

Recetario de gnuplot (versión 5.2)

26 de febrero de 2018

Inicio de gnuplot

\$ gnuplot

```
$ gnuplot

  G N U P L O T
  Version 5.2 patchlevel rc4   last modified 2017-07-30

  Copyright (C) 1986-1993, 1998, 2004, 2007-2017
  Thomas Williams, Colin Kelley and many others

  gnuplot home:      http://www.gnuplot.info
  faq, bugs, etc:   type "help FAQ"
  immediate help:   type "help" (plot window: hit 'h')

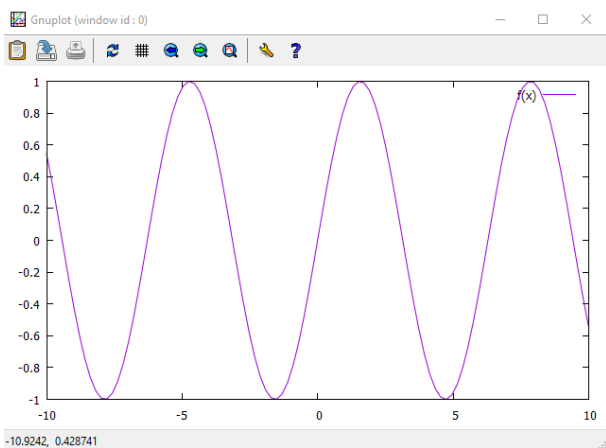
Terminal type is now 'wxt'
Options are '0 enhanced'
Encoding set to 'cp1252'.
gnuplot>
```

Representación de funciones de una variable $y = f(x)$.

Supongamos la función $f(x) = \sin(x)$. Vamos a representarla usando el siguiente texto en línea de comandos:

```
gnuplot>
gnuplot> f(x)=sin(x)
gnuplot> plot f(x) w l
gnuplot>
```

la salida es:



Se puede cambiar el rango en el eje X y en el eje Y de representacion de la función con los comandos:

- ▶ **set xrange [xmin:xmax], set yrange [ymin:ymax], set zrange [zmin:zmax]** (esta última para representaciones en 3D, ver más adelante)

Representación de datos $y = f(x)$ almacenados en un fichero, en la forma (x, y)

Tenemos un fichero que donde tenemos almacenados dos columnas de datos en el formato (x, y) . Listamos el contenido del fichero con el comando unix “**more file.dat**”. La salida es:

```
MINGW32- MINGW32-~/FISICA_COM- |
# Curve 0 of 1, 100 points
# Curve title: "sin(x)*rand(0)"
# x y type
-10 0.0830378 i
-9.79798 0.249472 i
-9.59596 0.124985 i
-9.39394 -0.0170436 i
-9.19192 -0.0152347 i
-8.9899 -0.414337 i
-8.78788 -0.529669 i
-8.58586 -0.00815533 i
-8.38384 -0.582387 i
-8.18182 -0.360689 i
-7.9798 -0.405765 i
-7.77778 -0.84953 i
-7.57576 -0.358111 i
-7.37374 -0.396343 i
-7.17172 -0.380777 i
-6.9697 -0.134892 i
-6.76768 -0.457475 i
-6.56566 -0.210489 i
-6.36364 -0.0422447 i
-6.16162 0.0829004 i
-5.9596 0.0286624 i
-5.75758 0.110811 i
-5.55556 0.267146 i
-5.35354 0.476204 i
-5.15152 0.299611 i
-4.94949 0.0821215 i
-4.74747 0.370798 i
file.dat
Ready
```

Nota: gnuplot no lee las líneas que empiezan con #

Representamos dichos datos, comenzamos gnuplot y hacemos:

```
$ gnuplot

G N U P L O T
Version 5.2 patchlevel rc4   last modified 2017-07-30

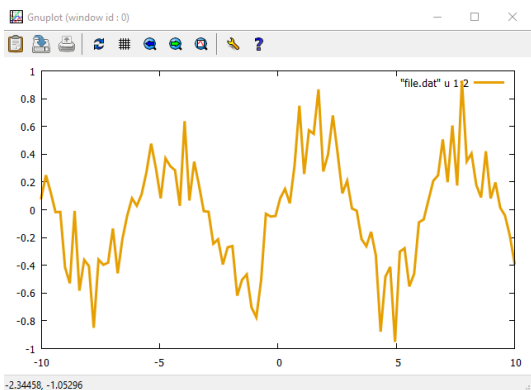
Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
Options are '0 enhanced'
Encoding set to 'cp1252'.
gnuplot> plot "file.dat" u 1:2 w l lt 4 lw 3
gnuplot>

Ready
```

cuya salida es:



Vamos a explicar dicho línea de comandos: Primero se escribe “**plot**” que es el comando general para dibujar en dos dimensiones(x, y). Después se incluye el fichero a leer que se pone entre comillas “” o entre ‘ ’. A continuación se usa el comando **using** (u abreviado) que va a indicar la columnas a representar, en este caso la columna 1 (para la x) y la 2 (para la y). A continuación un comando que indica como se va pintar los puntos del fichero y cuyos valores mas usados son:

w l	une los puntos con líneas
w p	dibuja los puntos y sin líneas
w lp	dibuja con líneas y puntos
w steps	dibuja con líneas horizontales entre los puntos

Cada uno de estos comandos tiene una serie de opciones como son el tipo de línea (**lt**), la anchura de la línea (**lw**), el tipo de punto (**pt**), el tamaño del punto (**ps**), etc, cada uno seguido de un numero que indica el valor asignado a cada opción (ver la ayuda help de gnuplot para mas opciones). En el ejemplo de mas arriba se ha representado los puntos con líneas, con el tipo de línea 4, y con el grosor de línea 3.

La función del último ejemplo se ha generado multiplicando la función $\sin(x)$ por un número aleatorio entre $(0,1)$. Representamos ambas funciones (función $\sin(x)$ y los datos del fichero:

```
$ gnuplot
```

```
GNUPLOT
Version 5.2 patchlevel rc4   last modified 2017-07-30

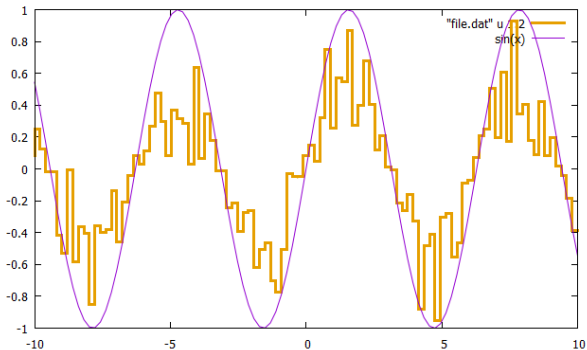
Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help"   (plot window: hit 'h')
```

```
Terminal type is now 'wxt'
Options are '0 enhanced'
Encoding set to 'cp1252'.
gnuplot> plot 'file.dat' u 1:2 w steps lt 4 lw 3, sin(x) w l
gnuplot> plot 'file.dat' u 1:2 w steps lt 4 lw 3, sin(x) w l lt 1
gnuplot>
```

```
Ready
```

que da como salida:



Notad dos cosas: 1) la figura en el fichero la hemos representado con `w steps`, y segundo la función original la hemos representado con `l t 1`, que es la línea violeta. Si no lo hubiésemos incluido la hubiese dibujado en azul que es el tipo de línea 2, ya que la función seno se pinta en segundo lugar en el anterior ejemplo y por defecto le asigna el tipo de línea 2.

Poniendo etiquetas a los ejes

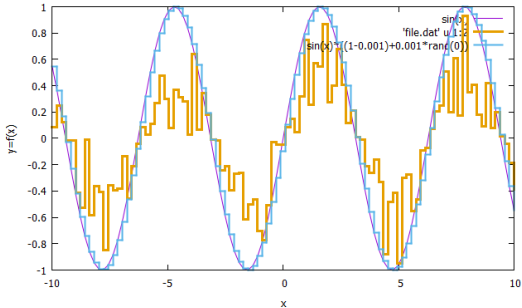
Se utiliza el comando `set xlabel 'xlabel'`, y `set ylabel 'ylabel'`

```
$ gnuplot
      G N U P L O T
      Version 5.2 patchlevel rc4   last modified 2017-07-30
      Copyright (C) 1986-1993, 1998, 2004, 2007-2017
      Thomas Williams, Colin Kelley and many others

      gnuplot home:      http://www.gnuplot.info
      faq, bugs, etc:    type "help FAQ"
      immediate help:    type "help" (plot window: hit 'h')

      Terminal type is now 'wxt'
      Options are '0 enhanced'
      Encoding set to 'cp1252'.
      gnuplot> plot sin(x) w l, 'file.dat' u 1:2 w steps lt 4 lw 3, sin(x)*((1-0.001)+0.001*rand(0)) w ste
      ps lw 2
      gnuplot> set xlabel 'x'
      gnuplot> set ylabel 'y=f(x)'
      gnuplot> replot
      gnuplot>
```

que da como salida:



El tamaño de las etiquetas de los ejes se puede cambiar y también admite expresiones latex y formulas matemáticas (ver terminal epslatex)

Nota: en todas la gráficas, por defecto, se incluye en la esquina superior derecha información de lo que representa cada línea. Esta “key” se puede quitar usando el comando

gnuplot> unset key

Nota: Siempre que hagamos una modificación a la gráfica podemos incluirla después del nuevo comando introducido haciendo

gnuplot>replot

Representación de superficies en 3D, $z = f(x, y)$

La representación en 3D es una extensión de la representación en 2D, incluyendo una nueva variable. Por lo tanto todos los comandos de las representación en 2D se pueden extender a 3D. Vamos a ver un ejemplo. Vamos a representar la función

$$z = \text{sinc}(x, y) = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$$

El comando básico es “**splot**”. Veamos un ejemplo

```
$ gnuplot

  G N U P L O T
  Version 5.2 patchlevel rc4   last modified 2017-07-30

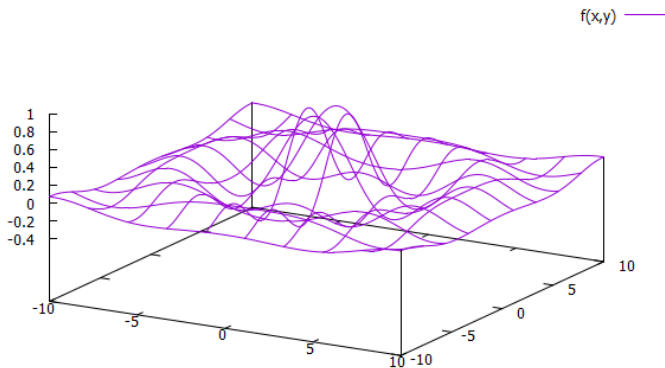
  Copyright (C) 1986-1993, 1998, 2004, 2007-2017
  Thomas Williams, Colin Kelley and many others

  gnuplot home:      http://www.gnuplot.info
  faq, bugs, etc:   type "help FAQ"
  immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
Options are '0 enhanced'
Encoding set to 'cp1252'.
gnuplot> f(x,y)=(sin(sqrt(x*x+y*y)))/(sqrt(x*x+y*y))
gnuplot> splot f(x,y) w l
gnuplot>
```

Ready

cuya salida es:

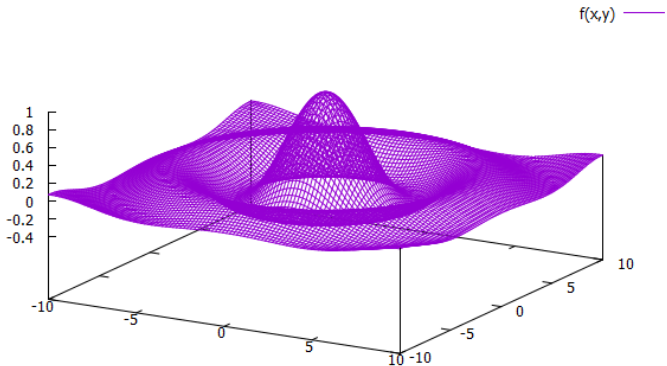


que no es una figura muy óptima. Hay varias formas de optimizarla.

Primero aumentaremos la rejilla de la superficie con el comando

```
gnuplot> set isosamples 100,100
```

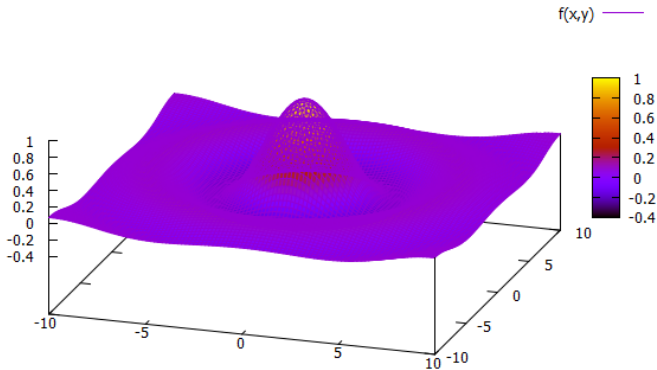
La salida es una rejilla 100x100 (el que se pone por defecto es una rejilla 10x10):



Podemos incluir una paleta de colores para la superficie con el comando

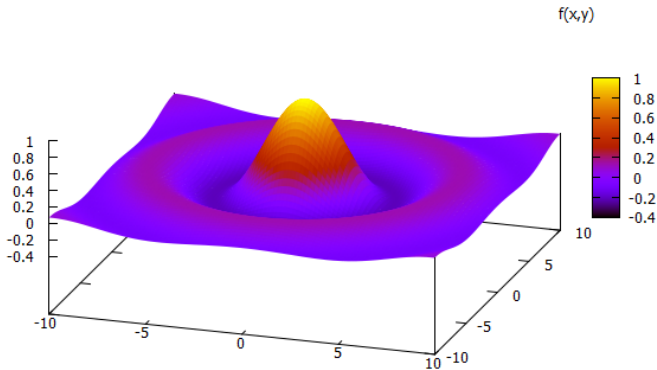
```
gnuplot> set pm3d
```

que después de repintar nos queda



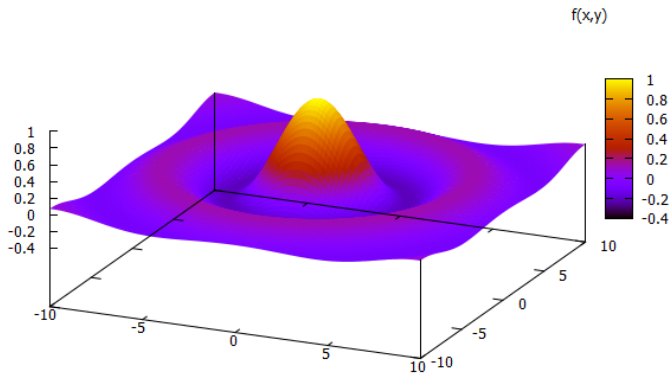
Podemos quitar la rejilla, pues en este caso oculta la superficie, con el comando

gnuplot> **unset surface**
que después de repintar da



Podemos hacer la superficie transparente y mostrar los ejes con el comando

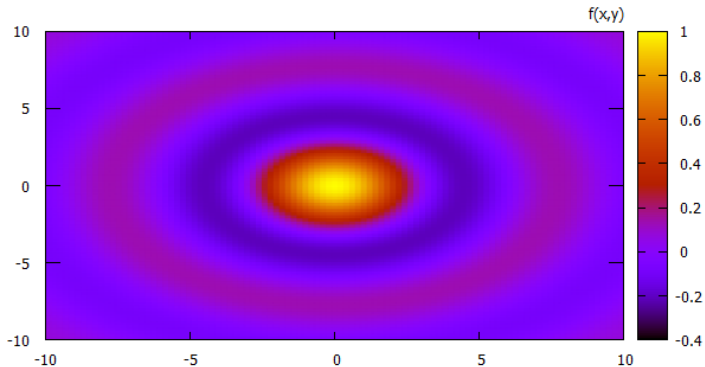
gnuplot> set hidden3d



Por último podemos ver la proyección de dicha superficie en el plano (x,y) usando el comando

gnuplot> set view map

que después de repintar con “replot” da la salida:



A continuación se ponen los comandos usados para llegar a esta última gráfica

```
$ gnuplot

G N U P L O T
Version 5.2 patchlevel rc4   last modified 2017-07-30

Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
Options are 'O enhanced'
Encoding set to 'cp1252'.
gnuplot> f(x,y)=(sin(sqrt(x*x+y*y)))/(sqrt(x*x+y*y))
gnuplot> splot f(x,y) w l
gnuplot> set isosamples 100,100
gnuplot> replot
gnuplot> set pm3d
gnuplot> replot
gnuplot> unset surface
gnuplot> replot
gnuplot> set hidden3d
gnuplot> replot
gnuplot> set view map
gnuplot> rep
gnuplot> □
```

Ready

32x100

Haciendo películas con gnuplot

Hay varias formas de hacer películas con gnuplot. Veamos dos formas:

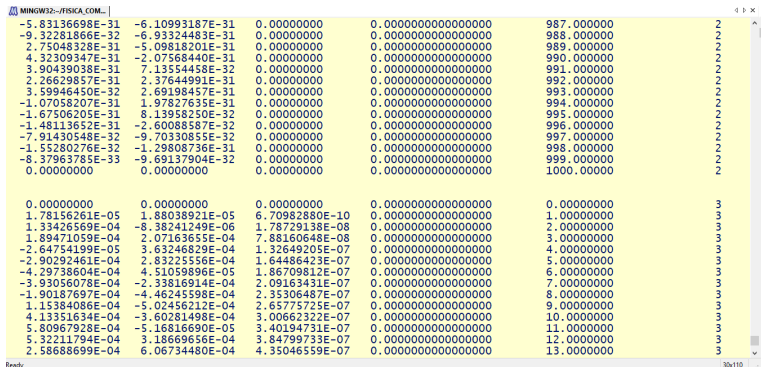
- A. La forma tradicional de hacer una película es haciendo un gif animado de un conjunto de **ficheros png**, cada uno de ellos siendo un fotograma de la película. Para esto hay que incluir bucles en gnuplot donde en cada paso del bucle se genera un fotograma (en formato png) y después se unen todos haciendo un gif animado. Ver la página <http://www.gnuplotting.org/tag/animation/> para más información.
- B. Lo podemos hacer de forma más sencilla **de forma interactiva con un bucle en la línea de comandos**. La sintaxis para hacer el bucle es la que sigue:

```
$ gnuplot
G N U P L O T
Version 5.2 patchlevel rc4   last modified 2017-07-30
Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:   http://www.gnuplot.info
faq, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
Options are '0 enhanced'
Encoding set to 'cp1252'.
gnuplot> do for [i=1:300] {plot 'prueba.dat' index i u 5:3 w l, '' index i u 5:4 w l}
```

Para que la película funcione hay que incluir una línea en blanco en el fichero de datos después de los datos correspondientes a cada fotograma, como la línea en blanco que se muestra a continuación (entre los datos del fotograma 2 y 3):



The image shows a screenshot of a text editor window titled "MINGW32--/FISICA_COM...". The window contains a list of data points, each consisting of a numerical value followed by a unit code. The data is organized into two main groups. The first group contains 20 lines of data, each with five columns of values and a column of unit codes (mostly "-31" and "-32"). The second group also contains 20 lines of data, each with five columns of values and a column of unit codes (mostly "-05", "-04", and "-07"). A single blank line is inserted between the 20th line of the first group and the 21st line of the second group. The status bar at the bottom of the window shows "Ready" and "30x110".

-5.83136698E-31	-6.10993187E-31	0.00000000	0.000000000000000000	987.000000	2
-9.32281866E-32	-6.93324448E-31	0.00000000	0.000000000000000000	988.000000	2
2.75048328E-31	-5.09818201E-31	0.00000000	0.000000000000000000	989.000000	2
4.32309347E-31	-2.07568440E-31	0.00000000	0.000000000000000000	990.000000	2
3.90439038E-31	7.13554458E-32	0.00000000	0.000000000000000000	991.000000	2
2.26629857E-31	2.37644991E-31	0.00000000	0.000000000000000000	992.000000	2
3.59946450E-32	2.69198457E-31	0.00000000	0.000000000000000000	993.000000	2
-1.07058207E-31	1.97827635E-31	0.00000000	0.000000000000000000	994.000000	2
-1.67506205E-31	8.13958250E-32	0.00000000	0.000000000000000000	995.000000	2
-1.48113652E-31	-2.60088587E-32	0.00000000	0.000000000000000000	996.000000	2
-7.91430548E-32	-9.70330855E-32	0.00000000	0.000000000000000000	997.000000	2
-1.55280276E-32	-1.29808736E-31	0.00000000	0.000000000000000000	998.000000	2
-8.37963785E-33	-9.69137904E-32	0.00000000	0.000000000000000000	999.000000	2
0.00000000	0.00000000	0.00000000	0.000000000000000000	1000.000000	2
0.00000000	0.00000000	0.00000000	0.000000000000000000	0.00000000	3
1.78156261E-05	1.88038921E-05	6.70982880E-10	0.000000000000000000	1.00000000	3
1.33426569E-04	-8.38241249E-06	1.78729138E-08	0.000000000000000000	2.00000000	3
1.89471059E-04	2.07163655E-04	7.88160648E-08	0.000000000000000000	3.00000000	3
-2.64754199E-05	3.63246829E-04	1.32649205E-07	0.000000000000000000	4.00000000	3
-2.90292461E-04	2.83225556E-04	1.64486423E-07	0.000000000000000000	5.00000000	3
-4.29738604E-04	4.51059896E-05	1.86709812E-07	0.000000000000000000	6.00000000	3
-3.93056078E-04	-2.33816914E-04	2.09163431E-07	0.000000000000000000	7.00000000	3
-1.90187697E-04	-4.46245598E-04	2.35306487E-07	0.000000000000000000	8.00000000	3
1.15384086E-04	-5.02456212E-04	2.65775725E-07	0.000000000000000000	9.00000000	3
4.13351634E-04	-3.60281498E-04	3.00662322E-07	0.000000000000000000	10.00000000	3
5.80967928E-04	-5.16816690E-05	3.40194731E-07	0.000000000000000000	11.00000000	3
5.32211794E-04	3.18669656E-04	3.84799733E-07	0.000000000000000000	12.00000000	3
2.58688699E-04	6.06734480E-04	4.35046559E-07	0.000000000000000000	13.00000000	3

Esta forma de crear películas por línea de comandos es muy útil pues nos permite ver de forma interactiva la película. Además podemos crear de forma muy sencilla el gif animado desde la línea de comandos usando la terminal **“gif animate”**:

```
gnuplot> set term gif animate
Terminal type is now 'gif'
Options are 'nocrop enhanced animate delay 10 loop 0 nooptimize size 640,480 font "arial,12.0" '
gnuplot> set output "animate.gif"
gnuplot> do for [i=1:300] {plot 'prueba.dat' index i u 5:3 w l, '' index i u 5:4 w l}
```

Ejercicio: comprobar que la salida es un gif animado.

Uso de scripts

Todos los comandos que ponemos en la línea de comandos se pueden incluir de forma secuencial en un fichero o script, que llamaremos por ejemplo "figura.plot". Después desde cualquier ventana unix, podemos ejecutar el comando:

\$ gnuplot figura.plot

que nos dará la salida deseada según los comandos incluidos en el script

Ejemplo:

```
#contenidos del fichero figura.plot
set terminal pdfcairo size 8cm,6cm
set output 'figura.pdf'
set origin 0,0
sinc(x,y)=sin(sqrt(x*x+y*y))/(sqrt(x*x+y*y))
set samples 1000
set isosamples 100,100
set pm3d
unset surface
plot sinc(x,y) w l
```

Ejercicio: comprobar la salida del script anterior