# Encapsulamiento, Herencia y Polimorfismo.

# Encapsulamiento

El encapsulamiento es un principio fundamental de la programación orientada a objetos (POO) que consiste en ocultar los atributos y el estado interno de un objeto, permitiendo que el acceso a ellos se realice únicamente a través de métodos públicos llamados getters y setters. Con este mecanismo se protege la integridad de los datos, ya que evita que sean modificados de manera directa desde fuera de la clase. Además, el encapsulamiento facilita la seguridad, la reutilización de código y la facilidad de mantenimiento, porque garantiza que el control sobre los atributos se realice de forma controlada y coherente.

# Herencia

La herencia es un mecanismo que permite crear nuevas clases a partir de otras ya existentes, estableciendo una relación de jerarquía. La clase que se hereda se llama clase padre o superclase, mientras que la clase que recibe los atributos y métodos se llama clase hija o subclase. Gracias a la herencia, se logra la reutilización de código, evitando tener que reescribir las mismas funcionalidades en distintas clases. También permite crear clases más especializadas a partir de clases generales, lo cual mejora la organización y comprensión del programa.

# Polimorfismo

El polimorfismo es la capacidad que tienen los objetos de adoptar distintas formas o comportamientos a través de un mismo método. En la programación orientada a objetos, esto significa que un método definido en una clase padre puede ser sobrescrito por las clases hijas, de manera que, al invocarlo, el comportamiento cambie según el tipo de objeto que lo utilice. Este concepto hace que el código sea más flexible y dinámico, ya que permite trabajar con colecciones de objetos de diferentes tipos sin necesidad de saber exactamente de qué clase provienen, fomentando la extensibilidad y el mantenimiento de los programas.