

# COMUNICACIÓN

## Presentaciones científicas en formato PDF de Acrobat con L<sup>A</sup>T<sub>E</sub>X: una alternativa a Microsoft Powerpoint

**Cañavate Bernal, Roberto J.**

Profesor Titular de Escuela Interino

Departamento de Métodos Cuantitativos e Informáticos

Universidad Politécnica de Cartagena

E-mail: r.canavate@upct.es

**Cobacho Tornel, M<sup>a</sup> Belén**

Ayudante de Escuela Universitaria

Departamento de Métodos Cuantitativos e Informáticos

Universidad Politécnica de Cartagena

E-mail: belen.cobacho@upct.es

**Sánchez García, Juan Francisco**

Ayudante de Escuela Universitaria

Departamento de Métodos Cuantitativos e Informáticos

Universidad Politécnica de Cartagena

E-mail: jf.sanchez@upct.es

**Dirección de correo de los autores**

Campus Alfonso XIII

Facultad de Ciencias de la Empresa

Paseo Alfonso XII, 50

30203 CARTAGENA

## Resumen

Los editores de textos informáticos diseñados bajo la filosofía WYSIWYG, tales como Microsoft Word, Corel WordPerfect, Lotus WordPro, etc., suelen estar poco y mal adaptados para la inserción de fórmulas y caracteres matemáticos. Si a esto le añadimos otros inconvenientes habituales como su elevado precio, mala portabilidad o gran tamaño de los archivos resultantes, no es de extrañar la rápida expansión que han tenido en los últimos tiempos otros editores científicos alternativos, especialmente  $\text{\LaTeX}$ . Este desarrollo, junto al auge en la utilización de sistemas multimedia y de Internet, ha hecho necesaria la creación de nuevo software específico para la exposición de presentaciones científicas cuya base sean documentos creados en formato  $\text{\LaTeX}$ . Aunque tradicionalmente ha sido escaso el éxito obtenido por el software disponible para tal fin, debido en parte a sus limitadas posibilidades gráficas y dinámicas, en la actualidad existen paquetes que, basándose en presentaciones con formato PDF, generan resultados finales con características estéticas similares a las obtenidas con las aplicaciones tradicionales pero con la ventaja operativa que ofrece el lenguaje  $\text{\LaTeX}$ . En este trabajo se analizarán las capacidades del nuevo software existente para la generación de presentaciones PDF a partir de documentos escritos en  $\text{\LaTeX}$ , y se pondrán de manifiesto las considerables ventajas del mismo, así como sus desventajas frente al programa que con más frecuencia se utiliza en este área: Microsoft PowerPoint.

**Palabras clave:**  $\text{\LaTeX}$ , PDF, Presentación multimedia.

## 1 El editor de textos $\text{\LaTeX}$

Durante los últimos años, especialmente a lo largo de la década de los 90, el lenguaje de programación  $\text{\LaTeX}$  para la edición de textos ha tenido una notable difusión, principalmente en los ámbitos científico, técnico y editorial. Buena prueba de ello es que importantes editoriales científicas (Addison-Wesley, Springer-Verlag, Birkhäuser, etc.) han hecho de  $\text{\LaTeX}$  su principal sistema de impresión. La razón de este auge radica en las significativas ventajas que este lenguaje de programación presenta para la realización de documentos, especialmente libros y textos que contiene

fórmulas matemáticas, y en la extraordinaria calidad tipográfica que se obtiene, muy próxima a la de las imprentas profesionales. Entre las características que explican su superioridad frente a los editores de textos tradicionales (Microsoft Word, Corel WordPerfect, etc.) podemos destacar las siguientes:

- La sencillez y potencia que proporciona para el formateo de fórmulas matemáticas.
- La gratuidad del software. En efecto, tanto el cuerpo principal del programa como los accesorios (*paquetes* de comandos) están disponibles de forma libre en Internet, mayoritariamente en [8].
- Los escasos requerimientos de hardware, que capacitan para su utilización prácticamente a la totalidad de las computadoras.
- La completa portabilidad de los resultados. Inicialmente  $\text{T}_{\text{E}}\text{X}$  (la base sobre la que se construyó  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ) se creó con la intención de ser un lenguaje independiente a la plataforma, es decir, que el aspecto final que aparece en pantalla o en la impresora no dependa de las características del hardware o del software utilizado, y de ahí que las salidas que genera sean archivos con extensión DVI, abreviatura de *device independent*. En este sentido,  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  fue el precursor de otros formatos independientes de la plataforma muy extendidos en la actualidad como PostScript, HTML y PDF, a los que se puede convertir de forma simple y gratuita.
- El reducido tamaño de los archivos resultantes, lo que facilita su almacenamiento e incrementa enormemente su capacidad de transmisión (vía correo electrónico o ftp). Este hecho, junto con la independencia del resultado respecto del sistema, pone de manifiesto la idoneidad de las características de  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  para su uso en Internet.

Sin embargo, frente a los aspectos positivos anteriores existen ciertos inconvenientes que explican el uso extensivo de otros programas alternativos, especialmente

Microsoft Word. Aunque  $\text{\LaTeX}$  no fue concebido como un editor de textos de uso general, la principal causa de su restricción al ámbito científico-técnico (generalmente en conexión con la universidad) proviene de su dificultad de aprendizaje (el formato de textos se realiza mediante líneas de comandos) y sus inicialmente reducidas capacidades gráficas, inconvenientes que, si bien se ha superado en la actualidad con la aparición de numerosos paquetes complementarios, no mejoran las de otros programas encuadrados bajo la filosofía WYSIWYG.

## 2 Transparencias y presentaciones multimedia

La utilización de proyectores de transparencias se ha convertido en una herramienta habitual en ámbitos didácticos, ya sea en exposiciones públicas orales (conferencias, congresos, etc.) o con fines docentes. Sin embargo, su uso se encuentra en claro retroceso en los últimos años debido a la creciente competencia que sufren por parte de las presentaciones multimedia (ejecutables con la ayuda de un ordenador), de mejor calidad y mayores posibilidades, tanto gráficas como dinámicas. Además, las presentaciones multimedia son un vehículo adecuado para cubrir las nuevas necesidades docentes surgidas de la utilización de Internet como método de enseñanza a distancia, lo que les proporciona un importante valor añadido.

Puesto que Microsoft Word es el programa más generalizado para edición de textos, no es de extrañar que sea precisamente Microsoft PowerPoint la utilidad más extendida para la creación de transparencias y presentaciones. Si bien su éxito se debe en gran medida a su sencillez de manejo y a las enormes posibilidades que ofrece, el programa no carece de importantes inconvenientes, la mayor parte de ellos compartidos con Microsoft Word: dificultad para la edición de fórmulas matemáticas, precio elevado, dependencia del sistema, considerables necesidades de hardware y gran tamaño de los archivos finales. A todo esto se añade la frustración de aquellos usuarios de  $\text{\LaTeX}$  que además de encontrar inadecuada la capacidad de formato de fórmulas matemáticas de Microsoft PowerPoint, se encuentran con la imposibilidad de transferir directamente material de un documento generado con  $\text{\LaTeX}$  a una presentación con PowerPoint.

## 2.1 Microsoft PowerPoint y T<sub>E</sub>XPoint

Con el objeto de evitar algunos de estos inconvenientes, especialmente los relacionados con la escritura de fórmulas matemáticas y la utilización de PowerPoint por usuarios de L<sup>A</sup>T<sub>E</sub>X, ha aparecido recientemente una utilidad complementaria a Microsoft PowerPoint denominada T<sub>E</sub>XPoint, que permite incorporar los símbolos propios de L<sup>A</sup>T<sub>E</sub>X en presentaciones realizadas con PowerPoint. Para ello dispone de dos formas de adición de los objetos L<sup>A</sup>T<sub>E</sub>X:

- Modo línea. En este modo T<sub>E</sub>XPoint incorpora una nueva fuente tipográfica a Windows que posee los símbolos de L<sup>A</sup>T<sub>E</sub>X, ya que muchos de los mismos no existen en Windows. Uno de los inconvenientes más importantes de este método es que sólo se puede utilizar en cuadros de texto, por lo que no es posible realizar directamente construcciones complejas como fórmulas presentadas paso a paso, fracciones, etc.
- Modo presentación. Con este procedimiento T<sub>E</sub>XPoint convierte cualquier texto L<sup>A</sup>T<sub>E</sub>X en una imagen que se inserta en la presentación en curso. Para conseguir tal fin, compila el documento internamente con L<sup>A</sup>T<sub>E</sub>X generando el correspondiente fichero DVI, lo transforma en un fichero **PostScript** y finalmente lo convierte en una imagen de mapa de bits que es la que se incorpora en la presentación.

La principal limitación de este método radica en que la imagen generada no es escalable, por lo que se observa una importante pérdida de calidad si se modifica su tamaño. Además, la utilización de este modo requiere que el ordenador tenga instalada una distribución de L<sup>A</sup>T<sub>E</sub>X y el programa Ghostscript (se puede descargar gratuitamente de [7]) con una versión igual o superior a 6.

En cualquier caso, el uso de T<sub>E</sub>XPoint no solventa muchas de las objeciones ya comentadas para PowerPoint, de modo que no puede considerarse una herramienta totalmente satisfactoria, pero sí una herramienta intermedia entre los usuarios de L<sup>A</sup>T<sub>E</sub>X y PowerPoint.

## 2.2 Alternativas con L<sup>A</sup>T<sub>E</sub>X

La primera opción desarrollada expresamente para la creación de transparencias con L<sup>A</sup>T<sub>E</sub>X es la clase de documento `slides` que viene incorporada en su distribución básica. Sus posibilidades son muy reducidas, e incluso su creador, Leslie Lamport, desaconsejaba en [4] su uso de manera generalizada. Esta debilidad produjo la pronta aparición de nuevos paquetes para la impresión de transparencias con ostensibles mejoras respecto al anterior, entre los que cabe destacar las clases `FoilTEX` y `seminar`. Sin embargo, la finalidad de ambos paquetes es la impresión de transparencias para ser usadas con retroproyectores, preferiblemente en formato `PostScript`, pero no la de creación de presentaciones multimedia. Para éstas han surgido en los últimos años complementos específicos que generan presentaciones multimedia con nuevas características adaptadas al entorno informático, tales como enlaces internos y externos, diferentes transiciones entre transparencias, etc. El archivo final tiene formato `PDF`, lo que además de permitir las opciones anteriores añade los beneficios del lenguaje `PDF`, similares a los del propio L<sup>A</sup>T<sub>E</sub>X.

A pesar de que los paquetes para la creación en formato `PDF` de presentaciones multimedia (`pdfscreen`, `pdfslide`, etc.) producen en general resultados satisfactorios, carecen de una importante característica que los sitúa en un nivel claramente inferior al de las utilidades más extendidas en este campo: la exhibición incremental de los elementos de la presentación, esto es, la posibilidad de mostrar el contenido de las transparencias de forma gradual mediante la adición de nuevo material a la parte ya expuesta. Esta desventaja comparativa se solventó con la aparición de `PPower4`, una utilidad gratuita escrita en Java que mediante un procesamiento del archivo `PDF` final permite insertar pausas (el paquete `pause`) en el desarrollo de la exposición. El sistema utilizado para simular la aparición gradual de nuevos elementos en la presentación consiste en crear una transparencia independiente con el material que aparece hasta cada una de las pausas. Posteriormente la exposición se realiza proyectando cada una de las transparencias consecutivamente, y puesto que cada una de ellas posee nuevo material que se añade al anterior, el efecto creado es el de que sólo existe una transparencia a la que se le van agregando diferentes

elementos. Como se comprobará más adelante, esta técnica es la misma que se aplica en el paquete **prosper**.

Aunque el avance cualitativo que significa la aparición de **PPower4** es considerable, existen algunas acciones que todavía no es posible realizar con este proceso, como la sustitución de elementos de una transparencia. Este aspecto, junto con el deseo de evitar el procesamiento ulterior del archivo PDF una vez generado mediante el software de creación de transparencias, constituye el origen de los paquetes **T<sub>E</sub>XPower** y **prosper**. El primero de ellos todavía se encuentra en una fase de desarrollo *pre-alfa* y, por el momento, incluye un post procesador similar a **PPower4** pero de mayor potencia, lo que sólo permite su utilización en combinación con algún paquete de creación de presentaciones multimedia, como los ya mencionados **pdfscreen**, **pdfslide**, etc. La clase de documentos **prosper** está algo más desarrollada (la versión actual es 1.0i) y constituye un paquete completo de generación de presentaciones dinámicas a través de L<sup>A</sup>T<sub>E</sub>X.

Tanto **prosper** como **T<sub>E</sub>XPower** proporcionan a los usuarios de L<sup>A</sup>T<sub>E</sub>X la oportunidad de producir presentaciones similares a las de PowerPoint, aunque quizás algo más sobrias. La situación actual permite avanzar que la aparición futura de nuevas versiones de **T<sub>E</sub>XPower** lo configurará como una referencia esencial en este campo, aunque por ahora es un paquete que, si bien resulta muy completo, posee numerosos comandos que complican su manejo habitual y que hace que el usuario prefiera utilizar **prosper**, un complemento potente y fácil de aprender.

### 3 La clase de documentos **prosper**

**Prosper** es un paquete para L<sup>A</sup>T<sub>E</sub>X desarrollado recientemente por Frédéric Goualard (véase [3]) para la realización e impresión de transparencias y la creación de presentaciones informáticas. Constituye una nueva clase de documentos para el editor de textos L<sup>A</sup>T<sub>E</sub>X pero respetando los comandos habituales de este lenguaje, de modo que cualquier texto creado con L<sup>A</sup>T<sub>E</sub>X puede ser incorporado directamente en una presentación creada con **prosper**, y por otro lado, cualquier usuario con una distribución de L<sup>A</sup>T<sub>E</sub>X instalada puede utilizar libremente el paquete **prosper**, por

ejemplo descargándolo de forma gratuita de [3].

**Prosper** es capaz de producir resultados en formato **PostScript** para la impresión de transparencias y en formato **PDF** para la realización de presentaciones multimedia. Esta doble posibilidad resulta interesante en determinadas ocasiones, por ejemplo cuando se desea entregar a la audiencia de una exposición multimedia una copia impresa de su contenido.

Es conveniente observar que para las exposiciones multimedia el formato del archivo final es **PDF**. Este hecho, no es casual puesto que este tipo de ficheros comparten gran parte de la filosofía del lenguaje **L<sup>A</sup>T<sub>E</sub>X**: gratuidad, portabilidad y bajo tamaño. En efecto, los archivos en formato **PDF** poseen unas dimensiones netamente inferiores al de otros formatos (lo que facilita su distribución a través de Internet o su almacenamiento en soportes físicos) y pueden ser leídos con una utilidad gratuita, **Acrobat Reader**, que ofrece la propia compañía en [6], y que posee versiones para todos los sistemas informáticos: Windows, Linux, Macintosh, etc. De este modo, los archivos de presentaciones generadas con **prosper** pueden ser ejecutados gratuitamente en cualquier sistema operativo, lo que, junto con el bajo tamaño de los ficheros, supone para estas presentaciones unas cotas de portabilidad difícilmente alcanzables por otros medios.

### 3.1 Opciones generales y estructura

Puesto que **prosper** compone una clase de documentos **L<sup>A</sup>T<sub>E</sub>X**, la primera línea del fichero fuente ha de ser `\documentclass[Opciones]{prosper}`, donde las opciones hacen referencia a configuraciones generales del documento. En la Figura 1 se presenta la estructura general de un documento fuente de **prosper** y las opciones disponibles para esta clase de documentos, entre las que cabe señalar la posibilidad de seleccionar **pdf** o **ps**, lo que determina, respectivamente, si el archivo se compila para producir una copia impresa (en formato **PostScript**) o para generar una presentación informática (en formato **PDF**). Para una descripción más extensa de las opciones y comandos del paquete **prosper** véase [2].

En el preámbulo, es decir, antes de la línea `\begin{document}` del documento



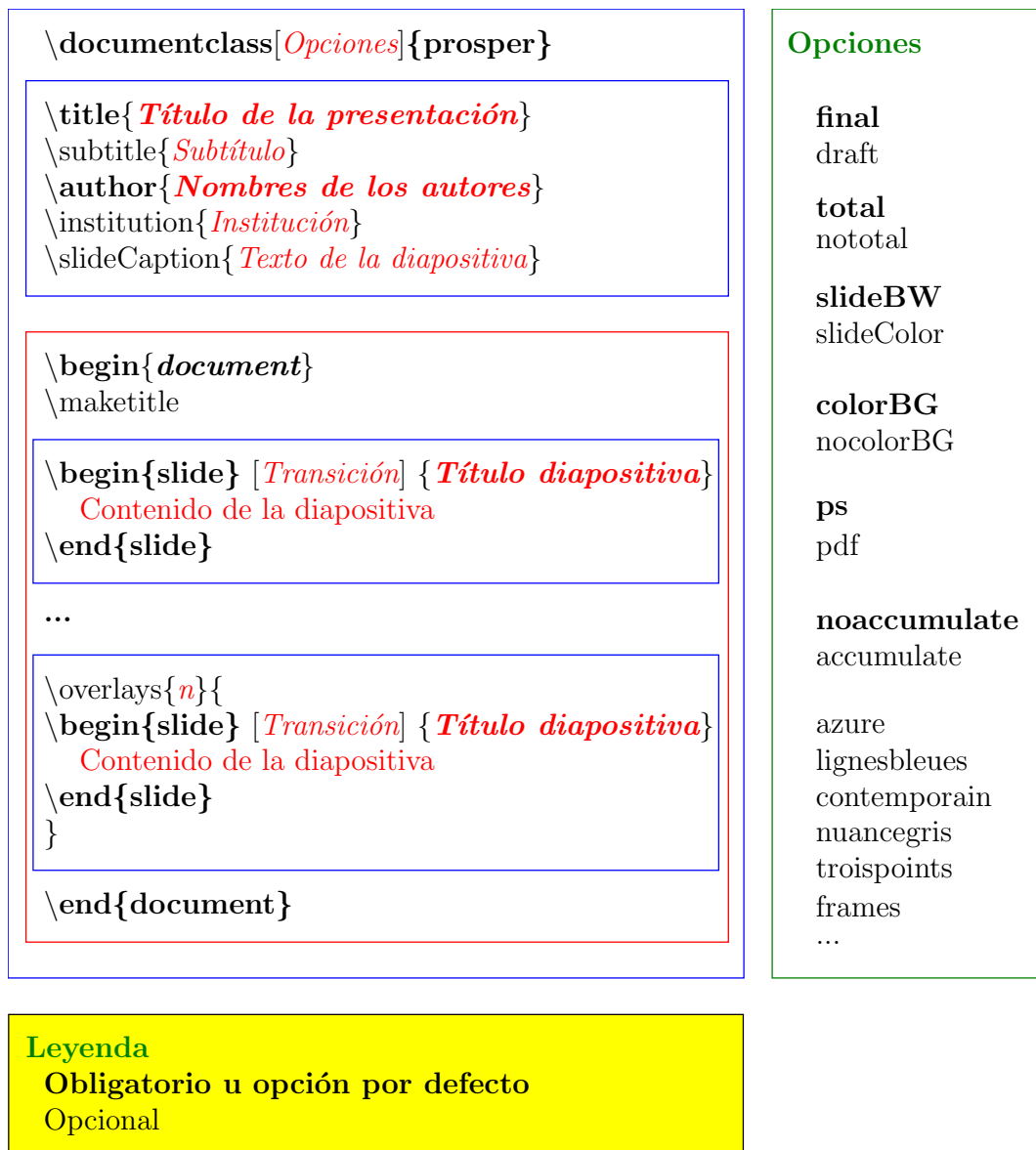


Figura 1: Estructura de comandos del paquete `prosper`

fuentes, se incluyen diversos comandos que afectan a la presentación, entre los que destacan los existentes para la composición de una página de título inicial mediante el comando `\maketitle` (que, como es usual, se sitúa en la línea inmediatamente posterior a `\begin{document}`):

`\title{Texto}`. El título de la presentación es *Texto*.

`\subtitle{Texto}`. El segundo título de la presentación es *Texto*.

`\author{Nombre}`. El nombre del autor/es de la presentación es *Nombre*.

`\email{Dirección}`. La dirección electrónica del autor/es es *Dirección*.

`\institution{Nombre}`. La institución a la que pertenece el autor/es es *Nombre*.

También es posible incluir en cada transparencia una pequeña imagen a modo de logotipo con el comando `\Logo`. Sin embargo con esta opción el tamaño del archivo resultante aumenta de forma notable, especialmente si se utiliza con frecuencia el comando `\overlays` (debido al sistema que emplea **prosper** para generar presentaciones incrementales de transparencias, descrito en la página siguiente), por lo que es recomendable sopesar previamente el interés de dicha opción y el efecto sobre la magnitud del fichero final.

## 3.2 Principales comandos

Una de las virtudes de **prosper** es la introducción de un pequeño número de comandos que generan una estructura de creación de transparencias clara y sencilla. El entorno en el que se introducen todos los elementos de cada transparencia es

```
\begin{slide}[Transición]{Título} ... \end{slide}
```

donde *Transición* es un argumento opcional que indica el tipo de transición<sup>1</sup> con el que aparecerán todos los elementos de la transparencia y *Título* es el título que se le asigna.

La exhibición incremental del material contenido en una diapositiva concreta se realiza cuando se incluye el entorno `slide` anterior dentro del comando

```
\overlays{n}{...}
```

donde *n* indica el número de partes en las que se dividirá la exposición de la diapositiva. Cuando aparece este comando, **prosper** transforma la transparencia creada con `slide` en *n* *subtransparencias*, todas ellas con el mismo título e igual número de página. En el archivo PDF final se muestran consecutivamente cada una de las

---

<sup>1</sup>Actualmente **prosper** soporta los tipos Replace, Split, Blinds, Box, Wipe, Dissolve y Glitter.

subtransparencias haciendo una pausa tras cada una de ellas. La existencia de contenidos en unas subtransparencias pero no en otras otorga al espectador la impresión de que se trata de una única transparencia a la que se añaden y quitan elementos. Pieza clave en este proceso es la determinación de qué contenidos aparecen en cada una de las subtransparencias, labor que se lleva a cabo con la ayuda de los comandos<sup>2</sup>

$$\backslash\text{fromSlide}^*\{p\}\{\dots\} \quad \backslash\text{untilSlide}^*\{p\}\{\dots\} \quad \backslash\text{onlySlide}^*\{p\}\{\dots\}$$

que indican, respectivamente, los elementos que aparecerán desde la subtransparencia  $p$  hasta la  $n$ , desde la subtransparencia 1 hasta la  $p$  y únicamente en la subtransparencia  $p$ . La utilización adecuada de estos comandos permite obtener toda clase de efectos relacionados con la adición y eliminación de material de una transparencia, incluso dentro de una fórmula matemática o una figura, lo que proporciona un completo abanico de posibilidades. Por ejemplo, si se desea que aparezca el objeto *Texto* en las subtransparencias 4, 5 y 8 de una diapositiva dividida en 10 subtransparencias, basta con reproducir el esquema de la Figura 2.

En determinadas circunstancias, no basta con la simple combinación de los comandos anteriores para obtener correctamente el efecto de insertar y eliminar contenidos de una diapositiva. En muchos de estos casos, el camino para conseguir los resultados esperados pasa por la utilización acertada del comando

$$\backslash\text{phantom}\{Argumento\}$$

de L<sup>A</sup>T<sub>E</sub>X. Esta orden se emplea para dejar espacios de separación adecuados en la página, ya que escribe *Argumento* con tinta invisible, como si de un fantasma (*phantom* en inglés) se tratara. En las Figuras 3 y 4 se muestra un sencillo ejemplo en el que se pone de manifiesto la necesidad de  $\backslash\text{phantom}$  para obtener un resultado satisfactorio. En la Figura 3 se comprueba cómo al escribir la expresión matemática para que sea presentada en dos pasos, y puesto que ésta es centrada por L<sup>A</sup>T<sub>E</sub>X, la segunda subtransparencia produce el efecto de desplazar la fórmula a la izquierda,

---

<sup>2</sup>Prosper posee comandos similares a los que se presentan sin el asterisco y con la primera letra en mayúscula y sin asterisco (véase [2]).

```

\overlays{10}{
\begin{slide}{Título}

...

\fromSlide*{4}{
    \untilSlide*{5}{Texto}
}

...

\onlySlide*{8}{Texto}

...

\end{slide}
}

```

Figura 2: Ejemplo de uso conjunto de los comandos `\fromSlide`, `\untilSlide` y `\onlySlide`.

ya que la primera parte de la expresión se escribe en cada una de las subtransparencias en posiciones horizontales distintas. La explicación de este *movimiento* es que la longitud de la fórmula es diferente en cada subtransparencia, por lo que al ser centrada por  $\text{\LaTeX}$  se produce un desajuste. Para salvar este problema es preciso construir en las dos subtransparencias fórmulas con la misma longitud, y dado que la segunda parte de la expresión no ha de ser exhibida hasta el paso final, el comando `\phantom` teniendo como argumento la última parte de la igualdad se revela como la mejor solución posible (véase Figura 4).

Uno de los aspectos más recurrentes en la mayoría de las exposiciones es la enumeración secuencial de distintos elementos, así que **prosper** posee un entorno especialmente creado para estos casos

```

\begin{itemstep}
\item Ítem 1

...

\item Ítem n
\end{itemstep}

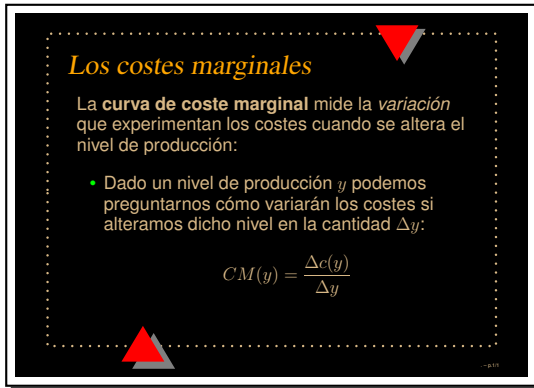
```

```

1 CM(y)=\frac{\Delta c(y)}{\Delta y}%
2 \fromSlide*{2}={\frac{c(y+\Delta y)-c(y)}{\Delta y}}

```

(a) Código fuente de la fórmula matemática



(b) Primera subtransparencia



(c) Segunda subtransparencia

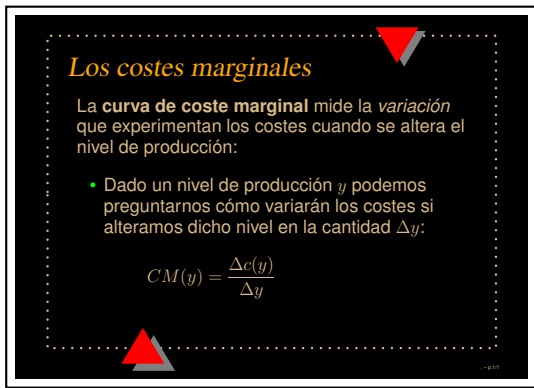
Figura 3: Presentación incremental sin el comando `\phantom`

```

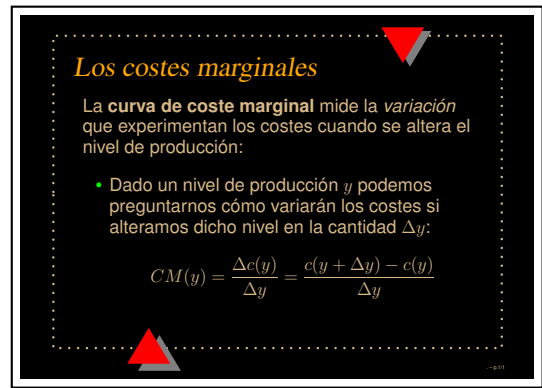
1 CM(y)=\frac{\Delta c(y)}{\Delta y}%
2 \onlySlide*{1}{%
3     \phantom{=\frac{c(y+\Delta y)-c(y)}{\Delta y}}%
4 }%
5 \fromSlide*{2}={\frac{c(y+\Delta y)-c(y)}{\Delta y}}

```

(a) Código fuente de la fórmula matemática



(b) Primera subtransparencia



(c) Segunda subtransparencia

Figura 4: Presentación incremental con el comando `\phantom`

que configura los ítem de modo que el primero de ellos aparezca en la primera subtransparencia, en la segunda subtransparencia se *añada* el segundo ítem, etc.

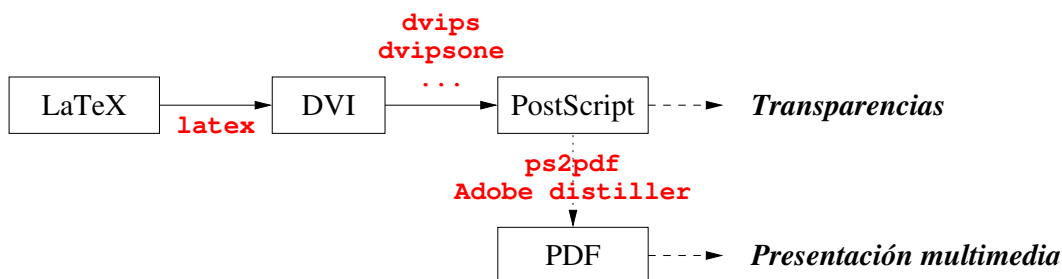


Figura 5: Esquema de compilación del paquete `prosper`

### 3.3 Obtención del fichero PDF

Al contrario de lo que suele suceder en  $\text{\LaTeX}$ , el proceso de compilación del archivo fuente hasta obtener el archivo PDF final no requiere de la utilidad  $\text{PDF}\text{\LaTeX}$ , sino del programa Ghostscript (se puede descargar gratuitamente de [7]) en una versión igual o superior a la 6.0. La razón de esto es que los fondos de las transparencias se crean con comandos incluidos en el programa `PSTricks`, el cual necesita que el fichero  $\text{\LaTeX}$  sea primero compilado y después convertido a `PostScript` con algún programa para tal fin, como por ejemplo `DVIPS`<sup>3</sup>, incluido en todas las distribuciones estándar de  $\text{\LaTeX}$ . Posteriormente el archivo `PostScript` es transformado en PDF mediante la utilidad `ps2pdf` incluida en Ghostscript.

En la Figura 5 se esquematizan los pasos a seguir para la obtención de los archivos en formato `PostScript` (para imprimir) y PDF (para presentaciones multimedia). Obsérvese que el último paso, el que proporciona el fichero PDF a partir del formato `PostScript`, puede llevarse a cabo también utilizando el software no gratuito Acrobat Distiller (véase [6]).

### 3.4 Estilos para prosper

Con el objetivo de obtener exposiciones de alta calidad gráfica, el paquete `prosper` incorpora el concepto de *estilo de transparencias*, conjunto de comandos para la construcción de cada página en los que se fijan los colores y tamaños de las fuentes, se

---

<sup>3</sup>Para conseguir que el archivo PDF final se genere con fuentes vectoriales, lo que lo convierte en independiente de la resolución, hay que incluir las opciones `-Pcmz` `-Pamz` al ejecutar `DVIPS`.

**Los costes marginales**

La **curva de coste marginal** mide la *variación* que experimentan los costes cuando se altera el nivel de producción:

- Dado un nivel de producción  $y$  podemos preguntarnos cómo variarán los costes si alteramos dicho nivel en la cantidad  $\Delta y$ :

$$CM(y) = \frac{\Delta c(y)}{\Delta y} = \frac{c(y + \Delta y) - c(y)}{\Delta y}$$

(a) Estilo *Alienglow*

**Los costes marginales**

La **curva de coste marginal** mide la *variación* que experimentan los costes cuando se altera el nivel de producción:

- Dado un nivel de producción  $y$  podemos preguntarnos cómo variarán los costes si alteramos dicho nivel en la cantidad  $\Delta y$ :

$$CM(y) = \frac{\Delta c(y)}{\Delta y} = \frac{c(y + \Delta y) - c(y)}{\Delta y}$$

(b) Estilo *Autumn*

**Los costes marginales**

La **curva de coste marginal** mide la *variación* que experimentan los costes cuando se altera el nivel de producción:

- Dado un nivel de producción  $y$  podemos preguntarnos cómo variarán los costes si alteramos dicho nivel en la cantidad  $\Delta y$ :

$$CM(y) = \frac{\Delta c(y)}{\Delta y} = \frac{c(y + \Delta y) - c(y)}{\Delta y}$$

(c) Estilo *Azure*

**Los costes marginales**

La **curva de coste marginal** mide la *variación* que experimentan los costes cuando se altera el nivel de producción:

- Dado un nivel de producción  $y$  podemos preguntarnos cómo variarán los costes si alteramos dicho nivel en la cantidad  $\Delta y$ :

$$CM(y) = \frac{\Delta c(y)}{\Delta y} = \frac{c(y + \Delta y) - c(y)}{\Delta y}$$

(d) Estilo *Contemporain*

**Los costes marginales**

La **curva de coste marginal** mide la *variación* que experimentan los costes cuando se altera el nivel de producción:

- Dado un nivel de producción  $y$  podemos preguntarnos cómo variarán los costes si alteramos dicho nivel en la cantidad  $\Delta y$ :

$$CM(y) = \frac{\Delta c(y)}{\Delta y} = \frac{c(y + \Delta y) - c(y)}{\Delta y}$$

(e) Estilo *Darkblue*

**Los costes marginales**

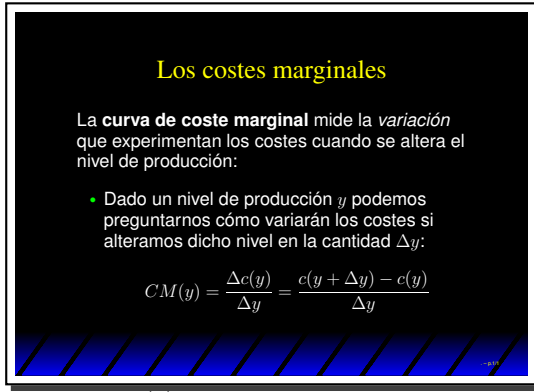
La **curva de coste marginal** mide la *variación* que experimentan los costes cuando se altera el nivel de producción:

- Dado un nivel de producción  $y$  podemos preguntarnos cómo variarán los costes si alteramos dicho nivel en la cantidad  $\Delta y$ :

$$CM(y) = \frac{\Delta c(y)}{\Delta y} = \frac{c(y + \Delta y) - c(y)}{\Delta y}$$

(f) Estilo *Frames*

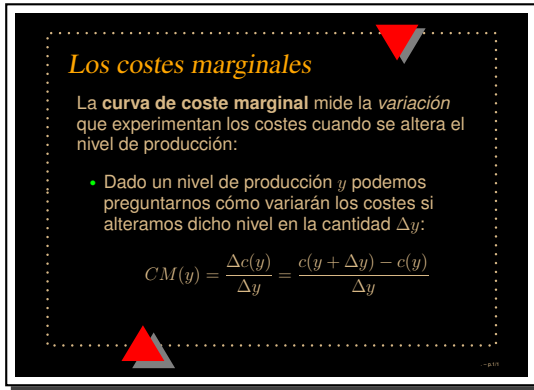
crea el esquema gráfico que es usado como fondo y se determinan otros aspectos de composición, como la ubicación del título de la transparencia. Cada estilo se almacena en un fichero independiente con nombre `PPRestilo.sty`, y su utilización en la creación de una presentación se realiza incluyendo la opción *estilo* dentro de la línea inicial del fichero fuente, es decir,



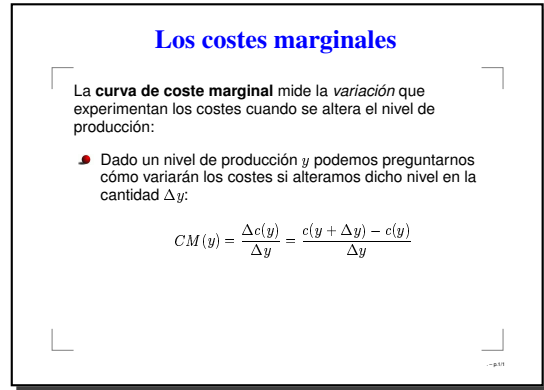
(g) Estilo *Lignesbleues*



(h) Estilo *NuanceGris*



(i) Estilo *TroisPoints*



(j) Ningún estilo seleccionado

Figura 6: Transparencias creadas con `prosper` y diferentes estilos

`\documentclass[Opciones,estilo]{prosper},`

utiliza como estilo de las transparencias *estilo*.

La distribución del paquete incluye algunos estilos creados por el propio autor (véanse imágenes (a)-(i) de la Figura 6) y otros fruto de las contribuciones de usuarios de `prosper`. Del mismo modo, cualquier usuario puede modificar los estilos ya creados o crear otros nuevos (véase [3] para un listado completo de estilos para `prosper`), siempre que tenga la precaución de utilizar nombres diferentes a los existentes, tal y como exige el autor en la documentación del programa (véase [2]). La realización de cambios en los estilos requiere la posesión de algunos conocimientos del paquete `PSTricks` y de programación en  $\text{\LaTeX}$ , para lo cual pueden consultarse los trabajos [1] y [5] respectivamente. En cualquier caso, esta labor queda simplificada enormemente gracias a la forma intuitiva y autoexplicativa con la que se suelen



construir los comandos de  $\text{\LaTeX}$ , lo que permite la ejecución de pequeñas modificaciones mediante una observación cuidadosa del fichero de estilo y de los resultados que produce.

## 4 Incorporación de gráficos e imágenes

La inserción de imágenes externas en archivos  $\text{\LaTeX}$  ha sido, desde hace mucho tiempo, uno de los principales lastres del editor  $\text{\LaTeX}$ . En realidad  $\text{\LaTeX}$  no es capaz de incluir imágenes, sino que esta labor queda postergada a la acción del *controlador* (generalmente *DVIPS*) que es el encargado de visualizar y/o imprimir el archivo gráfico. Los paquetes **graphics** y **graphicx** fueron desarrollados para simplificar estas tareas al usuario, básicamente incluyendo el comando

$$\backslash\text{includegraphics}[Opciones]\{Archivo\}$$

para insertar el fichero gráfico *Archivo* en  $\text{\LaTeX}$ . Dado que las presentaciones creadas con **prosper** han de ser transformadas a formato **PostScript**, normalmente con *DVIPS*, los archivos gráficos que pueden incluirse en la presentación han de tener formato **PostScript**, o **EPS** (**PostScript** encapsulado).

Si la adición de imágenes en **prosper** puede ser en ciertos casos algo tediosa, la creación de dibujos y figuras supone un aspecto favorable para la utilización de  $\text{\LaTeX}$  en general, y de **prosper** en particular. El paquete **PSTricks** (de libre distribución, por ejemplo en [8]) proporciona una amplia gama de comandos compatibles con  $\text{\LaTeX}$  que, mediante directivas del lenguaje **PostScript**, crean objetos gráficos insertados en el fichero  $\text{\LaTeX}$  cuando éste se transforma al formato **PostScript**. Frente al inconveniente de construir el objeto gráfico mediante comandos y no con la ayuda del ratón como es usual en PowerPoint, el paquete **PSTricks** presenta numerosas ventajas al usuario de  $\text{\LaTeX}$  y **prosper**, entre las que destacan las siguientes:

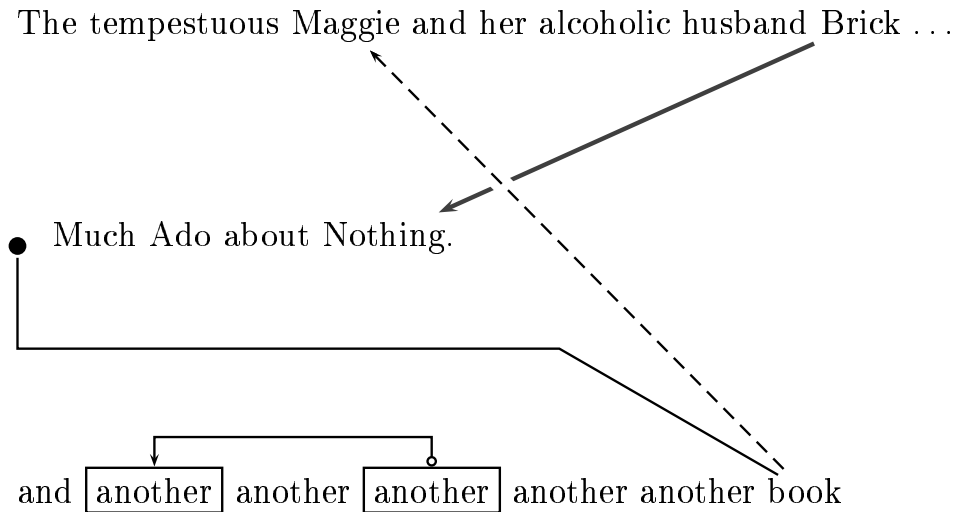
- La utilización de comandos para la configuración de objetos gráficos implica que el incremento de tamaño que éstos suponen para el archivo fuente, el **PostScript** o el **PDF** es mínimo, incrementando de este modo la portabilidad de las presentaciones de **prosper**.

```

1 The tempestuous \rnode{E}{Maggie} and her alcoholic husband
2 \rnode{A}{Brick}\ldots\vskip 2cm
3 \cnode*{3pt}{D}\hskip 1em Much Ado about \rnode{B}{Nothing}.\vskip 1in
4 and \rnode{J}{\psframebox{another}} another
5 \rnode{K}{\psframebox{another}} another another \rnode{C}{book}
6 \ncline[linecolor=darkgray,linewidth=1.5pt,nodesep=3pt]{->}{A}{B}
7 \ncline[nodesep=3pt,linestyle=dashed,border=3pt]{->}{C}{E}
8 \ncangle[nodesep=1pt,angleA=-90,angleB=150,armB=3cm]{D}{C}
9 \ncbar[angle=90]{<-oo}{J}{K}

```

(a) Código fuente



(b) Resultado

Figura 7: Combinación de texto y objetos gráficos con PSTricks

- Los comandos gráficos y el texto del documento se encuentran al mismo nivel, lo que añade un gran abanico de posibilidades para las presentaciones multimedia con muy poco esfuerzo (véase la combinación de texto y objetos gráficos en Figura 7).
- Los gráficos producidos son vectoriales, y por tanto, totalmente escalables y de la máxima calidad, tanto al visualizarlos como al imprimirlos.
- Existen gran diversidad de conjuntos de comandos o *subpaquetes* de PSTricks para facilitar la realización de las tareas más diversas, como nodos, estructuras en árbol, gráficos en 3 dimensiones, etc. Como ilustración de esta enorme diversidad de subpaquetes se presenta una imagen (véase Fig. 8) de un ca-

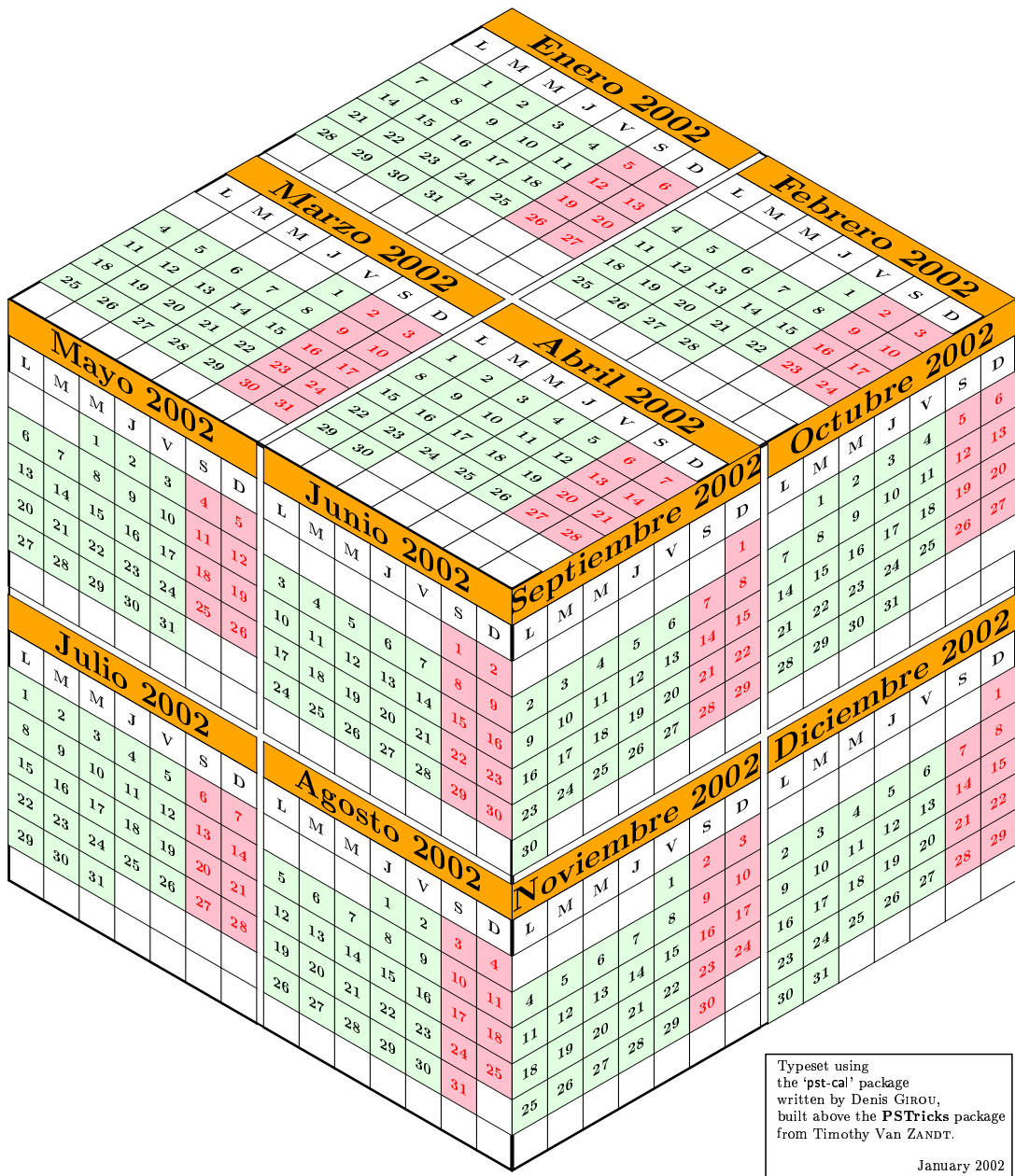


Figura 8: Calendario en tres dimensiones con el paquete `pst-cal`.

lendarario tridimensional para el presente año realizado sobre PSTricks con el paquete `pst-cal`, que cuando se distribuya próximamente permitirá la creación automática de calendarios de este tipo para el año que se desee.

## 5 Conclusiones

En el trabajo se ha puesto de manifiesto que, pese a la masiva utilización en la actualidad de Microsoft PowerPoint para la realización de presentaciones multimedia, existen importantes inconvenientes para su utilización, como su alto precio, las notables necesidades de hardware, el gran tamaño de los archivos resultantes o la incompatibilidad con numerosos sistemas informáticos, e incluso con otras versiones del software. Estos aspectos negativos cobran especial importancia en ciertas situaciones como en el caso de exposiciones con abundantes fórmulas matemáticas o presentaciones que necesitan de un alto grado de portabilidad, por ejemplo, cuando son compartidas a través de Internet para su descarga por cualquier usuario.

Como alternativa a Microsoft PowerPoint se ha presentado el paquete **prosper** para el editor  $\text{\LaTeX}$ , cuya difusión se ha visto limitada entre el gran público debido a que también presenta algunos inconvenientes. El más importante de ellos es la dificultad de aprendizaje entre usuarios ajenos a  $\text{\LaTeX}$ , ya que su uso se limita, al igual que el del lenguaje  $\text{\LaTeX}$ , a líneas de comandos. Este hecho contrasta con la fácil e intuitiva utilización de Microsoft PowerPoint por parte de nuevos usuarios, basada en el uso combinado de barras de herramientas y ratón, y sustentada en el estilo homogéneo de la suite Microsoft Office, conocida por la mayor parte de los usuarios informáticos.

**Prosper** es una utilidad gratuita diseñada para la creación de presentaciones multimedia en formato PDF con una alta calidad gráfica. Tanto el hecho de que el archivo final tenga formato PDF como que el software esté basado en el lenguaje  $\text{\LaTeX}$  aseguran que los resultados obtenidos con **prosper** resuelvan los problemas de PowerPoint mencionados anteriormente. Aunque la inserción de archivos de imágenes creadas con otras aplicaciones se restringe únicamente a los formatos **PostScript** y **EPS**, el uso de este programa en combinación con el paquete **PSTricks** para la creación de objetos y figuras permite obtener resultados gráficos superiores a los de PowerPoint.

# Referencias

- [1] B. Cascales Salinas, P. Lucas Saorín, J.M. Mira Ros, A. Pallarés Ruiz, S. Sánchez-Pedreño Guillén (2000). *ΛT<sub>E</sub>X: una imprenta en sus manos*. Aula Documental de Investigación, Madrid.
- [2] F. Goualard (2000). *Manual for the prosper class*. Documentación distribuida con el paquete.
- [3] F. Goualard. *Prosper @ Source Forge (tm)*. <http://prosper.sourceforge.net/>.
- [4] L. Lamport (1994). *ΛT<sub>E</sub>X – A document Preparation System. User’s guide and manual references*. Addison-Wesley, 2<sup>a</sup> edición, 1994.
- [5] T. Van Zandt (1993). *PSTricks: PostScript macros for Generic T<sub>E</sub>X. User’s Guide*. Documentación distribuida con el paquete.
- [6] *Adobe Systems Incorporated*. <http://www.adobe.com>.
- [7] *Ghostscript, Ghostview and GSview*. <http://www.cs.wisc.edu/~ghost/>.
- [8] *Welcome to CTAN: the Comprehensive TeX Archive Network*. <http://www.ctan.org/>.