

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

ESTRUCTURAS REPETITIVAS O BUCLES

- UN BUCLE ES UNA ESTRUCTURA DE PROGRAMACIÓN QUE PERMITE LA REPETICIÓN CONTROLADA DE UN CONJUNTO DE INSTRUCCIONES.
- OCTAVE DISPONE DE LAS SIGUIENTES ESTRUCTURAS REPETITIVAS:
 - ✓ for (para)
 - ✓ while (mientras)
 - ✓ do-until (repetir...hasta)
- ESTE TIPO DE ESTRUCTURAS, Y EN PARTICULAR LAS INSTRUCCIONES while y for SON UTILIZADAS DE FORMA GENERALIZADA EN LA INMENZA MAYORÍA DE LOS LENGUAJES DE PROGRAMACIÓN.

ESTRUCTURAS REPETITIVA: bucles for

- EL BUCLE for SE UTILIZA CUANDO NOS INTERESA REPETIR UN BLOQUE DE INSTRUCCIONES UN NÚMERO PREDETERMINADO DE VECES.

- LA ESTRUCTURA for ES DE LA SIGUIENTE FORMA:

```
for i = vector
  instrucciones
end
```

- EL CONJUNTO DE INSTRUCCIONES SE REPITE PARA CADA ELEMENTO DEL VECTOR, DENOMINÁNDOSE ITERACIÓN A CADA UNA DE ESTAS REPETICIONES.

- EN CADA ITERACIÓN, i TOMA DE FORMA ORDENADA EL VALOR DE CADA ELEMENTO DEL VECTOR (SI ES UNA MATRIZ, CADA CICLO i ES UNA COLUMNA DE DICHA MATRIZ) .

Ejemplo:

```
for i = [1:5]
    j=2.*i
    disp(j)
end
```

Por pantalla se mostrará: ans = 2 4 6 8 10

- EL OPERADOR “:” ES MUY USADO EN LOS BUCLES for. EN ESTE CASO, LOS BUCLES TENDRÁN LA SIGUIENTE ESTRUCTURA (LOS CORCHETES NO SON NECESARIOS):

```
for i = [inicial:incremento:final]
    instrucciones
end
```

Ejemplo 28: Usar un bucle for para calcular los cuadrados de los cinco primeros números impares.

```
for i = [1:2:9]
    j=2.*i
    disp(j)
end
```

Ejemplo 29: Escribir un programa que sume los elementos de un vector.

```

suma.m
1  %PROGRAMA QUE SUMA LOS ELEMENTOS DE UN VECTOR
2
3  v=1:2:9
4
5  for x=v
6      suma=suma+x
7  end
    
```

DEFINIMOS EL VECTOR

```

octave-3.2.4.exe:23>
octave-3.2.4.exe:23> suma
v =
     1     3     5     7     9
error: invalid call to script
error: called from:
    
```

AL EJECUTAR EN OCTAVE DA UN ERROR

- EL ERROR SE DEBE A NO INICIALIZAR LA VARIABLE SUMA (ACUMULADOR).

```

suma.m
1  %PROGRAMA QUE SUMA LOS ELEMENTOS DE UN VECTOR
2
3  v=1:2:9
4
5  suma=0;
6
7  for x=v
8      suma=suma+x
9  end
    
```

INICIALIZAMOS SUMA

```

octave-3.2.4.exe:23> suma
v =
     1     3     5     7     9
suma = 1
suma = 4
suma = 9
suma = 16
suma = 25
octave-3.2.4.exe:24>
    
```

EL PROGRAMA NO DA ERROR

Ejemplo 30: Escribir un programa que sume los elementos de un vector y calcule la media (EMPLEAR UN CONTADOR).

```

1  %PROGRAMA QUE SUMA LOS ELEMENTOS DE UN VECTOR
2  % CALCULA LA MEDIA
3
4  v=1:2:9
5
6  suma=0;
7
8  contador=0;
9
10 for x=v;
11     suma=suma+x;
12     contador=contador+1;
13 end
14
15 media=suma/contador;
16
17 fprintf('La media es: %0.2f\n', media)
    
```

INICIALIZAMOS EL CONTADOR Y EL ACUMULADOR

EN EL BUCLE AUMENTAMOS EN UNA CANTIDAD CONSTANTE EL CONTADOR

CON EL ACUMULADOR Y EL CONTADOR HACEMOS LA MEDIA

```

octave-3.2.4.exe:28> Ejemplo30_T5
v =
     1     3     5     7     9
La media es: 5.00
octave-3.2.4.exe:29>
    
```

ESTRUCTURAS REPETITIVA: bucles for

length



SIRVE PARA OBTENER EL NÚMERO DE ELEMENTOS DE UN VECTOR.
ES MUY UTILIZADO EN LOS BUCLES for EN LOS QUE NO SABEMOS LA LONGITUD DEL VECTOR.

```
ejemplo_for_generar_datos.m
1: b=(1:2:100);
2:
3: a=length(b)
4:
5: for i=1:a
6:     j(i)=2.*i;
7: end
8:
9: b=j.*5;
10:
```

ESTRUCTURAS REPETITIVA: bucles for

En un bucle for sino guardamos los datos de cada una de las repeticiones en una variable, el resultado del bucle será el ultimo dato que ha calculado en dicho bucle.

```

ejemplo_for_generar_datos.m
ejemplo_for_sin_datos.m
1  b=(1:2:100);
2
3  a=length(b)
4
5  for i=1:1:a
6      j=2.*i;
7  end
8
9  b=j.*5
  
```

No guardamos los resultados de cada bucle

```

ejemplo_for_generar_datos.m
1  b=(1:2:100);
2
3  a=length(b)
4
5  for i=1:1:a
6      j(i)=2.*i;
7  end
8
9  b=j.*5
10
  
```

Con nombre_variable(i) guardamos los resultados de cada bucle

```

octave:4> ejemplo_for_sin_datos
a =
    50
b =
   500
octave:5>
  
```

El ultimo dato del bucle es 100 , con ello hace el siguiente calculo que esta fuera del bucle

```

octave:3> ejemplo_for
a =
    50
b =
   500

Columns 1 through 13:
    10    20    30    40    50    60    70    80    90   100   110   120   130
Columns 14 through 26:
   140   150   160   170   180   190   200   210   220   230   240   250   260
Columns 27 through 39:
   270   280   290   300   310   320   330   340   350   360   370   380   390
Columns 40 through 50:
   400   410   420   430   440   450   460   470   480   490   500
octave:4>
  
```

El cálculo fuera del bucle se ejecuta con todos los datos generados en el bucle

ESTRUCTURAS REPETITIVA: bucles while

- EL BUCLE `while` ES UNA ESTRUCTURA QUE SE UTILIZA PARA REPETIR UN CONJUNTO DE INSTRUCCIONES MIENTRAS SE CUMPLA UNA CONDICIÓN LÓGICA DETERMINADA.

- LA ESTRUCTURA `while` ES DE LA SIGUIENTE FORMA:

```
while condicion
  instrucciones
end
```

- MIENTRAS LA CONDICIÓN ES VERDADERA, SE EJECUTAN LAS INSTRUCCIONES, TRAS LO CUAL SE VUELVE A COMPROBAR LA CONDICIÓN. EN EL MOMENTO QUE ES FALSA, SE TERMINA EL BUCLE.
- LA VARIABLE QUE SE EVALÚA DEBE CAMBIAR CADA VEZ QUE SE REPITE EL BUCLE, DE LOS CONTRARIO SERÍA UN CICLO INFINITO.

ESTRUCTURAS REPETITIVA: bucles while

Ejemplo:

```

i=0
while i < 3
    disp(i)
    i=i+1
end
disp('Fin del programa')
    
```

- PARA LAS CONDICIONES while SE PUEDEN UTILIZAR TODOS LOS OPERADORES Y FUNCIONES LÓGICAS ESTUDIADOS.
- EL BUCLE for SE USA CUANDO SE QUIEREN REPETIR LAS INSTRUCCIONES UN NÚMERO PREDETERMINADO DE VECES Y while BUSCA EL CUMPLIMIENTO DE UNA CONDICIÓN PARA LA FINALIZACIÓN DEL CICLO.

ESTRUCTURAS REPETITIVA: bucles do-until

- EL BUCLE do-until (REPETIR HASTA) ES MUY SIMILAR AL BUCLE while, LA FINALIZACIÓN DE BUCLE ESTA LIGADA AL CUMPLIMIENTO DE UNA CONDICIÓN.
- EL BUCLE do-until LA CONDICIÓN SE COMPRUEBA AL FINAL DE LA ESTRUCTURA Y SE EJECUTAN LAS ORDENES HASTA QUE SE CUMPLE LA CONDICIÓN (Y NO MIENTRAS, COMO OCURRE EN while).
- LA ESTRUCTURA do-until ES DE LA SIGUIENTE FORMA:

do

instrucciones

until condicion

- LAS INSTRUCCIONES DEL BUCLE SE EJECUTAN AL MENOS UNA VEZ.

ESTRUCTURAS REPETITIVA: bucles do-until

Ejemplo: mostrar números por pantalla hasta que llegue al número 3

i=0

do

disp(i)

i=i+1

until i>2

disp('Fin del programa')

SELECCIÓN DEL TIPO DE BUCLE

for



REPETICIÓN DE UN CONJUNTO DE INSTRUCCIONES UN NÚMERO PREDETERMINADO DE VECES.

while



INCUMPLIMIENTO DE UNA CONDICIÓN AL INICIO DEL BUCLE PARA LA FINALIZACIÓN DEL CICLO. LAS INSTRUCCIONES DEL BUCLE SE EJECUTAN 0 O MÁS VECES.

do-while



CUMPLIMIENTO DE UNA CONDICIÓN AL FINAL DEL BUCLE PARA LA FINALIZACIÓN DEL CICLO. LAS INSTRUCCIONES DEL BUCLE SE EJECUTAN 1 O MÁS VECES.

ESTRUCTURAS REPETITIVAS O BUCLES

break



SE UTILIZA PARA TERMINAR UN BUCLE PREMATURAMENTE UNA VEZ NO SE CUMPLA LA CONDICIÓN, PARA QUE NO CONTINÚE HACIENDO EL BUCLE.

Ejemplo:

n=0

while (n<10)

a=input('Ingrese un valor mayor que cero')

n=n+1

if a<=0

disp('Debe ingresar un numero positivo')

break

end

disp('El logaritmo de este número es')

disp(log(a))

end

CUANDO SE INTRODUCE UN NÚMERO NEGATIVO NO SE REPITE MAS EL BUCLE

ESTRUCTURAS REPETITIVAS O BUCLES

continue



SE UTILIZA PARA CONTINUAR UN BUCLE AUNQUE NO SE HAYA CUMPLIDO LA CONDICIÓN.

Ejemplo:

n=0

while (n<10)

a=input('Ingrese un valor mayor que cero')

n=n+1

if a<=0

disp('Debe ingresar un numero positivo')

continue

end

disp('El logaritmo de este número es')

disp(log(a))

end

**CUANDO SE INTRODUCE UN NÚMERO NEGATIVO
EL BUCLE SE REPITE DE NUEVO**

Ejemplo 31a: Se tienen N disoluciones numeradas de 1 a N y se mide el pH y la temperatura de cada disolución.

a) Escribe el pseudocódigo del algoritmo de un programa que pida al usuario el número de disoluciones N y los valores de temperatura y pH de esas N disoluciones. Mostrará por pantalla:

- La temperatura media de las disoluciones claramente ácidas ($\text{pH} < 6.5$).
- La temperatura media de las disoluciones claramente básicas ($\text{pH} > 7.5$).
- La temperatura media de las disoluciones neutras ($6.5 \leq \text{pH} \leq 7.5$).

b) Puede que al escribir el programa anterior no hayas tenido en cuenta que pudiera no haber disoluciones en uno de los grupos de pH (por ejemplo, podría no haber disoluciones ácidas). En ese caso, al ejecutar el programa se generaría un error al calcular la media (por dividir por cero). Si no lo has hecho, incluye las modificaciones necesarias para que, en el caso de que no haya disoluciones de un grupo, el programa lo indique al usuario.

Ejemplo 31b.

```

1  % PROGRAMA QUE PIDE AL USUARIO EL NUMERO DE DISOLUCIONES
2  % PIDE PH Y T DE CADA UNA DE ELLAS Y HACE LA IMEDIA
3
4  clear, clc
5
6  acidas=0;
7  basicas=0;
8  neutras=0;
9  TMneutras=0;
10  TMacidas=0;
11  TMbasicas=0;
12
13  N=input('Introduce el numero de disoluciones que quiera medir: ');
14
15  for i=1:N;
16      ph=input('Introduzca el pH de la disolución: ');
17      T=input('introduzca la temperatura de la disolucion en Kelvin: ');
18      if ph<6.3;
19          acidas=acidas+1;
20          TMacidas=TMacidas+T;
21      elseif ph>7.5;
22          basicas=basicas+1;
23          TMbasicas=TMbasicas+T;
24      else
25          neutras=neutras+1;
26          TMneutras=TMneutras+T;
27      end
28  end
  
```

INICIALIZAMOS LOS CONTADORES Y ACUMULADORES

BUCLE for DESDE 1 HASTA N

Ejemplo 31c.

```

29 if acidas==0;
30     disp('No hay disoluciones acidas');
31 else
32     fprintf('La temperatura media de las disoluciones acidas es: %0.3f K\n', TMacidas/acidas)
33 end
34
35 if basicas==0;
36     disp('No hay disoluciones basicas');
37 else
38     fprintf('La temperatura media de las disoluciones basicas es: %0.3fK\n', Tmbasicas/basicas)
39 end
40
41 if neutras==0;
42     disp('No hay disoluciones neutras');
43 else
44     fprintf('La temperatura media de las disoluciones neutras es: %0.3fK \n', TMneutras/neutras)
45 end
    
```

BUCLE if PARA SALIDAS POR PANTALLA PARA PROPORCIONAR TEMPERATURAS MEDIAS

```

Introduce el numero de disoluciones que quiere medir: 3
Introduzca el pH de la disoluci%  
n: 5
introduzca la temperatura de la disolucion en Kelvin: 300
Introduzca el pH de la disoluci%  
n: 6
introduzca la temperatura de la disolucion en Kelvin: 393
Introduzca el pH de la disoluci%  
n: 8
introduzca la temperatura de la disolucion en Kelvin: 500
La temperatura media de las disoluciones acidas es: 346.500 K
La temperatura media de las disoluciones basicas es: 500.000K
No hay disoluciones neutras
octave-3.2.4.exe:16>
    
```

Ejemplo propuesto: Modifica el programa anterior para que evalúe cual es la disolución de mayor temperatura y que muestre el pH de dicha disolución.

```

disoluciones.m | disolucionesTmayor.m
% PROGRAMA QUE PIDE AL USUARIO EL NUMERO DE DISOLUCIONES
% PIDE PH Y T DE CADA UNA DE ELLAS Y HACE LA T MEDIA
% INDICA CUAL ES EL PH DE LA DISOLUCION DE MAYOR TEMPERATURA

clear, clc

acidas=0;
basicas=0;
neutras=0;
TMneutras=0;
TMacidas=0;
TMbasicas=0;
Tmayor=0;

N=input('Introduce el numero de disoluciones que quiere medir: ');

for i=1:N;
    ph=input('Introduzca el pH de la disolucion: ');
    T=input('Introduzca la temperatura de la disolucion en Kelvin: ');
    if ph<6.5;
        acidas=acidas+1;
        TMacidas=TMacidas+T;
        if T>Tmayor;
            Tmayor=T;
            pH_Tmayor=ph;
        end
    elseif ph>7.5;
        basicas=basicas+1;
        TMbasicas=TMbasicas+T;
        if T>Tmayor;
            Tmayor=T;
            pH_Tmayor=ph;
        end
    else
        neutras=neutras+1;
        TMneutras=TMneutras+T;
        if T>Tmayor;
            Tmayor=T;
            pH_Tmayor=ph;
        end
    end
end

if acidas==0;
    disp('No hay disoluciones acidas');
else
    fprintf('La temperatura media de las disoluciones acidas es: %0.3f K\n', (TMacidas)/acidas)
end

if basicas==0;
    disp('No hay disoluciones basicas');
else
    fprintf('La temperatura media de las disoluciones basicas es: %0.3fK\n', (TMbasicas)/basicas)
end

if neutras==0;
    disp('No hay disoluciones neutras');
else
    fprintf('La temperatura media de las disoluciones neutras es: %0.3fK \n', (TMneutras)/neutras)
end

fprintf('La disolucion que presenta la mayor temperatura tiene un pH de %0.1f \n',pH_Tmayor)

```

Ejemplo propuesto: Modifica el programa anterior para que evalúe cual es la disolución de mayor temperatura y que muestre el pH de dicha disolución.

```

Introduce el numero de disoluciones que quiere medir: 6
Introduzca el pH de la disolucion: 6
introduzca la temperatura de la disolucion en Kelvin: 300
Introduzca el pH de la disolucion: 5.5
introduzca la temperatura de la disolucion en Kelvin: 273
Introduzca el pH de la disolucion: 4
introduzca la temperatura de la disolucion en Kelvin: 400
Introduzca el pH de la disolucion: 8
introduzca la temperatura de la disolucion en Kelvin: 300
Introduzca el pH de la disolucion: 9
introduzca la temperatura de la disolucion en Kelvin: 283
Introduzca el pH de la disolucion: 8.2
introduzca la temperatura de la disolucion en Kelvin: 393
La temperatura media de las disoluciones acidas es: 324.333 K
La temperatura media de las disoluciones basicas es: 325.333K
No hay disoluciones neutras
La disolucion que presenta la mayor temperatura tiene un pH de 4.0
octave-3.2.4.exe:11> _
    
```

**ES IMPORTANTE PROBAR QUE EL PROGRAMA FUNCIONA
CORRECTAMENTE CON UN EJEMPLO REAL**

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



BLOQUE TEMÁTICO II

FUNDAMENTOS DE PROGRAMACIÓN Y APLICACIONES EN INGENIERÍA

TEMA 4. Fundamentos de programación

TEMA 5. Programación básica en MATLAB[®] / Octave

TEMA 6. Programación de macros en Excel empleando Visual Basic