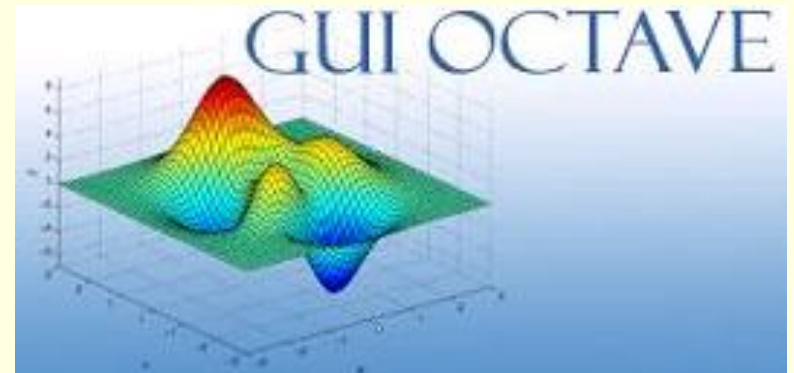
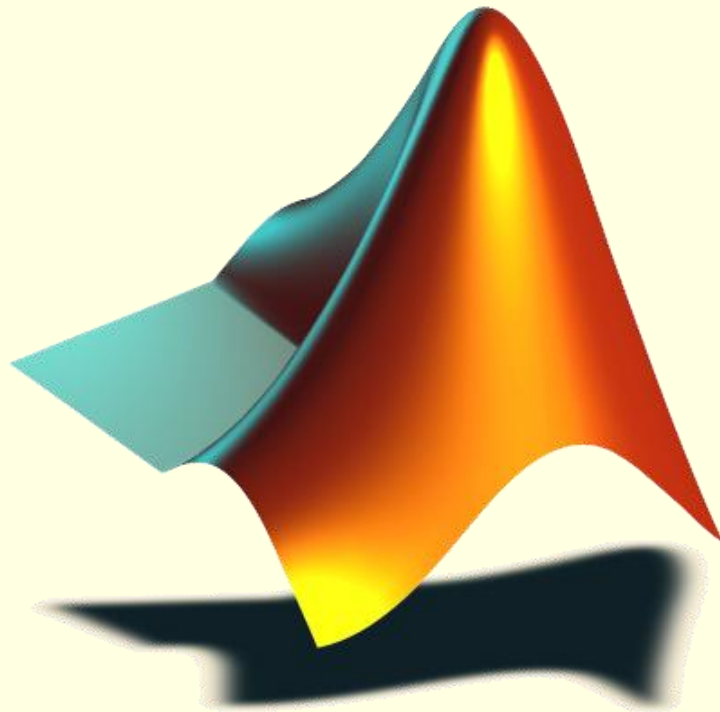


Laboratorio de Computación Científica

Primer semestre

Programación MATLAB y OCTAVE

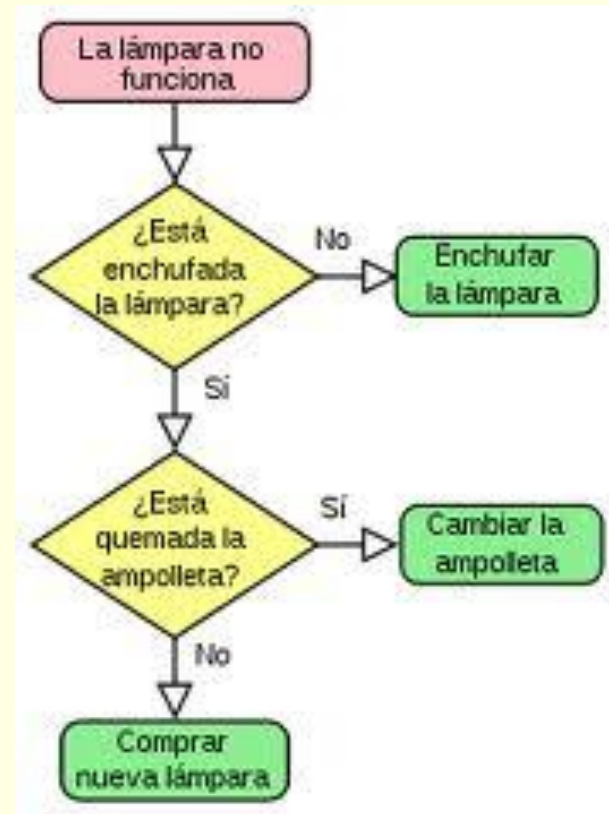


Temario

- Programación
 - Bucles *for/while*
 - Condicionales *if*
- Funciones y Scripts (M-files)
- Representación gráfica
- Entrada/Salida

Programación

- Diagrama flujo
 - Esquema gráfico del programa
 - Ayuda a comprender su evolución
 - Traducción casi directa a un lenguaje alto-nivel



Programación

■ Bucles *for/while*

■ *for*: itera entre valores conocidos

```
for variable = expresión  
    <orden>  
    ...  
end
```



```
>> for x = 1:5,  
    disp ('x toma el valor');  
    disp (x)  
end
```

■ *while*: itera hasta que se deja de cumplir una condición

```
while <expresión>  
    <orden>  
    ...  
end
```



```
>> a=3;  
>> while a < 5,  
    disp ('a es menor que 5 ya que vale')  
    disp (a)  
    a = a + 1;  
en
```

Programación

■ Operadores condicionales

Operador	Significado
<	Menor que
<=	Menor o igual
>	Mayor que
>=	Mayor o igual
==	Igual que
~=	Distinto que

■ Operadores lógicos

Operador	Significado
&	y
	o
~	no

■ Y además

- xor (x,y) operación “o” exclusiva,
- any (x) → ¿algún?
- all (x) → ¿todos?

Programación

- Sentencia if
 - Evalúa una expresión lógica

```
if <expresión1>
    <orden>
    ...
elseif <expresión2>
    <orden>
    ...
else
    <orden>
    ...
end
```



```
>> if x<5,
        disp('x es menor que 5');
end
```

Programación

■ Funciones o M-file

- Un función, .m o M-file, es un fichero de texto que agrupa un conjunto de instrucciones
- Matlab/octave interpreta las instrucciones del fichero en secuencia
- Sintaxis del fichero nombre_funcion.m:
function [var_salida] = nombre_funcion (var_entrada)

```
function sum=suma(a,b)
    sum=a+b;
```

```
>> sum=suma(3,4)
    sum=7
```

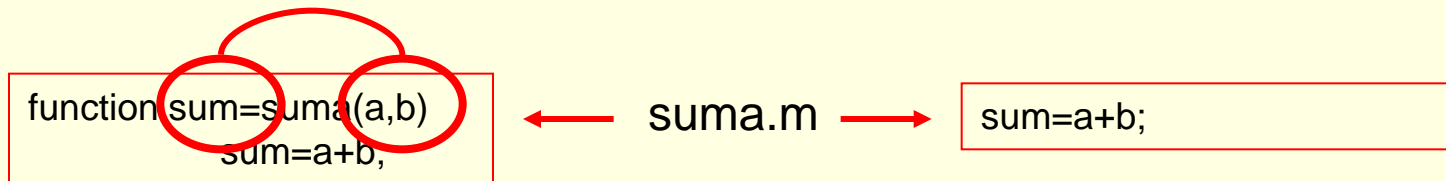
Programación

■ Scripts

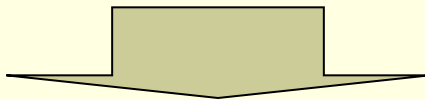
- Colección de sentencias que se ejecutan de forma secuencial escritas en un fichero
- Diferencia scripts-funciones
 - Scripts (no tiene parámetros de entrada ni salida)
 - Funciones (tienen parámetros de entrada y salida)
 - Scripts (variables definidas tienen ámbito global)
 - Funciones (variables definidas tienen ámbito local)

Programación

■ Ejemplo diferencia función/script parámetros de entrada y salida

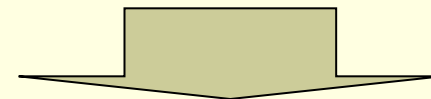


```
>> x=suma(3,4)
      x=7
>>a
???' Undefined function or variable 'a'.
```



a, b y sum no definidos
x se ha definido y vale 7

```
>> a=3; b=4;
>> suma;
>> sum
      sum =
      7
```

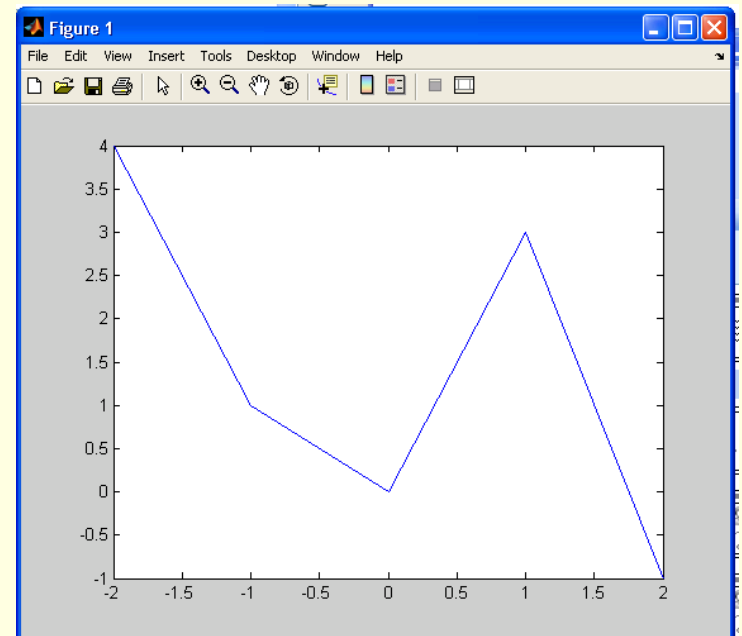
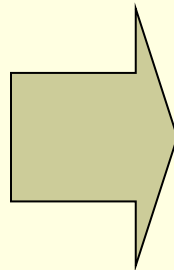


a, b y sum definidos
a vale 3
b vale 4
sum vale 7

Representación gráfica

- Difieren la visualización en Matlab/octave
- Figuras 2D
 - `plot(xdata1,ydata1,'estilo_color', xdata2.....);`
 - Dibuja un (o varios) vector frente a otro

```
>> x = -2:1:2;  
>> y = [4 1 0 3 -1];  
>> plot (x,y)
```



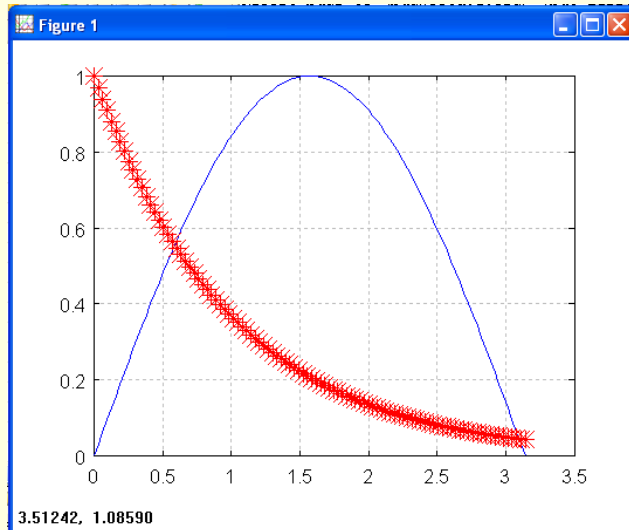
Representación gráfica (plot)

- plot
 - title('título'): añade el título
 - xlabel('etiqueta'): añade etiqueta en eje x
 - Ylabel: funciona igual. ylabel off lo elimina
 - text (x,y,'texto'): introduce la cadena 'texto' en las coordenadas x,y
 - legend(): define la leyenda de la figura
 - grid: activa la rejilla
 - hold on/off: activa/desactiva la incorporación de nuevas gráficas
 - Estilo de líneas:

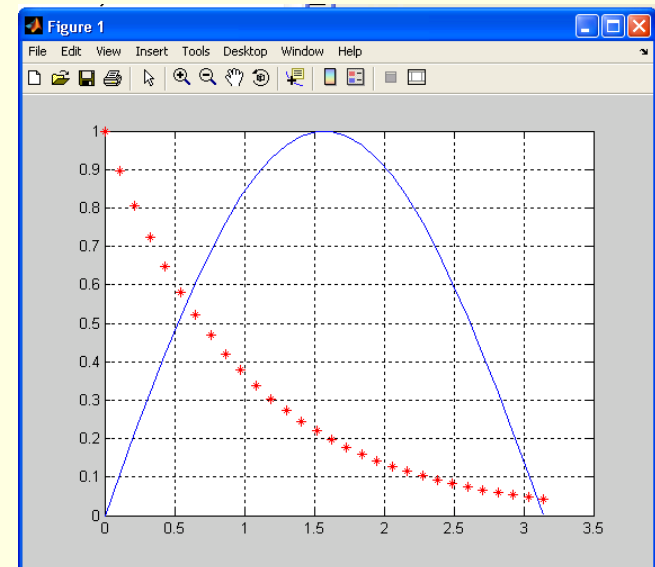
COLOR	MARKER TYPE	LINE STYLE
y yellow	. point	- solid
m magenta	o circle	: dotted
c cyan	x x-mark	-. dashdot
r red	+ plus	-- dashed
g green	* star	
b blue	s square	
w white	d diamond	
k black	h hexagram	
	p pentagram	
	v triangle (down)	
	> triangle (right)	
	< triangle (left)	
	^ triangle (up)	

Representación gráfica (plot)

```
>> x = linspace(0,pi,30);  
>> y = sin(x);  
>> plot(x,y);  
>> grid on;  
>> hold on;  
>> plot(x,exp(-x),'r*')
```



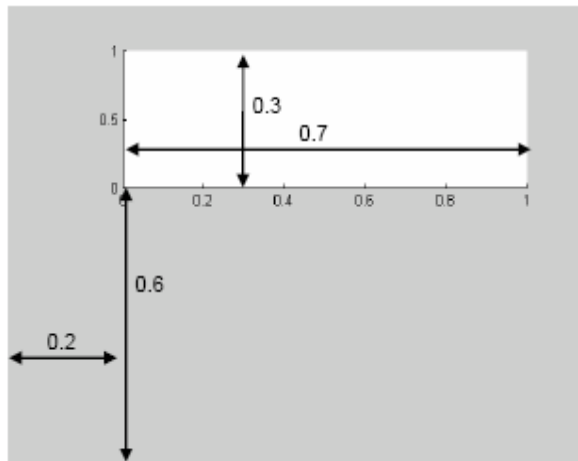
matlab



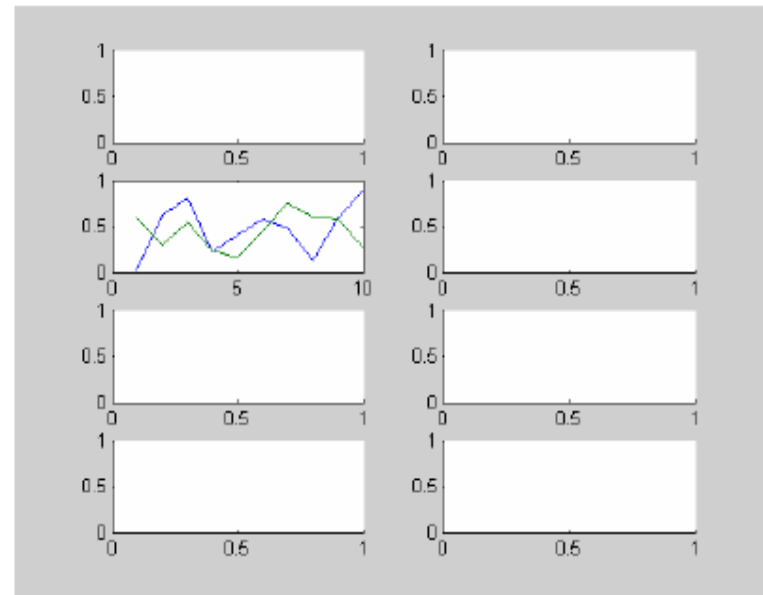
GUIoctave

Representación gráfica (subplot)












```
subplot(4,2,3)  
plot(rand(10,2))
```



```
subplot('position',[0.2,0.6,0.7,0.3])
```



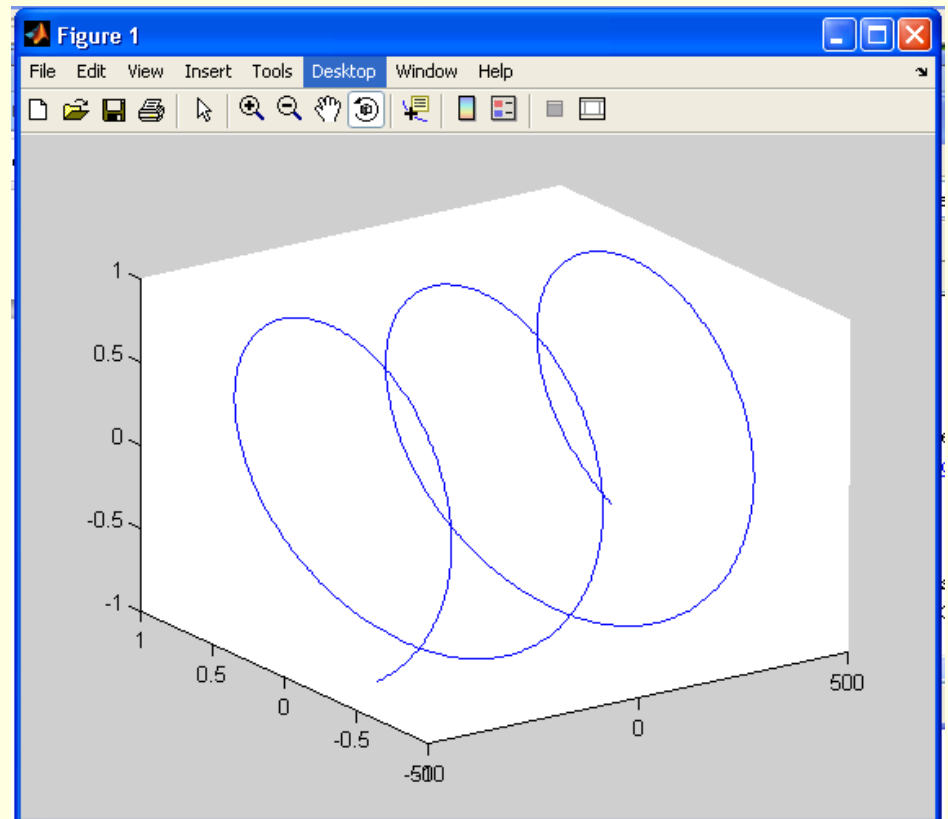
Representación gráfica 2D

Orden	¿Qué hace?	Imagen
area	colorea el area bajo la gráfica	
bar	diagrama de barras (verticales)	
barh	diagrama de barras (horizontales)	
hist	histograma	
pie	sectores	
rose	histograma polar	
stairs	gráfico de escalera	
stem	secuencia de datos discretos	
loglog	como plot pero con escala logarítmica en ambos ejes	
semilogx	como plot pero escala logarítmica en el eje x	
semilogy	como plot pero escala logarítmica en el eje y	

Representación gráfica (plot3)

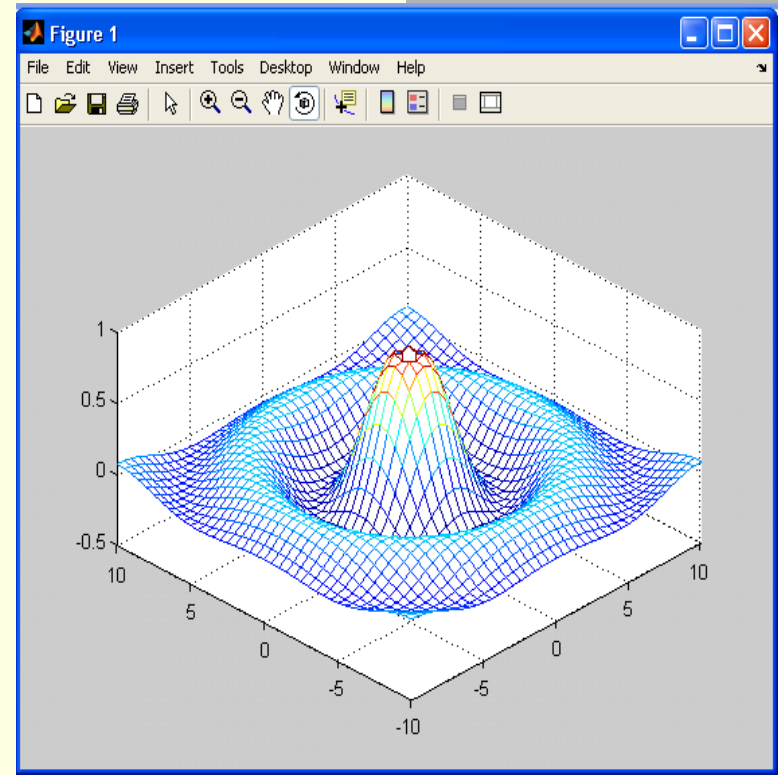
- Extensión del plot 2D a 3D

```
>> %Muelle  
>> x = -500:500;  
>> y = sind(x);  
>> z = cosd(x);  
>> plot3(x,y,z)
```



Representación gráfica (mesh)

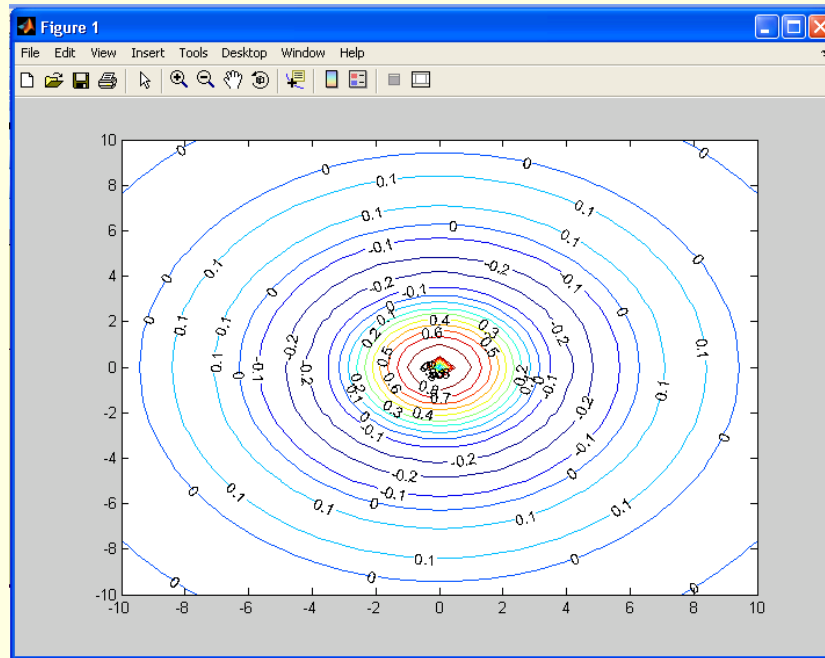
- Visualización de una malla
 - meshgrid crea las matrices malla
 - ... y mesh representa gráficamente la malla



```
>> x = -10:0.5:10; y = -10:0.5:10;  
>> [X,Y] = meshgrid (x,y); % crea matrices para hacer la malla  
>> Z = sin (sqrt (X .^2 + Y .^2)) ./ sqrt (X .^ 2 + Y .^ 2 + 0.1);  
>> mesh (X,Y,Z) % dibuja la gráfica
```


Representación gráfica (contour)

■ Líneas de contorno



```
>> x = -10:0.5:10; y = -10:0.5:10;  
>> [X,Y] = meshgrid (x,y); % crea matrices para hacer la malla  
>> Z = sin (sqrt (X .^2 + Y .^2)) ./ sqrt (X .^ 2 + Y .^ 2 + 0.1);  
>> [C,h] = contour (X,Y,Z);  
>> clabel (C,h) % incluye los valores de las isolíneas
```

Entrada/Salida

- Matlab/octave permite guardar y cargar datos de los archivos del computador
 - save: guarda las variables y su contenido del workspace
 - Sintaxis: `save nombre_fichero [variables ...]`
 - load: carga las variables y su contenido de alguna sesión anterior en el workspace
 - Sintaxis: `load nombre_fichero [variables ...]`

Entrada/salida

- Por defecto MATLAB guarda los datos en un fichero binario
 - No se puede visualizar con un editor
 - -ascii: permite guardar los datos visualizables en un editor

```
>> save guardar_miworkspace.mat -ascii X Y Z
```

- Existen funciones *fopen*, *fclose*, *fprintf*, *fscanf*, *fgets*,... similares a la funciones de C/C++ para el manejo de ficheros

Entrada/salida

- También permite comunicarse a través de la pantalla/teclado
 - `disp('cadena')`: muestra una en pantalla
 - `input('cadena')`: devuelve un dato recogido por teclado
 - `fprintf/fscanf`: similares a los de C/C++

```
>> disp ('Hola Mundo!!');  
Hola Mundo!!  
>> a = input('introduce un numero ');  
Introduce un numero: 3  
>> a  
  
3  
>> fprintf (1,'el numero a=%i\n', a);  
El numero a=3
```