

## Apunte Actividad 2

### Estructuras de Repetición: while, for, do-while, Contadores y Acumuladores

#### 1. Introducción

Las **estructuras de repetición** permiten ejecutar un bloque de código varias veces, hasta que una condición deje de cumplirse. Son esenciales cuando necesitamos automatizar tareas repetitivas.

En Java tenemos tres estructuras principales:

- while
- for
- do-while

#### 2. Bucle while

El ciclo while ejecuta un bloque de código **mientras** una condición sea verdadera.

```
int contador = 1;

while (contador <= 5) {
    System.out.println("Número: " + contador);
    contador++;
}
```

- Este código imprime los números del 1 al 5.
- Cuidado: si olvidamos incrementar la variable, se produce un **bucle infinito**.

#### 3. Bucle for

El ciclo for se utiliza cuando sabemos cuántas veces queremos repetir algo.

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

- Hace lo mismo que el while anterior, pero de forma más compacta.

#### 4. Bucle do-while

El ciclo do-while se diferencia en que el bloque se ejecuta **al menos una vez**, aunque la condición sea falsa desde el inicio.

```
int numero = 0;

do {
    System.out.println("Número: " + numero);
    numero++;
} while (numero < 3);
```

- Este ejemplo imprime los números 0, 1, 2.

## 5. Contadores y Acumuladores

- **Contador:** Variable que aumenta o disminuye en cada iteración.
- **Acumulador:** Variable que **suma valores** en cada iteración.

*Ejemplo:*

```
int suma = 0; // acumulador
int contador = 0; // contador

for (int i = 1; i <= 5; i++) {
    suma += i;        // acumula la suma de los números
    contador++;       // cuenta las iteraciones
}

System.out.println("Suma total: " + suma);        // 15
System.out.println("Cantidad de números: " + contador); // 5
```

## 6. Ciclos Anidados

Podemos usar un bucle dentro de otro. Esto se conoce como **ciclo anidado**.

*Ejemplo:* tabla de multiplicar

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 3; j++) {
        System.out.println(i + " x " + j + " = " + (i * j));
    }
}
```

- Genera la tabla de multiplicar del 1 al 3.

## 7. Control de Flujo dentro de bucles

- **break**: interrumpe el bucle.
- **continue**: salta a la siguiente iteración.

```
for (int i = 1; i <= 5; i++) {  
    if (i == 3) {  
        continue; // salta el 3  
    }  
    if (i == 5) {  
        break;    // detiene en el 5  
    }  
    System.out.println("Número: " + i);  
}
```