

Gráficos y color en L^AT_EX

J. M. López y L. M. Nieto

17 de noviembre de 2004

Resumen

En este breve artículo pretendemos condensar los aspectos más importantes de la inclusión de gráficos en L^AT_EX y el uso del color.

Índice

1. Para los más impacientes	2
2. Un poco de historia	3
2.1. Formatos de gráficos	3
2.2. ¿Qué tipo de formato uso?	3
2.3. ¿Cuales son los mejores formatos para incluir en L ^A T _E X?	4
3. Inclusión de gráficos EPS	4
3.1. Ajustando el tamaño	7
3.2. Rotando figuras	8
4. El entorno figure	9
5. Personalización del entorno figure	10
6. Emplazamiento de las figuras	12
6.1. Fijando figuras	14
7. Más de un gráfico en una figura	14
8. El entorno subfigure	14
9. Recortando figuras	16
10. Personalización espacios	17
11. Figuras en otros formatos	18

12. Rotando y escalando objetos	18
13. Color. Introducción básica	21
14. Modelos de color	21
15. Definiendo colores	22
16. Usando colores	22
16.1. Colores predefinidos	22
16.2. Usando las especificaciones de color directamente	22
17. Páginas en color	23
18. Cajas coloreadas	23
19. Nombre de los colores por nombre en dvips	25
20. Definición de los colores de la página anterior	26
21. Trucos en windows para obtener ficheros postscript	27
21.1. Obteniendo ficheros postscript	28
21.2. El fichero tiene mucho “blanco” alrededor	28

Índice de figuras

1. Esta es la representación de la superficie...	10
2. Esta es la representación de la superficie...	11
3. Esta es la representación de la superficie...	12
4. Esta es la representación de la superficie...	12
5. Esta es la representación de la superficie...	13
6. Esta es la representación de la superficie...	13
7. Esta es una figura con un número único y dos gráficas	14
8. Pie de la figura total	15
9. Esta es la representación de la superficie silla mono recortada	16
10. Silla mono recortada automáticamente con gsvievw	17

1. Para los más impacientes

Los gráficos en formato postscript encapsulado se pueden incluir en un texto escrito en \LaTeX incluyendo en el preámbulo `\usepackage{graphicx}` y utilizando el comando `\includegraphics{fichero.eps}` en el lugar donde queremos que aparezca el gráfico. Este gráfico aparece en su tamaño natural, ajustes de tamaño, rotaciones, inclusión de pie

de figura, numeración, referencias cruzadas a la figura, etc son un poco más complicadas de generar y no se puede resumir en unas líneas.

2. Un poco de historia

Cuando $\text{T}_{\text{E}}\text{X}$ se escribió no existía postscript, EPS, JPEF, GIF, TIFF, etc. eso significa que originalmente la inclusión de gráficos no se contempló, pero Knuth dejó una puerta abierta para incorporar a $\text{T}_{\text{E}}\text{X}$ ficheros no ASCII, el comando `\special` en el que se basan los paquetes de inclusión de gráficos, que permiten que $\text{T}_{\text{E}}\text{X}$ componga la página sin necesidad de conocer el contenido de figura, necesita solamente conocer su tamaño para reservar el espacio necesario en la composición de la página, luego el programa de impresión se encargará de “traducir” el gráfico a lenguaje de impresora para situarlo sobre el papel.

Para los que empiezan con $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ la inclusión de gráficos en un documento suele ser uno de los aspectos que más problemas generan, la razón primordial es la gran diversidad de formatos e incompatibilidades que existen entre ellos.

2.1. Formatos de gráficos

Esencialmente existen dos formas diferentes de almacenar en un fichero un gráfico: **Mapas de bits** y **Formato vectorizado**, el primero consiste en una gran tabla en la que se informa sobre el color de cada uno de los pixels (puntos) del gráfico, cuando se quiere visualizar simplemente se copia la tabla de colores en una pantalla (o impresora). Este tipo de formato es el que usan las cámaras fotográficas. Tiene el inconveniente que si queremos ampliar el gráfico esos puntos (que originalmente son del tamaño del punto más pequeño que se puede mostrar en la pantalla) acaban transformándose en cuadros de color apreciables a simple vista y por tanto la imagen pierde calidad. Las extensiones más usuales de gráficos de este tipo son `bmp`, `jpg` (`jpeg`), `gif`, `png` y `tiff` y muchos programas generan este tipo de archivos.

En el formato vectorizado en lugar de guardar la información como un mapa de colores se almacenan una serie de instrucciones que permiten al ordenador regenerar el gráfico cada vez que se quiere mostrar en pantalla (o enviar a una impresora), de esa forma cuando se amplía el tamaño el gráfico se escala adecuadamente manteniendo la calidad original. Formatos de este tipo son `wmf`, `ps`, `eps`, `pdf`. Los formatos vectorizados pueden guardar, como parte del gráfico, mapas de bits, por tanto en ese caso esa parte del gráfico sufrirá los mismos problemas comentados anteriormente.

2.2. ¿Qué tipo de formato uso?

Una regla sencilla sobre que formato utilizar puede ser la siguiente:

fotos e imágenes con muchos colores `jpg` o `jpeg` (son el mismo)

mapas de bits con pocos colores `bmp`, `gif`, `png`

gráficos de funciones, imágenes con líneas, etc formatos vectoriales, `ps`, `eps`, `pdf`

2.3. ¿Cuales son los mejores formatos para incluir en L^AT_EX?

Los gráficos más fáciles de incluir en un documento L^AT_EX son los gráficos en postscript encapsulado (eps), por esa razón empezaremos por este tipo de gráficos ¹

3. Inclusión de gráficos EPS

Un documento mínimo para inclusión del gráfico `grafico.eps` es el siguiente

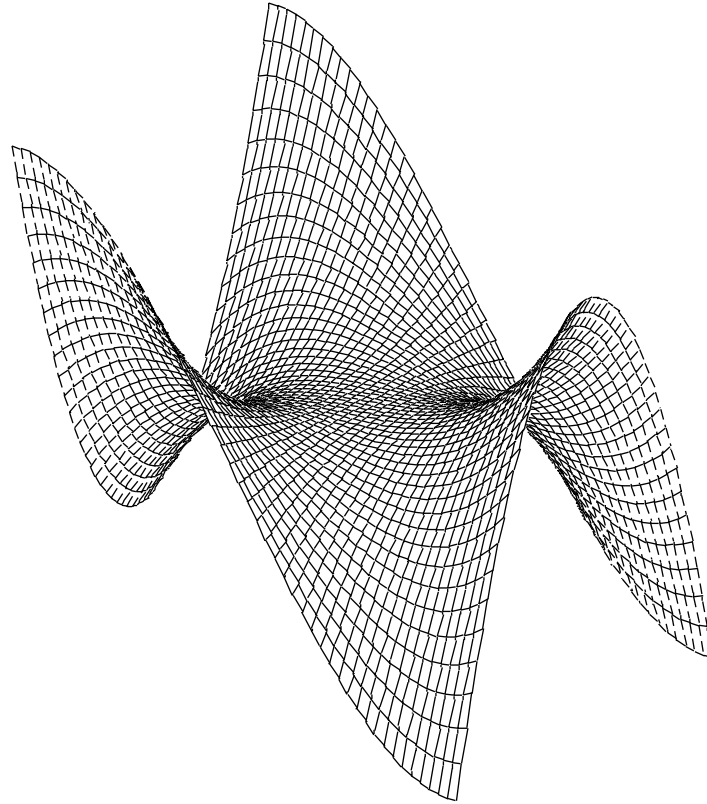
```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
  \includegraphics{grafico.eps}
\end{document}
```

Por ejemplo:

El comando `\includegraphics{sillamono.ps}` inserta la figura que vemos a continuación.

¹Esto no es una gran restricción, muchos programas de generación de gráficos exportan en ese formato, si esa posibilidad no existe siempre se puede instalar una impresora postscript e imprimir con esa impresora el gráfico en un archivo, que por tanto estará en postscript. Para mas detalles veáse la sección [21](#)

Silla de mono



Como puede verse la incluye en su tamaño natural y por tanto tiene que poner la imagen en la siguiente página puesto que en la anterior no tiene espacio, la imagen ocupa la página completa y se desborda!!!

Existen varias opciones en el comando `\includegraphics` que permiten controlar varios aspectos de la inclusión, el comando, de forma general, puede expresarse:

```
\includegraphics[opción1,opcion2,...]{grafica.eps}
```

Algunas de las posibles opciones son:

- height
- width
- angle
- origin
- viewport
- clip
- draft
- final
- etc

El fichero `sillamono.ps` contiene una línea:

```
%%BoundingBox: 50 50 554 770
```

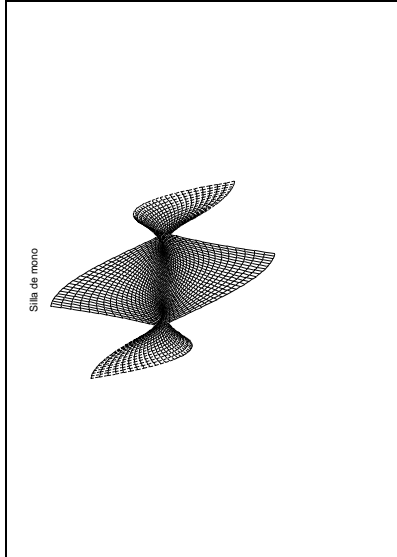
Esta línea es la única que \LaTeX lee del fichero gráfico, en ella se especifica el tamaño del gráfico. Los cuatro números indican las coordenadas $x=50$ e $y=50$ del punto inferior izquierdo de la figura y las coordenadas $x=554$ e $y=770$ del punto superior derecho del gráfico en unidades de longitud postscript que corresponde a un valor de $\frac{1}{72}$ de pulgada (puntos pp)², por tanto el tamaño de la gráfica es de 7×10 pulgadas. Los números del BoundingBox indican a la impresora donde debe situar el rectángulo cubierto por la gráfica, tomando como origen, $(0,0)$, las coordenadas de la esquina inferior izquierda del papel en posición vertical, es decir, con el lado mayor como eje y y el lado menor como eje x .

² Los puntos manejados por \TeX son ligeramente diferentes $\frac{1}{72,27}$ de pulgada (estos puntos se denominan pt)

3.1. Ajustando el tamaño

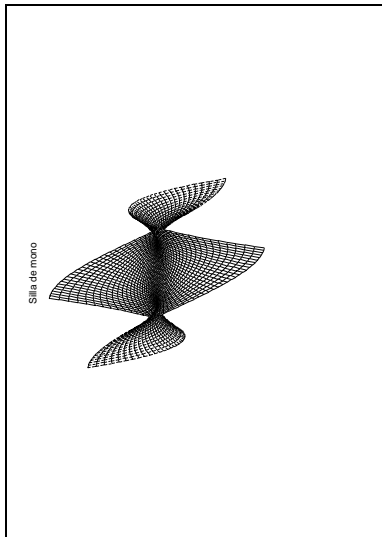
Podemos ajustar el tamaño de la gráfica imponiendo una determinada anchura a la gráfica (la longitud se ajusta automáticamente)

```
\includegraphics[width=5cm]{sillamono.ps}
```



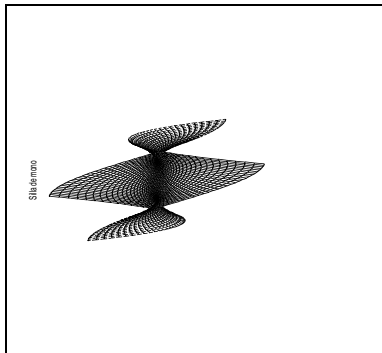
Las medidas absolutas son un problema para la portabilidad, podemos poner una medida relativa al ancho del texto (la anchura del texto está determinado por `\textwidth`)

```
\includegraphics[width=0.3\textwidth]{sillamono.ps}
```



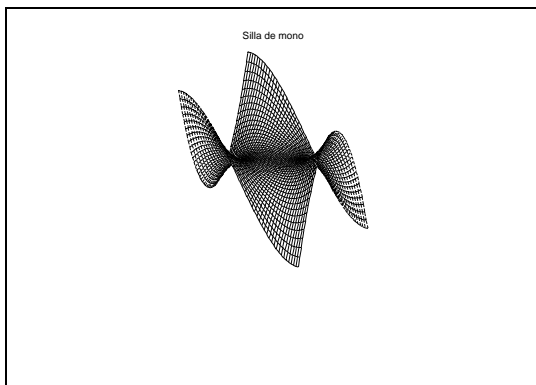
Si especificamos la longitud y anchura la gráfica se deforma

```
\includegraphics[width=0.3\textwidth,  
height=0.2\textheight]{sillamono.ps}
```



3.2. Rotando figuras

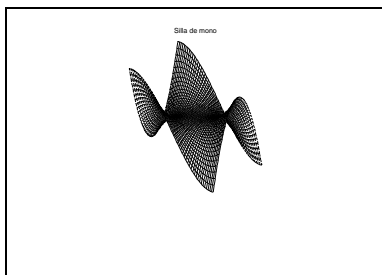
Se puede rotar una figura un ángulo arbitrario alrededor de un punto dado (por defecto se rota alrededor del punto inferior izquierdo del gráfico)



la figura anterior se ha generado con:

```
\includegraphics[width=0.3\textwidth,angle=-90]{sillamono.ps}
```

El orden en que se ponen las opciones es importante:



```
\includegraphics[angle=-90,width=0.3\textwidth]{sillamono.ps}
```

4. El entorno figure

Como hemos podido apreciar en lo visto anteriormente, para \LaTeX una figura no es más que una caja de unas dimensiones que tiene que componer dentro de una página, dadas las dimensiones de esta “caja” la composición de la página es realmente difícil y poco estética, cuando una página está a medio llenar de texto y debemos incluir una figura, si ésta es más grande que el espacio de la página \TeX se ve obligado a pasarla a la página siguiente dejando una página incompleta. Para prevenir este comportamiento \LaTeX dispone de un entorno que permite “flotar” a la figura hasta encontrar el lugar adecuado para situarse sin dejar páginas a medio componer y siguiendo unas reglas de enmaquetado de páginas bastante estrictas que permiten un acabado muy profesional de los documentos.

Este proceso en que las figuras “flotan” sobre el texto y se sitúan en el lugar adecuado para que la página tenga un aspecto profesional es aceptado con dificultad por algunos, que se sienten más cómodos situando las figuras justo donde se ponen en el fichero fuente, sin embargo fijar de antemano el lugar donde queremos colocar la figura obliga en ocasiones a tener que cambiar el texto del documento para ajustarlo al espacio que la figura deja libre e impide hacer modificaciones posteriores del texto sin un tedioso proceso de recolocado de todas las figuras. Para evitar esto debemos tener en cuenta solamente dos principios:

- No debemos escribir documentos que sean dependientes de la posición de los gráficos, por tanto debemos evitar frases como “Esta figura...” o “La siguiente figura...” que debemos sustituir por “En la figura 12...”, que permite situar la figura en cualquier punto del documento.
- Tranquilidad sobre la posición de los gráficos, ese es el trabajo encomendado a \LaTeX y lo hace muy bien (él es nuestro tipista)

El entorno figure:

```
\begin{figure}
\includegraphics[opciones]{grafico.eps} \caption{Esta figura
representa...} \label{etiqueta-fig}
\end{figure}
```

Trata a esta figura como un cuerpo flotante que se acomoda en la posición de la página adecuada y numera la figura para poder luego referirse a ella por su etiqueta:

Ahora podemos referirnos a esta figura simplemente diciendo una superficie curiosa puede verse en la figura 1 en la página 10

Ahora podemos referirnos a esta figura simplemente diciendo una superficie curiosa puede verse en la figura~\ref{sillamono} en la página~\pageref{sillamono}

Las reglas “profesionales” de enmaquetado de figuras son bastante estrictas:

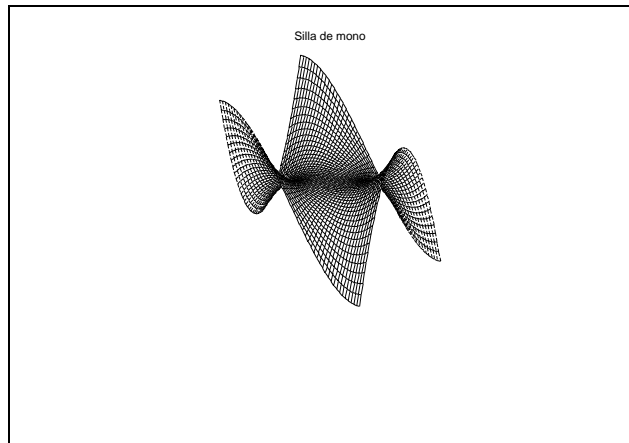


Figura 1: Esta es la representación de la superficie...

- Una figura no puede cubrir más del 80 % de una página que tenga texto
- La fracción que una figura puede ocupar, cuando se sitúa en la parte superior de una página que contenga texto es de 0.7 de la longitud total del texto en una página normal.
- Si se pone más de una figura en la parte superior de una página con texto la fracción máxima es del 0.6
- La máxima fracción de texto ocupada por una figura situada en el fondo de una página es de 0.4 de la altura total de texto.
- La mínima superficie que tiene que ser ocupada por las figuras en una página de solo figuras es del 50 %

Todos estas fracciones pueden cambiarse pero si no es estrictamente necesario es mejor no hacerlo.

5. Personalización del entorno figure

Las figuras que aparecen anteriormente aparecen enmarcadas en un rectángulo, esa es una opción que se consigue simplemente con un comando `\fbox{}` antes de incluir la figura, por ejemplo la figura 1 se ha obtenido con el código siguiente:

```
\begin{figure}
  \centering
  \fbox{
\includegraphics[angle=-90,width=0.5\textwidth]{sillamono.ps}
  }
  \caption{Esta es la representación de la
superficie...}\label{sillamono}
```

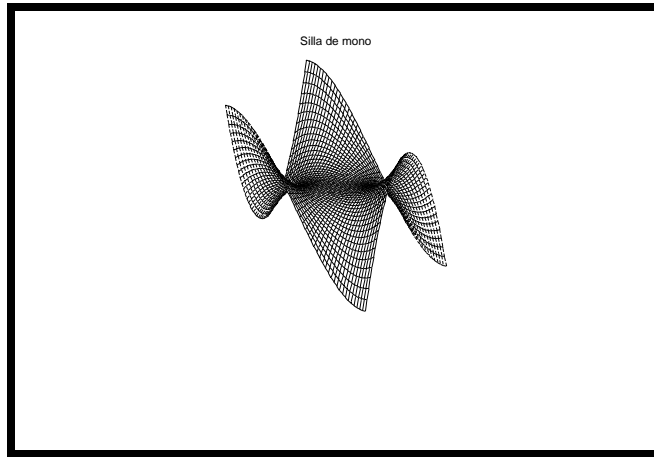


Figura 2: Esta es la representación de la superficie...

Los recuadros pueden personalizarse, el grosor de la línea se controla con

```
\setlength{\fboxrule}{n pt}
```

donde n es un número, como puede apreciarse en la figura 2, obtenida a partir de:

```
\begin{figure}
  \centering
  \setlength{\fboxrule}{3 pt}
  \fbox{
\includegraphics[angle=-90,width=0.5\textwidth]{sillamono.ps}
  }
  \caption{Esta es la representación de la
superficie...}\label{sillamonobis}
\end{figure}
```

La separación entre la línea del cuadro y la figura también se puede controlar con el comando `\setlength{\fboxsep}{n cm}` siendo n un número, como puede apreciarse en la figura 3, obtenida a partir de:

```
\begin{figure}
  \centering
  \setlength{\fboxrule}{3 pt}
  \setlength{\fboxsep}{1 cm}
  \fbox{
\includegraphics[angle=-90,width=0.5\textwidth]{sillamono.ps}
  }
  \caption{Esta es la representación de la
superficie...}\label{sillamonobisbis}
\end{figure}
```

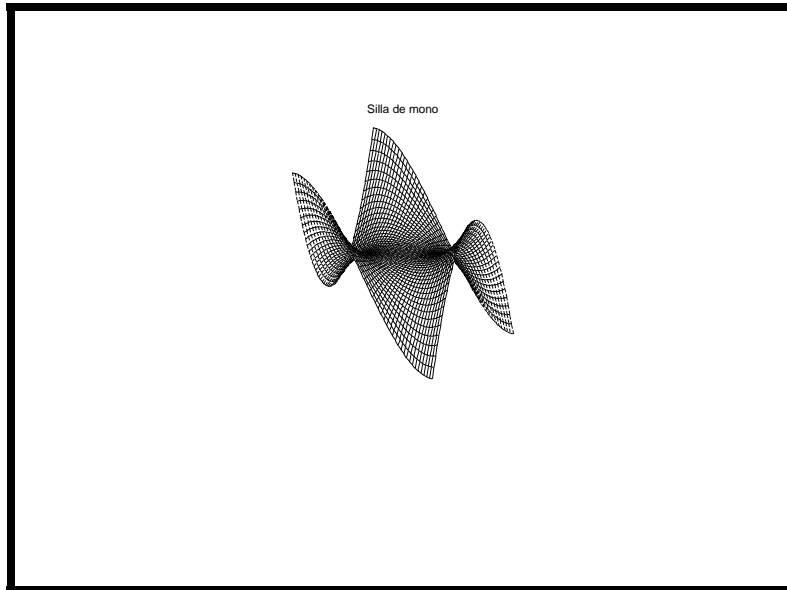


Figura 3: Esta es la representación de la superficie...

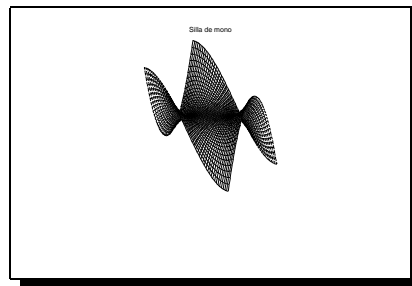


Figura 4: Esta es la representación de la superficie...

Se pueden conseguir personalizaciones más completas con el paquete `\usepackage{fancybox}` como puede verse en la figura 4 (`\shadowbox{ }`), en la figura 5 (`\doublebox{ }`) y en la figura 6 (`\ovalbox{ }`):

6. Emplazamiento de las figuras

El entorno `figure` tiene un argumento opcional que permite a los usuarios especificar emplazamientos posibles por orden de preferencia. Este argumento opcional puede tomar los valores:

- `h` (here) se le pide a `TEX` que trate de colocar la figura en el lugar del texto donde se incluye
- `t` (top) sitúese la figura al principio de la página
- `b` (bottom) sitúese la figura en el fondo de la página

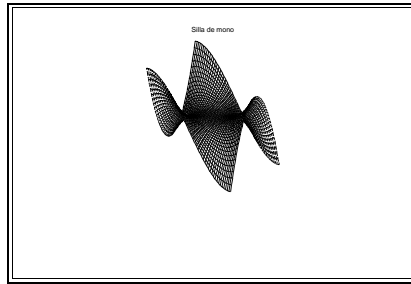


Figura 5: Esta es la representación de la superficie...

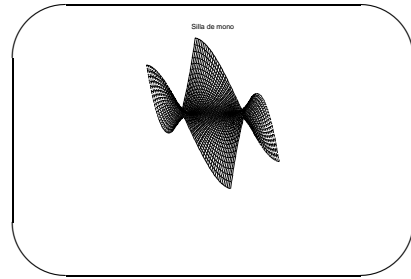


Figura 6: Esta es la representación de la superficie...

- p sitúese la figura en una página de objetos flotantes (figuras, tablas, etc)

Si no se especifica este argumento opcional la posición por defecto es equivalente a `\begin{figure}[tbp]`

El orden en que se escriben los emplazamientos no supone diferencia, el orden es siempre `h t b p`

Cuando un documento tiene poco texto y muchas gráficas puede resultar difícil encontrar el sitio adecuado para situar las figuras por lo que podemos encontrarnos con muchas gráficas no procesadas esperando turno para encontrar un hueco (una gráfica no se procesa hasta que todas las gráficas que aparecieron anteriormente en el texto lo están), esta situación puede acumularse hasta el final del documento, con lo cual todas las figuras aparecerán al final del mismo. Para evitar que figuras no procesadas correspondientes a una parte del documento no se acomoden en la parte siguiente se puede poner “barreras”, para ello se puede usar el paquete `placeins`:

```
\usepackage[section]{placeins}
```

que pone como barrera para el procesamiento de las figuras el comienzo de la sección siguiente.

`\usepackage[below]{placeins}` que permite situar figuras hasta (incluida) la página donde comienza una nueva sección si en la misma aparece algo de texto de la sección precedente.

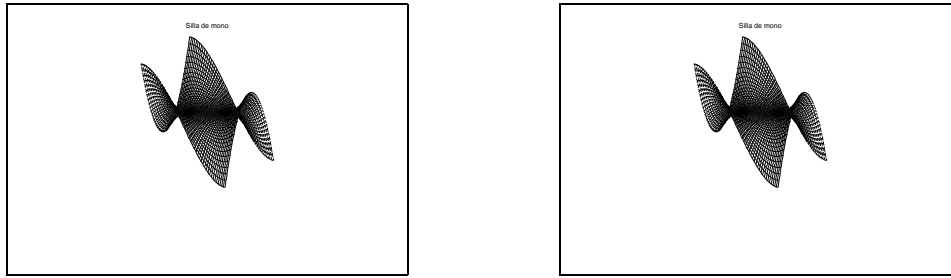


Figura 7: Esta es una figura con un número único y dos gráficas

6.1. Situando figuras en posiciones determinadas del texto

En ocasiones es necesario situar una figura en un punto determinado del texto de forma incondicional, si tenemos esa necesidad podemos seguir manteniendo el mismo tipo de entorno pero evitando que “flote”. Para lograrlo tenemos que cargar el paquete `float` después de cargado cuando abrimos el entorno `figure`, dentro de las opciones de posición utilizamos `H` que significa que la figura se sitúe en esa posición exactamente. Los problemas derivados de esta obligatoriedad son evidentes, si la página de texto está incompleta y el espacio que que no es suficiente para situar la figura, ésta pasará a la página siguiente dejando una página a medio componer, etc.

7. Más de un gráfico en una figura

En ocasiones es muy útil poner más de un gráfico con un pie común. Como ejemplo veamos la figura 7, que se ha obtenido a partir de:

```
\begin{figure}
  \centering
  \fbox{
\includegraphics[angle=-90,width=0.3\textwidth]{sillamono.ps}
    }%
    \hspace{2cm}%
  \fbox{
\includegraphics[angle=-90,width=0.3\textwidth]{sillamono.ps}
    }
\caption{Esta es una figura con un número único y dos gráficas}
  \label{dosfiguras}
\end{figure}
```

8. El entorno subfigure

Dos gráficas juntas con un mismo pie común pueden representar resultados a los que debemos referirnos en el texto conjuntamente o por separado, para ello debemos numerar

las figuras con un número común y una letra para diferenciar una subfigura de la otra; esto se puede conseguir utilizando el paquete subfigure, para obtener algo como lo que se puede apreciar en la figura 8. Ahora podemos referirnos a la figura completa (figura 8) o cada una de las subfiguras por separado, en la figura 8(a) o en la figura 8(b)

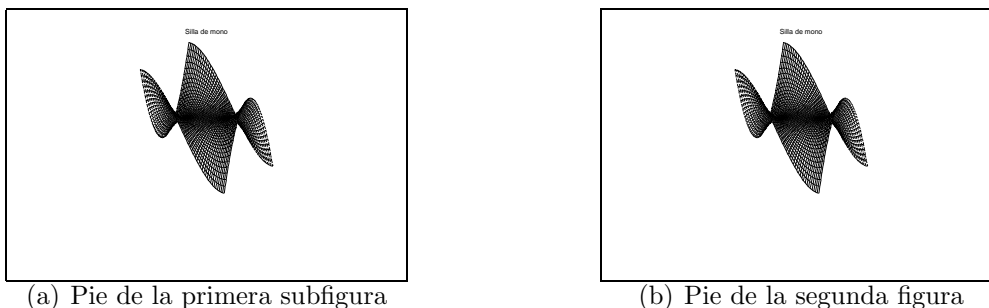


Figura 8: Pie de la figura total

La figura 8 se ha obtenido a partir de:

```
\begin{figure}[h]
  \centering
  \subfigure[Pie de la primera subfigura]{\label{subfiguraa}
    \fbox{
      \includegraphics[angle=-90,width=0.3\textwidth]{sillamono.ps}
    }
  }

  \hspace{2cm}
  \subfigure[Pie de la segunda figura]{\label{subfigurab}
    \fbox{
      \includegraphics[angle=-90,width=0.3\textwidth]{sillamono.ps}
    }
  }

  \caption{Pie de la figura total}\label{subfiguras}
\end{figure}
```

El texto en que se hace referencia a la figura y subfiguras se ha generado a partir de:

Dos gráficas juntas con un mismo pie común pueden representar resultados a los que debemos referirnos en el texto conjuntamente o por separado, para ello debemos numerar las figuras con un número común y una letra para diferenciar una subfigura de la otra; esto se puede conseguir utilizando el paquete subfigure, para obtener algo como lo que se puede apreciar en la figura~\ref{subfiguras}. Ahora podemos referirnos a la figura completa (figura~\ref{subfiguras}) o cada una de las subfiguras por separado, en la figura~\ref{subfiguraa} o en la figura~\ref{subfigurab}

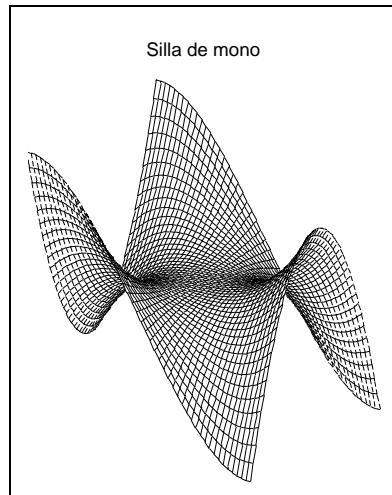


Figura 9: Esta es la representación de la superficie silla mono recortada

9. Recortando figuras

Como ha podido apreciarse a lo largo de este documento la figura usada está “rodeada” de mucho espacio en blanco, esto sucede con frecuencia puesto que muchos programas, independientemente del tamaño del gráfico, se incluye como tamaño de la figura la página completa. Para recortar el espacio que no interesa que aparezca en el gráfico se pueden usar varias de las opciones del comando `\includegraphics`, en concreto `viewport` y `clip`

`viewport` especifica la porción de gráfico para ver, esta porción se representa por cuatro números que corresponden a las coordenadas del punto inferior izquierdo y del superior derecho de la ventana sobre el gráfico, tomando como origen el punto inferior izquierdo del BoundingBox, por ejemplo, la figura de la silla del mono tiene como BoundingBox los números 50 50 554 770, por tanto un `viweport` de coordenadas 0 0 504 720 corresponde a la ventana actual de la figura.

La figura 9 se ha obtenido con el siguiente código:

```
\begin{figure}
  \centering
  \fbox{
    \includegraphics[viewport= 0 230 354 500,clip,
angle=-90,width=0.3\textwidth]{sillamono.ps}}
    \caption{Esta es la representación de la
superficie silla de mono recortada}\label{sillamonorecortada}
  \end{figure}
```

La determinación de los valores que debemos poner en el `viewport` pueden obtenerse fácilmente abriendo la figura postscript con `gsview` (en windows) o `gv` en (linux) y observando que la posición del cursor nos indica las coordenadas del punto donde se encuentra (en las unidades correspondientes).

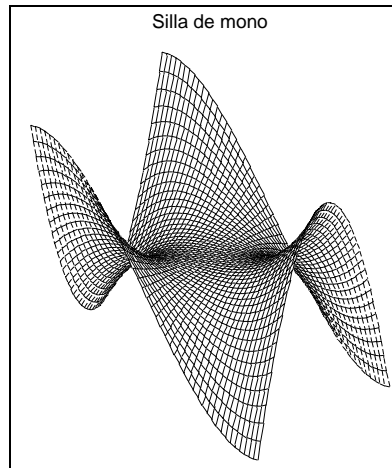


Figura 10: Silla mono recortada automáticamente con gsvie

Un método alternativo más inmediato (en windows) es:

Se abre el fichero con gsvie. En File → PS to EPS, se contesta si a a la pregunta que aparece y se elige un nombre para el fichero, éste último tendrá un BoundingBox ajustado al gráfico. Este método no funciona siempre depende de la “calidad” del fichero postscript considerado, algunos ficheros postscript contienen comandos que impiden que el proceso anterior funcione bien, por lo que en ocasiones debemos utilizar necesariamente el método inicial.

Puede verse el resultado de la utilización de este último método en la [figure 10](#)

10. Personalización de espacios entre figuras, pies de figura, etc.

El espacio entre dos figuras seguidas, entre la figura y el texto que le sigue o precede, la forma de la letra y formato de los pies de figura, la existencia o no de líneas de separación, etc puede personalizarse en gran medida

1. `\setlength{\floatsep}{10pt plus 3pt minus 2pt}`
2. `\setlength{\textfloatsep}{20pt plus 2pt minus 2pt}`
3. `\setlength{\intextsep}{12pt plus 2pt minus 2pt}`
4. `\newcommand{\topfigrule}{\hrule\vspace{-0.4pt}}`
5. etc.

que modifican, respectivamente

1. La separación entre gráficos

2. Separación entre texto y gráfico
3. La separación entre texto y gráfico cuando éste esta en medio
4. Crea una linea de separación entre las figuras situadas en la parte superior de la página y el texto
5. Existen otras muchas personalizaciones del entorno que no mencionamos aquí³

11. Figuras en otros formatos

Figuras en otros formatos también se pueden incluir, si bien es preferible (es una opinión personal) “traducirlas” a postscript y después tratarlas normalmente.

Los formatos más populares son jpeg y tiff, para ambos formatos existen programas que permiten pasar las figuras a postscript de forma rápida y eficiente. Estos programas se pueden compilar para diferentes sistemas operativos.

La incompatibilidad entre formatos impide que en un mismo documento puedan incluirse figuras en formato postscript y en otro formato. Si queremos incluir figuras en formato jpeg, tiff gif, pdf, etc no podremos incluir figuras en formato eps.

La inclusión de figuras con formatos jpeg, gif, tiff, pdf, etc se hace de la misma forma simplemente se carga el paquete `graphicx` con la opción `pdftex`:

```
\usepackage[pdftex]{graphicx}
```

y se crea el fichero final en formato pdf con el compilador pdfL^AT_EX en lugar de utilizar L^AT_EX. El método de inclusión de la figura es el mismo que con las figuras eps, sustituyendo el fichero nombre.eps por nombre.jpg, etc.

12. Rotando y escalando objetos

El paquete `graphicx` incluye algunos comandos que permiten escalar y rotar cualquier objeto L^AT_EX

- `\scalebox{escala horizontal}[escala vert]{argumento}`
- `\resizebox{ancho}{alto}{argumento}`
- `\rotatebox[opciones]{ángulo}{argumento}`

IMPORTANTE: muchos visores de dvi no son capaces de visualizar los cambios producidos por estos comandos, para poder verlos debemos pasarlos a postscript.

Ejemplos: `\scalebox{10}[10]{pepito}`

`\scalebox{10}{pepito}`

`\scalebox{-5}[5]{pepito}`

³Para más información puede verse el trabajo “*Using imported graphics in L^AT_EX2_ε*” que puede encontrarse en <http://www.tex.ac.uk/tex-archive/help/Catalogue/entries/epslatex.html>

producen (en postscript!!!):

pepito
pepito
oʃiqəq

```
\resizebox{3cm}{2cm}{pepito}  
genera
```

pepito

```
podemos rotar \rotatebox{33}{pepito} genera
```

```
podemos rotar pepito
```

Las opciones posibles de `\rotatebox` son:

[*x,y*] coordenadas del punto sobre el que se gira (referidas al punto de la línea base izquierda del objeto)

lt left top

ct center top

rt right top

lc left center

c center

rc right center

lB left baseline

cB center baseline

rB right baseline

lb left baseline

cb center bottom

rb right bottom

Algunos ejemplos:

El siguiente texto

En este ejemplo el `\rotatebox[origin=c]{33}{texto siguiente esta rotado 33 grados respecto al centro}`

produce:

En este ejemplo el

texto siguiente esta rotado 33 grados respecto al centro

En este ejemplo el `\rotatebox[x=-10cm,y=-10cm]{33}{texto est\'a rotado alrededor de}` un centro situado en el punto $(-10\text{cm}, -10\text{cm})$ referido al punto de referencia por defecto de la caja.

texto está rotado alrededor de

En este ejemplo el un centro situado en el punto (-10cm,-10cm) referido al punto de referencia por defecto de la caja.

13. Color. Introducción básica

Este paquete⁴ `color`, al igual que los paquetes gráficos, no se basa en comandos intrínsecos a `TeX` para generar el color, deben aprovechar las facilidades del dispositivo de impresión o del visor para poder realizar esa labor, por tanto son dependientes del sistema.

Los dispositivos (“drivers”) disponibles y que pueden interpretar los comandos del paquete directamente son:

```
dvips, xdvi, dvipdf, dvipdfm, pdftex, dvipsone, dviwindo, emtex  
dviwin, pctexps, pctexwin, pctexhp, pctex32, truetex, tcidvi,  
vtex, oztex, textures.
```

14. Modelos de color

El soporte de color para `TeX` se construye a partir de modelos de color, los modelos de color soportados por un dispositivo pueden variar de dispositivo a dispositivo pero en general casi todos incluyen los modelos `rgb` `cmymk` `gray` y `named`

rgb red green blue, este es un sistema de composición de color a partir de la combinación de los colores rojo, verde y azul como colores primarios la identificación de un color se hace por una terna de números comprendidos entre 0 y 1 que fijan la proporción de cada uno en la mezcla

⁴ En la última distribución de MiTeX se puede ver el catálogo de CTAN en formato html apuntando el navegador a la dirección: `c:\texmf\doc\html\catalogue\catalogue.html` en donde pueden encontrarse mucha información sobre numerosos paquetes

cmymk cyan magenta yellow black, se representa por un conjunto de cuatro números comprendidos entre 0 y 1 representando la proporción de cada uno de los cuatro colores en la mezcla (método usado en impresoras)

gray escala de grises, un número único comprendido entre 0 y 1.

named los colores se identifican por un nombre, blue, green, JungleGreen, etc.

Los visores de documentos en pantalla en ocasiones no son capaces de interpretar los comandos de color y pueden dar una imagen diferente de la que se obtiene en el documento impreso, por esa razón debemos “traducir” el fichero dvi a postscript o pdf para asegurarnos de la apariencia real del documento.

15. Definiendo colores

El usuario puede definir sus propios colores que mediante el comando:
`\definecolor{nombre color}{modelo color}{especificación color}`
por ejemplo `\definecolor{azul-claro}{rgb}{0.8,0.85,1}`

16. Usando colores

16.1. Colores predefinidos

El coloreado de texto se puede hacer de dos formas diferentes:

- `\color{nombre color}` cambia el color del texto que sigue hasta que otro comando del mismo estilo cambie a un nuevo color.
- `\textcolor{nombre color}{texto en ese color}`, este comando es equivalente a

```
{\color{nombre color} texto en color}
```

y colorea solamente `texto en color`.

16.2. Usando las especificaciones de color directamente

Normalmente definiremos todos los colores que se usarán en un documento en el preámbulo del mismo, sin embargo en ocasiones es necesario utilizar colores no definidos anteriormente, los comandos son equivalentes a los vistos en la sección 16.1:

`\color[modelo de color]{especificación color}` ejemplo:

```
\color[rgb]{1,0,0} (color rojo)
```

`\textcolor[modelo]{especificación color}{ texto color}` ejemplo:

```
\textcolor[cmymk]{0,0,1,0}{texto color amarillo}
```

17. Páginas en color

Los colores de fondo de las páginas se pueden conseguir con los comandos

- `\pagecolor{nombre color}`
- `\pagecolor[modelo]{especificación}`

Ambos comandos cambian el color del fondo de la página actual y las siguientes hasta que se cambie de nuevo el color de fondo, o bien se use `\pagecolor{white}` para regresar a páginas sin color de fondo.

18. Cajas coloreadas

Podemos conseguir cajas coloreadas con los siguientes comandos

- `\colorbox{nombre color}{texto dentro}`
- `\colorbox[modelo]{especificacionc color}{texto}`
- `\fcolorbox{nombre color contorno}{color caja}{texto}`
- `\fcolorbox[modelo]{especificacion col contorno}{especificacion col. caja}{texto}`

En la página siguiente vemos algunos ejemplos.

Ejemplos: Esta página tiene como color de fondo el verde, que hemos obtenido con `\pagecolor{green}`

Varias cajas con color

texto en una caja de color azul

caja mezcla de rojo y azul al 50%

caja de fondo amarillo y contorno rojo

caja de fondo amarillo y contorno rojo

caja de fondo amarillo y contorno rojo

Estas cajas se han obtenido con el código:

```
\colorbox{blue}{este texto esta en una caja de
color azul}\[0.3cm]
\colorbox[rgb]{0.5,0.5,1}{esta caja es una mezcla de rojo
y azul al 50\%}\[0.3cm]
\fcolorbox{red}{yellow}{caja de fondo amarillo y
contorno rojo}\[0.3cm]
\setlength{\fboxrule}{3 pt}
\fcolorbox{red}{yellow}{caja de fondo amarillo y
contorno rojo}\[0.3cm]
\setlength{\fboxsep}{5pt}
\fcolorbox{red}{yellow}{caja de fondo amarillo y contorno rojo}
```

19. Nombre de los colores por nombre en dvips

	GreenYellow		Rhodamine		SkyBlue
	Yellow		Mulberry		Turquoise
	Goldenrod		RedViolet		TealBlue
	Dandelion		Fuchsia		Aquamarine
	Apricot		Lavender		BlueGreen
	Peach		Thistle		Emerald
	Melon		Orchid		JungleGreen
	YellowOrange		DarkOrchid		SeaGreen
	Orange		Purple		Green
	BurntOrange		Plum		ForestGreen
	Bittersweet		Violet		PineGreen
	RedOrange		RoyalPurple		LimeGreen
	Mahogany		BlueViolet		YellowGreen
	Maroon		Periwinkle		SpringGreen
	BrickRed		CadetBlue		OliveGreen
	Red		CornflowerBlue		RawSienna
	OrangeRed		MidnightBlue		Sepia
	RubineRed		NavyBlue		Brown
	WildStrawberry		RoyalBlue		Tan
	Salmon		Blue		Gray
	CarnationPink		Cerulean		Black
	Magenta		Cyan		White
	VioletRed		ProcessBlue		

20. Definición de los colores de la página anterior

```
\DefineNamedColor{named}{GreenYellow} {cmyk}{0.15,0,0.69,0}
\DefineNamedColor{named}{Yellow} {cmyk}{0,0,1,0}
\DefineNamedColor{named}{Goldenrod} {cmyk}{0,0.10,0.84,0}
\DefineNamedColor{named}{Dandelion} {cmyk}{0,0.29,0.84,0}
\DefineNamedColor{named}{Apricot} {cmyk}{0,0.32,0.52,0}
\DefineNamedColor{named}{Peach} {cmyk}{0,0.50,0.70,0}
\DefineNamedColor{named}{Melon} {cmyk}{0,0.46,0.50,0}
\DefineNamedColor{named}{YellowOrange} {cmyk}{0,0.42,1,0}
\DefineNamedColor{named}{Orange} {cmyk}{0,0.61,0.87,0}
\DefineNamedColor{named}{BurntOrange} {cmyk}{0,0.51,1,0}
\DefineNamedColor{named}{Bittersweet} {cmyk}{0,0.75,1,0.24}
\DefineNamedColor{named}{RedOrange} {cmyk}{0,0.77,0.87,0}
\DefineNamedColor{named}{Mahogany} {cmyk}{0,0.85,0.87,0.35}
\DefineNamedColor{named}{Maroon} {cmyk}{0,0.87,0.68,0.32}
\DefineNamedColor{named}{BrickRed} {cmyk}{0,0.89,0.94,0.28}
\DefineNamedColor{named}{Red} {cmyk}{0,1,1,0}
\DefineNamedColor{named}{OrangeRed} {cmyk}{0,1,0.50,0}
\DefineNamedColor{named}{RubineRed} {cmyk}{0,1,0.13,0}
\DefineNamedColor{named}{WildStrawberry} {cmyk}{0,0.96,0.39,0}
\DefineNamedColor{named}{Salmon} {cmyk}{0,0.53,0.38,0}
\DefineNamedColor{named}{CarnationPink} {cmyk}{0,0.63,0,0}
\DefineNamedColor{named}{Magenta} {cmyk}{0,1,0,0}
\DefineNamedColor{named}{VioletRed} {cmyk}{0,0.81,0,0}
\DefineNamedColor{named}{Rhodamine} {cmyk}{0,0.82,0,0}
\DefineNamedColor{named}{Mulberry} {cmyk}{0.34,0.90,0,0.02}
\DefineNamedColor{named}{RedViolet} {cmyk}{0.07,0.90,0,0.34}
\DefineNamedColor{named}{Fuchsia} {cmyk}{0.47,0.91,0,0.08}
\DefineNamedColor{named}{Lavender} {cmyk}{0,0.48,0,0}
\DefineNamedColor{named}{Thistle} {cmyk}{0.12,0.59,0,0}
\DefineNamedColor{named}{Orchid} {cmyk}{0.32,0.64,0,0}
\DefineNamedColor{named}{DarkOrchid} {cmyk}{0.40,0.80,0.20,0}
\DefineNamedColor{named}{Purple} {cmyk}{0.45,0.86,0,0}
\DefineNamedColor{named}{Plum} {cmyk}{0.50,1,0,0}
\DefineNamedColor{named}{Violet} {cmyk}{0.79,0.88,0,0}
\DefineNamedColor{named}{RoyalPurple} {cmyk}{0.75,0.90,0,0}
\DefineNamedColor{named}{BlueViolet} {cmyk}{0.86,0.91,0,0.04}
\DefineNamedColor{named}{Periwinkle} {cmyk}{0.57,0.55,0,0}
\DefineNamedColor{named}{CadetBlue} {cmyk}{0.62,0.57,0.23,0}
\DefineNamedColor{named}{CornflowerBlue} {cmyk}{0.65,0.13,0,0}
\DefineNamedColor{named}{MidnightBlue} {cmyk}{0.98,0.13,0,0.43}
\DefineNamedColor{named}{NavyBlue} {cmyk}{0.94,0.54,0,0}
```

```

\DefineNamedColor{named}{RoyalBlue}      {cmyk}{1,0.50,0,0}
\DefineNamedColor{named}{Blue}           {cmyk}{1,1,0,0}
\DefineNamedColor{named}{Cerulean}       {cmyk}{0.94,0.11,0,0}
\DefineNamedColor{named}{Cyan}           {cmyk}{1,0,0,0}
\DefineNamedColor{named}{ProcessBlue}    {cmyk}{0.96,0,0,0}
\DefineNamedColor{named}{SkyBlue}        {cmyk}{0.62,0,0.12,0}
\DefineNamedColor{named}{Turquoise}      {cmyk}{0.85,0,0.20,0}
\DefineNamedColor{named}{TealBlue}       {cmyk}{0.86,0,0.34,0.02}
\DefineNamedColor{named}{Aquamarine}     {cmyk}{0.82,0,0.30,0}
\DefineNamedColor{named}{BlueGreen}      {cmyk}{0.85,0,0.33,0}
\DefineNamedColor{named}{Emerald}        {cmyk}{1,0,0.50,0}
\DefineNamedColor{named}{JungleGreen}    {cmyk}{0.99,0,0.52,0}
\DefineNamedColor{named}{SeaGreen}       {cmyk}{0.69,0,0.50,0}
\DefineNamedColor{named}{Green}          {cmyk}{1,0,1,0}
\DefineNamedColor{named}{ForestGreen}    {cmyk}{0.91,0,0.88,0.12}
\DefineNamedColor{named}{PineGreen}      {cmyk}{0.92,0,0.59,0.25}
\DefineNamedColor{named}{LimeGreen}      {cmyk}{0.50,0,1,0}
\DefineNamedColor{named}{YellowGreen}    {cmyk}{0.44,0,0.74,0}
\DefineNamedColor{named}{SpringGreen}    {cmyk}{0.26,0,0.76,0}
\DefineNamedColor{named}{OliveGreen}     {cmyk}{0.64,0,0.95,0.40}
\DefineNamedColor{named}{RawSienna}      {cmyk}{0,0.72,1,0.45}
\DefineNamedColor{named}{Sepia}          {cmyk}{0,0.83,1,0.70}
\DefineNamedColor{named}{Brown}          {cmyk}{0,0.81,1,0.60}
\DefineNamedColor{named}{Tan}            {cmyk}{0.14,0.42,0.56,0}
\DefineNamedColor{named}{Gray}           {cmyk}{0,0,0,0.50}
\DefineNamedColor{named}{Black}          {cmyk}{0,0,0,1}
\DefineNamedColor{named}{White}          {cmyk}{0,0,0,0}

```

21. Trucos en windows para obtener ficheros posts-cript

Un truco que permite obtener ficheros postscript válidos para incluir en un fichero \LaTeX consiste en instalar una impresora postscript para imprimir en un fichero. Una impresora que funciona bien es la HP **Color LaserJet PS**. El proceso es el normal para instalar una impresora:

- Se añade una impresora nueva (en panel de control impresoras)
- Elegimos “impresora conectada a este equipo” y como puerto se elige “FILE”. Como modelo **Color LaserJet PS**
- Después de instalada, en el panel de impresoras, pincha con el botón derecho del

ratón elige Propiedades → General → Preferencias de impresión → Opciones avanzadas → Opciones postscript → Postscript Encapsulado (EPS)

21.1. Obteniendo ficheros postscript

Podemos ahora obtener un fichero postscript a partir de cualquier programa de windows, simplemente enviamos el fichero deseado a la impresora postscript, le damos el nombre adecuado “nombre.eps” y listo (algunos sistemas se empeñan en añadir una extensión .prn al fichero, si es así se cambia el nombre para que solamente sea ”nombre.eps”).

21.2. El fichero tiene mucho “blanco” alrededor

Muchos de los controladores de windows y algunos programas cometen un importante error, generan un BoundingBox incorrecto (generalmente del tamaño de la mancha de texto), para corregir esto véase la sección [9](#)