

Módulo 3: Lógica de programación y depuración

- Ciclos.
- Depurador paso a paso.

Ciclo For

Usado cuando se conoce el número de veces que un código debe usarse.

```
for ( i = 1 : 5 )  
    disp( i^2 ) % Esto se ejecuta 5 veces  
end
```

```
1  
4  
9  
16  
25
```

```
S = -10 : 0.5 : 0;  
  
for ( j = 1 : length(S) )  
    disp( S(j) ) % Recorrer un vector y mostrar lo que hay almacenado  
                  % en cada posición  
end
```

```
-10  
-9.5000  
-9  
-8.5000  
-8  
-7.5000  
-7  
-6.5000  
-6  
-5.5000
```

-5
-4.5000
-4
-3.5000
-3
-2.5000
-2
-1.5000
-1
-0.5000
0

```
S = -10 : 1 : 10;

for ( j = 1 : length(S) )
    if ( mod( S(j), 2) ~= 0 )
        palabra = ['El número ', num2str( S(j) ), ' es impar.' ]
        disp( palabra )
        % Ver si el valor almacenado en cada posición de S es impar
    end
end
```

```
palabra =
'El número -9 es impar.'
El número -9 es impar.
palabra =
'El número -7 es impar.'
El número -7 es impar.
palabra =
'El número -5 es impar.'
El número -5 es impar.
palabra =
'El número -3 es impar.'
El número -3 es impar.
palabra =
'El número -1 es impar.'
El número -1 es impar.
palabra =
'El número 1 es impar.'
El número 1 es impar.
palabra =
'El número 3 es impar.'
El número 3 es impar.
palabra =
```

```

'El número 5 es impar.'
El número 5 es impar.
palabra =
'El número 7 es impar.'
El número 7 es impar.
palabra =
'El número 9 es impar.'
El número 9 es impar.

```

Ex. 1. Recorrer una matriz y construir una tabla de multiplicar en ella.

```

N = 10;
M_table = zeros( N );
for (i = 1 : N)
    % Cualquier código aquí se ejecuta N veces
    for (j = 1 : N)
        % Cualquier código aquí se ejecuta N² veces
        M_table( i, j ) = i*j;
    end
end
M_table

```

```

M_table = 10x10
     1     2     3     4     5     6     7     8     9    10
     2     4     6     8    10    12    14    16    18    20
     3     6     9    12    15    18    21    24    27    30
     4     8    12    16    20    24    28    32    36    40
     5    10    15    20    25    30    35    40    45    50
     6    12    18    24    30    36    42    48    54    60
     7    14    21    28    35    42    49    56    63    70
     8    16    24    32    40    48    56    64    72    80
     9    18    27    36    45    54    63    72    81    90
    10    20    30    40    50    60    70    80    90   100

```

Ciclo While

Bloque de código que debe repetirse siempre que una condición lógica se cumpla.

```

R = [1, 2, 4, -1, 4, 2];
k = 1

```

```

k = 1

```

```

while ( R(k) > 0 )
    disp( R(k) ) % Imprimir valores de R hasta encontrar el primer valor
    k = k + 1;   % negativo
end

```

2

4

Ex 2. Encontrar la suma de todos los múltiplos de 3 o 5 menores a 1000.

```
i = 1;
suma_total = 0;
while (i < 1000)
    if ( ( mod(i, 3) == 0 ) | ( mod(i, 5) == 0 ) )
        suma_total = suma_total + i;
    end
    i = i + 1;
end
suma_total
```

```
suma_total = 233168
```