

Apunte Actividad 2

toString(), static, equals() y hashCode()

1. El método toString()

En Java, **todas las clases heredan de Object**, y con ello el método toString().

Por defecto, devuelve algo poco legible (ejemplo: Persona@3f99bd52).

- **Sobrescribirlo** permite mostrar el estado del objeto de manera clara:

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
  
    public Persona(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
  
    @Override  
    public String toString() {  
        return "Persona [nombre=" + nombre + ", edad=" + edad + "]";  
    }  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
        Persona p = new Persona("Ana", 25);  
        System.out.println(p); // Persona [nombre=Ana, edad=25]  
    }  
}
```

2. Miembros estáticos (static)

La palabra clave **static** indica que un atributo o método pertenece a la **clase**, no a los objetos.

- **Atributo estático**: compartido por todas las instancias.
- **Método estático**: se llama desde la clase, sin necesidad de crear un objeto.

Ejemplo:

```
public class Calculadora {  
  
    public static final double PI = 3.1416;  
  
    public static int sumar(int a, int b) {  
        return a + b;  
    }  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
        System.out.println(Calculadora.PI);           // 3.1416  
        System.out.println(Calculadora.sumar(5, 3)); // 8  
    }  
}
```

Útil para **constantes** y **funciones utilitarias**.

3. equals() y hashCode()

En Java, el operador == compara **referencias** (si apuntan al mismo objeto en memoria).

Para comparar **contenido**, debemos sobrescribir equals().

3.1 Sobrescribir equals()

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
  
    public Persona(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
  
    @Override  
    public boolean equals(Object obj) {  
        if (this == obj) {  
            return true; // misma referencia  
        }  
        if (obj == null || getClass() != obj.getClass()) {  
            return false;  
        }  
        Persona otra = (Persona) obj;  
        return edad == otra.edad && nombre.equals(otra.nombre);  
    }  
}
```

3.2 Sobrescribir hashCode()

Cuando sobrescribimos equals(), también debemos sobrescribir hashCode(). Esto es clave en estructuras como HashSet o HashMap.

```
@Override  
public int hashCode() {  
    return java.util.Objects.hash(nombre, edad);  
}
```

Ejemplo de uso:

```
Persona p1 = new Persona("Ana", 25);  
Persona p2 = new Persona("Ana", 25);  
  
System.out.println(p1.equals(p2)); // true  
System.out.println(p1.hashCode() == p2.hashCode()); // true
```