Material descargable

egg

# Teoría JAVA I

## Bucles

En programación, los bucles **son estructuras de control** que facilitan la ejecución repetida de un bloque de código mientras se cumpla una condición específica. Son esenciales para realizar tareas repetitivas de manera eficiente.

Existen tres tipos principales de bucles en Java:

- **Bucle for:** Este bucle es una estructura de control más compacta que combina la inicialización, la condición y el incremento (o decremento) en una sola línea. El bucle for se utiliza generalmente cuando el número de iteraciones es conocido de antemano.
- **Bucle while:** Este bucle ejecuta un bloque de código mientras una condición especificada sea verdadera. La condición se evalúa antes de cada iteración. Por lo tanto, si la condición es falsa desde el principio, el bloque de código dentro del bucle no se ejecutará ni una sola vez.
- **Bucle do-while:** Este bucle es similar al bucle while, pero la condición se evalúa después de que se ha ejecutado el bloque de código. Por lo tanto, el bloque de código se ejecutará al menos una vez, incluso si la condición es falsa desde el principio.

En este apartado, profundizaremos en el **bucle for**

### For i

**El bucle "for" se utiliza cuando conocemos la cantidad de veces que se debe repetir un bloque de código**. Tiene tres partes:

- **Inicialización** → Aquí es *donde se inicializa la variable de control* del bucle. *En la mayoría de los casos, esta variable se denomina "i"*.
  El uso de "i" es simplemente una convención; se utiliza tradicionalmente para indicar *"índice"*, pero en realidad puedes usar cualquier nombre de variable válido en Java.

```
for (int i = 0; ... ) {
    //Código del bucle
}
```

  En este ejemplo, *"int i = 0;"* inicializa la variable *"i"* con el valor 0.

- **Condición** → Esta es la *condición que se verifica antes de cada iteración del bucle*. Si la condición es verdadera (“true”), se ejecuta el bloque de código dentro del bucle. Si es falsa (“false”), el bucle se detiene.

```
for (... ; i < 5; ... ) {
    //Código del bucle
}
```

  En este ejemplo, *"i < 5"* es la condición. Por lo tanto, mientras el valor de *"i"* sea menor que 5, el bloque de código del bucle continuará ejecutándose.

- **Actualización** → En esta parte, *se actualiza la variable de control*. En la mayoría de los casos, simplemente se *incrementa* o *decrementa* la variable.

```
for (... ; ... ; i++) {
    //Código del bucle
}
```

  Aquí, *"i++"* incrementa el valor de *"i"* en 1 en cada iteración del bucle.

A continuación, te mostramos el *bucle "for"* completo a modo de referencia:

```
public static void main(String[] args) {
    for (int i = 0; i < 5; i++) {
        System.out.println("El valor de i es: " + i);
    }
}
```

Este bucle imprimirá los números del 0 al 4 en la consola. Después de cada iteración, "i" se incrementa en uno (i++), y mientras "i" sea menor que 5 (i < 5), el bucle continuará. Cuando "i" llegue a 5, la condición "i < 5" será falsa y el bucle se detendrá.

## For i & arrays

El **bucle "for" con "i"** es muy útil para acceder a los elementos de un array a través de sus *índices*.

Observemos estos dos ejemplos de cómo mostrar los elementos de un arreglo sin y con un bucle:

- *Sin* un bucle “for”:

```
public static void main(String[] args) {
    String[] paises = {"Uruguay","Argentina","Brasil","Venezuela"};
    System.out.println(paises[0]);
    System.out.println(paises[1]);
    System.out.println(paises[2]);
    System.out.println(paises[3]);
}
```

- *Con* un bucle “for”:

```
public static void main(String[] args) {
    String[] paises = {"Uruguay","Argentina","Brasil","Venezuela"};
    for (int i = 0; i < paises.length ; i++) {
        System.out.println(paises[i]);
    }
}
```

## For each

**El bucle "for-each" se utiliza para recorrer elementos en arreglos o colecciones sin tener que lidiar con índices**. (Las colecciones las veremos más adelante).

```
public static void main(String[] args) {
    int[] arr = {1, 2, 3, 4, 5};
    for (int num : arr) {
        System.out.println("El valor es: " + num);
    }
}
```

En este ejemplo, la *variable "num"* toma cada valor en el *arreglo "arr"* en cada iteración del bucle, lo que simplifica el proceso de iterar a través del arreglo sin necesidad de utilizar un índice.

💡Siempre es recomendable utilizar la **documentación oficial** para obtener más detalles sobre el material proporcionado, su uso y su implementación.