Índice

1. Introducción

En este documento se presenta el paquete **xymatrix** que se puede utilizar para generar diagramas directamente en el fichero `tex`.

También se presenta una selección de herramientas (Metapost, JFig, Dia, eps2pdf, fig2dev) que se pueden utilizar para generar gráficos o convertir entre diferentes formatos.

2. xymatrix

`xymatrix` es un paquete que se puede utilizar para dibujar diagramas cuyos elementos se puedan colocar en las celdas de una matriz.

Para indicar que se desea utilizar este paquete hay que poner este preámbulo (para trabajar con pdflatex).

```
\documentclass[a4paper,spanish,11pt]{article}

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

%Para trabajar con latex comentar las dos anteriores y descomentar estas
%\usepackage[dvips]{graphics}
%\usepackage[dvips]{color}


\usepackage{amsmath}

%Para trabajar con acentos
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}

\usepackage[all]{xy}

\begin{document}

\xymatrix{...}

\end{document}
```

Una matriz se puede insertar dentro de una ecuación utilizando el entorno `matrix`.

Ejemplo

Una matriz en Latex

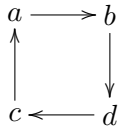
$$\begin{matrix} a & b \\ c & d \end{matrix}$$

```
\[
  \begin{bmatrix}
    a & b \\
    c & d
  \end{bmatrix}
\]
```

Aquellos diagramas que tengan una estructura similar a una matriz se pueden dibujar utilizando el comando `xymatrix`. Una modificación del ejemplo anterior para unir con flechas los cuatro elementos sería:

Ejemplo

Un primer ejemplo de diagrama




```
\xymatrix{
  a  \ar[r] & b \ar[d] \\
  c  \ar[u] & d \ar[l]
}
```

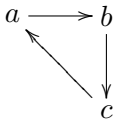
donde \ar[·] indica que se desea dibujar una flecha y el argumento entre corchetes indica la dirección en la que se debe dibujar.

Carácter	Significado
r	una columna hacia la derecha (right)
l	una columna hacia la izquierda (left)
u	una fila hacia arriba (up)
d	una fila hacia abajo (down)

Estas direcciones se pueden combinar, así `ru` significaría hacia arriba y hacia la derecha. Estas direcciones deben apuntar a elementos que existan en la matriz.

Ejercicio 1

Realiza un documento que contenga la siguiente figura:



El estilo de la flecha se puede cambiar utilizando \ar@*estilo*[·]

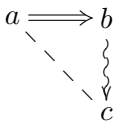
Ejemplo

Estilos de las flechas

```
\xymatrix{
   {\bullet} \ar@{=>}[dr] &  & {\bullet} \ar@{~>}[dr] & & {\bullet} \ar@{-}[dr] &  \\
    & {\bullet} \ar@{:>}[ur] &  & {\bullet} \ar@{-->}[ur] & & {\bullet}
}
```

Ejercicio 2

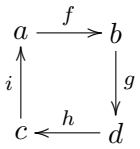
Añade al documento anterior la figura:



Si se desean poner etiquetas a las flechas se puede hacer del siguiente modo:

Ejemplo

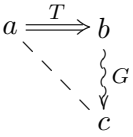
Etiquetas sobre las flechas



```
\xymatrix{
  a   \ar[r]^f & b \ar[d]^g \\
  c   \ar[u]^i & d \ar[l]_h
}
```

Ejercicio 3

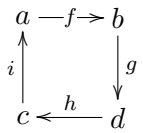
Añade al documento anterior la figura:



También es posible que la etiqueta no esté por encima o por debajo (o a la izquierda o derecha si la flecha es vertical) sino que esté en medio, para ello hay que indicar que se debe producir un corte.

Ejemplo

Etiquetas partiendo las flechas



```
\xymatrix{
  a  \ar[r] |{f} & b \ar[d]^g \\
  c  \ar[u]^i & d \ar[l]_h
}
```

Las flechas no tienen porqué ser rectas, se pueden dibujar con curvatura.

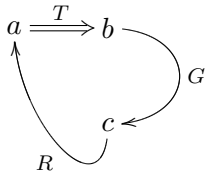
Ejemplo

Flechas curvas

`@/^/`		`\xymatrix{A \ar@/^/[r] & B}`
`@/_/`		`\xymatrix{A \ar@/_/[r] & B}`
`@/_1mm/`		`\xymatrix{A \ar@/_1mm/[r] & B}`
`@(out,in)`		`\xymatrix{A \ar@(u,u)[r] & B}`
`@(out,in)`		`\xymatrix{A \ar@(d,l)[r] & B}`

Ejercicio 4

Añade al documento anterior la figura:



Se puede decorar el texto con rectángulos, circunferencias, se puede hacer que el objeto sea mayor o menor...

La sintaxis es `*{modificador}{elemento}`

- `+` Aumenta un poco el tamaño. Se puede especificar la cantidad utilizando `+<cantidad>`
- `+=` Hace que el tamaño del objeto sea cuadrado aumentando la dimensión menor. `-=` hace que se disminuya la mayor de las dimensiones.
- `[F]` Pone un rectángulo alrededor del texto.
- `[F=]` Pone un rectángulo con linea doble alrededor del texto.
- `[F.]` Pone un rectángulo punteado alrededor del texto.
- `[F--]` Pone un rectángulo con línea discontinua alrededor del texto.
- `[F-,]` Pone un rectángulo sombreado alrededor del texto.
- `[o][F]` Pone una circunferencia alrededor del texto.

Ejemplo

Decoraciones

+	$A \longrightarrow B$	`\xymatrix{*+<5em>{A} \ar[r] & B}`
[F]	$\boxed{A} \longrightarrow B$	`\xymatrix{*[F]{A} \ar[r] & B}`
[F]	$\boxed{A} \longrightarrow B$	`\xymatrix{*+[F]{A} \ar[r] & B}`
[F]	$\boxed{A} \longrightarrow B$	`\xymatrix{*+<1cm>[F]{A} \ar[r] & B}`
[F-,]	$\boxed{A} \longrightarrow B$	`\xymatrix{*+<1cm>[F]{A} \ar[r] & B}`
[o][F]	$\textcircled{A} \longrightarrow B$	`\xymatrix{*+[o][F]{A} \ar[r] & B}`

El texto que aparece está en formato de matemáticas (que es conveniente para poner variables o expresiones matemáticas pero no es conveniente para poner palabras o frases). Cuando sea necesario poner palabras se puede utilizar el comando `\txt` lo mismo es aplicable a las etiquetas que aparecen en las flechas.

Ejemplo

Palabras o frases como elementos

$\boxed{Estoestexto} \longrightarrow B$ `\xymatrix{*+[F]{Esto es texto} \ar[r] & B}`

$\boxed{\text{Esto es texto}} \longrightarrow B$ `\xymatrix{*+[F]\txt{Esto es texto} \ar[r] & B}`

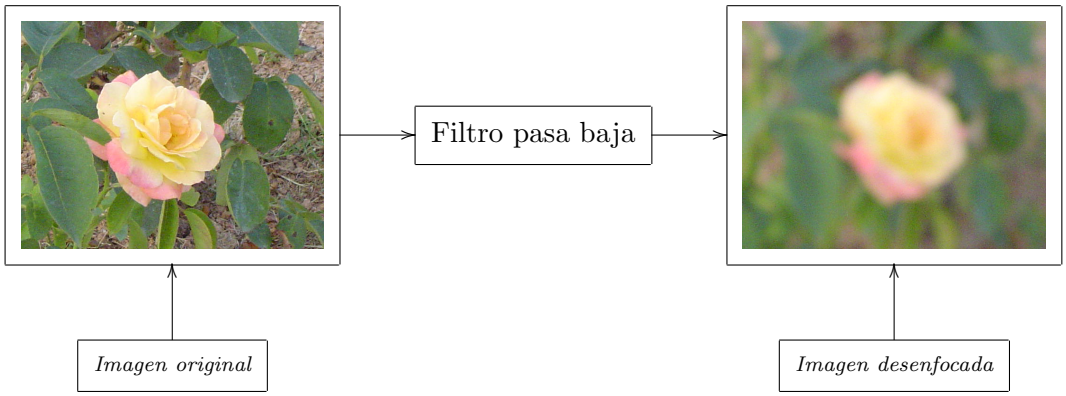
$\boxed{\scriptstyle \text{Esto es texto}} \longrightarrow B$ `\xymatrix{*+[F]{\scriptsize \txt{Esto es texto}} \ar[r] & B}`

Al utilizar el comando `\xymatrix` se pueden pasar argumentos que se aplicarán a todos los elementos de ese diagrama. Por ejemplo, es posible definir que todos los elementos se separen una determinada cantidad, es posible definir cual debe ser el espaciado entre filas o columnas, o que no se tenga en cuenta el tamaño de los elementos,...

- `\xymatrix @=1cm` separa todos los elementos en un centímetro.
- `\xymatrix @R=1cm` separa las filas en un centímetro.
- `\xymatrix @C=1cm` separa las columnas en un centímetro.
- Si antes de utilizar `\xymatrix` ponemos `\entrymodifiers={modificadores}` provoca que todos los elementos incluyan el modificador indicado. Por ejemplo `\entrymodifiers={++[o][F-]}` provoca que todas los elementos aparezcan rodeados con un círculo.

Ejemplo

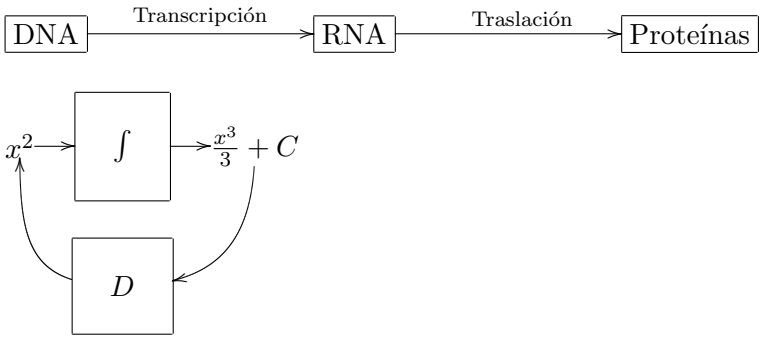
Decoraciones en todos los elementos

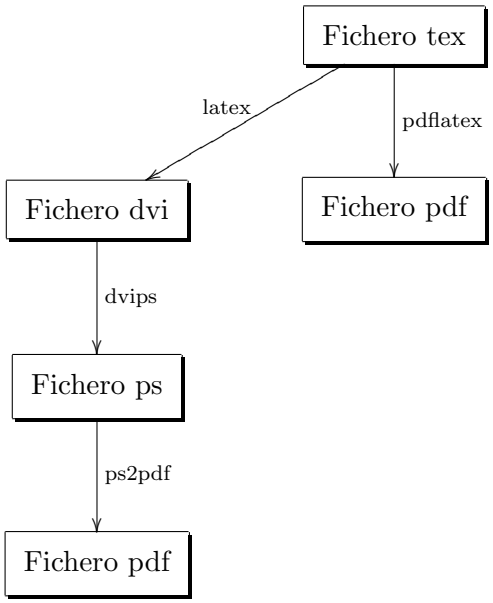



```
\begin{center}
\entrymodifiers={++[F]}
\xymatrix@=1cm{
   \includegraphics[width=4cm]{imagenes/im1.png} \ar[r] & \txt{Filtro pasa baja} \ar[r] &
   \includegraphics[width=4cm]{imagenes/im2.png} \\
   \txt{\scriptsize \em Imagen original} \ar[u] & *{} & \txt{\scriptsize \em Imagen
       desenfocada} \ar[u]
}
\end{center}
```

Ejercicio 5

Añade al documento anterior las figuras:





3. Metapost

Metapost es un lenguaje de programación de gráficos desarrollado por John Hobby en los laboratorios Bell que permite producir gráficos de alta calidad. Está basado en en Metafont de Donald Knuth, pero genera una salida PostScript.

Las figuras se almacenan en un fichero (habitualmente con extensión `mp`). La estructura de este fichero se muestra a continuación:

```
beginfig(1)
  Instrucciones para generar la figura 1
```

```
endfig;

beginfig(2)
  Instrucciones para generar la figura 2
endfig;

...

beginfig(n)
  Instrucciones para generar la figura n
endfig;


end;
```

Supongamos que nuestro fichero con las figuras se llama `figuras.mp`.

Este fichero debe ser procesado mediante la utilidad `mpost` (incluida con la distribución de Miktex). La instrucción será:

```
mpost figuras.mp
```

Este procesado consiste en la generación de un fichero en formato PostScript (que no contiene las fuentes) por cada una de las figuras que había en el fichero original. Así si el fichero anterior se llama `figuras.mp` se generarán los fichero `figuras.1`, ... , `figuras.n` siendo n el número de figuras.

3.1. Metapost y Latex

Las figuras generadas tras ejecutar `mpost` se pueden utilizar en un documento latex tal y como muestra el siguiente código:

```
\documentclass[a4paper,spanish,11pt]{article}

\usepackage[dvips]{graphics}

\usepackage{amsmath}

%Para trabajar con acentos
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}

\begin{document}

\includegraphics{imagenes/figura.1}

\end{document}
```

para ser procesado mediante Latex.

3.2. Metapost y pdflatex

Si en lugar de utilizar Latex se está interesado en utilizar pdflatex hay que realizar los siguientes pasos:

1. Cambiar los nombres de los ficheros `figuras.numero` por `nombrefiguras.mps`

   En el ejemplo anterior se podría cambiar el nombre `figuras.1` por `figuras1.mps`.

2. Incluir la figura en el documento, por ejemplo:

```
\documentclass[a4paper,spanish,11pt]{article}

\usepackage[pdftex]{graphicx}

\usepackage{amsmath}

%Para trabajar con acentos
\usepackage[latin1]{inputenc}
\usepackage[spanish]{babel}

\begin{document}
...
\includegraphics[nombrefiguras.mps]
...
\end{document}
```

El cambio de la extensión del fichero es importante ya que cuando pdflatex encuentra un fichero con extensión `mps` lo convierte automáticamente a un formato compatible con PDF. Si no se hace esto las figuras no se visualizarán.

Este cambio de extensión nos ahorra realizar la conversión por nuestra cuenta (que se puede realizar utilizando la aplicación `mptopdf`).

3.3. Ejemplos y ejercicios

Ejemplo

Un ejemplo de dibujo en Metapost

```
beginfig(1)

draw (0,0)--(10,0)--(10,10)--(0,10)--(0,0);
endfig;
end;
```

Supongamos que el texto del ejemplo se guarda en un fichero llamado `figura.mp`. Este fichero se puede procesar utilizando la utilidad `mpost`:

```
mpost figura.mp
```

generándose dos ficheros:

- `figura.1` que contiene la figura en formato PostScript y
- `figura.log` que contiene información sobre el proceso realizado.

La ruta definida por los puntos

```
(0,0)--(10,0)--(10,10)--(0,10)--(0,0)
```

se puede almacenar en una variable de forma que puede ser utilizada posteriormente, esta variable es de tipo `path`.

Ejemplo

Utilización de una variable para almacenar una ruta

```
beginfig(2)

path p;
p = (0,0)--(10,0)--(10,10)--(0,10)--cycle;

draw p;

endfig;
```

Ejemplo

Desplazamiento de una figura

```
beginfig(3)

path p;
p = (0,0)--(10,0)--(10,10)--(0,10)--cycle;

draw p;
draw p shifted (10,0);

endfig;
```

Ejemplo

Rotación de una figura

```
beginfig(4)

path p;
p = (0,0)--(10,0)--(10,10)--(0,10)--cycle;

for x=10 step 10 until 360:
   draw p rotated x;
endfor;

endfig;
```

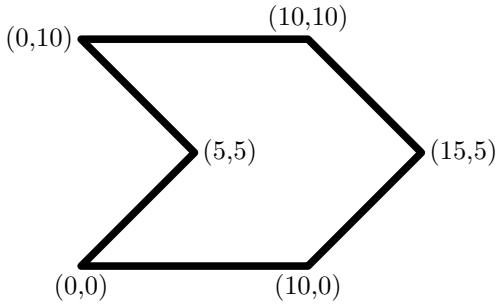
Ejemplo

Desplazamiento y rotación de una figura

```
beginfig(5)

path p;
p = (0,0)--(10,0)--(10,10)--(0,10)--cycle;

for x=5 step 5 until 360:
   draw p rotated x shifted (x,0);
endfor;

endfig;
```

Ejercicio 6

Realiza lo mismo que en los 5 ejemplos anteriores cambiando el cuadrado por la forma cuyas coordenadas se muestran a continuación:



Guarda las figuras en un único fichero con nombre `figuras.mp`. Procesa este fichero con la aplicación `mpost`. Se generarán 5 ficheros `figuras.1`, ..., `figuras.5`. Cambia el nombre a las figuras tal y como se ha explicado anteriormente e inclúyelas en un documento que se procese mediante pdflatex.

Ejemplo

Desplazamiento de una figura rellena con color variable

```
beginfig(6)

path p;
p = (0,0)--(5,0)--(5,10)--(0,10)--cycle;

for x=0 step 0.025 until 1:
   show x*200;
   fill p shifted (x*200,0) withcolor x*red
        + (1-x)*blue;
endfor;

endfig;
```

Ejemplo

Escalado y rotación de una figura rellena

```
beginfig(7)

path p;
p = (0,0)--(10,0)--(10,10)--(0,10)--cycle;

for x=30 step 30 until 360:
   fill p scaled 2 rotated x withcolor 0.6red;
endfor;

fill fullcircle scaled 1cm withcolor white;

endfig;
```

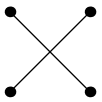
Hemos visto que la variable `path` almacena una ruta. Hay otros tipos de variables que almacenan otros tipos de datos, por citar algunos:

- `numeric` para almacenar un valor numérico.
- `pair` para almacenar un punto con coordenadas x e y.
- `color` para almacenar un color.
- `string` para almacenar texto.
- `picture` para almacenar la figura.

Ejemplo

Uso de la variable pair

```
beginfig(8)

pair pa,pb;
pair pc,pd;

pa:=(10,10);
pb:=(40,40);

draw pa -- pb;
draw pa withpen pencircle scaled 4bp;
draw pb withpen pencircle scaled 4bp;

pc:=(10,40);
pd:=(40,10);

draw pc -- pd;
draw pc withpen pencircle scaled 4bp;
draw pd withpen pencircle scaled 4bp;

endfig;
```

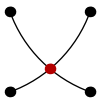


La ruta entre puntos puede ser curva.

Ejemplo

Rutas curvas y puntos de intersección

```
beginfig(9)

pair pa,pb;
pair pc,pd;

pa:=(10,10);
pb:=(40,40);


path lineaa;
lineaa := pa{dir=20}..pb;

draw lineaa;
draw pa withpen pencircle scaled 4bp;
draw pb withpen pencircle scaled 4bp;

pc:=(10,40);
pd:=(40,10);

path lineab;
lineab := pc{dir=-70}..pd;

draw lineab;
draw pc withpen pencircle scaled 4bp;
draw pd withpen pencircle scaled 4bp;

draw lineaa intersectionpoint lineab withpen
    pencircle scaled 4bp withcolor 0.7red;


endfig;
```



Es posible añadir texto utilizando `label`. Si se desea poner un punto y una etiqueta se puede utilizar `dotlabel`.

Se puede controlar la posición en la que aparecerá el texto respecto a la posición utilizando:

```
dotlabel.pos("texto",punto)
```

o,

```
label.pos("texto",punto)
```

donde `pos` puede ser:

top	Sobre el punto
bot	Debajo del punto
rt	A la derecha del punto
lft	A la izquierda del punto
urt	Sobre el punto a la derecha
lrt	Bajo el punto a la derecha
ulft	Sobre el punto a la izquierda
llft	Bajo el punto a la izquierda

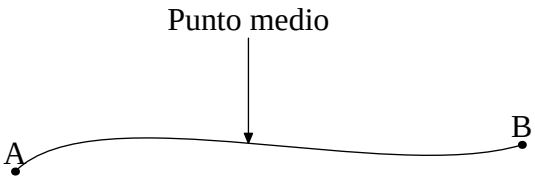
Si se desea tener control sobre el texto se puede poner entre `btex` y `etex` (que son abreviaturas de begintex y endtex respectivamente). Por ejemplo para poner $\sqrt{2}$ a la derecha del punto $(0,0)$ se puede hacer lo siguiente:

```
label.rt(btex $\sqrt 2$ etex, (0,0))
```

Ejemplo

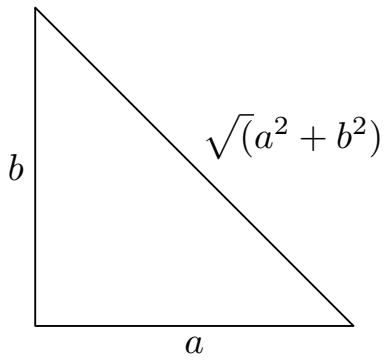
Etiquetas, flechas y punto medio

```
beginfig(10)

defaultfont := "tir";
defaultscale := 12pt/fontsize(defaultfont);

pair pa,pb;
pair med,etiq;

pa:=(10,10);
pb:=(200,20);

path lineaa;
lineaa := pa.. controls (40,40) and (150,5) ..
    pb;
draw lineaa;

%Punto medio
med := point 1/2length(lineaa) of lineaa;


dotlabel.top("A",pa);
dotlabel.top("B",pb);

etiq := (xpart med,60);
drawarrow etiq--med;
label.top("Punto medio",etiq);


endfig;
```



Ejercicio 7

Realiza la siguiente figura:

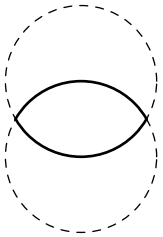


Metapost es capaz de encontrar la intersección entre dos caminos cerrados.

Ejemplo

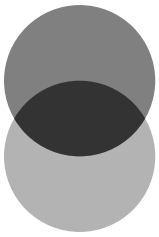
Intersección entre rutas

```
beginfig(11)
  u:=2cm;
  path c[];
  c[1] := fullcircle scaled u;
  c[2] := c[1] shifted (0,.5u);
  draw c[1] dashed evenly;
  draw c[2] dashed evenly;
  draw buildcycle(c[1],c[2]) withpen pencircle
      scaled 1bp;
endfig;
```



Ejercicio 8

Realiza la siguiente figura:



También es posible trabajar con funciones. En el ejemplo siguiente se dibuja la función

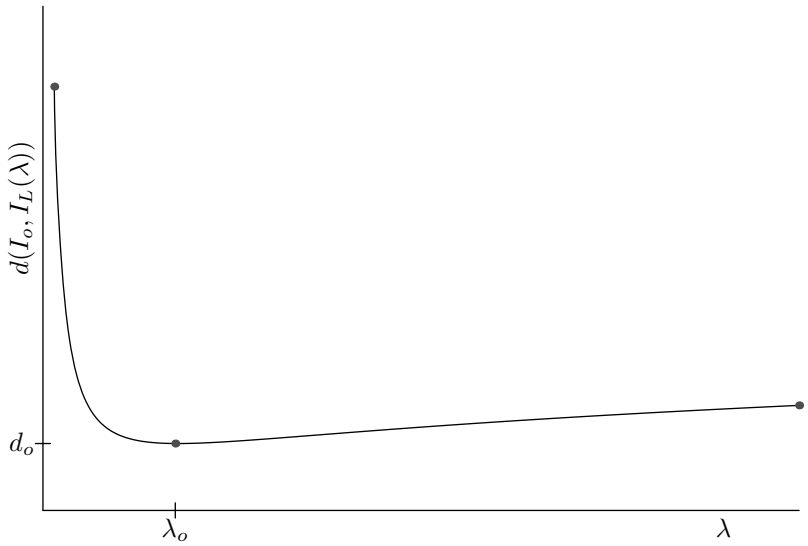
$$\frac{1}{x} + 0{,}56\ln(x)$$

en el rango de valores $x \in [0{,}15, 10]$ utilizando un incremento de 0,1. Además se encuentra el mínimo de la función y se dibujan los ejes utilizando símbolos matemáticos.

Ejemplo (Complejo)

Funciones

```
beginfig(12);
defaultfont := "tir";
defaultscale := 18pt/fontsize(defaultfont);

numeric xmin, xmax, ymin, ymax;
xmin := 0.15; xmax := 10; ymax := 1/xmin;
u := 1cm;

%Definicion de la funcion ln
vardef ln(expr x) = (mlog(x)/256) enddef;
%Definicion de la funcion f
vardef f(expr x) = 1/x + 0.56*ln(x) enddef;

xinc := 0.1;
path pts_f;

%Esto construye el path de la curva (se utiliza la funcion f)
pts_f := (xmin,f(xmin))*u
for x=xmin+xinc step xinc until xmax+xinc:
.. (x,f(x))*u
endfor;
draw pts_f;

%Esto sirve para encontrar el minimo de la funcion
numeric minx, miny,valy;
minx = xmin;
miny = f(xmin);

for x=xmin+xinc step xinc until xmax+xinc:
   valy := f(x);
   if ( valy < miny):
     minx := x;
     miny := valy;
   fi;
endfor;

pair minimo, inicial, final;
minimo := (minx,miny)*u;

inicial = (xmin,f(xmin))*u;
final = (xmax,f(xmax))*u;

draw minimo withpen pencircle scaled 3pt withcolor 0.3white;
draw inicial withpen pencircle scaled 3pt withcolor 0.3white;
draw final withpen pencircle scaled 3pt withcolor 0.3white;

path hline, vline;
hline = (0,0)*u -- (xmax,0)*u;
vline = (0,0)*u -- (0,ymax)*u;
draw hline;
draw vline;
label.bot(btex $\lambda$ etex, (0.9xmax,0)*u);
label.lft(btex $d(I_o,I_L(\lambda))$ etex rotated 90, (0,0.6ymax)*u);

draw (xpart minimo,-u/10) -- (xpart minimo,u/10);
label.bot(btex $\lambda_o$ etex,(xpart minimo, 0));

draw (-u/10,ypart minimo) -- (u/10,ypart minimo);
label.lft(btex $d_o$ etex,(0,ypart minimo));
endfig;
```



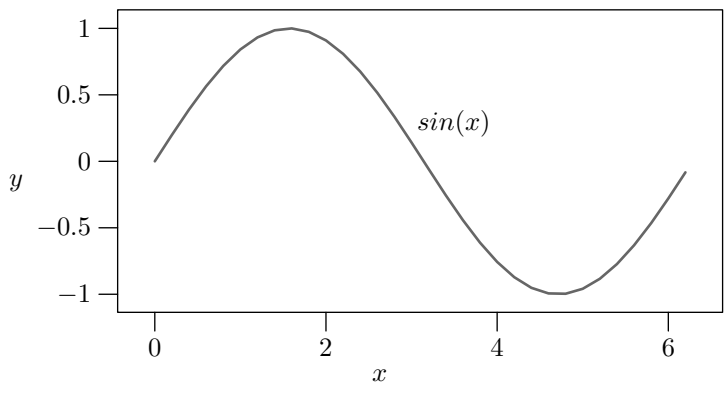
También es posible dibujar un gráfico a partir de una tabla de datos. Para ello hay que importar el fichero `graph.mp` e indicar el fichero que contiene los datos.

Ejemplo

Gráficas de datos

Supongamos un fichero `datos.dat` que contiene pares de puntos del tipo $(x_n, sin(x_n))$ en cada fila. Para representarlos se puede hacer lo siguiente:

```
input graph;

beginfig(13);

draw begingraph(8cm,4cm);
   glabel.lft( btex \vbox{\hbox{$y$}} etex, OUT);
   glabel.bot( btex \vbox{\hbox{$x$}} etex, OUT);
   gdraw "datos.dat" withpen pencircle scaled 1pt withcolor 0.4white;
   glabel.urt(btex $sin(x)$ etex,15);
endgraph;

endfig;
```



4. Dia

Este programa se puede utilizar para hacer diagramas de flujo, diagramas eléctricos, diagramas UML, etc

Permite exportar las figuras a `png` a `eps` e incluso a Metapost (con lo cual se puede editar y cambiar lo que deseemos).

El gráfico exportado no coincide exactamente con lo que se ve en la pantalla (lo cual es un poco desagradable) pero... es *freeware*.

5. JFig y fig2dev

JFig es una aplicación similar al famoso Xfig de Unix. JFig está realizado en Java por lo que es necesario tener instalada una máquina virtual de Java (Java Runtime Environment). Antes era *freeware* pero ahora es *shareware*.

Guarda las figuras en formato `*.fig`. Se puede utilizar `fig2dev` para realizar conversiones a otros formatos (como por ejemplo postscript).

Por ejemplo si tenemos una figura realizada con JFig y deseamos exportarla a Postscript encapsulado (suponiendo que `fig2dev` está en el path):

```
fig2dev -L eps -m 0.5 figura.fig figura.eps
```

donde lo que sigue a la opción `-L` es el lenguaje al que se desea exportar y lo que sigue a la opción `-m` es el escalado global.

6. eps2pdf

Si el programa de gráficos que estamos utilizando permite exportar a Postscript encapsulado pero no a PDF, y deseamos trabajar con pdflatex (que no admite Postscript como formato gráfico) se pueden transformar los ficheros `*.eps` mediante la utilidad `eps2pdf`.

La figura exportada a eps en la sección anterior se podría incluir en un fichero para ser procesado mediante `latex` pero si se desea incluir en un fichero que sea procesado mediante `pdflatex` hay que realizar un paso adicional para convertirla a pdf:

```
eps2pdf /f figura.eps
```