

UNIVERSIDAD POLITÉCNICA DE VICTORIA



INTEGRACIÓN DE SISTEMAS MECATRÓNICOS

TAREA 2

ESCUELA: Universidad Politécnica de Victoria

MAESTRO: MC Enrique Martínez Peña

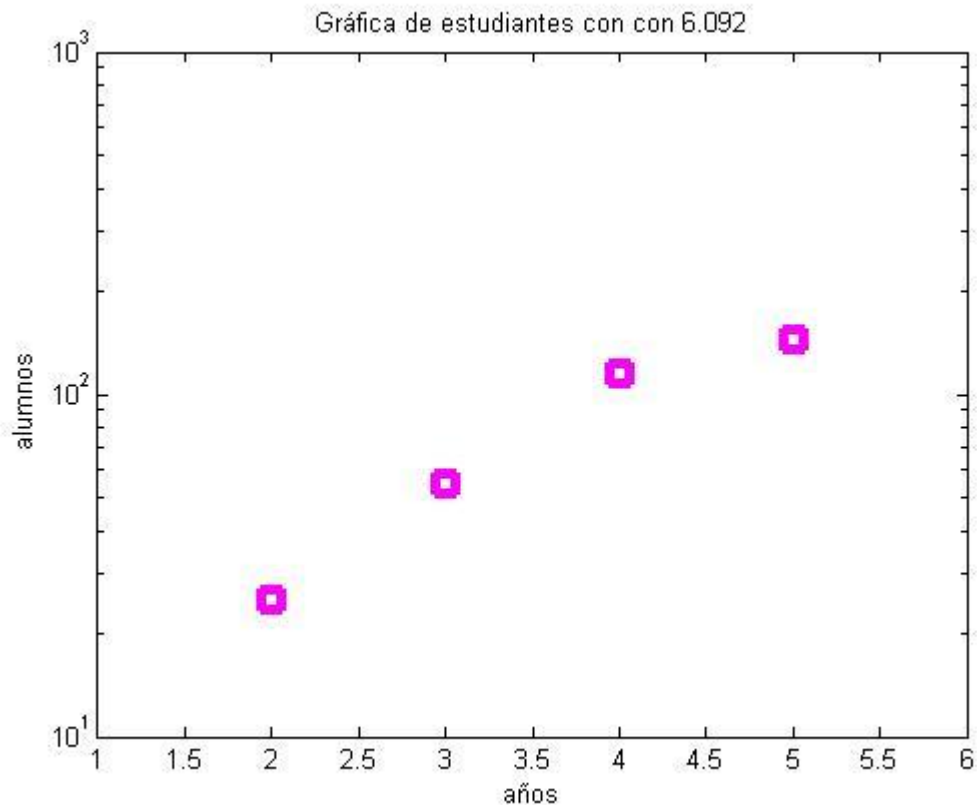
ASIGNATURA: Integración de Sistemas
Mecatrónicos

ALUMNO: Ángel Arturo Ramírez Suárez

GRUPO: IM 8-1

Ejercicio 1. - Graficación de una función semilogarítmica.

Dentro de los 5 años pasados, el número de estudiantes en 6.094 ha sido de 15, 25, 55, 115 y 144. El tamaño de la clase parece tener un crecimiento exponencial. Para verificarlo, grafica estos valores en una gráfica con una escala logarítmica y etiquétala `semilogy`, `xlabel`, `ylabel`, `title`. Usa símbolos magenta cuadrados de tamaño de cuadro 10 y ancho de línea 4 sin línea conectándolos. Puede que tengas que cambiar los límites de x para visualizar todos los símbolos. Si la relación es exponencial, se verá lineal en la gráfica.



Código.

```
%*****EJERCICIO 1*****
```

```
%Se declara el vector de datos.
```

```
y=[15 25 55 115 144 0];
```

```
%Se realiza la graficación con los parámetros especificados.
```

```
semilogy(1:1:6,y,'ms','LineWidth',4,'MarkerSize',10)
```

```
%Se colocan las etiquetas.
```

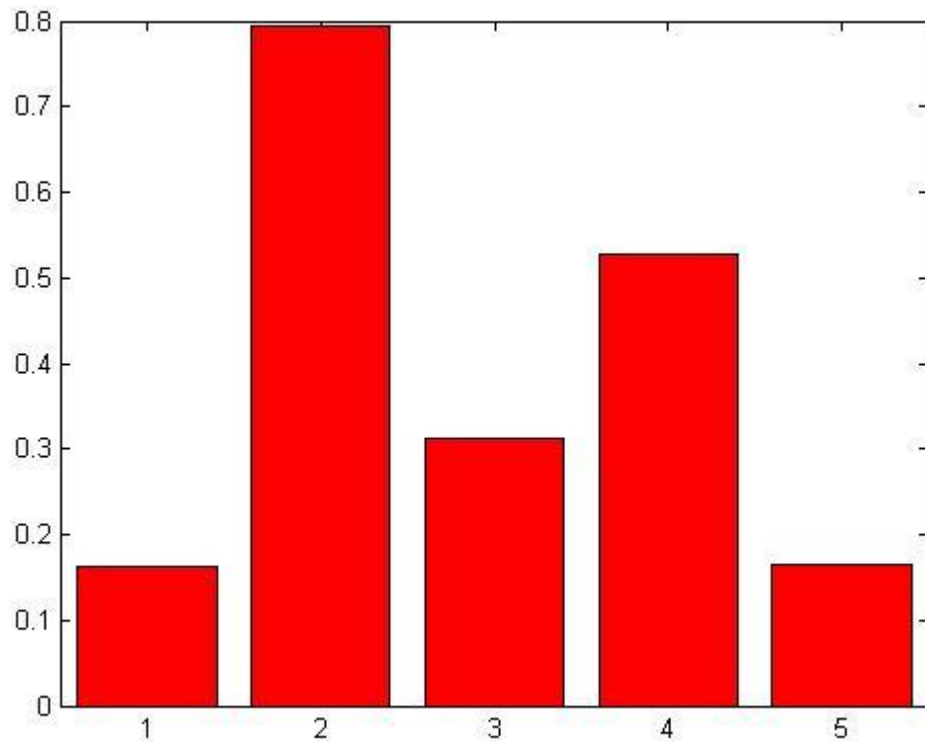
```
xlabel('años')
```

```
ylabel('alumnos ')
```

```
title('Gráfica de estudiantes con con 6.092')
```

Ejercicio 3. – Gráfica de barras.

Elabora un vector con 5 valores aleatorios y gráficelos en una gráfica de barras usando barras rojas.



Código.

```
%****EJERCICIO 3****
```

```
%Se genera un vector aleatorio.
```

```
A=rand(1,5);
```

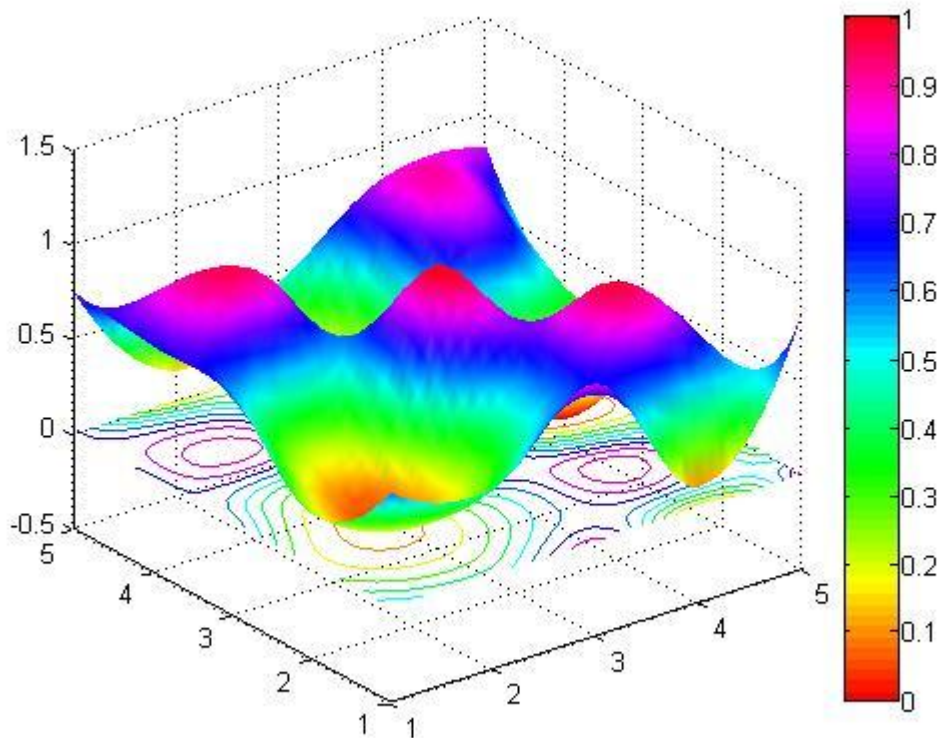
```
%Se genera la gráfica de barras y se especifica color rojo.
```

```
bar(A,'r')
```

Ejercicio 4.- Interpolación y graficación de superficies.

Elabora un script llamado randomSurface.m para que realice lo siguiente:

1. Elabore una superficie aleatoria, utilizando una matriz de valores aleatorios del rango [0,1] utilizando rand.
2. Elaborar un meshgrid de X0 y Y0 con el vector 1:5. X0, Y0 y Z0 definen 25 puntos en la superficie.
3. Interpolamos valores intermedios para suavizar la superficie.
4. Interpolamos X0, Y0 y Z0 en las posiciones X1 y Y1 usando interpolación cúbica.
5. Graficar una superficie Z1. Establecer el mapa de color a hsv y la sombra a interp.
6. Tomar los ejes y graficar 15 líneas de contorno en los mismos.
7. Añadir una barra de color.
8. Establecer el eje de colores de 0 a 1.



Código.

```
% *****EJERCICIO 4*****
```

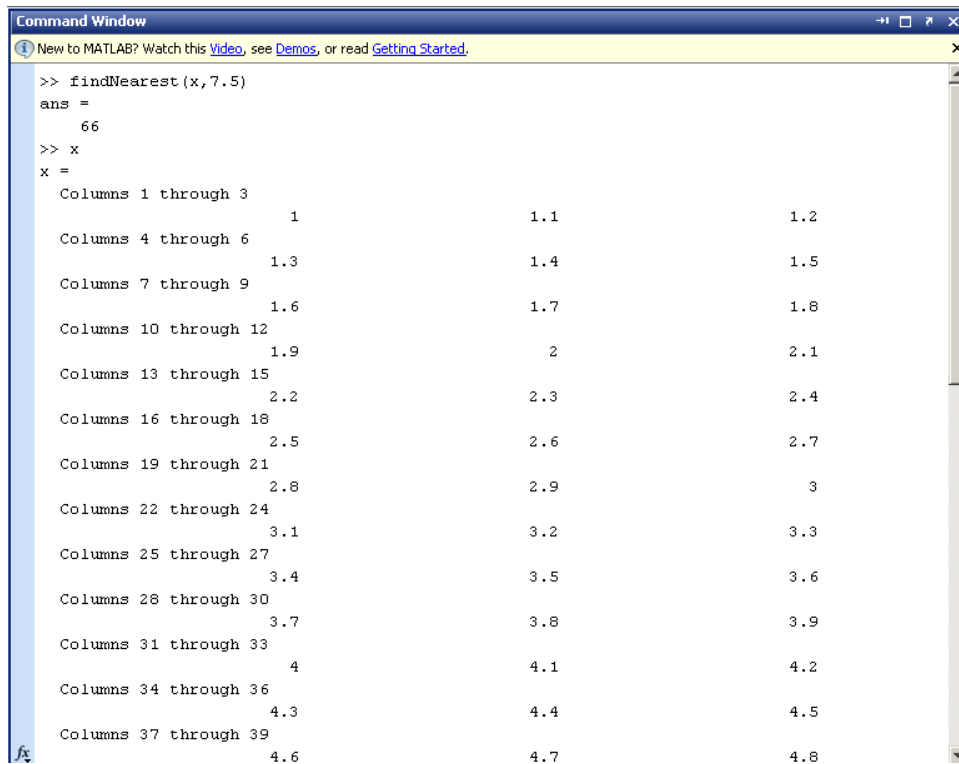
```
%Se genera el vector aleatorio.
```

```
Z0 = rand(5);
```

```
[X0,X0] = meshgrid(1:5,1:5);  
[X1,X1] = meshgrid(1:.1:5);  
%Se realiza la interpolación.  
Z1 = interp2(X0,Y0,Z0,X1,Y1,'cubic');  
%Se definen parámetros de las gráficas.  
surf(X1,Y1,Z1)  
colormap(hsv)  
shading INTERP  
hold on  
contour(X1,Y1,Z1)  
colorbar  
V = [0 1];  
caxis (V)
```

Ejercicio 5. – Find.

Escribe una función para regresar el índice del valor más cercano a un valor deseado.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> findNearest(x,7.5)
ans =
    66
>> x
x =
Columns 1 through 3      1      1.1      1.2
Columns 4 through 6    1.3      1.4      1.5
Columns 7 through 9    1.6      1.7      1.8
Columns 10 through 12  1.9      2      2.1
Columns 13 through 15  2.2      2.3      2.4
Columns 16 through 18  2.5      2.6      2.7
Columns 19 through 21  2.8      2.9      3
Columns 22 through 24  3.1      3.2      3.3
Columns 25 through 27  3.4      3.5      3.6
Columns 28 through 30  3.7      3.8      3.9
Columns 31 through 33  4      4.1      4.2
Columns 34 through 36  4.3      4.4      4.5
Columns 37 through 39  4.6      4.7      4.8
```

```
%*****EJERCICIO 5*****
```

```
function r = findNearest(x,determinedVal)
```

```
%Se determina el intervalo.
```

```
x=abs(x-determinedVal);
```

```
%X obtiene los valores de x.
```

```
y=x(:);
```

```
%Se localiza el valor mínimo.
```

```
m=min(y);
```

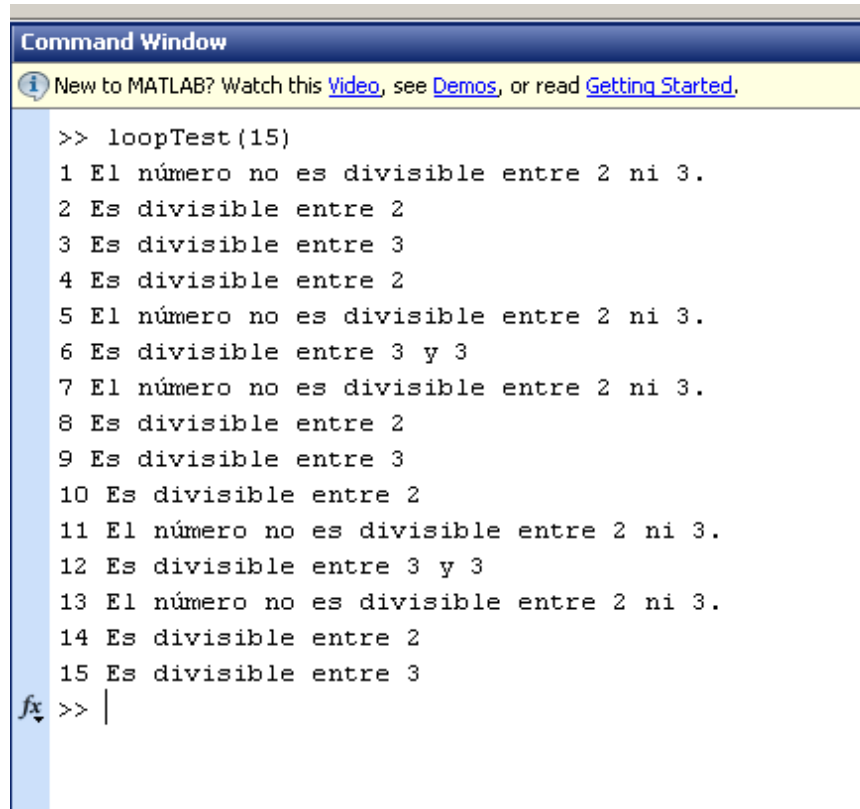
```
%Se encuentra el índice en el cual y = m.
```

```
r=find(y==m);
```

```
end
```

Ejercicio 6. – Ciclos y control de flujo.

Elabora una función llamada loopTest(N) que itere los valores 1 a N y por cada número n debe mostrar si n es divisible entre 2 y 3, n es divisible por 2 y 3 o n no es divisible por 2 y 3.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> loopTest(15)
1 El número no es divisible entre 2 ni 3.
2 Es divisible entre 2
3 Es divisible entre 3
4 Es divisible entre 2
5 El número no es divisible entre 2 ni 3.
6 Es divisible entre 3 y 3
7 El número no es divisible entre 2 ni 3.
8 Es divisible entre 2
9 Es divisible entre 3
10 Es divisible entre 2
11 El número no es divisible entre 2 ni 3.
12 Es divisible entre 3 y 3
13 El número no es divisible entre 2 ni 3.
14 Es divisible entre 2
15 Es divisible entre 3
fx >> |
```

Código.

```
%***** EJERCICIO 6*****
```

```
function loopTest(n)
```

```
vec = 1:n;
```

```
for n=1:n
```

```
    val1 = vec(n);
```

```
    val2 = num2str(val1);
```

```
    if (rem(val1,2))==0 && (rem(val1,3))==0
```

```
        display([val2 ' Es divisible entre 3 y 3'])
```

```
    elseif (rem(val1,2))==0
```

```
    display([val2 ' Es divisible entre 2'])  
elseif (rem(val1,3))==0  
    display([val2 ' Es divisible entre 3'])  
else  
    display([val2 ' El número no es divisible entre 2 ni 3.'])  
end  
end
```