

Elementos y Estructuras de Control - II

1ºDAM IoT

Estructura iterativa / repetitiva / bucle

Bloque de sentencias que puede repetir su ejecución siguiendo uno de estos tres criterios:

1. Un número predeterminado de veces.
2. Mientras que se cumpla una condición (de permanencia).
3. Hasta que se cumpla una condición (de salida).

Casi todos los lenguajes de programación soportan estos tres tipos a la hora de realizar iteraciones/repeticiones/bucles.

Dependiendo de la sintaxis del lenguaje concreto, estos tres casos se pueden implementar de forma diferente (incluso haber alguna forma más de implementación).

Bucle PARA

Bucle PARA (FOR)

El bucle PARA ejecuta las instrucciones que contiene dentro, una serie de veces. Indicamos el valor inicial de un contador, cómo actualizar el contador y la condición que debe cumplir ese contador para seguir ejecutándose.

Su estructura suele ser:

Una variable inicial que se inicializa al comienzo del bucle con un valor.

La variable inicial es incrementada o decrementada a cada paso según el valor que le demos.

```
Para variable_numerica <- valor_inicial Hasta valor_final Con Paso paso Hacer  
    secuencia_de_acciones  
FinPara
```

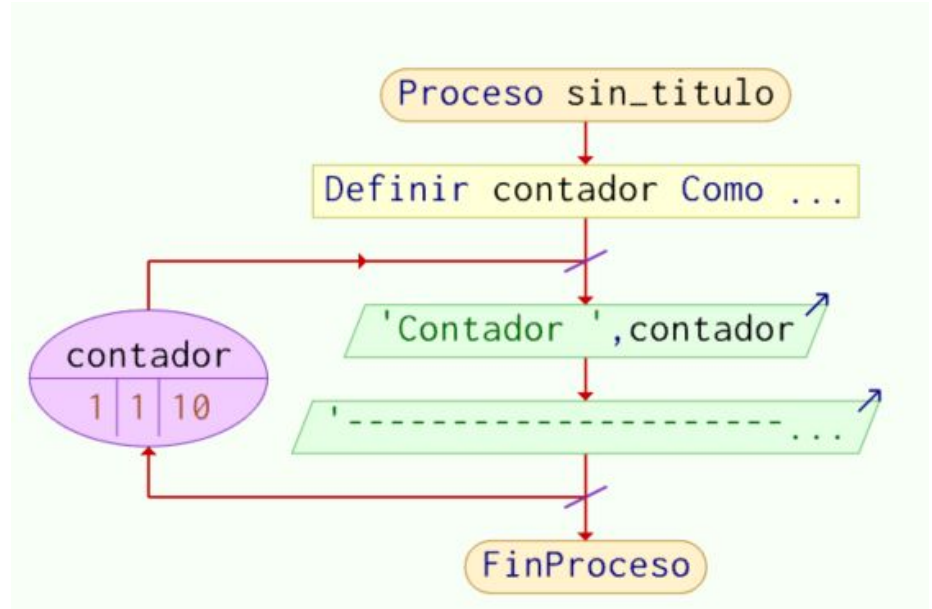
Cuando la variable numérica toma el valor final salimos del bucle.

Importante que en algún momento la variable numérica llegue al valor final, sino nunca saldremos del bucle y se producirá un **bucle infinito**.

Bucle PARA (FOR)

Ejemplo: Escribe por pantalla utilizando un bucle PARA los números del 1 al 10.

```
Proceso sin_titulo
  Definir contador Como Entero;
  Para contador ← 1 Hasta 10 Con Paso 1 Hacer
    Escribir "Contador ", contador;
    Escribir "-----";
  FinPara
FinProceso
```



Bucle PARA (FOR)

Ejercicio: ¿Cómo imprimo sólo los números pares del 1 al 100 sin utilizar una condicional SI?

Bucle PARA (FOR)

Ejercicio: ¿Cómo imprimo sólo los números pares del 1 al 100 sin utilizar una condicional SI?

Proceso `sin_titulo`

Definir contador Como Entero;

Definir rangoInferior, rangoSuperior Como Entero;

rangoInferior ← 2;

rangoSuperior ← 100;

Para contador ← rangoInferior Hasta rangoSuperior Con Paso 2 Hacer

 Escribir contador;

FinPara

FinProceso

Bucle PARA (FOR)

Ejercicio: Imprime los números del 100 al 1, esta vez en orden decreciente.

Bucle PARA (FOR)

Ejercicio: Imprime los números del 100 al 1, esta vez en orden decreciente.

```
Proceso sin_titulo
    Definir contador Como Entero;
    Definir rangoSuperior, rangoInferior Como Entero;
    //Puedo usar dos variables para definir el rango inferior y superior
    rangoInferior ← 100;
    rangoSuperior ← 1;
    Para contador←rangoInferior Hasta rangoSuperior Con Paso -1 Hacer
        Escribir "Contador ", contador;
        Escribir "-----";
    FinPara
FinProceso
```

Bucle PARA (FOR)

Este bucle lo solemos utilizar cuando sabemos el número de veces que debe ejecutarse el bucle.

Es una sentencia de control muy cómoda porque ella se encarga de realizar el incremento/decremento a cada paso.

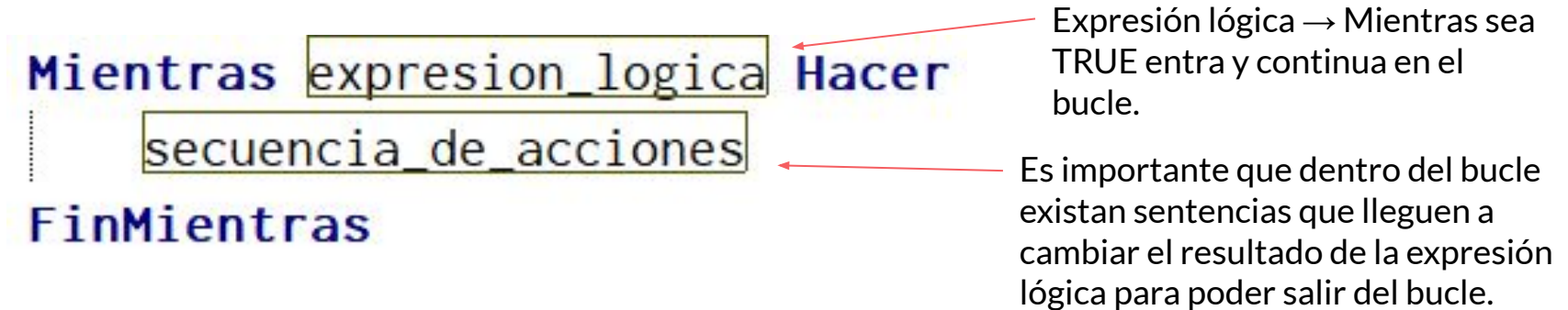
Bucle MIENTRAS

Bucle MIENTRAS (WHILE)

Este bucle se ejecuta MIENTRAS se cumpla una condición.

La condición es de PERMANENCIA (es decir, nos mantenemos en el bucle mientras su evaluación da como resultado “true”).

La condición se suele evaluar al principio del bucle, de forma que, de no cumplirse, el bloque de sentencias asociado no se ejecuta ni una sola vez.



Bucle MIENTRAS (WHILE)

Ejemplo: ¿Cómo imprimo los números del 1 al 10?

Proceso sin_titulo

Definir contador Como Entero;

contador ← 1;

Mientras contador ≤ 10 Hacer

 Escribir "Contador ", contador;

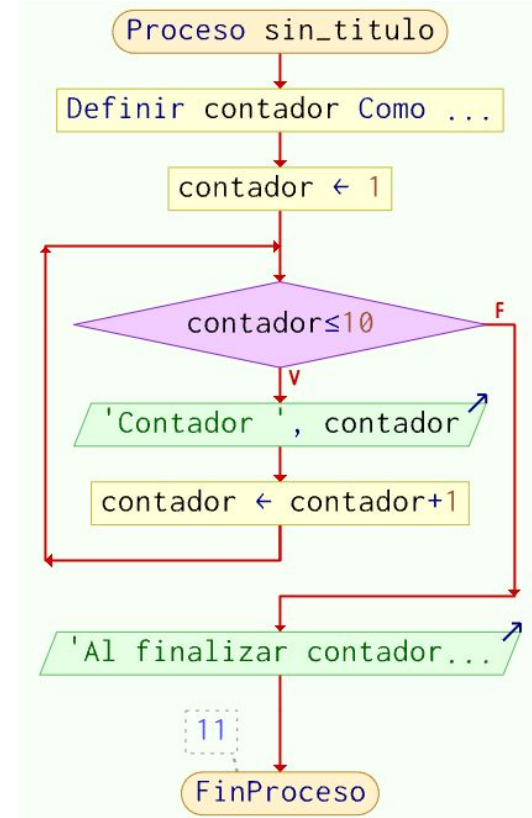
 contador ← contador + 1;

FinMientras

Escribir "Al finalizar contador tiene el valor ", contador; //

FinProceso

Contador 1
Contador 2
Contador 3
Contador 4
Contador 5
Contador 6
Contador 7
Contador 8
Contador 9
Contador 10
Al finalizar contador tiene el valor 11

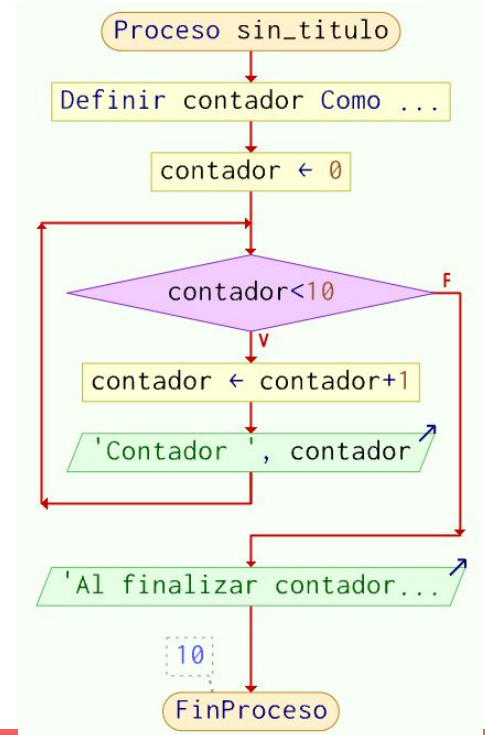


Bucle MIENTRAS (WHILE)

Ejemplo: Otra posible solución inicializando contador a 0. Esta vez al salir del bucle contador vale 10.

```
Proceso sin_titulo
    Definir contador Como Entero;
    contador ← 0;
    Mientras contador < 10 Hacer
        contador ← contador + 1;
        Escribir "Contador ", contador;
    FinMientras
    Escribir "Al finalizar contador tiene el valor ", contador; //10
FinProceso
```

```
Contador 1
Contador 2
Contador 3
Contador 4
Contador 5
Contador 6
Contador 7
Contador 8
Contador 9
Contador 10
Al finalizar contador tiene el valor 10
```



Bucle MIENTRAS (WHILE)

Ejercicio: ¿Cómo imprimo sólo los números pares del 1 al 100 utilizando MIENTRAS?

Bucle MIENTRAS (WHILE)

Ejercicio: ¿Cómo imprimo sólo los números pares del 1 al 100 utilizando MIENTRAS?

Proceso sin_titulo

Definir contador Como Entero;

contador \leftarrow 2;

Mientras contador \leq 100 Hacer

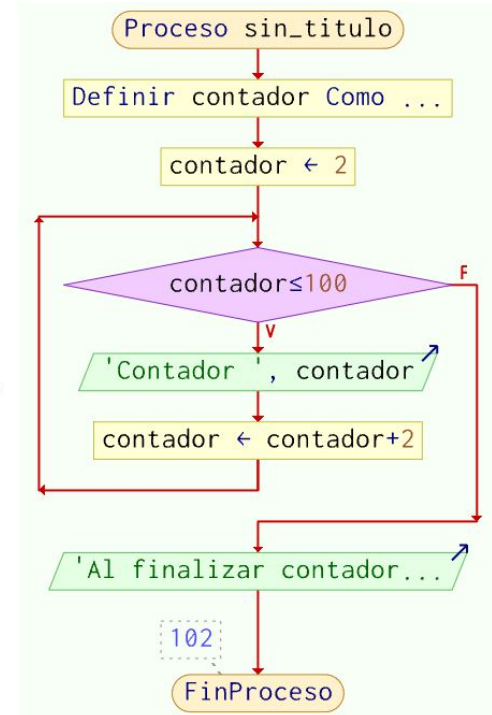
 Escribir "Contador ", contador;

 contador \leftarrow contador + 2;

FinMientras

Escribir "Al finalizar contador tiene el valor ", contador; //102

FinProceso



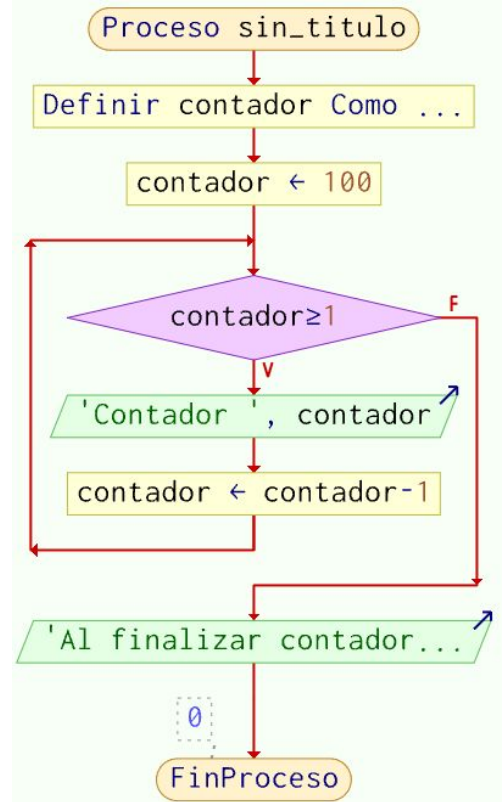
Bucle MIENTRAS (WHILE)

Ejercicio: Imprime los números del 100 al 1, esta vez en orden decreciente.

Bucle MIENTRAS (WHILE)

Ejercicio: Imprime los números del 100 al 1, esta vez en orden decreciente.

```
Proceso sin_titulo
  Definir contador Como Entero;
  contador ← 100;
  Mientras contador ≥ 1 Hacer
    Escribir "Contador ", contador;
    contador ← contador - 1;
  FinMientras
  Escribir "Al finalizar contador tiene el valor ", contador; //0
FinProceso
```



Bucle REPETIR

Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Permanecemos en el bucle HASTA QUE se cumpla una condición.

La condición es de SALIDA (es decir, nos mantenemos en el bucle mientras su evaluación da como resultado “false”).

La condición se evalúa al final del bucle, esto hace que este tipo de bucles se ejecutan al menos una vez siempre.

```
//Secuencias de instrucciones
Repetir
|   secuencia_de_acciones
Hasta Que expresion_logica
//Secuencias de instrucciones
```

De nuevo, es importante que dentro del bucle existan sentencias que lleguen a cambiar el resultado de la expresión lógica para poder salir del bucle.

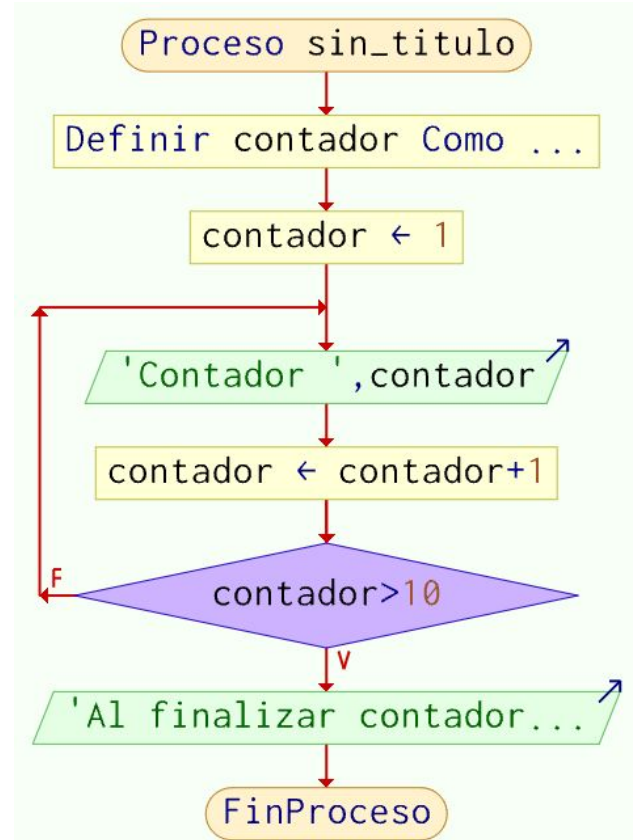
Se repite hasta que la expresión lógica es TRUE.
Se repite MIENTRAS que la expresión lógica sea FALSE.

Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Ejemplo: ¿Cómo imprimo los números del 1 al 10?

```
Definir contador como Entero;  
contador ← 1;  
Repetir  
|   Escribir "Contador ", contador;  
|   contador ← contador + 1;  
Hasta Que contador > 10  
Escribir "Al finalizar contador tiene el valor ", contador;
```

```
Contador 1  
Contador 2  
Contador 3  
Contador 4  
Contador 5  
Contador 6  
Contador 7  
Contador 8  
Contador 9  
Contador 10  
Al finalizar contador tiene el valor 11
```

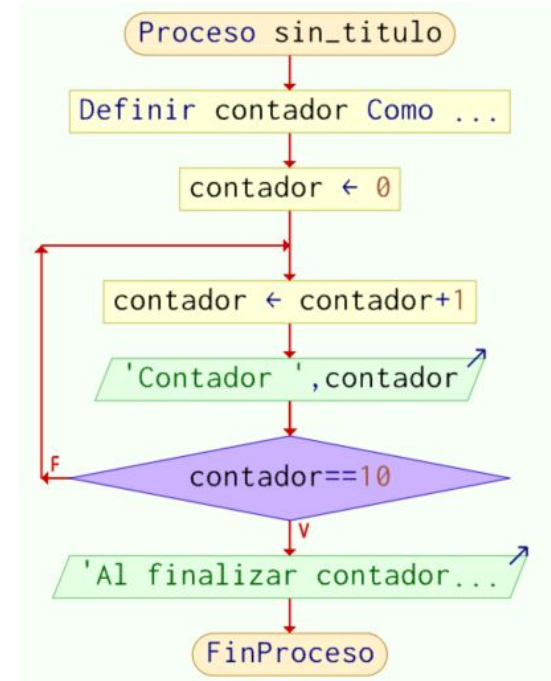


Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Ejemplo: Otra posible solución inicializando contador a 0.

```
Proceso sin_titulo
  Definir contador como Entero;
  contador ← 0;
  Repetir
    contador ← contador + 1;
    Escribir "Contador ",contador;
  Hasta Que contador ==10
  Escribir "Al finalizar contador tiene el valor ", contador;
FinProceso
```

```
Contador 1
Contador 2
Contador 3
Contador 4
Contador 5
Contador 6
Contador 7
Contador 8
Contador 9
Contador 10
Al finalizar contador tiene el valor 10
```



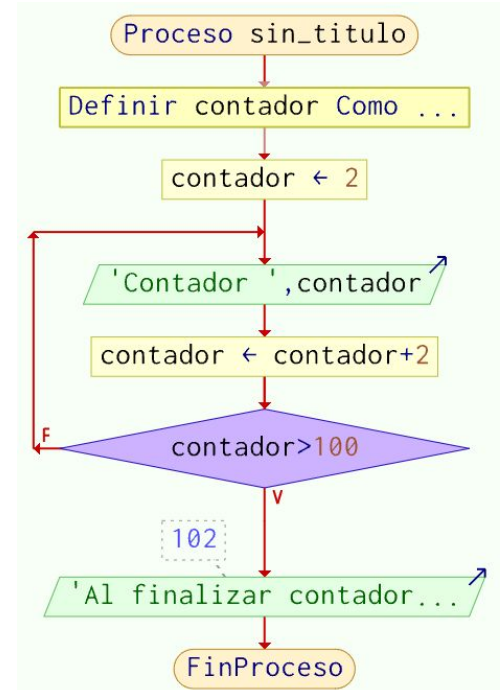
Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Ejercicio: ¿Cómo imprimo sólo los números pares del 1 al 100 utilizando REPETIR?

Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Ejercicio: ¿Cómo imprimo sólo los números pares del 1 al 100 utilizando REPETIR?

```
Proceso sin_titulo
  Definir contador como Entero;
  contador ← 2;
  Repetir
    Escribir "Contador ", contador;
    contador ← contador + 2;
  Hasta Que contador > 100
  Escribir "Al finalizar contador tiene el valor ", contador; //102
FinProceso
```



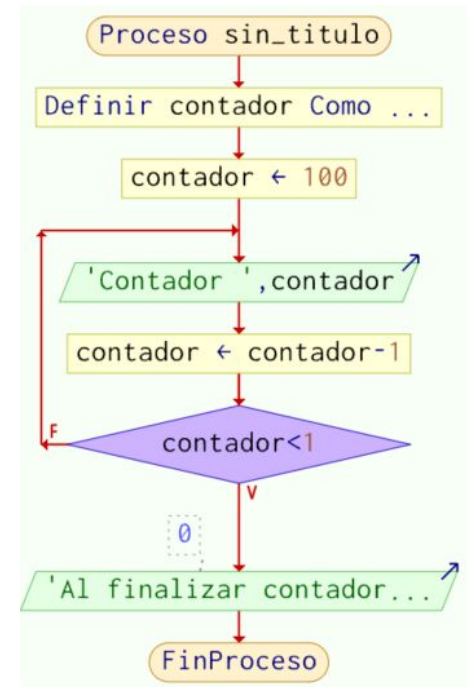
Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Ejercicio: Imprime los números del 100 al 1, esta vez en orden decreciente.

Bucle REPETIR ... HASTA QUE (REPEAT ... UNTIL)

Ejercicio: Imprime los números del 100 al 1, esta vez en orden decreciente.

```
Proceso sin_titulo
  Definir contador como Entero;
  contador ← 100;
  Repetir
    Escribir "Contador ",contador;
    contador ← contador -1;
  Hasta Que contador < 1
  Escribir "Al finalizar contador tiene el valor ", contador; //0
FinProceso
```



¿Cuándo usar cada bucle?

Cualquier algoritmo en el que se necesite repetir una serie de sentencias puede utilizar cualquiera de los 3 bucles vistos.

PARA → Normalmente se utilizan con un tipo de datos llamado arrays que veremos más adelante. O cuando queremos repetir las instrucciones un número de veces y el contador que utilizamos no lo modificamos nosotros dentro del bucle, sino que dejamos que sea el propio PARA quien lo haga.

MIENTRAS → Cuando queremos ejecutar unas instrucciones mientras se cumple una condición.

REPETIR → Queremos que se repita al menos una vez la secuencia de instrucciones. Por ejemplo, mostramos siempre un menú con opciones al usuario hasta que decide salir.

Tarea - 4