



## CONTROL DE FLUJO – CICLOS

### Autor(es):

Gustavo Eduardo Gil Prado  
Docente TC

### Resumen:

El objetivo de esta presentación es servir de guía en realizar un repaso de los conceptos de estructuras de control de flujo y ciclos previo al inicio del curso de Programación Orientada a Objetos

### FACULTAD DE INGENIERÍA

Ingeniería Informática  
Institución Universitaria Colegio Mayor del Cauca



Guía de Curso – 30/05/2022

### Descripción:

Servir de apoyo al desarrollo del curso de Programación Orientada a Objetos

### Palabras clave:

Algoritmo, Clase, Instancia, Aplicación, Atributo, Método

Referencie este documento así: Gil, G.E. (2023). CONTROL DE FLUJO – CICLOS [Guía de Curso]. Institución Universitaria Colegio Mayor del Cauca.

---

# **CONTROL DE FLUJO - CICLOS**

**GUSTAVO EDUARDO GIL PRADO**



# En la sesión anterior aprendimos...

---

Ejecutar condicionalmente instrucciones

Expresiones lógicas (comparaciones) y operadores lógicos.

Instrucción (if)

Instrucción (if - Else) y anidamiento



# Instrucción if / else: Ejemplo – Recordemos.

---

```
if (condition)
{
    /*
    Bloque de instrucciones que se ejecutarán si la condición se
    cumple ósea si es igual a true (1)
    */
}
else
{
    /*
    Bloque de instrucciones que se ejecutarán si la condición NO
    se cumple ósea si es igual a false (0)
    */
}
}
```

# Instrucción if / else / if : Ejemplo

---

```
if (condition)
{
    /*
    Bloque de instrucciones que se ejecutarán si la condición se cumple ósea
    si es igual a true (1)
    */
}
else if(condition2)
{
    /*
    Bloque de instrucciones que se ejecutarán si la condición NO se cumple
    ósea si es igual a false (0)
    */
}
```

# ¿Qué vamos a aprender hoy?

---

Repetición de bloques de instrucciones: Ciclos

Instrucciones cíclicas

Tipos de ciclo



# Por que utilizar ciclos...

---

La necesidad de repetir instrucciones, aprovechado la capacidad de las maquinas de realizar operaciones simples de manera muy rápida.

Muchos de las operaciones requieren de la repetición lógica de pasos.

Ejemplo:

- El funcionamiento de los sistemas numéricos.
- Las operaciones matemáticas como: multiplicación, factorial, etc.

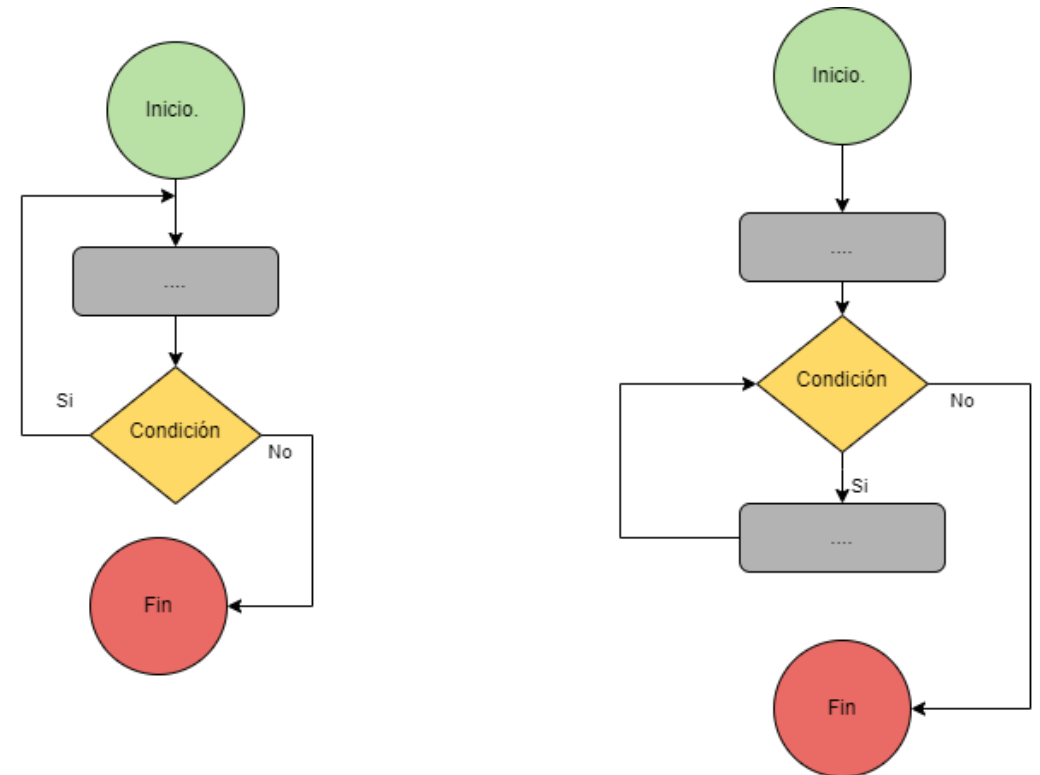


# Tipos de instrucciones cíclicas.

En JavaScript así como en gran parte de los lenguajes que comparten su sintaxis (C++, JAVA...), en la se distinguen 3 instrucciones cíclicas:

- Ciclo **for**.
- Ciclo **While**
- Ciclo **do-while**

¿Cuál debemos utilizar?





# Sintaxis – ciclo `for`

---

```
for(int i = valor_inicial; i <= valor_final; i = i + paso)
{
/*....
Bloque de Instrucciones....
....*/
}
```

Por sus características este ciclo es apropiado cuando conocemos la cantidad de iteraciones y/o requerimos el uso de un contador o número con un valor de paso por cada ciclo.

# Sintaxis – ciclo `while`

---

```
while(condición_de_finalización) //por ejemplo numero == 100
{
/*....
Bloque de Instrucciones....
....*/
}
```

Por sus características este ciclo es muy versátil ya que podemos ajustar las iteraciones acordes a una condición lógica dentro del código.

# Sintaxis – ciclo `do-while`

---

```
do
{
/*....
Bloque de Instrucciones....
....*/
}
while(condición_de_finalización); //por ejemplo numero != 23
```

Por sus características este ciclo es apropiado cuando requerimos que el bloque de instrucciones se ejecute por lo menos una vez antes de comprobar el condicional.

# Ejemplo simple de ciclos....

---

Realizar una aplicación que permita mostrar los primeros 20 números pares.

# Ejercitemos

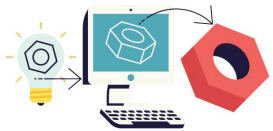
---



Leer como entrada un número entero  $n$ , e imprimir la tabla de multiplicar del  $n$ , desde 0 hasta 10.



Leer como entrada un número entero  $n$ , e imprimir una cuenta regresiva desde  $n$  hasta 1, seguido de “boom”.



Leer como entrada un número entero  $n$ , e imprimir en orden todos los números pares menores o iguales a  $n$ .

# Ejercicios

---

Leer como entrada dos números enteros **a** y **b**, y calcule e imprima todos los múltiplos de **a** que son menores que **b**.

Leer como entrada un número real **x** y un número entero positivo **n**, y calcular la potencia  $x^n$ .

Leer como entrada dos números enteros **a** y **b**, y calcule el resto que queda después de dividir de forma entera **a** entre **b** (sin utilizar el operador `%` del lenguaje).

Leer como entrada un número entero de hasta 9 dígitos, y contar la cantidad de dígitos que tiene y la suma de esos dígitos.