extras para mostrar partes diferentes de la película.

```
\movie[label=celulas,width=4cm,height=3cm,poster,showcontrols,duration=5s]{}{celulas.
avi}
\hyperlinkmovie[start=5s,duration=7s]{células}{\beamerbutton{Mostrar la etapa media}}
\hyperlinkmovie[start=12s,duration=5s]{células}{\beamerbutton{Mostrar la etapa final}}
```

Una película puede servir como destino de un tipo especial de hiperliga, a saber una hiperliga que se introduce usando el comando siguiente:

```
\hyperlinkmovie[<opciones>]{<etiqueta de la película>}{<texto>}
```

Causa que el texto se vuelva una hiperliga de película. Cuando se pulsa sobre el *<texto>*, la película con la *<etiqueta de la película>* se empieza a reproducir (o detener o pausar o reanudar dependiendo de las *<opciones>*). La película y la hiperliga deben estar en la misma página.

Se pueden dar las siguientes opciones, la mayoría son las mismas que para el comando `\movie`; si se da a la liga una opción diferente a la de la propia película, tiene prioridad la opción para la liga:

- **duration**=*<tiempo>*s Como para `\movie`, causa que se reproduzca la película sólo el número de segundos dado.
- **loop** y **repeat** Como para `\movie`, causa que la película se reproduzca repetidamente.
- **once** Como para `\movie`, causa que se reproduzca la película sólo una vez.
- **palindrome** Como para `\movie`, causa que la película se reproduzca sucesivamente hacía adelante y luego hacía atrás.
- **pause** Causa que se haga una pausa en la reproducción de la película, si es que se le estaba reproduciendo. Si no, no pasa nada.
- **play** Causa que se reproduzca la película desde cualquier posición de inicio indicada. Si la película ya se está reproduciendo, se detendrá y se reiniciará en la posición de inicio. Éste es el valor por defecto.
- **resume** Reanuda la reproducción de la película si se había pausado previamente. Si no se había hecho una pausa, ni se inició o ya se estaba reproduciendo, no pasa nada.

- **showcontrols**=*(true o false)* Como para `\movie`, causa que durante la reproducción se muestre o no una barra de mando.
- **start**=*(tiempo)*s Como para `\movie`, si se usa `play` para iniciar la película, causa que se salten los primeros segundos de la película.
- **stop** Causa que se detenga la reproducción de la película.

Animaciones creadas al mostrar diapositivas en sucesión rápida

Se puede crear una animación de una manera portable usando los comandos de cubierta del paquete BEAMER para crear una serie de diapositivas que, cuando se muestran en sucesión rápida, presentan una animación. Éste es un método flexible, pero tales animaciones normalmente son bastante estáticas ya que tardan algún tiempo para avanzar de una diapositiva a la siguiente. Este método es útil principalmente para las animaciones donde se quiere explicar cada “gráfico” de la animación. Cuando se adelantan las diapositivas “a mano”, es decir, al apretar el botón *AvPág*, típicamente toma por lo menos un segundo para que se muestre la próxima diapositiva.

Se pueden crear animaciones con más vida al confiar en la capacidad del programa visor. Algunos programas sólo pueden mostrar diapositivas por un cierto número de segundos durante una presentación (para Acrobat Reader esto sólo funciona en el modo de pantalla entera). Al establecer el número de segundos en cero se puede crear una sucesión rápida de diapositivas.

Para facilitar la creación de animaciones que usan esta característica, se pueden usar los comandos siguientes: `\animate` y `\animatevalue`.

`\animate`*<(especificación de cubierta)>*

Las diapositivas que satisfacen la *<(especificación de cubierta)>* se muestran tan brevemente como sea posible.

Ejemplo:

```
\begin{frame}
  \frametitle{Animaci\'}{o}n de cinco diapositivas}
  \animate<2-4>
```

La primera diapositiva se muestra de manera normal. Cuando se muestra la segunda (presumiblemente después de presionar una tecla de hacia adelante), la segunda, tercera y cuarta diapositivas “pasan como de rayo”. Al final se muestra el contenido de la quinta diapositiva.

... código para crear una animación con cinco diapositivas

```
\end{frame}
```

ARTÍCULO En este modo se ignora el comando

`\animatevalue`*<(diapositiva inicial)>-<(diapositiva final)>{(nombre)}{(valor inicial)}{(valor final)}*

El *<(nombre)>* debe ser el nombre de un contador o una dimensión y varía entre dos valores. Para las diapositivas en el rango especificado, el contador o dimensión se establece como un valor interpolado que depende del número de la diapositiva actual. En las diapositivas antes de *<(diapositiva inicial)>*, el contador o la dimensión se establece como *<(valor inicial)>*; en las diapositivas después de *<(diapositiva final)>* se establece como *<(valor final)>*.

Ejemplo:

```
\newcount\opaqueness
\begin{frame}
  \animate<2-10>
  \animatevalue<1-10>{\opaqueness}{100}{0}
```

```

\begin{colormixin}{\the\opaqueness!averagebackgroundcolor}
  \frametitle{Marco Desvaneci\'}{e}ndose}

  Este texto (y todo lo demás contenido en el marco) se desvanecerá cuando se
  muestre la segunda diapositiva. Esto funciona incluso con
  {\color{green!90!black}colored} \alert{texto}.
\end{colormixin}
\end{frame}

\newcount\opaqueness
\newdimen\offset
\begin{frame}
  \frametitle{Teoremas Voladores (;Realmente no se debe hacer!)}

  \animate<2-14>

  \animatevalue<1-15>{\opaqueness}{100}{0}
  \animatevalue<1-15>{\offset}{0cm}{-5cm}
  \begin{colormixin}{\the\opaqueness!averagebackgroundcolor}
  \hskip\offset
  \begin{minipage}{\textwidth}
    \begin{theorem}
      Este teorema vuela hac\'{i}a afuera.
    \end{theorem}
  \end{minipage}
\end{colormixin}

  \animatevalue<1-15>{\opaqueness}{0}{100}
  \animatevalue<1-15>{\offset}{-5cm}{0cm}
  \begin{colormixin}{\the\opaqueness!averagebackgroundcolor}
  \hskip\offset
  \begin{minipage}{\textwidth}
    \begin{theorem}
      Este teorema vuela hac\'{i}a adentro
    \end{theorem}
  \end{minipage}
\end{colormixin}
\end{frame}

```

ARTÍCULO En este modo se ignora el comando.

Si los “gráficos” de animación residen en archivos de gráfico externos individuales, también se puede considerar el comando `\multiinclude` (ver Sección 13.1.3) junto con `\animate`. Por ejemplo, suponiendo que se han creado los archivos gráficos nombrados `animación.1` hasta `animación.10`, se puede crear una animación como sigue:

```

\begin{frame}
  \animate<2-9>
  \multiinclude[start=1]{animación}
\end{frame}

```


Incluyendo animaciones externas que residen en múltiples archivos de imagen

Algunas animaciones residen en archivos externos de la manera siguiente: para cada etapa de la animación hay un archivo de imagen que contiene una imagen para esta etapa. Se puede incluir convenientemente tal serie de imágenes usando el estilo `mpmulti.sty` del paquete `ppower4`. Este estilo, escrito por Klaus Guntermann, introduce un comando llamado `\multiinclude` que toma el nombre base de un archivo de gráfico como `migraphic` y luego busca archivos llamados `migraphic.0`, `migraphic.1`, etcétera, hasta que no encuentra un archivo más. Luego incluye estos archivos de gráficos usando el comando `\includegraphics`, pero los pone “uno sobre el otro”. Además, y ésta es la parte importante, inserta un comando `\pause` después de cada gráfico. Este comando se define en el paquete `ppower4` y tiene el mismo efecto que el comando `\pause` de BEAMER.

Por esta razón, tanto `ppower4` como BEAMER, primero despliegan el gráfico básico y luego de manera adicional muestran el siguiente gráfico en cada diapositiva. Si se intenta usar directamente `mpmulti.sty`, uno se encuentra con el problema que incluye un archivo llamado `pause.sty` que es parte del paquete `ppower4`.

También se puede considerar el uso del estilo `xmpmulti.sty` que viene con BEAMER. Este archivo es casi idéntico a `mpmulti` salvo dos diferencias: primero, no incluye `pause.sty`, un estilo que conceptualmente desentona con BEAMER, aunque BEAMER contiene un método que esquiva el problema. Segundo, extiende el comando `\multiinclude` permitiendo una especial especificación de cubierta predefinida. El efecto de esto se explica más adelante.

```
\usepackage{xmpmulti}
```

Define el comando `\multiinclude`. El código de este paquete es de Klaus Guntermann con adiciones de Tantau. Se puede usar junto con BEAMER y `ppower4`, es decir, se puede usar como un reemplazo para `mpmulti` si también se incluye el paquete `pause` en una presentación `ppower4`.

```
\multiinclude[<especificación de cubierta predefinida>][<opciones>]{<nombre de archivo base>}
```

Excepto por la posibilidad de especificar una *<especificación de cubierta predefinida>*, este comando es idéntico al comando `\multiinclude` del paquete `ppower4`.

Si no se da ninguna especificación de cubierta, el comando busca archivos llamados *<nombre de archivo base>*.*<número>* para una sucesión creciente de números *<número>* empezando por cero. En cuanto encuentra estos archivos emite sobre ellos un comando `\includegraphics`. Los archivos que siguen al primero se ponen “en la parte superior” del primero.

Se inserta el comando `\pause` entre cualesquiera dos invocaciones de `\includegraphics`. Se puede modificar este comportamiento dando las *<opciones>* convenientes.

Ejemplo: Supongamos que MetaPost creó los archivos `gra.0`, `gra.1` y `gra.2`. Entonces se puede crear un marco que consiste de tres diapositivas que incrementalmente muestran su gráfico:

```
\begin{frame}
  \multiinclude{gra}
\end{frame}
```

El efecto de proporcionar una *<especificación de cubierta predefinida>* es el siguiente: primero, entre los gráficos no se inserta ningún comando `\pause`. En cambio, cada gráfico es rodeado por un ambiente `actionenv` con la especificación de cubierta establecida como *<especificación de cubierta predefinida>*.

Ejemplo: se puede crear el mismo efecto del ejemplo anterior usando `\multiinclude[<+>]{gra}`.

Ejemplo: para un uso más interesante de la *<especificación de cubierta predefinida>*, consideremos lo siguiente:

```
\multiinclude[<alert@+| +>]{gra}
```

Esto siempre pinta de rojo la parte del gráfico añadida mas recientemente (asumiendo que no se usan colores especiales en el gráfico mismo).

Ejemplo: para tener cada gráfico *reemplazando* completamente al anterior, se puede usar `\multiinclude[<+>]{gra}`.

Se pueden dar las siguientes *opciones* (son iguales a las del comando original en el paquete ppower4):

- **pause**=*<comando>* reemplaza el comando predefinido `\pause` por *<comando>*.
Si se da una *<especificación de cubierta predefinida>* el comando predefinido de pausado es vacío; en caso contrario es `\pause`.
Notamos que algunos comandos, como `\pauselevel`, no están disponibles en `\beamer`.
- **graphics**=*<opciones>* pasa las *<opciones>* al comando `\includegraphics`.
Ejemplo: `\multiinclude[graphics={height=5cm}]{gra}`
- **format**=*<extensión>* causa que se cambien los nombres de los archivos de *<nombre de archivo base>*.*<número>* a *<nombre de archivo base>*-*<número>*.*<extensión>*. Notamos el cambio del punto por un guión. Esta opción permite incluir, digamos, archivos con extensión `.jpg`.
- **start**=*<número>* especifica el *<número>* inicial. El valor por defecto es cero.
- **end**=*<número>* especifica el *<número>* final. El valor por defecto es el infinito.

Notamos que, si no se usa **format**=*<opciones>*, el comando `\includegraphics` estará sin saber en que formato está realmente el archivo gráfico. Después de todo, acaba con el ambiguo “sufijo de formato” `.0` o `.1`. Usando el comando siguiente se le puede decir a `\includegraphics` que cualquier archivo del que no sabe nada acerca de su sufijo realmente está en formato, digamos, `.mps`:

```
\DeclareGraphicsRule{*}{mps}{*}{}
```

13.2. Sonidos

Se pueden incluir sonidos en una presentación. Tal sonido se puede reproducir cuando se abre una diapositiva o cuando se presiona cierto botón. Los comandos para incluir sonidos están definidos en el paquete `multimedia`.

Como ya se apunto antes, se puede incluir un sonido en una presentación en PDF tratándolo como si fuera una película y se usa el comando `\movie`. Aunque en la mayoría de los casos esto es suficiente, hay dos casos donde este método no es satisfactorio:

1. cuando se cierra una página, se detiene inmediatamente la reproducción. Así, no se puede usar el comando `\movie` para crear sonidos que persisten durante un tiempo más largo.
2. no se pueden reproducir dos películas al mismo tiempo.

La especificación PDF introduce objetos de sonido especiales que se tratan de manera bastante diferente a los objetos de película. Se puede crear un objeto de sonido usando el comando `\sound` que es algo similar a `\movie`.

También existe allí un comando `\hyperlinksound` que es similar a `\hyperlinkmovie`. Aunque conceptualmente es mejor usar `\sound` para los sonidos, antes de usarlo hay que considerar varias cosas:

- se *pueden* reproducir varios sonidos al mismo tiempo. En particular, es posible reproducir un sonido general paralelamente a una película.
- la reproducción del sonido *puede* persistir después de que se cierra la página actual (aunque no se necesite).

- los datos de un archivo de sonido se *pueden* encajar completamente en un archivo PDF, liberándonos de la necesidad de “andar llevando” otros archivos.
- los objetos de sonido *no* funcionan con dvips mas ps2pdf. Sólo funcionan con pdf_latex.
- hay mucho menor control sobre cómo se debe reproducir la parte de un sonido. En particular, no se muestra ninguna barra de mando y no se puede especificar ni el tiempo de inicio ni la duración.
- un error en algunas versiones de Acrobat Reader hace necesario que se proporcionen detalles muy exactos sobre la codificación del archivo de sonido. Se tiene que proporcionar la proporción de la muestra, el número de canales (mono o estéreo), el número de bits por muestra y el método de codificación de la muestra (raw, signed, Alaw o μ law). Si no se saben estos datos o se proporcionan incorrectamente, el sonido se reproduce incorrectamente.
- parece que sólo se pueden incluir datos de sonido sin comprimir, los cuales se pueden hacer muy grandes fácilmente. La especificación no lo requiere, pero uno es incapaz de reproducir en Acrobat Reader sonido comprimido. Formatos de datos que funcionan son .aif y .au.

`\sound[opciones]{texto póster del sonido}{nombre del archivo de sonido}`

Este comando inserta el sonido con el nombre de archivo *nombre del archivo de sonido* en el archivo PDF. Como para `\movie`, el archivo debe estar disponible cuando se reproduzca el sonido. Sin embargo, a diferencia de `\movie`, se puede usar la opción `inlinesound` para realmente encajar los datos de sonido en el archivo PDF.

Como para una película, el *texto póster del sonido* se pondrá en una caja que, cuando se le pulse, empezará a reproducir el sonido. Sin embargo, también se puede dejar vacía esta caja y usar la opción `autostart`. Una vez que inició la reproducción del sonido, sólo se le puede detener iniciando la reproducción de un sonido distinto o usando el comando `\hyperlinkmute`.

Los formatos de sonido soportados dependen de la aplicación visor. Versiones recientes de Acrobat Reader soportan las extensiones .aif y .au.

Este comando sólo funciona junto con pdf_latex. Si se usa dvips todavía se muestra el póster, pero el pulsarlo no tiene efecto y de ninguna manera se encaja el sonido.

Ejemplo: `\sound[autostart,samplingrate=22050]{}{aplauso.au}`

Se pueden dar las siguientes opciones:

- **autostart** Causa la reproducción inmediata del sonido cuando se muestra la página.
- **automute** Causa que enmudezcan los sonidos cuando se abandona la página actual.
- **bitpersample**=*(8 o 16)*. Especifica el número de bits por muestra en el archivo de sonido. No se necesita especificar esta opción si el número es 16.
- **channels**=*(1 o 2)* Especifica si el sonido es mono o estéreo. No se necesita especificar esta opción si el sonido es mono.
- **depth**=*(dimensión T_EX)* Anula la profundidad de la caja del *texto póster del sonido* y lo establece como la dimensión dada.
- **encoding**=*(método)* Especifica el método de codificación que puede ser Raw, Signed, muLaw, o ALaw. No se necesita especificar esta opción si el método es muLaw.
- **height**=*(dimensión T_EX)* Anula la altura de la caja del *texto póster del sonido* y lo establece como la dimensión dada.

- **inlinesound** Causa que los datos de sonido se almacenen directamente en el archivo PDF.
- **label**=*<etiqueta de sonido>* Asigna una etiqueta al sonido de tal manera que ésta puede ser aludida posteriormente por medio del comando `\hyperlinksound`, el cual se puede usar para iniciar un sonido. La *<etiqueta de sonido>* no es una etiqueta normal.
- **loop** o **repeat** Causa la reproducción continua del sonido.
- **mixsound**=*<true o false>* Si se establece **true**, el sonido se reproduce adicionalmente a otro sonido que ya se está reproduciendo. Si se establece **false**, todos los demás sonidos (aunque no el sonido de películas) se detienen antes de que se reproduzca el sonido. El valor por defecto es **false**.
- **samplingrate**=*<número>* Especifica el número de muestras por segundo en el archivo de sonido. No se necesita especificar esta opción si el número es 44100.
- **width**=*<dimensión TEX>* Funciona como la opción **height**, sólo que para el ancho de la caja del póster.

Ejemplo: lo siguiente crea un “sonido de fondo” para la diapositiva, asumiendo que `aplauso.au` está codificado correctamente (44100 muestras por segundo, mono, codificación μ law, 16 bits por muestra).
`\sound[autostart]{\aplauso.au}`

Igual que en las películas, los sonidos también pueden servir como destinos de hiperligas de sonido especiales.

`\hyperlinksound[<opciones>]{<etiqueta de sonido>}{<texto>}`

Causa que el *<texto>* se convierta en una hiperliga de sonido. Cuando se pulsa sobre el *<texto>* se empieza a reproducir el sonido con la etiqueta *<etiqueta de sonido>*.

Se pueden dar las siguientes *<opciones>*:

- **loop** o **repeat**. Causa que vuelva a iniciar el sonido después de llegar al final del mismo.
- **mixsound**=*<true o false>*. Si se establece **true**, el sonido se reproduce adicionalmente a otro sonido que ya se está reproduciendo. Si se establece **false**, todos los demás sonidos (aunque no el sonido de películas) se detienen antes de que se reproduzca el sonido. El valor por defecto es **false**.

Dado que no existe una manera directa de detener la reproducción de un sonido, el siguiente comando es útil:

`\hyperlinkmute{<texto>}`

Causa que el *<texto>* se convierta en hiperliga que, cuando se le presiona, detiene la reproducción de todos los sonidos.

13.3. Transiciones de diapositiva

PDF en general y Acrobat Reader en particular, ofrecen una manera estandarizada de definir *transiciones de diapositiva*. Tal transición es un efecto visual que se usa para mostrar una diapositiva. Por ejemplo, en lugar de simplemente exhibir inmediatamente la siguiente diapositiva, cualquier cosa que se mostró antes se puede “disolver” lentamente y ser reemplazada por el contenido de la diapositiva.

Hay varios comandos que se pueden usar para especificar qué efecto se debe usar cuando se presenta la diapositiva actual. Consideremos lo siguiente:

```
\frame{\pgfuseimage{joven}}
\frame{
  \transdissolve
```

```
\pgfuseimage{hombre}
}
```

El comando `\transdissolve` causa que la diapositiva del segundo marco se muestre de “manera desvanecida”. Notamos que el desvanecimiento es una propiedad del segundo marco no del primero. Podemos poner el comando en cualquier parte del marco.

Los comandos de transición son conscientes de especificación de cubierta. Podemos colapsar los dos marcos anteriores en uno solo, así:

```
\begin{frame}
  \only<1>{\pgfuseimage{joven}}
  \only<2>{\pgfuseimage{hombre}}
  \transdissolve<2>
\end{frame}
```

Esto declara que en la primera diapositiva se debe mostrar el joven, en la segunda diapositiva se debe mostrar el hombre viejo y cuando se muestra la segunda diapositiva ésta se debe mostrar de una “manera desvanecida”.

A continuación se listan los diferentes comandos para crear efectos de transición. Todos ellos toman un argumento opcional que puede contener una lista de pares $\langle clave \rangle = \langle valor \rangle$. Son posibles las siguientes opciones:

- `duration = \langle segundos \rangle` Especifica el número de $\langle segundos \rangle$ necesarios para el efecto de transición. Por defecto es un segundo, pero a menudo es apropiado uno más corto (como 0.2 segundos). Las aplicaciones visor, sobre todo Acrobat Reader, pueden interpretar esta opción de una manera algo extraña.
- `direction = \langle grado \rangle` Para efectos “dirigidos”, esta opción especifica la dirección del efecto. Los valores permitidos son 0, 90, 180, 270 y para el efecto glitter también 315.

ARTÍCULO Todos estos comandos se ignoran en este modo.

`\transblindshorizontal` $\langle especificación de cubierta \rangle$ [$\langle opciones \rangle$]
Muestra a la diapositiva como si se abriera una persiana horizontal.

Ejemplo: `\transblindshorizontal`

`\transblindsvertical` $\langle especificación de cubierta \rangle$ [$\langle opciones \rangle$]
Muestra a la diapositiva como si se abriera una persiana vertical.

Ejemplo: `\transblindsvertical<2,3>`

`\transboxin` $\langle especificación de cubierta \rangle$ [$\langle opciones \rangle$]
Muestra a la diapositiva desplazando los cuatro lados hacia su centro.

Ejemplo: `\transboxin<1>`

`\transboxout` $\langle especificación de cubierta \rangle$ [$\langle opciones \rangle$]
Muestra a la diapositiva ampliando un área rectangular a partir de su centro.

Ejemplo: `\transboxout`

`\transdissolve` $\langle especificación de cubierta \rangle$ [$\langle opciones \rangle$]

Muestra a la diapositiva desvaneciendo lo que se mostró antes.

Ejemplo: `\transdissolve[duration=0.2]`

`\transglitter`*(especificación de cubierta)* [*(opciones)*]

Muestra a la diapositiva con un efecto de brillo que barre en la dirección especificada.

Ejemplo: `\transglitter<2-3>[direction=90]`

`\transsplitverticalin`*(especificación de cubierta)* [*(opciones)*]

Muestra a la diapositiva con dos líneas verticales barriendo desde los lados hacia el centro.

Ejemplo: `\transsplitverticalin`

`\transsplitverticalout`*(especificación de cubierta)* [*(opciones)*]

Muestra a la diapositiva con dos líneas verticales barriendo desde el centro hacia los lados.

Ejemplo: `\transsplitverticalout`

`\transsplithorizontalin`*(especificación de cubierta)* [*(opciones)*]

Muestra a la diapositiva con dos líneas horizontales barriendo desde los lados hacia el centro.

Ejemplo: `\transsplithorizontalin`

`\transsplithorizontalout`*(especificación de cubierta)* [*(opciones)*]

Muestra a la diapositiva con dos líneas horizontales barriendo desde el centro hacia los lados.

Ejemplo: `\transsplithorizontalout`

`\transwipe`*(especificación de cubierta)* [*(opciones)*]

Muestra a la diapositiva con una sola línea barriendo en la dirección especificada, “limpiando” el contenido anterior.

Ejemplo: `\transwipe[direction=90]`

También se puede especificar el tiempo que se debe mostrar una diapositiva dada, usando el siguiente comando consciente de especificación de cubierta.

`\transduration`*(especificación de cubierta)*{*(número de segundos)*}

En el modo de pantalla completa, muestra la diapositiva durante *(número de segundos)*. Si se especifica cero, la diapositiva se muestra el menor tiempo posible. Esto se puede usar para crear pseudo animaciones interesantes.

Ejemplo: `\transduration<2>{1}`

Cambiando la Apariencia de las Cosas

BEAMER ofrece varias maneras de cambiar la apariencia de una presentación a todos los niveles de detalle. En un nivel superior, se pueden usar *temas* para cambiar de una manera conveniente la apariencia global. En un nivel inferior, las *plantillas* permiten especificar de manera individual la apariencia de cada mínimo detalle.

Dos aspectos importantes de la “apariencia” de una presentación se tratan en secciones extras: los colores y las fuentes. Aquí también se pueden usar los temas de color y fuente para cambiar globalmente los colores y las fuentes de una presentación, mientras que todavía se puede cambiar independientemente de todo lo demás el color o la fuente de, digamos, los títulos de bloques.

14. Temas

14.1. Cinco sabores de temas

Los *temas* facilitan el cambio de apariencia de una presentación. La clase BEAMER usa cinco diferentes tipos de temas:

Temas de Presentación Conceptualmente, un tema de presentación determina qué apariencia se va a usar para cada detalle en una presentación. Así, al elegir un tema de presentación particular se establecerá, digamos, que color tendrán los números en una enumeración, qué color de fondo tendrán, qué fuente se usará para hacerlos, si se dibujará detrás de ellos un círculo, una bola, un rectángulo o cualquier otra cosa. Así, cuando se elige un tema de presentación, ésta se verá de la manera que alguien (el creador del tema) pensó que se debería ver una presentación. Los temas de presentación típicamente sólo eligen un tema de color, un tema fuente, un tema interno y un tema externo particular, los cuales se combinan bien.

Temas de color Un tema de color sólo determina qué colores se van a usar en una presentación. Si se elige un tema de presentación particular y luego se elige un tema de color, sólo cambiarán los colores de la presentación. Un tema de color puede especificar colores de una manera muy detallada: por ejemplo, puede cambiar específicamente los colores para hacer, digamos, el borde de un botón, el fondo de un botón y el texto en un botón.

Temas de fuente Un tema de fuente determina qué fuentes o atributos de las fuentes se van a usar en una presentación. Como con los colores, las fuentes de todos los elementos de texto usados en una presentación se pueden especificar de manera independiente.

Temas internos Un tema interno determina cómo se editan ciertos elementos de una presentación. Esto incluye todos los elementos que están “dentro” del marco, es decir, que no son parte del encabezado, pie de página o barras laterales. Esto incluye todas las enumeraciones, ambientes de listado, ambientes de bloque, ambientes de teorema o la tabla de contenidos. Por ejemplo, un tema interno podría especificar que en una enumeración el número se debe editar sin un punto y que se dibuje un círculo pequeño detrás de él. El tema interno *no* especifica qué color se debe usar para el número o el círculo (éste es trabajo del tema de color) ni qué fuente se debe usar (éste es trabajo del tema de fuente).

Temas externos Un tema externo especifica la apariencia de lo que esta “afuera” o en la “frontera” de las diapositivas de la presentación. Especifica si hay encabezados y pies de pagina, lo que se muestra en ellos, si hay una barra lateral, donde va el logotipo, donde los símbolos y barras de navegación, etcétera. También especifica donde se pone el título del marco y cómo se edita.

Los diferentes temas residen en los cuatro subdirectorios `theme`, `color`, `font`, `inner` y `outer` del directorio `beamer/themes`. Internamente, un tema se almacena como un archivo de estilo normal. Sin embargo, para usar un tema se deben usar los siguientes comandos especiales:

```
\usetheme[<opciones>]{<lista de nombres>}
```

Instala el tema de presentación llamado *<nombre>*. Actualmente, el efecto de este comando es el mismo que poner `\usepackage` para el archivo de estilo llamado `beamertheme<nombre>.sty` para cada *<nombre>* en la *<lista de nombres>*.

```
\usecolortheme[<opciones>]{<lista de nombres>}
```

Como `\usetheme` sólo que para temas de color. Sus archivos de estilo se llaman `beamercolortheme<nombre>.sty`

```
\usefonttheme[<opciones>]{<nombre>}
```

Como `\usetheme` sólo que para temas de fuente. Sus archivos se llaman `beamerfonttheme<nombre>.sty`


```
\useinnertheme[<opciones>]{<nombre>}
```

Como `\usetheme` sólo que para temas internos. Sus archivos se llaman `beamerinnertheme(<nombre>).sty`

```
\useoutertheme[<opciones>]{<nombre>}
```

Como `\usetheme` sólo que para temas externos. Sus archivos se llaman `beameroutertheme(<nombre>).sty`

Si no se usa ninguno de estos comandos, para todos ellos se usa un tema sobrio *predefinido*. A continuación se describen los temas de presentación que vienen con la clase BEAMER. Los temas de elemento, distribución, color y fuente se presentan en las siguientes secciones.

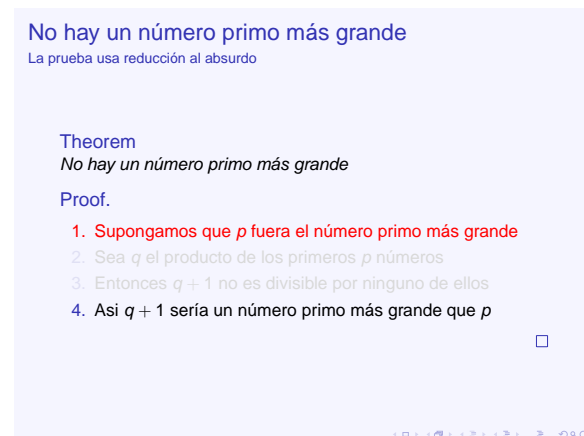
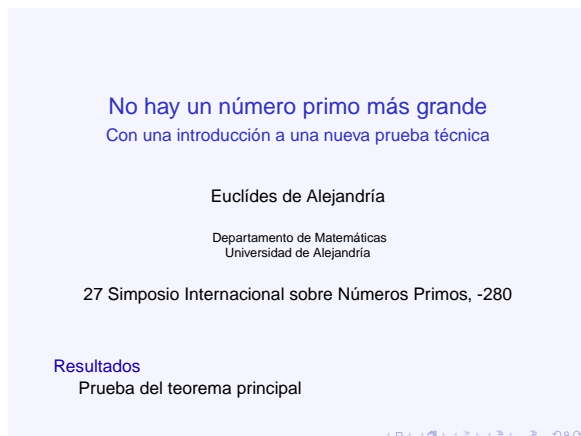
14.2. Temas de presentación sin barras de navegación

Un tema de presentación dicta la apariencia de cada detalle de la presentación. Habiendo elegido un tema de presentación particular, no se necesita especificar nada, ni tener que hacer algo con la apariencia—el creador del tema ya lo hizo. Sin embargo todavía se *pueden* cambiar cosas después, al usar un color, fuente o elemento distinto o incluso el tema layout; o al cambiar colores específicos, fuentes o las plantillas directamente.

Salvo dos casos especiales, todos los temas de presentación son nombres de ciudades, las cuales visitó el creador de BEAMER al asistir a una conferencia o taller. Hay temas que no desarrolló Tantau y esto se indica en el tema correspondiente. Les hizo cambios ligeros o “corrigió” algunos temas pero todavía se nombran los autores originales.

```
\usetheme{default}
```

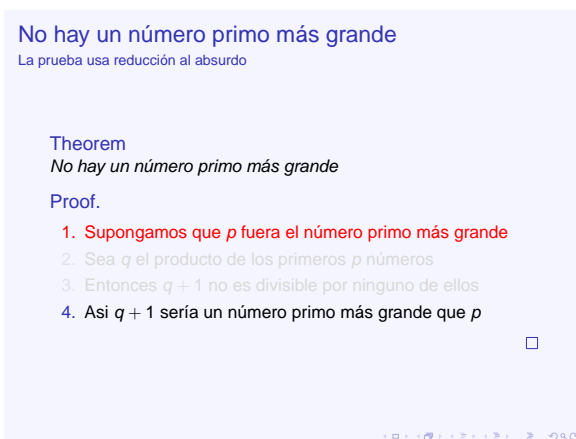
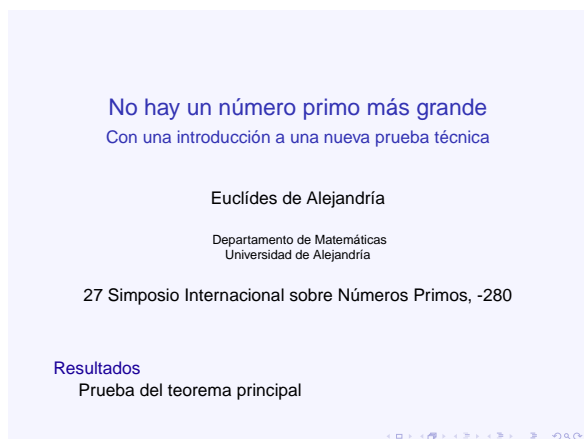
Ejemplo:



Como lo sugiere el nombre este tema se instala por defecto. Es un tema sin sentido, sobrio, que hace uso mínimo de variaciones de color o fuente. Este tema es útil para todo tipo de pláticas excepto para las que son muy largas.

```
\usetheme[headheight=<altura de la cabeza>,footheight=<altura del pie>]{boxes}
```

Ejemplo:



Para este tema se puede especificar un número arbitrario de plantillas para las cajas en el encabezado y en el pie de página. Se puede agregar una plantilla para otra caja usando los siguientes comandos.

`\addheadbox{⟨color beamer⟩}{⟨plantilla de caja⟩}`

Cada vez que se invoca este comando se agrega una caja nueva a la línea de encabezado, mostrando a la izquierda la primera caja añadida. Todas las cajas tendrán el mismo tamaño. Se usa el `⟨color beamer⟩` para establecer los colores del primer plano y del fondo de la caja.

Ejemplo:

`\addheadbox{section in head/footer}{\tiny\quad 1. Caja}`

`\addheadbox{structure}{\tiny\quad 2. Caja}`

Se logra directamente un efecto similar al de los comandos previos instalando una plantilla de encabezado que contiene dos `beamercolorbox`:

```
\setbeamertemplate{headline}
{\leavevmode
\begin{beamercolorbox}[width=.5\paperwidth]{section in head/footer}
\tiny\quad 1. Caja
\end{beamercolorbox}%
\begin{beamercolorbox}[width=.5\paperwidth]{structure}
\tiny\quad 2. Caja
\end{beamercolorbox}
}
```

Mientras más complicado sea, los comandos previos ofrecen más flexibilidad.

`\addfootbox{⟨color beamer⟩}{⟨plantilla de caja⟩}`

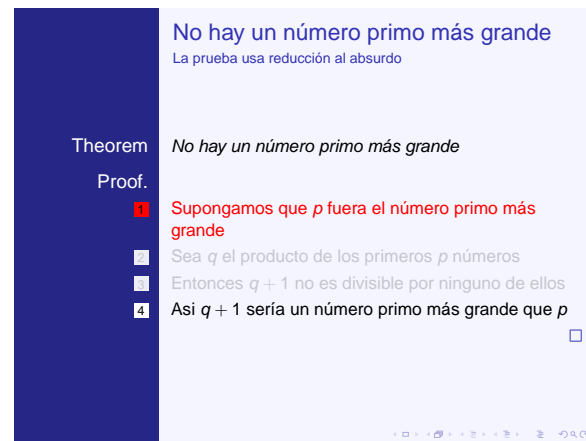
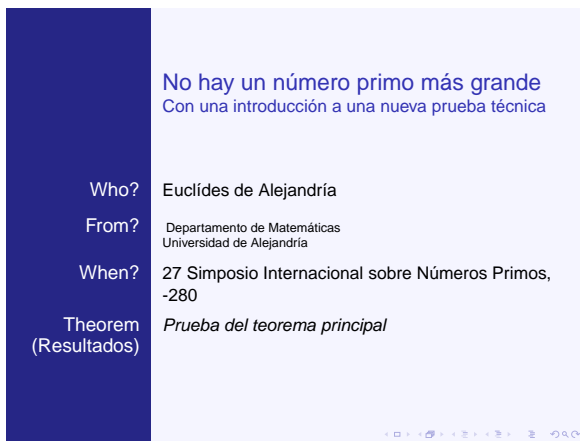
Ejemplo:

`\addfootbox section in head/footer}{\tiny\quad 1. Caja}`

`\addfootbox{structure}{\tiny\quad 2. Caja}`

`\usetheme[⟨opciones⟩]{Bergen}`

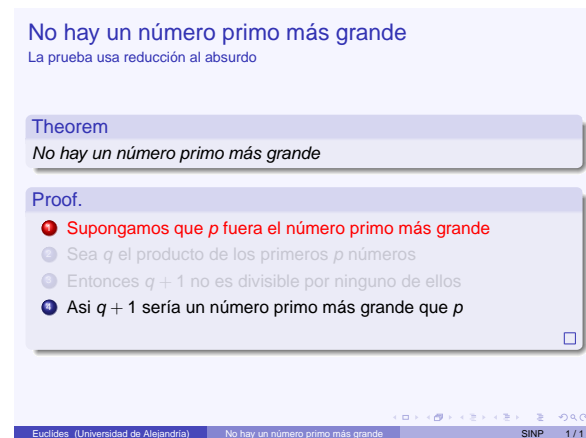
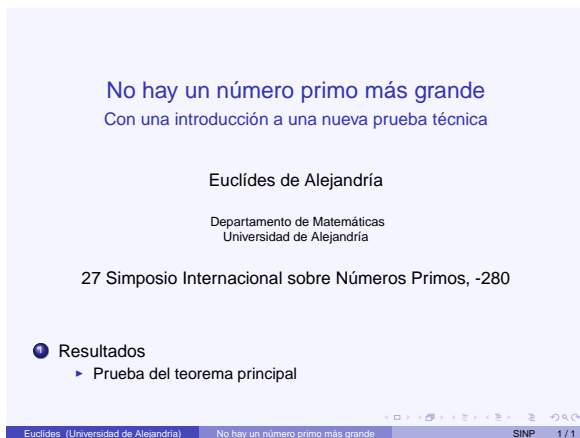
Ejemplo:



Es un tema que se basa en los temas internos `inmargin` y `rectangle`. Usar este tema no es trivial dado que conseguir el espacio correcto puede ser más truculento que con la mayoría de los demás temas. Además, este tema no funciona con columnas. Se pueden consultar los comentarios en el tema interno `inmargin`. Bergen es un pueblo en Noruega.

`\usetheme[<opciones>]{Boadille}`

Ejemplo:



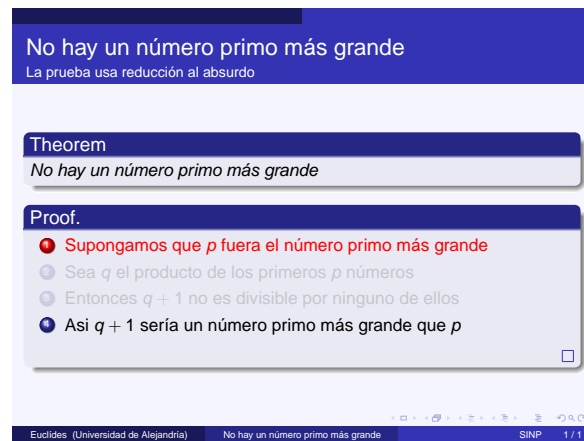
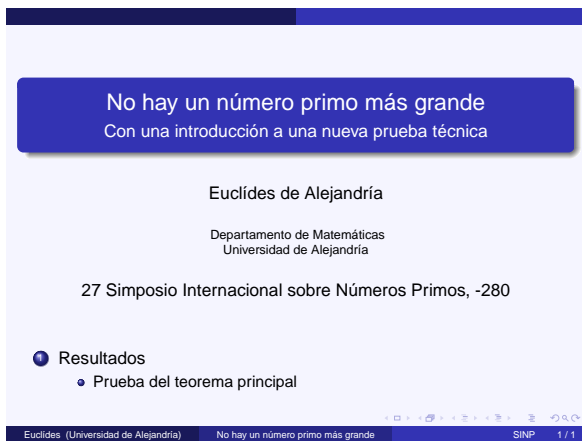
Un tema que da mucha información en poco espacio. Se pueden dar las siguientes `<opciones>`:

- `secheader` causa que se inserte un encabezado mostrando la sección y subsección actual. Por defecto este encabezado no se muestra.

Autor del tema: Manuel Carro. Boadilla es un pueblo cerca de Madrid, sede del departamento de Ciencia Informática de la Universidad.

`\usetheme[<opciones>]{Madrid}`

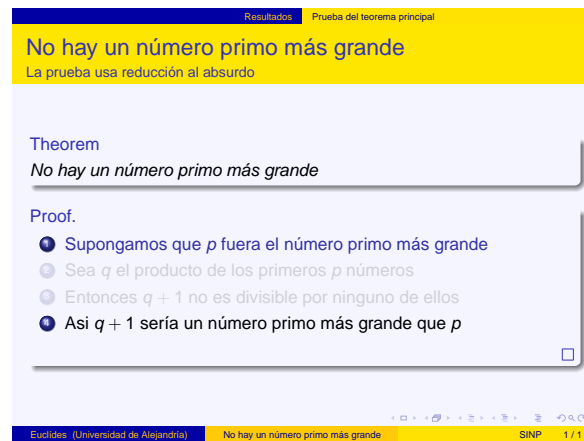
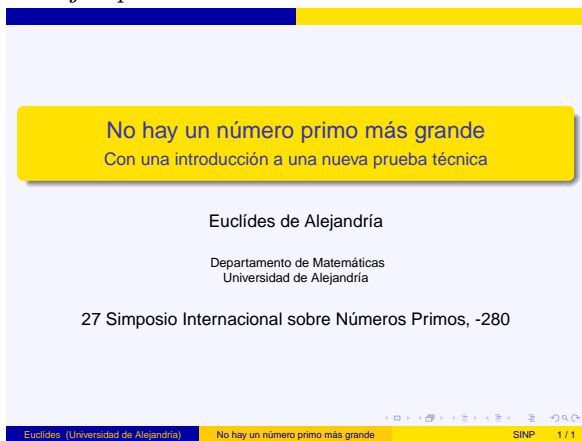
Ejemplo:



Como el tema `Boadille` sólo que usa colores más fuertes y no se modifican los iconos de listados. Se pueden dar las mismas *opciones* que el tema `Boadilla`. *Autor del tema:* Manuel Carro.

`\usetheme{AnnArbor}`

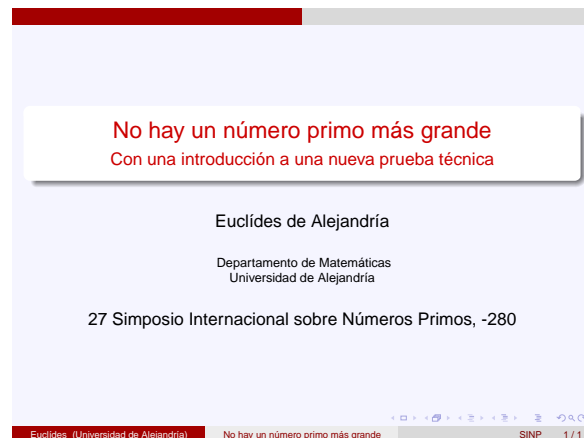
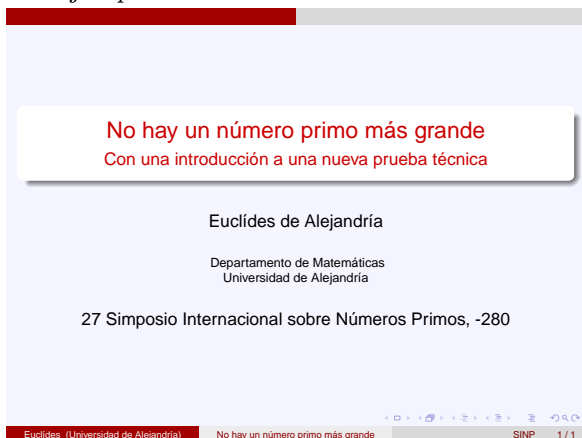
Ejemplo:



Como `Boadille` pero usa los colores de la Universidad de Michigan localizada en Ann Arbor. *Autor:* Madhusudan Singh.

`\usetheme{CambridgeUS}`

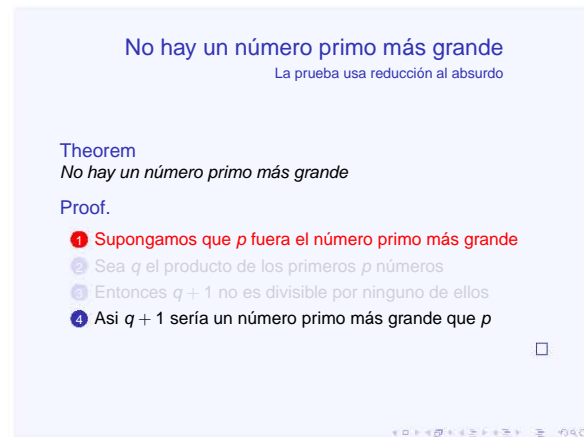
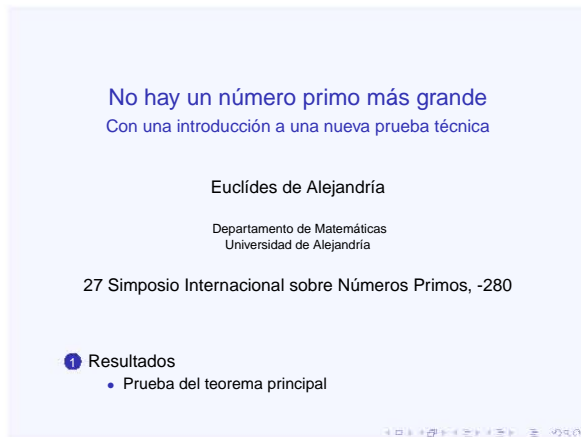
Ejemplo:



Como `Boadille` pero usa los colores del MIT. *Autor:* Madhusudan Singh.

```
\usetheme{Pittsburgh}
```

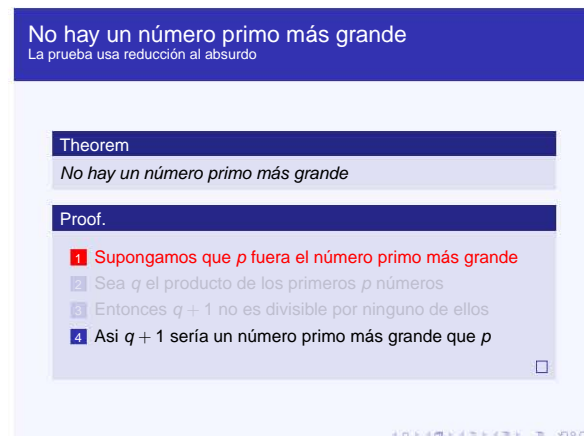
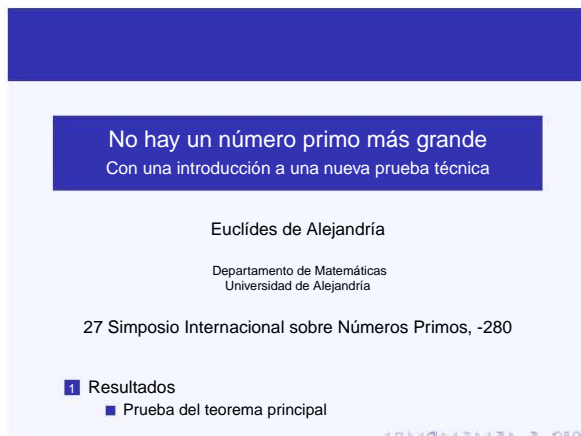
Ejemplo:



Un tema sobrio. Ubica los encabezados a la derecha creando una “tensión” interesante dentro de cada marco. Pittsburgh es un pueblo en el este de EU.

```
\usetheme[<opciones>]{Rochester}
```

Ejemplo:



Un tema dominante sin elementos de navegación. Se puede hacer menos dominante usando un tema con un color diferente. Rochester es un pueblo en el norte de Nueva York, EU.

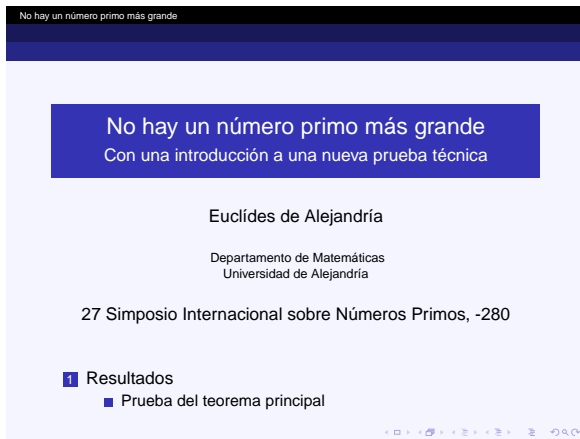
Se pueden dar las siguientes *<opciones>*:

- `height=<dimensión>` establece la altura de la barra de título del marco.

14.3. Temas de presentación con barra de navegación tipo árbol.

```
\usetheme{Antibes}
```

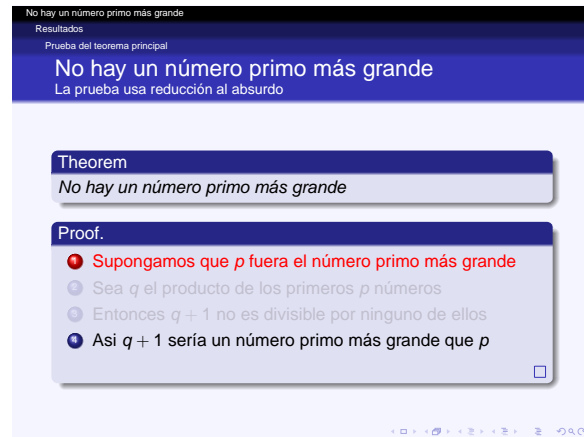
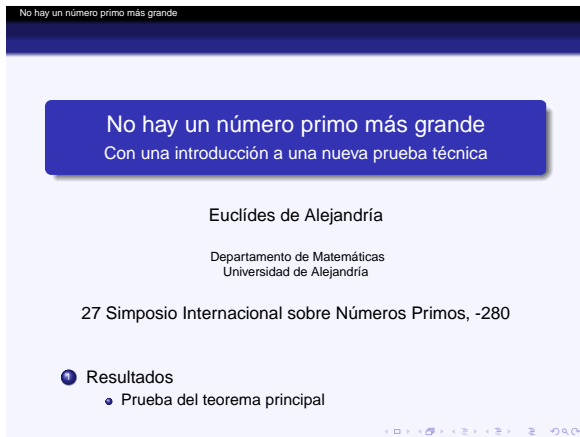
Ejemplo:



Un tema dominante con una barra de navegación tipo árbol en la parte superior. Los elementos rectangulares en la parte superior reflejan la navegación de modo rectangular. El tema se puede hacer menos dominante usando un tema con un color diferente. Antibes es un pueblo en el sur de Francia.

`\usetheme{JuanLesPins}`

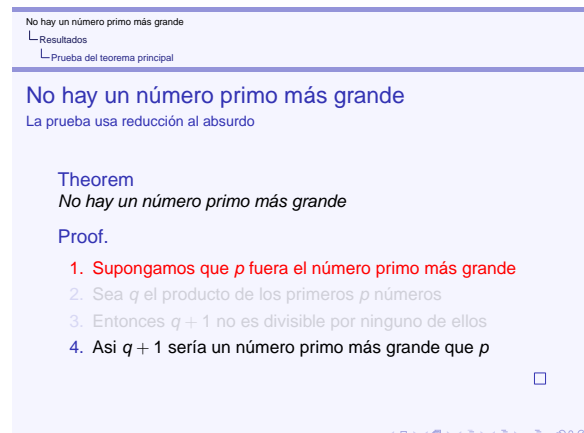
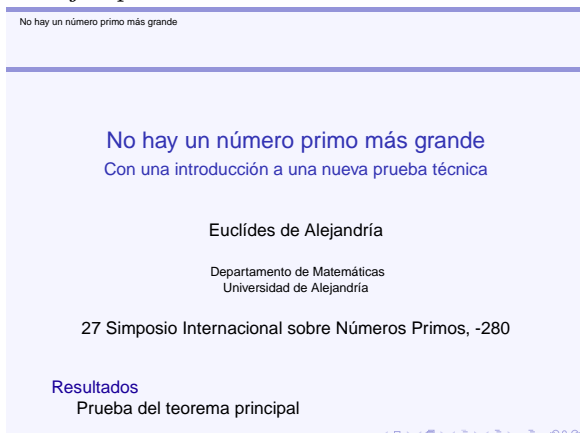
Ejemplo:



Una variación del tema *Antibes* que tiene una apariencia mucho “más suave”. Se puede hacer menos dominante usando un tema con un color diferente. Juan-Les-Pins es un acogedor pueblo cerca de Antibes.

`\usetheme{Montpellier}`

Ejemplo:



Un tema sobrio que da una pizca de navegación básica. El encabezado se puede hacer más dominante

usando un tema con un color diferente. Montpellier está en el sur de Francia.

14.4. Temas de presentación con barra lateral con la tabla de contenido

`\usetheme[<opciones>]{Berkeley}`

Ejemplo:



Un tema dominante. Si la barra de navegación está a la izquierda, ésta domina desde que se ve por primera vez. La altura del encabezado del marco se fija en dos y media líneas, así que se debe tener cuidado con títulos muy largos. Se pone un logotipo en el área de la esquina. Las áreas rectangulares dominan la distribución. El tema se puede hacer menos dominante usando un tema de color diferente.

Por defecto, la entrada actual en la tabla de contenidos de la barra lateral se resalta usando un color más vibrante. Una buena alternativa es resaltar la entrada actual usando un color diferente para el fondo del punto actual. El tema de color `sidebartab` instala los colores apropiados para que simplemente se tenga que poner `\usecolortheme{sidebartab}`.

Este tema de color funciona con todos los temas que muestran una tabla de contenidos en la barra lateral.

Este tema es útil para pláticas largas, como en conferencias que requieren de una tabla de contenidos que sea visible todo el tiempo.

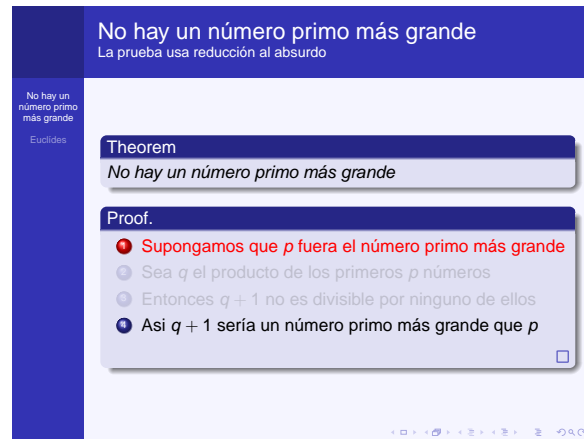
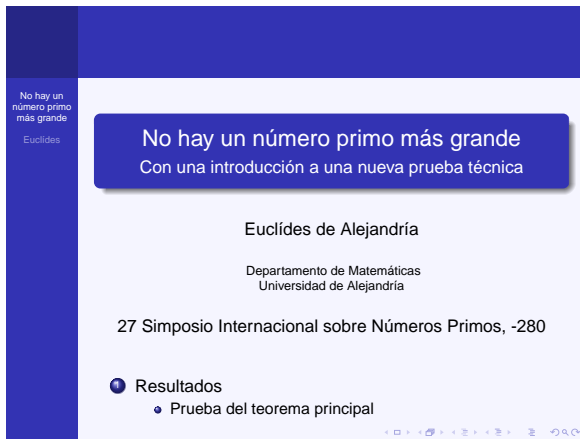
Se pueden dar las siguientes opciones:

- `hideallsubsections` causa que se muestren sólo las secciones en la barra lateral. Es útil si se necesita ahorrar espacio.
- `hideothersubsections` causa que se muestren sólo las subsecciones en la barra lateral. Es útil si se necesita ahorrar espacio.
- `left` (valor por defecto) pone la barra lateral a la izquierda.
- `right` pone la barra lateral a la derecha.
- `width=<dimensión>` establece el ancho de la barra lateral. No se crea la barra lateral si se pone 0.

Berkeley está en la costa occidental de E.U., cerca de San Francisco.

`\usetheme[<opciones>]{PaloAlto}`

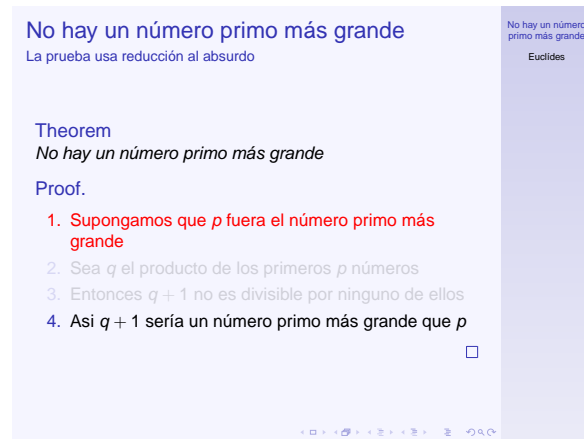
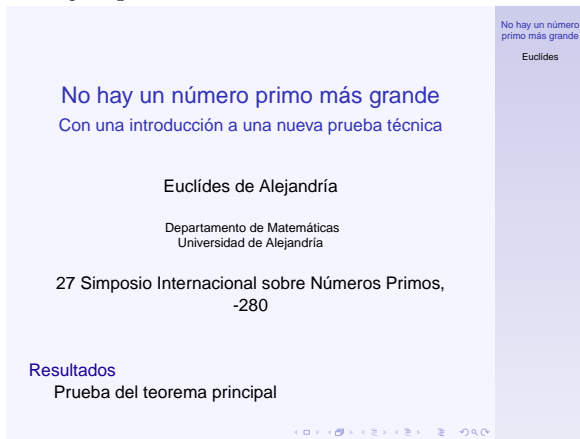
Ejemplo:



Una variación del tema Berkeley con áreas rectangulares menos dominantes. Se pueden dar las mismas *opciones* que el tema Berkeley. Palo Alto está cerca de San Francisco.

`\usetheme[opciones]{Goettingen}`

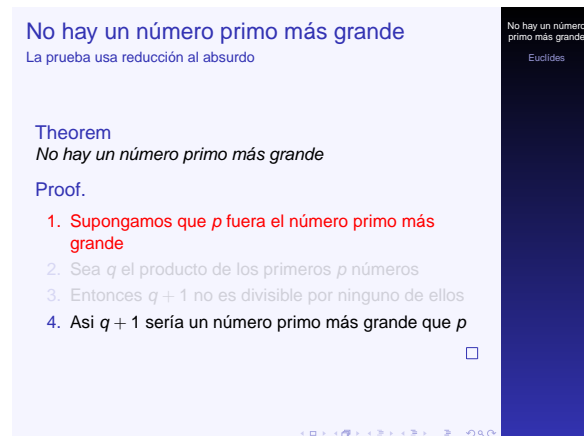
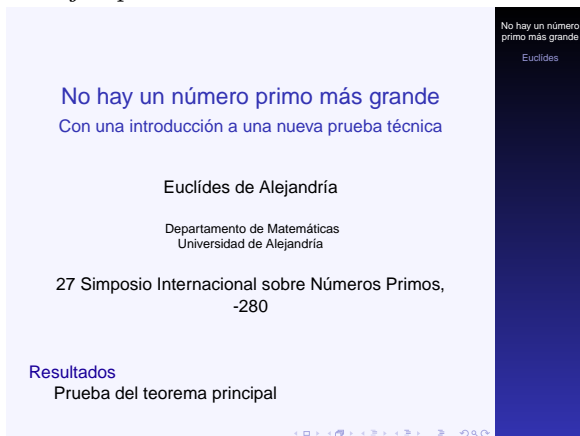
Ejemplo:



Un tema relativamente sobrio, útil para una plática más larga que exige una barra lateral conteniendo la tabla de contenido. Se pueden dar las mismas *opciones* que el tema Berkeley. Göttingen es un pueblo en Alemania.

`\usetheme[opciones]{Marburg}`

Ejemplo:

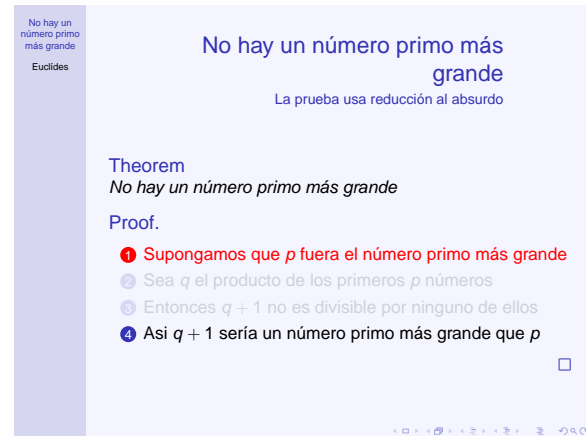
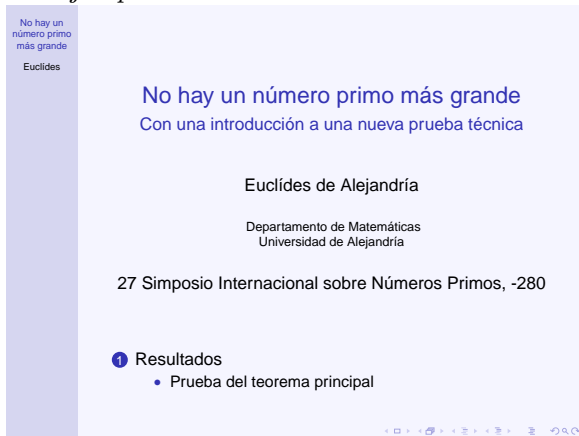


Una variación muy dominante del tema Goettingen. Se pueden dar las mismas *opciones*. Marburg

es un pueblo en Alemania.

`\usetheme[opciones]{Hannover}`

Ejemplo:



En este tema, la barra lateral a la izquierda se equilibra con el desplazamiento a la derecha de los encabezados. Hannover es un pueblo en Alemania.

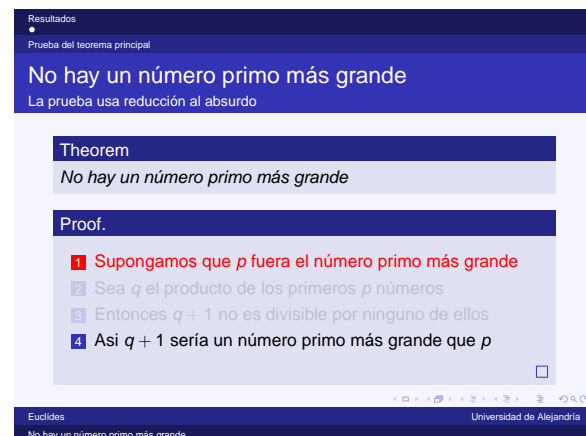
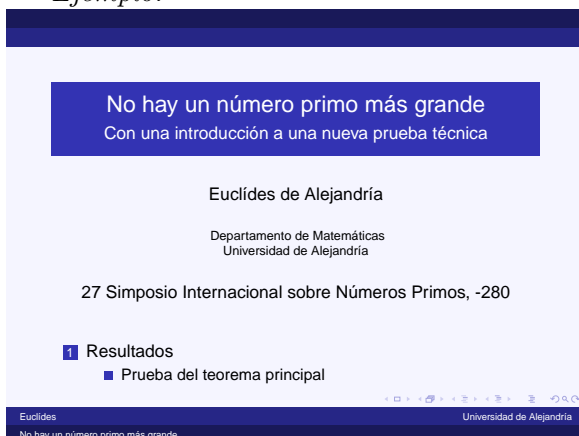
Se pueden dar las siguientes opciones:

- `hideallsubsections` causa que se muestren sólo las secciones en la barra lateral. Es útil si se necesita ahorrar espacio.
- `hideothersubsections` causa que se muestren sólo las subsecciones en la barra lateral. Es útil si se necesita ahorrar espacio.
- `width=<dimensión>` establece el ancho de la barra lateral.

14.5. Temas de presentación con un mini marco de navegación

`\usetheme[opciones]{Berlin}`

Ejemplo:



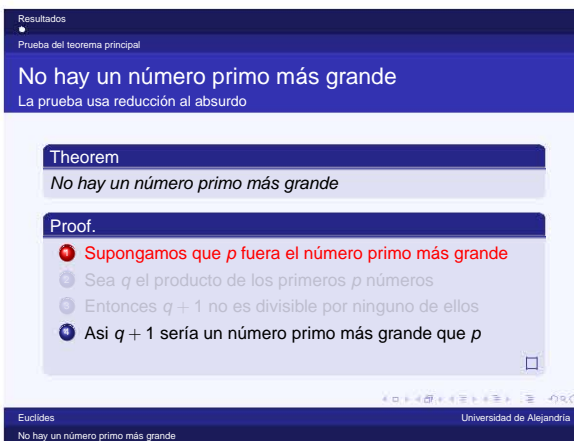
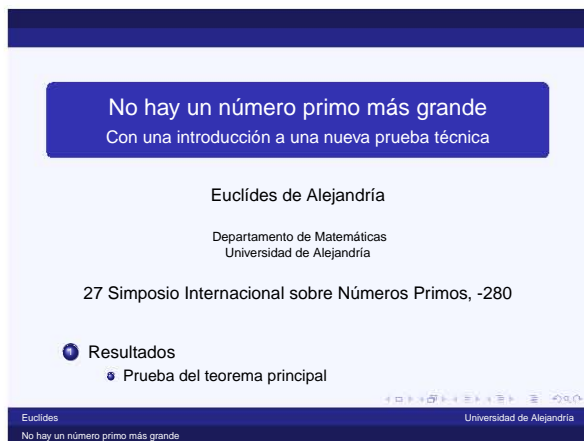
Un tema dominante con colores fuertes y áreas rectangulares dominantes. Los encabezados y pies de página dan mucha información, dejando poco espacio para las diapositivas reales. Este tema es útil para las conferencias donde es probable que el público no conozca el título de la plática o la persona que la está presentando. El tema se puede hacer menos dominante usando un tema de color diferente.

Se pueden dar las siguientes *opciones*:

- **compress** causa que los mini marcos en el encabezado sólo usen una línea. Es útil para ahorrar espacio.

`\usetheme[⟨opciones⟩]{Ilmenau}`

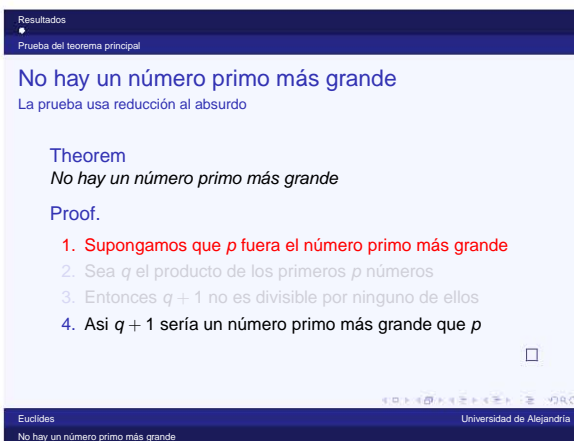
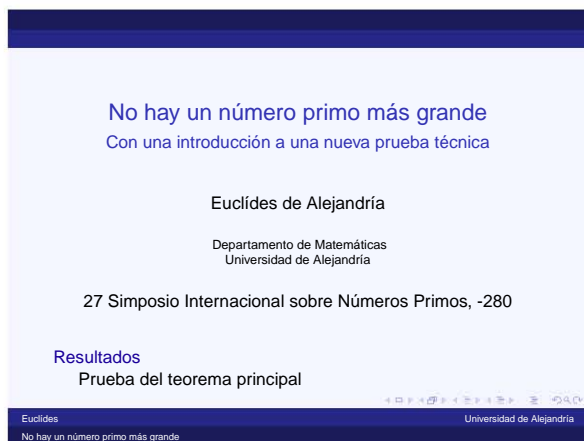
Ejemplo:



Una variación del tema **Berlín**. Se pueden dar las mismas *⟨opciones⟩*. Ilmenau es un pueblo en Alemania.

`\usetheme{Dresden}`

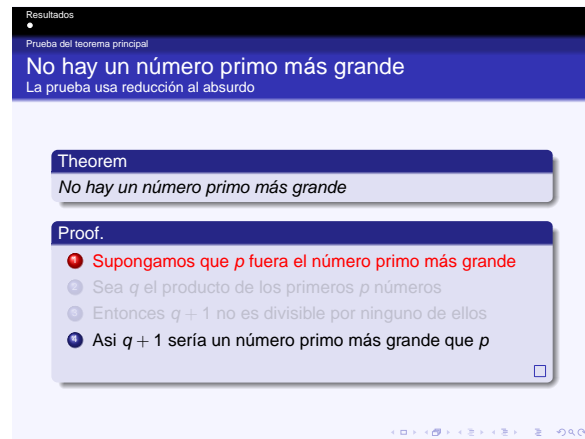
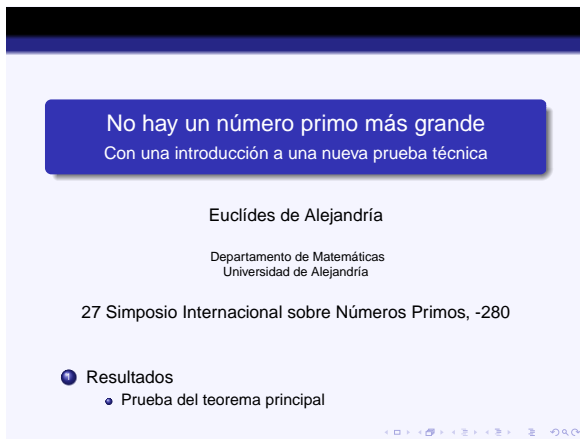
Ejemplo:



Una variación del tema **Berlín**, con una separación fuerte entre el material de navegación en la parte superior/inferior y el texto principal. Se pueden dar las mismas opciones. Dresden es un pueblo en Alemania.

`\usetheme{Darmstadt}`

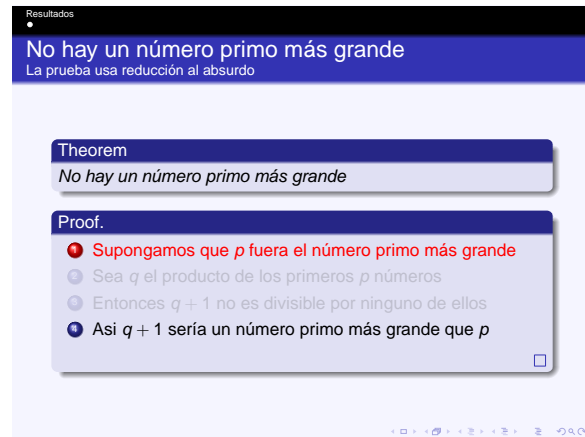
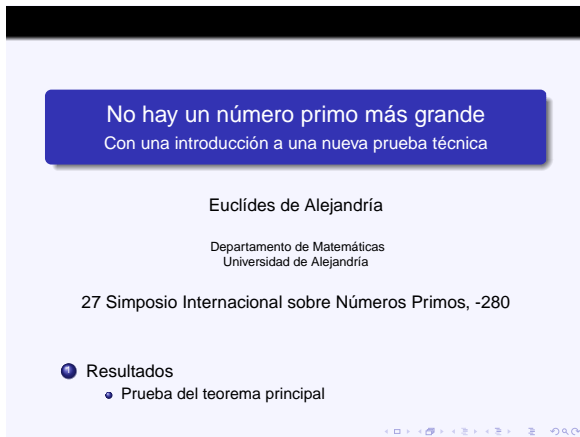
Ejemplo:



Un tema con una separación fuerte entre la parte superior de navegación y la parte informativa principal. Esta separación se puede disminuir usando un tema de color diferente. Darmstadt es un pueblo en Alemania.

`\usetheme{Frankfurt}`

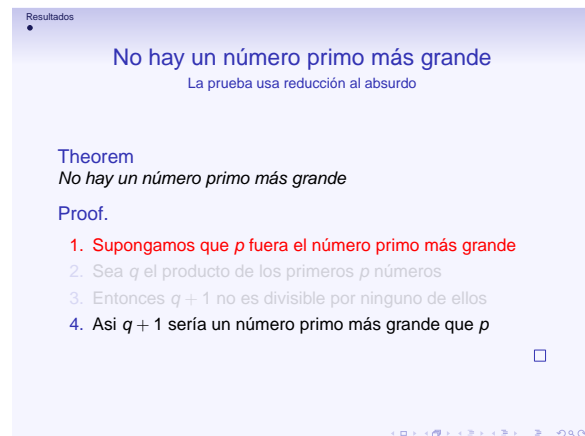
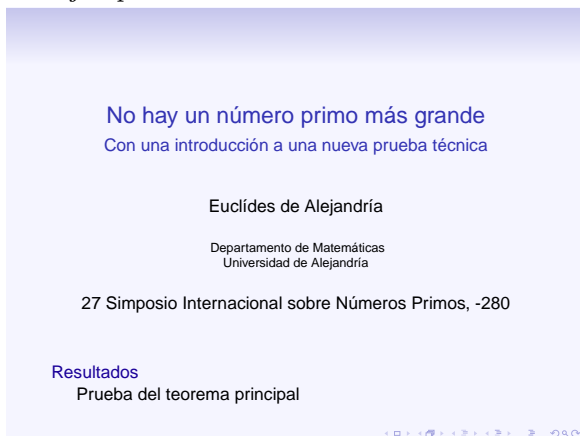
Ejemplo:



Una variación del tema Darmstadt, omite la información de la subsección por lo que es ligeramente menos de-sordenado. Frankfurt es un pueblo en Alemania.

`\usetheme{Singapore}`

Ejemplo:

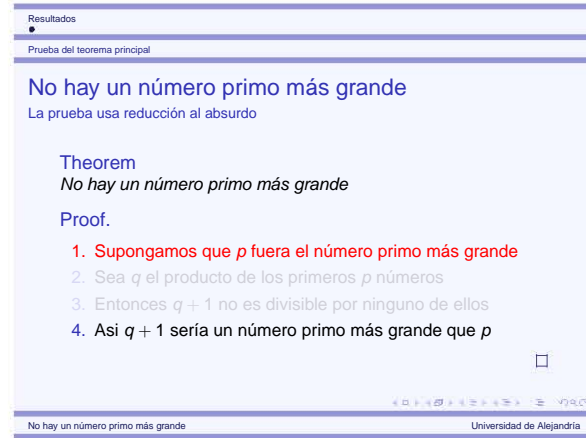
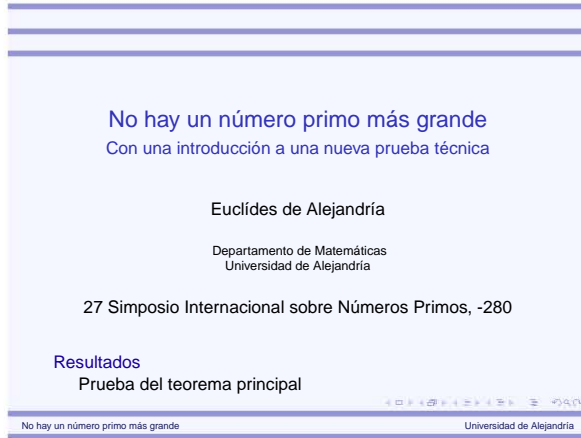


Un tema no muy sobrio con una parte de navegación que no domina. Singapur se localiza en el sureste

de Asia.

`\usetheme{Szeged}`

Ejemplo:

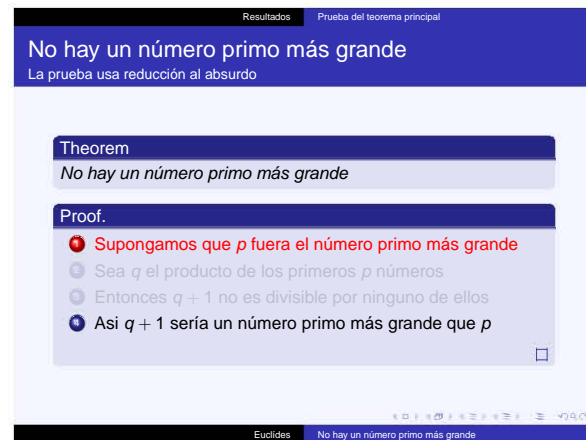
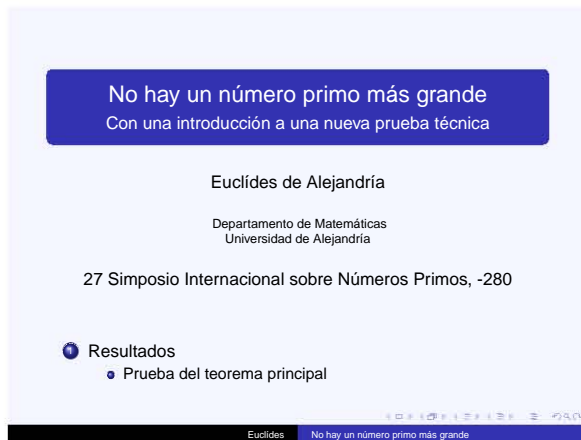


Un tema sobrio con líneas horizontales muy dominantes. Szeged está en la frontera sur de Hungría.

14.6. Temas de presentación con tablas de sección y subsección

`\usetheme{Copenhagen}`

Ejemplo:



Un tema no tan dominante. Este tema da información comprimida sobre la sección y subsección actual en la parte superior y sobre el título y el autor en la parte inferior. No se usan sombras dando a la presentación un aspecto plano. El tema se puede hacer menos dominante usando un tema de color diferente.

`\usetheme{Luebeck}`

Ejemplo:

Una variación del tema Copenhagen. Luebeck es un pueblo en el noreste de Alemania.

`\usetheme{Malmoe}`

Ejemplo:

Una variación más sobria del tema Copenhagen. Malmoe es un pueblo en el sur de Suecia.

`\usetheme{Warsaw}`

Ejemplo:

Una variación dominante del tema Copenhagen.

14.7. Temas de presentación incluidas para tener compatibilidad

Las anteriores versiones de BEAMER incluyeron algunos otros temas. Éstos todavía están disponibles para tener compatibilidad, aunque están implementados de manera diferente (instalan principalmente el color, la fuente, los temas internos y externos apropiados). Sin embargo, ellos pueden o no respetar ningún tema de color y no se les apoyará en el futuro. La siguiente lista muestra cuál de los temas nuevos se debe usar en lugar de los antiguos. (Al cambiar, se puede usar el tema de fondo `structurebold` con la fuente `onlysmall`.)

Tema antiguo Opciones de reemplazo
`none` Usar `compatibility`.
`bars` Intentar con `Dresden`.
`classic` Intentar con `Singapore`.
`lined` Intentar con `Szeged`.
`plain` Nada o intentar con `Pittsburgh`.
`sidebar` Intentar con `Goettingen` para la versión luminosa y `Marburg` para la oscura.
`shadow` Intentar con `Warsaw`.
`split` Intentar con `Malmoe`.
`tree` Intentar con `Montpellier` y para la versión con barras `Antibes` o `JuansLesPins`.

15. Temas Internos, Temas Externos y Plantillas

Esta sección discute los temas internos y externos que están disponibles en BEAMER. Estos temas instalan ciertas *plantillas* para los diferentes elementos de una presentación. El mecanismo de una plantilla se explica al final de la sección.

Antes de meternos con los detalles, convenimos alguna terminología para esta sección. En BEAMER un *elemento* es una parte de una presentación que posiblemente se edita de alguna manera especial. Ejemplos de elementos son el encabezado del marco, el nombre del autor o el pie de página. La apariencia de cada elemento se controla por una *plantilla* para este elemento. Las plantillas apropiadas se instalan por temas internos y externos, donde los temas *internos* sólo instalan plantillas para elementos que están típicamente “dentro del texto principal”, mientras que los temas *externos* instalan plantillas para los elementos “alrededor del texto principal”. Así, desde el punto de vista de las plantillas, no hay diferencia real entre los temas internos y externos.

15.1. Temas internos

Un tema interno instala plantillas que determinan cómo se van editar los elementos siguientes:

- Páginas de título y de parte.
- Ambientes itemize.
- Ambientes enumerate.
- Ambientes description.
- Ambientes block.
- Ambientes theorem y proof.
- Figuras y tablas.
- Pies de página.
- Entradas de bibliografía.

En los siguientes ejemplos se usan los temas de color `seahorse` y `rose` para mostrar donde y cómo se respetan los colores de fondo. Además, se han especificado los colores de fondo para todos los elementos que los respetan en el tema predefinido. En el tema de color predefinido, todas las áreas rectangulares grandes son transparentes.

```
\useinnertheme{default}
```

Ejemplo:

No hay un número primo más grande
Con una introducción a una nueva prueba técnica

Euclides de Alejandría
Departamento de Matemáticas
Universidad de Alejandría

27 Simposio Internacional sobre Números Primos, -280

Resultados
Prueba del teorema principal

No hay un número primo más grande
La prueba usa reducción al absurdo

Theorem
No hay un número primo más grande

Proof.

1. Supongamos que p fuera el número primo más grande
2. Sea q el producto de los primeros p números
3. Entonces $q + 1$ no es divisible por ninguno de ellos
4. Así $q + 1$ sería un número primo más grande que p

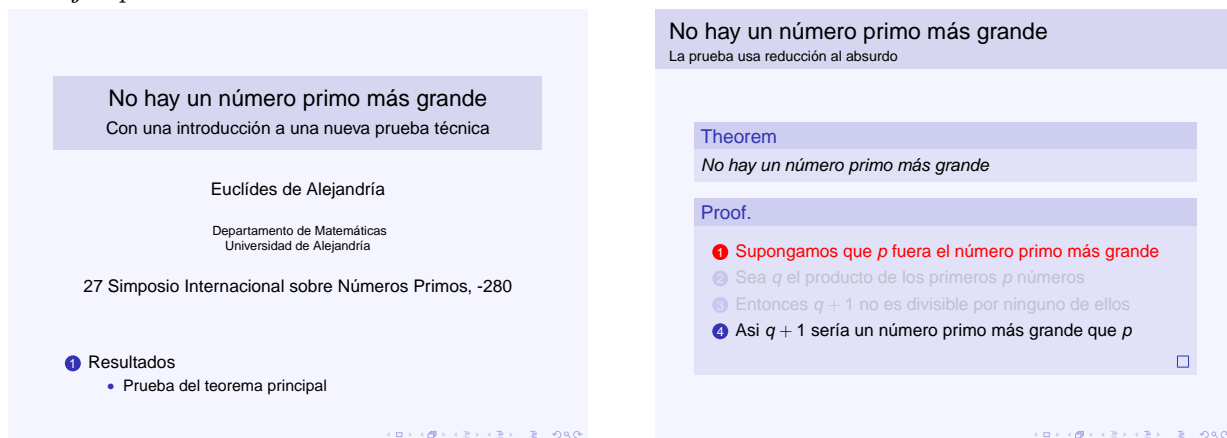
El tema de elemento predefinido es bastante sobrio. La única extravagancia es que en los ambientes `itemize` se usa un triángulo pequeño en lugar del punto usual.

En algunos casos el tema respetará las especificaciones de color del fondo para los elementos. Por ejemplo, si se establece el color de fondo para los títulos de bloque en verde, los títulos de bloque tendrán fondo verde. Actualmente a los siguientes elementos se les respetan las especificaciones de fondo:

- Los campos `title`, `author`, `institute` y `date` en la página de título.
- Los ambientes `block`, tanto para el encabezado como para el cuerpo.

`\useinnertheme{circles}`

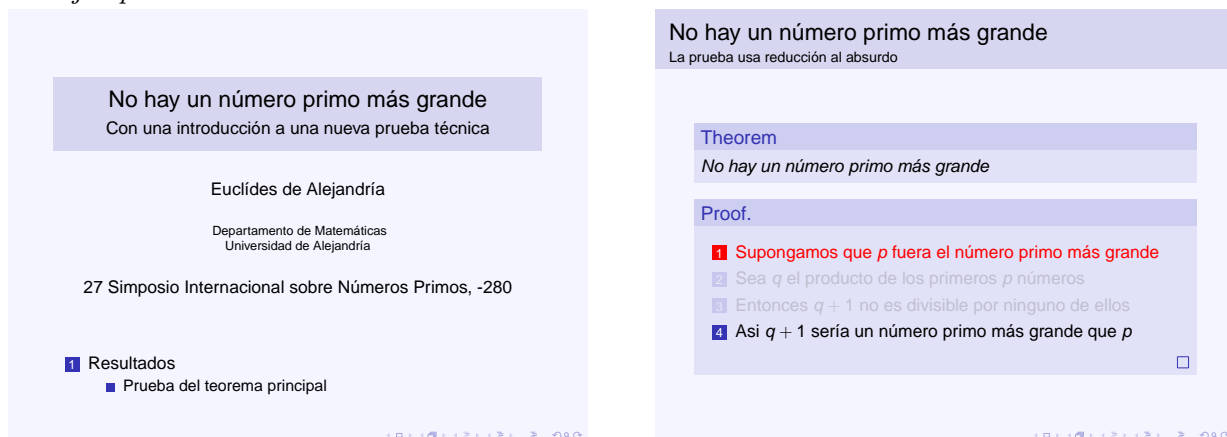
Ejemplo:



En este tema, los objetos de `itemize` y `enumerate`, así como las entradas en la tabla de contenidos, inician con un círculo pequeño.

`\useinnertheme{rectangles}`

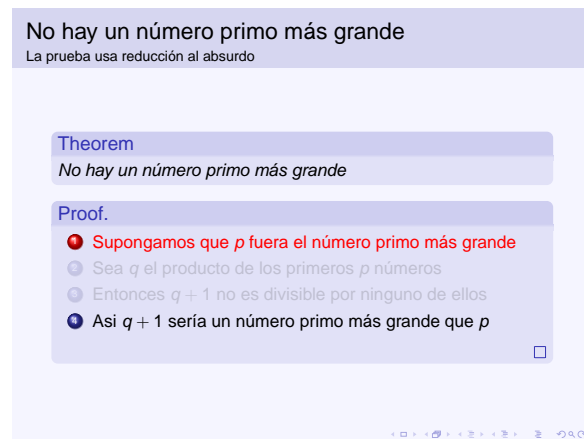
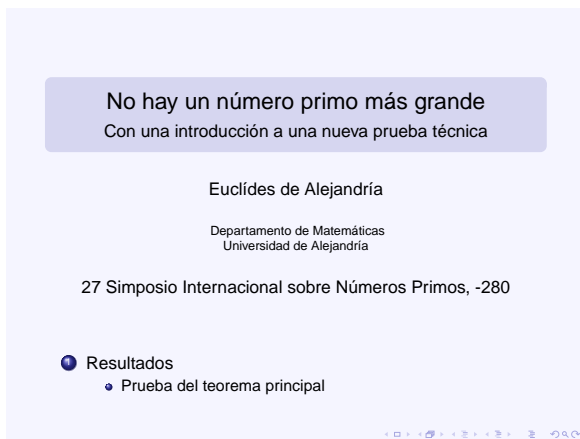
Ejemplo:



En este tema, los objetos de `itemize` y `enumerate`, así como las entradas en la tabla de contenidos, inician con un rectángulo pequeño.

`\useinnertheme[<opciones>]{rounded}`

Ejemplo:



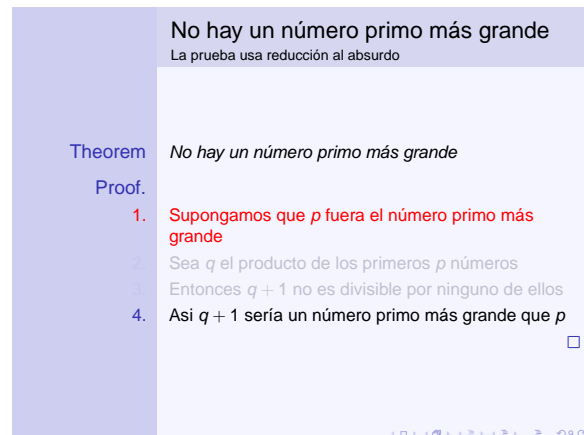
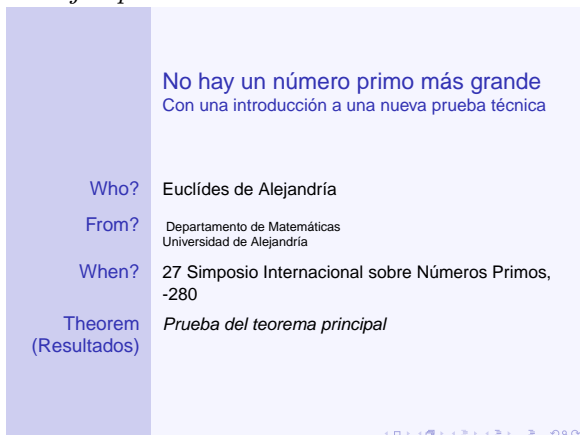
En este tema, los objetos de `itemize` y `enumerate`, así como las entradas en la tabla de contenidos, inician con una bola pequeña. Si se especifica un fondo para los bloques entonces se redondearán las esquinas de los rectángulos de fondo.

Se puede usar la siguiente opción:

- `shadow` agrega una sombra a todos los bloques.

`\useinnertheme{inmargin}`

Ejemplo:



La idea detrás de este tema es tener la información “estructurada” a la izquierda y la información “normal” a la derecha. Para este fin, se redefinen los bloques de tal manera que el título del bloque se muestra a izquierda y su cuerpo a la derecha.

El código que se usa para poner texto en el margen es un poco frágil. A menudo se puede necesitar ajustar el espacio “a mano”, esto se hará bajo nuestro propio riesgo.

Los objetos de `itemize` se redefinen de tal manera que aparecen a la izquierda. Sin embargo, sólo se puede cambiar su posición por medio de algunos parámetros de espaciado; por lo demás, el código usado para dibujar los objetos no cambia. Debido a esto, se puede cargar primero otro tema interno y luego cargar este tema.

Este es un tema interno “sucio” dado que molesta con cosas que un tema interno no debería. En particular, cambia el ancho de la barra lateral izquierda a un valor muy grande. Sin embargo, todavía se puede usar junto con la mayoría de los temas externos.

Es problemático usar columnas en este tema, la mayoría de las veces el resultado no es el esperado.

15.2. Temas externos

Un tema externo determina (bruscamente) la distribución global de los marcos. Especifica donde debe ir cualquier elemento de navegación (como una mini tabla de contenidos o un mini marco de navegación) y cómo deben verse. Típicamente, un tema externo especifica cómo se representan los siguientes elementos:

- El encabezado y pie de página.
- Las barras laterales.
- El logotipo.
- El título del marco.

Un tema externo no especifica cómo se representan algunas cosas, como el ambiente `itemize`, ese es trabajo de un tema interno.

En los siguientes ejemplos se usa el tema de color `seahorse`. Dado que el tema de color predefinido deja la mayoría de los fondos vacíos, con tal tema la mayoría de los temas externos se ven sin estructura.

`\useoutertheme{default}`

Ejemplo:

El tema de distribución predefinido es el más sobrio y minimalista. Jala el título del marco a la izquierda y no instala ningún encabezado ni pie de página. Sin embargo, este tema respeta el color de fondo indicado para el encabezado del marco. Si se especifica un color, detrás del encabezado del marco se pone una barra que ocupa el ancho de la página. Se ignora el color de fondo del subtítulo del marco.

`\useoutertheme{infolines}`

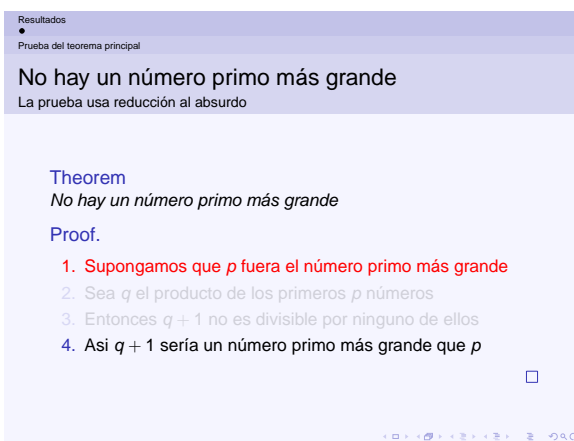
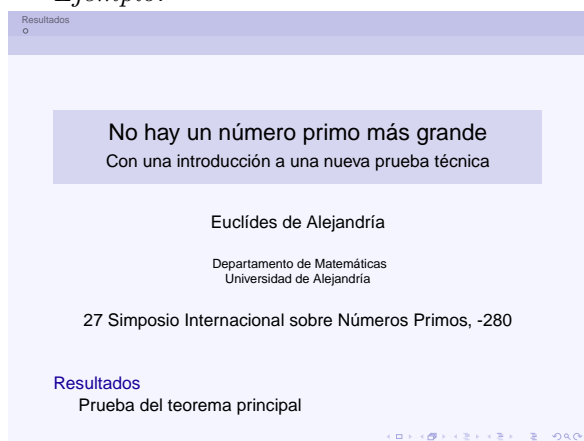
Ejemplo:

Este tema instala un encabezado que muestra la sección y subsección actual. Instala un pie de página que muestra el nombre del autor, la institución, el título de la presentación, la fecha actual y un contador de marcos. Este tema usa poco espacio.

Los colores usados en el encabezado y los pies de página se dibujan de `palette primary`, `palette secondary` y `palette tertiary`.

```
\useoutertheme[⟨opciones⟩]{miniframes}
```

Ejemplo:



Este tema instala un encabezado que muestra una barra de navegación horizontal. Esta barra contiene una entrada para cada sección de la presentación. Debajo de cada entrada de sección se muestran círculos pequeños que representan los diferentes marcos en la sección. Los marcos se colocan a manera de subsecciones, es decir, hay una línea de marcos para cada subsección. Si se da la opción de clase `compress`, los marcos en cambio se colocan en una sola fila para cada sección. Las barras de navegación obtienen su color de `section in head/foot`.

Debajo de la barra de navegación, hay una línea que muestra el título de la subsección actual. El color se obtiene de `subsection in head/foot`.

En la parte inferior hay dos líneas que contienen información como el nombre del autor, la institución o el título del artículo. Lo que se muestra exactamente depende de las `⟨opciones⟩` dadas. Los colores se obtienen de los colores-BEAMER apropiados como `author in head/foot`.

Si se establece el color de fondo `separation line`, en la parte superior e inferior del encabezado como del pie de página y entre la barra de navegación y el nombre de la subsección, se dibujan unas líneas de separación. Esta línea de separación tendrá una altura de 3pt. Se puede incluso tener un control mas fino sobre los colores de las líneas de separación al establecer colores apropiados como `lower separation line head`.

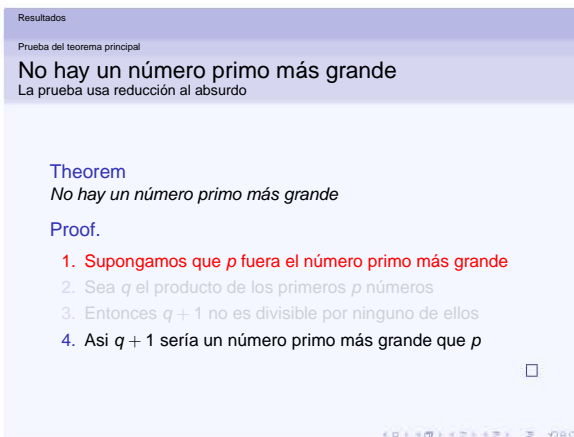
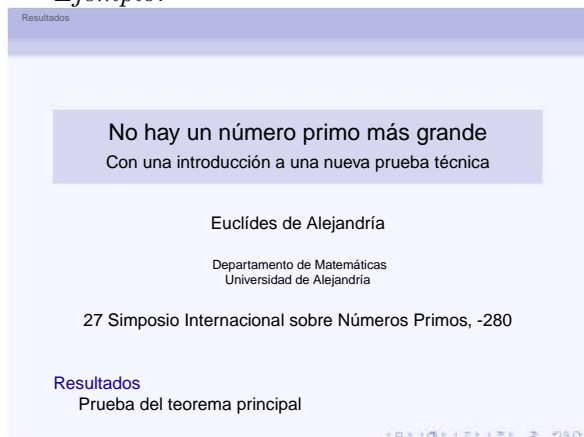
Se pueden dar las siguientes opciones:

- `footline=empty` suprime el pie de pagina (valor por defecto).
- `footline=authorinstitute` muestra el nombre del autor y del instituto en el pie de página.
- `footline=authortitle` muestra el nombre del autor y el título de la plática en el pie de página.
- `footline=institutetitle` muestra el instituto y el título de la plática en el pie de página.
- `footline=authorinstitutetitle` muestra el nombre del autor, del instituto y el título de la plática en el pie de página.

- `subsection=(true or false)` muestra o suprime la línea que muestra la subsección en el encabezado. Se muestra por defecto.

`\useoutertheme[⟨opciones⟩]{smoothbars}`

Ejemplo:



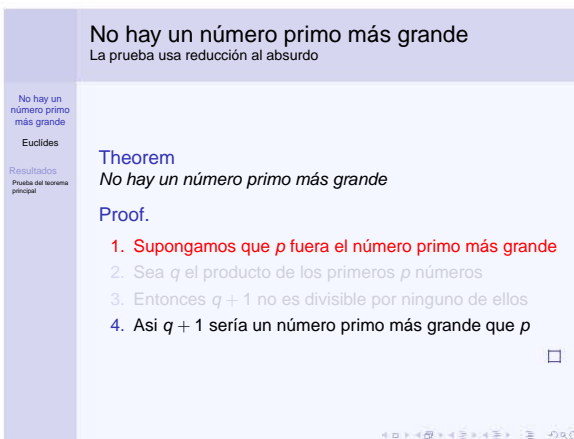
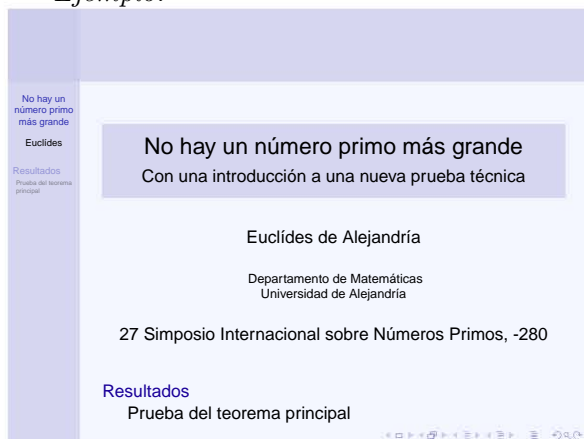
Se comporta de manera muy parecida al tema `miniframes`, por lo menos respecto al encabezado. La única diferencia es que se instalan transiciones suaves entre los colores de fondo de la barra navegación, la barra (opcional) para el nombre de la subsección y el fondo de título del marco. No se crea pie de página. Se puede conseguir el pie de página del tema `miniframes` cargando primero ese tema y luego el tema `smoothbars`.

Se pueden dar las siguientes opciones:

- `subsection=(true or false)` muestra o suprime la línea que muestra la subsección en el encabezado. Se muestra por defecto.

`\useoutertheme[⟨opciones⟩]{sidebar}`

Ejemplo:



En esta distribución se muestra una barra lateral que contiene una pequeña tabla de contenidos con la sección, subsección o subsubsección actual resaltada. El encabezado del marco se centra verticalmente en un área rectangular en la parte superior, que siempre ocupa la misma cantidad de espacio en todos los marcos. Finalmente, el logotipo se muestra en la “esquina” que resulta de la barra lateral y el rectángulo del encabezado del marco.

Hay varias maneras de modificar la distribución usando las `⟨opciones⟩`. Si se pone el ancho de la barra lateral en `0pt` entonces no se muestra, produciendo una distribución en la que el encabezado del marco no “tiembla” dado que siempre ocupa la misma cantidad de espacio en todas las diapositivas. Recíprocamente, si se pone la altura del área rectangular que contiene el encabezado del marco en `0pt`, éste no

se usa y el encabezado del marco se inserta de manera normal (ocupando tanto espacio como necesite en cada diapositiva).

El color de fondo de la barra lateral se toma de `sidebar`, el color de fondo del encabezado del marco de `frametitle` y el color de fondo de la esquina del logotipo de `logo`.

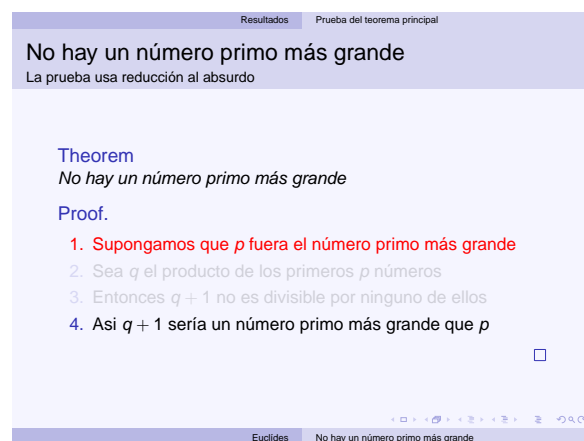
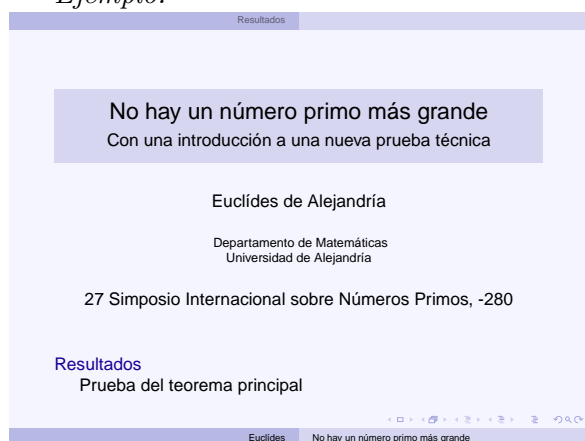
Los colores de las entradas en la tabla de contenidos se toman del color-BEAMER en `section in sidebar` y `section in sidebar current`, así como los correspondientes colores-BEAMER para las subsecciones. Si una entrada no cabe en una línea automáticamente se divide en varias líneas.

Se pueden dar las siguientes opciones:

- `height=<dimensión>` especifica la altura del rectángulo de encabezado del marco, si se pone en `0pt` entonces no se crea y el encabezado del marco se inserta de manera normal. El valor por defecto es 2.5 veces la altura de la línea base para la fuente en el encabezado del marco. Así, hay suficiente espacio para un título de dos líneas más un subtítulo de una línea.
- `hideothersubsections` causa que se supriman en la tabla de contenidos todas las subsecciones excepto aquéllas de la sección actual. Es útil si se tienen muchas subsecciones.
- `hideallsubsections` causa que se supriman en la tabla de contenidos todas las subsecciones.
- `left` pone la barra lateral en el lado izquierdo. Notar que en una cultura que lee de izquierda a derecha esto es lo primero que ve la gente. Notar además que la tabla de contenidos normalmente *no* es la parte más importante del marco, así que no necesariamente se quiere que la gente la vea al principio. No obstante, es el valor por defecto.
- `right` pone la barra lateral en el lado derecho.
- `width=<dimensión>` especifica el ancho de la barra lateral, se suprime si se pone en `0pt`. El valor por defecto es 2.5 veces la altura de la línea base para la fuente en el encabezado del marco.

`\useoutertheme{split}`

Ejemplo:

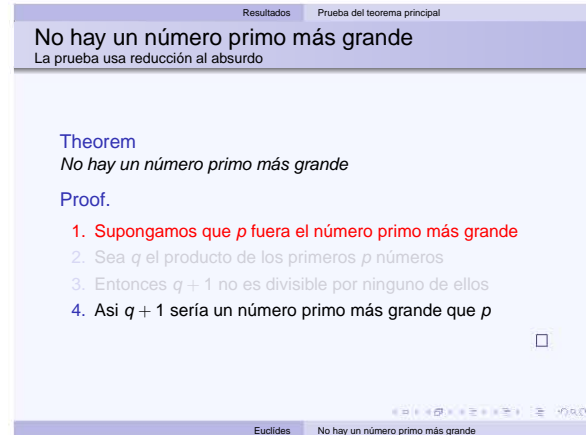
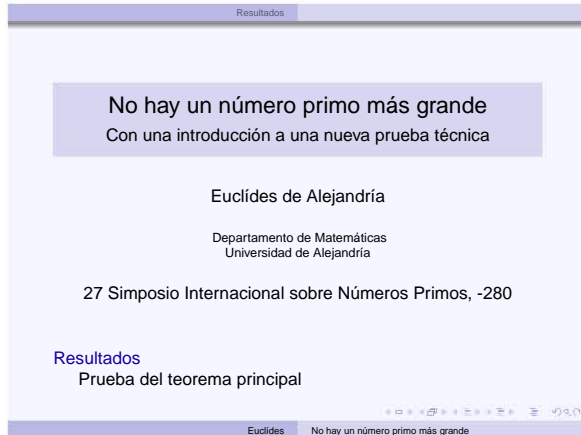


Este tema instala un encabezado que en su parte izquierda muestra las secciones de la plática y en la derecha las subsecciones de la sección actual. Si se da la opción de clase `compress`, las secciones y subsecciones se ponen en una línea; normalmente hay una línea por sección o subsección.

El pie de página muestra el nombre del autor a la izquierda y el título de la plática a la derecha. Los colores se toman de `palette primary` y `palette fourth`.

`\useoutertheme{shadow}`

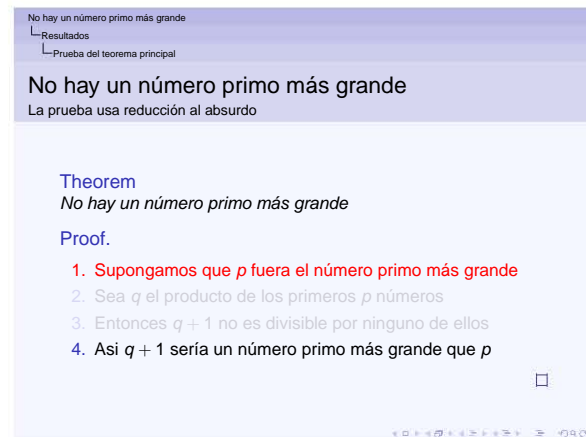
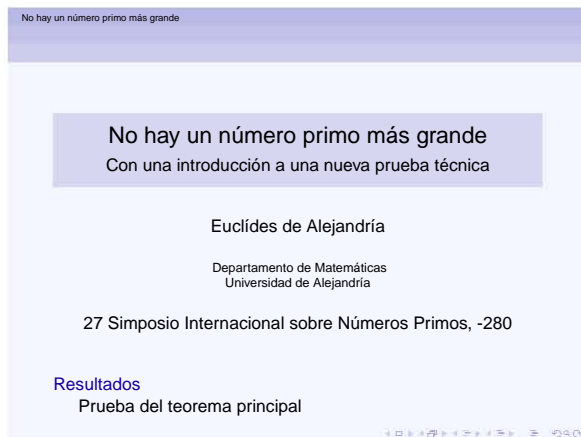
Ejemplo:



Este tema de distribución extiende el tema `split` poniendo un sombreado horizontal detrás del título del marco añadiendo un poco de “sombra” en la parte inferior del encabezado.

```
\useoutertheme[<opciones>]{tree}
```

Ejemplo:



En esta distribución, el encabezado contiene tres líneas que muestran el título de la plática actual, la sección actual en esta plática y la subsección actual en la sección. Los colores se toman de `title in head/foot`, `section in head/foot` y `subsection in head/foot`.

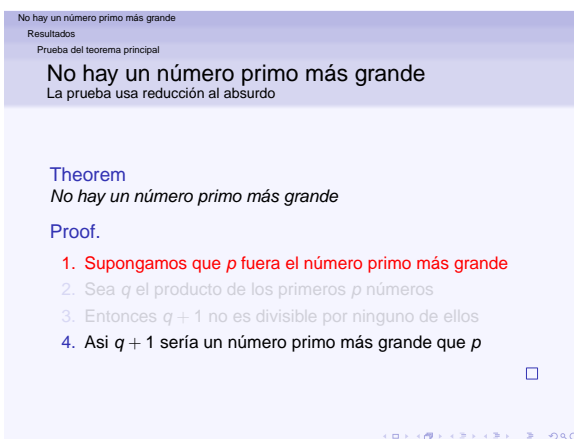
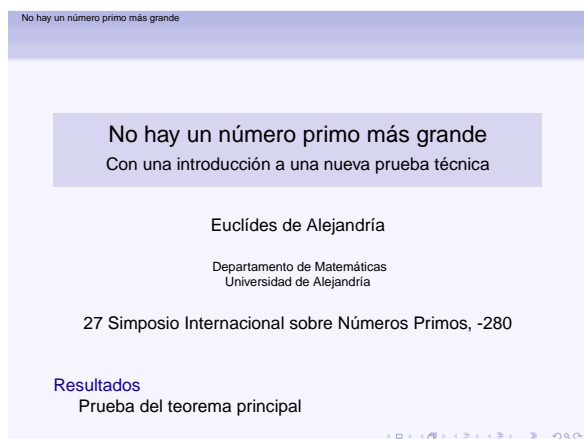
Además, se muestran líneas de separación de altura 3pt *si* se pone el fondo de `separation line`. Se puede tener un control mas fino sobre los colores de estas líneas poniendo `upper separation line head` y `lower separation line head`.

Se pueden dar las siguientes opciones:

- `hooks` causa que se dibujen pequeños “ganchos” delante de la entradas de sección y subsección, para aumentar la apariencia de árbol.

```
\useoutertheme{smoothtree}
```

Ejemplo:



Es similar a la distribución `tree`. La diferencia principal es que los colores de fondo cambian suavemente.

15.3. Cambiando las plantillas para elementos de una presentación

Esta sección explica cómo funciona el administrador de plantillas de BEAMER.

Resumen del administrador de plantillas de Beamer

Si sólo se quiere modificar la apariencia de pocos elementos, no es necesario crear un nuevo tema interno o externo completo. En cambio, se puede modificar la plantilla apropiada.

Una plantilla indica cómo se edita un elemento de una presentación. Por ejemplo, la plantilla `frametitle` determina donde se pone el título del marco, qué fuente usar, etcétera. Como lo sugiere el nombre, una plantilla se especifica escribiendo el código `LATEX` exacto que también se usaría al editar a mano un solo título de marco. Sólo que, en lugar del título real, se usa el comando `\insertframetitle`.

Ejemplo: Si se quiere editar el encabezado del marco en rojo, centrado y en negritas y si sólo se va a editar un encabezado de marco a mano, se podría hacer así:

```
\begin{frame}
  \begin{centering}
    \color{red}
    \textbf{El t\{i}tulo de este marco.}
  \par
\end{centering}
```

Bla, bla.

```
\end{frame}
```

Para editar de esta manera el título del marco en todas las diapositivas, se cambia la plantilla de título del marco como sigue:

```
\setbeamertemplate{frametitle}
{
  \begin{centering}
    \color{red}
    \textbf{\insertframetitle}
  \par
\end{centering}
}
```

Se usa el siguiente código para obtener el efecto deseado:

```
\begin{frame}
  \frametitle{El título de este marco.}
  Bla, bla.
\end{frame}
```

Al representar el marco, BEAMER usa el código de la plantilla de título del marco para editar el título del marco y reemplazar cada ocurrencia de `\insertframetitle` por el título del marco actual.

Podemos llevar este ejemplo un poco más allá. Sería mejor si no se tuviera que “hardwire” el color del título de marco, excepto si este color se pudiera especificar independientemente del código para la plantilla. De esta manera, un tema de color podría cambiar este color. Dado que este es un problema común en la mayoría de las plantillas, BEAMER establece de manera automática el color-BEAMER `frametitle` cuando se usa la plantilla `frametitle`. Así que se puede quitar el comando `\color{red}` si en algún punto se pone el color-BEAMER `frametitle` en rojo.

```
\setbeamercolor{frametitle}{fg=red}
\setbeamertemplate{frametitle}
{
  \begin{centering}
    \textbf{\insertframetitle}
  \par
  \end{centering}
}
```

A continuación, también se puede hacer “temable” la fuente. Al igual que el color, la fuente-BEAMER `frametitle` se instala antes de que se edite la plantilla `frametitle`. Así, se debe reescribir el código como sigue:

```
\setbeamercolor{frametitle}{fg=red}
\setbeamerfont{frametitle}{series=\bfseries}
\setbeamertemplate{frametitle}
{
  \begin{centering}
    \insertframetitle\par
  \end{centering}
}
```

Usuarios, temas o cualquiera puede ahora cambiar fácilmente el color o la fuente del título de marco sin tener que meterse con el código usado para editarlo.

ARTÍCULO En este modo, la mayoría de los mecanismos de plantilla se desactivan y no tienen efecto. Sin embargo, están disponibles algunas plantillas. Si este es el caso, se indica de manera especial.

Aquí están algunos consejos que podrían ser útiles al establecer una plantilla:

- Usualmente se desea copiar código de una plantilla existente. El código a menudo cuida de algunos aspectos que tal vez uno no ha pensado todavía. Los temas internos y externos predefinidos podrían ser puntos de partida útiles. El archivo `beamerbaseauxtemplates.sty` también contiene interesantes plantillas auxiliares.
- Al copiar código de otra plantilla e insertarlo en el preámbulo del documento (no en otro archivo de estilo), se puede haber “activado” el carácter (@). Para hacerlo, se agrega el comando `\makeatletter` antes del comando `\setbeamertemplate` y después de `\makeatother`.

- La mayoría de las plantillas que tienen que ver con los componentes del marco (encabezados, barras laterales, etc.) sólo se pueden cambiar en el preámbulo. Otras si se pueden cambiar durante el documento.
- La altura de las plantillas de encabezado y pie de página se calcula automáticamente. Esto se hace editando las plantillas y luego “echando una mirada” a sus alturas. Este recalcu se hace justo al inicio del documento, *después* de que se han cargado todos los paquetes e incluso *después* de que éstos han ejecutado su inicialización `\AtBeginDocument`.
- A menudo es un poco molesto conseguir que las cajas encajen bien dentro de cualquier plantilla. Se puede consultar el libro de T_EX para los detalles gloriosos sobre “Fabricando Cajas”. Si el encabezado es simple, también se puede intentar poner todo en un ambiente `pgfpicture` que hace más fácil su colocación.

Usando las plantillas de beamer

Como usuario de la clase BEAMER uno típicamente no “usa” o “invoca” las plantillas por sí mismo, directamente. Por ejemplo, la plantilla de título de marco se invoca automáticamente por BEAMER en alguna parte profunda dentro del proceso de edición del marco. Lo mismo se cumple en la mayoría de las otras plantillas. Sin embargo, si por cualquier razón, uno mismo desea invocar una plantilla, se puede usar el siguiente comando.

```
\usebeamertemplate***{<nombre del elemento>}
```

Si no se pone ningún asterisco, el texto del *<nombre del elemento>* se inserta directamente en la posición actual. Este texto se debe especificar previamente usando el comando `\setbeamertemplate`. No se inserta ningún texto si este comando no fue llamado antes.

Ejemplo:

```
\setbeamertemplate{miplantilla}{correcta}
```

...

La respuesta es `\usebeamertemplate{miplantilla}`.

Si se agrega un asterisco pasan tres cosas. Primero, la plantilla se pone dentro de un grupo-T_EX y por eso limita la mayoría de los efectos laterales de comandos usados dentro de la plantilla. Segundo, dentro de este grupo se usa el color-BEAMER llamado *<nombre del elemento>* y se selecciona el color del primer plano. Tercero, también se usa la fuente-BEAMER *<nombre del elemento>*. Normalmente esta es la mejor versión a usar.

Si se agrega un segundo asterisco pasa casi lo mismo que en el caso anterior. Sin embargo, además, el color se usa con el comando `\setbeamercolor*`. Esto causa que si no se especifica ningún primer plano o fondo especial por el color-BEAMER *<nombre del elemento>* entonces los colores se reestablecen al color del texto normal. Así, en este caso de doble asterisco, se garantiza que el color usado para la plantilla sea independiente del color que estaba actualmente en uso.

Finalmente, agregar un tercer asterisco también causará que se añada un asterisco al comando `\setbeamerfont*`. Esto también causa que la fuente usada para la plantilla sea reestablecida a texto normal, a menos que la fuente-BEAMER *<nombre del elemento>* especifique otra cosa. Ésta (con tres asteriscos) es la versión disponible con “mayor protección”.

```
\ifbeamertemplateempty{<nombre plantilla beamer>}{<ejecutado si vacío>}{<ejecutado en caso contrario>}
```

Este comando verifica si una plantilla está definida y tiene un texto no vacío. Si el texto está vacío o la plantilla no está definida en absoluto, se ejecuta *<ejecutado si vacío>*. En caso contrario, se ejecuta *<ejecutado en caso contrario>*.

```
\expandbeamertemplate{<nombre plantilla beamer>}
```

Este comando hace lo mismo que `\usebeamertemplate{<nombre plantilla beamer>}`. La diferencia es que realiza una expansión directa y no examina asterisco alguno. Esto es importante dentro de, por ejemplo, un `\edef`. Si no se sabe la diferencia entre `\def` y `\edef`, no es necesario este comando.

Estableciendo plantillas de Beamer

Para establecer una plantilla-BEAMER se puede usar el siguiente comando:

```
\setbeamertemplate{<nombre del elemento>}[<opción predefinida>]{<args>}
```

En el caso más simple, si no se da ninguna *<opción predefinida>*, el *<args>* debe ser un solo argumento y el texto de la plantilla *<nombre del elemento>* se establece como este texto. Se usa este texto en la invocación posterior de la plantilla con el comando `\usebeamertemplate`.

Ejemplo:

```
\setbeamertemplate{respuesta}{correcta}
```

...

Tu respuesta es `\usebeamertemplate*{respuesta}`.

Si se especifica una *<opción predefinida>*, este comando se comporta un poquito diferente. En este caso, alguien usó el comando `\defbeamertemplate` al predefinir una plantilla para nosotros. Al dar el nombre de esta plantilla predefinida como el parámetro opcional *<opción predefinida>*, se causa que la plantilla *<nombre del elemento>* sea establecida como esta plantilla.

Ejemplo: `\setbeamertemplate{bibliography item}[book]` causa que los objetos de la bibliografía se vuelvan pequeños iconos de libro. Este comando causa una llamada subsiguiente de `\usebeamertemplate{bibliography item}` para insertar el código predefinido al insertar un libro.

Algunas opciones de plantilla predefinidas toman parámetros por sí mismos. En tal caso, los parámetros se dan como *<args>*.

Ejemplo: La *<opción predefinida>* `grid` para la plantilla `background` toma un argumento opcional:

```
\setbeamertemplate{background}[grid][step=1cm]
```

El argumento entre corchetes es el argumento opcional.

En las descripciones de elementos, si hay una posible *<opción predefinida>*, la descripción muestra cómo se puede usar la *<opción predefinida>* junto con sus argumentos, pero se omite `\setbeamertemplate{xxxx}`. Así, el ejemplo anterior se documentaría en la descripción del elemento de fondo así:

- `[grid]` [*<opciones de salto>*] causa que una malla . . .

```
\addtobeamertemplate{<nombre del elemento>}{<pre-texto>}{<post-texto>}
```

Este comando agrega el *<pre-texto>* antes del texto que está actualmente instalado como la plantilla *<nombre del elemento>* y agrega el *<post-texto>* después de él. Esto permite la modificación de plantillas existentes en forma limitada.

Ejemplo: Los siguientes comandos tienen el mismo efecto:

```
\setbeamertemplate{miplantilla}{¡Hola mundo!}
```

```
\setbeamertemplate{miplantilla}{mundo}
```

```
\addtobeamertemplate{miplantilla}{¡Hola }{!}
```

Se anula cualquier adición si se instala una nueva plantilla. Por otro lado, se puede usar repetidamente este comando para agregar múltiples cosas.

```
\defbeamertemplate<especificación de modo>*<nombre del elemento>{<opción predefinida>}[<número argumento>][<argumento opcional predefinido>]{<texto predefinido>}[<acción>]{<comando de acción>}
```

Este comando instala una *opción predefinida* para la plantilla *<nombre del elemento>*. Una vez que se ha usado este comando, los usuarios pueden acceder a la plantilla predefinida usando el comando `\setbeamertemplate`.

Ejemplo: `\defbeamertemplate{itemize item}{doble flecha}{\Rightarrow$}`

Después de que se invoca el comando anterior, los dos comandos siguientes tienen el mismo efecto:

```
\setbeamertemplate{itemize item}{\Rightarrow$}
\setbeamertemplate{itemize item}[doble flecha]
```

A veces, una plantilla predefinida necesita conseguir un argumento cuando se le instala. Por ejemplo, supongamos que se quiere definir una plantilla predefinida que dibuja un cuadrado como objeto de una lista y además hacer configurable el tamaño de este cuadrado. En este caso, se puede especificar el *<número argumento>* de la opción predefinida de la misma manera que se hace para el comando `\newcommand`:

```
\defbeamertemplate{itemize item}{cuadrado}[1]{\hrule width #1 height #1}
% Lo siguiente tiene el mismo efecto:
\setbeamertemplate{itemize item}[cuadrado]{3pt}
\setbeamertemplate{itemize item}{\hrule width 3pt height 3pt}
```

Respecto al comando `\newcommand`, también se puede especificar un *<argumento opcional predefinido>*:

```
\defbeamertemplate{itemize item}{square}[1][1ex]{\hrule width #1 height #1}
% Lo siguiente tiene el mismo efecto:
\setbeamertemplate{itemize item}[square][3pt]
\setbeamertemplate{itemize item}{\hrule width 3pt height 3pt}
% También lo siguiente:
\setbeamertemplate{itemize item}[square]
\setbeamertemplate{itemize item}{\hrule width 1ex height 1ex}
```

La versión con asterisco del comando instala la opción de plantilla predefinida, pero luego inmediatamente llama `\setbeamertemplate` para esta opción. Esto es útil para las plantillas predefinidas. Si hay algunos argumentos necesarios, éstos se ponen como `\relax`.

En ciertos casos, si se elige una opción de plantilla predefinida, no sólo se desea que se instale el texto de plantilla, sino que también se lleven a cabo ciertas “acciones” extra. Por ejemplo, se debe definir un sombreado de tal manera que no se debe redefinir cada vez que se va a usar posteriormente. Para implementar tales “acciones”, se puede usar el argumento opcional *<acción>* siguiendo a la palabra clave `[action]`. Así, después del uso normal de `\defbeamertemplate` se agrega el texto `[action]` y luego cualquier comando, el cual se debe ejecutar una vez que se selecciona la *<opción predefinida>* para el comando `\setbeamertemplate`.

Ejemplo:

```
\defbeamertemplate{background canvas}{misombreado}[2]
{
  \pgfuses shading{misombreado}% bastante simple
}
[action]
{
  \pgfdeclareverticalshading{misombreado}{\the\paperwidth}
  {color(0cm)=(#1); color(\the\paperheight)=(#2)}
}
...
```

```

\setbeamertemplate{background canvas}{misombreado}{red!10}{blue!10}
% Define el sombreado misombreado justo aquí. Subsecuentes llamadas a
% \usebeamertemplate{background canvas} producirá
% ‘‘\pgfuses shading{misombreado}’’.

```

ARTÍCULO Normalmente no tiene efecto en este modo. Sin embargo, si se da una *especificación de modo*, este comando se aplica para los modos especificados. Así, este comando se comporta como el comando `\`, que también tiene la especificación de modo implícita *presentation* si no se da ninguna otra especificación.

Ejemplo: `\defbeamertemplate{miplantilla}{default}{algo}` no tiene efecto en el modo `article`.

Ejemplo: `\defbeamertemplate<article>{miplantilla}{default}{algo}` no tiene efecto en los modos `presentation`, pero tiene un efecto en el modo `article`.

Ejemplo: `\defbeamertemplate<all>{miplantilla}{default}{algo}` se aplica a todos los modos.

A menudo es útil tener acceso a la misma opción de plantilla vía diferentes nombres. Para esto, se puede usar el siguiente comando para crear seudónimos:

```

\defbeamertemplatealias{<nombre del elemento>}{<nuevo nombre de opción predefinida>}{<nombre existente de opción predefinida>}

```

Causa que las dos opciones predefinidas tengan el mismo efecto.

No hay ninguna relación de herencia entre las plantillas como la hay para los colores y fuentes. Esto es debido al hecho de que las plantillas para un elemento raramente tienen sentido para otra. Sin embargo, a veces ciertos elementos se “comportan similarmente” y a uno le gustaría que un comando `\setbeamertemplate` se aplicara a un conjunto entero de plantillas vía herencia. Por ejemplo, se podría querer que `\setbeamertemplate{items}[circle]` cause que todos los objetos usen la opción `circle`, aunque los efectos para `itemize item` tan opuestos a los de `itemize subsubitem` y tan opuestos a los de `enumerate item` deben ser ligeramente diferentes.

El mecanismo de plantilla-BEAMER implementa una forma simple de herencia vía *plantillas padre*. En las descripciones de elementos, las plantillas padre se indican vía una marca entre paréntesis.

```

\defbeamertemplateparent{<nombre de plantilla padre>}[<nombre de opción predefinida>]{<lista de plantilla hijo>}[<número de argumento>][<argumento opcional predefinido>]{<argumentos para los hijos>}

```

El efecto de este comando es que siempre que alguien llama `\setbeamertemplate{<nombre de plantilla padre>}{<args>}`, se invoca el comando `\setbeamertemplate{<nombre de plantilla hijo>}{<args>}` para cada *nombre de plantilla hijo* en la *lista de plantilla hijo*.

Los *argumentos para los hijos* entran en juego si se invoca el comando `\setbeamertemplate` con un nombre de opción predefinida (no necesariamente el mismo que *nombre de opción predefinida*).

Si `\setbeamertemplate` se llama con algún nombre de opción predefinido, los hijos se llaman en cambio con los *argumentos para los hijos*.

Ejemplo: Lo que sigue es un típico uso simple:

```

\defbeamertemplateparent{itemize items}{itemize item,itemize subitem,itemize subsubitem}{}

```

% El siguiente comando tiene el mismo efecto que los tres comandos de abajo:

```

\setbeamertemplate{itemize items}[circle]

```

```

\setbeamertemplate{itemite item}[circle] %en realidad se añade el argumento ‘‘vacío’’

```

```

\setbeamertemplate{itemize subitem}[circle]

```

```
\setbeamertemplate{itemize subsubitem}[circle]
```

Ejemplo: En el siguiente caso, se pasa a los hijos un argumento:

```
\defbeamertemplateparent{sections/subsections in toc shaded}  
{section in toc shaded,subsection in toc shaded}[1][20]  
{[#1]}
```

% El siguiente comando tiene el mismo efecto que los dos comandos que lo siguen:

```
\setbeamertemplate{sections/subsection in toc shaded}[default][35]
```

```
\setbeamertemplate{section in toc shaded}[default][35]
```

```
\setbeamertemplate{subsection in toc shaded}[default][35]
```

% Otra vez:

```
\setbeamertemplate{sections/subsection in toc shaded}[default]
```

```
\setbeamertemplate{section in toc shaded}[default][20]
```

```
\setbeamertemplate{subsection in toc shaded}[default][20]
```

En detalle, pasa lo siguiente: cuando se encuentra `\setbeamertemplate` para una plantilla padre, BEAMER primero chequea si sigue una opción predefinida. Si no, lee un solo argumento y se llama `\setbeamertemplate` para todos los hijos de esta plantilla. Si hay un conjunto de opciones de plantilla predefinida, BEAMER evalúa los *argumentos para los hijos*. Puede contener parámetros como #1 o #2. Estos parámetros se llenan con los argumentos que siguen la llamada de `\setbeamertemplate` para la plantilla padre. El número de argumentos debe ser el número dado como *número argumento*. También se puede especificar un argumento opcional de la manera usual. Una vez que se han calculado los *argumentos para los hijos*, se llama `\setbeamertemplate` para todos los hijos de la plantilla predefinida y con los argumentos calculados.

Uno se puede preguntar que pasa cuando ciertas opciones predefinidas toman un cierto número de argumentos, pero otra opción predefinida toma un número diferente de argumentos. En este caso, el mecanismo antes mencionado no puede diferenciar entre las opciones predefinidas y no es nada claro cual o incluso cuántos argumentos deben estar contenidos en *argumentos para los hijos*. Por esta razón, se puede dar el argumento opcional *nombre de opción predefinido* al llamar `\defbeamertemplateparent`. Si se especifica este argumento opcional, la paternidad de la plantilla sólo se aplica a este particular *nombre de opción predefinido*.

Así, si alguien llama `\setbeamertemplate` para este *nombre de opción predefinido*, se usan los *argumentos para los hijos* dados. Para otros nombres de opción predefinidos se usa posiblemente una definición diferente. Se puede imaginar que omitiendo la opción *nombre de opción predefinido* significa “estos *argumentos para los hijos* se aplican a todos los nombres de opción predefinidos que no han sido especialmente definidos de manera diferente”.

16. Colores

El administrador de color de BEAMER nos permite especificar el color de cada elemento (a saber, el color de las entradas de sección en la tabla de contenidos, o el color de las entradas en una mini tabla de contenidos en una barra lateral). Aun cuando el sistema es bastante poderoso su uso no es común. Para simplificar el uso del sistema de color, debemos considerar un tema predefinido el cual se encarga de todo.

En lo que sigue, primero se explican los temas de color. El resto de la sección consiste en explicar cómo funciona internamente el administrador de color. Es necesario leer estas secciones sólo si se desea escribir temas de color propios; o si se está a gusto con los temas predefinidos pero se insiste en que el texto matemático debe ser de un color rosa encantador.

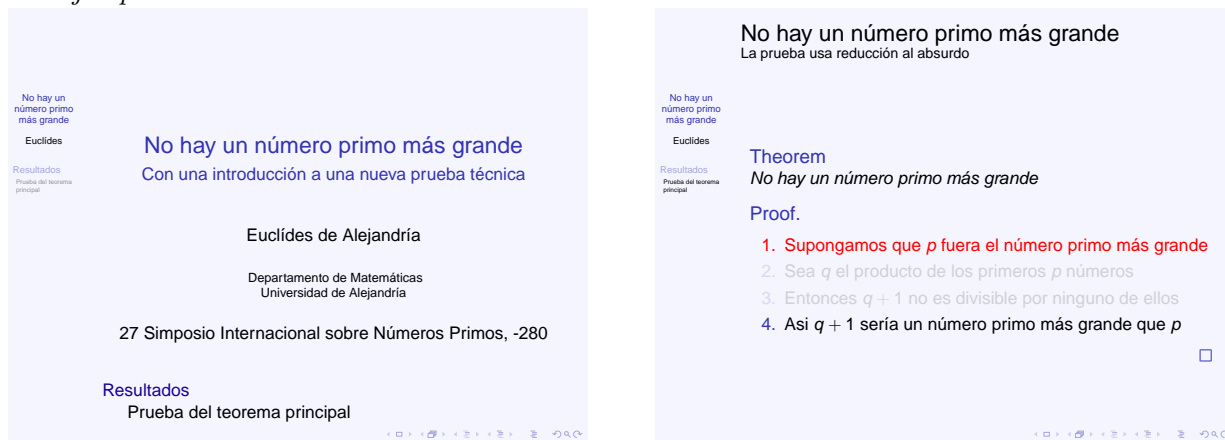
16.1. Temas de color

Para mostrar también los efectos de los diferentes temas de color sobre la barra lateral, en los ejemplos siguientes se usan los temas de color junto con el tema exterior *sidebar*.

Temas de color predefinidos y de propósito especial

```
\usecolortheme{default}
```

Ejemplo:



El tema de color `default` es muy sobrio. Instala colores poco especiales e incluso con menos fondos. El color predefinido establece las relaciones de padre predefinidas entre los diferentes colores-BEAMER.

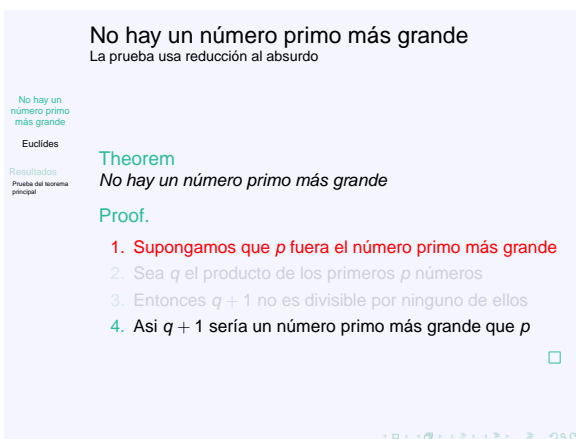
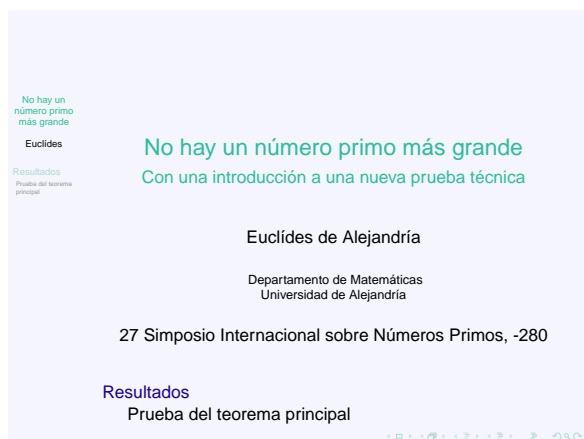
Los principales colores que se establecen en el tema de color `default` son los siguientes:

- `normal text` es negro sobre fondo blanco
- `alerted text` es rojo
- `example text` es de color verde oscuro (verde con 50 % de negro)
- `structure` se establece como una versión ligera de MidnightBlue (mas precisamente, 20 % rojo, 20 % verde y 70 % de azul).

Se usa este tema para una presentación ridícula. Dado que este tema se carga por defecto, no se puede “recargar” después de haber cargado otro tema de color.

```
\usecolortheme[opciones]{structure}
```

Ejemplo:



El ejemplo se creó usando `\usecolortheme[named=SeaGreen]{structure}`.

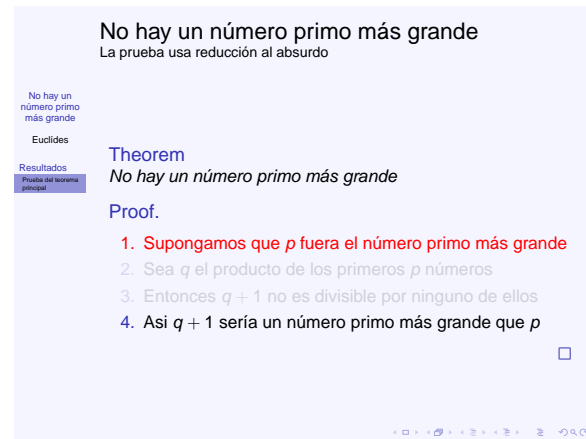
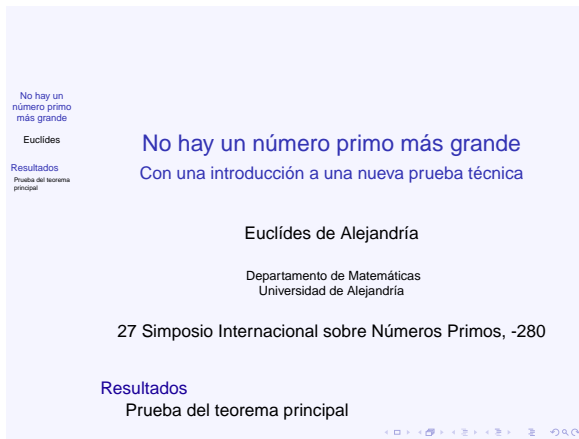
Este tema ofrece una manera conveniente de cambiar el color que se usa para elementos estructurales. Más precisamente, sólo cambia el primer plano del color-BEAMER `structure`. Esto también se puede lograr directamente invocando la función `\setbeamercolor`, pero este tema de color hace las cosas un poco más fáciles.

El tema ofrece varias *opciones* que se pueden usar para especificar el color a usar por los elementos estructurales:

- `rgb={\langle tripleta rgb \rangle}` establece el primer plano de `structure` como la tripleta rojo-verde-azul especificada. Los argumentos deben ser números decimales entre 0 y 1. Por ejemplo, `rgb={0.5,0,0}` proporciona un color rojo oscuro.
- `RGB={\langle tripleta rgb \rangle}` hace lo mismo que `rgb`, excepto que los números van entre 0 y 255. Por ejemplo, `RGB={128,0,0}` proporciona un rojo oscuro.
- `cmk={\langle tupla cmyk \rangle}` establece el primer plano de `structure` como la tupla cian-magenta-amarillo-negro especificada. Los argumentos deben ser números decimales entre 0 y 1. Por ejemplo, `cmyk={0,1,1,0.5}` proporciona un rojo oscuro.
- `cmk={\langle tripleta cmy \rangle}` es similar a `cmk` pero no se especifica el componente negro.
- `hsb={\langle tripleta hsb \rangle}` establece el primer plano de `structure` como la tripleta tono-saturación-brillo especificada. Los argumentos deben ser números decimales entre 0 y 1. Por ejemplo, `hsb={0,1,0.5}` proporciona un rojo oscuro.
- `named={\langle nombre del color \rangle}` pone el primer plano de `structure` como el color nombrado. Este color se debe definir previamente usando el comando `\DefineNamedColor`. Al agregar la opción de clase `xcolor=dvipsnames` se instala una larga lista de nombres estándar. Para la lista ver el archivo `dvipsnam.def`.

`\usecolortheme{sidebartab}`

Ejemplo:



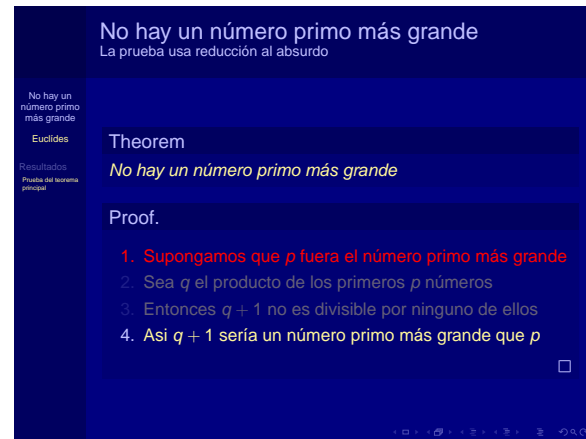
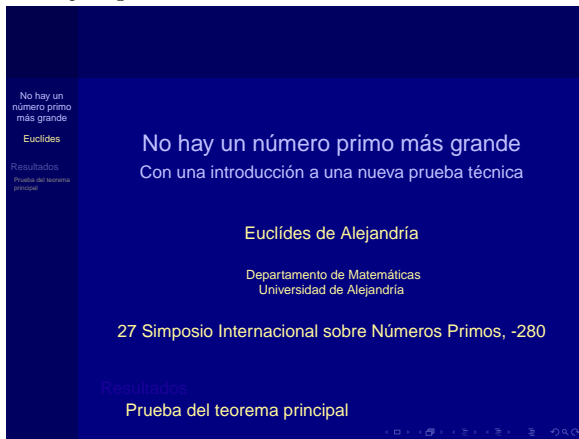
Este tema cambia los colores en una barra lateral de tal manera que la entrada actual en una tabla de contenido mostrada allí, se resalta mostrando un fondo diferente detrás de ella.

Temas de color completos

Un tema de color “completo” es un tema de color que especifica totalmente todos los colores para todas las partes de un marco. Instala colores específicos y no deriva los colores, digamos, del color-BEAMER structure. Ocurre que los temas de color completos tienen nombres de aves.

```
\usecolortheme{albatross}
```

Ejemplo:



El tema de color es un tema “oscuro” o “invertido” que usa amarillo sobre azul como colores principales. El tema de color también instala para los bloques un color de fondo ligeramente más oscuro, lo que es necesario para temas de presentación que usan sombras, pero que es indeseable para todos los demás temas de presentación. Se pueden quitar los fondos para bloques usando el tema de color lily junto con este tema.

Al usar un tema claro-sobre-oscuro como éste, hay que saber que existen ciertas desventajas:

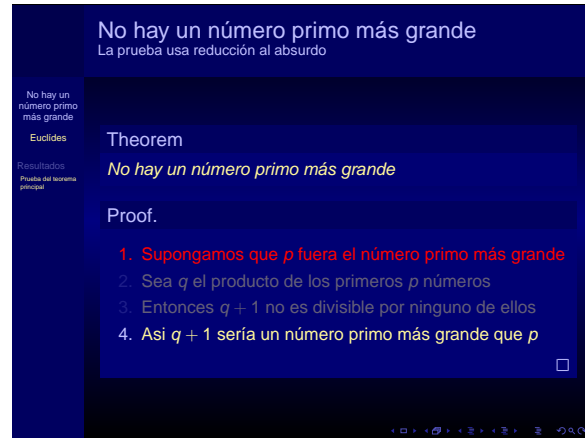
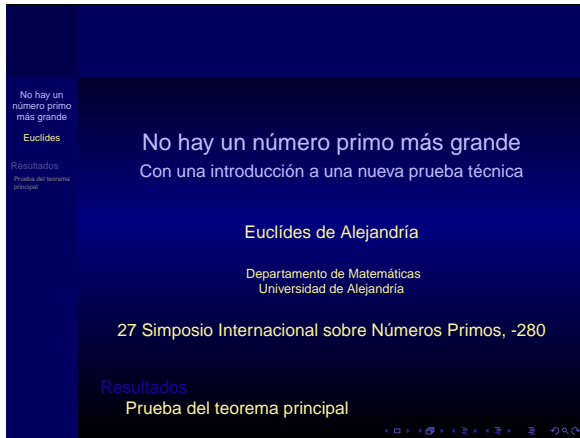
- Si se ha “oscurecido” el auditorio donde se da la conferencia, si se usa tal tema al público se le dificulta tomar o leer notas.
- Cuando el cuarto se pone muy oscuro, la pupila se dilata dificultando el enfoque del ojo. Esto puede dificultar la lectura del texto.
- En el mejor de los casos, es difícil imprimir tales diapositivas.

Por otro lado, a menudo una presentación claro-sobre-oscuro parece ser más “elegante” que un sencillo oscuro-sobre-claro.

Se pueden dar las siguientes opciones:

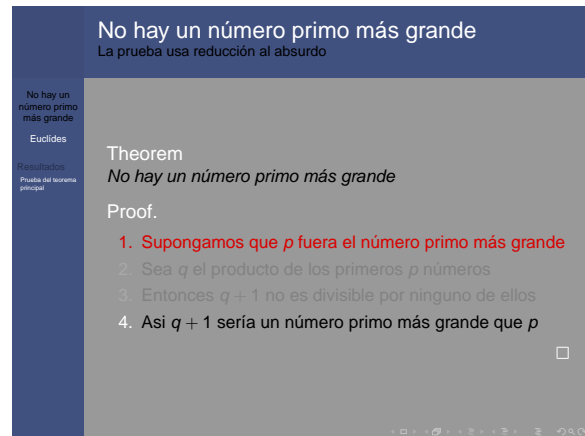
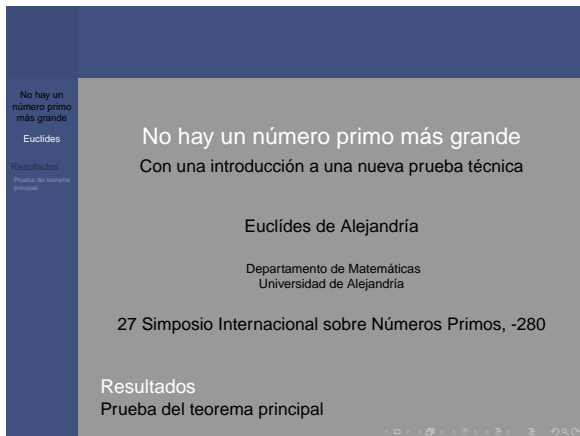
- **overlystylish** instala un lienzo de fondo que en cierta modo es demasiado elegante. Al usar esta opción, probablemente sea muy buena idea usar también el tema de color **lily**.

Ejemplo: La opción **overlystylish** junto con el tema de color **lily**:



`\usecolortheme{beetle}`

Ejemplo:

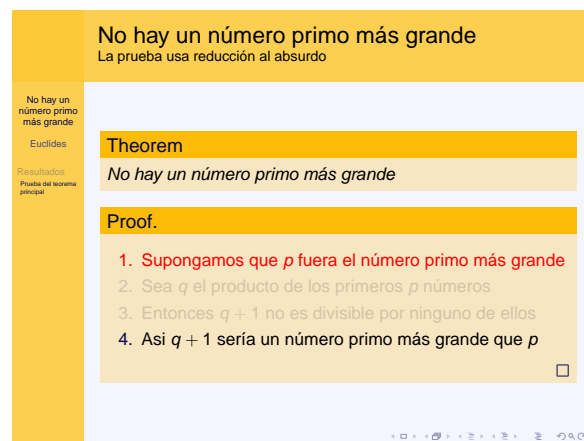
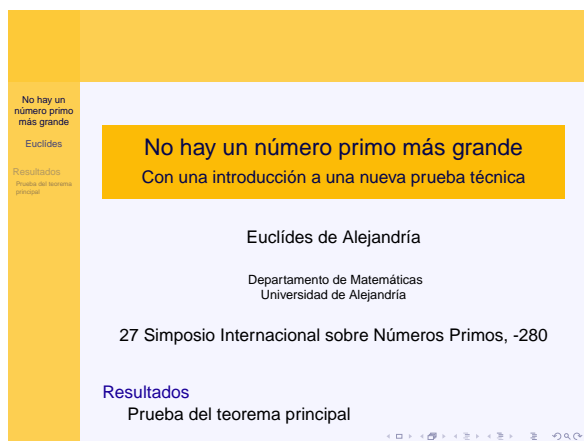


El principal “tema detrás de este tema” es usar texto blanco y negro sobre fondo gris. El texto blanco se usa para dar especial énfasis, el texto negro para el texto normal. Sin embargo, el “material exterior” como el encabezado y el pie de pagina usan un color azulado. Para cambiar este color, se debe cambiar el fondo de `palette primary`.

Se debe tener mucho cuidado con este tema ya que los contrastes blanco/gris y negro/gris son mucho más bajos respecto a otros temas. Hay que asegurarnos que el contraste sea suficiente para la presentación real. Se puede cambiar el fondo “grisáceo” cambiando el fondo de `normal text`.

`\usecolortheme{crane}`

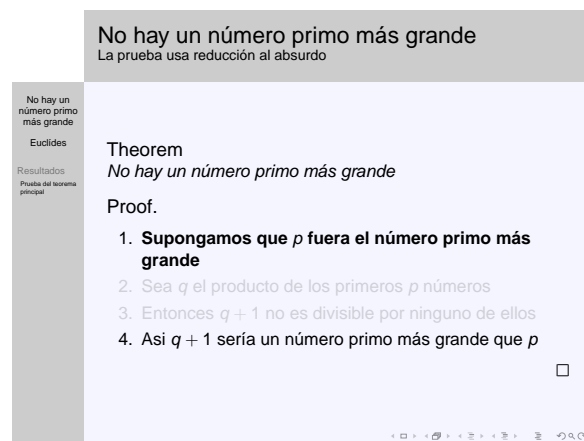
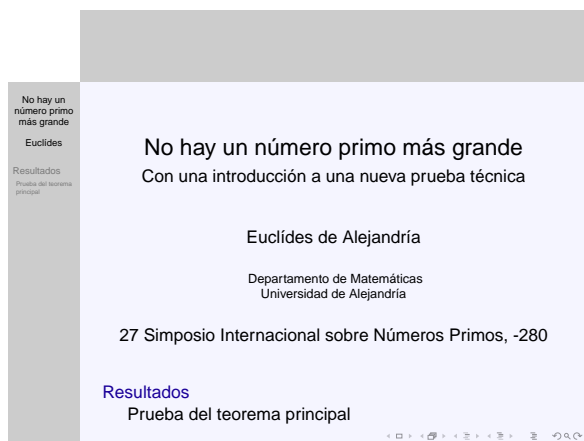
Ejemplo:



Este tema usa los colores de Lufthansa cuyo logotipo es una grulla. Sin embargo, *no* es un tema oficial para la compañía.

`\usecolortheme{dove}`

Ejemplo:

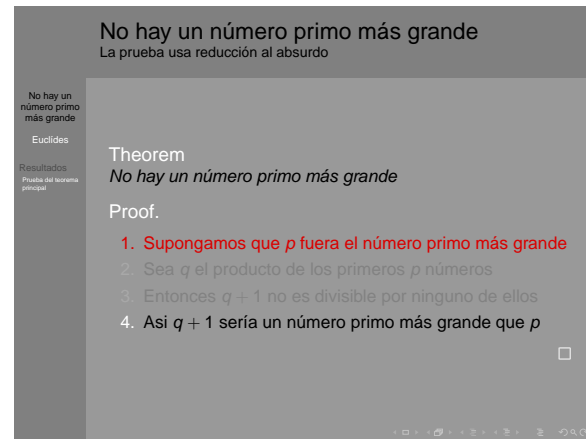
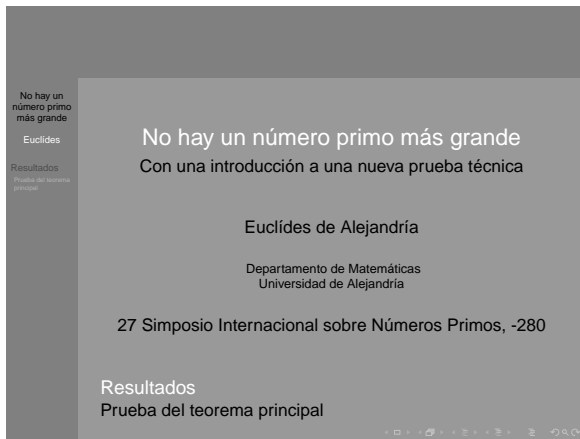


Este tema casi es un tema de blanco y negro, útil para crear presentaciones que se imprimen fácilmente en una copiadora de blanco y negro. En ciertos casos ineludibles, el tema usa escala de grises pero nunca color. También cambia la fuente del texto resaltado a negritas.

Al usar este tema se debe considerar el uso de la opción de clase `gray`, que asegura que todos los colores se conviertan a una escala de grises. También se debe considerar el uso del tema de fuente `structurebold`.

`\usecolortheme{fly}`

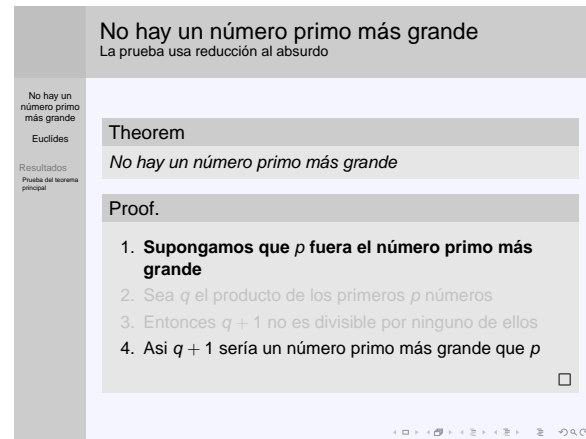
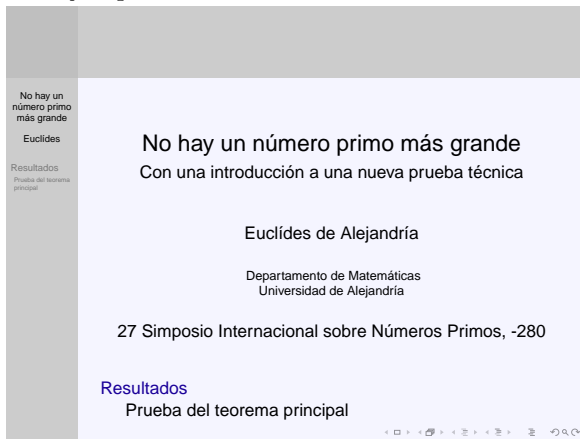
Ejemplo:



Este tema es la versión “que sigue” de `beetle` y todo el tiempo usa blanco/negro/gris. No va particularmente bien con temas que usan sombras.

`\usecolortheme{seagull}`

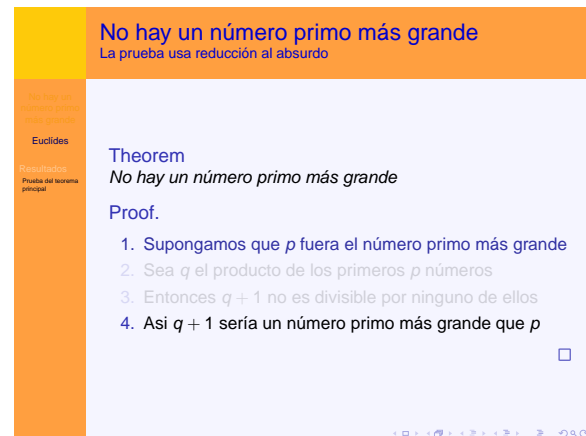
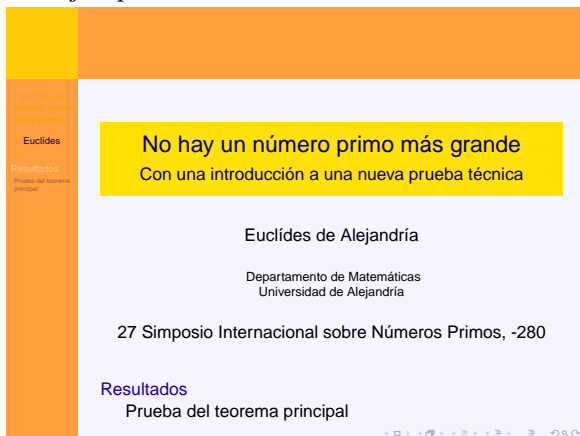
Ejemplo:



Como el tema de color `dove`, este tema es útil para imprimir las diapositivas con una copiadora de blanco y negro. Sin embargo, usa intensamente diferentes sombras de gris que pueden o no, tener una buena vista en una transparencia.

`\usecolortheme{wolverine}`

Ejemplo:

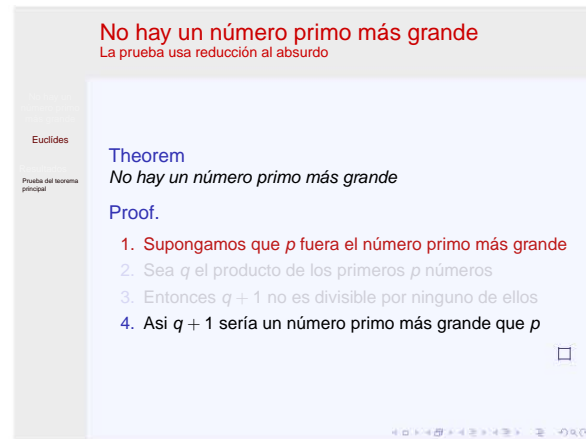
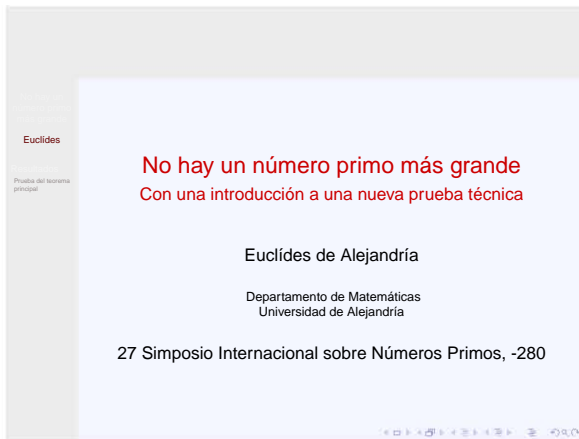


El tema se basa en los colores de la mascota (un lobo) de la Universidad de Michigan. *Autor del tema:*

Madhusudan Singh.

```
\usecolortheme{beaver}
```

Ejemplo:



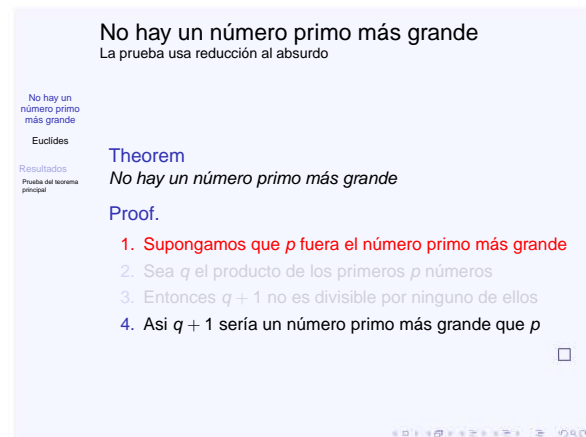
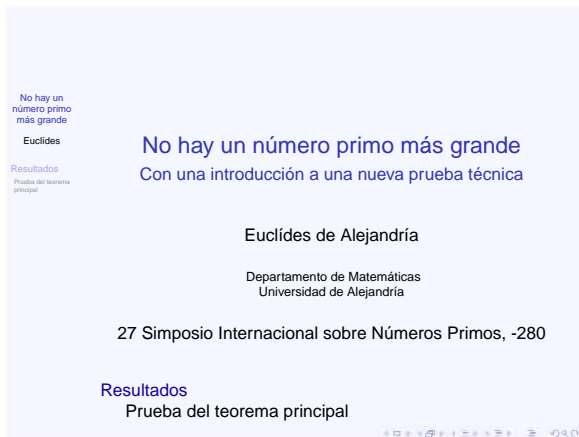
El tema se basa en los colores de la mascota (un castor) del MIT. *Autor del tema* Madhusudan Singh.

Temas de color internos

Los temas de color internos sólo especifican los colores de elementos usados en temas internos. Más evidentemente, especifican los colores usados para los bloques. Se pueden usar junto con otros temas de color. Si se usan para cambiar los colores internos instalados por un tema de presentación u otro tema de color, obviamente se deben especificar *después* de que se cargó el otro tema. Ocurre que los temas de color internos tienen nombres de flores.

```
\usecolortheme{lily}
```

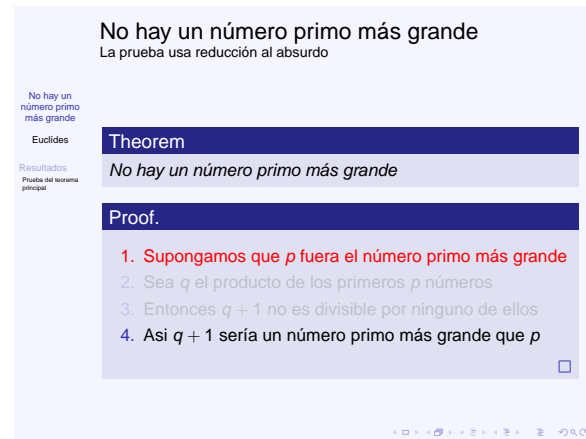
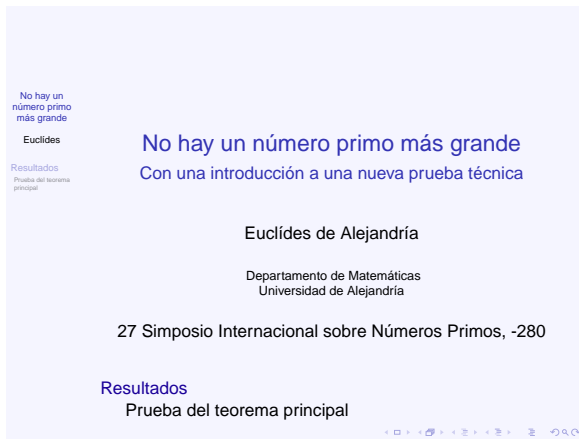
Ejemplo:



Este tema se usa principalmente para *desinstalar* cualquier color de bloque instalado por otro tema, restaurando los colores usados en el tema `default`. En particular, al usar este tema se quitan todos los colores de fondo para los bloques.

```
\usecolortheme{orchid}
```

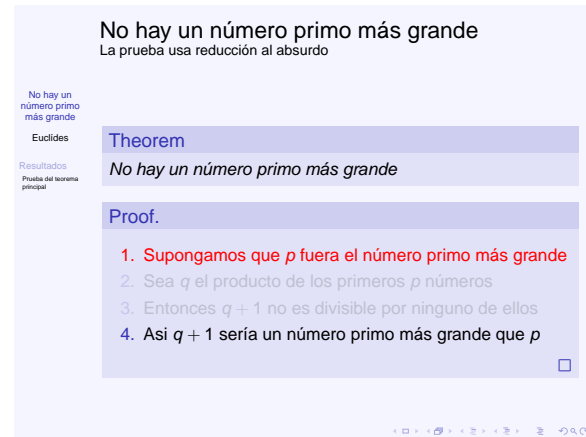
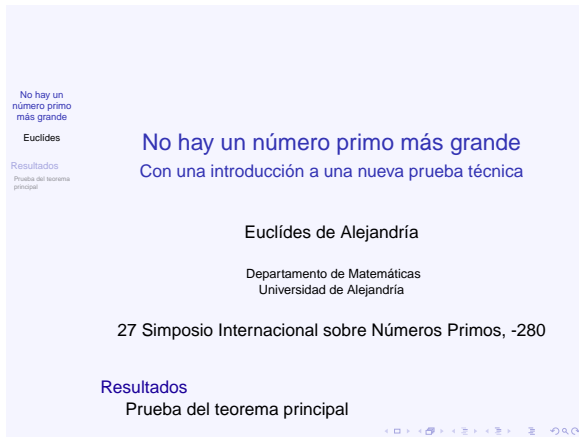
Ejemplo:



Este tema instala títulos de bloque blanco-sobre-oscuro. El fondo del título de un bloque normal se establece como el primer plano del color-BEAMER `structure`, el primer plano se establece como blanco. El fondo de bloques *alerta* se establece como rojo y el de bloques *ejemplo* como verde. El cuerpo de los bloques tiene un fondo casi transparente.

`\usecolortheme{rose}`

Ejemplo:



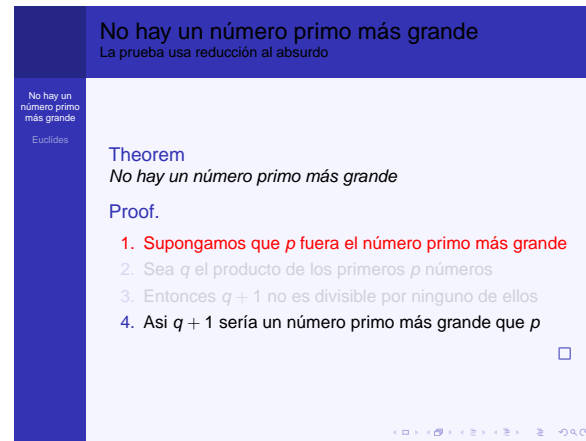
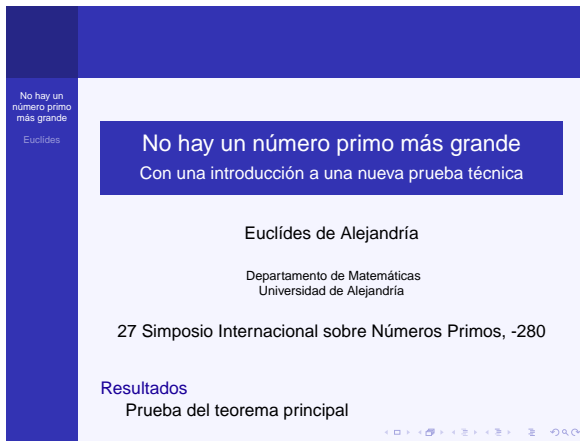
Este tema instala fondos casi transparentes tanto para el título de bloques como para el cuerpo de bloques. Este tema es mucho menos “agresivo” que el tema `orchid`. Los colores de fondo se derivan del primer plano del color-BEAMER `structure`.

Temas de color externos

Un tema de color externo cambia la paleta de colores sobre la que se basan por defecto los colores usados en el encabezado, pie de página y la barra lateral. Los temas de color externos normalmente no cambian el color de elementos internos, excepto posiblemente para `titlelike`. Ocurre que los temas de color externos tienen nombres de animales marinos.

`\usecolortheme{whale}`

Ejemplo:



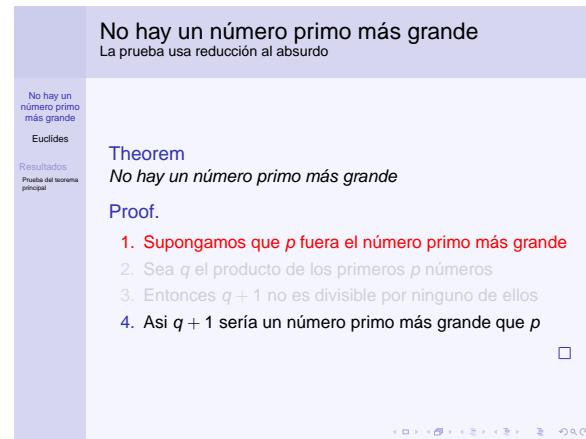
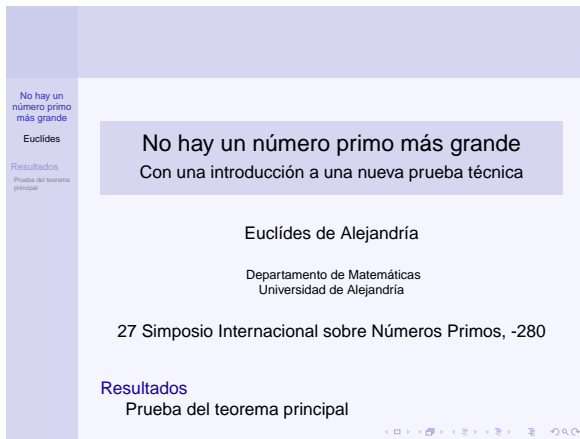
Instala una paleta blanco-sobre-oscuro para el encabezado, pie de página y barra lateral. En los fondos usados en esas partes, hay que poner sombras entre el color-BEAMER `structure` y negro. El primer plano se establece como blanco.

Mientras que este tema de color puede parecer agresivo, se debe notar que una barra oscura en el borde del marco tiene una apariencia algo diferente durante una presentación que la que tiene en el papel: durante una presentación la proyección en la pared usualmente está rodeada de oscuridad. Así, una barra oscura no crea un contraste tan opuesto a como lo hace en el papel. De hecho, al usar este tema se acentúa la atención hacia la parte principal del marco.

La contraparte de este tema en lo que respecta a bloques es el tema `orchid`. Sin embargo, si se usa junto con el tema de color `rose` se obtiene algo interesante.

```
\usecolortheme{seahorse}
```

Ejemplo:

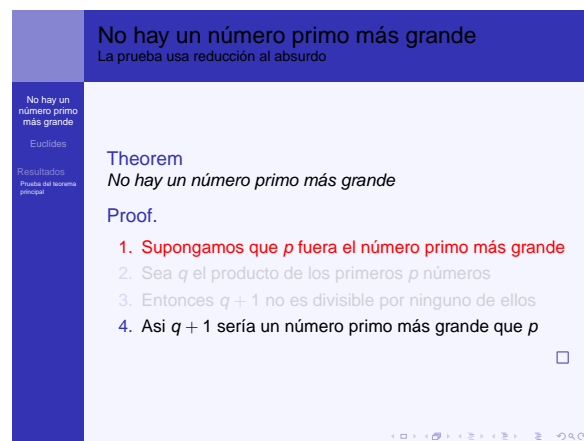
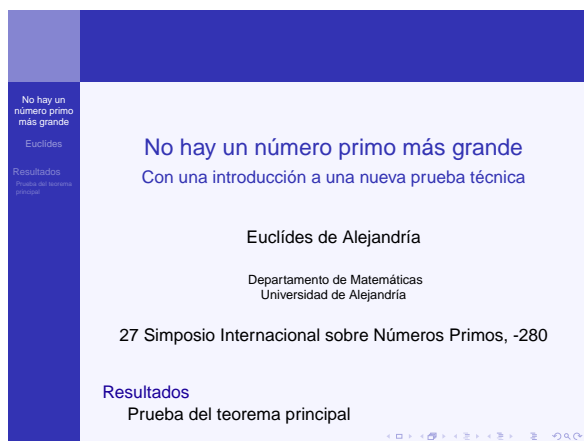


Instala un fondo casi transparente para el encabezado, pie de página y barra lateral. Al usar este tema se causa que los elementos de navegación sean mucho menos “dominantes” que cuando se usa el tema `whale`.

Va bien con el tema de color `rose` o `lily`. Si se usa junto con `orchid` se sobre enfatizan los bloques.

```
\usecolortheme{dolphin}
```

Ejemplo:



Un tema de color que está entre whale y seahorse. Fue donado cortésmente por Manuel Carro. Como seahorse, va bien con los temas de color rose y lily.

16.2. Cambiando los colores de elementos de una presentación

Esta sección explica cómo funciona el administrador de color de BEAMER.

Perspectiva general del administrador de color de Beamer

En la filosofía de BEAMER, cada elemento de una presentación puede tener un color diferente. Desafortunadamente resulta ser que la simple asignación de un solo color para cada elemento de una presentación no es tan buena idea. Antes que nada, a veces queremos que los colores de los elementos cambien durante una presentación, como el color de los indicadores de objeto cuando se les resalta o cuando están dentro de un bloque ejemplo. Segundo, de manera natural, algunos elementos tienen dos colores, a saber un primer plano y un fondo pero no siempre es así. Tercero, a veces el elemento (de algún modo) no debe tener un color especial sino que simplemente debe “marchar” con el color de su entorno. Finalmente, el dar un color especial a cada elemento dificulta el cambio de colores global (como cambiar todas las distintas cosas de tipo azul en tipo rojo) e incluso hace aun más difíciles las extensiones.

Por todas estas razones, el color de un elemento en BEAMER es un objeto estructurado, al que se le llama *color-BEAMER*. Cada *color-BEAMER* tiene dos partes: un primer plano y un fondo. Cualquiera de éstos puede estar “vacío”, lo que significa que cualquier primer plano o fondo que estuvo activo antes, debe permanecer activo cuando se usa el color.

Los colores-BEAMER se pueden *heredar* de otros colores-BEAMER y los temas predefinidos hacen un extensivo uso de esta característica. Por ejemplo, hay un *color-BEAMER* que se llama **structure** y toda una clase de elementos heredada de este color. Así, si alguien cambia **structure**, el color de todos estos elementos automáticamente cambia como corresponde. No obstante, cuando un color se hereda de otro todavía puede anular o el primer plano o el fondo.

También es posible “heredar” de otro *color-BEAMER* de una manera más sofisticada, que es más como *usar* el otro *color-BEAMER* de manera indirecta. Se puede especificar que, digamos, el fondo del título debe ser un 90 % del fondo de texto normal y 10 % del primer plano de **structure**. Heredar y usar otros colores-BEAMER se hace dinámicamente. Esto significa que si uno de los colores-BEAMER padre cambia durante la presentación, automáticamente los colores derivados también cambian.

El tema de color predefinido, que siempre se carga, instala numerosos colores-BEAMER y relaciones de herencia entre ellos. Estos colores se explican a lo largo de este trabajo. El color usado para, digamos, títulos de marco se discute en la sección sobre títulos de marco, etc.

Usando colores de Beamer

Un color-BEAMER no es como un color normal definido por los paquetes `color` y `xcolor` y, por consiguiente, no se puede usar directamente como un comando tipo `\color` o `\colorlet`. En cambio, para usar un color-BEAMER, se debe llamar primero al comando `\usebeamercolor` que se explica más adelante. Este comando establece dos colores (normales) llamados `fg` (para el primer plano) y `bg` (para el fondo). Se puede indicar `\color{fg}` para establecer el color del primer plano y `\color{bg}` para establecer el color de fondo. También se pueden usar los colores `fg` y `bg` en cualquier contexto en el que normalmente se usa un color como, digamos, `red`. Si un color-BEAMER no tiene primer plano o fondo, los colores `fg` o `bg` (o ambos) permanecen inalterados.

Dentro de las plantillas, tal vez ya se le llamó a este comando con la opción `[fg]`.

```
\usebeamercolor*[\fg o bg]{\nombre color-beamer}
```

Este comando (posiblemente) cambia los dos colores `fg` y `bg` a los colores del primer plano y fondo del `\nombre color-beamer`. Si el color-BEAMER no especifica un primer plano, `fg` queda inalterado; si no especifica un fondo, `bg` queda inalterado.

A menudo se desea usar el color `fg` o `bg` después de usar este comando. Para esta situación común es útil el argumento opcional `\fg o bg`, que puede ser `fg` o `bg`. Dar esta opción causa que se instale inmediatamente el primer plano `fg` o el fondo `bg` después de que han sido establecidos. Así, el siguiente comando

```
\usebeamercolor[fg]{normal text}
es un atajo para
\usebeamercolor{normal text}
\color{fg}
```

Si se usa la versión con asterisco de este comando, se usa el color-BEAMER `normal text` antes de que se invoque el comando. Esto asegura que, excepto por malvadas artimañas, se establezcan los colores `fg` y `bg` independientemente de los colores que estaban en uso cuando se invocó el comando.

Éste comando tiene efectos laterales especiales. Primero, el color (normal) `parent.bg` se establece como el valor de `bg` antes de esta llamada. Así que se puede acceder al color que estaba en uso antes de llamar a este comando por medio del color `parent.bg`.

Segundo, el color especial `\nombre color-beamer.fg` se establece *globalmente* como el mismo valor de `fg` y `\nombre color-beamer.bg` se establece globalmente como el valor de `bg`. Esto permite acceder al primer plano o fondo de un cierto `\nombre color-beamer` después que se ha usado otro color-BEAMER. Sin embargo, en relación a estos colores globales especiales uno se debe ceñir al mínimo inevitable y se debe hacer tan localmente como sea posible, ya que un cambio del color-BEAMER no reflejará un cambio de los colores `\nombre color-beamer.fg` y `\nombre color-beamer.bg` hasta la próxima invocación de `\usebeamercolor`. También, si `\nombre color-beamer` no especifica un color para el primer plano o fondo, entonces los valores de los colores especiales pasan a ser el primer plano o fondo al momento de la última invocación de `\usebeamercolor`.

Así, intente no entrar en el hábito de escribir `\color{structure.fg}` todo el tiempo, por lo menos no sin estar cerca de `\usebeamercolor{structure}`.

Ejemplo:

Este texto está `{\usebeamercolor[fg]{alerted text} resaltado}`. La siguiente caja usa el primer plano y fondo de `frametitles`:

```
{
  \usebeamercolor[fg]{frametitle}
  \colorbox{bg}{T'i}tulo de marco}
```



```
}
```

ARTÍCULO Este comando no tiene efecto en modo `article`.

```
\ifbeamercolorempy[⟨fg o bg⟩]{⟨nombre color-beamer⟩}{⟨si indefinido⟩}{⟨si definido⟩}
```

Este comando se puede usar para verificar si está vacío el primer plano o fondo de algún `⟨nombre color-beamer⟩`. Si el primer plano o fondo de `⟨nombre color-beamer⟩` está definido, se ejecutará `⟨si definido⟩`, de otro modo se ejecutará `⟨si indefinido⟩`.

Ejemplo:

```
\ifbeamercolorempy[bg]{frametitle}
{% ‘Fondo transparente’
  \usebeamercolor[fg]{frametitle}
  \insertframetitle
}
{% Fondo opaco
  \usebeamercolor[fg]{frametitle}
  \colorbox{bg}{\insertframetitle}
}
```

Estableciendo colores de Beamer

Para establecer o cambiar un color-BEAMER se puede usar el comando `\setbeamercolor`.

```
\setbeamercolor*{⟨nombre color-beamer⟩}{⟨opciones⟩}
```

Establece o cambia un color-BEAMER. El `⟨nombre color-beamer⟩` debe ser un texto razonablemente simple (no debe ser truculento y evitando símbolos de puntuación), pero puede contener espacios. Así, `normal text` es un `⟨nombre color-beamer⟩` válido y también lo es `Mi Color Número 2`.

En el caso más simple, se especifica un primer plano sólo dando la opción `fg=⟨opciones⟩` y, posiblemente, también un fondo usando la opción `bg=⟨opciones⟩`.

Ejemplo:

```
\setbeamercolor{normal text}{fg=black,bg=migrisluminoso}
```

Ejemplo:

```
\setbeamercolor{alerted text}{fg=red!80!black}
```

El efecto de este comando es acumulativo, así que los dos siguientes comandos

```
\setbeamercolor{section in toc}{fg=blue}
```

```
\setbeamercolor{section in toc}{bg=white}
```

tienen el mismo efecto que

```
\setbeamercolor{section in toc}{fg=blue,bg=white}
```

Naturalmente, una segunda llamada con la misma clase de `⟨opciones⟩` inicia un valor diferente que anula una llamada anterior.

La versión con asterisco primero restablece todo, por eso se desactiva el efecto acumulativo. Se usa esta versión con asterisco para restablecer completamente la definición de algún color-BEAMER.

Se pueden dar las siguientes opciones:

- `fg=⟨color⟩` establece el color del primer plano de `⟨nombre color-beamer⟩` como el `⟨color⟩` (normal) dado. El `⟨color⟩` también puede ser una expresión de color como `red!50!black`.

Si `⟨color⟩` está vacío, el `⟨nombre color-beamer⟩` “no tiene un primer plano especial” y cuando se usa el color, el primer plano actualmente vigente no debe cambiar.

Al especificar un primer plano de esta manera se anula cualquier color de primer plano heredado.

- **bg**=*<color>* hace lo mismo que la opción **fg** pero para el fondo.
- **parent**=*<color(es)-beamer padre>* especifica lo que *<nombre color-beamer>* debe heredar de los *<color(es)-beamer padre>* especificados. Cualquier color de primer plano y/o fondo establecido por los padres será usado también cuando se haga uso de *<nombre color-beamer>*. Si los múltiples padres especifican un primer plano, el último “gana”; lo mismo aplica para fondos.

Ejemplo:

```
\setbeamercolor{padre}{fg=red}
\setbeamercolor{madre}{bg=green}
\setbeamercolor{hijo}{parent={padre,madre}}
\begin{beamercolorbox}{hijo}
  Terrible texto rojo sobre verde.
\end{beamercolorbox}
\setbeamercolor{padre}{fg=blue}
\begin{beamercolorbox}{hijo}
  Ahora terrible texto azul sobre verde, ya que se cambió el padre.
\end{beamercolorbox}
```

Notamos que un cambio del primer plano o fondo de un padre cambia el correspondiente primer plano o fondo del hijo (a menos que se anule).

Un color-BEAMER puede no sólo tener padres, también abuelos, etcétera.

- **use**=*<otro color-beamer>* se usa para asegurar que se estableció correctamente otro color-BEAMER antes de que se evalúen las especificaciones de color del primer plano o fondo.

Supongamos que se quiere que el primer plano de objetos sea una mezcla de 50% del primer plano de elementos estructurales y 50% del color del primer plano normal. Se puede intentar

```
\setbeamercolor{item}{fg=structure.fg!50!normal text.fg}
```

Sin embargo, esto no necesariamente da el resultado deseado: si el color-BEAMER `structure` cambia, el color (normal) `structure.fg` no se actualiza inmediatamente. Para garantizar que el color normal `structure.fg` es correcto, se debe usar lo siguiente:

```
\setbeamercolor{item}{use={structure,normal text},fg=structure.fg!50!normal text.fg}
```

Esto garantiza que los colores `structure.fg` y `text.fg` estén correctamente establecidos cuando se calcula el primer plano de *item*. Para ver la diferencia, consideremos el siguiente ejemplo:

```
\setbeamercolor{abuelo}{fg=red}
\setbeamercolor{abuela}{bg=white}
\setbeamercolor{padre}{parent={abuelo,abuela}}
\setbeamercolor{madre}{fg=black}
{
  \usebeamercolor{padre}\usebeamercolor{madre}
  %% Define padre.fg y madre.fg globalmente
}
\setbeamercolor{mi color A}{fg=padre.fg!50!madre.fg}
```

```

\setbeamercolor{mi color B}{use={padre,madre},fg=padre.fg!50!madre.fg}

{\usebeamercolor[fg]{mi color A} texto rojo oscuro}
{\usebeamercolor[fg]{mi color B} texto rojo tambi\'}{e}n oscuro}

\setbeamercolor{abuelo}{fg=green}
{\usebeamercolor[fg]{mi color A} todav\'}{i}a texto rojo oscuro}
{\usebeamercolor[fg]{mi color B} ahora texto verde oscuro}

```

16.3. El color del texto matemático

Por defecto, el texto matemático no tiene ningún color especial- sólo hereda el color “de alrededor”. Algunos prefieren que el texto matemático tenga un color especial. Aunque *no* se recomienda esto, BEAMER facilita (razonablemente) el cambio de color del texto matemático. Simplemente hay que cambiar los siguientes colores:

Color-Beamer `math text`

Este color es el padre de `math text inlined` y de `math text displayed`. Por defecto está vacío.

Color-Beamer `math text inlined`

Color Padre: `math text`

Si se establece el primer plano de este color, el texto matemático en línea se edita usando este color. Esto se hace vía `\everymath` y puede no funcionar en todos los casos, así que se tiene que intentar encontrar una manera de evitar el problema. Actualmente se ignora el fondo.

Color-Beamer `math text displayed`

Color Padre: `math text`

Como `math text inlined`, sólo que para el llamado texto matemático “desplegado”. Este es texto matemático entre `\[` y `\]` o entre `$$` y `$$` o dentro de los ambientes como `equation` o `align`. La organización de este color es algo frágil, se debe usar bajo nuestro propio riesgo. Actualmente se ignora el fondo.

Color-Beamer `normal text in math text`

Si se establece el primer plano de este color, el texto normal dentro del texto matemático (qué se introduce usando los comandos `\text`) se edita usando este color. El fondo se ignora actualmente.

16.4. Las paletas de color

Cuando se diseña un tema de color uno se enfrenta al siguiente problema: supongamos que en el encabezado se quieren colores que cambien gradualmente de negro a, digamos, azul. Cualquier cosa que esté en la parte superior del encabezado debe ser negra, lo que viene justo debajo debe ser azul oscuro y en el fondo del encabezado deben ser sólo azules.

Desafortunadamente, temas externos distintos ponen cosas distintas en la parte superior. Un tema puede poner el autor en la parte superior, otro tema puede poner el título del documento allí mismo. Esto imposibilita la asignación directa de los tres colores “negro”, “azul oscuro” y “azul” para los diferentes elementos que están típicamente representados en el encabezado. No importa cómo los asignemos, las cosas se verán mal para ciertos temas externos.

Para sortear este problema, BEAMER usa una capa de *colores de paleta*. Los temas de color típicamente sólo cambian estos colores de la paleta. Por ejemplo, un tema de color podría hacer azul al `color-BEAMER palette primary`, hacer azul oscuro a `palette secondary` y hacer negro a `palette tertiary`. Los temas externos ahora pueden establecer cosas de tal manera que lo que muestran en la parte superior del encabezado herede de `palette tertiary`, lo que viene justo debajo herede de `palette secondary` y lo

que está al fondo herede de `palette tertiary`. De esta manera, los temas de color pueden cambiar la manera, incluso complicada, de cómo se ven los temas externos y pueden hacerlo de forma consistente.

Notamos que todavía se puede cambiar el color de cada elemento simplemente anulando el color (o colores) de los elementos en el encabezado. En cierto modo, los colores de la paleta son sólo una “sugerencia” de cómo las cosas deben ser coloreadas por un tema exterior.

En detalle, los temas externos usan los colores de paleta siguientes:

Color-Beamer `palette primary`

Los temas externos basan el color de los elementos de navegación y, posiblemente de otros elementos, sobre los colores de las cuatro paletas. La paleta “primaria” se debe usar para los elementos de navegación más importantes, que normalmente son los que cambian con mayor frecuencia y que requieren mayor atención del público. La “secundaria” y “terciaria” son menos importantes, la “cuaternaria” es la menos importante.

Por defecto, los colores de la paleta no tienen fondo y el primer plano varía de `structure.fg` a negro. Para la barra lateral hay un conjunto extra de colores de paleta, ver `palette sidebar primary`.

Color-Beamer `palette secondary`

Ver `palette primary`.

Color-Beamer `palette tertiary`

Ver `palette primary`.

Color-Beamer `palette quaternary`

Ver `palette primary`.

Color-Beamer `palette sidebar primary`

Similar a `palette primary`, sólo que los temas externos basan los colores de elementos en la barra lateral sobre los colores de las cuatro paletas de la barra lateral.

Color-Beamer `palette sidebar secondary`

Ver `palette sidebar primary`.

Color-Beamer `palette sidebar tertiary`

Ver `palette sidebar primary`.

Color-Beamer `palette sidebar quaternary`

Ver `palette sidebar primary`.

16.5. Miscelánea de colores

En esta sección se listan algunos colores “básicos” que no “pertenecen” a ningún comando especial.

Color-/Fuente-Beamer `normal text`

Este color se usa para texto normal. Al principio del documento el color del primer plano es instalado como `\normalcolor`. El fondo de este color es usado por el lienzo de fondo predefinido para el fondo de la presentación. El fondo también es el valor predefinido del color normal `bg`.

Dado que este color es la “raíz” de los otros colores-BEAMER, se deben instalar tanto un primer plano como un fondo. En particular, para conseguir un fondo de lienzo transparente, se deja vacío el fondo del color-BEAMER `background canvas`, no el fondo de este color.

Actualmente no se usa la fuente-BEAMER. En particular, redefinir esta fuente no tendrá efecto alguno. Es probable que esto cambie en el futuro.

Color-/Fuente-Beamer `example text`

Este color/fuente se usa cuando se edita el texto dentro de un bloque `example`.

Color-/Fuente-Beamer `titlelike`

Este color/fuente es una forma más especializada del color/fuente `structure`. Es la base para todos los elementos que son “como títulos”. Esto incluye el título y subtítulo del marco así como el título y subtítulo del documento.

Color-Beamer `separation line`

El primer plano de este color se usa para separar líneas. Si el primer plano está vacío, no se dibuja ninguna línea de separación.

Color-Beamer `upper separation line head`

Color Padre: `separation line`

Caso especial para la línea de separación más alta en un encabezado.

Color-Beamer `middle separation line head`

Color Padre: `separation line`

Caso especial para la línea de separación media en un encabezado.

Color-Beamer `lower separation line head`

Color Padre: `separation line`

Caso especial para la línea de separación más baja en un encabezado.

Color-Beamer `upper separation line foot`

Color Padre: `separation line`

Caso especial para la línea de separación más alta en un pie de página.

Color-Beamer `middle separation line foot`

Color Padre: `separation line`

Caso especial para la línea de separación media en un pie de página.

Color-Beamer `lower separation line foot`

Color Padre: `separation line`

Caso especial para la línea de separación más baja en un pie de página.

16.6. Efectos de transparencia

Por defecto, los objetos *cubiertos* no se muestran durante una presentación. Si se escribe `\uncover<2>{Texto.}`, el texto no se muestra en ninguna diapositiva excepto en la segunda. En las otras diapositivas, el texto no se imprime usando el color de fondo sino que no se muestra en absoluto. Este efecto es muy útil si el fondo no tiene un color uniforme.

Sin embargo, a veces se quiere que los objetos cubiertos no estén completamente cubiertos. Más bien, nos gustaría que se mostraran de una manera muy tenue o sombreada. Esto le permite al público tener una sensación de lo que esta por venir, sin distraerse. También se puede querer texto que este cubierto “una vez más” pero que todavía sea visible hasta cierto punto.

Idealmente, existiría una opción para hacer “transparente” el texto cubierto. Esto significa que cuando se muestra el texto cubierto, éste se debería mezclar con el fondo detrás de él. Desafortunadamente `pgf` todavía no soporta la transparencia real. En cambio, la transparencia se crea mezclando el color del objeto

que se quiere mostrar con el color de fondo actual (el color `bg`, que ojala se haya establecido de tal manera que es el color promedio del fondo en el que se debe poner el objeto). Para instalar este efecto se puede usar:

```
\setbeamercovered{transparent}
```

Este comando permite especificar de una manera bastante general cómo se debe representar un objeto cubierto. Se pueden especificar diferentes maneras de representar el objeto, dependiendo de cuánto tiempo tomará antes de que se muestre este objeto o por cuánto tiempo ha estado cubierto una vez más. El efecto de transparencia se aplica automáticamente a todo los colores, *excepto* a los colores en imágenes.

```
\setbeamercovered{<opciones>}
```

Este comando ofrece varias opciones diferentes, siendo la más importante `transparent`. Todas las opciones internamente se trazan como las dos opciones `still covered` y `again covered`.

Se pueden dar las siguientes *<opciones>*:

- `invisible` es el valor predefinido y causa que el texto cubierto “desaparezca completamente”.
- `transparent=<opacidad>` causa que el texto cubierto se edite de una manera “transparente”. Por defecto esto significa que 85 % del color del fondo se mezcla con todos los colores o que la *<opacidad>* del texto es de 15 %. Se puede especificar un *<porcentaje>* diferente, donde 0 significa “totalmente transparente” y 100 significa “totalmente opaco”.

Desgraciadamente este valor depende de cada proyector “específico”. Lo que se ve bien en la pantalla no necesariamente se ve bien durante una presentación.

- `dynamic` Hace que todo el texto cubierto sea bastante transparente pero de una manera dinámica.
- `highly dynamic` Tiene el mismo efecto que `dynamic` pero el efecto es más fuerte.
- `still covered=<lista todavía no>` especifica cómo representar artículos cubiertos que todavía no han sido descubiertos. La *<lista todavía no>* debe ser una lista de comandos `\opaqueness`.

Ejemplo:

```
\setbeamercovered{%
still covered={\opaqueness<1>{15}\opaqueness<2>{10} \opaqueness<3>{5}\opaqueness<4->
{2}},
again covered={\opaqueness<1->{15}}}
```

- `again covered=<lista una vez más>` especifica cómo representar objetos cubiertos que han sido una vez más cubiertos, es decir, que se habían mostrado antes pero que están ahora cubiertos una vez más.

```
\opaqueness<especificación de cubierta>{<porcentaje de opacidad>}
```

La *<especificación de cubierta>* especifica en que diapositivas el texto cubierto debe tener *<porcentaje de opacidad>*. A diferencia de otras especificaciones de cubierta, esta *<especificación de cubierta>* es una especificación de cubierta “relativa”. Por ejemplo, la especificación “3” aquí significa “cosas que se descubrirán tres diapositivas más adelante”, respectivamente “cosas que han sido cubiertas una vez más por tres diapositivas”. Más precisamente, si se descubre un objeto por más de una diapositiva y luego se cubre una vez más, sólo el “primer momento de descubrimiento” se usa para el cálculo de cuánto tiempo ha estado cubierto el objeto una vez más.

Una opacidad de 100 es totalmente opaco y 0 es totalmente transparente. Actualmente, dado que todavía no se implementa la transparencia real, este comando causa que todos los colores tengan una mezcla de *<porcentaje de opacidad>* del `bg` actual. En un futuro próximo este comando podría producir transparencia real.

La extensión alterna pgf usada dentro de un área opaca es $\langle \text{porcentaje de opacidad} \rangle \text{opaque}$. En caso de llamadas anidadas, sólo se usa la especificación de opacidad más profunda.

Ejemplo:

```
\setbeamercovered{still covered={\opaqueness<1->{15}},  
again covered={\opaqueness<1->{15}}}  
\pgfdeclareimage{libro}{libro}  
\pgfdeclareimage{libro.!15opaque}{nombredearchivoalibrocasitransparente}
```

Hace que todo lo que se descubre dentro de dos diapositivas sólo sea 15 por ciento opaco.

17. Fuentes

La primera subsección introduce los temas de fuente predefinidos que vienen con BEAMER y que facilitan el cambio de las fuentes usadas en una presentación. La siguiente subsección describe otros comandos especiales para cambiar algunos atributos básicos de las fuentes usadas en una presentación. La última subsección explica cómo conseguir un control mucho más fino sobre las fuentes de cada elemento individual de una presentación.

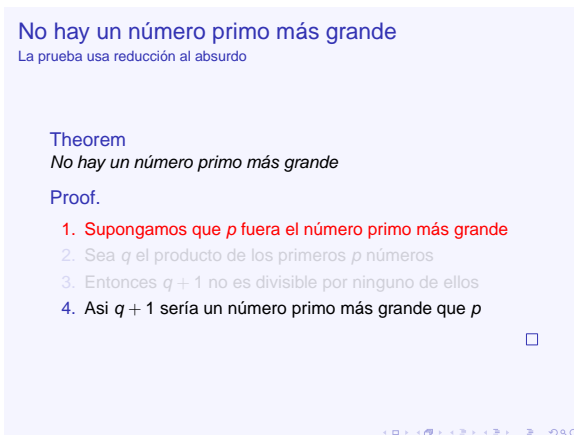
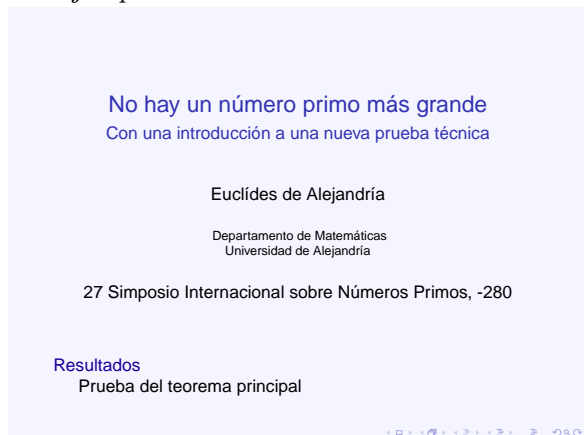
17.1. Temas de Fuente

BEAMER viene con un conjunto de temas de fuente. Cuando se usan tales temas, ciertas fuentes se pueden cambiar como se describe mas adelante. Se pueden usar conjuntamente varios temas de fuente. Por razones históricas no se pueden cambiar, usando temas de fuente, todos los aspectos de las fuentes utilizadas -en algunos casos se necesitan comandos y opciones especiales las cuales se describen en la próxima subsección.

Los siguientes temas de fuente sólo cambian ciertos atributos de la fuente y no eligen familias de fuente especiales (aunque eso también podría ser posible y en el futuro se podrían agregar temas que simplemente hagan eso). Actualmente, para cambiar la familia de fuente, se necesitan cargar paquetes especiales como se explica en la próxima subsección.

```
\usefonttheme{default}
```

Ejemplo:



El tema de fuente predefinido instala una fuente sans serif para todo el texto de la presentación. El tema predefinido instala diferentes tamaños de fuente para cosas como títulos o encabezados y pies de página, pero no usa negritas o cursivas para “resaltar”. Para cambiar parte o todo el texto a una fuente serif se debe usar el tema `serif`.

Nota: El comando `\mathrm` siempre produce texto serif vertical (no inclinado) y el comando `\mathsf` siempre produce texto sans-serif vertical. El comando `\mathbf` produce texto vertical en negritas sans-serif o serif (dependiendo de si se usa `mathsans` o `mathserif`).

Dependiendo de si se usa `mathsans` o `mathserif`, para producir un texto vertical sans serif o serif, se puede usar el comando `\operatorname` del paquete `amsmath`. Al usar directamente este comando en lugar de `\mathrm` o `\mathsf`, se ajusta automáticamente el texto matemático vertical si se cambia de sans-serif a serif o al revés.

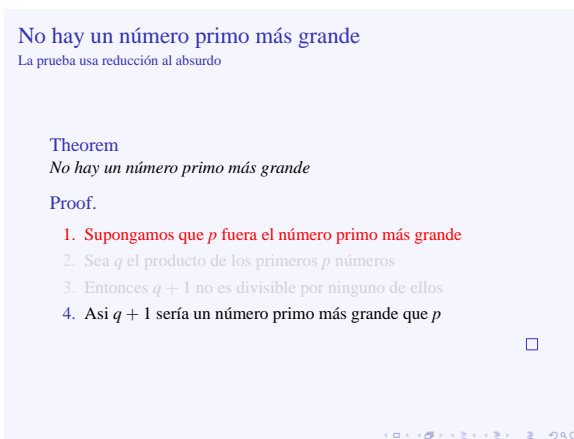
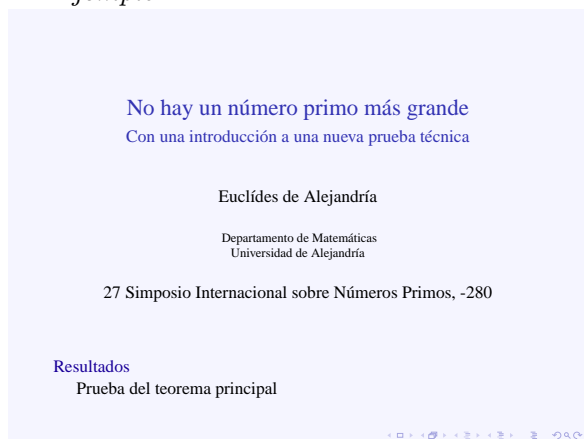
```
\usefonttheme{professionalfonts}
```

Este tema de fuente realmente no cambia ninguna fuente. Más bien *suprime* ciertos reemplazos internos realizados por BEAMER. Si usamos “fuentes profesionales” (fuentes que uno compra y que vienen con un conjunto completo de cada símbolo en todos los modos), no queremos que BEAMER se meta con

ellas. BEAMER normalmente reemplaza ciertos glyphs de carácter en texto matemático por versiones mas apropiadas. Por ejemplo, BEAMER normalmente reemplaza glyphs de tal manera que para las variables en texto matemático se usan caracteres en cursivas de la fuente principal. Si el paquete de fuente profesional ya cuida de esto, debe anularse la intromisión de BEAMER. Note que la substitución de BEAMER se anula automáticamente si se carga alguno de los siguientes paquetes: `mathtime`, `mathpmt`, `lucidabr`, `mtpro` y `hvmath`. Si nuestro paquete favorito de fuente profesional no está entre éstos, se debe usar la opción `professionalfont`.

```
\usefonttheme[⟨opciones⟩]{serif}
```

Ejemplo:



Este tema causa que todo el texto se edite usando la fuente predefinida serif (excepto si se especifican ciertas *⟨opciones⟩*).

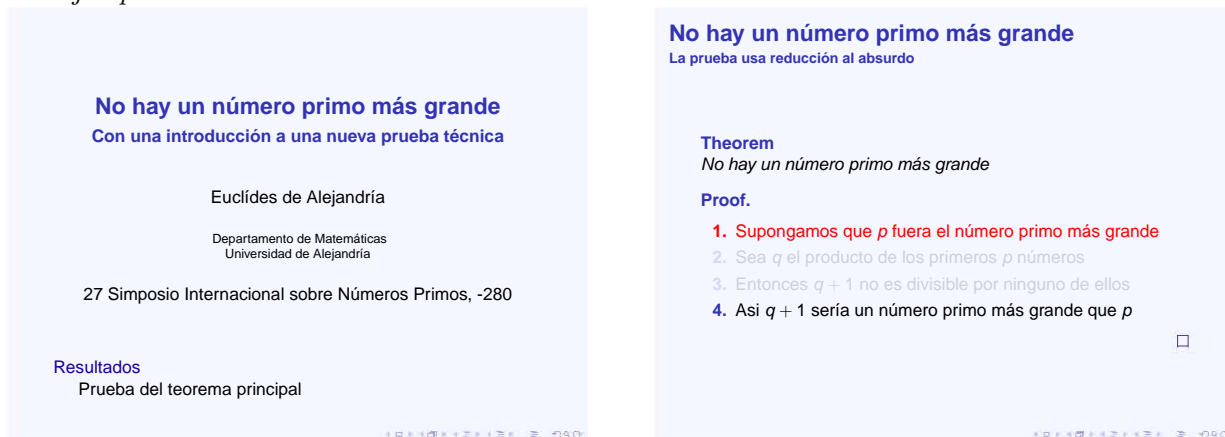
Se pueden dar las siguientes *⟨opciones⟩*

- `stillssansserifmath` causa que el texto matemático todavía se edite usando sans serif. Esta opción sólo tiene sentido si se usa también la opción `stillssansseriftext` ya que texto matemático sans serif dentro de texto serif se ve ridículo.
- `stillssansserifsmall` causa que el texto “pequeño” todavía se edite usando sans serif. Esto se refiere al texto en el encabezado, pie de página y barras laterales. Se aconseja usar esta opción ya que frecuentemente el texto pequeño en sans serif es más legible.
- `stillssansseriflarge` causa que el texto “grande” todavía se edite usando sans serif. Esto se refiere al título de la presentación o al título del marco. Los títulos sans serif con texto serif son una combinación popular en tipografía.
- `stillssansseriftext` causa que el texto normal (ninguno de los casos anteriores) todavía se edite usando sans serif. Si se usa esta opción, probablemente se deben usar también las dos primeras. Sin embargo, al no usar `stillssansseriflarge` se consigue un título serif (posiblemente en cursivas) sobre un texto sans serif. Esto puede ser un efecto visual interesante. Naturalmente “efecto tipográfico interesante” puede significar “efecto tipográfico terrible” si se eligen las fuentes o tamaños incorrectos. Se necesita algo de experiencia tipográfica para juzgar que es correcto.
- `onlymath` es una forma corta para seleccionar todas las opciones anteriores excepto la primera. Así, al usar esta opción se causa que sólo el texto matemático se edite usando una fuente serif. Recordar que, por defecto, las fórmulas matemáticas se editan usando letras sans serif. En la mayoría de los casos, esto es visualmente más agradable y una manera más legible de editar formulas matemáticas si el texto alrededor está editado con sans serif. Sin embargo, en los textos matemáticos la fuente que se usa para representar, digamos, una variable a veces es usada para diferenciar entre los diferentes significados de esta variable. En tal caso, es necesario editar el texto matemático usando letras serif.

Además, si se tiene mucho texto matemático, la audiencia puede “analizarlo” más rápidamente si se le edita en la forma en que generalmente se lee: en una fuente serif.

`\usefonttheme[⟨opciones⟩]{structurebold}`

Ejemplo:



Este tema de fuente causa que se editen títulos y texto en los encabezados, pies de páginas y barras laterales con una fuente en negritas.

Se pueden dar las siguientes opciones:

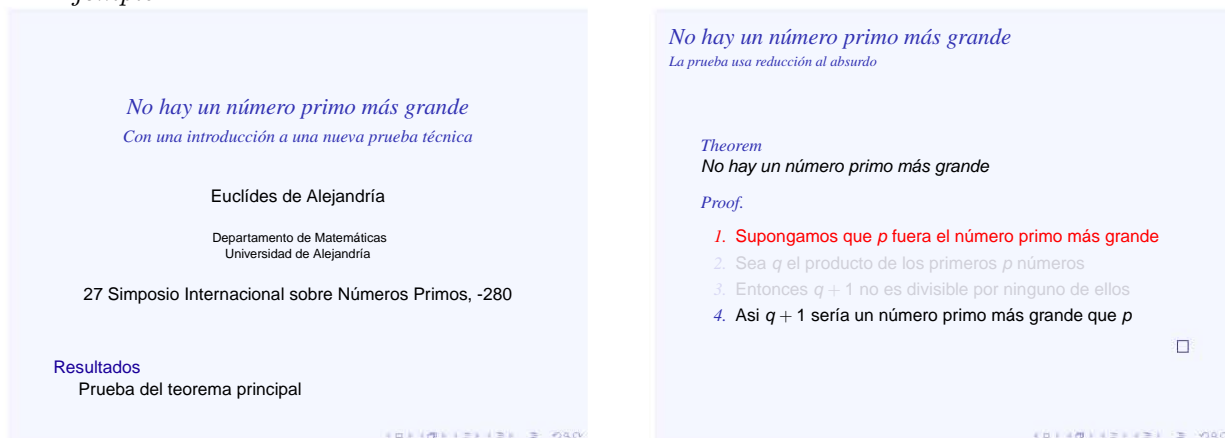
- **onlysmall** causa que sólo el texto “pequeño” se edite en negritas. Más precisamente, sólo se cambia el texto en el encabezado, pie de página y barra lateral para que se edite en negritas. No se afectan los títulos grandes.
- **onlylarge** causa que sólo el texto “grande” se edite en negritas. Estos son el título principal, los títulos de marcos y las entradas de sección en la tabla de contenidos.

Como se señaló en la Sección 5.6.1, se debe usar este tema (posiblemente con la opción **onlysmall**) si la fuente no se reduce apropiadamente o para texto brillante sobre oscuro.

Los temas normales no instalan este tema por defecto, mientras que los temas compatibles anteriores si lo hacen. Puesto que se puede recargar este tema una vez que ha sido cargado, no se puede usar este tema con temas compatibles anteriores para establecer también los títulos con una fuente en negritas.

`\usefonttheme[⟨opciones⟩]{structureitalicserif}`

Ejemplo:

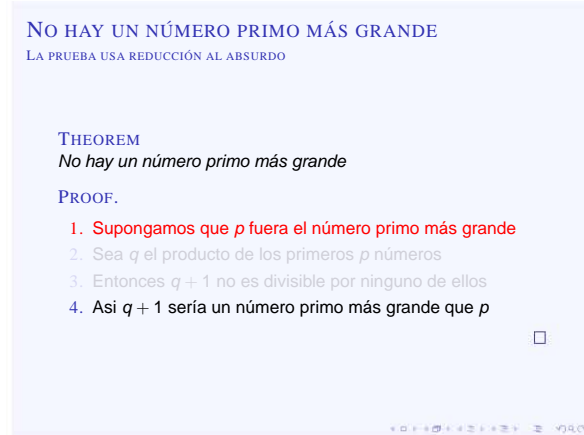
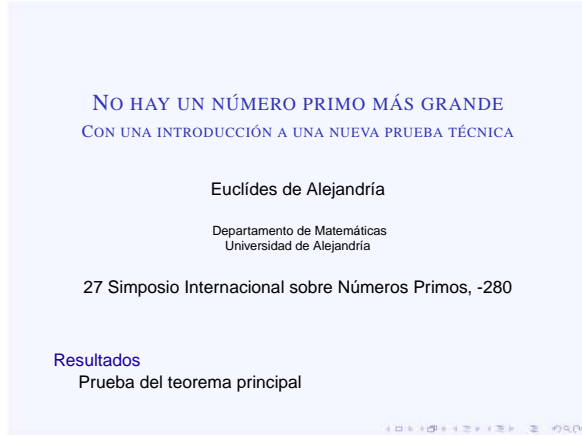


Este tema es parecido al tema de fuente **structurebold**, pero donde **structurebold** crea texto en negritas, este tema lo edita en cursivas y con la fuente estándar serif. Soporta las mismas *⟨opciones⟩* que

el tema `structurebold`. Ver la sección 5.6.3 para los pros y contras de usar cursivas.

```
\usefonttheme[opciones]{structuresmallcapsserif}
```

Ejemplo:



Otra vez, este tema hace exactamente lo mismo que el tema de fuente `structurebold`, sólo que esta vez el texto usa mayúsculas pequeñas y la fuente serif. Soporta las mismas opciones que el tema `structurebold`. Ver la sección 5.6.3 para los pros y contras de usar letras mayúsculas pequeñas.

17.2. Haciendo cambio de fuente sin usar temas de fuente.

Mientras que la mayoría de las elecciones de fuente se pueden hacer usando temas de fuente, por razones históricas algunos cambios sólo se pueden hacer usando opciones de clase o cargando paquetes especiales. Estas opciones se explican a continuación. Posiblemente estas opciones se reemplazarán en el futuro por un tema.

Eligiendo un tamaño de fuente para el texto normal.

Como se vio en la sección 5.6.1, no es una buena idea medir el tamaño predefinido de la fuente de una presentación en puntos. No obstante, como es usual, BEAMER simplemente establece el tamaño predefinido en 11 pt. Esto puede ser ridículamente pequeño, pero el tamaño real de cada marco es de sólo 128mm por 96 mm y la aplicación visor alarga la fuente. Al especificar un tamaño de fuente más pequeño que el predefinido (11pt) se puede poner más en cada diapositiva, al especificar un tamaño de fuente mas grande se puede meter menos.

Para especificar el tamaño de la fuente, se pueden usar las siguientes opciones de clase:

```
\documentclass[8pt]{beamer}
```

Este modo es demasiado pequeño. Requiere que se instale el paquete `extsize`.

```
\documentclass[9pt]{beamer}
```

También es demasiado pequeño. Requiere que se instale el paquete `extsize`.

```
\documentclass[10pt]{beamer}
```

Esta opción se usa si realmente se necesita meter mucho texto en cada marco. Funciona sin el paquete `extsize`.

```
\documentclass[smaller]{beamer}
```

Lo mismo que la opción 10pt.

```
\documentclass[11pt]{beamer}
```

Tamaño de la fuente por defecto. No se necesita especificar esta opción.

```
\documentclass[12pt]{beamer}
```

Hace que la fuente sea un poco más grande, lo que hace al texto más legible. El inconveniente es que cabe menos texto en cada marco.

```
\documentclass[bigger]{beamer}
```

Lo mismo que la opción 12pt.

```
\documentclass[14pt]{beamer}
```

Hace que todas las fuentes sean algo más grandes. Requiere que se instale el paquete `extsize`.

```
\documentclass[17pt]{beamer}
```

Es casi el tamaño predefinido de PowerPoint. Requiere que se instale el paquete `extsize`.

```
\documentclass[20pt]{beamer}
```

Realmente es muy grande. Requiere que se instale el paquete `extsize`.

Eligiendo una familia de fuente.

Por defecto, BEAMER usa las fuentes Computer Modern. Para cambiar esto, se puede usar alguno de los paquetes preparados del mecanismo de fuente de L^AT_EX. Por ejemplo, para cambiar a Times/Helvéctica se pone en el preámbulo

```
\usepackage{mathptmx}
\usepackage{helvet}
```

Notamos que si no se usa el tema de fuente `serif`, se selecciona Helvéctica (no Times) como la fuente de texto.

Puede haber otras fuentes disponibles en la instalación. Normalmente algunos de los siguientes paquetes deben estar disponibles: `avant`, `bookman`, `chancery`, `charter`, `euler`, `helvet`, `mathtime`, `mathptm`, `mathptmx`, `newcent`, `palatino`, `pifont`, `utopia`.

Eligiendo una codificación de fuente.

Una misma fuente puede venir con diferentes codificaciones, que son (burdamente hablando) las formas en que los caracteres de un texto son mapeados en glyphs (la forma real de un carácter particular en una fuente particular de un tamaño particular). En T_EX a menudo se usan dos codificaciones con caracteres Latinos: la codificación T1 y la codificación OT1 (la antigua codificación T1).

Conceptualmente, la codificación más reciente T1 es preferible sobre la vieja codificación OT1. Desafortunadamente, las fuentes EC, esto es las fuentes Computer Modern T1-codificadas, se distribuyen sobre instalaciones pequeñas (como las fuentes de Metafont) y sólo tienen traducciones de mapas de bits de cada glyph. Por esta razón, al usar las fuentes EC T1-codificadas sobre tales instalaciones pequeñas se producirán archivos PDF que se representan pobremente.

MiKTEX (para plataformas en Windows) y t_ETeX (para Unix/Linux) se pueden instalar con diferentes niveles de completez. En cuanto a las fuentes Computer Modern, se pueden instalar los siguientes paquetes: `cm-super fonts`, `lmodern` (Latín moderno) fonts y `lgc fonts`, el último contiene los alfabetos latino, griego y cirílico. En cuanto a las otras fuentes, `txfonts` y `pxfonts` son dos conjuntos extendidos de las fuentes Times y Palatino PostScript, ambos paquetes contienen conjuntos extendidos de glyphs matemáticos. La mayoría de las fuentes Poscript estándar también están disponibles en la codificación T1.

Entre los paquetes que hacen disponibles las fuentes Computer Modern en la codificación T1, se puede sugerir el paquete `lmodern`. Si se usa `lmodern`, están disponibles varias fuentes extras (como una

fuente matemática sans-serif en negritas) y símbolos extras. Para seleccionar la codificación T1, se usa `\usepackage[T1]{fontenc}`. De esta manera, si se tienen instaladas las fuentes Latin Modern, se puede escribir:

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

para conseguir hermosas fuentes y la colocación correcta de guiones. Hacemos notar sin embargo, que ciertas versiones anteriores del conjunto LM no incluyeron glyphs correctos para los ligados como “fi”, lo cual puede causar problemas. Cerciorarse de que los ligados se desplieguen correctamente y si no, actualizar.

17.3. Cambiando las fuentes de los elementos de una presentación.

Esta sección explica como funciona el administrador de fuentes de BEAMER.

Resumen del administrador de fuentes de Beamer

El mecanismo de fuente de BEAMER es algo parecido al mecanismo de color de BEAMER, pero no exactamente igual. Como para los colores, cada elemento BEAMER, como los títulos de marco, el título del documento, los pies de nota, etc. tiene una cierta fuente-BEAMER. Como para los colores, por un lado se puede especificar la fuente de cada elemento individualmente; por otro lado las fuentes también usan la herencia, por eso se facilita el cambio global de las fuentes usadas para, digamos, “cosas tipo-título” o para “cosas tipo-lista”.

Mientras que un color-BEAMER tiene un cierto primer plano y un cierto fondo (cualquiera de ellos puede estar vacío), una fuente-BEAMER tiene un tamaño, una forma, una serie y una familia, cada una de las cuales puede estar vacía. La relación de herencia entre las fuentes-BEAMER no necesariamente es la misma que entre los colores BEAMER, aunque Tantau ha intentado relacionarlos siempre que es posible.

La herencia múltiple desempeña un papel más importante para las fuentes que para los colores. Una fuente puede heredar los atributos de dos fuentes diferentes. Si uno de ellos especifica que la fuente debe ser, digamos, negritas y el otro especifica que la fuente debe ser, digamos, grande, entonces la fuente hija sería grande y en negritas.

Como para los colores, la descripción de la fuente usada para un elemento se da después de la descripción del elemento.

Usando fuentes de Beamer

Para usar una fuente-BEAMER se puede usar el comando `\usebeamerfont`. Dentro de las plantillas para elementos, este comando (generalmente) ya ha sido llamado, así que rara vez se tendrá que usar este comando.

```
\usebeamerfont*{<nombre fuente-beamer>}
```

Este comando cambia la fuente actual por la fuente especificada por `<nombre fuente-beamer>`. El `<nombre fuente-beamer>` no puede ser un texto demasiado elegante y puede contener espacios. Ejemplos típicos son `frametitle` o `section in toc` o `Mi Fuente 1`. La fuente-BEAMER puede tener (y debería) el mismo nombre que las plantillas-BEAMER y los colores-BEAMER.

Ejemplo: `\usebeamerfont{frametitle}`

En la versión sin asterisco de este comando, la fuente se cambia de acuerdo a los atributos especificados en el `<nombre fuente-beamer>`, pero los atributos no especificados permanecen sin cambios. Por ejemplo, si la fuente especifica que la fuente debe ser “negritas” pero no especifica nada más, y si la fuente actual es grande, entonces `\usebeamerfont` causa que la fuente actual se vuelva grande y en negritas.

En la versión con asterisco de este comando, primero se restablece la fuente antes de que se apliquen los atributos de la fuente. De esta manera, en el ejemplo anterior de una fuente-BEAMER que sólo tiene el atributo “negritas”, poniendo `\usebeamerfont*` *siempre* causa que la fuente actual se convierta en una fuente de tamaño normal, forma normal, en negritas, de la familia predefinida.

Estableciendo fuentes de Beamer

Como para los colores-BEAMER, hay un comando central para editar y cambiar fuentes-BEAMER.

```
\setbeamerfont*{<nombre fuente-beamer>}{<atributos>}
```

Este comando establece o restablece ciertos atributos de la fuente-BEAMER `<nombre fuente-beamer>`. En la versión sin asterisco, este comando sólo añade aquellos atributos que no se han mencionado en una llamada anterior y anula aquellos que se han mencionado. De esta manera, los siguientes dos bloques de comandos tienen el mismo efecto:

Ejemplo:

```
\setbeamerfont{frametitle}{size=\large}
\setbeamerfont{frametitle}{series=\bfseries}

\setbeamerfont{frametitle}{size=\large,series=\bfseries}
```

En la versión con asterisco, primero se restablecen completamente los atributos de la fuente, es decir, se establecen como vacíos.

Se pueden dar los siguiente `<atributos>`:

- `size=<comando de tamaño>` Establece el atributo de tamaño de la fuente-BEAMER.

El `<comando de tamaño>` debe ser un comando de L^AT_EX normal que se usa para establecer el tamaño de la fuente o debe estar vacío. Se incluyen los comandos útiles `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\huge` y `\Huge`. BEAMER también introduce los dos tamaños de fuente `\Tiny` y `\TINY` para texto *realmente* pequeño. Si se les usa se debe saber *exactamente* lo que se está haciendo. Queda la advertencia.

Notamos que hay una diferencia entre especificar un comando vacío y especificar `\normalsize`: hacer el atributo de tamaño “vacío” significa que el tamaño de la fuente no debe cambiar cuando se usa esta fuente, mientras que especificar `\normalsize` significa que el tamaño se debe establecer como el tamaño normal siempre que se usa esta fuente.

- `size*={<tamaño en pt>}{<salto línea de base>}`

Establece el atributo de tamaño de la fuente como el `<tamaño en pt>` dado y el salto de la línea base al valor dado. Notamos que, dependiendo del tipo de fuente que se use, no todos los tamaños de las fuentes están disponibles. También, ciertos tamaños de fuente son mucho menos deseables que otros; los comandos estándar tienen cuidado de elegir los tamaños adecuados. No usar estas opciones a menos de que se tenga una buena razón. Este comando tiene el mismo efecto que `size={\fontsize{<tamaño en pt>}{<salto línea de base>}}`.

- `shape=<comando de forma>` Establece el atributo de forma de la fuente. El comando debe ser como alguno de los comandos `\itshape`, `\slshape`, `\scshape` o `\upshape`.
- `shape*={<abreviación de atributo de forma>}` Establece el atributo de forma de la fuente usando la abreviación de L^AT_EX para los atributos.

Este comando tiene el mismo efecto que `shape={\fontshape{<abreviación de atributo de forma>}}`.

- `series=<comando de serie>` Establece el atributo de serie de la fuente. El comando debe ser como el comando `\bfseries`.

- **series*={⟨abreviación de atributo de serie⟩}** tiene el mismo efecto que `series={\fontseries{⟨abreviación de atributo de serie⟩}}`
- **family=⟨comando de familia⟩** Establece el atributo de familia de la fuente. El comando debe ser un comando de fuente-L^AT_EX como `\rmfamily` o `\sffamily`.
- **family*={⟨nombre de familia⟩}** Establece el atributo de familia de fuente como el ⟨nombre de familia⟩ dado. El comando tiene el mismo efecto que `family={\fontfamily{⟨nombre de familia⟩}}`. El ⟨nombre de familia⟩ normalmente es alguna abreviación cifrada de un nombre de familia de fuente que se instaló en algún lugar del sistema. Por ejemplo, el ⟨nombre de familia⟩ para Time pasa a ser `ptm`. Nadie puede recordar estos nombres, así que es perfectamente normal tener que buscarlos laboriosamente.
- **parent={⟨lista de padres⟩}** especifica una lista de fuentes padre. Cuando se usa la fuente-BEAMER primero se usan los padres. Así, cualquier atributo de fuente establecido por uno de los padres se hereda en esta fuente-BEAMER, excepto si la fuente anula este atributo.

Ejemplo:

```
\setbeamerfont{padre A}{size=\large}
\setbeamerfont{padre B}{series=\bfseries}
\setbeamerfont{hijo}{parent={padre A, padre B},size=\small}
```

```
\normalfont
```

Este texto est\’{a} en una fuente normal.

```
\usebeamerfont{padre A}
```

Este texto es grande.

```
\usebeamerfont{padre B}
```

Este texto es grande y est\’{a} en negritas.

```
\usebeamerfont{padre B}
```

Este texto es todav\’{i}a m\’{a}s grande y en negritas.

```
\usebeamerfont*{padre B}
```

Este texto s\’{o}lo est\’{a} en negritas, pero no es grande.

```
\usebeamerfont{hijo}
```

Este texto es pequeño y est\’{a} en negritas.

Parte IV

Creando material de apoyo.

El objetivo de la clase BEAMER es simplificar la creación de presentaciones usando un proyector. Sin embargo, una presentación raramente está aislada. El material que acompaña a una presentación incluye:

- Las presentaciones normalmente deberían estar acompañadas de *folletos*, texto escrito que el público puede leer durante o después de que se da la presentación.
- Quisiéramos tener notas que idealmente se muestren en la pantalla de nuestra computadora mientras el público ve la presentación.
- Quisiéramos crear una copia impresa de la presentación para checar posibles errores.
- Quisiéramos crear una versión en transparencia de la charla.

Esta parte discute cómo BEAMER nos ayuda con la creación de lo antes mencionado.

18. Añadiendo notas para el expositor

Una *nota* es texto que pretende ser un recordatorio para el expositor, de lo que debe decir o debe tener presente al presentar una diapositiva. Normalmente las notas se imprimen sobre papel, pero con el apoyo de dos pantallas éstas se pueden mostrar sobre el monitor de una computadora portátil mientras la presentación principal se muestra en el proyector.

18.1. Especificando el contenido de la nota.

Para añadir una nota a una diapositiva o a un marco, se usa el comando `\note`. Este comando se puede usar tanto dentro como fuera de los marcos, pero según sea el caso se comporta de manera totalmente diferente: dentro de los marcos los comandos `\note` se acumulan y añaden una sola página de nota después de la diapositiva actual; fuera de los marcos cada `\note` inserta directamente una sola página de nota con el parámetro dado como contenido. Es preferible usar el comando `\note` dentro de los marcos que fuera, ya que sólo los comandos emitidos dentro de los marcos se benefician de la opción de clase `onlyslideswithnotes`.

Dentro de un marco, el efecto de `\note<texto>` es el siguiente: cuando se usa en alguna parte dentro del marco en una diapositiva específica, se crea después de la diapositiva una página de nota conteniendo el `<texto>`. Dado que se puede añadir una especificación de cubierta al comando `\note`, se puede especificar después de que diapositiva se debe mostrar la nota. Si se usan múltiples comandos en una diapositiva, se “acumulan” y se muestran en la misma nota.

Para mejorar la acumulación de notas se puede usar el comando `\note` con la opción `[item]`. Las notas que se añaden con esta opción se acumulan en una lista `enumerate` que sigue a cualquier texto que se inserta usando `\note`.

El siguiente ejemplo produce una página de nota que tiene dos entradas y que sigue a una segunda diapositiva.

```
\begin{frame}
  \begin{itemize}
    \item<1-> Huevos
    \item<2-> Plantas
      \note[item]<2>{Cuenta el chiste acerca de las plantas.}
      \note[item]<2>{¡Hazlo rápido!.}
    \item<3-> Animales
  \end{itemize}
\end{frame}
```

Fuera de los marcos el comando `\note` crea una sola página de nota, que es “independiente” de cualquier uso de los comandos `\note` dentro del marco previo. Si se pone `\note` dentro de un marco y otro `\note` justo después de él, se crean *dos* páginas de nota.

En lo que sigue se describe la sintaxis y efectos del comando `\note` dentro de los marcos:

`\note<especificación de cubierta>[<opciones>]{<texto nota>}`

Efectos *dentro* de los marcos:

Este comando añade `<texto nota>` a la nota que sigue a la diapositiva actual. Se acumulan los usos múltiples de este comando en una diapositiva. Si no se da una `<especificación de cubierta>`, la nota se añade a *todas* las diapositivas del marco actual. A menudo esto no es lo que se quiere, así que generalmente es una buena idea añadir una especificación como `<1>`.

Se pueden dar las siguientes `<opciones>`:

- **item** hace que se ponga la nota como un objeto en una lista que se muestra al final de la página de nota.

Ejemplo: `\note<2>{No hablar más de dos minutos acerca de esto.}`

ARTÍCULO Las notas se ignoran en este modo.

A continuación se describe la sintaxis y los efectos del comando `\note fuera` de los marcos.

`\note[opciones]{texto nota}`

Fuera de los marcos este comando crea una página de nota. Este comando no es afectado por la opción `notes= onlyframeswithnotes`.

Se pueden dar las siguientes *opciones*:

- **itemize** por conveniencia encierra la página de nota completa en un ambiente `itemize`.
- **enumerate** encierra la página de nota completa en un ambiente `enumerate`.

Ejemplo:

```
\frame{alg\'}{u}n texto}
\note{No hablar m\'}{a}s de un minuto.}
\note[enumerate]
{
\item Recalca esto primero.
\item Luego esto.
}
```

ARTÍCULO Las notas se ignoran en este modo.

Los siguientes elementos dictan cómo se representan las páginas de nota:

Plantilla-/Color-/Fuente-Beamer note page

Esta plantilla se usa para editar una página de nota. La plantilla debe contener una mención de la inserción `\insertnote`, el cual contiene el texto de la nota. Para apretujar la página de nota se puede reducir el tamaño de la fuente `BEAMER note page`. El valor por defecto es `\small`.

Las siguientes opciones de plantilla están predefinidas:

- **[default]** La plantilla predefinida muestra la última diapositiva en la esquina superior derecha y algunos “consejos” que deben ayudar a hacer coincidir una página de nota con la diapositiva que se muestra actualmente.
- **[compress]** La opción produce una salida similar a la predefinida, sólo que encaja más texto en cada página de nota a cambio de la legibilidad.
- **[plain]** Sólo inserta el texto de la nota, ningún consejo de adorno.

Las siguientes dos inserciones son útiles para las páginas de nota:

- `\insertnote` inserta dentro de la plantilla el texto de la nota actual.
- `\insertslideintonotes{magnificación}` Inserta una “mini imagen” de la última diapositiva en la nota actual. Se escalará la diapositiva a la magnificación dada.

Ejemplo: `\insertslideintonotes{0.25}`

Esto da una mini diapositiva cuyo ancho y altura son un cuarto del tamaño usual.

18.2. Especificando contenido para múltiples notas.

A veces se quiere que algún texto se muestre en cada nota o al menos en cada nota de una larga serie de notas. Para lograr este efecto, se pueden usar los dos siguientes comandos:

`\AtBeginNote{<texto>}`

El texto se inserta al principio de cada nota dentro del alcance del comando. Para detener el efecto se usa `\AtBeginNote{}` o se encierra el área en un grupo `TEX`.

Es aconsejable agregar un comando `\par` o una línea vacía al final del `<texto>` ya que de otra forma cualquier texto de nota seguirá directamente al `<texto>` sin un cambio de línea.

Ejemplo:

```
\section{Mi Secci\'}{o}n

\AtBeginNote{Finalizar esta secci\'}{o}n a las 14:35.\par}
\begin{frame}
  ...
  \note{alguna nota}
\end{frame}
\begin{frame}
  ...
  \note{alguna otra nota}
\end{frame}
\AtBeginNote{}
```

`\AtEndNote{<texto>}`

Este comando se comporta de la misma manera que `\AtBeginNote` sólo que el texto se inserta al final (parte inferior). Se puede agregar `\par` al principio del `<texto>`.

18.3. Especificando qué notas y marcos se van a mostrar.

Dado que normalmente no se desea que las notas sean parte de la presentación, se debe indicar explícitamente en el preámbulo si se incluirán las notas en la presentación. Para esto se pueden usar las siguientes opciones de BEAMER:

`\setbeameroption{hide notes}`

No se muestran las notas. En una presentación ésta es la opción predefinida.

`\setbeameroption{show notes}`

Incluye las notas en el archivo de salida. También se incluyen las diapositivas normales y las páginas de nota se intercalan con ellas.

`\setbeameroption{show notes on second screen=<ubicación>}`

Cuando se da esta opción se crea una doble pantalla de la charla (ver Sección 21). La segunda pantalla, que por defecto se despliega a la derecha, muestra las notas. Al especificar una `<ubicación>` diferente, la segunda pantalla también se puede colocar a la izquierda, en la parte inferior o en la parte superior.

Ejemplo:

```
\documentclass{beamer}
\usepackage{pgfpages}
\setbeameroption{show notes on second screen}
\begin{document}
\begin{frame}
  Un marco.
```

```

\note{Esto se muestra a la derecha.}
\end{frame}
\end{document}

```

En detalle, pasa lo siguiente: la presentación se edita de manera normal y se muestra en la pantalla principal, o para ser precisos, en la página lógica de `pgfpage` número cero. La segunda pantalla (pantalla lógica número uno) se inicializa vacía.

Siempre que se edita una página de nota, ya sea porque un marco contiene el comando `\note` o porque el marco estaba seguido de un comando `\note`, se hace de manera normal. Luego la página de nota se pone en la segunda pantalla. Luego la página completa se envía al exterior. (Los detalles exactos son un poco más complejos pero básicamente esto es lo que sucede).

Un efecto importante de este comportamiento es que una página de nota *siguiendo* un marco se muestra al lado de éste. Normalmente esto es exactamente lo que se quiere y espera. Sin embargo, si hay múltiples páginas de nota para una sola diapositiva, sólo se muestra la última. Esto puede cambiar en un futuro, así que no confíe en este efecto.

Ejemplo:

```

\begin{frame}
  Primer marco.
\end{frame}
\note{Esta no se muestra en absoluto (actualmente).}
\note{Esta nota se muestra junto con el primer marco.}

\begin{frame}
  Segundo marco.
  \note{Esta nota se muestra junto con el segundo marco.}
\end{frame}

\begin{frame}
  Ning\,'{u}n texto de nota se muestra en este marco.
\end{frame}

```

Si realmente se necesitan múltiples páginas de nota para una sola diapositiva, se tendrá que usar algo más complicado como esto:

```

\begin{frame}<1-3>
  Primer marco.
  \note<1>{Primera p\,'{a}gina de nota para este marco.}
  \note<2>{Segunda p\,'{a}gina de nota para este marco.}
  \note<3>{Tercera p\,'{a}gina de nota para este marco.}
\end{frame}

```

```
\setbeameroption{show only notes}
```

Incluye sólo las notas en el archivo de salida y suprime todos los marcos. Esta opción es útil para imprimirlas. Si se especifica este comando *no* se actualizan los archivos `.aux` y `.toc`. Así que si se añade una sección y se vuelve a compilar la presentación, esto no se reflejará en las barras de navegación (que de cualquier forma no veremos ya que las notas son la única salida).

19. Creando transparencias.

El objetivo principal de la clase `BEAMER` es crear presentaciones para proyectores. Sin embargo, a menudo es útil imprimir transparencias como respaldo en caso de que falle la computadora u otra cosa. Una versión de transparencias de una plática a menudo tiene menos diapositivas que la versión principal dado que toma más tiempo cambiar las diapositivas, pero puede tener más diapositivas que la versión folleto. Por ejemplo, mientras que en un folleto una animación se puede condensar en una sola diapositiva, para la versión en transparencias se podrían imprimir varias diapositivas.

Para crear una versión de transparencias se especifica la opción de clase `trans`. Si no se especifica nada más, se eliminarán todas las especificaciones de cubierta. Para la mayoría de los casos esto crea de manera exacta el resultado deseado.

```
\documentclass[trans]{beamer}
```

Creará una versión que usa las especificaciones de cubierta `trans`.

En algunos casos se quiere un comportamiento más complejo. Por ejemplo, cuando se usan muchos comandos `\only` para dibujar una animación, no es una buena idea anular todas las especificaciones de cubierta, ya que ocasionará que se muestren al mismo tiempo todos los pasos de la animación. En algunos casos esto no es deseable. También se podría querer anular algunos comandos `\alert` que aplican sólo a diapositivas específicas en el folleto.

Para un control más fino de lo que se debe mostrar en un folleto, se pueden usar las *especificaciones de modo*. Éstas especifican qué diapositivas de un marco se deben mostrar en una versión especial, por ejemplo para la versión de folleto. Como se explicó en la Sección 8.2, una especificación de modo se escribe al lado de una especificación de cubierta normal dentro de paréntesis angulares. Está separada de la especificación normal por una barra vertical y un espacio. Aquí hay un ejemplo:

```
\only<1-3,5-9| trans:2-3,5>{Texto}
```

Esta especificación dice: “Normalmente (en modo `beamer`), inserta el texto en las diapositivas 1 – 3 y 5 – 9. Para la versión de transparencias, inserta el texto sólo en las diapositivas 2, 3 y 5”. Si no se da ninguna especificación de modo especial para el modo `trans` por defecto es “siempre”. Esto ocasiona el efecto deseado de que si no se especifica nada, la especificación de cubierta se elimina eficazmente para el folleto.

Una especificación especialmente útil es la siguiente:

```
\only<3| trans:0>{No se muestra en las transparencias.}
```

Dado que no hay diapositiva cero, el texto no se muestra. De la misma forma, `\alert<3-|trans:0>{Texto}` no resaltará el texto en una transparencia.

También se puede usar una especificación de modo para la especificación de cubierta del ambiente `{frame}` como en el ejemplo siguiente:

```
\begin{frame}<1-| trans:0>
  Texto...
\end{frame}
```

Esto ocasiona que se elimine el marco en la versión de transparencias. También se puede restringir la presentación de tal manera que se muestren en el folleto sólo diapositivas específicas del marco:

```
\begin{frame}<1-| trans:4-5>
  Texto...
\end{frame}
```

También es posible dar sólo una especificación de cubierta alterna. Por ejemplo, `\alert<trans:0>{...}` causa que se resalte el texto durante la presentación, pero nunca en la versión de transparencias. Así mismo, el comando `\frame<trans:0>{...}` causa que se elimine el marco para el folleto.

Finalmente hacemos notar que es posible dar más de una especificación de cubierta alterna y en cualquier orden. Por ejemplo, la siguiente especificación establece que el texto se debe insertar en las primeras tres diapositivas de la presentación, en las primeras dos diapositivas de la versión de transparencias y no en el folleto.

```
\only<trans:1-2| 1-3| handout:0>{Texto}
```

Si se desea dar la misma especificación en todas las versiones, se puede hacer especificando all. Por ejemplo,

```
\frame<all:1-2>{bla, bla, .....} asegura que el marco tiene dos diapositivas en todas las versiones.
```

20. Creando folletos y Notas de la Conferencia

Durante una presentación es muy deseable que el público tenga disponible un *folleto* o incluso las *notas de la conferencia*. Un folleto le permite al público retroceder de manera individual a las cosas que no haya entendido.

Los folletos siempre se deben proporcionar *lo más pronto posible*, de preferencia semanas antes de la plática. *No* retener el folleto hasta al final de la plática.

La clase BEAMER ofrece dos maneras diferentes de crear versiones especiales de la plática y se discuten a continuación. La primera forma es crear una versión de folleto agregando la opción **handout**, que causa que el documento se edite en modo **handout**. “Parecerá” una presentación pero se puede imprimir (se “derriban” las cubiertas). La segunda forma, más complicada y más poderosa, es crear una versión “article” independiente de la presentación. Esta versión coexiste en el archivo principal.

20.1. Creando folletos usando el modo **handout**

La manera más fácil de crear un folleto para el público (aunque no la más deseable) es usar la opción **handout**. Esta opción funciona exactamente igual que la opción **trans**. Un ejemplo elaborado de diferentes especificaciones de cubierta para la presentación, el folleto y las transparencias se puede encontrar en el archivo `beamerexample1.tex`.

```
\documentclass[handout]{beamer}
```

 Crea una versión que usa las especificaciones de cubierta **handout**.

Se puede elegir un color diferente y/o tema de presentación para el folleto.

Al imprimir un folleto creado de esta manera, generalmente se quieren por lo menos dos y posiblemente hasta cuatro diapositivas en cada página. Presumiblemente la manera más fácil de hacerlo es usar `pgfpages` como sigue:

```
\usepackage{pgfpages}
\pgfpagelayout{2 on 1}[a4paper,border shrink=5mm]
```

En lugar de 2 en 1 se puede usar 4 en 1 (pero entonces se tiene que agregar `landscape` a la lista de opciones) y se pueda usar, digamos, `letterpaper` en lugar de `a4paper`.

20.2. Creando folletos usando el modo **article**.

En lo que sigue, la “versión article” de la presentación se refiere a un texto en T_EX normal usando, por ejemplo, la clase de documento **article** o quizás `llncs` o una clase de documento similar. Esta versión de la presentación generalmente sigue diferentes reglas de edición e incluso puede tener una estructura diferente. No obstante, se puede querer que esta versión coexista con la presentación en un mismo archivo y se puede querer compartir alguna parte de ella (como una figura o una fórmula) con la presentación.

En general, la versión article de una plática se ve mejor como folleto, que el folleto que se crea usando el modo sencillo **handout**, dado que es más económico y puede incluir más información a fondo.

Iniciando el modo **article**

El modo de artículo de una presentación se crea al especificar **article** o **book** o alguna otra clase como la clase de documento en lugar de BEAMER y al cargar luego el paquete `beamerarticle`.

El paquete `beamerarticle` define virtualmente todos los comandos de BEAMER de una manera que es consciente para el modo **article**. También se pueden dar especificaciones de cubierta a comandos como `\textbf` o `\item` una vez que se ha cargado `beamerarticle`. Hacemos notar que, excepto para `\item`, estas especificaciones de cubierta también funcionan: al escribir `\section<presentation>{Nombre}` se suprime este comando de sección en la versión de artículo. Para saber los efectos exactos que tienen las especificaciones de cubierta en el modo **article**, se deben ver las descripciones de los comandos a los

cuales se las queremos aplicar.

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

Hace que la mayoría de los comandos de BEAMER estén disponibles para otra clase de documento. Se pueden dar las siguientes *opciones*:

- **activeospeccharacters** permite que el código del carácter de los paréntesis angulares como se especificó por otros paquetes. Normalmente BEAMER desactiva el comportamiento especial de los dos caracteres < y >. Al usar esta opción, se reinstala su comportamiento original pagando el precio de posibles problemas cuando se usen especificaciones de cubierta en el modo **article**.
- **noamsthm** suprime la carga del paquete **amsthm**. No se definirá ningún teorema.
- **notheorem** suprime la definición de los ambientes estándar como **theorem**, pero todavía se carga **amsthm** y el comando **\newtheorem** todavía hace que los ambientes que definimos sean conscientes de especificación de cubierta. El usar esta opción nos permite definir ambientes estándar en la forma que queramos, mientras se mantiene el poder de las extensiones a **amsthm**.
- **envcountsect** causa que teoremas, definiciones, etc., se numeren respecto a cada sección. Así, en vez de Teorema 1 se tiene Teorema 1.1. Se recomienda utilizar esta opción.
- **noxcolor** suprime la carga del paquete **xcolor**. No se definirá ningún color.

Ejemplo:

```
\documentclass{article}
\usepackage{beamerarticle}
\begin{document}
\begin{frame}
\frametitle{Un t\’{i}tulo de marco}
\begin{itemize}
\item<1-> Se pueden usar especificaciones de cubierta.
\item<2-> Esto es \’{u}til.
\end{itemize}
\end{frame}
\end{document}
```

Hay un problema restante: mientras que la versión de **article** puede pasar fácilmente todo el archivo a TeX, incluso en la presencia de comandos como **\frame<2>**, no queremos que el texto de artículo especial se inserte en nuestra presentación en BEAMER original. Eso significa que nos gustaría se eliminara todo el texto *entre* marcos (no el de adentro). Más precisamente, queremos que se elimine todo el texto, excepto los comandos como **\section**, etc. Se puede aplicar este comportamiento al especificar la opción **ignorenonframetext** en la versión de presentación. La opción inserta un **\mode*** al principio de la presentación.

El siguiente ejemplo muestra un uso simple del modo **article**.

```
\documentclass[a4paper]{article}
\usepackage{beamerarticle}
%\documentclass[ignorenonframetext,red]{beamer}

\mode<article>{\usepackage{fullpage}}
\mode<presentation>{\usetheme{Berlin}}

% para todo mundo:
\usepackage[english]{babel}
\usepackage{pgf}
```



```
\pgfdeclareimage[height=1cm]{miimagen}{nombredearchivo}
```

```
\begin{document}
```

```
\section{Introducci\'}{o}n}
```

Éste es el texto de introducción. Este texto no se muestra en la presentación, pero será parte del artículo.

```
\begin{frame}
```

```
\begin{figure}
```

```
% En el artículo, esta es una figura flotante,
```

```
% En la presentación, esta figura se muestra en el primer marco
```

```
\pgfuseimage{mimagen}
```

```
\end{figure}
```

```
\end{frame}
```

Una vez más este texto no se muestra en la presentación.

```
\section{Parte principal}
```

Mientras que este texto no se muestra en la presentación, el comando `section` también aplica a la presentación.

Podemos agregar una subsección que `s\'}{o}lo` es parte del `art\'}{i}culo` así:

```
\subsection<article>{Secci\'}{o}n sólo del artículo}
```

Con un poco más texto.

```
\begin{frame}
```

```
Este texto es parte tanto del artículo como de la presentación.
```

```
\begin{itemize}
```

```
\item Esta cosa tambi\'}{e}n se muestra en ambas versiones.
```

```
\item Esto tambi\'}{e}n.
```

```
\only<article>{\item Este objeto particular s\'}{o}lo es parte  
de la version article.}
```

```
\item<presentation:only@0> Este texto también s\'}{o}lo es parte del art\'}{i}culo.
```

```
\end{itemize}
```

```
\end{frame}
```

```
\end{document}
```

Hay un comando cuyo comportamiento es un poco especial en el modo `article`: el comando de cambio de línea `\\`. Dentro de los marcos este comando no tiene efecto en el modo `article`, excepto si está presente una especificación de cubierta. Entonces tiene el efecto normal dictado por la especificación. La razón de este comportamiento es que típicamente se insertan muchos comandos `\\` en una presentación para tener control sobre todos los cambios de línea. Estos cambios de línea están de más principalmente en el modo `article`. Si realmente se quiere un cambio de línea que aplique en todas las versiones, se escribe `\\<all>`. Notamos que el comando `\\` a menudo se redefine a través de ciertos ambientes para que no siempre sea consciente de especificación de cubierta. En tal caso se tiene que escribir algo como: `\only<presentation>{\\}`.

Flujo de trabajo

Los siguientes pasos de flujo de trabajo son opcionales pero pueden simplificar la creación de la versión de artículo.

- En el archivo principal `principal.tex` se borra la primera línea que es la que establece la clase del documento.
- Crear un archivo llamado, digamos, `principal.beamer.tex` con el siguiente contenido:

```
\documentclass[ignorenonframetext]{beamer} \input{principal.tex}
```

- Crear un archivo extra llamado `principal.artículo.tex` con el siguiente contenido:

```
\documentclass{article}
\usepackage{beamerarticle}
\setjobnamebeamerversion{principal.beamer}
\input{principal.tex}
```

- Ahora se puede ejecutar `pdflatex` o `latex` en los dos archivos `principal.beamer.tex` y `principal.artículo.tex`.

El comando `\setjobnamebeamerversion` le dice a la versión `article` dónde encontrar la versión de presentación. Esto es necesario si se desea incluir en un artículo diapositivas (como figuras) de la versión de presentación.

`\setjobnamebeamerversion`{*<nombre de archivo sin extensión>*}

Le dice a la clase BEAMER dónde encontrar la versión de presentación del archivo actual.

Un ejemplo de este flujo de trabajo se puede encontrar en el subdirectorio `examples` con el archivo `beamerexample2`.

Incluyendo diapositivas de la versión de presentación en la versión de artículo.

Si se usa el paquete `beamerarticle`, el comando `\frame` vuelve estar disponible en modo `article`. Al adaptar la plantilla de marco, en el artículo se puede “imitar” la apariencia de marcos editados con BEAMER. Sin embargo, a veces se quiere insertar “la cosa real” en la versión `article`, esto es, una “captura de pantalla” precisa de una diapositiva de la presentación. Los comandos que se introducen a continuación nos ayudan a hacer esto.

Para incluir una diapositiva de la presentación en la versión de artículo se deben hacer dos cosas: primero se le debe poner a la diapositiva una etiqueta de L^AT_EX normal usando el comando `\label`. Dado que este comando es consciente de especificación de cubierta, también se pueden seleccionar diapositivas específicas de un marco. Al agregar a un marco la opción `label=<nombre>` automáticamente se añade una etiqueta `<nombre><número de diapositiva>` a cada diapositiva del marco.

Una vez que se ha etiquetado la diapositiva, en la versión de artículo se puede usar el siguiente comando para insertarla dentro de él:

`\includeslide`[*<opciones>*]{*<nombre de la etiqueta>*}

Este comando llama a `\pgfimage` con las opciones dadas para el archivo especificado por

```
\setjobnamebeamerversion<nombre de archivo>
```

Además, la opción `page=<página del nombre de la etiqueta>` se pasa a `\pgfimage`, donde

\langle página del nombre de la etiqueta \rangle se lee internamente del archivo \langle nombre de archivo \rangle .smm.

Ejemplo:

```
\article
\begin{figure}
\begin{center}
\includeslide[height=5cm]{diapositiva1}
\end{center}
\caption{La primera diapositiva (altura 5cm). Note el segundo objeto parcialmente
cubierto.}
\end{figure}
\begin{figure}
\begin{center}
\includeslide{diapositiva2}
\end{center}
\caption{La segunda diapositiva (tamaño original). Ahora tambi\''{e}n se muestra el
segundo objeto}
\end{figure}
```

El efecto exacto de pasar la opción `page= \langle página del nombre de la etiqueta \rangle` al comando `\pgfimage` se explica en la documentación de `pgf`. En esencia, pasa lo siguiente:

- Para la versión antigua de `pdflatex` y para cualquier versión de `latex` junto con `dvips`, el paquete `pgf` busca un archivo llamado \langle nombre de archivo \rangle .page \langle página del nombre de la etiqueta \rangle . \langle extensión \rangle
- Para cada página de archivo `.pdf` o `.ps` que se va a incluir de esta manera, tal archivo se debe crear a mano. Por ejemplo, si el archivo PostScript de la versión de presentación se llama `principal.beamer.ps` y se desea incluir las diapositivas con números de página 2 y 3, se deben crear “a mano” los archivos de una sola página `principal.beamer.página2.ps` y `principal.beamer.página3.ps`. Si no se pueden encontrar estos archivos, `pgf` se quejará.
- Para las versiones recientes de `pdflatex`, `pdflatex` también busca los archivos según el esquema antes mencionado. Sin embargo, si no los encuentra (porque no se han creado), usa un mecanismo especial para extraer directamente la página deseada del archivo de presentación `principal.beamer.pdf`.

20.3. Detalles sobre los Modos

Aquí se describe cómo funcionan exactamente los modos y cómo se puede usar el comando `\mode` para controlar que parte del texto pertenece a que modo. Cuando BEAMER edita el texto, siempre está en alguno de los siguientes cinco modos:

- `beamer` es el modo predefinido.
- `second` es el modo que se usa cuando se está editando una diapositiva para una segunda pantalla opcional.
- `handout` es el modo para crear folletos.
- `trans` es el modo para crear transparencias.
- `article` es el modo cuando se ha transferido el control a otra clase, como `article.cls`. Notamos que el modo también es `article`, si se transfiere el control a, digamos, `book.cls`.

En adición a estos modos, BEAMER reconoce los siguientes nombres para conjuntos de modos:

- `all` se refiere a todos los modos.

- `presentation` se refiere a los primeros cuatro modos.

Dependiendo del modo actual, se puede querer que se inserte cierto texto sólo en ese modo. Por ejemplo, se puede omitir un determinado marco o tabla de la versión `article`. En algunas situaciones se puede usar el comando `\only` para este propósito. Sin embargo, el comando `\mode` que se describe a continuación es mucho más poderoso que `\only`.

En realidad hay tres tipos de este comando, que difieren ligeramente sólo en la sintaxis. El primero y más simple, es la versión que toma un argumento. Esencialmente se comporta de la misma manera que `\only`.

`\mode` \langle *especificación de modo* \rangle $\{$ *texto* $\}$

Causa que se inserte el \langle *texto* $\}$ sólo en los modos indicados. Se recuerda que una \langle *especificación de modo* \rangle es sólo una especificación de cubierta en el que no se menciona ninguna diapositiva.

El \langle *texto* $\}$ no debe hacer algo extravagante que involucre cambios de modos o incluya otros archivos. En particular, no se debe poner un comando `\include` dentro del \langle *texto* $\}$.

Ejemplo:

```
\mode<article>{Detalle extra mencionado sólo en la versión article.}

\mode
<beamer| trans>
{\frame{\tableofcontents[currentsection]}}
```

El segundo tipo del comando `\mode` no toma ningún argumento. “Ningún argumento” significa que no está seguido por una llave izquierda (`{`), sino de cualquier otro símbolo.

`\mode` \langle *especificación de modo* \rangle

En el modo especificado, este comando realmente no tiene efecto. La parte interesante es el efecto en los modos no especificados: en estos modos, el comando ocasiona que \TeX entre en un tipo de estado de agotamiento. Ahora se ignoran todas las líneas siguientes hasta la siguiente línea que tiene una sola ocurrencia de uno de los siguientes comandos: `\mode`, `\mode*`, `\begin{document}`, `\end{document}`. Incluso un comentario en esta línea hará que \TeX salte tal comando. Notamos que la línea con los comandos especiales que hacen que \TeX detenga el agotamiento, no puede seguir directamente a la línea donde inició el agotamiento. Más bien, debe haber una línea no vacía antes del comando o por lo menos dos líneas vacías.

Cuando \TeX encuentra un sólo comando `\mode` entonces ejecuta este comando. Si el comando es un comando `\mode` del primer tipo, \TeX reanuda su estado de “agotamiento” después de haber insertado (o no insertado) el argumento del comando `\mode`. Si el comando `\mode` es del segundo tipo, toma el control.

Usar este segundo tipo de `\mode` es menos conveniente que el primero, pero hay diferentes razones por las que se le podría necesitar:

- El agotamiento por línea es mucho más rápido que la del tercer tipo (ver adelante).
- El primer tipo lee completamente su argumento. Esto significa que no puede contener cualquier texto verbatim que contenga una sola llave.
- El primer tipo no puede aguantar argumentos que contienen `\include`.
- Si el texto principalmente pertenece a un modo con pequeñas cantidades de texto de otro modo insertado, este segundo tipo es agradable de usar.

Nota: Al buscar por líneas un comando `\mode` para sacudirlo de su estado de agotamiento, \TeX no reconoce un comando `\mode` si le sigue una especificación de modo en la misma línea. Así, tal especificación

se debe dar en la siguiente línea.

Nota: Cuando finaliza un archivo en T_EX, T_EX no debe estar en estado de agotamiento. Se desactiva este estado usando `\mode` en una línea y `<all>` en la siguiente.

Ejemplo:

```
\mode<article>
```

```
Este texto se edita sólo en modo |article|.
\verb!texto literal está bien {!
```

```
\mode
<presentation>
{% este texto sólo se inserta en el modo de presentación
\frame{\tableofcontents[currentsection]}}
```

```
Aquí estamos de vuelta al modo de artículo. Este texto
no se inserta en el modo de presentación
```

```
\mode
<presentation>
```

```
Este texto sólo se inserta en el modo de presentación.
```

```
El último tipo de comando mode se comporta completamente diferentemente.
```

\mode*

El efecto de este modo es ignorar todo el texto fuera de los marcos en los modos `presentation`. En modo de artículo no tiene efecto.

Este modo se debe introducir sólo fuera de los marcos. Una vez introducido, si el modo actual es un modo `presentation`, T_EX entra en un estado de agotamiento similar al estado de agotamiento del segundo “tipo” de comando `\mode`. La diferencia es que ahora el texto se lee por tokens y no por líneas. El texto se agota token a token hasta que se encuentra uno de los siguientes tokens: `\mode`, `\frame`, `\againframe`, `\part`, `\section`, `\subsection`, `\appendix`, `\note`, `\begin{frame}`, y `\end{document}` (los dos últimos en realidad son tokens, pero de cualquier manera se les reconoce).

Una vez que se encuentra uno de estos comandos, se detiene el agotamiento y se ejecuta el comando. Sin embargo, todos estos comandos restauran el modo que estaba en efecto cuando iniciaron. Así, una vez que finaliza el comando, T_EX regresa a su agotamiento.

Normalmente, `\mode*` hace exactamente lo que se quiere que T_EX haga fuera de los marcos: ignorar todo (excepto los comandos antes mencionados) fuera de los marcos en el modo `presentation`. Sin embargo, hay casos en los que se tiene que usar el segundo tipo del comando `\mode`: si se tiene texto verbatim que contiene uno de los comandos, si se tiene un texto muy largo fuera de los marcos o si se quiere algún texto fuera de un marco (como una definición) para que se ejecute también en el modo `presentation`.

La opción de clase `ignorenonframetext` activa `\mode*` al principio del documento.

Ejemplo:

```
\begin{document}
\mode*
```

```
Este texto no se muestra en la presentación.
```

```
\begin{frame}
  Este texto se muestra tanto en el modo de artículo como en el de presentación.
\end{frame}
```

De nuevo, este texto no se muestra en la presentación.

```
\section{Este comando tambi\'}{e}n tiene efecto en el modo presentaci\'}{o}n.}
```

De vuelta al material del artículo.

```
\frame<presentation>
{Este marco sólo se muestra en la presentación.}
\end{document}
```

El siguiente ejemplo muestra cómo se pueden incluir otros archivos en un archivo principal. El contenido de un `principal.tex`:

```
\documentclass[ignorenonframetext]{beamer}
\begin{document}
Ésto es materia del modo asterisco.
```

Incluyamos archivos:

```
\mode<all>
\include{a}
\include{b}
\mode*
```

De vuelta al modo asterisco

```
\end{document}
```

Y `a.tex` (y asimismo `b.tex`):

```
\mode*
\section{Primera secci\'}{o}n}
Texto extra en la versión article.
\begin{frame}
  Algún texto.
\end{frame}
\mode<all>
```

21. Tomando ventaja de pantallas múltiples

Esta sección describe las opciones suministradas por BEAMER para sacar ventaja de las computadoras que tienen más de una salida de video y que pueden desplegar diferentes salidas de ellas. Para tales sistemas se puede conectar una salida de video a un proyector mostrando la presentación principal. La segunda salida de video se puede conectar a un pequeño monitor extra (o sólo se visualiza en la pantalla de la computadora) mostrando, por ejemplo, las notas especiales para el expositor. Alternativamente, las dos salidas se pueden conectar a dos distintos proyectores. Se puede mostrar la presentación principal en la primera proyección y en la segunda, la tabla de contenidos. La segunda proyección podría mostrar también la versión traducida en diferentes idiomas o podría mostrar siempre la diapositiva “previa”.

La idea básica detrás del soporte en BEAMER de dos salidas de video es la siguiente: usando opciones especiales se le puede pedir a BEAMER que cree un archivo PDF en el que las “páginas” sean extraordinariamente anchas o altas. Por defecto, su altura es de 128 mm, pero su ancho será de 192 mm (el doble de los usuales 96mm). Estas páginas súper anchas muestran a la izquierda las diapositivas de la presentación principal y a la derecha el material auxiliar (se pueden intercambiar usando las opciones apropiadas, aunque las hiperligas sólo funcionan si la presentación esta a la izquierda y la segunda pantalla a la derecha).

Para la presentación se pueden conectar dos pantallas al sistema. El sistema de ventanas cree que la pantalla es dos veces mas ancha de lo que es realmente. Cualquier cosa que el sistema de ventanas pone en la mitad izquierda de esta gran pantalla virtual se redirecciona a la primera salida de video, todo lo que esta en la mitad derecha se redirecciona a la segunda salida de video.

Cuando el programa de presentación despliega la especialmente preparada presentación-BEAMER superancha, exactamente la mitad izquierda de la pantalla se llenará con la presentación principal, la parte derecha se llenará con el material auxiliar. No todos los programas de presentación soportan esta característica especial. Por ejemplo, Acrobat Reader 6.0.2 sólo usa un modo de pantalla completa en MacOS X.

BEAMER usa el paquete `pgfpages` para editar presentaciones en dos pantallas. Así, el primer paso para crear tal presentación es incluir estos paquetes:

```
\documentclass{beamer}
\usepackage{pgfpages}
```

El siguiente paso es elegir una opción apropiada para mostrar algo especial en la segunda pantalla. Estas opciones se discuten en las siguientes secciones.

Una de las cosas que hacen es establecer una cierta distribución de páginas en formato pgf que es apropiada para las presentaciones en dos pantallas. Sin embargo, todavía se puede cambiar posteriormente tal distribución. Por ejemplo, se podrían agrandar las páginas virtuales. Para más detalles, ver la documentación de `pgfpages`.

21.1. Mostrando notas en la segunda pantalla

Un primer uso es mostrar la presentación en la pantalla principal y las notas en la segunda pantalla. Para hacer esto se usa la opción `shownotes` (ver página ?).

21.2. Mostrando material en un segundo modo en la segunda pantalla.

Un segundo uso es mostrar “una versión diferente” de la presentación en la segunda pantalla. Esta versión diferente podría ser una traducción o simplemente la tabla de contenidos actual.

Para especificar lo que se va a mostrar en la segunda pantalla, se puede usar un modo especial de BEAMER llamado `second`. Este modo se comporta de manera similar a los modos `handout` o `beamer` pero

su efecto depende de las opciones usadas:

```
\setbeameroption{second mode text on second screen=<ubicación>}
```

Esta opción causa que la segunda pantalla muestre el material del segundo modo. La *<ubicación>* de la segunda pantalla puede ser `left`, `right`, `bottom` o `top`.

En detalle, pasa lo siguiente: cuando se necesita editar un nuevo marco, BEAMER verifica si se dio la opción especial `typeset second`. Si no fue así, el marco se edita de manera normal y las diapositivas se ponen en la pantalla de la presentación principal (más precisamente, en la página lógica de `pgfpages` número cero). La segunda pantalla (página lógica número uno) muestra lo que se mostró antes de que se editara el marco.

Si se da la opción de marco especial `typeset second`, después de cada diapositiva del marco el contenido del marco se edita una vez más, pero esta vez para el modo `second`. Esto produce otra diapositiva que se pone en la segunda pantalla (página lógica número uno). Entonces se “emite” la página entera.

El modo `second` se comporta más como el modo `beamer` que otros modos: cualquier especificación de cubierta para `beamer` también aplicará para el modo `second`, a menos que también se de explícitamente una especificación de modo para `second`. En particular, `\only<1-2>{Texto}` se muestra en las diapositivas 1 y 2 en el modo `second`, pero sólo en la primera diapositiva en el modo `handout` o modo `trans`.

Ejemplo:

```
\documentclass{beamer}
\usepackage{pgfpages}
\setbeameroption{second mode text on second screen}
\begin{document}
\begin{frame}[type second]
  Este texto se muestra a la izquierda y a la derecha.
  \only<second>{Este texto se muestra s\'{o}lo a la derecha.}
  \only<second:0>{Este texto se muestra s\'{o}lo a la izquierda.}
\end{frame}
\begin{frame}
  Este texto se muestra a la izquierda. A la derecha se muestra lo mismo que en el
  marco previo.
\end{frame}
\begin{frame}[type second]
  \alt<second>{El comando \string\alt es \'{u}til para el modo second.
  Mostremos la tabla de contenidos, aqu\'{i}: \tableofcontents}
  {Aqu\'{i} viene algo de texto normal para la primera diapositiva.}
\end{frame}
\end{document}
```

El ejemplo siguiente muestra cómo se pueden agregar traducciones de una manera cómoda.

```
\documentclass{beamer}
\usepackage{pgfpages}
\setbeameroption{second mode text on second screen}
\DeclareRobustCommand\traducción[1]{\mitraducción# 1\relax}
\long\def\mitraducción#1|#2\relax{\alt<second>{#2}{#1}}
\title{\traducción{Preparando Presentaciones|Vortr" age vorbereiten}}
\author{Till Tantau}
\begin{document}
\begin{frame}[typeset second]
  \titlepage
\end{frame}
```



```

\begin{frame}[type second]
  \frametitle{\traducción{Esto es el t\'}{i}tulo del marco. |Dies ist der Titel des Rah
mens.}}
  \begin{itemize}
    \item<1-> \traducción{Primero|Erstens}.
    \item<2-> \traducción{Segundo|Zweitens}.
    \item<3-> \traducción{Tercero|Drittens}.
  \end{itemize}
  \traducción{No descubrir l\'}{i}nea por l\'}{i}nea. |Man sollte Text nicht
Zeile f\"ur Zeile aufdecken.}
\end{frame}
\end{document}

```

En la última parte del ejemplo anterior, es un poco tedioso tener que añadir la opción `typeset second` en cada marco. La siguiente opción establece globalmente esta opción:

```
\setbeameroptionalways typeset second mode=<true or false>}
```

Cuando esta opción se establece como `true`, cada marco que sigue tendrá la opción `typeset second` establecida como `true`.

21.3. Mostrando la diapositiva anterior en la segunda pantalla.

```
\setbeameroption{previous slide on second screen=<ubicación>}
```

Esta opción causa que la segunda pantalla muestre la diapositiva anteriormente editada, a menos que esto se anule por un marco con la opción `[typeset second]`. La idea es que si se tienen dos proyectores siempre se puedan presentar las “últimas dos” diapositivas y hablar sobre ellas.

Al usar esta opción se desactiva la actualización de archivos externos como lo es la tabla de contenidos.

Parte V

Los Cómos

Esta parte contiene explicaciones de cómo hacer algunas cosas. Estas explicaciones realmente no son parte del corazón de BEAMER. Más bien, explican como usar BEAMER para lograr un cierto efecto o hacer algo especial.

22. Cómo descubrir cosas por partes

22.1. Descubriendo una enumeración por partes

Un uso común de las cubiertas es mostrar por partes una lista de puntos en una enumeración. La manera más fácil y flexible de hacer esto es como sigue:

```
\begin{itemize}
\item<1-> Primer punto.
\item<2-> Segundo punto.
\item<3-> Tercer punto.
\end{itemize}
```

La ventaja de este método es que se retiene el control total sobre el orden en el que se muestran los objetos. Al cambiar, por ejemplo, la última especificación a <2-> entonces se tienen descubiertos los últimos dos puntos al mismo tiempo.

Una desventaja del método es que se tiene que reenumerar todo si se agrega un nuevo objeto. Esto normalmente no es un problema grave pero puede ser una molestia. Para automatizar el descubrimiento se puede usar el siguiente código:

```
\begin{itemize}[<+>]
\item Primer punto.
\item Segundo punto.
\item Tercer punto.
\end{itemize}
```

El efecto de [<+>] es la de instalar una *especificación de cubierta predefinida*, ver la definición de `itemize`.

Ahora supongamos que se quiere mostrar al mismo tiempo el segundo y tercer punto. Se puede lograr esto agregando la especificación <2-> ya sea al segundo o tercer comando `\item`. Sin embargo, otra vez se tiene que hacer algún reenumerado si se agrega un nuevo objeto al principio. Un mejor método es usar temporalmente una especificación de cubierta diferente y la notación de punto.

```
\begin{itemize}[<+>]
\item Primer punto.
\item[<.->] Segundo punto.
\item Tercer punto.
\end{itemize}
```

Se pueden construir macros propios basándonos en estas ideas (como un ambiente `itemstep` o un comando `\itemcomoelprevio`).

22.2. Resaltando el objeto actual en una enumeración

Si se descubre por partes una enumeración, a veces es una idea buena resaltar el último punto descubierto para atraer la atención del público. Esto se logra como sigue:

```
\begin{itemize}
\item<1-| alert@1> Primer punto.
\item<2-| alert@2> Segundo punto.
\item<3-| alert@3> Tercer punto.
\end{itemize}
o
\begin{itemize}[<+ -| alert@+>]
```

```

\item Primer punto.
\item Segundo punto.
\item Tercer punto.
\end{itemize}

```

Notamos que el pequeño símbolo de objeto se dibuja de color rojo.

22.3. Cambiando el símbolo que aparece en una enumeración.

Al descubrir una lista de tareas o problemas, se puede querer que el símbolo delante del último símbolo descubierto sea una X en una boleta de votar, mientras que para los objetos anteriores sea una marca de control (estos caracteres se encuentran en algunas fuentes Dingbats).

La mejor manera de lograr esto es implementar un nuevo ambiente de acción. Si se activa esta acción, cambia temporalmente la plantilla del símbolo de objeto a otro símbolo:

```

\newenvironment{boletaenv}
{\only{
  \setbeamertemplate{itemize item}{código para mostrar una boleta}%
  \setbeamertemplate{itemize subitem}{código para mostrar una boleta más pequeña}%
  \setbeamertemplate{itemize subsubitem}{código para mostrar una boleta más pequeña}}}
{}

\setbeamertemplate{itemize item}{código para mostrar una marca de control}
\setbeamertemplate{itemize subitem}{código para mostrar una marca de control más pequeña}
\setbeamertemplate{itemize subsubitem}{código para mostrar una marca de control más pequeña}

```

El efecto del código es instalar una marca de control como plantilla predefinida. Si la acción `boleta` es solicitada por algún objeto, esta plantilla se reemplaza temporalmente por las plantillas *boleta*.

Notamos que `boletaenv` se invoca con la especificación de cubierta dada por la acción que le sigue directamente. Esto ocasiona que se invoque `\only` exactamente en las cubiertas indicadas.

Estos son algunos ejemplos de su uso:

```

\begin{itemize}
\item<1-| boleta@1> Primer punto.
\item<2-| boleta@2> Segundo punto.
\item<3-| boleta@3> Tercer punto.
\end{itemize}

```

y

```

\begin{itemize}[<+>| boleta@+>]
\item Primer punto.
\item Segundo punto.
\item Tercer punto.
\end{itemize}

```

En el ejemplo siguiente, cada vez más artículos tienen marca de control de diapositiva a diapositiva:

```

\begin{itemize}[<boleta@+-| visible@1-,+(1)>]
\item Primer punto.
\item Segundo punto.
\item Tercer punto.
\end{itemize}

```

El punto importante es `boleta@+`. El peculiar `visible@1-,+(1)` tiene el efecto siguiente: aunque no tiene ningún efecto respecto a lo que se muestra (después de todo, se aplica a todas las diapositivas), asegura que en la enumeración se mencione la diapositiva número 4. De esta manera habrá una diapositiva en la que los tres puntos tienen marca de control.

22.4. Descubriendo por partes las fórmulas etiquetadas

Supongamos que se tiene una fórmula de tres líneas como la siguiente:

```
\begin{align}
A &= B \\
&= C \\
&= D
\end{align}
```

Descubrir esta fórmula línea por línea es un poco complicado. Una primera idea es usar los comandos `\pause` o `\onslide`. Desgraciadamente éstos no funcionan dado que `align` internamente reprocesa su entrada varias veces, lo cual desordena los delicados interiores de los comandos. Lo siguiente funciona un poco mejor:

```
\begin{align}
A &= B \\
&\uncover<2->{&= C \\
&\uncover<3->{&= D}
\end{align}
```

Desgraciadamente esto no funciona en presencia de etiquetas (funciona para el ambiente `align*`). Lo que sucede es que la etiqueta de la última línea se muestra en todas las diapositivas. El problema aquí es que la etiqueta se crea cuando se encuentra `\\` o cuando se encuentra `\end{align}`. En la última línea éstos ya están “antes del” comando `\uncover`.

Para resolver este problema se puede agregar una línea vacía sin etiqueta y luego insertar un salto vertical negativo para deshacer la última línea:

```
\begin{align}
A &= B \\
&\uncover<2->{&= C \\
&\uncover<3->{&= D \\
&\notag
\end{align}
\vskip-1.5em
```

22.5. Descubriendo una tabla por renglones

Cuando se quiere destapar una tabla renglón por renglón, uno se encuentra con toda clase de problemas si existen líneas verticales y horizontales en la tabla. La razón es que la primera línea vertical en el extremo izquierdo se dibuja incluso antes de que se lea el renglón (y así, en particular, antes de que se pueda leer cualquier comando `\onslide`). Sin embargo, colocar antes `\pause` o `\uncover` al final del renglón tampoco es útil dado que entonces se elimina la línea horizontal debajo de la última línea descubierta.

Una posible manera de resolver este problema es no usar líneas horizontales o verticales. En cambio, una buena alternativa para estructurar la tabla es colorear las líneas usando el paquete `colortbl`. Ver el ejemplo, donde la tabla se descubre por renglones:

```
\rowcolors[] {1}{blue!20}{blue!10}
\begin{tabular}{l!{\vrule}cccc}
Clase & A & B & C & D \\
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6
\end{tabular}
```

```
Z & 5 & 6 & 7 & 8
\end{tabular}
```

Al usar `\onslide` en lugar de `\pause`, se puede conseguir un control mas fino sobre qué línea se muestra en qué diapositiva.

22.6. Descubriendo una tabla por columnas

Al intentar descubrir una tabla columna por columna surgen los mismos problemas que se tenían al descubrir renglón por renglón.

Una vez más, el paquete `colortbl` ofrece una solución. En el siguiente ejemplo, el encabezado de `tabular` se usa para insertar comandos `\onslide`, uno para cada columna, que cubre las entradas en la columna de una cierta diapositiva en adelante. Al final de la última columna, el `\onslide` sin especificación asegura que la primera columna en el siguiente renglón se muestra una vez más de manera normal.

Insertar una línea horizontal es complicado ya que sobrepasa el ancho total de la tabla incluso en la versión cubierta. Lo mejor es sólo usar barras horizontales.

```
\rowcolors[] {1}{blue!20}{blue!10}
\begin{tabular}{l!{\vrule}c<{\onslide<2->}c<{\onslide<3->}
c<{\onslide<4->}c<{\onslide}c}
Class & A & B & C & D \\
X & 1 & 2 & 3 & 4 \\
Y & 3 & 4 & 5 & 6 \\
Z & 5 & 6 & 7 & 8
\end{tabular}
```

23. Conclusiones

En los últimos diez años se ha extendido el uso de \LaTeX en el ambito académico y científico ya que es un lenguaje que permite la elaboración de artículos de investigación, tesis y libros técnicos, con una calidad tipográfica comparable a la de una editorial científica de primera línea. Es como tener una imprenta en casa (por decir lo menos).

Aunado a lo anterior, el lenguaje \LaTeX se destaca por ser considerado software libre, el cual puede ser bajado de cualquier espejo ligado al proyecto CTAN (Comprehensive \TeX Archive Network). El proyecto CTAN constantemente actualiza y aumenta los paquetes disponibles en el lenguaje \LaTeX . En la actualidad, de los nodos principales del servidor de CTAN se descargan algo mas de 6 TeraBytes (1 TB= 2^{40} bytes) al mes, sin contar las descargas de los 75 espejos alrededor del mundo. De hecho, la Facultad de Ciencias de la UNAM cuenta con una máquina que sirve como espejo de CTAN y de la cual se pueden bajar los paquetes ligados a \LaTeX .

Como era de esperarse, el presente trabajo fué editado usando \LaTeX ; se utilizaron una cantidad importante de paquetes para poder insertar gráficos, para insertar código en \LaTeX y para lograr la apariencia final mostrada.

Una vez que el académico ha obtenido su trabajo final y tiene que mostrar sus resultados ante una audiencia amplia, se ve ante la necesidad de preparar una presentación. Debido a los avances tecnológicos, la utilización de acetatos (transparencias) ha caído en desuso y se ha venido fomentando el uso de los proyectores para hacer presentaciones electrónicas, las cuales proporcionan dinamismo, fluidez, y sobre

todo, mejoran la visibilidad.

Es aquí donde entra en escena el paquete BEAMER de L^AT_EX. Dado que el académico quiere preservar la apariencia de sus ecuaciones, fórmulas, gráficos, etc. (cosa que no puede hacer PowerPoint, por ejemplo), BEAMER le proporciona una solución: permite la reutilización de código en L^AT_EX además de que proporciona aspectos visuales que van a apoyar el contenido científico, técnico.

El presente trabajo está pensado para ser un apoyo para la realización de presentaciones desde muy sencillas hasta aquellas que utilizan varios recursos multimedia. Lo más importante, esto no sólo se circunscribe al material escrito, sino que va acompañado de un CD que contiene: los programas necesarios para trabajar, la presente tesis en formato PDF acompañado del código fuente en L^AT_EX que le dió origen, un extenso tutorial (junto con su código fuente) en formato PDF que enseña el uso de BEAMER, además de presentaciones reales de investigadores del Instituto de Matemáticas de la UNAM en Congresos Internacionales (que aprendieron a usar BEAMER en cursos impartidos por el Director de esta tesis, quién utilizó el mencionado tutorial).

Más allá del impacto local que ya ha tenido el presente trabajo, este se potenciará de manera importante ya que todo este material se va a subir al espejo de CTAN ubicado en la Facultad de Ciencias de la UNAM.

Se espera que este material tenga una gran acogida entre la comunidad científica de habla hispana, ya que a veces no dispone de suficiente tiempo para hacer traducciones extensas (más de 200 páginas de texto en inglés), y lo que es mejor, tendrá un tutorial que le enseña el uso de BEAMER y que además le permite copiar y pegar los comandos necesarios para hacer su presentación sin que tenga que aprender L^AT_EX.

Bibliografía

- [1] Gratzer, G., *Math Into Latex*, Birkhauser, tercera edición, pag. 616, 2000.
- [2] Goossens M., Rahtz S., Mittelbach F., *The L^AT_EX Graphics Companion*, Addison-Wesley Professional; primera edición, pags. 608, 1997.
- [3] Knuth D., *TEXbook*, Addison-Wesley Professional; edición Spiral, pags. 496, 1984.
- [4] Lamport L., *LaTeX: A Document Preparation*, System Addison-Wesley Professional; segunda edición, pags. 288, 1994.
- [5] Tantau T., *The beamer class; Manual for version 3.06*, 2005.