TEMA 5. PROGRAMACIÓN BÁSICA EN MATLAB ®/OCTAVE

5.1. Introducción a Matlab y Octave

5.2. Entrada y salida con formato

5.3. Programas: script y funciones

5.4. Estructuras alternativas o condicionales

5.5. Estructuras repetitivas o bucles

5.6. Aplicación de MATLAB ® / Octave a casos prácticos de ingeniería

• LAS SECCIONES DEL CÓDIGO DE LOS PROGRAMAS DE CÓMPUTO SE PUEDEN CATEGORIZAR EN UNA DE ESTAS TRES ESTRUCTURAS: SECUENCIAS, ESTRUCTURAS ALTERNATIVAS Y ESTRUCTURAS DE REPETICIÓN

• LAS SECUENCIAS SON LISTAS DE COMANDOS QUE SE EJECUTAN UNA DESPUÉS DE OTRA.

• UNA ESTRUCTURA ALTERNATIVA PERMITE AL PROGRAMADOR EJECUTAR UN COMANDO (O CONJUNTO DE COMANDOS) SI ALGÚN CRITERIO ES VERDADERO Y UN SEGUNDO COMANDO O CONJUNTO CUANDO ES FALSO. PARA ESTO SE EMPLEAN CONDICIONES LÓGICAS QUE SON EVALUADAS MEDIANTE OPERADORES RELACIONALES Y LÓGICOS.

• UNA ESTRUCTURA REPETITIVA O BUCLE PERMITE QUE UN GRUPO DE ENUNCIADOS SE EJECUTE VARIAS VECES. EL NÚMERO DE VECES QUE SE EJECUTA DEPENDE DE UN CONTADOR O DE LA EVALUACIÓN DE UNA CONDICIÓN LÓGICA.

OPERADORES RELACIONALES Y LÓGICOS

- **LAS ESTRUCTURAS ALTERNATIVAS Y DE REPETICIÓN DEPENDEN DE OPERADORES RELACIONALES Y LÓGICOS.**
- **OCTAVE TIENE LOS SIGUIENTES OPERADORES:**

RELACIONALES

Operador de relación	Interpretación
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Igual que
~=	Distinto que

Comparación		Valor
Verdadero	SI	1
Falso	NO	0

RESPUESTA EN OCTAVE

LÓGICOS

Operador lógico	Símbolo
no	~
y	&
o	|

TABLA DE LA VERDAD

A	B	~A	A|B	A&B
falso	falso	verdadero	falso	falso
falso	verdadero	verdadero	verdadero	falso
verdadero	falso	falso	verdadero	falso
verdadero	verdadero	falso	verdadero	verdadero

Ejemplo:

operadores.m

```
% EJEMPLOS PARA OPERADORES LÓGICOS

% CON ESCALARES OP.RELACIONAL
x=5;
y=1;
x<y

% CON VECTORES OP.RELACIONAL
x=1:5;
y=x-4;
x<y

% CON MATRICES OP.RELACIONAL
x=[1,2,3,4,5];
y=[-2,0,2,4,6];
x<y

% OP.RELACIONALES + LÓGICOS
x=[1,2,3,4,5];
y=[-2,0,2,4,6];
z=[8,8,8,8,8];

z>x & z>y % Z ES MAYOR QUE X Y MAYOR QUE Y

x>y | x>z % X ES MAYOR QUE Y O MAYOR QUE Z
```

```
octave-3.2.4.exe:15> operadores
ans = 0
ans =

   0   0   0   0   0

ans =

   0   0   0   0   1

ans =

   1   1   1   1   1

ans =

   1   1   1   0   0

octave-3.2.4.exe:16>
```

FUNCIONES LÓGICAS

OCTAVE OFRECE TANTO LAS ESTRUCTURAS ALTERNATIVAS TRADICIONALES (IF) COMO UNA SERIE DE FUNCIONES LÓGICAS QUE REALIZAN LA MISMA TAREA.

find →

SE USA TANTO EN ESTRUCTURAS ALTERNATIVAS COMO EN REPETITIVAS.
SIRVE PARA IDENTIFICAR QUE ELEMENTOS DE UNA MATRIZ SATISFACEN UN CRITERIO DADO.

find(nombre_vector<>=valor)

Ejemplo:

```
altura=[63,67,65,72,69,78,75]   % VECTOR DE DATOS DE ALTURA
aceptados=find(altura>=66)      % CON find BUSCAMOS AQUELLOS QUE SON > O = A 66

aceptados=2  4  5  6  7         % PROPORCIONA LAS POSICIONES DEL VECTOR QUE
                                % SON MAYORES O IGUALES A 66
altura (aceptados)              % COMANDO PARA SABER EL DATO DE CADA UNO
ans= 67  72  69  78  75
```

Ejemplo 17a. Usando el comando find ,y tras una serie de alturas que corresponden con diferentes personas, mostrar aquellas cuya altura es mayor o igual a 66 y las que no lo son.

Ejemplo17_T5.m

```
% EJEMPLO PARA VER LA APLICACION DEL COMANDO find

alturas=[63,67,65,72,69,78,75];

% BUSCAMOS LOS ACEPTADOS (ALTURA>=66)
aceptados=find(alturas>=66);

alturas(aceptados);

% BUSCAMOS LOS NO ACEPTADOS (ALTURA<66)
no_aceptados=find(alturas<66);

alturas(no_aceptados);

% LOS MOSTRAMOS POR PANTALLA
disp('Los siguientes candidatos satisfacen el requisito de estatura')

fprintf('Candidato %4.0f mide %4.1f pulgadas de alto\n',...
[aceptados;alturas(aceptados)])

disp('Los siguientes candidatos no satisfacen el requisito de estatura')

fprintf('candidato %4.0f mide %4.1f pulgadas de alto\n',...
[no_aceptados;alturas(no_aceptados)])
```

VALORES DE ALTURA

OBTENEMOS LA POSICIÓN DE LAS PERSONAS >=66

VECTOR CON LOS VALORES DE LAS ALTURAS

IGUAL CON ALTURAS <66

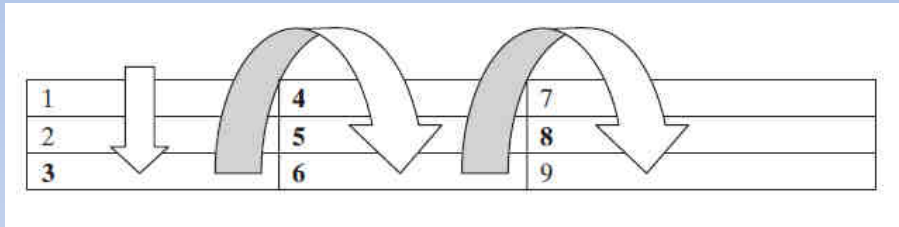
Ejemplo 17b.

```
15  % LOS MOSTRAMOS POR PANTALLA
16  disp('Los siguientes candidatos satisfacen el requisito de estatura')
17
18  fprintf('Candidato %4.0f mide %4.1f pulgadas de alto\n',...
19  [aceptados;alturas(aceptados)])
20
21  disp('Los siguientes candidatos no satisfacen el requisito de estatura')
22
23  fprintf('candidato %4.0f mide %4.1f pulgadas de alto\n',...
24  [no_aceptados;alturas(no_aceptados)])
25
```

fprintf EMPLEANDO MATRICES

```
octave-3.2.4.exe:23> Ejemplo17_T5
Los siguientes candidatos satisfacen el requisito de estatura
Candidato    2 mide 67.0 pulgadas de alto
Candidato    4 mide 72.0 pulgadas de alto
Candidato    5 mide 69.0 pulgadas de alto
Candidato    6 mide 78.0 pulgadas de alto
Candidato    7 mide 75.0 pulgadas de alto
Los siguientes candidatos no satisfacen el requisito de estatura
candidato    1 mide 63.0 pulgadas de alto
candidato    3 mide 65.0 pulgadas de alto
octave-3.2.4.exe:24>
```

find → **CUANDO SE EMPLEA ESTE COMANDO CON MATRICES HAY QUE TENER EN CUENTA QUE EL ORDEN PARA REALIZAR LA BÚSQUEDA DE AQUELLOS ELEMENTOS QUE CUMPLEN LAS CONDICIONES ES:**



Ejemplo 18a.

Disponemos de la siguiente matriz de valores de temperatura de diferentes puntos de una instalación. Cada columna representa un termopar diferente y las filas un punto de la instalación. Desarrollar un programa en Octave que indique aquellos puntos cuya temperatura sea mayor que 98.6 ºC.

	T1	T2	T3
Punto 1	95.3	100.2	98.6
Punto 2	97.2	99.2	98.9
Punto 3	100.1	99.3	97

Ejemplo 18b.

Ejemplo18_T5.m

```
% MATRIZ DE VALORES DE TEMPERATURA

temp=[95.3,100.2,98.6;97.4,99.2,98.9;100.1,99.3,97]

% BUSCAMOS LOS PUNTOS DE TEMPERATURA SUPERIOR A 98.6
% EL RESULTADO SERÁ UN VECTOR CON LAS POSICIONES COLUMNA A COLUMNA

puntos_calientes=find(temp>98.6)

% PARA TENER POR SEPARADO QUE VALORES DE FILAS Y COLUMNAS SUPERAN LA TEMPERATURA
% DAMOS DOS VARIABLES AL COMANDO find

[fila,col]=find(temp>98.6)

% GENERAMOS LA MATRIZ DE RESULTADOS (TRANSPONIENDO LOS VECTORES OBTENIDOS)

resultado=[fila,col,temp(puntos_calientes)]'

% DAMOS LOS RESULTADOS POR PANTALLA

fprintf('Los puntos %4.0f de los termopares %4.0f tuvieron una temperatura de %8.1f\n',...
resultado)

```

Ejemplo 18c.

```
temp =

    95.300   100.200    98.600
    97.400    99.200    98.900
   100.100    99.300    97.000

puntos_calientes =

   3
   4
   5
   6
   8

fila =

   3
   1
   2
   3
   2

col =
   1
   2
   2
   2
   3

resultado =

     3.0000     1.0000     2.0000     3.0000     2.0000
     1.0000     2.0000     2.0000     2.0000     3.0000
   100.1000   100.2000    99.2000    99.3000    98.9000

Los puntos     3 de los termopares     1 tuvieron una temperatura de     100.1
Los puntos     1 de los termopares     2 tuvieron una temperatura de     100.2
Los puntos     2 de los termopares     2 tuvieron una temperatura de      99.2
Los puntos     3 de los termopares     2 tuvieron una temperatura de      99.3
Los puntos     2 de los termopares     3 tuvieron una temperatura de      98.9
lines 18-41/41 (END) -- (f)orward, (b)ack, (q)uit
```

Ejemplo 19a.

USO DEL COMANDO find CON MATRICES Y CUMPLIMIENTO DE VARIAS CONDICIONES

Dados los siguientes valores de estatura (pulgadas) y edad (años) encontrar, empleando el comando find, aquellos que cumplen que la altura es >=66 y la edad entre 18 y 35.

Estatura (pulg)	Edad (años)
63	18
67	19
65	18
72	20
69	36
78	34
75	12

Ejemplo 19b.

Ejemplo18_T5.m

```
% BUSCAR AQUELLAS PERSONAS QUE SU ALTURA SEA MAYOR QUE 66 Y
% SU EDAD ENTRE 18 Y 35

%CREAMOS LA MATRIZ CON LOS DATOS

participantes=[63 18;67 19;65 18;72 20;69 36;78 34;75 12];

%BUSCAMOS LOS QUE CUMPLEN LOS CRITERIOS EN CADA COLUMNA (:,Nº COLUMNA)

aceptados=find(participantes(:,1)>=66 & participantes(:,2)>=18&...
participantes(:,2)<35)

% HAGO UNA MATRIZ DE RESULTADOS

resultado=[aceptados,participantes(aceptados,1),...
participantes(aceptados,2)]'

% LOS MUESTRO POR PANTALLA

fprintf('Participante %4.0f mide %4.0f pulgadas de alto y tiene %4.0f años de edad\n',...
resultado)
```

(:,1) SIRVE PARA SABER SI CUMPLE LAS CONDICIONES LA COLUMNA 1

CREAMOS LA MATRIZ DE RESULTADOS (TRANSPONER) Y LOS MOSTRAMOS POR PANTALLA

Ejemplo 19c.

```
octave-3.2.4.exe:27>
octave-3.2.4.exe:27> Ejemplo18_I5
aceptados =

   2
   4
   6

resultado =

    2    4    6
   67   72   78
   19   20   34

Participante    2 mide   67 pulgadas de alto y tiene   19 a±os de edad
Participante    4 mide   72 pulgadas de alto y tiene   20 a±os de edad
Participante    6 mide   78 pulgadas de alto y tiene   34 a±os de edad
octave-3.2.4.exe:28>
```

FUNCIONES LÓGICAS

any → AL SER APLICADA A UN VECTOR O UNA MATRIZ DEVUELVE UN ESCALAR QUE INDICA SI ALGUNO DE SUS ELEMENTOS ES CERO

Ejemplo:

```
a=[1 0 2 3 4 0]
any(a)
ans=2
```

all → AL SER APLICADA A UN VECTOR O UNA MATRIZ DEVUELVE UN ESCALAR QUE INDICA SI ALGUNO DE SUS ELEMENTOS ES DISTINTO DE CERO

Ejemplo:

```
a=[1 0 2 3 4 0]
all(a)
ans=4
```

FUNCIONES LÓGICAS

any **all**

ESTAS FUNCIONES SON MUY ÚTILES EN COMBINACIÓN CON OPERADORES LÓGICOS Y RELACIONALES.

Ejemplo: Si queremos saber si el vector a tiene algún valor negativo lo hacemos con la orden *any(a<0)* (¿hay algún número menor que cero en el vector a?).

Si nos interesa saber si todos sus elementos son negativos escribiríamos *all(a<0)* (¿todos los componentes de a son menores que cero?)

a=[1 0 2 3 4 0]
all(a<0)
ans=0

ESTRUCTURAS ALTERNATIVAS

- COMO TODO LENGUAJE DE PROGRAMACIÓN, OCTAVE DISPONE DE INSTRUCCIONES DE BIFURCACIÓN QUE PERMITEN EL CONTROL DEL FLUJO DE ORDENES DE UN PROGRAMA.

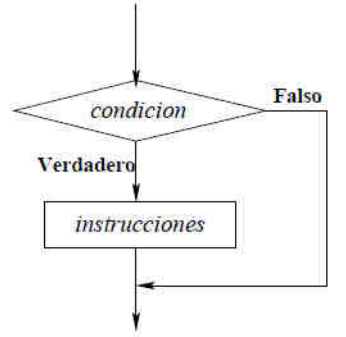
- DISPONE DE LAS SIGUIENTES ESTRUCTURAS:
  - ✓ if
  - ✓ if…else
  - ✓ if…elseif…else
  - ✓ switch-case
  - ✓ Otras funciones lógicas especificas de Octave (find)

- LA APLICACIÓN DIRECTA DE OPERADORES LÓGICOS A VECTORES Y MATRICES RESULTA MUCHO MÁS POTENTE QUE EN OTROS LENGUAJES, EN LOS QUE ESTA POSIBILIDAD NI TAN SI QUIERA EXISTE.

ESTRUCTURAS ALTERNATIVA: if

- LA INSTRUCCIÓN if SIMPLE NOS PERMITE COMPROBAR SI SE CUMPLE UNA CIERTA CONDICIÓN ANTES DE EJECUTAR UNA SERIE DE ORDENES.
- UN ENUNCIADO if SIMPLE TIENE LA SIGUIENTE FORMA:

```
if condición
    instrucciones
end
```



- OCTAVE EVALÚA SI LA CONDICIÓN ES CIERTA Y EN ESE CASO EJECUTA LAS INSTRUCCIONES. SI LA CONDICIÓN ES FALSA, LAS INSTRUCCIONES NO SE EJECUTAN.
- LA INSTRUCCIÓN if SIMPLE FUNCIONA BIEN CUANDO SE TRABAJA CON ESCALARES. SI TRABAJAMOS CON UN VECTOR O MATRIZ, SÓLO SE EJECUTARÁN LAS ACCIONES SI LA CONDICIÓN SE CUMPLE PARA TODOS LOS ELEMENTOS QUE LO COMPONEN.

Ejemplo 20.

Ejemplo20_Tema5.m

```
% EJEMPLO DE LA ESTRUCTURA ALTERNATIVA

G=input('Introduzca un numero: ');

if (G<50)
disp(G)
end

if (G<50)
    disp(G)
end

if (G<50)
end
```

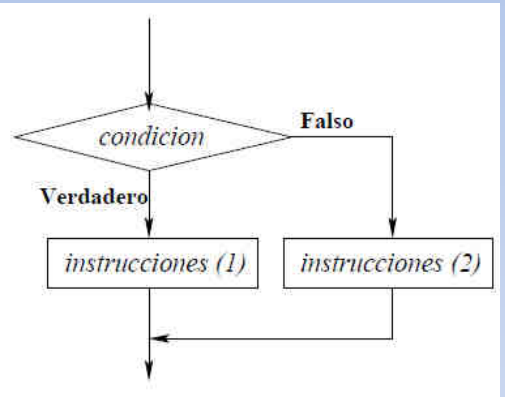
LA ESTRUCTURA if PUEDE ESCRIBIRSE DE DIFERENTES FORMAS. PARA MAYOR FACILIDAD DE CÓDIGO UTILIZAR TABULACIONES

```
octave-3.2.4.exe:47> Ejemplo20_Tema5
Introduzca un numero: 23
 23
 23
 23
octave-3.2.4.exe:48>
```

ESTRUCTURAS ALTERNATIVA: if…else

- ES IGUAL QUE LA ESTRUCTURA if SALVO QUE LA CLAUSULA else HACE QUE SI LA CONDICIÓN ES VERDADERA SE EJECUTEN UNAS ACCIONES Y SINO SE LLEVEN A CABO OTRO CONJUNTO DE ACCIONES.

```
if condición
        instrucciones
else
        instrucciones
end
```



- SI SE TRABAJA CON MATRICES, AL IGUAL QUE CON if, LAS INSTRUCCIONES SE EJECUTAN SÓLO CUANDO LA CONDICIÓN SE CUMPLE SOBRE TODOS LOS ELEMENTOS DE LA MATRIZ O VECTOR.

Ejemplo 21.

NO DA ERROR DE EJECUCIÓN SI NO PONEMOS ENTRE PARÉNTESIS LA CONDICIÓN

Ejemplo21_T5.m

```
% EJEMPLO DE UNA ESTRUCTURA if...else

x=input('Introduzca un numero: ');

if x>0
    y=log(x);
    fprintf('El logaritmo neperiano del numero es %0.3f\n',y)
else
    beep
    disp('El numero debe ser positivo')
end
```

beep PROPORCIONA UN SONIDO EN EL ORDENADOR CUANDO NO CUMPLE LA CONDICIÓN Y VA A LAS INSTRUCCIONES DEL else

```
octave-3.2.4.exe:54> Ejemplo21_T5
Introduzca un numero: 5
El logaritmo neperiano del numero es 1.609
octave-3.2.4.exe:55> Ejemplo21_T5
Introduzca un numero: -2
El numero debe ser positivo
octave-3.2.4.exe:56>
```

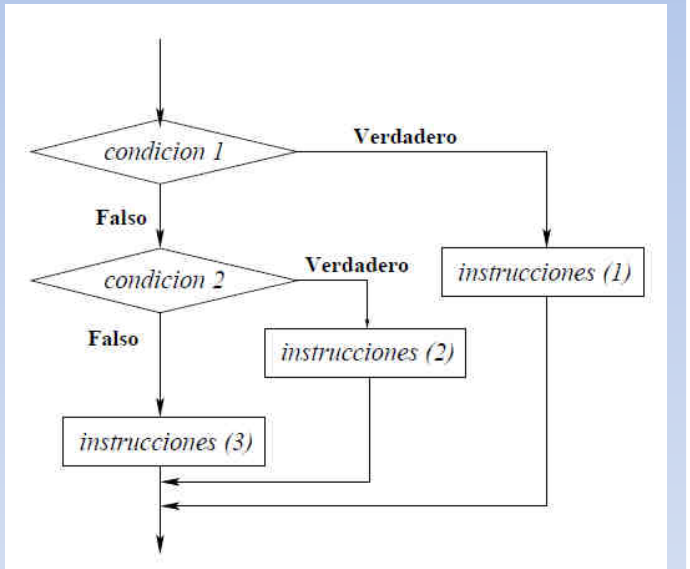
SE EJECUTAN ACCIONES DE if

SE EJECUTAN ACCIONES DE else

ESTRUCTURAS ALTERNATIVA: elseif…

• CUANDO EL NÚMERO DE CONDICIONES QUE DEBEMOS COMPROBAR ES MAYOR QUE UNO, LO CUAL NOS OBLIGARÍA A UTILIZAR CONDICIONES if…else ANIDADAS, RESULTA MÁS CONVENIENTE UTILIZAR LA CONDICIÓN elseif.

```
if condición1
      instrucciones1 (si condicion 1 es VERDADERA)
elseif condicion2
      instrucciones2 (si condicion 2 es VERDADERA)
….
else
     instrucciones (si todas las anteriores son falsas)
end
```



• LA ORDEN elseif PUEDE REPETIRSE TANTAS VECES COMO SE QUIERA DENTRO DE UNA ESTRUCTURA if.

Ejemplo 22. Un reactor químico debe funcionar entre 90 y 100 ºC. Por encima de 100, el reactor entra en situación crítica, peligrando la seguridad de la planta; entre 50 y 90 grados, el reactor funciona aunque su rendimiento no es óptimo, mientras que por debajo de 50 el funcionamiento no es correcto, debiéndose desconectar el equipo.

Escribir un programa que pida al usuario la temperatura del reactor y de la información del estado del mismo.

Ejemplo22_T5.m

```
% PROGRAMA SOBRE INFORMACIÓN DEL ESTADO DEL REACTOR

temp=input('Introduzca la temperatura del reactor: ');

if temp>100
    disp('Demasiado caliente-Peligro de explosion')
elseif temp>90
    disp('Temperatura dentro de los limites')
elseif temp>50
    disp('Temperatura por debajo del limite adecuado')
else
    disp('Desconectar el reactor, T muy baja')
end
```

SEGÚN LA TEMPERATURA INTRODUCIDA POR EL USUARIO, EL MENSAJE POR PANTALLA ES DIFERENTE

```
octave-3.2.4.exe:57> Ejemplo22_T5
Introduzca la temperatura del reactor: 95
Temperatura dentro de los limites
octave-3.2.4.exe:58> Ejemplo22_T5
Introduzca la temperatura del reactor: 45
Desconectar el reactor, T muy baja
octave-3.2.4.exe:59> Ejemplo22_T5
Introduzca la temperatura del reactor: 110
Demasiado caliente-Peligro de explosion
octave-3.2.4.exe:60> Ejemplo22_T5
Introduzca la temperatura del reactor: 65
Temperatura por debajo del limite adecuado
octave-3.2.4.exe:61>
```

ESTRUCTURAS ALTERNATIVA: elseif…

- OCTAVE ANALIZA SI ES VERDADERA LA PRIMERA CONDICIÓN, SI NO LO ES SIGUE CON LAS SIGUIENTES HASTA QUE ENCUENTRA LA PALABRA else.

- SI TRABAJAMOS CON MATRICES, LA CONDICIÓN DEBE SER VERDADERA PARA TODO EL ELEMENTO DE LA MATRIZ.

Ejemplo 23. La siguiente matriz nos indica la edad de una serie de conductores:

edad=[15,17,25,55,75]

El siguiente código evalúa si se emite o no la licencia de conducir en base a la edad del solicitante:

NO TODOS LOS ELEMENTOS SON VERDADEROS EN NINGUNA DE LAS CONDICIONES

```
if edad<18
    disp('Los siento, tendrá que esperar')
elseif edad >18 y edad < 70
    disp('Puede obtener un permiso para conducir')
else
    disp('Los conductores mayores de 70 requieren una licencia especial')
end
```

Ejemplo 24a. Crear una función para determinar calificaciones de examen y suponga una sola entrada a la función. las calificaciones se basas en los siguientes criterios:

- **Calificación A: promedio de puntos de 90 a 100.**
- **Calificación B: promedio de puntos de 80 a 90.**
- **Calificación C: promedio de puntos de 70 a 80.**
- **Calificación D: promedio de puntos de 60 a 70.**
- **Calificación E: <60.**

EL ARCHIVO .m DEBE LLAMARSE IGUAL QUE LA FUNCIÓN

puntos.m

```
%FUNCION PARA MOSTRAR LA CLASIFICACÍON EN BASE A LOS PUNTOS
%OBTENIDOS POR EL ALUMNO
%LA FUNCION TIENE UNA SOLA ENTRADA Y DEBE SER ESCALAR

function calificacion=puntos(x)

if x>=90
    calificacion='A';
elseif x>=80
    calificacion='B';
elseif x>=70
    calificacion='C';
elseif x>=60
    calificacion='D';
else
    calificacion='E';
end
```

```
octave-3.2.4.exe:62>
octave-3.2.4.exe:62> puntos(25)
ans = E
octave-3.2.4.exe:63> puntos(80)
ans = B
octave-3.2.4.exe:64> puntos(-52)
ans = E
octave-3.2.4.exe:65> puntos(108)
ans = A
```

CON PUNTOS <0 Y >100 LA FUNCIÓN DA UN RESULTADO (ERROR)

Ejemplo 24b. Debemos modificar la función de manera que no de ninguna respuesta de calificación cuando los puntos sean por encima de 100 o por debajo de 0.

puntos.m | puntosb.m

```
%FUNCION PARA MOSTRAR LA CLASIFICACIÒN EN BASE A LOS PUNTOS
%OBTENIDOS POR EL ALUMNO
%LA FUNCION TIENE UNA SOLA ENTRADA Y DEBE SER ESCALAR

function calificacion=puntosb(x)
if x>=0 & x<=100
    if x>=90
        calificacion='A';
    elseif x>=80
        calificacion='B';
    elseif x>=70
        calificacion='C';
    elseif x>=60
        calificacion='D';
    else
        calificacion='E';
    end
else
    calificacion='Puntos no vàlidos';
end
```

```
octave-3.2.4.exe:67>
octave-3.2.4.exe:67> puntosb(25)
ans = E
octave-3.2.4.exe:68> puntosb(80)
ans = B
octave-3.2.4.exe:69> puntosb(-52)
ans = Puntos no vβlidos
octave-3.2.4.exe:70> puntosb(108)
ans = Puntos no vβlidos
```

INTRODUCIMOS UN PRIMER if QUE RESTRINGE LOS PUNTOS QUE INTRODUCE EL USUARIO

ESTRUCTURAS ALTERNATIVA: switch y case

- LA ESTRUCTURA switch/case SE USA CON FRECUENCIA CUANDO EXISTE UNA SERIE DE OPCIONES DE RUTA DE PROGRAMACIÓN PARA UNA VARIABLE DADA DEPENDIENDO DE SU VALOR.

- CUALQUIER PROGRAMA QUE PUEDA HACERSE CON switch/case PUEDE HACERSE CON if/elseif/else. SIN EMBARGO, EL CÓDIGO ES MAS FACIL DE LEER CON EL PRIMERO.

- CON switch/case LOS CRITERIOS PARA SELECCIONAR LAS ACCIONES A EJECUTAR PUEDEN SER UN ESCALAR O UNA CADENA DE CARACTERES (EL ÚLTIMO ES LO MÁS FRECUENTE)

- **LA ESTRUCTURA switch/case ES DE LA SIGUIENTE FORMA:**

```
switch variable
case opcion1
    instrucciones1 (si la variable es igual a opcion1)
case opcion2
    instrucciones2 (si la variable es igual a opcion2)
....
case opcionN
    instruccionesN (si la variable es igual a opcionN)
otherwise
    instrucciones (si la variable no es igual a ninguna opción anterior)
end
```

- **LA PARTE otherwise NO ES NECESARIA PARA QUE FUNCIONE switch/case. SIN EMBARGO, DEBE INCLUIRSE SI HAY ALGUNA OPCIÓN DE QUE EL USUARIO INTRODUZCA UN VALOR QUE NO ESTE REFLEJADO EN NINGÚN CASO.**

Ejemplo 25a. Realizar un programa que pida al usuario el nombre de una ciudad y de por pantalla el precio del billete del avión.

Ejemplo25_T5.m

```
%PROGRAMA QUE PIDE AL USUARIO EL NOMBRE DE UNA CIUDAD
%DEVUELVE EL DINERO DEL BILLETE DEL AVIÓN
%OPCIONES POSIBLES:BOSTON, HONOLULU, BERLIN

city=input('Introduzca el nombre de la ciudad: ','s');
% AL PONER 's' EL USUARIO PUEDE INTRODUCIR EL NOMNRE SIN APOSTOFRE

switch city
    case 'Boston'
        disp('El precio del billete es 345 euros')
    case 'Berlin'
        disp('El precio del billete es 150 euros')
    case 'Honolulu'
        disp('El precio del billete es muy caro, quedese en casa')
    otherwise
        disp('Esa ciudad no esta disponible')
end
```

EL USUARIO INTRODUCE UNA CADENA DE CARACTERES

ESTRUCTURA switch/case

```
octave-3.2.4.exe:72>
octave-3.2.4.exe:72> Ejemplo25_T5
Introduzca el nombre de la ciudad: Berlin
El precio del billete es 150 euros
octave-3.2.4.exe:73>
octave-3.2.4.exe:73> Ejemplo25_T5
Introduzca el nombre de la ciudad: Honolulu
El precio del billete es muy caro, quedese en casa
octave-3.2.4.exe:74>
octave-3.2.4.exe:74> Ejemplo25_T5
Introduzca el nombre de la ciudad: Madrid
Esa ciudad no esta disponible
octave-3.2.4.exe:75>
```

ESTRUCTURAS ALTERNATIVA: switch y case

función menu

- **LA FUNCIÓN menu SE UTILIZA MUCHO CON LA ESTRUCTURA switch/case.**

- **AL USAR ESTA FUNCIÓN EL USUARIO NO TIENE QUE ESCRIBIR SINO SOLAMENTE ELEGIR UNA DE LAS OPCIONES QUE SE LE MUESTRAN POR PANTALLA.**

Nombre_variable=menu('Mensaje al usuario', 'texto opcion 1', 'texto opcion 2', etc.)

Ejemplo 25b. Realizar un programa que pida al usuario el nombre de una ciudad y de por pantalla el precio del billete del avión (EMPLEAR FUNCIÓN menu)

Ejemplo25_T5B.m

```
%PROGRAMA QUE PIDE AL USUARIO EL NOMBRE DE UNA CIUDAD
%DEVUELVE EL DINERO DEL BILLETE DEL AVIÓN
%OPCIONES POSIBLES:BOSTON, HONOLULU, BERLIN

city=menu('Elige una ciudad del menu', 'Boston','Berlin','Honolulu');

switch city
    case 1;
        disp('El precio del billete es 345 euros')
    case 2;
        disp('El precio del billete es 160 euros')
    case 3;
        disp('El precio del billete es muy caro, quedese en casa')
end
```

UTILIZAMOS LA FUNCION menu

EL USUARIO TIENE QUE SELECCIONAR UN NÚMERO

```
octave:10> Ejemplo25_T5B
Elige una ciudad del menu

  [ 1] Boston
  [ 2] Berlin
  [ 3] Honolulu

pick a number, any number: 1
El precio del billete es 345 euros
octave:11> _
```

Ejemplo 26a.

Ciertos países no usan el sistema métrico de medida, por ejemplo, en EEUU se emplean los galones como unidad de volumen (1 galón=3.7854 litros).

Escribe un programa para que se pueda usar para comprar gasolina.

1) **Pide al usuario si quiere comprar en galones o en litros (no hay más opciones).**
2) **Pregunte al usuario cuanta cantidad quiere comprar.**
3) **Calcula el coste total.**

Ejemplo 26b.

compra_gasolina.m

```
1  %PROGRAMA QUE CALCULA EL COSTE DE LA GASOLINA
2  %ADMITE PEDIDOS EN GALONES Y LITROS
3  % 1 LITRO = 1.39 EUROS y  1 galon = 3.7854 litros
4
5  clear, clc
6  precio_gasolina=1.39
7
8  % PEDIMOS AL USUARIO EN QUE UNIDAD VA A COMPRAR
9  compra=input('Indique si la gasolina la va a comprar en galones o litros: ',
10
11 % USAMOS SWITCH/CASE PARA CALCULAR UN FACTOR DE CONVERSION
12 switch compra
13     case 'galones'
14         factor=3.784;
15     case 'litros'
16         factor=1;
17     otherwise
18         disp('Unidad no disponible'
19         factor=0;
20 end
21
22 % PEDIMOS LA CANTIDAD QUE QUIERE
23 cantidad=input('Indique la cantidad que quiere comprar: ');
24
25 %CALCULAMOS EL COSTE DE LA GASOLINA
26 if factor~0
27     coste=cantidad*factor*precio_gasolina;
28     fprintf('El coste de la gasolina es de: %0.2f\n euros', coste)
29 end
```

VARIABLE PARA CALCULAR EL COSTE

EL USUARIO INDICA galones o litros

EN FUNCIÓN DE LO QUE DICE EL USUARIO CALCULA UN FACTOR DIFERENTE

INDICA LA CANTIDAD

SI EL FACTOR ES DISTINTO DE CERO, CALCULA EL COSTE

Ejemplo 26c.

```
if factor~0
    coste=cantidad*factor*precio_gasolina;
    fprintf('El coste de la gasolina es de: %0.2f\n euros', coste)
end
```

AL NO PONER ; SE MUESTRA POR PANTALLA ans=1 (VERDADERO)

```
Indique si la gasolina la va a comprar en galones o litros: galones
Indique la cantidad que quiere comprar: 10
ans =  1
El coste de la gasolina es de: 52.60
 eurosoctave-3.2.4.exe:81>
```

```
26  if factor~0;
27      coste=cantidad*factor*precio_gasolina;
28      fprintf('El coste de la gasolina es de: %0.2f\n euros', coste)
29  end
```

```
Indique si la gasolina la va a comprar en galones o litros: galones
Indique la cantidad que quiere comprar: 10
El coste de la gasolina es de: 52.60
 eurosoctave-3.2.4.exe:82> _
```