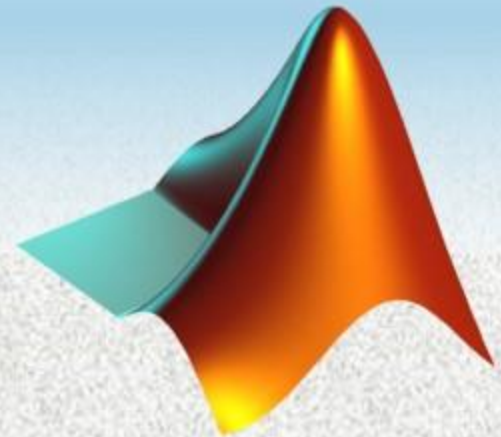




**UNIVERSIDAD
NACIONAL DE
INGENIERÍA**

MATLAB

R2016a

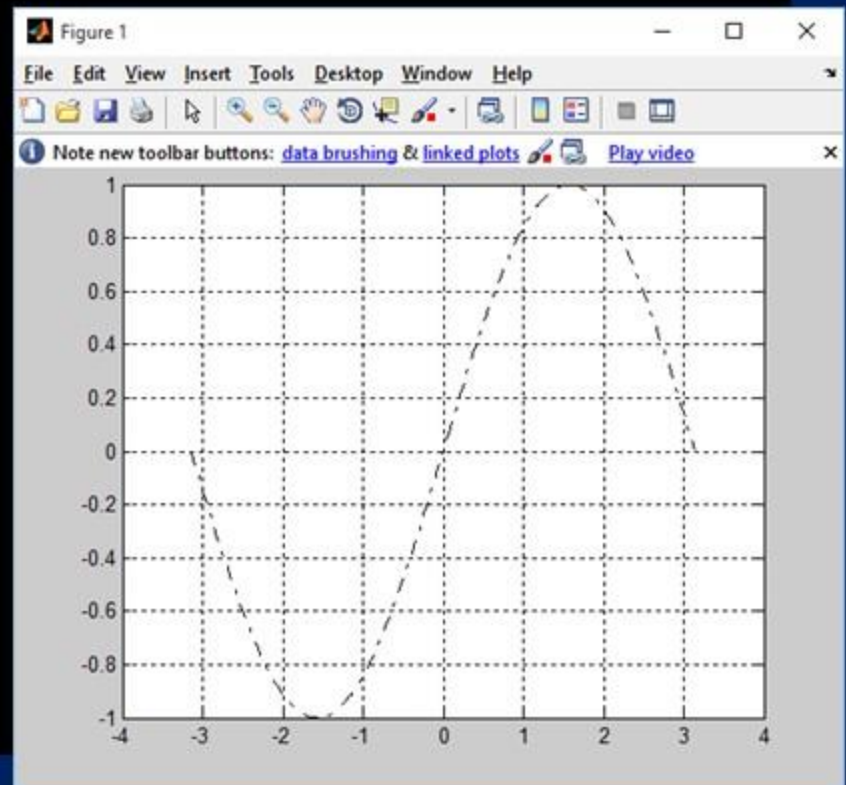


Maria Pimentel Herrera
uni.kernel@gmail.com

M-File: sd004153.m

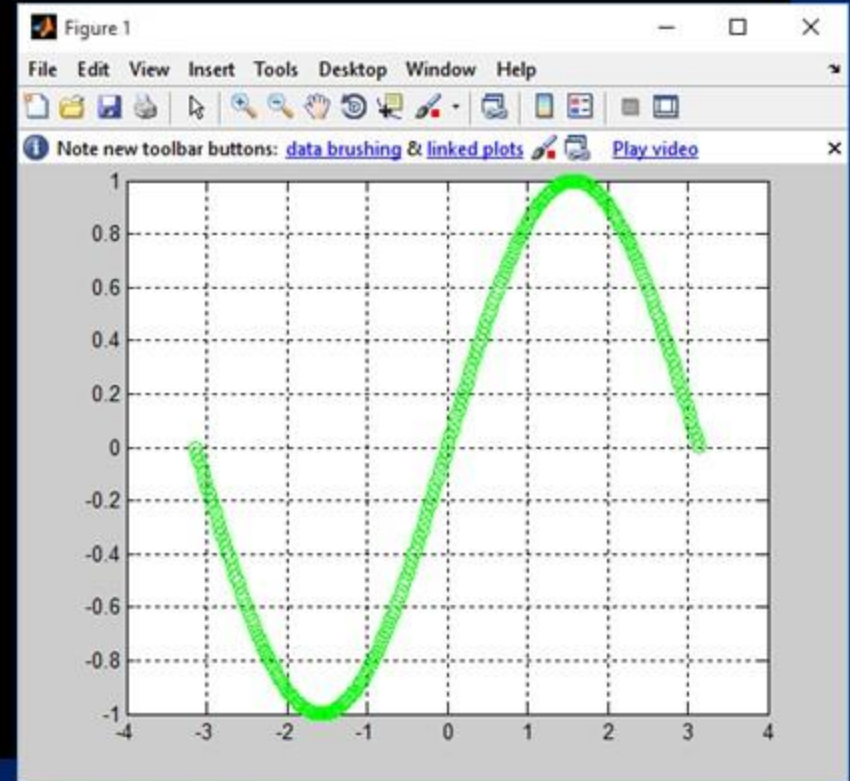
```
x=linspace(-pi,pi,250);  
y=sin(x);  
plot(x,y,'k-.')  
grid
```

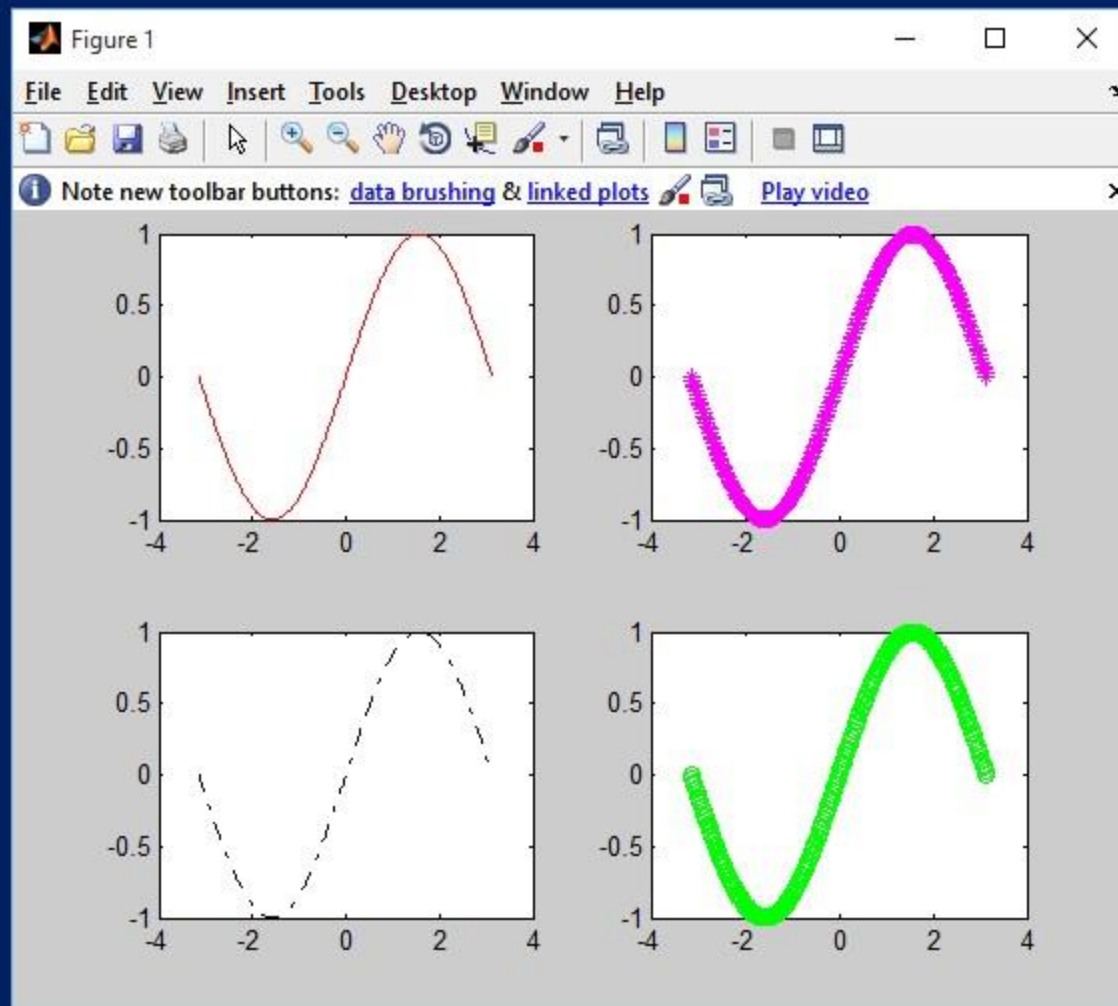
Tipos de línea	Símbolo
Continua	-
Guiones	--
Punteada	:
Guiones y punto	-.



M-File: sd004154.m

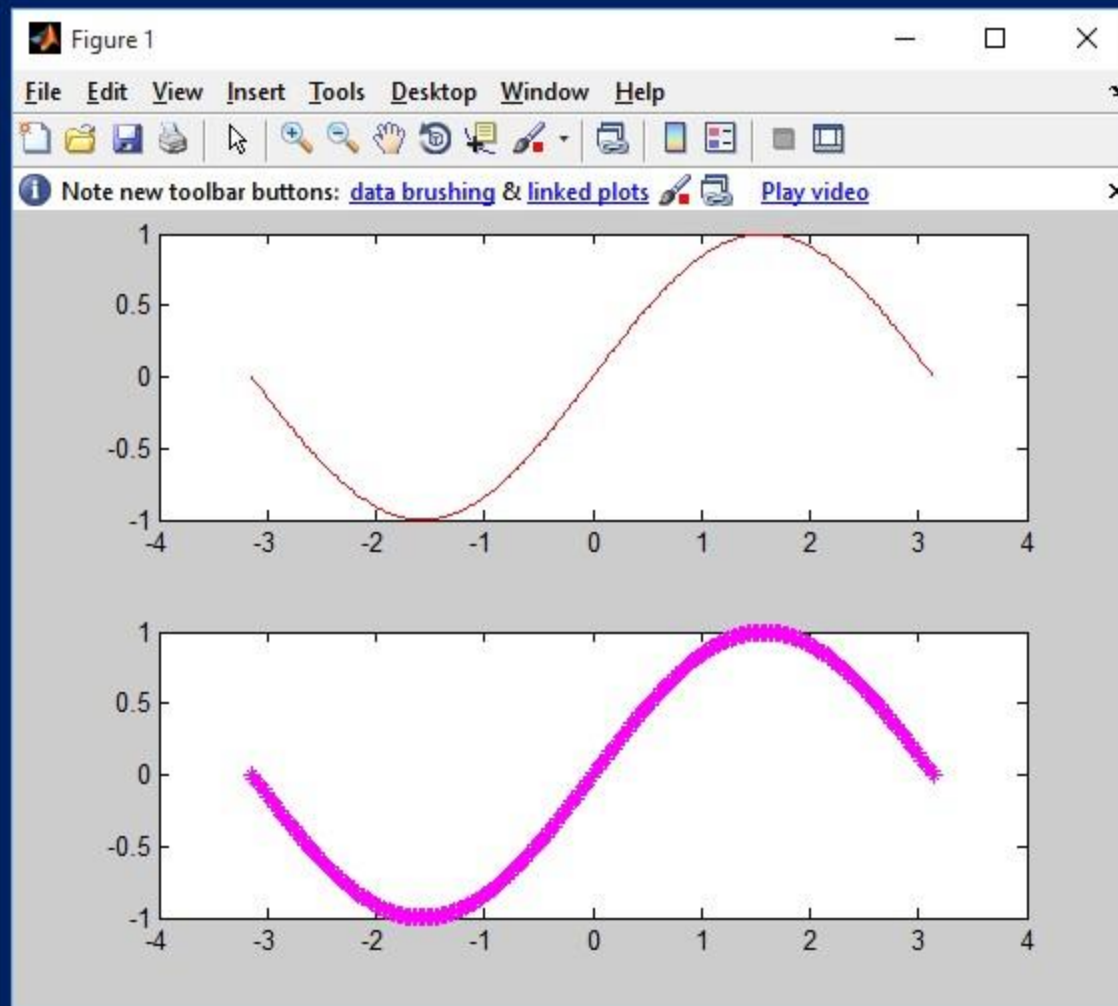
```
x=linspace(-pi,pi,250);  
y=sin(x);  
plot(x,y,'go')  
grid
```





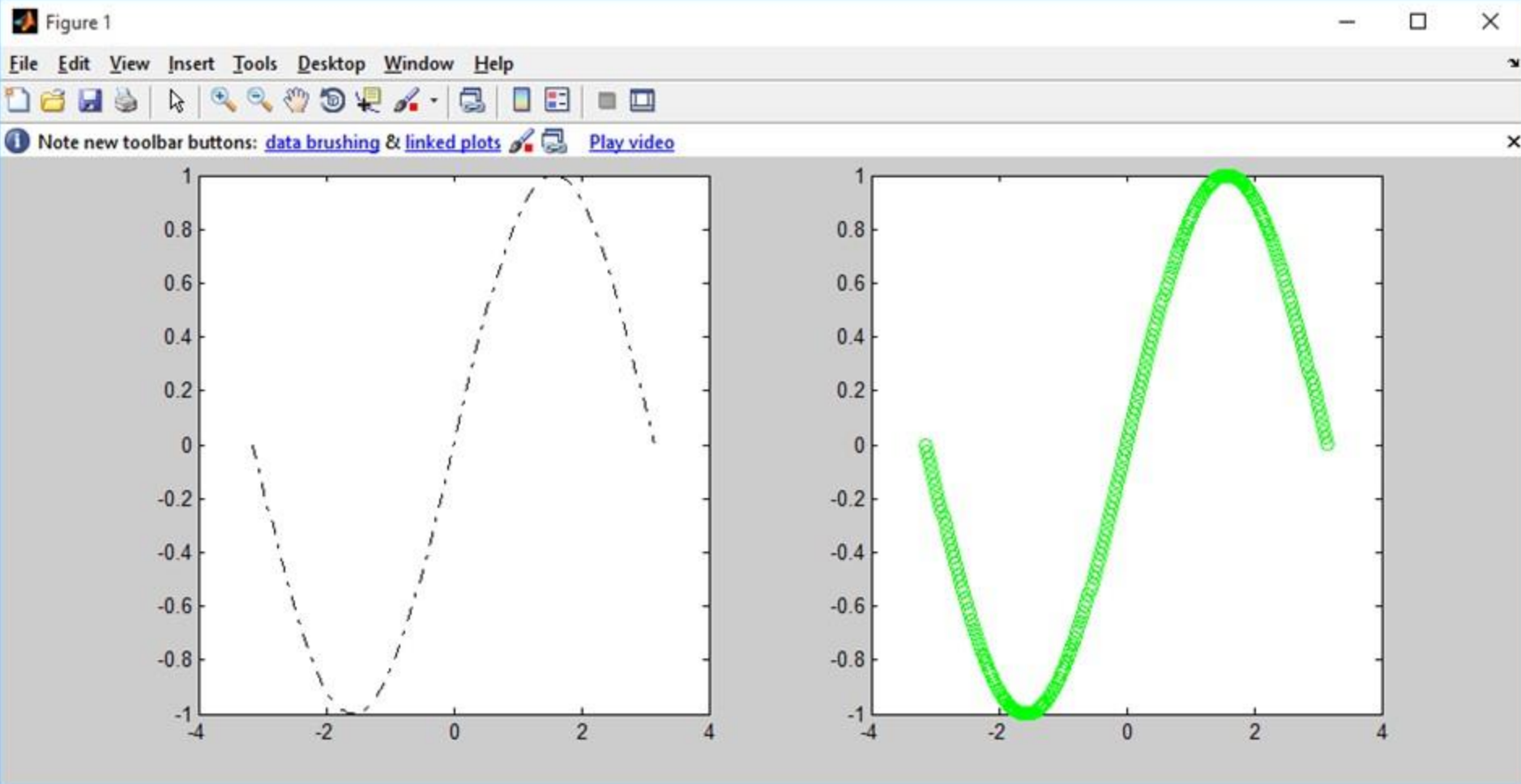
M-File: sd004155.m

```
subplot(1,1,1)
x=linspace(-pi,pi,250);
y=sin(x);
subplot(2,2,1), plot(x,y,'r')
subplot(2,2,2), plot(x,y,'m*')
subplot(2,2,3), plot(x,y,'k-.')
subplot(2,2,4), plot(x,y,'go')
```



M-File: sd004156.m

```
subplot(1,1,1)
x=linspace(-pi,pi,250);
y=sin(x);
subplot(2,1,1), plot(x,y,'r')
subplot(2,1,2), plot(x,y,'m*')
```



M-File: sd004157.m

```
subplot(1,1,1)
x=linspace(-pi,pi,250);
y=sin(x);
subplot(1,2,1), plot(x,y,'k-.')
subplot(1,2,2), plot(x,y,'go')
```

Operadores relacionales

<	Menor que
<=	Menor igual que
>	Mayor que
>=	Mayor igual que
==	Igual que
!=	No es Igual que

1. Script: Grafica en 2D

```
sd0051.m  x  +
1      % Ventana para un grafico
2 -    subplot(1,1,1)
3      % Entrada de dato
4 -    fx=input('Frecuencia en x: ');    % pi/4
5 -    dx=input('Fase en x: ');          % 0.05
6 -    fy=input('Frecuencia en y: ');    % pi/2
7 -    dy=input('Fase en y: ');          % 0.03
8      % Proceso
9 -    t=0:pi/100:2*pi;
10 -    x=cos(fx*t + dx);
11 -    y=sin(fy*t + dy);
12     % Salida
13 -    plot(x,y,'m*')
```

2. Script: Derivadas

```
sd0052.m x +
1 -   clc
2 -   syms x
3 -   g = x^5-x^3+15*x^2
4 -   fprintf('1ra. derivada: %0s\n',diff(g,x,1));
5 -   fprintf('2da. derivada: %0s\n',diff(g,x,2));
6 -   fprintf('3ra. derivada: %0s\n',diff(g,x,3));
7 -   fprintf('4ta. derivada: %0s\n',diff(g,x,4));
8 -   fprintf('5ta. derivada: %0s\n',diff(g,x,5));
```

```
g =

x^5 - x^3 + 15*x^2

1ra. derivada: 30*x - 3*x^2 + 5*x^4
2da. derivada: 20*x^3 - 6*x + 30
3ra. derivada: 60*x^2 - 6
4ta. derivada: 120*x
5ta. derivada: 120
```

3. Script: Derivadas parciales

```
clc
syms x y
f=sin(x*y)+cos(x*y^2);
fx=diff(f,x,1)
fy=diff(f,y,1)
fxy=diff(fx,y,1)
fyx=diff(fy,x,1)
```


$f_x =$

$$y \cos(xy) - y^2 \sin(xy^2)$$

$f_y =$

$$x \cos(xy) - 2xy \sin(xy^2)$$

$f_{xy} =$

$$\cos(xy) - 2y \sin(xy^2) - xy \sin(xy) - 2xy^3 \cos(xy^2)$$

$f_{yx} =$


$$\cos(xy) - 2y \sin(xy^2) - xy \sin(xy) - 2xy^3 \cos(xy^2)$$

4. Función con un resultado:

guardar con el nombre **sumamayores**

```
function suma = sumamayores(vector,min)
%{
sumamayores(v,m)
suma los numeros de un vector que sean
mayores o iguales que uno dado
argumentos:
v : vector de numeros a sumar
m : minimo valor que puede sumar
resultado: suma
%}
suma = sum(vector(vector>=min));
```

- En la ventana de comando o en otro script, crear un vector $v=[1 \ 2 \ 3 \ 4 \ 5]$, luego escribir **sumamayores(v,4)** resultado: 9

```
 function suma=sumamayores(vector,min)  
    suma=sum(vector(vector>=min));
```

```
>> v=[2 3 -4 5 9]; sm=sumamayores(v,4)
```

```
sm =
```

```
14
```

```
>> total=sumamayores([7:1.5:15],11)
```

```
total =
```

```
39
```

5. Función con dos resultados: guardar con el nombre **sumaymen**

```
function [may, men] = sumaymen(vector,min)
%{
sumaymen(v,m)
suma los numeros de un vector que sean
mayores o iguales que uno de rererencia
y los menores que este
argumentos:
v : vector de numeros a sumar
m : minimo valor que puede sumar
resultado: [mayores menores]
mayores: suma de los mayores o iguales
menores: suma de los menores
%}
elementos = vector>=min;
may = sum(vector(elementos));
men = sum(vector(~elementos));
```

- En la ventana de comando o en otro script, crear un vector $v=[1 \ 2 \ 3 \ 4 \ 5]$, escribir `[mayor menor]=sumaymen(v,4)` resultado: 9 y 6

sumaymen.m



```
1 function [may, men] = sumaymen(vector,min)
2 -     elementos = vector >= min;
3 -     may = sum(vector(elementos));
4 -     men = sum(vector(~elementos));
```

```
>> [a b]=sumaymen([-3 5 -4 6 7 9 -1],6)
```

```
a =
```

```
22
```

```
b =
```

```
-3
```


6. Script: Intercambio de filas y columnas en matrices

```
clc
A = [1 2 3 4;5 6 7 8;9 10 11 12;13,14,15,16]
disp('Intercambiar fila 1 con fila 3');
% guardar los datos de la fila 1 en un vector auxiliar
tmp = A(1,:);
% asignar los valores de la fila 3 a la fila 1
A(1,:) = A(3,:);
% asignar los valores del tmp a la fila 3
A(3,:) = tmp
disp('Intercambiar columna 2 con columna 3');
tmp = A(:,2);
A(:,2) = A(:,3);
A(:,3) = tmp
```

7. Script: aplicación if

Intercambio de valores si $a > b$

```
clc;
a=input('ingrese un valor para a:');
b=input('ingrese un valor para b:');
if a > b
    fprintf('intercambia valores si %0.0f > %0.0f\n',a,b)
    tmp=a;
    a=b;
    b=tmp;
    fprintf('nuevo valor de a es: %0.0f\n',a);
    fprintf('nuevo valor de b es: %0.0f\n',b);
end
```

Agregar las líneas sombreadas

```
clc;  
a=input('ingrese un valor para a = ');  
b=input('ingrese un valor para b = ');  
if a > b  
    fprintf('intercambia valores si %0.0f > %0.0f\n',a,b)  
    tmp=a;  
    a=b;  
    b=tmp;  
    fprintf('nuevo valor de a es: %0.0f\n',a)  
    fprintf('nuevo valor de a es: %0.0f\n',b)  
else  
    fprintf('no hay intercambia%0.0f >= %0.0f\n',b,a)  
end
```

Agregary modificar

```
clc;
a=input('ingrese un valor para a = ');
b=input('ingrese un valor para b = ');
if a > b
    fprintf('intercambia valores si %0.0f > %0.0f\n',a,b)
    tmp=a;
    a=b;
    b=tmp;
    fprintf('nuevo valor de a es: %0.0f\n',a)
    fprintf('nuevo valor de a es: %0.0f\n',b)
elseif a==b
    fprintf('valores iguales %0.0f = %0.0f\n',a,b)
else
    fprintf('se cumple a<b %0.0f<%0.0f\n',a,b)
end
```

8. Script: aplicación if

Si promedio ≥ 13 , Aprobado, si promedio > 16.5 Excelente, en caso contrario Desaprobado.

```
promedio = input('ingrese promedio ');  
if promedio > 16.5  
    disp('Excelente');  
elseif promedio  $\geq$  13  
    disp('Aprobado');  
else  
    disp('Desaprobado');  
end
```


Polinomios

Polinomios

$$f(x) = a_0 x^N + a_1 x^{N-1} + a_3 x^{N-3} + a_2 x^{N-2} + \dots \\ + a_{N-2} x^2 + a_{N-1} x + a_N$$

- Variable x
- Coeficientes $a_i, i=0..N$
- Grado N

9. Script: Polinomios

```
clc
format;
% Polinomios
p = [ 1 7 1 -1]      % representa:  x^3 + 7x^2 + x - 1
q = [ 3 2 1 0]      % 3x^3 + 2x^2 + x
% Suma de polinomios
suma = p+q           % 4x^3 + 9x^2 + 2x -1
% Diferencia
diferencia = p-q     % -2x^3 + 5x^2 - 1
% Producto por un escalar
por_escalar = 3*q     % 9x^3 + 6x^2 + 3x
% Producto entre polinomios
producto = conv(p,q)  % 3x^6+23x^5+18x^4+6x^3-x^2-x
r1 = [1 1]
producto =conv(r1,q)  % 3x^4+5x^3+3x^2+x
r1 = [0 0 1 1]
producto =conv(r1,q)  % 3x^4+5x^3+3x^2+x
% División entre polinomios
[cociente resto] = deconv([3 5 3 1 0], q)      % x+1
```

```

% Raices de un polinomio
format rat;
raices = roots(p)
raices = roots(q)
% Construir un polinomio a partir de sus raices
polinomio = poly([1 1])           %  $x^2 - 2x + 1$ 
polinomio = poly([1 1 -1])        %  $x^3 - x^2 - x + 1$ 
polinomio = poly([-1 -1 -1])      %  $x^3 + 3x^2 + 3x + 1$ 
% Evaluar un polinomio
evaluarp = polyval([1 1],2)       %  $p(x) = x + 1$  /  $p(2)=3$ 
evaluarp = polyval([1 1 1],-1)    %  $p(x) = x^2 + x + 1$  /  $p(-1)= 1$ 
% Evaluar un polinomio con un matriz
evaluarpm = polyvalm([3,2,1],[1,0;0,1]) %  $3x^2+2x+1$ 
% Derivada de un polinomio
polinomio = [3,2,1]
vpa(poly2sym(polinomio))
derivadap = polyder(polinomio)
vpa(poly2sym(derivadap))
% Integral de un polinomio
integralp = polyint(polinomio)
vpa(poly2sym(integralp))

```