

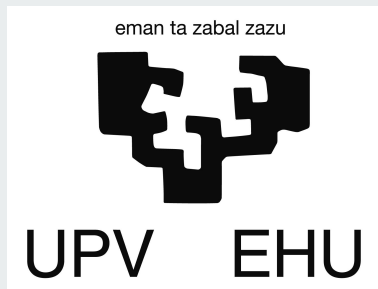


# Fundamentos de Informática

## Estructuras Iterativas

Dept. Lenguajes y Sistemas Informáticos (LSI)  
2018/2019

Rodrigo Agerri, Xabier Larrucea, Mari Carmen Otero, Juan Francisco Ramirez



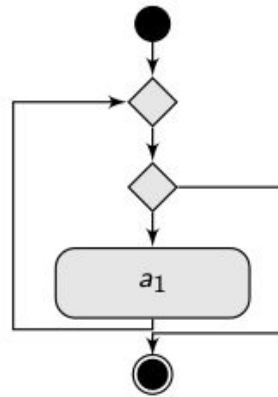
# Índice



1. Introducción
2. Estructuras "for"
3. Estructuras "while"

# Motivación

- Las estructuras iterativas son necesarias para repetir múltiples veces una misma operación



# Estructuras for

---

```
for index = values
    enunciados
end
```

```
for num = 1:5
    fprintf('%d\t', num);
end
```

```
for v = [1 5 8 17]
    disp(v)
end
```

1

5

8

17

# Estructuras for

Avance en incrementos de  $-0.2$  y muestre los valores.

```
for v = 1.0:-0.2:0.0
    disp(v)
end
```

1

0.8000

0.6000

0.4000

0.2000

0

# Estructuras for



```
for I = eye(4,3)
    disp('Current unit vector:')
    disp(I)
end
```

Current unit vector:

1  
0  
0  
0

Current unit vector:

0  
1  
0  
0

Current unit vector:

0  
0  
1  
0

# Estructuras for



```
for I = eye(4,3)
    disp('Current unit vector:')
    disp(I)
end
```

Current unit vector:

1  
0  
0  
0

Current unit vector:

0  
1  
0  
0

Current unit vector:

0  
0  
1  
0

# Estructuras for



```
M = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
for elem=M  
    M(elem) = M(elem) + 1  
end
```

```
M = [2 3 4 5; 6 7 8 9; 10 11 12 13]
```



# Estructuras for



```
M = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
for elem=M'  
    M(elem) = M(elem) + 1  
end
```

```
M = [2 3 4 5; 6 7 8 9; 10 11 12 13]
```

# Estructuras for anidadas and if

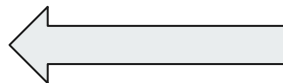
Crea una matriz de 1s.

```
nrows = 4;  
ncols = 6;  
A = ones(nrows,ncols);
```

A = 4x6

|    |    |    |    |    |   |
|----|----|----|----|----|---|
| 2  | -1 | 0  | 0  | 0  | 0 |
| -1 | 2  | -1 | 0  | 0  | 0 |
| 0  | -1 | 2  | -1 | 0  | 0 |
| 0  | 0  | -1 | 2  | -1 | 0 |

Realice un bucle a través de la matriz y asigne a cada elemento un nuevo valor. Asigne 2 en la diagonal principal, -1 en las diagonales adyacentes y 0 en cualquier otro lugar.



```
for c = 1:ncols  
    for r = 1:nrows  
  
        if r == c  
            A(r,c) = 2;  
        elseif abs(r-c) == 1  
            A(r,c) = -1;  
        else  
            A(r,c) = 0;  
        end  
    end  
end  
A
```

# while



Para repetir mientras la condición sea verdadera

Utilice un bucle `while` para calcular `factorial(10)`.

```
n = 10;
f = n;
while n > 1
    n = n-1;
    f = f*n;
end
disp(['n! = ' num2str(f)])
```

n! = 3628800

Evalúa una [expresión](#) y repite la ejecución de un grupo de instrucciones en un bucle mientras que la expresión es verdadera.

- Una expresión es verdadera cuando su resultado no está vacío y contiene sólo elementos no nulos (numéricos reales o lógicos). De lo contrario, la expresión es falsa.

## Sintaxis

```
while expression
    statements
end
```