

APOYA



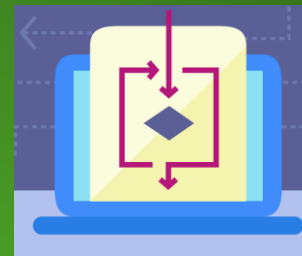


Introducción a la Programación

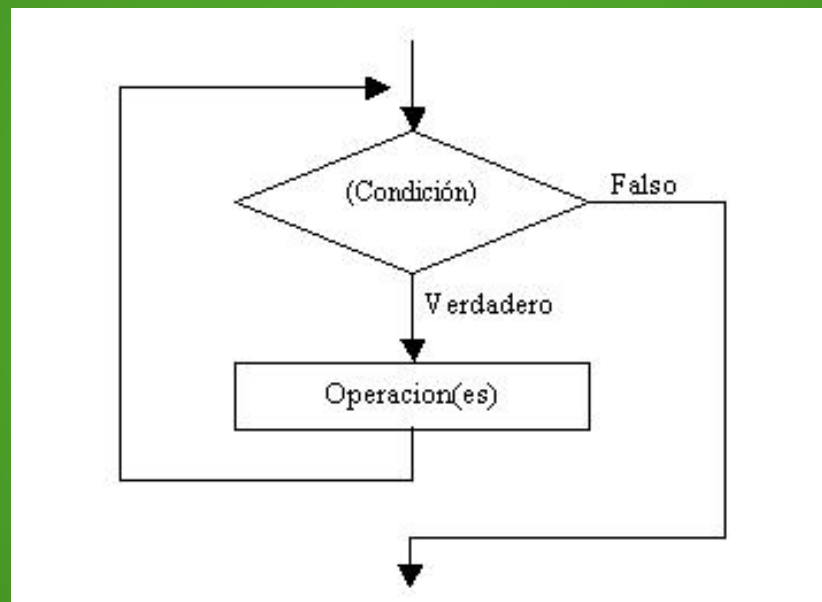
Comisión “B”

Año 2020

Profesor: Ing. Gabriel Guismin



Estructuras Repetitivas



Estructura repetitiva - Diagrama de flujo

Acumuladores

Es una variable numérica (entera o real) que se inicializa con un valor indicado (generalmente en cero) y se incrementa (o decrementa) con el contenido de otra variable (o varias variables). Se lo suele utilizar en bucles para obtener totales de valores que se almacenaron sucesivamente en una o varias otras variables.

```
total ← 0          { Se inicializa el acumulador total en 0. }  
...  
total ← total + a  { El acumulador se incrementa con el valor almacenado en a, }  
...               { cualquiera sea el contenido de a }
```

*Acumulador: la variable **total** va almacenando los distintos valores de la variable **a***

Contadores

Es una variable numérica entera que se inicializa con un valor dado (generalmente en cero) y se incrementa (o decrementa) siempre con un valor constante (por lo general uno). Se lo suele utilizar en bucles para contar sucesos, datos o acciones.

```
 $a \leftarrow 0$            { Se inicializa el contador  $a$  en 0 }  
...  
 $a \leftarrow a + 1$      { El contador  $a$  se incrementa en 1 (ahora almacena el valor 1) }  
...  
 $a \leftarrow a + 1$      { El contador  $a$  se incrementa en 1 (ahora almacena el valor 2) }  
...  
 $a \leftarrow a + 1$      { El contador  $a$  se incrementa en 1 (ahora almacena el valor 3) }
```

Contador: la variable a va tomando los valores de manera sucesiva

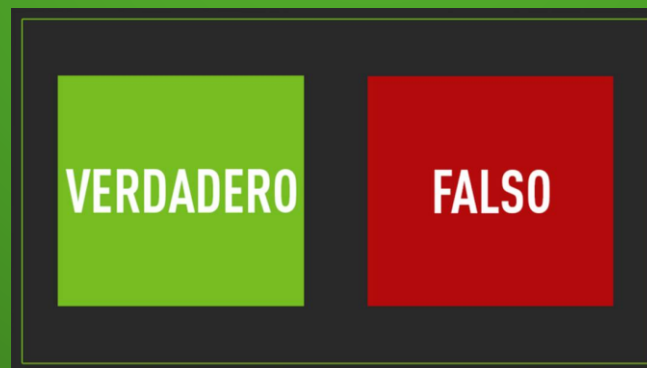
Interruptor o Bandera

Es una variable que se utiliza para indicar ciertos sucesos durante la ejecución de un programa.

Permite comunicar sucesos de una parte del programa a otra.

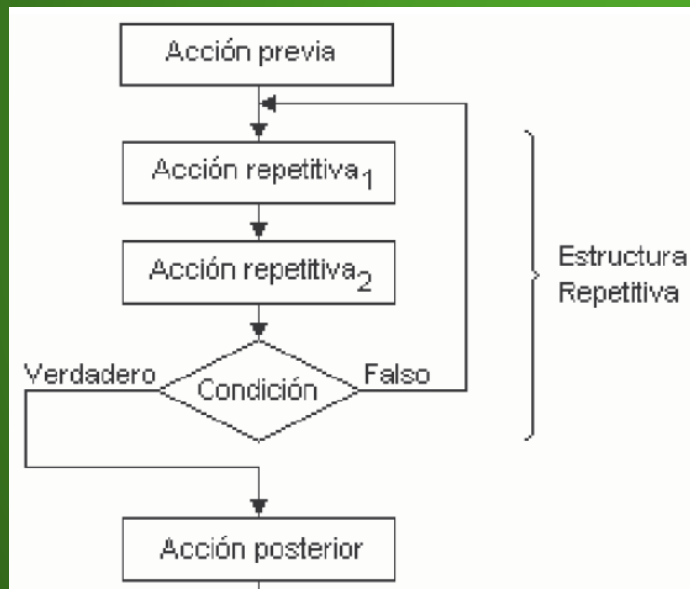
Generalmente se utiliza una variable de tipo lógica. Estas permiten indicar si un suceso ocurrió (verdadero) o no (falso).

Se inicializa con un valor (por ejemplo falso), y si el suceso esperado ocurre, entonces se cambia al otro valor (verdadero), caso contrario se mantiene el primero (falso). Más adelante en la ejecución del programa se podrá saber si el suceso ocurrió (valor verdadero en la bandera) o no (bandera en falso) según el valor almacenado en esta variable.



Estructura Repetitiva: Repetir

Con esta estructura, las acciones encerradas en el bucle se ejecutan al menos una vez y luego se repiten hasta que se cumpla la condición dada que se evalúa al final del bucle.



Estructura Repetir representada en diagrama de flujo

```
Repetir
{
    secuencia_de_acciones
}
Hasta Que expresion_logica
```

Estructura Repetir representada en pseudocódigo

Estructura Repetitiva: Repetir - Ejemplo

```
Algoritmo sin_titulo

    Definir cont Como Entero

    cont = 1

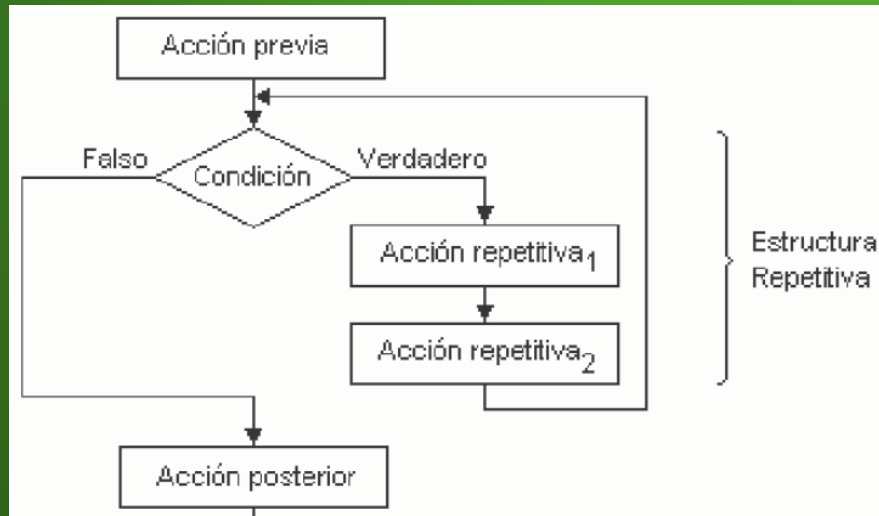
    Repetir
    |
    |     Escribir "Hola soy el número: " cont
    |
    |     cont= cont+1
    |
    Hasta Que (cont=11)

FinAlgoritmo
```

Estructura Repetir Ejemplo: El algoritmo cuenta de 1 a 10

Estructura Repetitiva: Mientras

Esta estructura se construye evaluando primero la condición de salida y luego se escribe su cuerpo repetitivo. La implementación de esta estructura de control comienza con la palabra reservada ***mientras*** seguida de la condición, continúa con las acciones que se ejecutarán varias veces (cuerpo del bucle, acciones repetitivas) y finaliza con la palabra reservada ***finmientras***.



Estructura Mientras representada en diagrama de flujo

```

Mientras expresion_logica Hacer
    secuencia_de_acciones
Fin Mientras
  
```

Estructura Mientras representada en pseudocódigo

Estructura Repetitiva: Mientras - Ejemplo

```
Algoritmo sin_titulo

    Definir cont Como Entero

    cont = 1

    Mientras (cont ≤ 10) Hacer
        Escribir "Hola soy el número: " cont
        cont=cont+1
    Fin Mientras

FinAlgoritmo
```

Estructura Mientras - Ejemplo: El algoritmo cuenta de 1 a 10

Bucle controlado por contador

Este tipo de bucles puede ser implementado con cualquiera de las estructuras de control.
Asumiendo que se cuenta con una variable entera llamada **cont** (como ejemplo) que se utilizará como contador.

Utilizando Repetir

```
cont ← <valor inicial>           { Inicialización del contador }
repetir                          { Inicio del bucle }
    leer (dato)                  { Obtención de los datos }
    <acción1>                    { Proceso de los datos }
    <acción2>
    ...
    <acciónn>
    cont ← cont + 1              { Incremento del contador }
hasta que (cont = <valor final>) { Control y cierre del bucle }
```

Utilizando Mientras

```
...
cont ← <valor inicial>           { Inicialización del contador }
mientras (cont < <valor final>) { Control e inicio del bucle }
    leer (dato)                  { Obtención de los datos }
    <acción1>                    { Proceso de los datos }
    <acción2>
    ...
    <acciónn>
    cont ← cont + 1              { Incremento del contador }
finmientras                     { Cierre del bucle }
```

Bucle controlado por centinela

Es un tipo de bucle sobre el cual el programador no posee conocimiento acerca de la cantidad de veces que se repetirán las acciones que constituyen el bucle.

La cantidad de veces que se debe repetir está dada por la cantidad de datos a procesar, siendo esta cantidad variable de una ejecución a otra. Debido a esto, es necesario definir un valor especial para indicar la salida del bucle.

```
leer (dato)                                { Obtención de datos }
mientras (dato <> <centinela>)             { Control e inicio del bucle }
    <acción1>                             { Proceso de los datos }
    <acción2>
    ...
    <acciónn>
    leer (dato)                             { Obtención de nuevos datos }
finmientras                               { Cierre del bucle }
```

Bucle controlado por centinela

Para conocer: en JAVA

```
do{
    instrucciones
}while(condicion);
```

"Repetir" en Java



```
public static void main(String[] args) {
    int num;
    do{
        String texto=JOptionPane.showInputDialog("Introduce un numero entre 0 y 10");
        num=Integer.parseInt(texto);
    }while(num>=10 || num<0);
    System.out.println("El numero introducido es "+num);
}
```

Ejemplo: "Repetir" en Java

```
while (condicion){
    instrucciones
}
```

Mientras en Java



```
public class PruebaApp {

    public static void main(String[] args) {

        int num=1;

        while (num<=5){
            System.out.println(num);
            num++;
        }

    }

}
```

Ejemplo: Mientras en Java