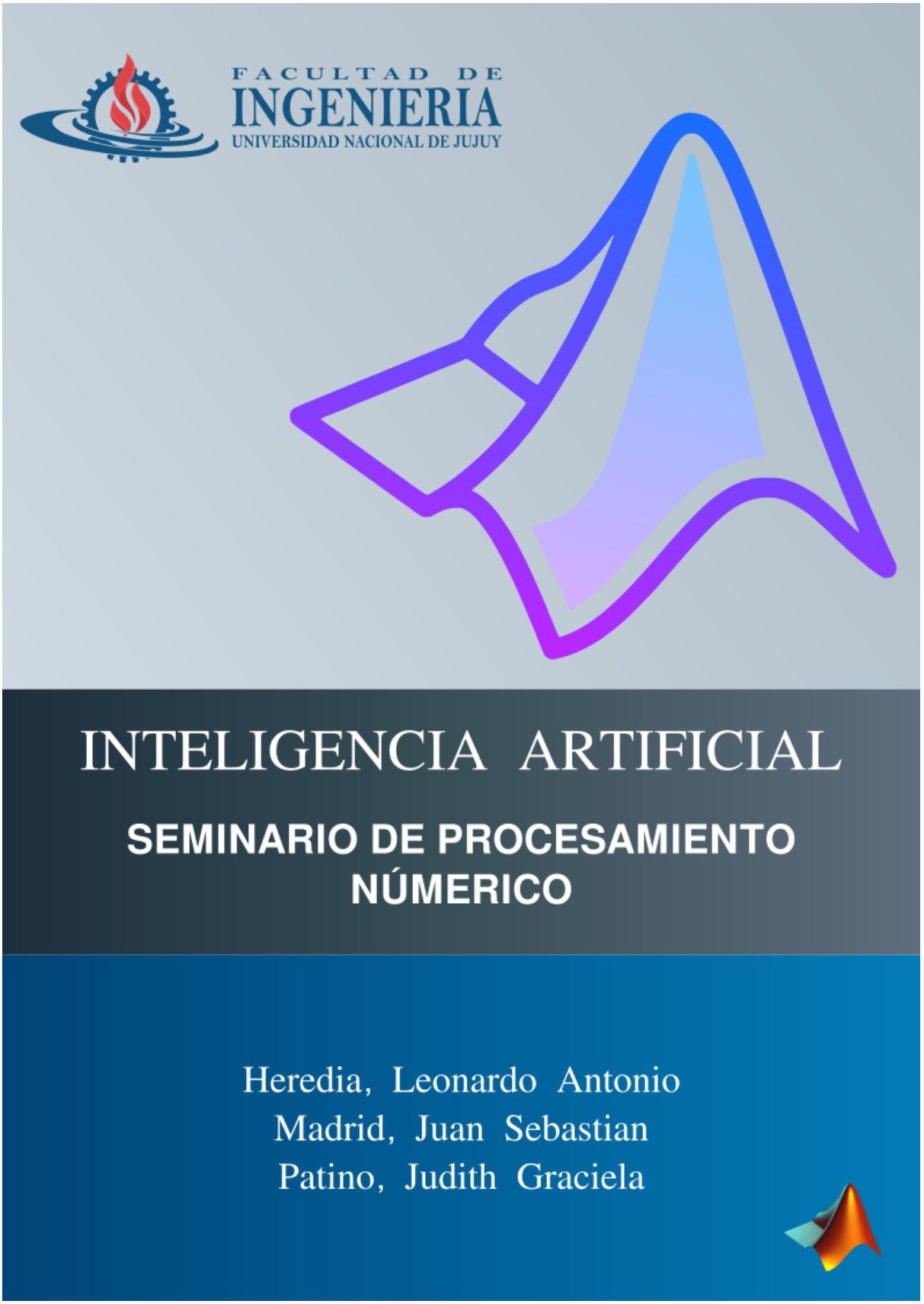
.....


FACULTAD DE
INGENIERIA
UNIVERSIDAD NACIONAL DE JUJUY
INTELIGENCIA ARTIFICIAL
SEMINARIO DE PROCESAMIENTO NÚMERICO
Heredia, Leonardo Antonio
Madrid, Juan Sebastian
Patino, Judith Graciela

.....

1. Trabajo con matrices. Generar matrices con las características que se indican:

a) Matriz A de 20x25 de valores enteros positivos. Extraer una matriz A1(3x8) desde la posición (5,5). Matriz A2 que resulte de la eliminación de las columnas 11 a 14 y las filas 9 a 13 de A.

```matlab
% Con clear borramos todas las variables del workspace
clear;
% Con clc limpiamos el command windows
clc;
% Con rand obtenemos valores aleatorios decimales entre 0 y 1, ej 0.0635
% Con randi obtenemos valores aleatorios enteros positivos randi()
% Con round obtenemos valores enteros redondeados
% Generamos la matriz A de 20x25 con valores enteros positivos.
% A = randi(100,20,25)
A = round(10*(rand(20,25)));
% Extraemos la matriz A1 de 3x8 desde la posición (5,5).
A1 = A(5:7,5:12)
% Asignamos A a A2 para poder eliminar las filas y columnas
A2=A
% Primero eliminamos las columnas.
A2(:,11:14) = [];
% Luego eliminamos las filas
A2(9:13,:) = []

```

```
A =

  Columns 1 through 20                                                                              ›lumns 21 through 25

     6     5     8     9     8     8    10     2     9     0     5     1     2     9     1     5     7     7     2     3     7     6    10     7     9
     6     8     6     6    10     6     1     2     7     7     6     4     8     6     3    10     5     9     9     4     8     3     6     7     4
     2     9     9     0     2     2     1     3     4     4     9     9     3     6     8     8     3     2     7     9     4     4     9     5     8
     1     8     1     1     8     5     3     9     7     8     2     5     4     9     2    10     8     1    10     6     4     8     7     3     6
     3     3     6     8     2     5     6     5    10     4     7     4     2     0     7     2     6     2     4    10    10     2     5    10     8
     9     5     3     5    10    10     5     4     5     4     6     2     8     9     7     5    10     4     9     2     6     3     6     5     1
     9     8     8     4     8     9     9     2     5     0     4     4     6     4     8     1     9     3     0     8     8     5     9     0    10
     7     1     2     8     4    10     5    10     3     0     9    10     7     0     8     8     4     9     6     7     3     3     2     7     8
     7     1     4     4     7     7     4     4     1     1     4     1     8     7     3     6     5     1     8     2     6     8     4     5     1
     2     3     4     5     5     4     5     8     2     6     2     5     1     2     3     9     3     6     2     5     6    10    10     1     5
     6     5     8     7     8     9     7     6     1     2     6     9    10     1     5    10     6     2     9     4    10     2     4     9     3
     8    10     7     9     4     5     0     4     5     8     6     0     5     6     3     9     8     8     8     6     1     2     7     3     6
     4     7     2     3     1     2     8     9     0     9     3     7     8     3     8     4     7     2     8     8     5     7     9     2     2
    10     3     3     7     6     4     1     8     9    10     8    10     7     3     8     0     1     5     6     1     5     4    10     1     6
     1     3     1    10     9     7     5     5     6     5    10     3     8     4     6     5     8    10     8     8     1    10     7     3     6
     3     9     7     1     2     6     3     8     0     2    10     1     2     4     3     2     0     4     3     8     9    10     1     2     6
     5     9     6     6     4     8     4     9     0     2     1     7     5     4     7     2     4     0     2     4     9     6     0     7     4
     1     6     2     4     7    10     7     4     2     5     2     9     6     6     2     3     7     2     3     4     4     9     6     1     0
     7     3     1     3     0    10     2     3     5     8     0     6     9     2     5     1     8     4     6     6     8     4     6     3     5
     6     1     5     3     9     5     3     6     1     3     6     9     8     2     4     7     4     3     8     7     1     6    10     3     4
```

```
A1 =

     2     5     6     5    10     4     7     4
    10    10     5     4     5     4     6     2
     8     9     9     2     5     0     4     4
```

…..

```
A2 =

  Columns 1 through 20

     6     5     8     9     8     8    10     2     9     0     1     5     7     7     2     3     7     6    10     7
     6     8     6     6    10     6     1     2     7     7     3    10     5     9     9     4     8     3     6     7
     2     9     9     0     2     2     1     3     4     4     8     8     3     2     7     9     4     4     9     5
     1     8     1     1     8     5     3     9     7     8     2    10     8     1    10     6     4     8     7     3
     3     3     6     8     2     5     6     5    10     4     7     2     6     2     4    10    10     2     5    10
     9     5     3     5    10    10     5     4     5     4     7     5    10     4     9     2     6     3     6     5
     9     8     8     4     8     9     9     2     5     0     8     1     9     3     0     8     8     5     9     0
     7     1     2     8     4    10     5    10     3     0     8     8     4     9     6     7     3     3     2     7
    10     3     3     7     6     4     1     8     9    10     8     0     1     5     6     1     5     4    10     1
     1     3     1    10     9     7     5     5     6     5     6     5     8    10     8     8     1    10     7     3
     3     9     7     1     2     6     3     8     0     2     3     2     0     4     3     8     9    10     1     2
     5     9     6     6     4     8     4     9     0     2     7     2     4     0     2     4     9     6     0     7
     1     6     2     4     7    10     7     4     2     5     2     3     7     2     3     4     4     9     6     1
     7     3     1     3     0    10     2     3     5     8     5     1     8     4     6     6     8     4     6     3
     6     1     5     3     9     5     3     6     1     3     4     7     4     3     8     7     1     6    10     3
```

https://github.com/openjuy/ia2022/tree/main/tp01/01a-Matrices

b) Matriz de 5x10 de números aleatorios enteros de dos dígitos. Determinar la posición y valor del menor y del mayor.

```
% ------------------------------------------------------------------------
% TRABAJO CON MATRICES
% ------------------------------------------------------------------------
% Matriz de 5 x 10 de números aleatorios enteros de dos dígitos. Determinar
% la posición y valor del menor y del mayor.
% ------------------------------------------------------------------------
% max
% Elementos máximos de un array
% M = max(A) devuelve los elementos máximos de un array.
% ------------------------------------------------------------------------
% [M,I] = max(___) encuentra los índices de los valores máximos de A y los
% devuelve en el vector de salida I, utilizando cualquiera de los
% argumentos de entrada de las sintaxis anteriores. Si el valor máximo se
% produce más de una vez, max devuelve el índice correspondiente a la
% primera aparición.
% ------------------------------------------------------------------------
clear; clc;
%
A = randi([-99,99],5,10)
% Aquí calculamos el máximo y su posición.
[max, posMax] = max (A)
% Aquí calculamos en mínimo y su posición.
[min, posMin] = min (A)

```

…..

```
A =

   -78    40    91    48    50    25    75   -86   -88     9
    -8    74    58     1   -26     8   -97    16    63    40
   -10   -89   -10   -60    88    30   -41    27     6    91
    10   -56   -33   -14   -96    45   -64    30    39   -11
    61    -8   -88   -66    65   -81    85    73   -57   -83


max =

    61    74    91    48    88    45    85    73    63    91


posMax =

     5     2     1     1     3     4     5     5     2     3
```

```
min =

   -78   -89   -88   -66   -96   -81   -97   -86   -88   -83


posMin =

     1     3     5     5     4     5     2     1     1     5
```

https://github.com/openjuy/ia2022/tree/main/tp01/01b-Matrices

c) Matriz aleatoria de 10x20, valores enteros en el intervalo (-25 ; 75). Ordenar por filas (orden creciente). Ordenar por columnas (orden decreciente).

```
% ------------------------------------------------------------------------
% TRABAJO CON MATRICES
% ------------------------------------------------------------------------
% Matriz aleatoria de 10x20, valores enteros en el intervalo(-25;75).
% Ordenar
% por filas(orden creciente). Ordenar por columnas(orden decreciente).
% ------------------------------------------------------------------------
% max
% Elementos máximos de un array
% M = max(A) devuelve los elementos máximos de un array.
% ------------------------------------------------------------------------
% [M,I] = max(___) encuentra los índices de los valores máximos de A y los
% devuelve en el vector de salida I, utilizando cualquiera de los
% argumentos de entrada de las sintaxis anteriores. Si el valor máximo se
% produce más de una vez, max devuelve el índice correspondiente a la
% primera aparición.
% ------------------------------------------------------------------------
clear; clc;
A=randi([-25 75],10,20)
FilAcs = sort(A,2)
ColDec = sort(A,'descend')
```

.....

```
A =

   -20    53    58     3   -11     3     8     9   -14   -15   -24     1     0    55    67   -16    30    15   -22    57
    38    67    58    43    34    73    65    60    55    61    72    51    68    -2    26    15    62    -8     4    -6
    55    54    32    66   -18   -22    25    -1    37    45    73    65   -19    35    73     4   -21    33    55   -13
    44     4    32    66    58     7    37    33   -18    49   -13    48     5   -14    -6     5    66    36     9    57
     9   -10     3    50    48    73    33    69   -19    40    22    16    34    27   -14   -15   -12    -4   -17    39
    70    60    45     1    68    11    45   -21   -12    27    41    69    -5    59     5    34    59    27    26   -24
    27    54    55    44    24     6   -23   -20    54     7     4     0    39    67    15     3    55    74    12    65
    71     2    19   -12    41   -13    28   -23   -16    41    51    28    55    25    17   -10    67    24    49    27
   -18    -2    20   -13    64    67   -22    43    -1   -14    31    71    25     3     6   -25   -12    45    27    29
    -5     7    22    -6    29   -12    58    35    -1   -11    18     2    40    40    45     3    25    16    56    36

FilAcs =

   -24   -22   -20   -16   -15   -14   -11     0     1     3     3     8     9    15    30    53    55    57    58    67
    -8    -6    -2     4    15    26    34    38    43    51    55    58    60    61    62    65    67    68    72    73
   -22   -21   -19   -18   -13    -1     4    25    32    33    35    37    45    54    55    55    65    66    73    73
   -18   -14   -13    -6     4     5     5     7     9    32    33    36    37    44    48    49    57    58    66    66
   -19   -17   -15   -14   -12   -10    -4     3     9    16    22    27    33    34    39    40    48    50    69    73
   -24   -21   -12    -5     1     5    11    26    27    27    34    41    45    45    59    59    60    68    69    70
   -23   -20     0     3     4     6     7    12    15    24    27    39    44    54    54    55    55    65    67    74
   -23   -16   -13   -12   -10     2    17    19    24    25    27    28    28    41    41    49    51    55    67    71
   -25   -22   -18   -14   -13   -12    -2    -1     3     6    20    25    27    29    31    43    45    64    67    71
   -12   -11    -6    -5    -1     2     3     7    16    18    22    25    29    35    36    40    40    45    56    58

ColDec =

    71    67    58    66    68    73    65    69    55    61    73    71    68    67    73    34    67    74    56    65
    70    60    58    66    64    73    58    60    54    49    72    69    55    59    67    15    66    45    55    57
    55    54    55    50    58    67    45    43    37    45    51    65    40    55    45     5    62    36    49    57
    44    54    45    44    48    11    37    35    -1    41    41    51    39    40    26     4    59    33    27    39
    38    53    32    43    41     7    33    33    -1    40    31    48    34    35    17     3    55    27    26    36
    27     7    32     3    34     6    28     9   -12    27    22    28    25    27    15     3    30    24    12    29
     9     4    22     1    29     3    25    -1   -14     7    18    16     5    25     6   -10    25    16     9    27
    -5     2    20    -6    24   -12     8   -20   -16   -11     4     2     0     3     5   -15   -12    15     4    -6
   -18    -2    19   -12   -11   -13   -22   -21   -18   -14   -13     1    -5    -2    -6   -16   -12    -4   -17   -13
   -20   -10     3   -13   -18   -22   -23   -23   -19   -15   -24     0   -19   -14   -14   -25   -21    -8   -22   -24
```

https://github.com/openjuy/ia2022/tree/main/tp01/01c-Matrices

d) Generar matriz de valores aleatorios binarios de 10x8, con un bit por celda. a1) Utilizar la función randi(); a2) utilizar la función rand() y ajustar para que no haya parte fraccionaria. Comparar ambos métodos.

```
1    clear; clc;
2    Arandi=randi([0 1],10,8)
3    Arand=round(rand(10,8))
4
```

```
Arandi =

     1     0     0     0     1     0     0     1
     1     1     0     0     0     0     1     1
     0     0     0     0     0     1     1     0
     0     1     1     1     0     1     0     0
     1     1     0     1     0     0     1     1
     0     0     0     0     1     0     0     0
     1     0     0     1     0     0     1     1
     0     1     1     1     0     1     1     1
     1     0     0     1     0     0     0     0
     0     0     1     1     1     1     0     1
```

…..

```
Arand =

     1     1     1     1     1     0     1     1
     1     1     1     0     0     1     0     0
     0     1     0     0     0     0     1     0
     0     0     0     1     1     1     1     1
     1     1     0     0     0     0     1     0
     1     0     0     0     0     0     1     1
     1     1     0     1     0     0     0     1
     1     1     0     1     1     0     0     1
     0     1     0     0     1     0     1     1
     1     1     1     1     0     0     1     0
```

https://github.com/openjuy/ia2022/tree/main/tp01/01d-Matrices

De la ejecución de los métodos *randi()* y *rand()* podemos concluir que, rand para números aleatorios uniformemente distribuidos, incluidos números flotantes, mientras que randi para números enteros.

2. Graficación. Desde la línea de comandos ejecutar las sentencias necesarias para graficar las siguientes funciones:

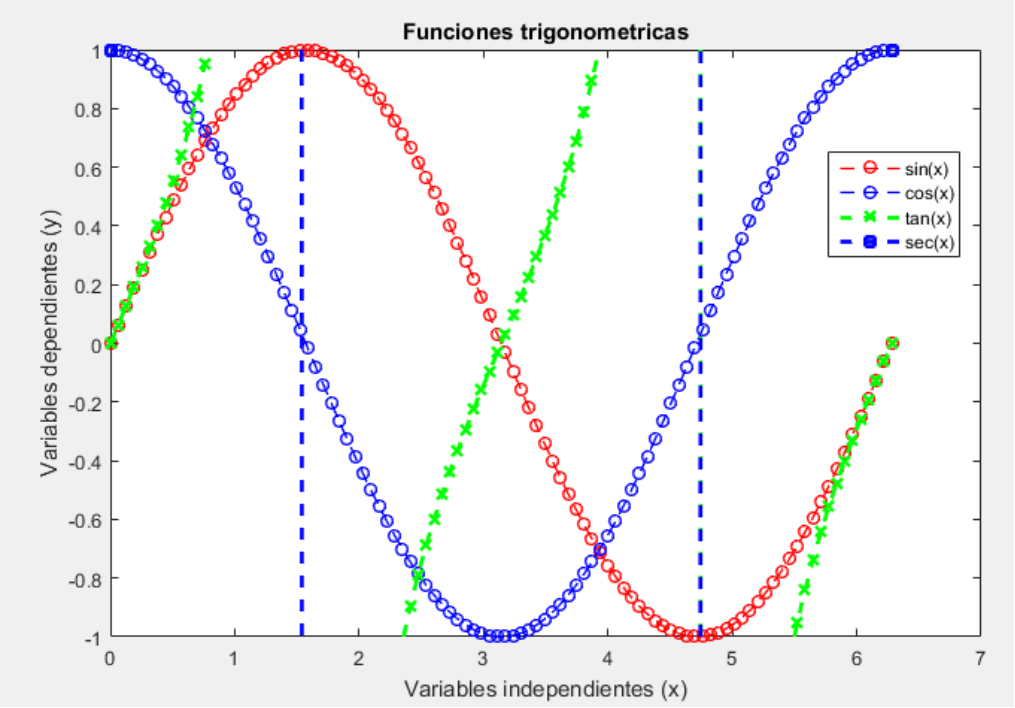
a) Funciones seno y coseno en el intervalo [0, 2π] con un mínimo de 100 puntos cada una, línea continua, color verde para tangente, grosor 2, marca x; color azul para secante, grosor 2, marca rombo. Graficar sobre el mismo sistema de ejes (ver comando **hold on**).

.....

```
% ------------------------------------------------------------------------
% GRAFICACIÓN
% ------------------------------------------------------------------------
% Dos vectores de la misma longitud se pueden representar uno con respecto
% al otro usando la función plot.
% plot(x,y)
% La función plot acepta un argumento adicional que permite especificar el
% color, el estilo de línea y el estilo de marcador utilizando diferentes
% símbolos entre comillas simples.
% plot(x,y,"r--o")
% El comando anterior representa una línea roja (r) de guiones (--) con un
% círculo (o) como marcador.
% ------------------------------------------------------------------------
% Observe que cada comando plot ha creado una gráfica independiente. Para
% representar una línea sobre otra, use el comando hold on para mantener la
% gráfica anterior mientras se agrega otra línea.
% ------------------------------------------------------------------------
% Funciones seno y coseno en el intervalo[0,2π] con un mínimo de 100 puntos
% cada una, línea continua, color verde para tangente, grosor 2, marca x;
% color azul para secante, grosor 2, marca rombo. Graficar sobre el mismo
% sistema de ejes (ver comando hold on)
% ------------------------------------------------------------------------
% La función plot acepta entradas adicionales opcionales que consisten en
% un nombre de propiedad y un valor asociado.
% plot(y,"LineWidth",5)
% El comando anterior representa una línea gruesa.
% ------------------------------------------------------------------------
% Crear cuadrículas
% linspace: Generar un vector espaciado linealmente
% y = linspace(x1,x2) devuelve un vector de fila de 100 puntos
% equidistantes entre x1 y x2.
% y = linspace(x1,x2,n) genera n puntos. El espaciado entre los puntos es
% (x2-x1)/(n-1).
% linspace es similar al operador de dos puntos, ":", pero proporciona
% control directo sobre el número de puntos y siempre incluye los extremos.
% "lin" en el nombre "linspace" se refiere a generar valores espaciados
% linealmente
% Si conoce el número de elementos que desea en un vector (en lugar del
```

.....

```
39      % espaciado entre cada elemento), podría utilizar en su lugar la función
40      % linspace:
41      % linspace(primero,último,número_de_elementos).
42      % Observe el uso de comas (,) para separar las entradas de la función
43      % linspace.
44      % x = linspace(0,1,5)
45      % -------------------------------------------------------------------------
46      clear; clc;
47      % genero un vector espaciado linealmente en el intervalo [0,2π] con 100
48      % puntos
49      x = linspace(0,2*pi,100);
50      A = sin(x);
51      B = cos(x);
52      C = tan(x);
53      % secante
54      D = sec(x);
55      % creamos la grafica
56      plot(x,A,"r--o","LineWidth",1)
57      title('Funciones trigonometricas')
58      % el programa permite etiquetar los ejes
59      hold on
60      plot(x,B,"b--o","LineWidth",1)
61      hold on
62      plot(x,C,"g--x","LineWidth",2)
63      hold on
64      plot(x,D,"b--s","LineWidth",2)
65      xlabel('Variables independientes (x)')
66      ylabel('Variables dependientes (y)')
67      % y agregar títulos
68      % title('Azul sec(x), Verde tan(x)')
69      ylim([-1 1])
70      legend("sin(x)","cos(x)","tan(x)","sec(x)")
71
```

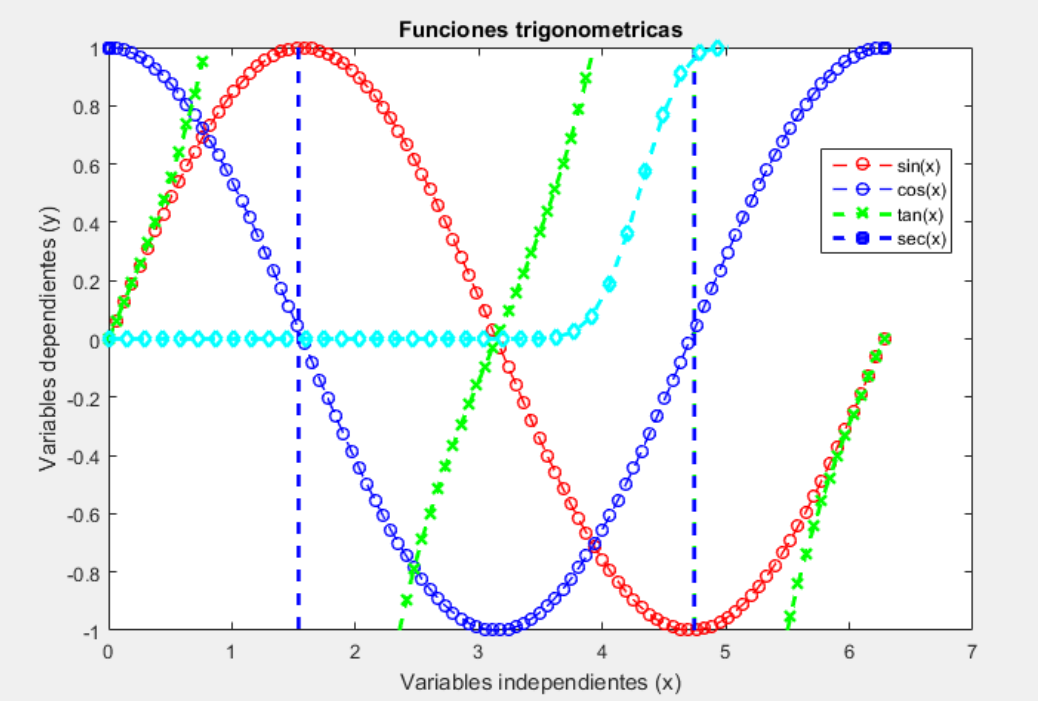


https://github.com/openjuy/ia2022/tree/main/tp01/02a-Graficacion

…..

b) $Y2 = e^{\frac{(x-5)^3}{0.5}}$ Intervalo (0 ; 10), línea de trazos, color cyan, grosor 2, marca rombo. Mínimo 70 puntos.

```
x=linspace(0,10,70);
y2=exp(((x-5).^3)/0.5);
plot(x,y2,'LineStyle','--','Color','c','LineWidth',2,'Marker','d')

```



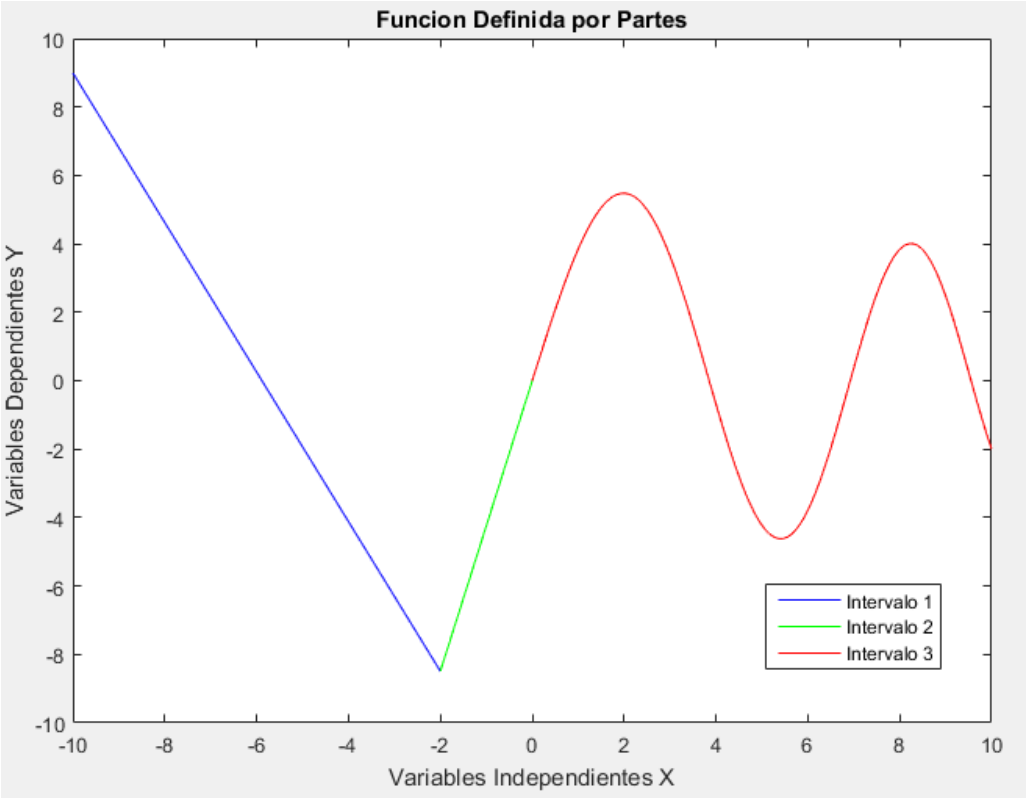
https://github.com/openjuy/ia2022/tree/main/tp01/02b-Graficacion

c) Graficar la función definida por partes que se indica. Cada intervalo debe contener por lo menos 20 puntos. Cada sección debe ser de un color diferente.

$$f(x) = \begin{cases} -2{,}186.x - 12{,}864 & si\ -10 \le x < -2 \\ 4{,}246.x & si\ -2 \le x < 0 \\ 10.e^{(-0.05.x-0{,}5)}.\sin(0.03.x^2 + 0{,}7.x) & si\ 0 \le x < -10 \end{cases}$$

```
x=linspace(-10,-2,80)
plot(x,-2.186*x-12.864,'b')
hold on
x=linspace(-2,0,20)
plot(x,4.246*x,'g')
x=linspace(0,10,100);
y1=10.*exp(-0.05.*x-0.5).*sin(0.03.*(x.^2)+0.7.*x)
plot(x,y1,'r')
```

…..



https://github.com/openjuy/ia2022/tree/main/tp01/02c-Graficacion

3. Series. Generar las series que se indican:

a) Generar una serie de 33 números aleatorios enteros de 3 cifras, en el intervalo [100;200], ordenados de menor a mayor.

```
clear
clc
%Punto 3a
%Crea una Serie de 33 elementos ordenados de menor a mayor enteros positivos
S33=sort(randi([100,200],1,33))
```

```
S33 =

  Columns 1 through 16

   103   103   109   112   114   115   117   127   128   139   142   149   155   163   166   166

  Columns 17 through 32

   168   171   175   176   180   180   182   185   191   192   192   194   196   196   196   197

  Column 33

   198
```

b) Generar una serie S1 en el intervalo [0 ; 2p] con un intervalo de p/3.3 (usar el operador '**:**'). Generar una serie S2 en el mismo intervalo que contenga 7 elementos (usar la función **linspace**). Comparar ambas series y explicar sus diferencias.

…..

```
7       %Punto 3b
8       % Crea una serie numerica entre 0 y 2*pi, con  un intervalo pi/3.3
9 -     S1=[0,pi/3.3,2*pi]
10      % Crea una Serie numerica entre 0 y 2*pi, con un 7 elementos
11 -    S2=linspace(0,2*pi,7)
```

```
S1 =

         0    0.9520    6.2832


S2 =

         0    1.0472    2.0944    3.1416    4.1888    5.2360    6.2832
```

c) Explicar cuál es el problema al intentar generar las siguientes series:

>> [105 : 2.5 : 25]<

El orden del intervalo es el problema, debido a que no es posible ir de 100 a 25 sumando 2.5 a cada elemento, en todo caso el intervalo sería de 25 a 100 con paso 2.5.

>> [-5 : 21 : 9]

```
ans =

    -5
```

En este caso el inconveniente es que la distancia entre elementos es mayor al intervalo [-5 9], por lo que generaría un solo elemento en la serie que será siempre -5.

>> [10 : 10 : 10]

```
ans =

    10
```

En este caso la serie solo va a tener un solo elemento que será el 10 para ampliar la cantidad de elementos,debería cambiar la cota superior a un número mayor y múltiplo de 10.

>> [p1/2 : p1/3 : pi/4]

```
Undefined function or variable 'p1'.
```

…..

Sin importar el valor de p, el problema es el intervalo, no es posible ir de p/2 > p/4 con paso p/3, se debería invertir el intervalo, o bien ir con paso -p/3

>> [x/2 : 4.7 : 25]

```
Undefined function or variable 'x'.
```

En este caso la serie va a depender del valor de x, y para que la misma tenga por lo menos un elemento, x deberá ser menor o igual a 25.

>> [6 : log(1) : 9]

```
Undefined function or variable 'sl'.
```

La presente serie tiene como valor de inicio 6, con paso 0 (como resultado de aplicar logaritmo de 1) y valor final 9. Por lo tanto, dicha serie puede generarse pero está vacía debido a que marca como paso de avance 0.

https://github.com/openjuy/ia2022/blob/main/tp01/03-Series/Punto3.m

4. **Escribir funciones en línea de comando. Detectar e indicar cuál es el problema que ocurre al intentar escribir en Matlab las siguientes funciones, solucionarlo y escribirlas adecuadamente:**

a) >> P1 = a * x^2 + b * x + c

Podemos analizar 2 posibles situaciones en la presente función. Por un lado, se puede considerar que cada variable representa una matriz cuadrada, con lo cual la función cuadrática no representa un problema.

Por otro lado, si cada una de las variables son un valor escalar el producto en la función está mal escrito ya que de esta forma Matlab lo tomaría como un producto vectorial. La forma correcta de ejecutar esta función sería: **P1 = a.* x.^2 + b.* x + c**

b) >> F(A,B,C) = 3 + 9^2 + pi

La presente función "*F*" requiere que se le asigne valores a las variables pasadas por parámetro "*A, B, C*" ya que definida como esta nos da error en Matlab. La forma correcta sería: **F = 3 + 9^2 + pi**

c) >> a=5; b=21; x=[4 5 6]; f1 = a*x+b*x^2

De acuerdo a los valores asignados a las variables que componen la función "*f1*", podemos afirmar que existe un problema en el producto ya que Matlab lo tomara como un producto vectorial, cuando debería ser una producto escalar. Por otro lado, tenemos la potencia que también se

…..

encuentra mal expresada. Para dar solución a esta función se debe escribir la función de la siguiente forma: **f1=a.*x+b.*x^2**

d) >> m=ones(2); n=magic(2); x=[7 8 9]; g1=m^2*x+x^*n

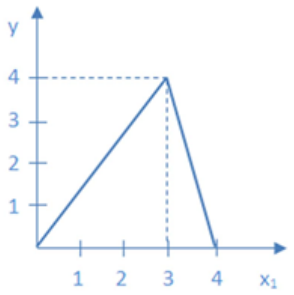
De acuerdo a los valores asignados a las variables de la función "*g1*" podemos asegurar que la ejecución de la misma no puede llevarse a cabo debido a que m y n son matrices cuadradas y x es un vector de 1 fila y 3 columnas. Dado que la función g1 trabaja un producto vectorial entre n y x, es condición que el 1er deba tener tantas columnas como filas tenga el 2do. Ya que esto no se cumple, no puede ejecutarse la función. Para solucionar esté problema, podemos asignar un valor de: **x = [1 2 ;3 4]**

e) ¿para qué sirve el operador '.' y el operador ';'?

El operador **"."** se utiliza para expresar operaciones escalares. El operador **";"** se utiliza para eliminar el eco en una línea de instrucción con lo cual no se muestra el resultado de la operación (pero si se actualizan los valores de las variables involucradas) y salta a la siguiente línea para ingresar la próxima sentencia.

5. Cambio de escala (reescalado).

a) Considerando la gráfica adjunta, escribir sus ecuaciones. Con ellas, generar una secuencia de 40 puntos para cada eje. Reescalar la secuencia de ordenadas (y) al intervalo (0,1). Graficar las secuencias (original y reescalada) sobre el mismo sistema de ejes. Calcular el centro de gravedad de ambas gráficas.



…..

```
clc,clear;
x1=linspace(0,3,30);
x2=linspace(3,4,10);
y1=4*x1/3;
y2=16-4*x2;
y_1=mapminmax(y1,0,1);
y_2=mapminmax(y2,0,1);

plot(x1,y1,'b');
hold on;
plot(x2,y2,'b');
plot(x1,y_1,'r');
plot(x2,y_2,'r');
xlim([0 4.5])
ylim([0 4.5])
grid
grid on

polyin1 = polyshape(x1,y1);
[CX1,CY1] = centroid(polyin1);
plot(CX1,CY1,'g');
hold on;
```

Nota: la función “polyshape” se reconoce y se ejecuta en una versión de Matlab 2018, la versión de Matlab2015b da error.

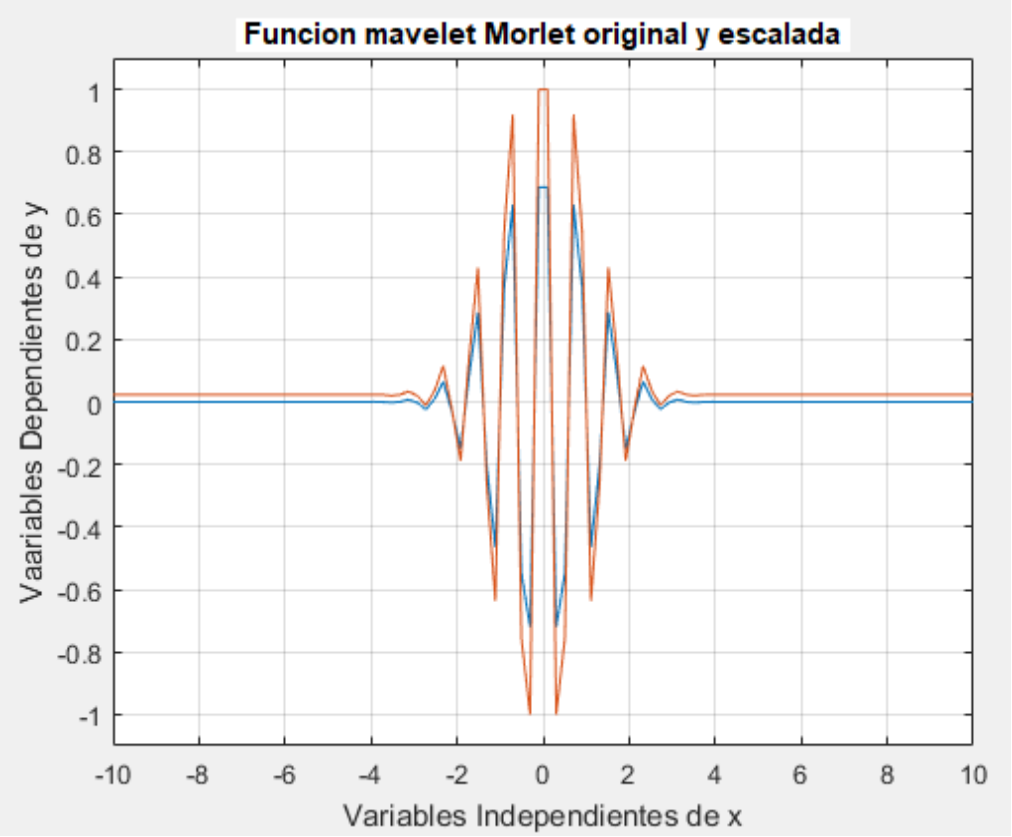
b) La ecuación siguiente representa a una función wavelet Morlet. Graficarla en el intervalo (-10,10) con un mínimo de 100 puntos.

$$y(x) = cos(8.x)e^{\frac{x^2}{2}}$$

Reescalar la secuencia (*sólo de ordenadas*) al intervalo (-1,0) y graficar conjuntamente con la función original.

…..

```
clc, clear;
x=linspace(-10,10,100);
y=cos(8*x).*exp((-x.^2)/2);

plot(x,y)
hold on

Y1=mapminmax(y,-1.0);

plot(x,Y1)
ylim([-1.1 1.1])

grid on
```



https://github.com/openjuy/ia2022/blob/main/tp01/05b-Reescalado/mavelet.m

6. Ruido. Sobre la secuencia temporal generada por la ecuación:

y = -10 * sin(3*t) - 5

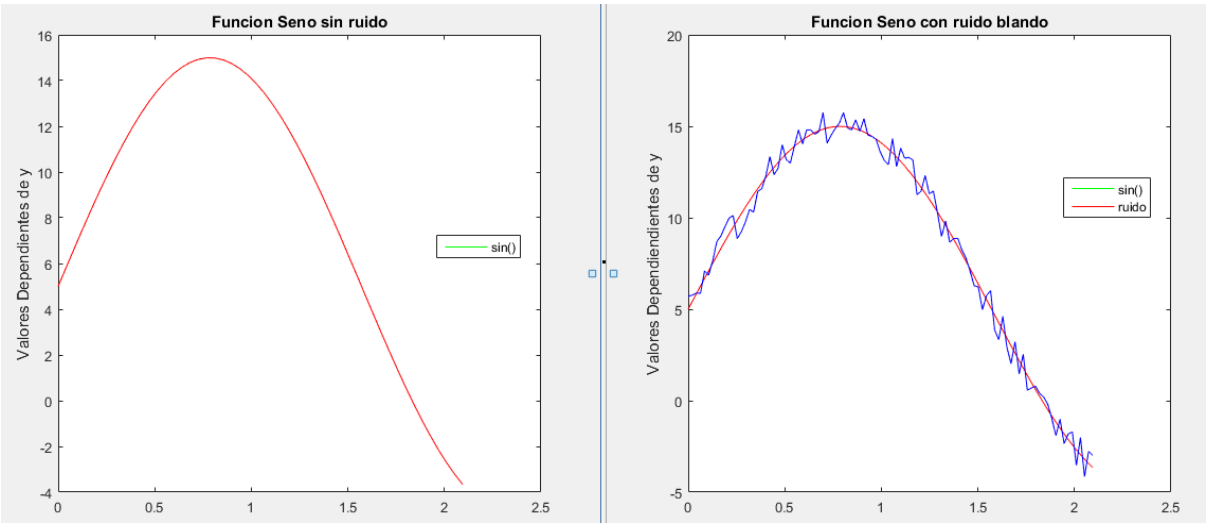
a) Obtener 50 puntos para **t** perteneciente a [0,120º]. *Verificar en qué unidades trabaja* **sin()**.

…..

```
1
2       x_1 = linspace(0,2/3*pi,50);
3       y_1 = 10*sin(2*x_1)+5;
4
5       x_2 = linspace(0,2/3*pi,100);
6       y_2 = 10*sin(2*x_2)+5 - 1 + 2 * rand(1,100);
7
8       plot(x_1, y_1, 'r');
9       hold on;
10      plot(x_2, y_2, 'b');
11
```

b) Agregar, a la secuencia, ruido blanco con una amplitud máxima (positiva o negativa) del 10% de la sinusoide de base (quitando la componente de continua → 5).

c) Graficar la señal sin ruido en rojo y superponer la señal con ruido en azul.



d) Calcular para ambas secuencias, la media (μ) y la desviación estándar (σ).

Orientación: La ventana de gráficas contiene el menú (click derecho sobre la curva), para modificar la mayoría de los parámetros.

```
12      %Media y Desviacion Standard de seno
13      mean(y_1)
14      std(y_1)
15
16      %Media y Desviacion Standard del ruido
17      mean(y_2)
18      std(y_2)
```

…..

```
>> %Media y Desviacion Standard de seno
mean(y_1)
std(y_1)

ans =

    8.4206


ans =

    5.8026
```

```
>> %Media y Desviacion Standard del ruido
mean(y_2)
std(y_2)

ans =

    8.5200


ans =

    5.7576
```

https://github.com/openjuy/ia2022/blob/main/tp01/06-Ruido/punto6.m

7. Scripts.

a) Se tiene una matriz de 10x10 de valores binarios (1 bit por celda). Escribir un script que reemplace los ‘1’ por ‘0’ y los ‘0’ por ‘-1’.

```
clear
clc
Matriz= randi([0 1],10);
B = Matriz;

for i=1:10
    for j=1:10
        if Matriz(i,j)==1
           Matriz(i,j)=0;
        else
            Matriz(i,j)=-1;
        end
    end
end

```

.....

```
B =

     0     1     0     1     0     1     1     1     1     0
     0     1     1     1     0     0     0     0     1     1
     1     0     0     1     1     0     1     0     0     1
     0     1     0     0     1     0     0     0     0     0
     0     0     1     1     0     0     0     0     1     1
     0     1     1     1     0     1     1     0     0     0
     1     1     1     1     0     1     1     0     1     1
     1     0     1     0     1     1     0     1     1     1
     1     0     0     0     1     1     0     0     1     1
     0     1     1     1     1     1     1     0     0     0
```

```
Matriz =

    -1     0    -1     0    -1     0     0     0     0    -1
    -1     0     0     0    -1    -1    -1    -1     0     0
     0    -1    -1     0     0    -1     0    -1    -1     0
    -1     0    -1    -1     0    -1    -1    -1    -1    -1
    -1    -1     0     0    -1    -1    -1    -1     0     0
    -1     0     0     0    -1     0     0    -1    -1    -1
     0     0     0     0    -1     0     0    -1     0     0
     0    -1     0    -1     0     0    -1     0     0     0
     0    -1    -1    -1     0     0    -1    -1     0     0
    -1     0     0     0     0     0     0    -1    -1    -1
```

https://github.com/openjuy/ia2022/blob/main/tp01/07a-Scripts/punto7a.m

b) Escribir un script para ejecutar en forma automática el problema 3.a) anterior. La cantidad de números de la serie debe ser introducida por el usuario (función **input**), lo mismo que la cantidad de cifras de los números y el intervalo de la serie.

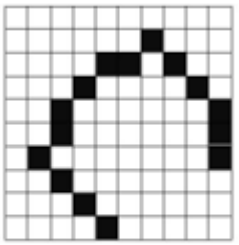
```
1  -    clc,clear;
2  -    prompt = 'Indique el numero mas grande de la serie: ';
3  -    maxserie=input(prompt);
4  -    prompt = 'Indique el numero mas pequeño de la serie: ';
5  -    minserie=input(prompt);
6  -    prompt = 'Indique la cantidad de numeros que tendra la serie: ';
7  -    numeros=input(prompt);
8
9  -    serie=round(minserie + (maxserie-minserie-1).*rand(1,numeros));
10 -    serieasc=sort(serie);
11
12 -    'La serie original es: ';
13 -    serie
14 -    'La serie ordenada de menor a mayor es: ';
15 -    serieasc
16
```

…..

```
ue el numero mas grande de la serie: 150
ue el numero mas pequeño de la serie: 120
ue la cantidad de numeros que tendra la serie: 33


rie original es:

 =

umns 1 through 19

5   140   146   148   136   124   124   127   144   127   144   127   147   130   126   127   138   134   130

umns 20 through 33

4   137   136   147   128   142   142   131   136   122   122   135   143   147


rie ordenada de menor a mayor es:

asc =

umns 1 through 19

2   122   124   124   126   127   127   127   127   128   130   130   131   134   135   135   136   136   136

umns 20 through 33

7   138   140   142   142   143   144   144   144   146   147   147   147   148
```

https://github.com/openjuy/ia2022/blob/main/tp01/07b-Scripts/punto7b.m

8. Script 2. Dada la siguiente matriz booleana, diseñar un script que:



Blanco = '0'

a) Genere 15 matrices booleanas, del mismo tamaño, conteniendo cada una 15 "unos" dispuestos en forma aleatoria.
b) Compare las 15 matrices con la matriz de muestra e identifique las matrices de mayor y menor similitud.
c) Por cada comparación realizada genere un índice de similitud (IS ∈ ℜ) entre 0 y 1 (un valor 1 significa que las matrices son iguales). *Orientación: investigar el índice de Jaccard.*
d) Re diseñe el script para generar 15 matrices con escala de grises. Conviértelas a booleanas con 1 si la celda posee un gris >= 75%, 0 si la celda contiene un gris <15% y un valor aleatorio para los restantes casos.

https://github.com/openjuy/ia2022/blob/main/tp01/08-Scripts/create_pattern_matrix.m

https://github.com/openjuy/ia2022/blob/main/tp01/08-Scripts/create_random_matrix.m

…..

https://github.com/openjuy/ia2022/blob/main/tp01/08-Scripts/compare_matrix.m

https://github.com/openjuy/ia2022/blob/main/tp01/08-Scripts/create_grey_based_random_matrix.m

9. Función 1. Identificar qué hace la función que sigue. Previamente deben ser encontrados y corregidos dos errores en el código.

```
function p = diagonal(matriz,valor)
FilCol=size(matriz);
if (valor~=0)&&(valor~=1)
    error('El 2do parámetro debe ser 0 o 1')
end
if FilCol(1)~=FilCol(2)
    error('La matriz debe ser cuadrada')
end
for i=1:FilCol(1)
     for j=1:FilCol(2)
          if i<j || i>j
             matriz(i,j)=valor;
          end
     end

p=matriz;
```

La presente función presenta 2 errores en la sintaxis de propia de la estructura utilizada. El primero es la finalización de la 1era estructura “for-end” y el segundo error es la finalización de la función p. Se procede a corregir estos error y se obtiene el siguiente código:

…..

```
function p = diagonal(matriz,valor)

FilCol=size(matriz);

if (valor~=0)&&(valor~=1)
    error('El 2do parámetro debe ser 0 o 1')
end

if FilCol(1)~=FilCol(2)
    error('La matriz debe ser cuadrada')
end

for i=1:FilCol(1)
    for j=1:FilCol(2)
        if i<j || i>j
            matriz(i,j)=valor;
        end
    end
end

p=matriz;

end
```

```
>> diagonal([2 4; 6 8], 1)

ans =

     2     1
     1     8
```

10. Función 2. Escribir una función con las características indicadas:

a) Debe generar una matriz A (dimensión 15x15) de números enteros aleatorios en el rango [-1, 1].
b) Debe encontrar todas las ocurrencias de una submatriz B 3x2, dentro de la matriz A, dada por el usuario, como argumento de la función.
c) La función devolverá las posiciones (fila y columna) de las submatrices encontradas en un vector de nx2.
d) Si no se encuentra la submatriz B, deberá buscarse su versión transpuesta.
e) La función debe verificar que la submatriz ingresada sea de 3x2 y que sus valores estén en el rango indicado para A.

.....

```
function [ x ] = punto10(matrix)

clear;clc;
rng('shuffle');
C= randi([-1,1],15,15)

D=fMdxd();
MatrixEncontrada(C,D);

function B=fMdxd()
    A=zeros(2);
    disp('Ingrese la matrix 3x2 \n');
    a=input('ingrese el primer valor=');
    b=input('ingrese el segundo valor=');
    c=input('ingrese el tercer valor=');
    d=input('ingrese el cuarto valor=');
    e=input('ingrese el quinto valor=');
    f=input('ingrese el sexto valor=');

    A(1,1)=a;
    A(1,2)=b;
    A(2,1)=c;
    A(2,2)=d;
    A(3,1)=e;
    A(3,2)=f;
    B=A;
end

function y= MatrixEncontrada(A,B)
  valor(1)=0;
  valor(2)=0;
    for i=1:(length(A)-1)
        for j=1:(length(A)-1)
            if A(i,j)==B(1,1)& A(i,j+1)==B(1,2)& A(i+1,j)==B(2,1)&A(i+1,j+1)==B(2,2)& A(i+1,j)==B(3,1)& A(i+1,j)==B(3,2)
            fprintf('Se encontro la submatriz en la fila %d y columna %d \n',i,j);
            valor(1)=i;
            valor(2)=j;
            end
        end
    end
    if (valor(1)== 0)
    disp('No se encontro la submatriz con la posicion ingresada ');
    end
  y=valor;
end

end
```

https://github.com/openjuy/ia2022/blob/main/tp01/10-Funciones/punto10.m