

---

# Introducción a MATLAB

Modelado, Optimización y Simulación  
Germán Montoya, [jermaintz@gmail.com](mailto:jermaintz@gmail.com)

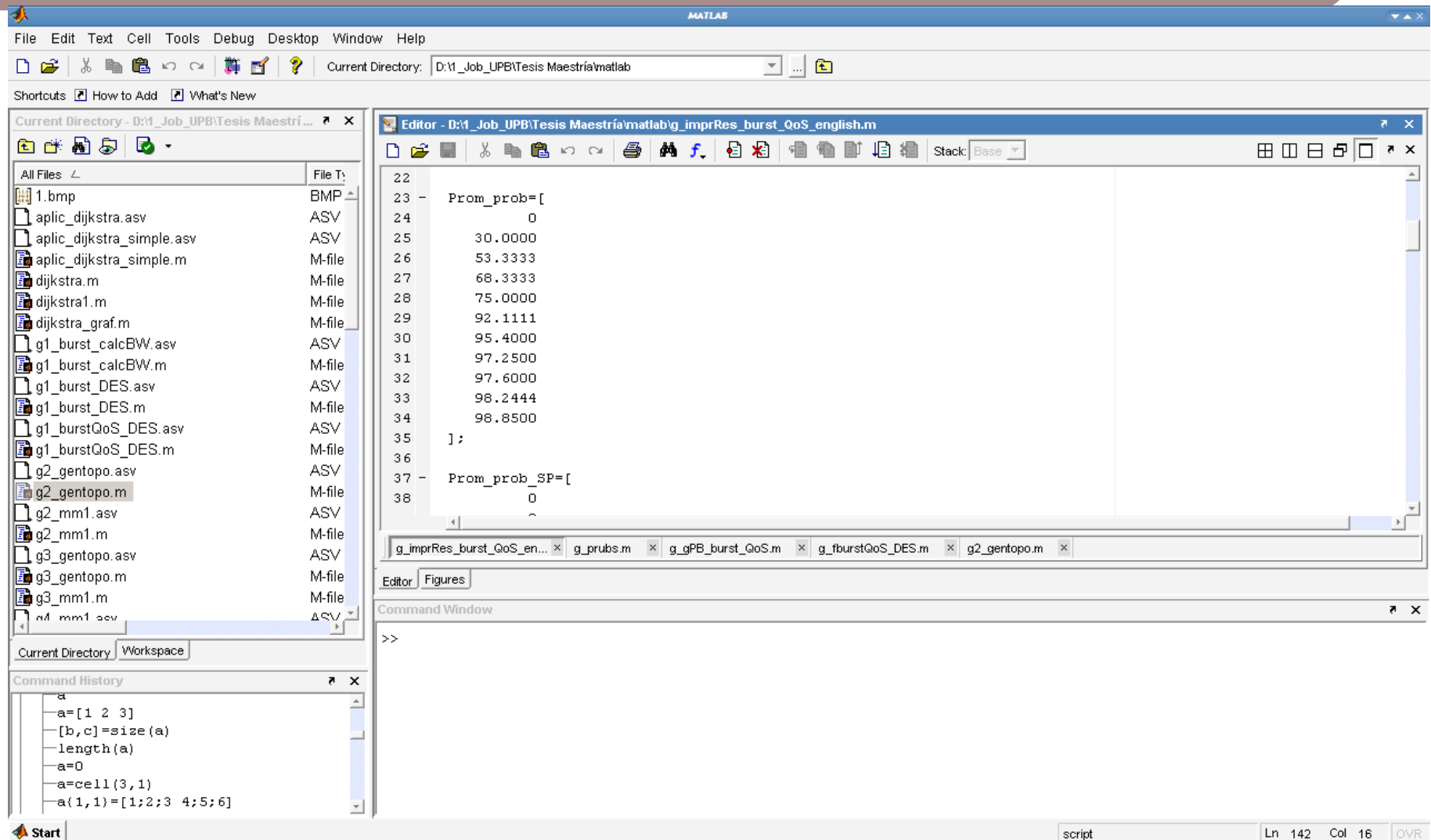
---

*Universidad de Los Andes  
Bogotá, Colombia*

# Índice

- Interfaz
- Números y Operaciones
- Vectores y Matrices
- Operaciones con Vectores y Matrices
- Funciones para Vectores y Matrices
- Gráficos 2D
- Programación
- Debugger

# Interfaz General



# Interfaz: workspace

The screenshot displays the MATLAB workspace and array editor interface. The workspace window on the left lists variables and their values, while the array editor on the right shows the data for the selected variable, Prom\_prob.

**Workspace**

Name	Value
Prom_prob	<11x1 double>
Prom_prob_CP	<11x1 double>
Prom_prob_SP	<11x1 double>
Prom_prob_SP...	<11x1 double>
anchol	1.5
color	'w'
color1	'r'
color2	'b'
color3	'g'
color4	'm'
h	769
linea1	'-r'
linea2	'-bs'
linea3	'-g>'
linea4	'-mo'
marca	6
n_paq	<11x1 double>
texto1	'Implementation without pr...
texto2	'Implementation with priori...
texto3	'Implementation with priori...
texto4	'Implementation with priority'

**Array Editor - Prom\_prob**

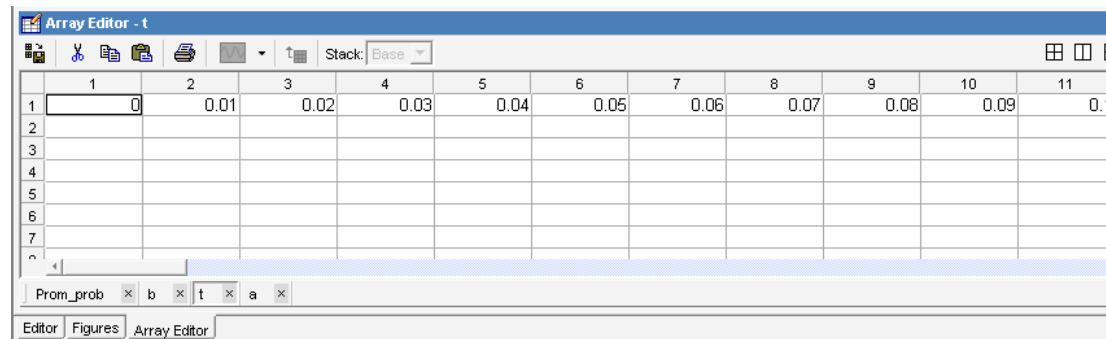
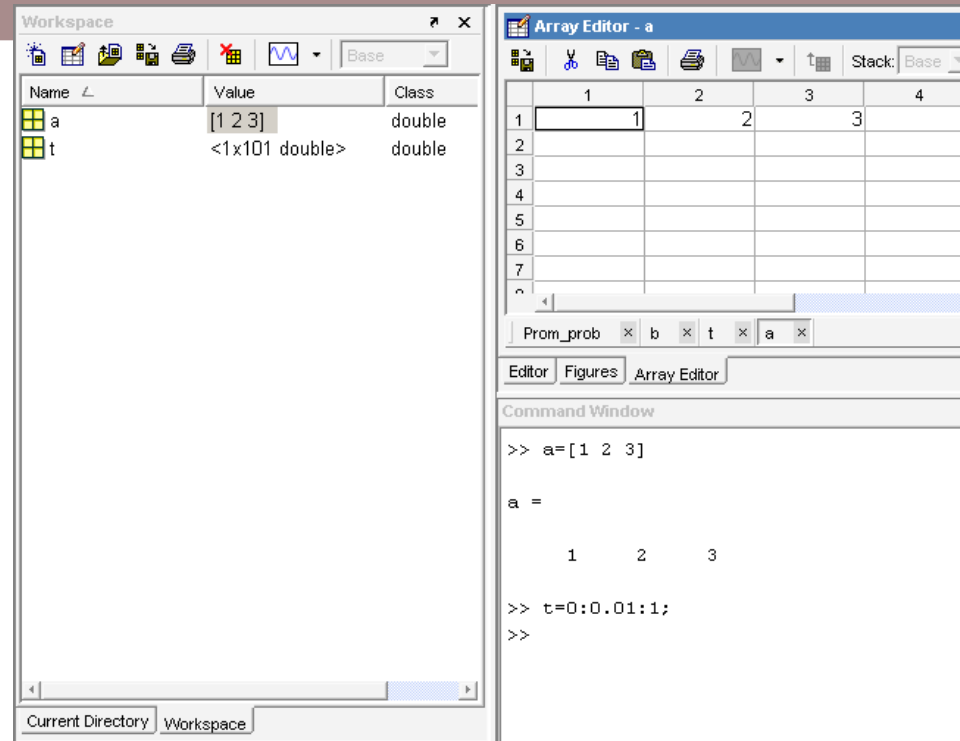
	1	2
1	0	
2	30	
3	53.333	
4	68.333	
5	75	
6	92.111	
7	95.4	
8	97.25	
9	97.6	
10	98.244	
11	98.85	
12		
13		
14		
15		
16		
17		
18		
19		

Editor Figures Array Editor

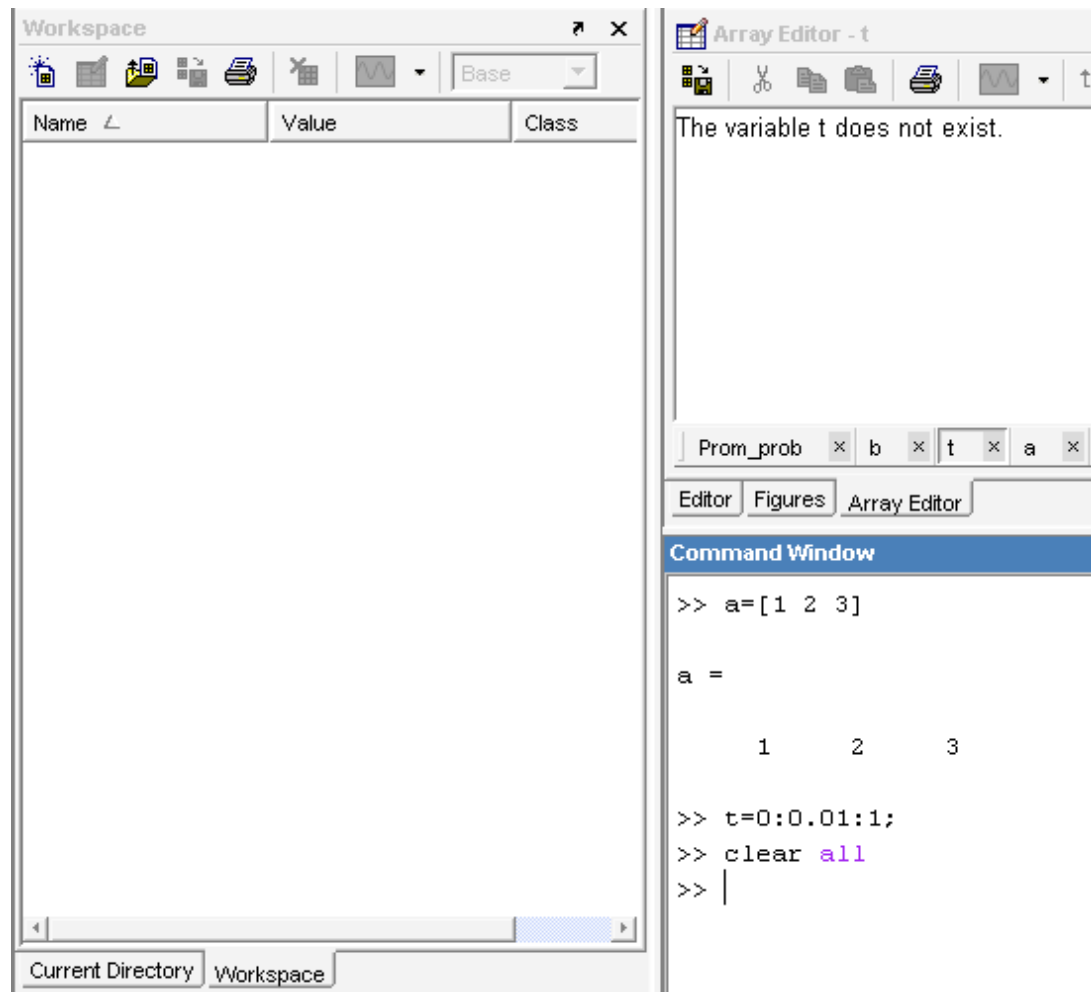
Command Window

>>

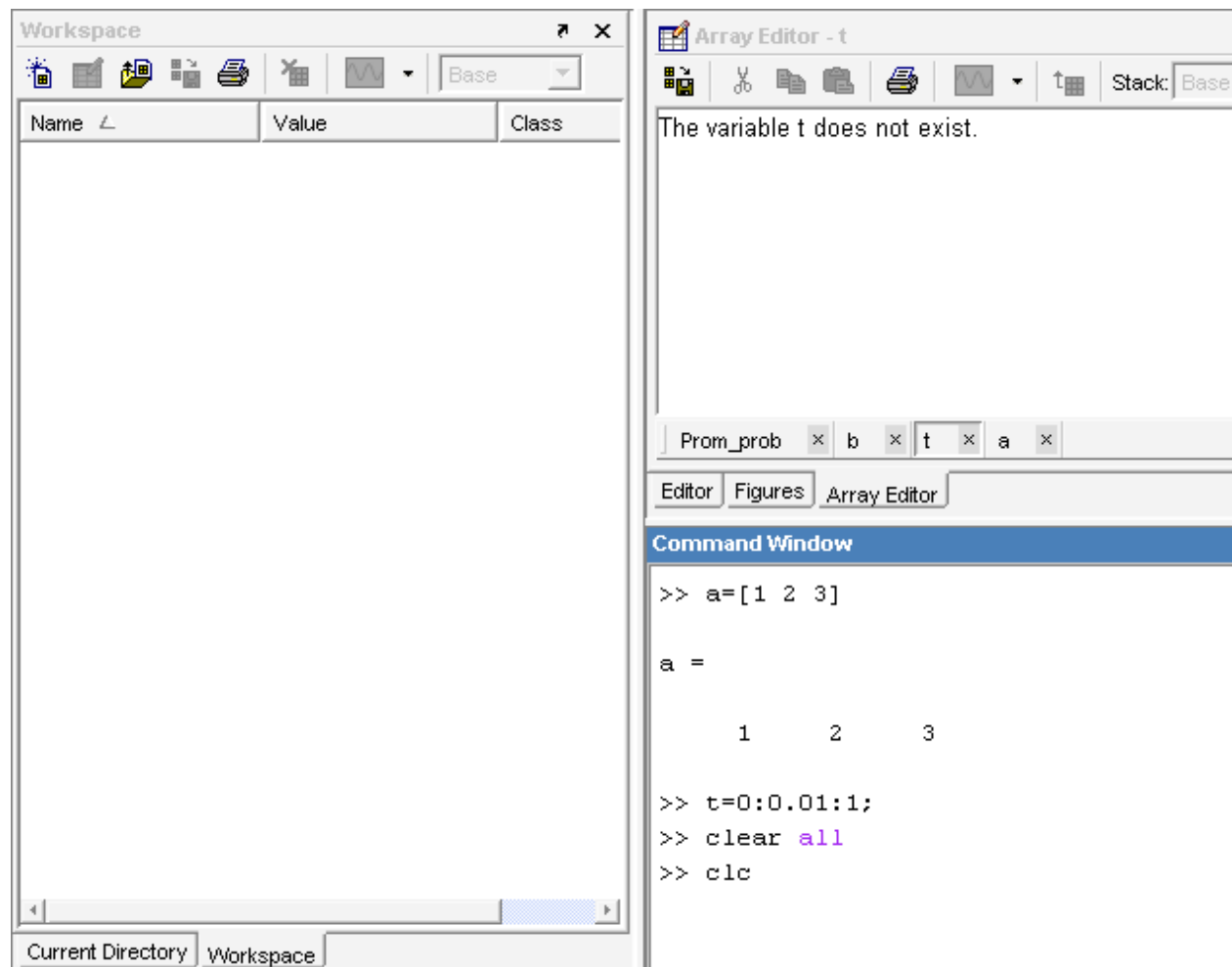
# Interfaz: command window



# Interfaz: borrado de variables



# Interfaz: "limpiar" command window



# Interfaz: command history

The image displays the MATLAB interface with three main windows:

- Workspace:** A table showing the current workspace variables.
- Array Editor - t:** A window for editing the array 't'.
- Command History:** A window showing the sequence of commands entered in the Command Window.

**Workspace Table:**

Name	Value	Class
t	<1x101 double>	double

**Array Editor - t:** A table showing the values of the array 't'.

	1	2	3
1	0	0.01	0.02
2			
3			
4			
5			
6			
7			

**Command History:**

```
clear all
clc
a=[1 2 3]
t=0:0.01:1;
clear all
clc
t=0:0.01:1;
clc
a=[1 2 3]
t=0:0.01:1;
clear all
```



# Números

- No hace falta definir variables enteras, reales, etc. como en otros lenguajes
  - Números enteros:  $a=2$
  - Números reales:  $x=-35.2$
  - Números potencias de 10:  $x=3e5 = 300000$  ó  $3*10^5 = 300000$
  - Números Imaginarios:  $x= 1 + 2i = 1.0000 + 2.0000i$
  - Euler:  $\exp(1) = 2.7182$
  - $\pi$ :  $pi = 3.1416$
  - Infinito:  $Inf=\infty$ .

# Matrices

- **Vector fila**; elementos separados por espacios en blanco o comas

```
>> v = [2 3 4]
```

- **Vector columna**: elementos separados por **punto y coma (;)**

```
>> w = [2;3;4;7;9;8]
```

- **Generación de vectores fila**:

- Especificando el incremento **h** de sus componentes  
**v=a:h:b**

- T=0:0.1:1

	1	2	3	4	5	6	7	8	9	10	11
1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

- Especificando su dimensión **n**: **linspace(a,b,n)**  
(por defecto n=100)

- T=linspace(0,1,11)

	1	2	3	4	5	6	7	8	9	10	11
1	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

# Matrices

- Matrices  $n \times m$ :
  - $A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$
  - Información:
    - De un elemento:  $A(2,3) = 6$
    - De una fila:  $A(1,:) = 1 \ 2 \ 3$
    - De una columna:  $A(:,2) = 2 \ 5 \ 8$
  - Matriz vacía:  $A = []$ 
    - Eliminación de una fila:  $A(1,:) = []$
    - Eliminación de una columna:  $A(:,2) = []$

# Matrices: Operadores

- Básicas: +, -, \*, /
  - $A=[1\ 2\ 3\ 4\ 5]; k=2; B=[5\ 4\ 3\ 2\ 1]$
  - $A/2 = 0.5\ 1\ 1.5\ 2\ 2.5;$
  - $2./A = 2\ 1\ 0.6667\ 0.5\ 0.4;$
  - $A./B = 0.2\ 0.5\ 1\ 2\ 5;$

# Matrices

- Almacenamiento en vectores:

```
> > a=[]
```

```
> > a=[a 1]
```

```
> > a=[a 2]
```

- Supresión en vectores:

```
> > a=[1 2 3 4 5]
```

```
> > a=a(2:length(a)); %recorto el primer  
elemento del vector.
```

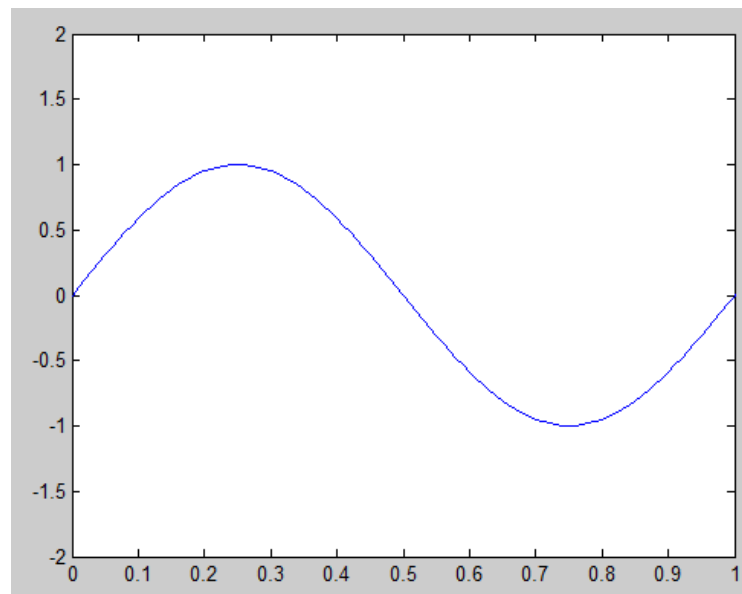
Como recorto el último??

# Funciones

- $A=[1\ 2\ 3\ 4\ 5]$ ;  $B=[1\ 2\ 3;4\ 5\ 6]$ ;
  - $\text{Length}(A)=5$
  - $[n,m]=\text{size}(B) \rightarrow n=2\ \text{y}\ m=3$
  - $\text{sum}(A)=15$ ;  $\text{prod}(A)=120$
  - $\text{Max}(A)=5$ ;  $\text{min}(A)=1$
  - $A'=[1;2;3;4;5]$
  - $\text{Zeros}(n,m)$ ;  $\text{zeros}(1,2)=[0\ 0]$
  - $\text{Ones}(n,m)$ ;  $\text{ones}(1,2)=[1\ 1]$

# Gráficos

- `T=0:0.1:1`
- `F=2`
- `X=sin(2*pi*F*T); %señal senoidal`
- `Figure`
- `Plot(t,x); %curva azul (por defecto)`



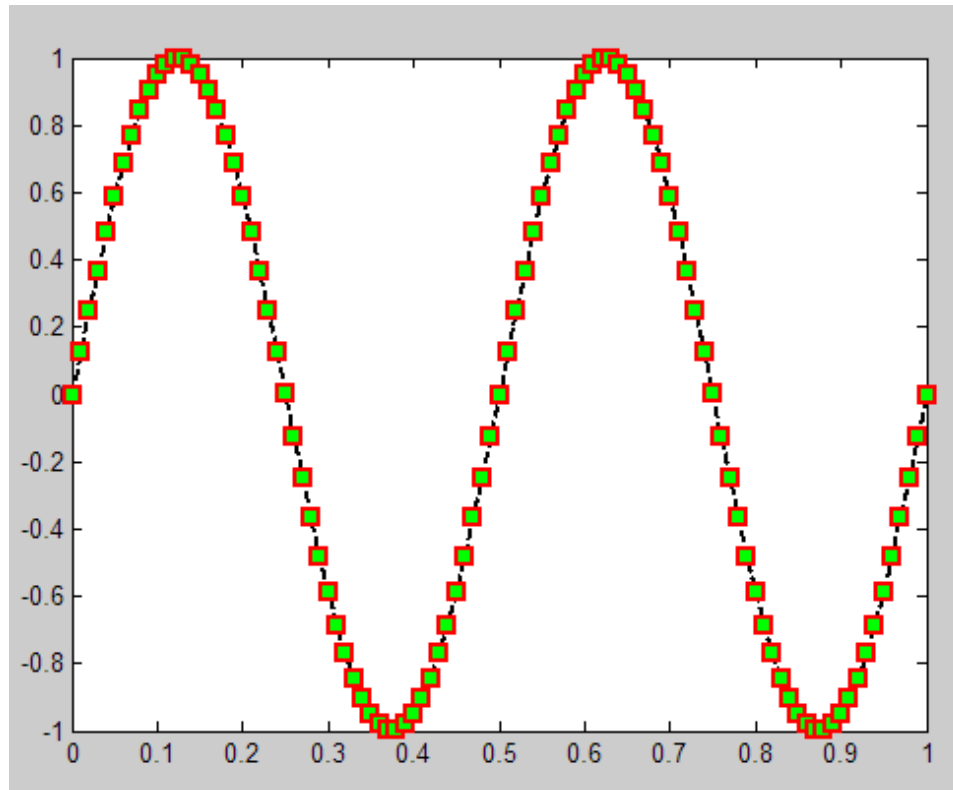
# Gráficos

- `Plot(t,x); %curva azul (por defecto)`
- `plot(t,x,'k');` %curva negra
- `plot(t,x,'--k');` %curva líneas espaciadas
- `plot(t,x,'--ks');` %figura en el valor grafico. Otros iconos en los puntos: d,h,o,p,s,x.
- `plot(t,x,'--ks','LineWidth',2);` %ancho de la linea
- `plot(t,x,'--ks', 'MarkerEdgeColor','r', ...  
'MarkerFaceColor','g')`



# Gráficos

- `plot(t,x,'--ks',  
    'LineWidth',2,'MarkerEdgeColor','r','MarkerFaceColor','g');`



# Gráficos: error común

- Hacer 2 graficas en una misma figura:

```
>> figure
```

```
>> plot(t,x)
```

```
>> plot(t,z) %sobrescribe al anterior plot
```

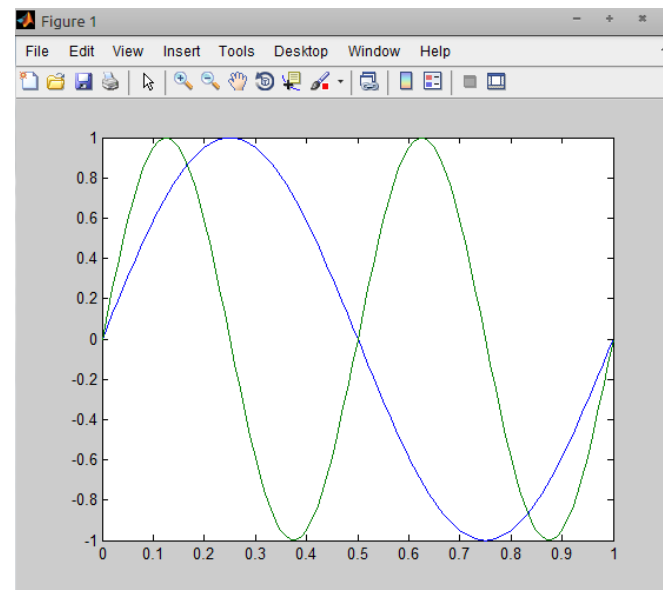
- Solución 1:

```
>> figure
```

```
>> hold on
```

```
>> plot(t,x)
```

```
>> plot(t,z)
```



# Gráficos: error común

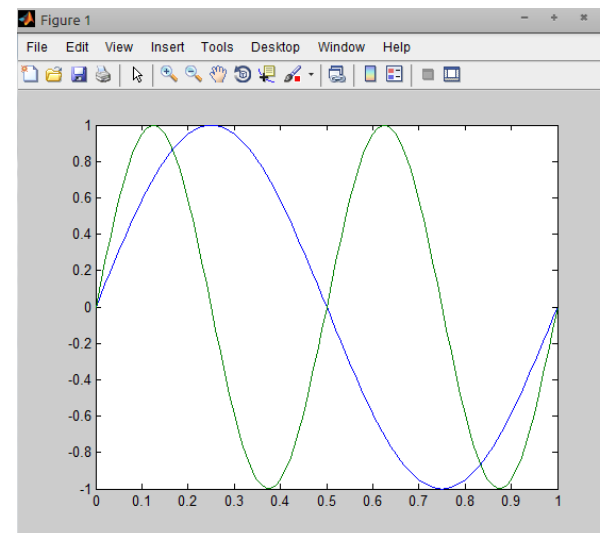
- Solución 2:

```
>> figure
```

```
>> plot(t,x,t,z)
```

```
>> figure
```

```
>> plot(t,x,'--ks',t,z,'o','LineWidth',2)
```



# Gráficos: estructura plot

>> Figure

>> Plot(t,x) %primera gráfica

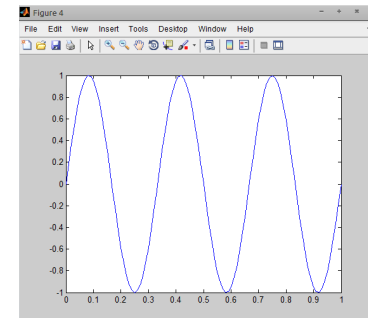
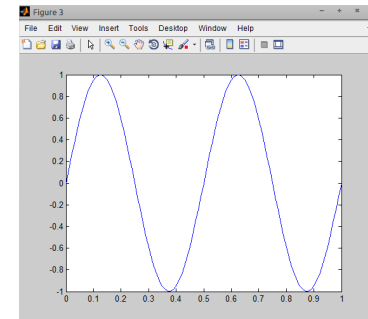
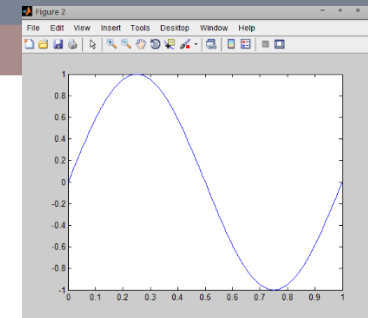
>> Figure

>> Plot(t,z) %segunda gráfica

•

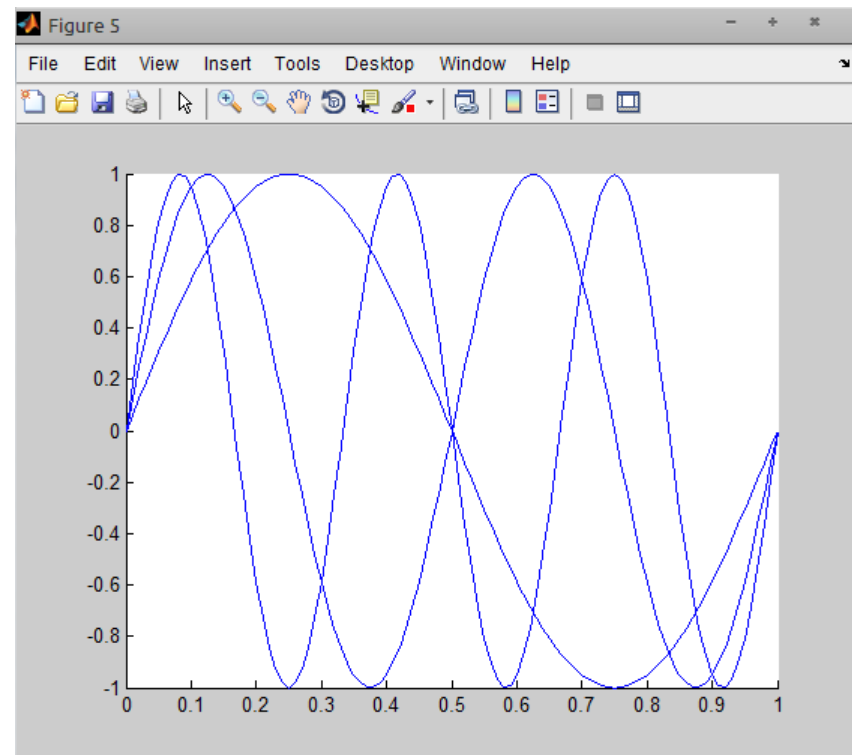
•

•



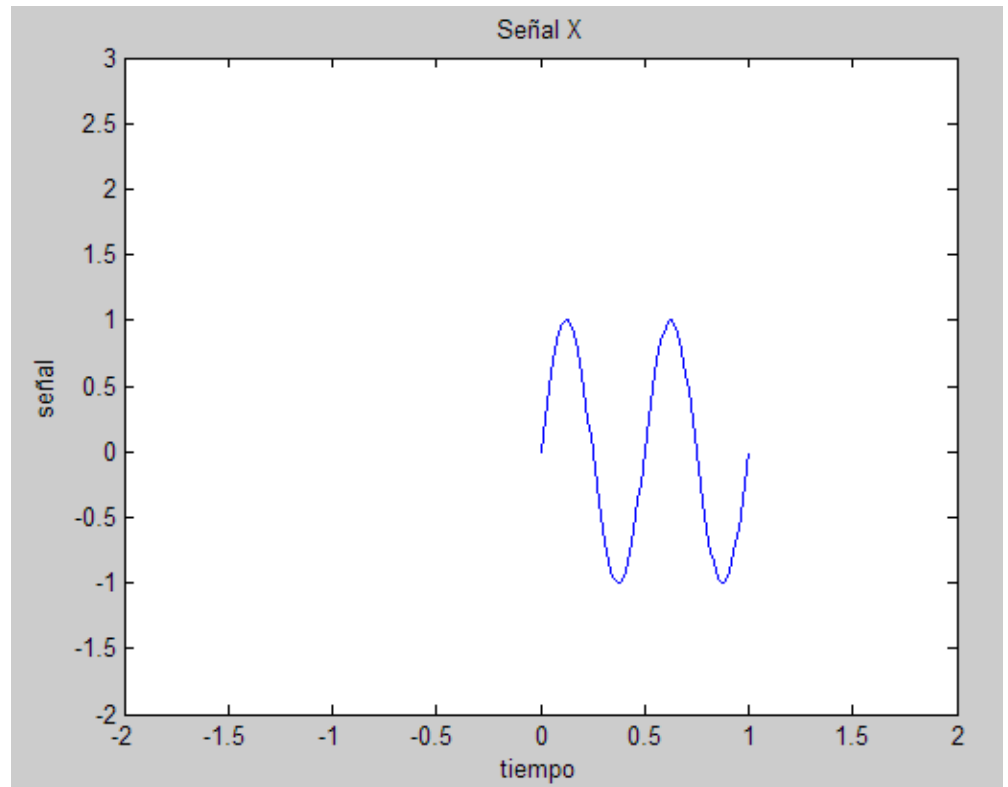
# Gráficos: estructura plot

```
>> Figure  
>> hold on  
>> plot(t,x)  
>> plot(t,y)  
.  
.  
.  
>> plot(t,z)
```



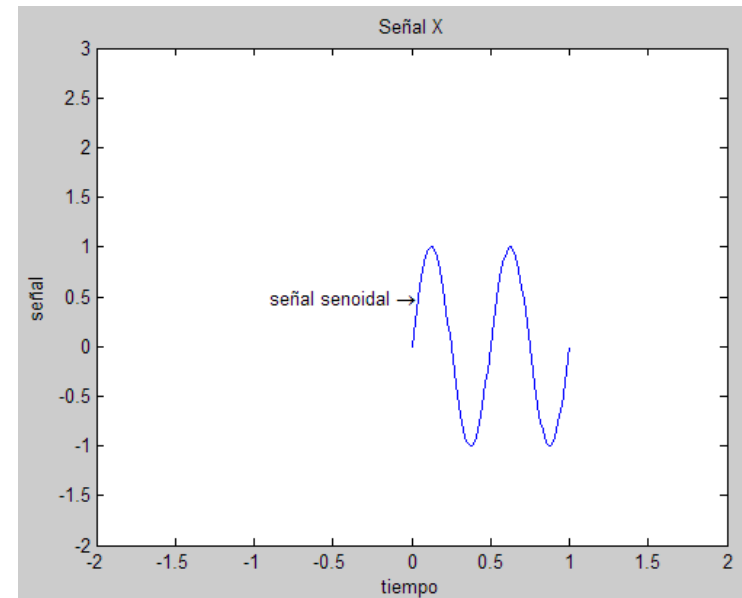
# Gráficos: ejes, abscisa(x) y ordenada(y)

```
>> figure;  
>> plot(t,x);  
>> title('Señal');  
>> xlabel('Tiempo');  
>> ylabel('Señal x');  
>> axis([-2 2 -2 3]);
```



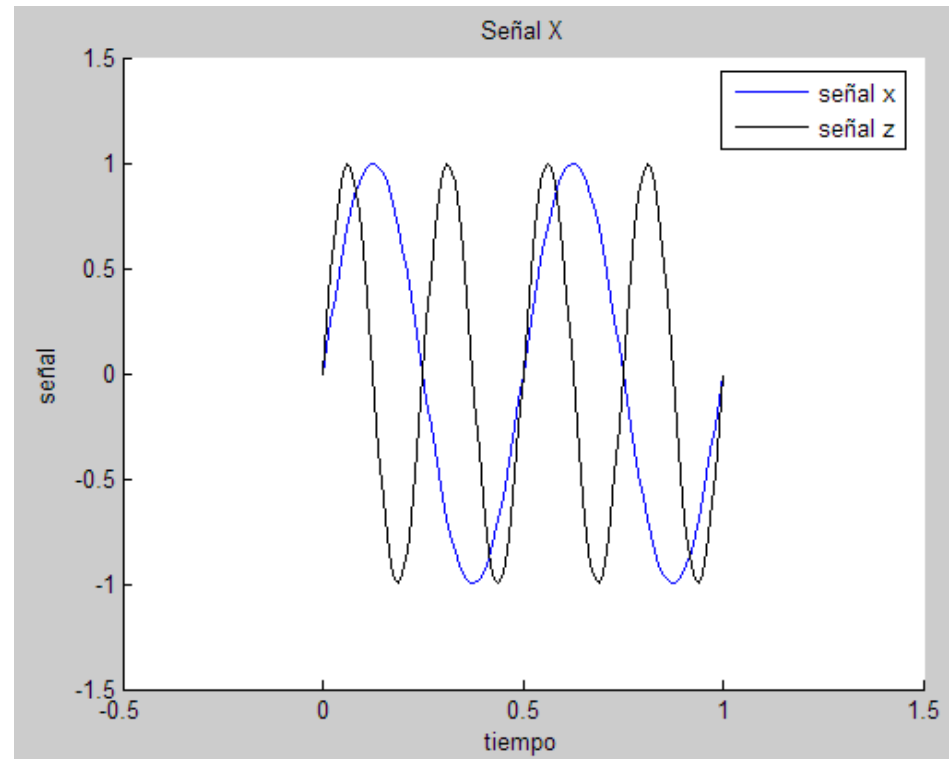
# Gráficos: texto dentro de la figura

```
>> figure;  
>> plot(t,x);  
>> xlabel('tiempo');  
>> ylabel('señal');  
>> axis([-2 2 -2 3]);  
>> text(-0.9,0.5,'señal senoidal  
  \rightarrow','FontSize',10);
```



# Gráficos: etiquetas

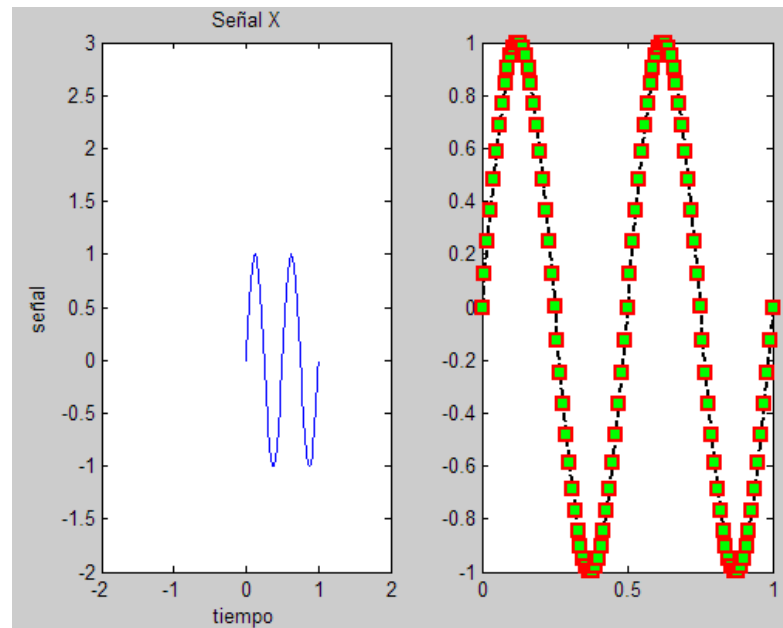
```
>> figure;  
>> hold on;  
>> plot(t,x,'b');  
>> title('Señal X');  
>> xlabel('tiempo');  
>> ylabel('señal');  
>> axis([-0.5 1.5 -1.5 1.5]);  
  
>> plot(t,z,'k');  
  
>> h = legend('señal x','señal z',1);
```





# Gráficos: subplot

```
>> subplot(1,2,1);  
>> plot(t,x);  
>> subplot(1,2,2);  
>> plot(t,x,'--ks',  
        'LineWidth',2,'MarkerEdgeColor','r','MarkerFaceColor','g');
```



# Gráficos: estructura de subplot

```
>> Figure;  
>> Subplot(n,m,i)  
>> Plot(t,x)  
>> Subplot(n,m,j)  
>> Plot(t,z)
```

.

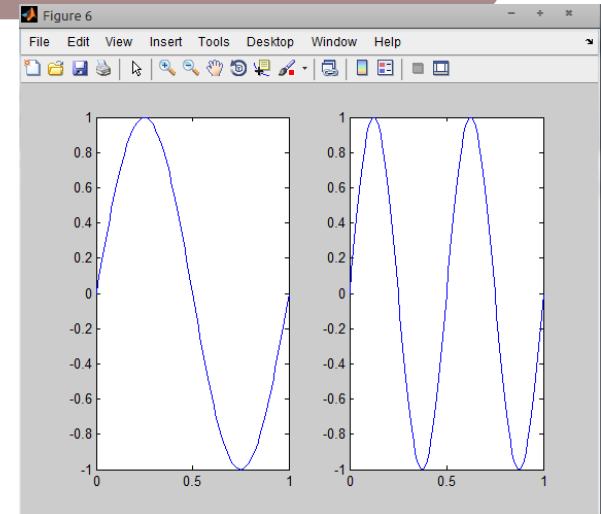
.

```
>> Figure;  
>> Subplot(p,q,i)
```

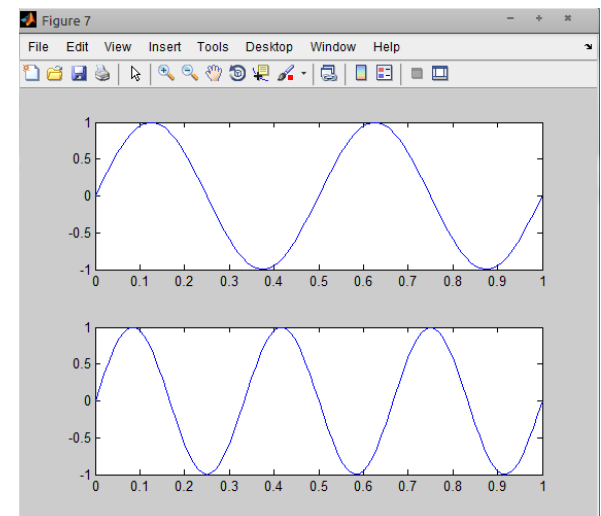
.

.

Primera ventana

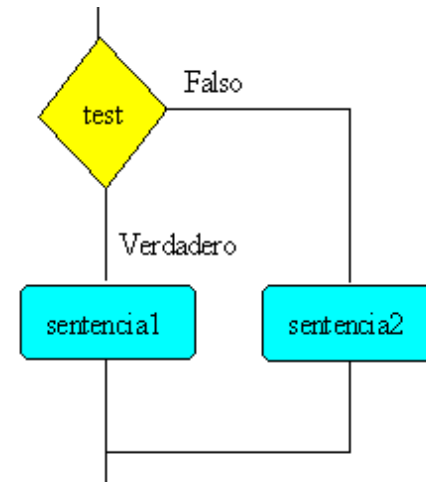
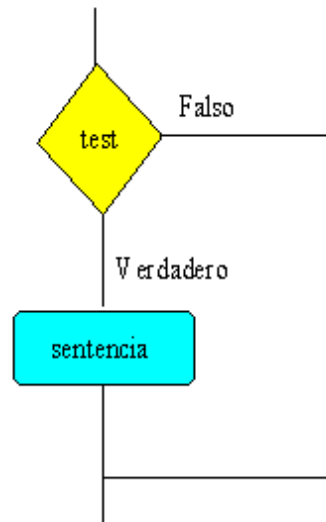


Segunda ventana



# Programación: if

- If condición  
    Sentencia;  
End;
- If condición  
    sentencia 1;  
Else  
    sentencia 2;  
End;

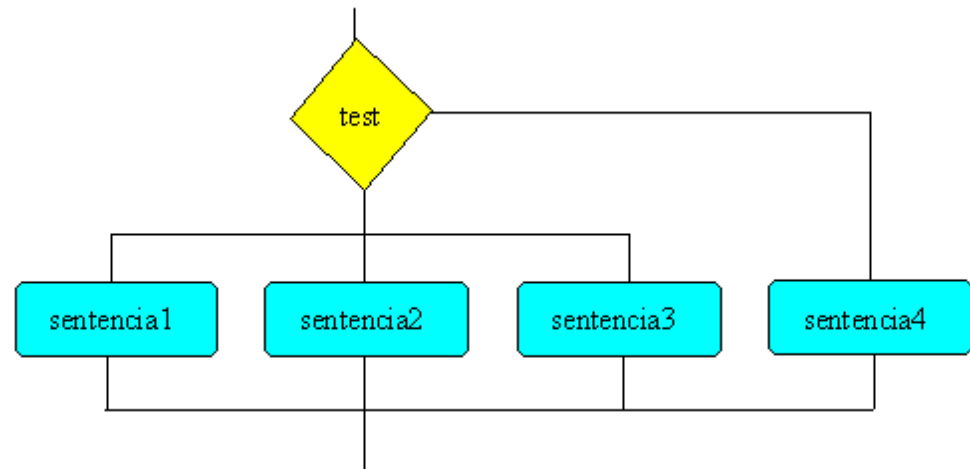


# Operaciones Lógicas

- $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ,  $==$  (igual)
- $|$  (or),  $\&$  (and)
- $\sim$  (no igual)
- If  $a > 0$   
     $a = a * b;$   
end;

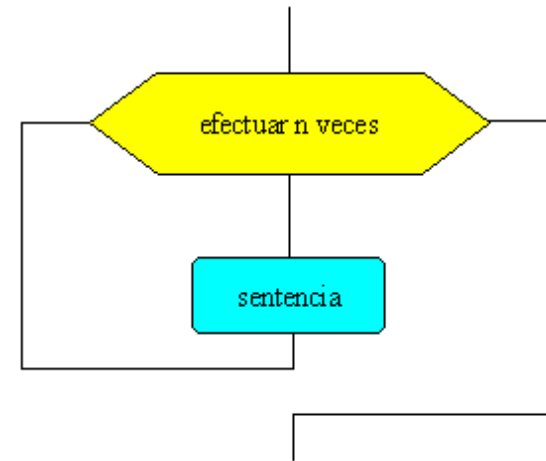
# Programación: if...elseif...else

```
if(a>0)
    sentencia1;
elseif(b>0)
    sentencia2;
elseif(c>0)
    sentencia3;
else
    sentencia4;
End;
```



# Programación: for

- for variable = expresión  
    sentencias  
End
- For i=1:3  
    a(i)=10;  
end

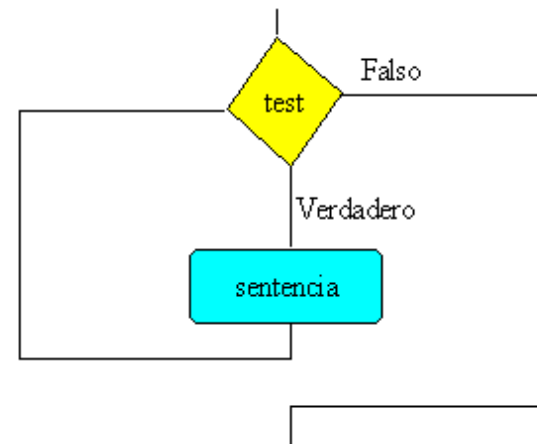


# Programación: while

- While expresión  
sentencias

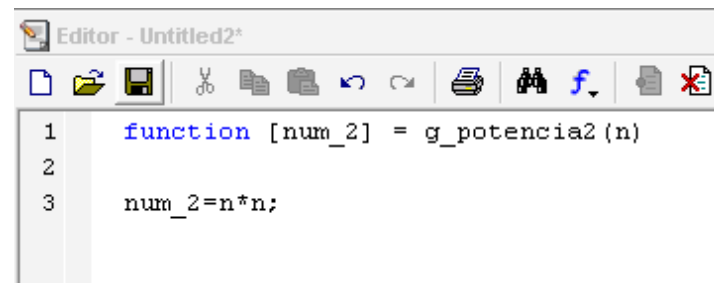
End

- While  $a == 0$   
     $a = a + 1;$   
end

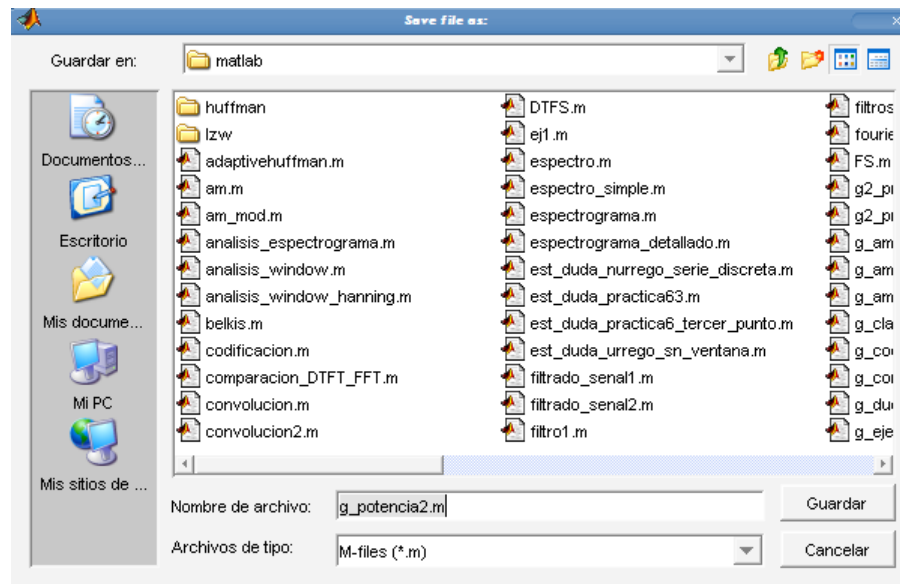


# Creación de Funciones

- M-file de la función "elevar al cuadrado":



```
1 function [num_2] = g_potencia2(n)
2
3 num_2=n*n;
```





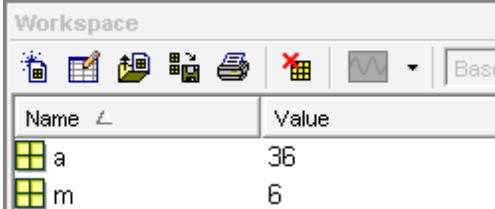
# Creación de Funciones

- M-file de la función:

```
function [num_2] = g_potencia2(n)  
num_2=n*n;
```

- M-file que usa la función:

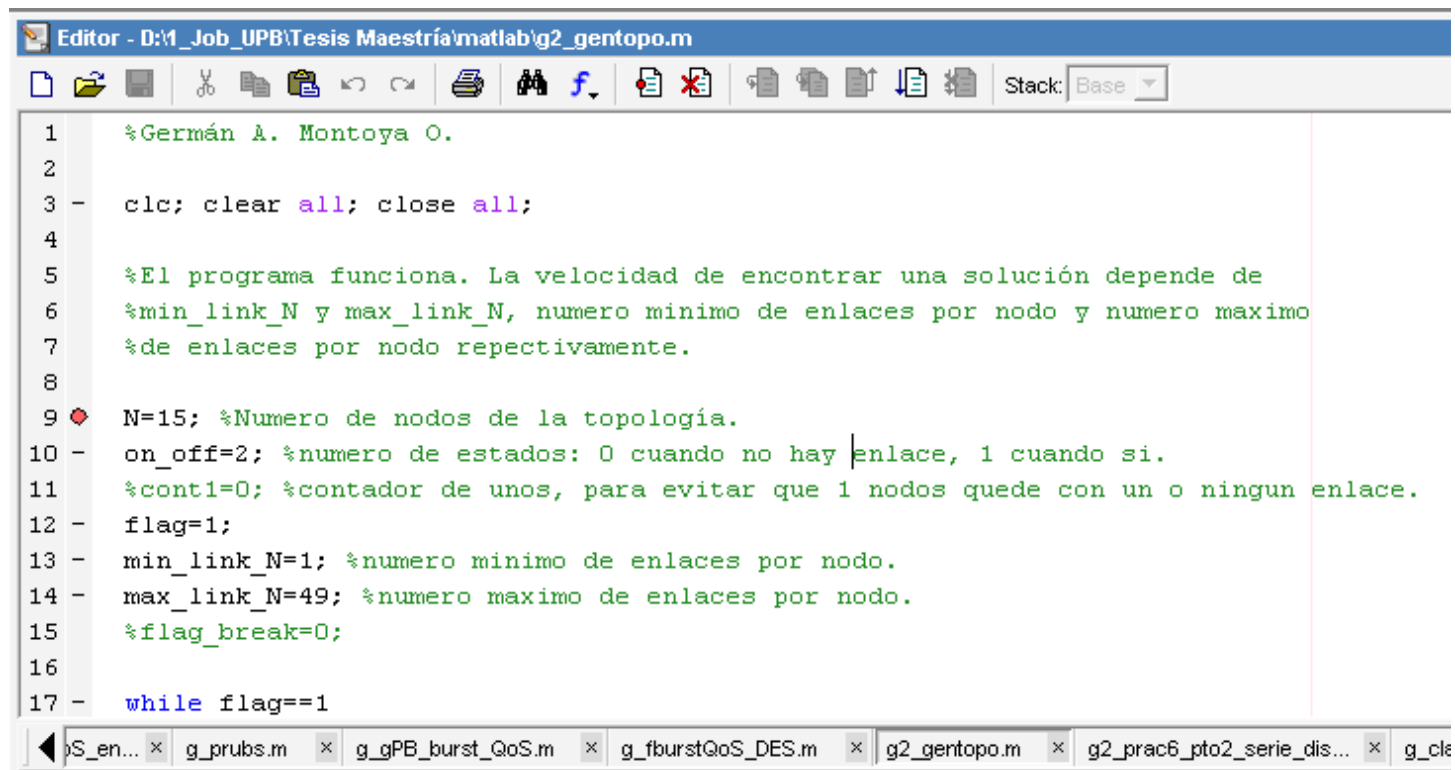
```
clc  
clear all;  
close all;  
m=6;  
a=g_potencia2(m);
```



The image shows a screenshot of the MATLAB Workspace window. It displays two variables: 'a' with a value of 36 and 'm' with a value of 6. The window has a toolbar with icons for saving, deleting, and other workspace actions. The 'Base' workspace is selected.

Name	Value
a	36
m	6

# Debugger



```
Editor - D:\1_Job_UPB\Tesis Maestría\matlab\g2_gentopo.m

1 %Germán A. Montoya O.
2
3 - clc; clear all; close all;
4
5 %El programa funciona. La velocidad de encontrar una solución depende de
6 %min_link_N y max_link_N, numero minimo de enlaces por nodo y numero maximo
7 %de enlaces por nodo repectivamente.
8
9 N=15; %Numero de nodos de la topología.
10 - on_off=2; %numero de estados: 0 cuando no hay enlace, 1 cuando si.
11 %cont1=0; %contador de unos, para evitar que 1 nodos quede con un o ningun enlace.
12 - flag=1;
13 - min_link_N=1; %numero minimo de enlaces por nodo.
14 - max_link_N=49; %numero maximo de enlaces por nodo.
15 %flag_break=0;
16
17 - while flag==1
```




# Debugger



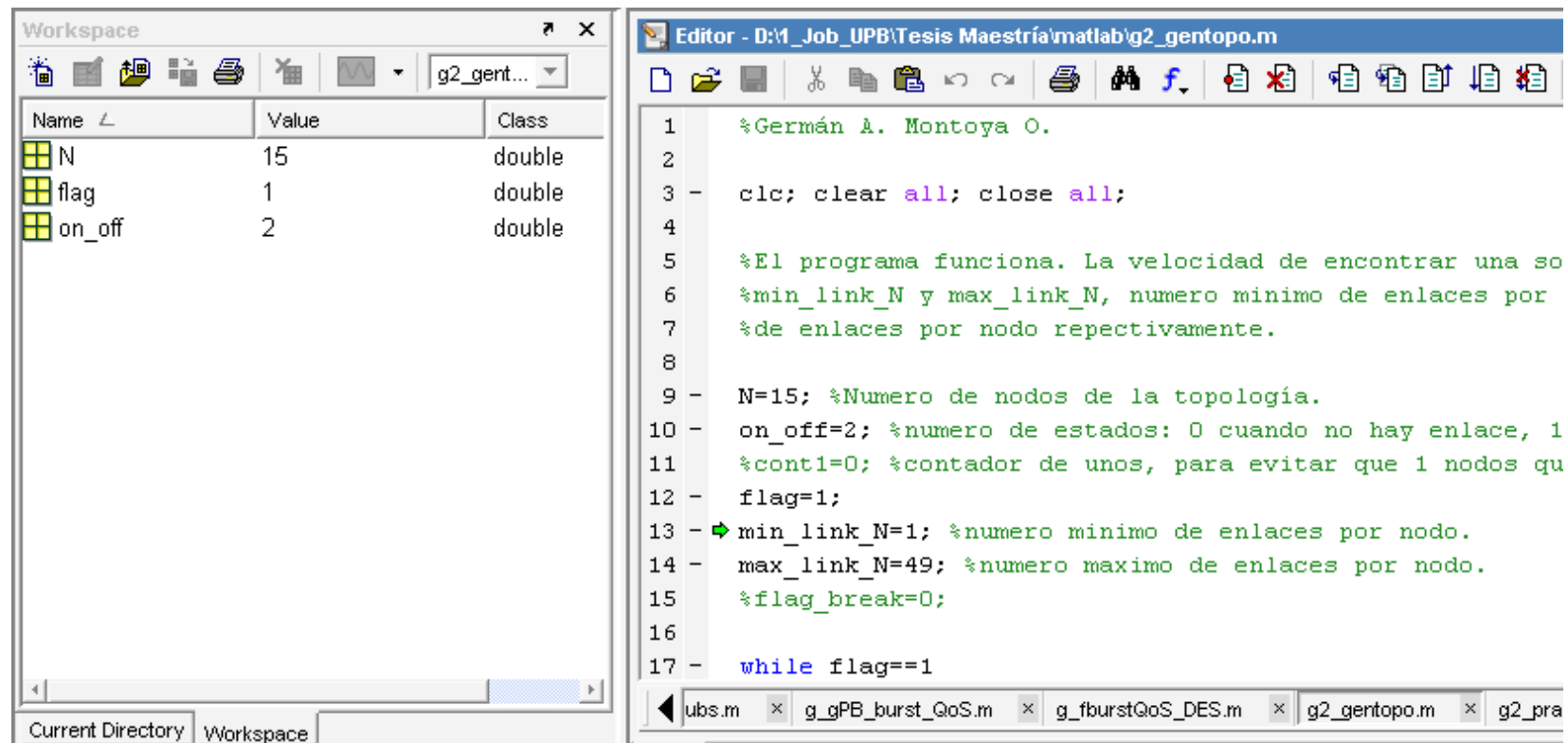
```
1 %Germán A. Montoya O.
2
3 - clc; clear all; close all;
4
5 %El programa funciona. La velocidad de encontrar una solución depende de
6 %min_link_N y max_link_N, numero minimo de enlaces por nodo y numero maximo
7 %de enlaces por nodo repectivamente.
8
9 - ➔ N=15; %Numero de nodos de la topología.
10 - on_off=2; %numero de estados: 0 cuando no hay enlace, 1 cuando si.
11 %cont1=0; %contador de unos, para evitar que 1 nodos quede con un o ningun enlace.
12 - flag=1;
13 - min_link_N=1; %numero minimo de enlaces por nodo.
14 - max_link_N=49; %numero maximo de enlaces por nodo.
15 %flag_break=0;
16
17 - while flag==1
```



# Debugger

- Step:  simulación paso a paso.
- Step In:  simulación paso a paso pero entra a funciones.
- Step Out:  sirve para salir de una función.

# Debugger



The screenshot displays the MATLAB environment with the Workspace and Editor windows open.

**Workspace Window:**

Name	Value	Class
N	15	double
flag	1	double
on_off	2	double

**Editor Window:** D:\Job\_UPB\Tesis Maestría\matlab\g2\_gentopo.m

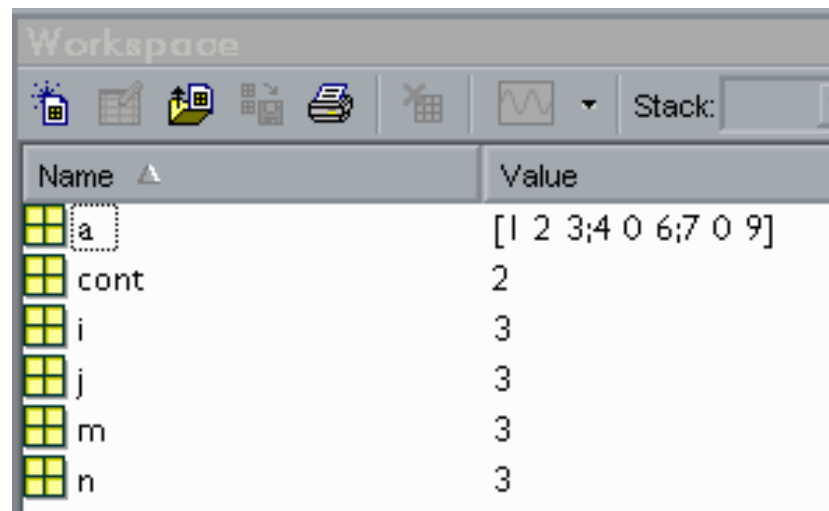
```
1 %Germán A. Montoya O.
2
3 - clc; clear all; close all;
4
5 %El programa funciona. La velocidad de encontrar una so
6 %min_link_N y max_link_N, numero minimo de enlaces por
7 %de enlaces por nodo repectivamente.
8
9 - N=15; %Numero de nodos de la topología.
10 - on_off=2; %numero de estados: 0 cuando no hay enlace, 1
11 %cont1=0; %contador de unos, para evitar que 1 nodos qu
12 - flag=1;
13 - min_link_N=1; %numero minimo de enlaces por nodo.
14 - max_link_N=49; %numero maximo de enlaces por nodo.
15 %flag_break=0;
16
17 - while flag==1
```

The taskbar at the bottom shows several open files: g2\_gentopo.m, g2\_pra, g2\_gPB\_burst\_QoS.m, g\_fburstQoS\_DES.m, and g2\_gentopo.m.

# Ejemplo: contar cuantos ceros tiene la matriz

```
a=[1 2 3;4 0 6;7 0 9];  
[n,m]=size(a);
```

```
cont=0;  
for i=1:n  
    for j=1:m  
        if a(i,j)==0  
            cont=cont+1;  
        end;  
    end;  
end;
```



The image shows a screenshot of the MATLAB Workspace window. It contains a table with two columns: 'Name' and 'Value'. The variables listed are 'a', 'cont', 'i', 'j', 'm', and 'n'. The value for 'a' is the matrix [1 2 3; 4 0 6; 7 0 9]. The value for 'cont' is 2. The values for 'i', 'j', 'm', and 'n' are all 3.

Name	Value
a	[1 2 3; 4 0 6; 7 0 9]
cont	2
i	3
j	3
m	3
n	3

# Tips avanzados

- Función `unidrnd(k)`:
  - Genera un número aleatorio entre 1 y  $k$ .
  - Cómo sería un número aleatorio entre 0 y  $k$ ?

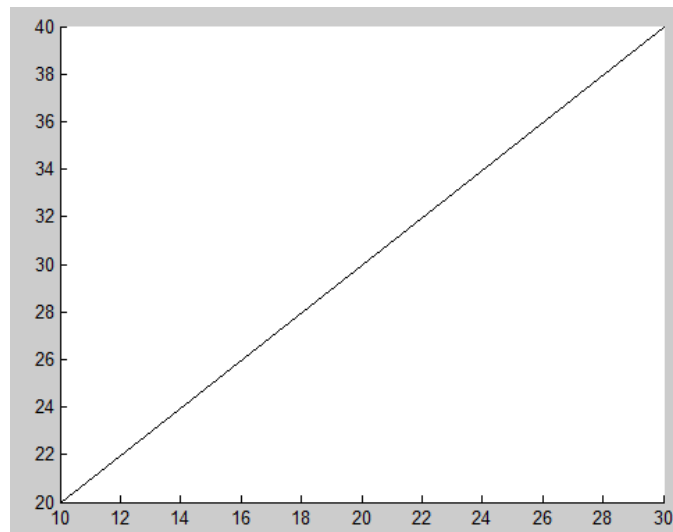
# Tips avanzados

- Función unidrnd(k):
  - Genera un número aleatorio entre 1 y k.
  - Cómo sería un número aleatorio entre 0 y k?
    - $\text{Unidrnd}(k+1) - 1$
- Estructuras:
  - $N(1).\text{posicionX}=20; N(1).\text{posicionY}=10;$
  - $N(1).\text{energia}=87;$
  - $N(2).\text{posicionX}=20; N(2).\text{posicionY}=10;$
  - $N(2).\text{energia}=87;$



# Tips avanzados

- Función line:
  - P1:  $x_1=10$ ;  $y_1=20$
  - P2:  $x_2=30$ ;  $y_2=40$ 
    - `Line([x1,x2],[y1 y2], 'LineStyle','-', 'Color','k', 'LineWidth',1)`

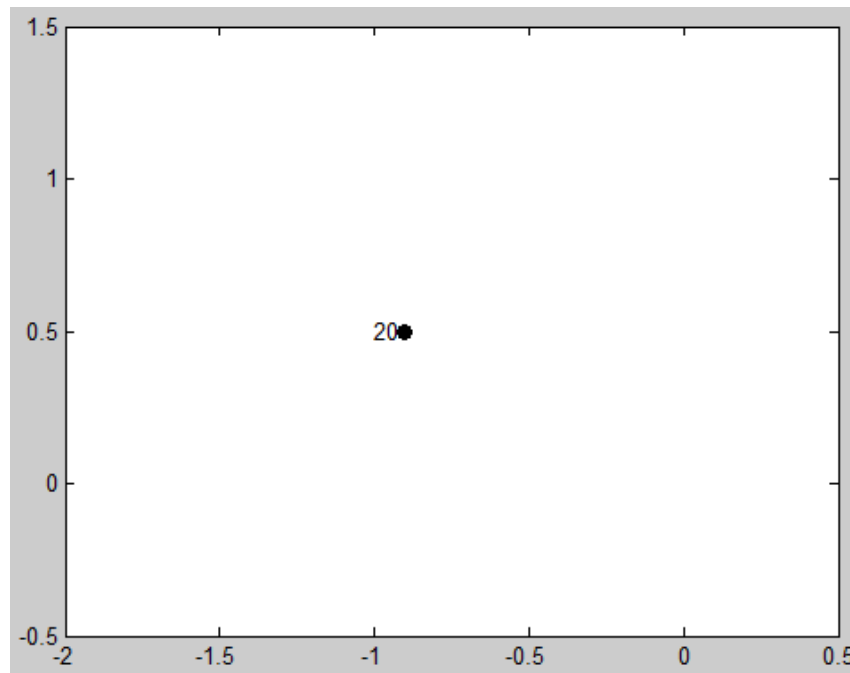


# Tips avanzados

- Función num2str:

```
plot(-0.9,0.5,'o', 'LineWidth',1,'MarkerEdgeColor','k', 'MarkerFaceColor','k',  
'MarkerSize',7);
```

```
text(-1,0.5,num2str(20),'FontSize',10);
```



# Tips avanzados

- Datos de entrada en medio de la ejecución de un m-file:

```
A=input('Guardar Red? 1-Si, 0-No: ')
```

```
If A==1
```

```
    Guardo Red;
```

```
end
```

# Tips avanzados

- Guardar datos en un archivo de excel:

```
A=[1 2 3 4 5];
```

```
xlswrite('valoresA.xls', A , 'Sheet1');
```

```
A=[1 2 3 4 5]; B=[5 4 3 2 1];
```

```
C=[A; B];
```

```
xlswrite('valoresC.xls', C , 'Sheet1');
```

	A	B	C	D	E
1	1	2	3	4	5
2	5	4	3	2	1

# Tips avanzados

- Leer datos de un archivo de excel:  
`C=xlsread(valoresC.xls', 'Sheet1');`
- Función `fprintf`:
  - Imprimir una frase: `fprintf('Hola mundo! \n');`
  - Imprimir el valor de una variable:  
`fprintf('El valor de A es: %f \n',A);`  
`fprintf('El valor de A es: %f y el de B es: %f \n',A,B);`

# Tips avanzados

- Exportar a un archivo de texto:

```
fileid=fopen('export.txt','w');  
fprintf(fileid, 'Parametro A \n');  
fprintf(fileid, 'A=%f \n',A);  
fclose(fileid);
```

