

Reflexión Final

La POO permite a los desarrolladores crear objetos que representan entidades del mundo real, como clientes, productos y pedidos. Estos objetos pueden interactuar entre sí, lo que facilita la creación de sistemas complejos y escalables. La POO también promueve la reutilización del código a través de la herencia y el polimorfismo, lo que reduce el tiempo de desarrollo y mejora la mantenibilidad del software.

Reflexión Personal

Clases:

Modularidad: Encapsulan datos y comportamientos relacionados, facilitando la comprensión y el mantenimiento del código.

Reutilización: Pueden ser reutilizadas en diferentes partes del sistema, reduciendo la duplicación de código.

Herencia:

Modularidad: Permite crear nuevas clases basadas en clases existentes, facilitando la extensión de funcionalidades.

Reutilización: Promueve la reutilización del código al heredar atributos y métodos de la clase base.

Polimorfismo:

Modularidad: Permite que diferentes clases respondan a la misma interfaz o método de manera específica, creando sistemas flexibles.

Reutilización: Permite utilizar objetos de diferentes clases de manera intercambiable, mejorando la integración de nuevas funcionalidades.