UNIDAD 3
ESTRUCTURAS REPETITIVAS

Programación
CFGS DAW

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

ÍNDICE

UD3. ESTRUCTURAS REPETITIVAS

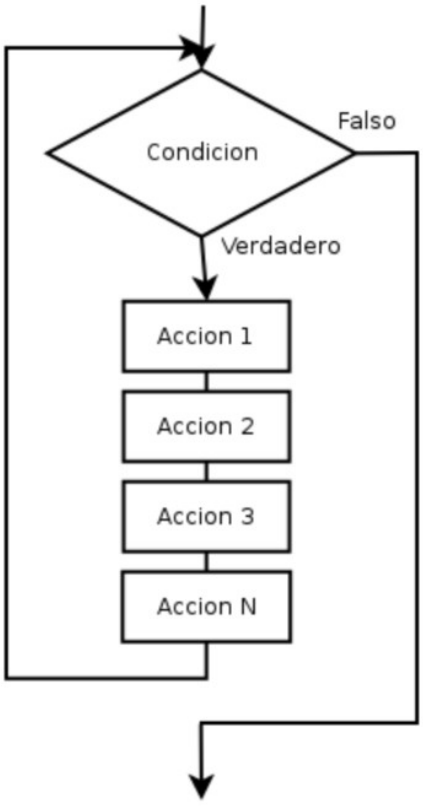
1. INTRODUCCIÓN

Las **instrucciones repetitivas (o bucles)** son aquellas que **permiten variar o alterar la secuencia normal de ejecución de un programa** haciendo posible que un grupo de operaciones (acciones) se repita un número determinado o indeterminado de veces, dependiendo del cumplimiento de una condición.

Veremos tres tipos: Bucle Mientras (WHILE), Bucle Hacer-Hasta (DO-WHILE) y Bucle Para (FOR)

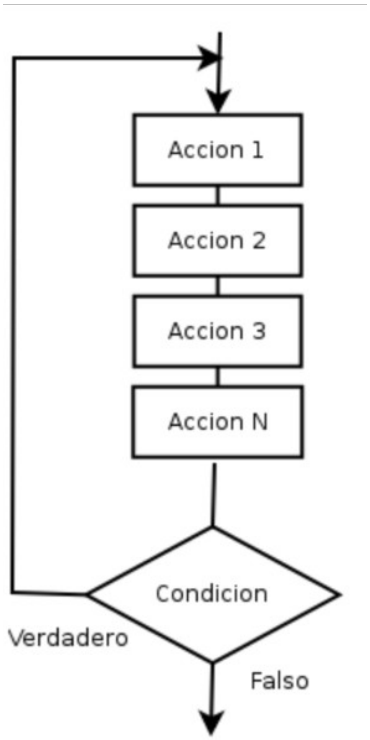
2. ESTRUCTURA MIENTRAS (WHILE)

En la estructura Mientras o "**WHILE**" el bloque de acciones **se repite mientras la condición sea cierta**, evaluándose siempre la condición antes de entrar en el bucle, por ello es posible que las acciones no se ejecuten nunca.

Pseudocódigo	Ordinograma
Mientras Condición hacer Instrucción 1 Instrucción 2 ... Instrucción N FinMientras	

3. ESTRUCTURA HASTA (DO-WHILE)

En la estructura hasta o "**DO-WHILE**", el bloque de instrucciones **se repite mientras que la condición sea cierta**, y la condición se evalúa al final del bloque por lo que siempre se ejecutarán al menos una vez el bloque de instrucciones.

Pseudocódigo	Ordinograma
Repetir Instrucción 1 Instrucción 2 ... Instrucción N Mientras Condición.	

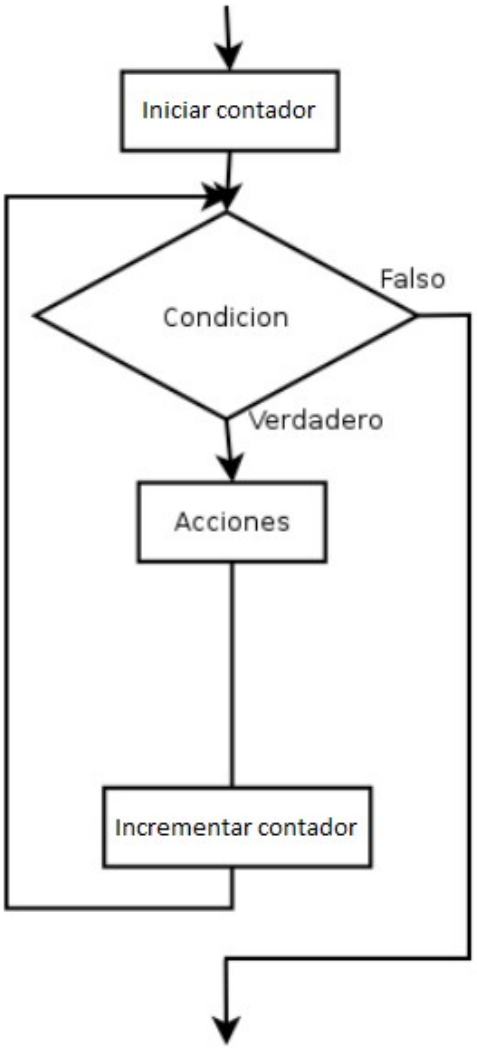
4. ESTRUCTURA PARA (FOR)

En la estructura Para o "FOR" se conoce de antemano el número de veces que se ejecutará el bloque de instrucciones.

> El bloque de acciones **se repite mientras que la condición sea cierta**, **evaluándose** siempre la condición **antes de entrar en el bucle**, por ello es posible que las acciones no se ejecuten nunca.

Esta explicación es idéntica a la del bucle WHILE, pero un bucle FOR debe cumplir las siguientes características:

1. La variable contador se inicializa con un valor inicial.
2. La condición siempre debe ser: variable_contador <= valor_final
3. En cada interacción, la variable contador se incrementa en un determinado valor.

Pseudocódigo	Ordinograma
Para Var_Cont de ValorInicial a ValorFinal con Incremento = n Instrucción 1 Instrucción 2 ... Instrucción N FinPara	

5. FORMAS DE ACABAR UN BUCLE

Uno de los peligros que se corren cuando se escribe un bucle es que éste no acabe nunca, lo que se denomina **bucle infinito**, para no cometer este error grave debemos recordar que las condiciones de los bucles deben poder cambiar dentro del bucle, es decir que si por ejemplo utilizamos una variable comparada con una constante, dicha variable debe poder cambiar de valor dentro del bucle.

> Las estructuras repetitivas deben incluir un mecanismo para que éstas se acaben.

Algunos de los métodos más usados para esa labor son los siguientes:

1. **Cuando sabemos el número de veces que se va a repetir la estructura, utilizaremos un contador.**

   Por ejemplo, en un enunciado del tipo: "imprimir la tabla del 7", sabemos que el proceso va desde 1 a 10, por lo tanto, usaremos un contador.

```
cont = 1
Mientas cont <=10
        Escribir cont * 7
        cont = cont + 1
FinMientras
```

2. **Preguntando si queremos seguir en el bucle.**

   Por ejemplo, en un ejercicio del tipo "introducir N alumnos y hallar su media", debemos preguntar si queremos introducir más alumnos:

```
. . .
seguir="s"
Mientras ((seguir="s") o (seguir="S"))
        . . .
        Escribir "Introducir más alumnos?"
        Leer seguir
FinMientras
. . .
```

3. **Usando un valor centinela.**

   Así, en el caso del siguiente problema: “Introducir N notas hasta introducir un 10”:

   ```
   . . .
   Leer nota
   Mientras (nota <> 10)
           . . .
           Leer nota
   FinMientras
   . . .
   ```

4. **Usando un interruptor que tomará el valor lógico true o false.**

   En un ejemplo como “Repetir ciertas instrucciones mientras la condición sea cierta”:

   ```
   . . .
   Mientras (SW = Verdadero)
           . . .
   FinMientras
   . . .
   ```

6. ELEMENTOS AUXILIARES

Los elementos auxiliares son variables que realizan funciones especificas dentro de un programa, y por su gran utilidad, frecuencia de uso y peculiaridades, conviene hacer un estudio separado de las mismas.

6.1 Contadores

Si vamos a repetir una acción un número determinado de veces y esa variable se va a incrementar siempre en una cantidad constante, se denomina **contador**. Sería útil llamarla algo así como CONT, CONTA, CONTADOR... Ai tuviéramos varios contadores dentro de un programa podríamos llamarlos CONT1, CONT2...

Se utilizan en los siguientes casos:

- Para contabilizar el número de veces que es necesario repetir una acción (variable de control de un bucle).
- Para contar un suceso particular solicitado por el enunciado del problema. Un contador debe inicializarse a un valor inicial (normalmente a cero) e incrementarse cada vez que ocurra un suceso.

6.2 Acumuladores

Si por el contrario, dicho objeto se va **incrementando de forma variable** se denomina **acumulador**. Deberemos llamarla ACU, ACUM, ACUMULA, ACUMULADOR, SUMA, ... u otra palabra significativa.

Se utiliza en aquellos casos en que se desea obtener el total acumulado de un conjunto de cantidades, siendo inicializado con un valor cero.

También en ocasiones hay que obtener el total acumulado como producto de distintas cantidades, en este caso se inicializará a uno.

Por ejemplo: imprimir la suma de N edades.

6.3 Interruptores

Por último, tenemos ciertas variables que pueden **tomar dos valores: cierto o falso**. Se les denomina **interruptores o switches** y su función es que ciertas instrucciones se ejecuten mientras tenga un valor determinado. A las variables de este tipo las denominaremos SW.

Se utiliza para:

- Recordar que un determinado suceso a ocurrido o no en un punto determinado del programa, y poder así realizar las decisiones oportunas.
- Hacer que dos acciones diferentes se ejecuten alternativamente dentro de un bucle.

Por ejemplo: introducir N edades y acabar al introducir un 99.