

Apunte Actividad 3

Encapsulamiento y Modificadores de Acceso

1. ¿Qué es el Encapsulamiento?

El **encapsulamiento** es un principio de la POO que consiste en **ocultar los detalles internos de una clase** y exponer únicamente lo necesario a través de métodos controlados.

Beneficios:

- Protege la **integridad de los datos**.
- Permite **controlar el acceso** a los atributos.
- Mejora la **modularidad y reutilización** del código.

2. Modificadores de Acceso

En Java, los modificadores de acceso definen el nivel de visibilidad de atributos, métodos y clases.

- **public** → accesible desde cualquier parte del programa.
- **private** → accesible solo dentro de la misma clase.
- **(default)** → accesible solo dentro del mismo paquete.
- **protected** → accesible dentro del mismo paquete y por clases hijas (herencia).

Ejemplo:

```
public class Persona {  
  
    public String nombre;    // acceso libre  
    private int edad;        // acceso restringido  
}
```

3. Getters y Setters

Como los atributos privados no pueden ser accedidos directamente, usamos **métodos públicos** llamados **getters** (para leer datos) y **setters** (para modificarlos).

Ejemplo:

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
  
    // Getter  
    public String getNombre() {  
        return nombre;  
    }  
  
    // Setter  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    // Getter y Setter para edad  
    public int getEdad() {  
        return edad;  
    }  
  
    public void setEdad(int edad) {  
        if (edad >= 0) { // controlamos la integridad de los datos  
            this.edad = edad;  
        }  
    }  
}
```

Uso en el main:

```
public class Main {  
  
    public static void main(String[] args) {  
        Persona p1 = new Persona();  
  
        p1.setNombre("Lucía");  
        p1.setEdad(22);  
  
        System.out.println("Nombre: " + p1.getNombre());  
        System.out.println("Edad: " + p1.getEdad());  
    }  
}
```

Observa cómo el **encapsulamiento** nos permite **controlar la validez de los datos**, ya que evitamos asignar edades negativas.

4. Encapsulamiento en NetBeans

En IDEs como NetBeans, puedes generar automáticamente **getters y setters** con un clic derecho → *Insert Code* → *Getter and Setter*.

Esto agiliza la creación de clases y refuerza el principio de encapsulamiento.