**Términos de Programación Orientada a Objetos**

**Encapsulamiento y Ocultación de Datos**

**Encapsulamiento** es el principio que consiste en agrupar los datos (atributos) y los métodos que operan sobre ellos dentro de una misma unidad, generalmente una clase. Esto permite **controlar el acceso** y modificar el estado interno del objeto solo a través de interfaces públicas definidas (métodos).

**Ocultación de datos** es una consecuencia del encapsulamiento y se refiere a **proteger los datos internos del objeto** del acceso directo por parte de código externo. Se logra utilizando modificadores de acceso como private, protected y public.

**Herencia**

La **herencia** es el mecanismo que permite a una clase (subclase o clase derivada) **heredar atributos y métodos** de otra clase (superclase o clase base). Esto favorece la reutilización de código y la creación de jerarquías

**Reutilización**

**Reutilización** es la capacidad de usar código existente (clases, métodos, estructuras) para **resolver nuevos problemas sin reescribirlo**. Se logra principalmente a través de la herencia y la composición.

**Polimorfismo**

**Polimorfismo** significa “muchas formas” y permite que **un mismo método o función tenga comportamientos diferentes** según el contexto o el tipo del objeto que lo invoque.

Hay dos tipos principales:

* **Polimorfismo en tiempo de compilación (estático)**: mediante sobrecarga de funciones.
* **Polimorfismo en tiempo de ejecución (dinámico)**: mediante herencia y métodos sobrescritos.

**Sobrecarga de Operadores**

La **sobrecarga de operadores** permite redefinir el comportamiento de los operadores estándar (+, -, \*, etc.) cuando se aplican a **objetos de clases definidas por el usuario**.

**Sobrecarga de funciones (Overloading)**

**Sobrecarga de funciones** permite definir **múltiples funciones con el mismo nombre**, pero con diferentes tipos o números de parámetros. Esto permite mayor flexibilidad y legibilidad.

Ejemplo: Una función calcular() puede tener versiones que reciben un entero, un flotante, o dos parámetros.

En Python se simula con parámetros opcionales o **\*args**

**Mensaje**

En POO, un **mensaje** es la forma en que **un objeto solicita a otro que ejecute uno de sus métodos**. Es el equivalente a **invocar un método en un objeto**.

Ejemplo: Objeto.mostrar() es un mensaje al objeto Objeto para que ejecute su método mostrar.