

---

# Estructuras de repetición

---

Fundamentos de la Programación.



Curso de Programación en Java  
**Juan Francisco** Maldonado León

Introducción .....	3
Estructuras de repetición .....	4
<i>Introducción</i> .....	4
<i>Tipos de Bucles</i> .....	6
Estructura de repetición Para (for) .....	6
Estructura de repetición Mientras (while) .....	10
Estructura de repetición Repetir Hasta (do-while).....	14

# Introducción

Las computadoras están especialmente diseñadas para todas aquellas aplicaciones que requieren una operación o conjunto de ellas que deben repetirse muchas veces.

Un tipo muy importante de estructura es el algoritmo necesario para repetir una o varias acciones un número determinado de veces.

# Estructuras de repetición

## Introducción

Las estructuras lógicas de repetición son construcciones que permiten repetir determinadas partes de una solución, de tal forma que en función de una condición o el valor de una variable, se pueda repetir un conjunto de pasos un determinado número de veces. Las estructuras lógicas de repetición son conocidas también como estructuras de iteración o bucles.

### **Bucles**

Son las estructuras que repiten una secuencia de instrucciones un número determinado de veces hasta que una condición se cumpla.

### **Iteración**

En programación, iteración se refiere a la repetición de una serie de instrucciones en un programa de computadora. Puede usarse tanto como un término genérico, así como para describir una forma específica de repetición con un estado mutable.

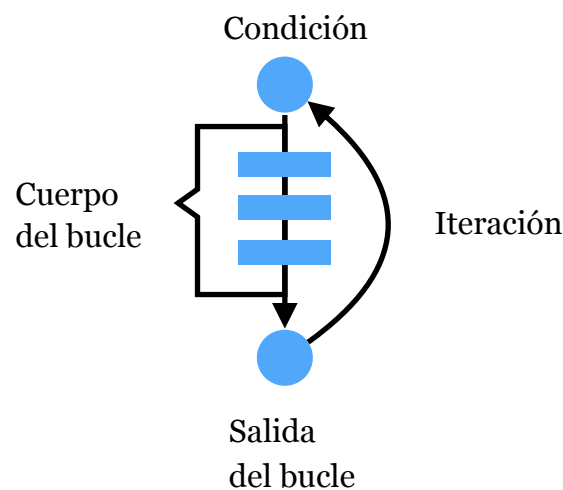
Por ejemplo, imagine que necesita leer 10 números y sumarlos.

```
suma = 0
leer numero1
suma = suma + numero1
leer numero2
suma = suma + numero2
leer numero3
suma = suma + numero3
...
```

Al analizar el ejemplo anterior, si utilizamos estructuras secuenciales, podemos determinar que las tareas son repetitivas, es decir la acción de leer un número y luego sumarlo a un resultado se realizara 10 veces. Ahora imagina que debes sumar 100 números. Para casos como esto es necesario usar estructuras de repetición que harán nuestros algoritmos mas eficientes.

Las dos principales preguntas ha realizarse en el diseño de un bucle son ¿Que contiene el bucle? ( leer un numero y sumarlo ) y ¿Cuántas veces se debe repetir? (10 veces).

Una estructura de repetición se compone de tres partes condición, cuerpo del bucle y salida del bucle.



Se debe establecer un mecanismo para determinar las tareas repetitivas. Este mecanismo es una condición que puede ser verdadera o falsa y que se comprueba una vez a cada paso o iteración del bucle (total de instrucciones que se repiten en el bucle).

# Tipos de Bucles.

Existen, principalmente, tres tipos de bucles, su utilización depende del tipo de programa que estemos realizando. Los bucles se definen básicamente por la forma en que les indicamos el número de iteraciones que debe realizar.

- Estructura de repetición Para (for)
- Estructura de repetición Mientras (while)
- Estructura de repetición Repetir hasta ( do-while ).

## Estructura de repetición Para (for)

Repite un bloque de código un número determinado de veces dependiendo del valor de una variable especial llamada contador.

Esta estructura, simplifica su uso incluyendo un contador como parte de ésta, es decir forma parte de la estructura, la inicialización de la variable, la condición de terminación y el valor del incremento.

Esta estructura declara en su primera parte la inicialización y nombre de variable, luego la condición de terminación y por último el valor del incremento, que puede ser negativo en caso que se requiera una cuenta regresiva.

## Pseudocódigo de la Estructura de Repetición Para

```
Para <inicializar> Hasta <condición> Con Paso <incremento> Hacer  
    Instrucción 1  
    Instrucción 2  
    Instrucción 3  
    ...  
FinPara
```

## Diagrama de Flujo de la Estructura de Repetición Para



La estructura de repetición Para se componen de tres elementos.

**<Inicializar>** : inicializa una variable específica y temporal para el bucle, esta instrucción solo se ejecuta la primera vez al inicio del bucle. Esta variable funciona como un índice que permite mantener el control sobre el número de veces que se repiten las instrucciones.

**<Condición>** : La segunda parte indica la condición de terminación para el ciclo, la cual esta directamente relacionada con el valor de la variable inicializada.

**<Incremento>** : Finalmente, la última parte especifica como serán manipulados los valores iniciales en cada iteración del ciclo.



## Ejemplos de utilización de estructura de repetición Para

### Ejemplo 1

Calcular la suma de los números naturales desde 1 hasta n.

Entrada	Proceso	Salida
Leer numero que define hasta que numero se debe sumar.	se deben sumar todos los números naturales desde 1 hasta el numero ingresado.	Mostrar la suma total.
<b>Leer</b> numero suma igual a 0	<b>para</b> numeroNatural = 0 <b>hasta</b> numero <b>Con</b> <b>Paso 1 Hacer</b> suma = suma + numeroNatural <b>FinSi</b>	<b>Mensaje</b> "Suma total"

### Pseudocódigo de la Estructura de Repetición Para

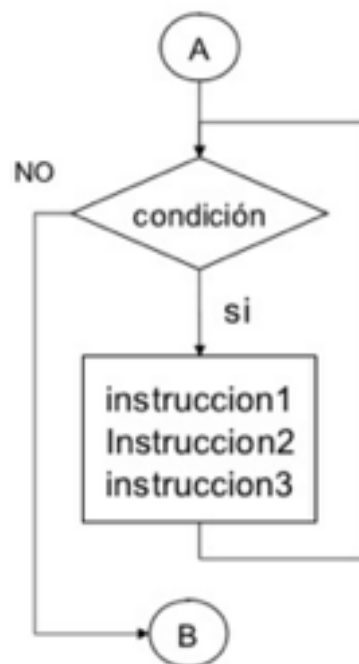
```
Proceso SumaNumeros
|   Leer numeroIngresado
|   suma = 0
|   Para numeroNatural<-1 Hasta numeroIngresado Con Paso 1 Hacer
|   |   suma = suma + numeroNatural
|   Fin Para
|   Escribir "Suma :", suma
FinProceso
```

## Estructura de repetición Mientras (while)

El ciclo mientras permite ejecutar un bloque de instrucciones mientras que una expresión lógica dada se cumpla, es decir, mientras su evaluación dé como resultado verdadero. La expresión lógica se denomina condición y siempre se evalúa antes de ejecutar el bloque de instrucciones. Si la condición no se cumple, el bloque no se ejecuta. Si la condición se cumple, el bloque se ejecuta, después de lo cual la instrucción vuelve a empezar, es decir, la condición se vuelve a evaluar.

En el caso en que la condición evalúe la primera vez como falsa, el bloque de instrucciones no será ejecutado, lo cual quiere decir que el número de repeticiones o iteraciones de este bloque será cero. Si la condición siempre evalúa a verdadero, la instrucción se ejecutará indefinidamente, es decir, un número infinito de veces.

Diagrama de flujo de la Estructura de Repetición Mientras



## Pseudocódigo de la Estructura de Repetición Mientras

```
Mientras <condición>  
    Instrucción 1  
    Instrucción 2  
    ...  
Fin Mientras
```

Donde, <condición> es la expresión lógica que se evalúa para determinar la ejecución o no del bloque de instrucciones, y <bloque instrucciones> (Instrucción 1, Instrucción 2, etc) es el conjunto de instrucciones que se ejecuta si la condición evalúa a **Verdadero**.

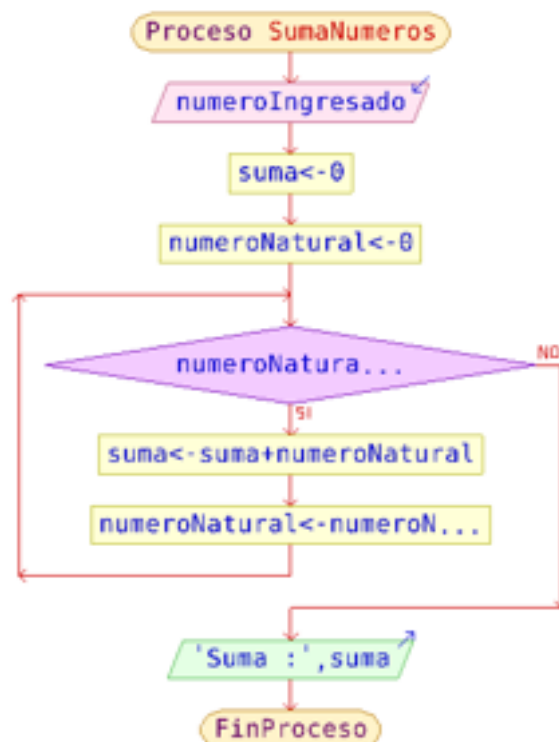
## Ejemplos de utilización de estructura de repetición Mientras

### Ejemplo 1

Calcular la suma de los números naturales desde 1 hasta n.

Entrada	Proceso	Salida
Leer numero que define hasta que numero se debe sumar.	se deben sumar todos los números naturales desde 1 hasta el numero ingresado.	Mostrar la suma total.
<b>Leer</b> numero suma igual a 0 contador igual a 0	<b>mientras</b> numeroNatural <= numeroIngresado sumar numeroNatural.	<b>Mensaje</b> "Suma total"

Diagrama de Flujo de la Estructura de Repetición Mientras



## Pseudocódigo de la Estructura de Repetición Mientras

**Proceso** SumaNumeros

**Leer** numeroIngresado

  suma = 0

  numeroNatural = 0

**Mientras** numeroNatural <=numeroIngresado **Hacer**

    suma = suma + numeroNatural

    numeroNatural = numeroNatural + 1

**Fin Mientras**

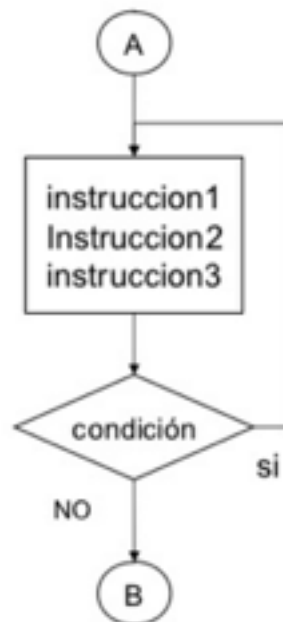
**Escribir** "Suma :", suma

**FinProceso**

## Estructura de repetición Repetir Hasta (do-while)

El ciclo repetir-hasta es similar al ciclo mientras, la diferencia radica en el momento de evaluación de la condición.

En el ciclo repetir-hasta la condición se evalúa después de ejecutar el bloque de instrucciones, por lo tanto, el bloque se ejecuta por lo menos una vez. Este bloque se ejecuta nuevamente si la condición evalúa a verdadero, y no se ejecuta más si se evalúa como falso.



**Repetir**

Instrucción 1

Instrucción 2

...

**Hasta Que** <condición>

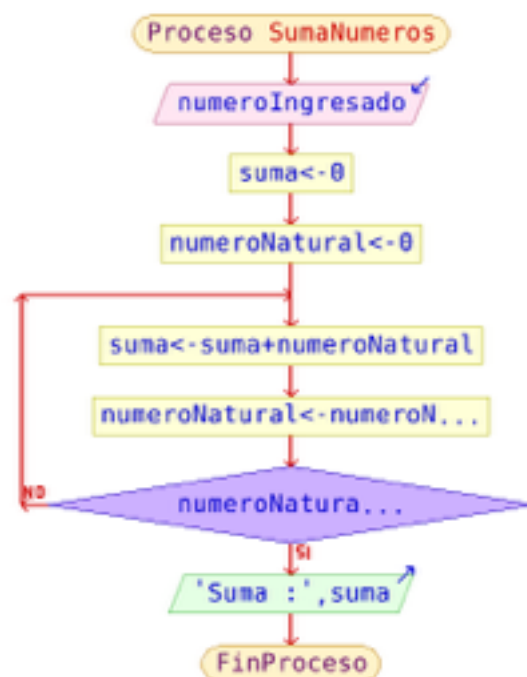
## Ejemplos de utilización de estructura de repetición Repetir-Hasta

### Ejemplo 1

Calcular la suma de los números naturales desde 1 hasta n.

Entrada	Proceso	Salida
Leer numero que define hasta que numero se debe sumar.	se deben sumar todos los números naturales desde 1 hasta el numero ingresado.	Mostrar la suma total.
<b>Leer</b> numero suma igual a 0 contador igual a 0	<b>Repetir</b> sumar numeroNatural. numeroNatural ++ Hasta numeroNatural <= numeroIngresado	<b>Mensaje</b> "Suma total"

Diagrama de Flujo de la Estructura de Repetición Mientras



## Pseudocódigo de la Estructura de Repetición Mientras

### Proceso SumaNumeros

Leer numeroIngresado

suma = 0

numeroNatural = 0

Repetir

    suma = suma + numeroNatural

    numeroNatural = numeroNatural + 1

Hasta Que numeroNatural > numeroIngresado

Escribir "Suma :", suma

FinProceso



Al momento de utilizar variables de una u otra manera, éstas pueden actuar de forma especial dentro del ciclo como lo veremos a continuación.

## Contador

un contador es una variable que dentro de un bucle aumenta o disminuye un valor fijo constante. un contador suele utilizarse para contar el numero de veces que itera un bucle. Pero, a veces se utiliza para contar solamente aquellas iteraciones de un bucle en las que se cumpla una determinada condición.

## Acumulador

Es una variable cuyo valor se incrementa o decremento de forma variable, es decir en cada iteración de un bucle aumenta secuencialmente. Un acumulador suele utilizarse para acumular resultados producidos en las iteraciones de un bucle.

## Centinela

un centinela es una variable que inicia con un valor y luego cambia o no este valor dependiendo de los estados que se presenten durante la ejecución de un programa, nos sirve para indicar si algo especifico ha ocurrido, generalmente tomo solo dos posibles valores que suelen ser verdadero o falso, pero su concepto se lo puede adaptar dependiendo de la necesidad.