

## Ciclos y sumatorias

<b>Ciclos y sumatorias</b>	<b>1</b>
¿Qué aprenderás?	2
Introducción	2
Sumatorias	3
Sumando de 1 a 100	3
Ejercicio guiado: Suma de números	5



**¡Comencemos!**

## ¿Qué aprenderás?

- Aplicar diagramas de flujos para resolver problemas con sumatorias.
- Distinguir la diferencia entre un contador y un acumulador.

## Introducción

Aprenderemos a crear programas que, además de utilizar ciclos, operen sobre otra variable. Esta estructura nos permite resolver diversos tipos de problemas, la mayoría de ellos corresponde a problemas matemáticos. Sin embargo, el entendimiento de estos contenidos nos otorgará las habilidades necesarias para aplicar esta estructura a otros contextos.

**¡Vamos con todo!**



## Sumatorias

Para algunos -que no gustan de las matemáticas- el término sumatoria puede sonar algo intimidante. Sin embargo, para resolver una sumatoria solo necesitamos saber una cosa: sumar.

La sumatoria consiste en sumar todos los números de una secuencia. Por ejemplo: sumar todos los números entre 1 y 100. Esto no solo sirve para resolver ecuaciones matemáticas, sino también para que generemos las habilidades de abstracción necesarias para resolver diversos problemas.

### Sumando de 1 a 100

$$1 + 2 + 3 + \dots + 99 + 100 = ?$$

Resolver esto es muy similar a contar las cien veces, pero además de contar, vamos a ir guardando la suma **en cada iteración**.

```
int i = 0;
while (i < 100){
    i+=1;
}
```

En el código anterior, tenemos una variable `i` que aumenta de 1 en 1 en cada iteración. Ahora necesitamos ir guardando la suma de cada uno de los números que vamos obteniendo de `i`.

Para ello crearemos una variable antes de entrar al ciclo para guardar esta suma, inicializando la variable en cero.

```
int i = 0;
int suma = 0; //Variable que guardará el resultado de la suma
while (i < 100){
    i+=1;
}
```

Si queremos sumar desde el valor 1 al 100, y como `i` parte en cero, debemos primero aumentar su valor en 1, y luego sumar.

```
int i = 0;
int suma = 0; //Variable que guardará el resultado de la suma
while (i < 100){
    i+=1;
    suma += i; //La variable suma, guarda el valor de suma + i
}
System.out.println(suma)
```

Por último, el diagrama de flujo de la solución, sería el siguiente:

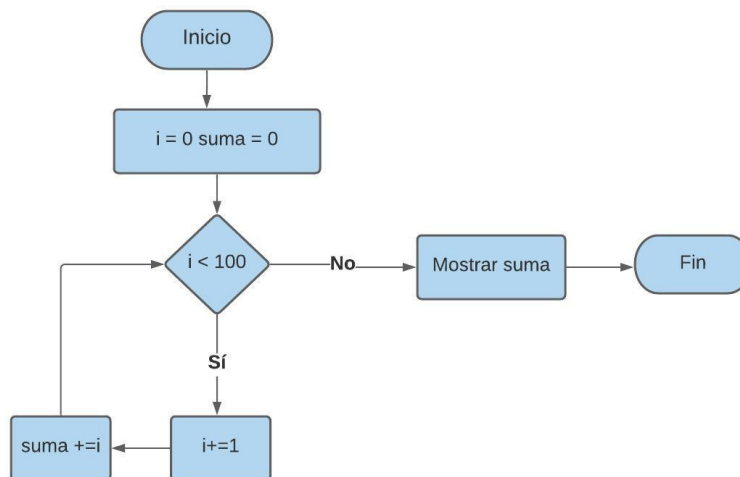


Imagen 1. Diagrama de flujo sumando hasta 100.

La instrucción `suma += i` es la encargada de aumentar el valor de la variable suma en cada iteración.

Este comportamiento, sumar y almacenar secuencialmente valores de variables, se conoce como la implementación de un **acumulador**. Un acumulador se diferencia de un contador, solo por el hecho que el acumulador se le suma un valor a la misma variable, y el contador aumenta su valor por un valor constante.

```
i = i + 1 // Contador
acum = acum + i //Acumulador
i++; //Cuenta de 1 en 1
i+=acum; //Acumula a i por el valor que tenga la variable acum
```

## Ejercicio guiado: Suma de números

### Contexto

Crear el programa `Suma.java`, donde el usuario ingresa un número, se muestra la suma de todos los números de 1 hasta ese número.

Ejemplo: si  $n = 100$ , el resultado es 5050.

1. Se debe crear el programa ejecutable main, donde se itere desde cero hasta el número ingresado por el usuario. En cada iteración se debe sumar cada número del conteo con el resultado anterior.

```
public class Ciclos {
    public static void main(String[] args) {
        //Se lee el valor ingresado por el usuario
        System.out.println("Ingrese un número");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        //i es la variable que contará hasta el número ingresado
        int i = 0;
        //suma es la variable que irá guardando el resultado
        int suma = 0;
        while (i < n){
            i+=1;
            suma += i;
        }
        System.out.printf("%d\n", suma);
    }
}
```