

BUCKLES

INSTRUCTOR

I.S.C. E.T.W. M.T.W. CARLOS URIEL DE JESÚS SÁNCHEZ GONZÁLEZ

[HTTPS://WWW.LINKEDIN.COM/IN/CARLOS-URIEL-DE-JESUS-SANCHEZ-GONZALEZ-8920A1372/](https://www.linkedin.com/in/carlosurieldejesussanchezgonzalez-8920a1372/)

WHILE
EJECUTA EL BLOQUE MIENTRAS LA CONDICIÓN SEA
VERDADERA.

```
int contador = 1;  
  
while (contador <= 5) {  
  
    System.out.println("Contador: " + contador);  
  
    contador++; // importante incrementar para evitar bucle infinito  
  
}
```

DO-WHILE
SE EJECUTA AL MENOS UNA VEZ, Y LUEGO REPITE
MIENTRAS LA CONDICIÓN SEA VERDADERA.

```
int numero = 1;  
  
do {  
  
    System.out.println("Número: " + numero);  
  
    numero++;  
  
} while (numero <= 5);
```

FOR CLÁSICO

IDEAL CUANDO SABES CUÁNTAS VECES QUIERES REPETIR.

```
for (int i = 1; i <= 5; i++) {  
    System.out.println("Iteración: " + i);  
}
```

FOR-EACH (ENHANCED FOR)
SE USA PARA RECORRER COLECCIONES O ARREGLOS DE
FORMA SENCILLA.

```
String[] frutas = {"Manzana", "Banana", "Naranja";  
for (String fruta : frutas) {  
    System.out.println("Fruta: " + fruta);  
}
```

STREAM (DESDE JAVA 8)

FORMA FUNCIONAL Y DECLARATIVA DE RECORRER COLECCIONES.

```
List<String> nombres = List.of("Ana", "Pedro", "Luis", "Maria");  
  
// forEach con lambda  
  
nombres.stream()  
  
    .forEach(nombre -> System.out.println("Nombre: " + nombre));  
  
// Filtrar y recorrer  
  
System.out.println("\nNombres con más de 3 letras:");  
  
nombres.stream()  
  
    .filter(n -> n.length() > 3)  
  
    .forEach(System.out::println);
```

BUENAS PRÁCTICAS CON BUCLES

- Usa `for-each` cuando solo necesitas recorrer la colección, sin índices.
- Usa `while` o `do-while` si no sabes cuántas iteraciones habrá (ejemplo: leer hasta que el usuario escriba "salir").
- Evita bucles infinitos asegurándote de actualizar la condición dentro del ciclo.
- Prefiere Streams cuando necesites operaciones de filtrado, mapeo o transformaciones en colecciones.
- No mezcles demasiada lógica dentro del bucle → extrae a métodos si es necesario.