

## Apunte Actividad 3

### Funciones (Métodos), Parámetros, Retorno y Javadoc

#### 1. ¿Qué es una función (método) en Java?

En Java, una **función** se denomina **método**.

Un método es un bloque de código con un nombre que puede ser llamado desde cualquier parte del programa para **evitar repeticiones** y **organizar mejor el código**.

Ventajas de usar métodos:

- Mejoran la **modularidad**.
- Evitan la **duplicación de código**.
- Facilitan la **lectura** y **mantenimiento**.

#### 2. Estructura de un método en Java

La sintaxis básica es:

```
modificador tipoRetorno nombreMetodo(parámetros) {  
  
    // cuerpo del método  
  
}
```

- **modificador**: nivel de acceso (public, private, etc.).
- **tipoRetorno**: tipo de dato que devuelve (int, double, String, void si no devuelve nada).
- **nombreMetodo**: identificador (por convención empieza en minúscula).
- **parámetros**: lista de variables que recibe el método (pueden ser cero o más).

#### 3. Métodos sin parámetros y sin retorno

Se usan cuando solo realizan una acción.

```
public void saludar() {  
    System.out.println("¡Hola, bienvenido!");  
}
```

Uso:

```
saludar();
```

#### 4. Métodos con parámetros

Permiten **reutilizar código** enviando diferentes valores.

```
public void saludarPersona(String nombre) {  
    System.out.println("Hola " + nombre + "!");  
}
```

Uso:

```
saludarPersona("Ana");  
saludarPersona("Pedro");
```

#### 5. Métodos con retorno

Devuelven un valor al lugar donde fueron llamados.

```
public int sumar(int a, int b) {  
    return a + b;  
}
```

Uso:

```
int resultado = sumar(5, 3);  
System.out.println("La suma es: " + resultado);
```

#### 6. Alcance de Variables (Scope)

El **alcance** define desde qué parte del programa una variable puede ser usada:

- **Variable local:** declarada dentro de un método, solo existe ahí.
- **Variable de clase (atributo):** declarada en la clase, accesible en todos los métodos de esa clase.

Ejemplo:

```
public class Ejemplo {  
  
    int contador = 0; // variable de clase  
  
    public void incrementar() {  
        int x = 5; // variable local  
        contador++;  
        System.out.println("Contador: " + contador);  
    }  
}
```

## 7. Sobrecarga de métodos (Overloading)

Podemos definir varios métodos con el **mismo nombre** pero con **diferentes parámetros**.

```
public int sumar(int a, int b) {  
    return a + b;  
}  
  
public double sumar(double a, double b) {  
    return a + b;  
}
```

## 8. Documentación con Javadoc

Javadoc permite **generar documentación automática** de los métodos.

Se escribe arriba del método con `/** ... */`.

*Ejemplo:*

```
/**  
 * Suma dos números enteros.  
 *  
 * @param a primer número  
 * @param b segundo número  
 * @return la suma de a y b  
 */  
public int sumar(int a, int b) {  
    return a + b;  
}
```

- Luego se puede generar la documentación con el comando:

`javadoc NombreArchivo.java`