

# 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 while y do-while**

### While

**El bucle "while" se utiliza cuando se desea repetir un bloque de código siempre que una condición sea verdadera.**

En este bucle, primero se evalúa la condición. Si la condición es verdadera, se ejecuta el bloque de código dentro del bucle. Después de ejecutar el bloque de código, se vuelve a evaluar la condición. Si la condición sigue siendo verdadera, el bloque de código se ejecuta nuevamente. Este proceso se repite hasta que la condición se evalúe como falsa.

Es importante tener cuidado al utilizar el bucle "while" para evitar caer en un bucle infinito, donde la condición nunca se evalúe como falsa y el bucle nunca termine de ejecutarse. Por lo tanto, dentro del bloque de código del bucle "while", generalmente se debe incluir algún mecanismo para cambiar la condición y eventualmente hacer que sea falsa. De lo contrario, el bucle se ejecutará indefinidamente.

Veamos el siguiente ejemplo:

```
public static void main(String[] args) {  
    int numeroAleatorio = 0;  
    while (numeroAleatorio < 8) {  
        numeroAleatorio = (int) (Math.random()*(10-0+1)+0);  
    }  
    System.out.println("Seguro es 8 o mayor a 8: "+numeroAleatorio);  
}
```

Aquí, el bloque de código se ejecuta mientras "numeroAleatorio" sea menor que 8, y en cada iteración del bucle, el valor de "numeroAleatorio" cambia.

## Do While

**El bucle "do-while" es una estructura de control en programación que garantiza que un bloque de código se ejecute al menos una vez, incluso si la condición es inicialmente falsa.** A diferencia del bucle "while", donde la condición se evalúa antes de ejecutar el bloque de código, en el bucle "do-while" la condición se evalúa después de la ejecución del bloque.

**Aquí hay un resumen de cómo funciona el bucle "do-while":**

- **Primero, se ejecuta el bloque de código dentro del "do".**
- **Luego, se evalúa la condición especificada después del "while".**
- **Si la condición es verdadera, el bloque de código se ejecuta nuevamente. Si es falsa, el bucle termina y la ejecución continúa después del bucle.**

El bucle "do-while" es útil cuando se necesita ejecutar un bloque de código al menos una vez, independientemente de la condición. También se utiliza cuando se desea repetir un bloque de código basado en una condición específica después de que se haya ejecutado al menos una vez.

Observemos el siguiente código:

```
public static void main(String[] args) {
    Scanner pepe = new Scanner(System.in);
    int num;
    do {
        System.out.print("Por favor, ingrese un número mayor a 0: ");
        num = pepe.nextInt();
    } while (num <= 0);
    System.out.println("Ingresaste: " + num);
}
```

En este ejemplo, el programa solicita al usuario que ingrese un número, y si el número es menor o igual a 0, el programa seguirá solicitando al usuario que ingrese otro número. Esto continuará hasta que el usuario ingrese un número mayor que 0.

Si intentamos lograr lo mismo con un *bucle "while"*, tendríamos que repetir el código para leer el número del usuario:

```
public static void main(String[] args) {
    Scanner pepe = new Scanner(System.in);
    System.out.print("Por favor, ingrese un número mayor a 0: ");
    int num = pepe.nextInt();
    while (num <= 0) {
        System.out.print("Por favor, ingrese un número mayor a 0: ");
        num = pepe.nextInt();
    }
    System.out.println("Ingresaste: " + num);
}
```

Esto se debe a que el *bucle "while"* evalúa la condición antes de entrar al bloque de código, por lo que necesitamos solicitar al usuario que ingrese un número antes del bucle para inicializar "num". Luego, dentro del bucle, necesitamos volver a solicitar al usuario que ingrese otro número si el número anterior no era válido.



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