

Concepto de encapsulamiento (abstracción)

Definición

El encapsulamiento es uno de los principios fundamentales de la POO. Consiste en ocultar los detalles internos de un objeto y exponer solo la interfaz necesaria para su uso.

Abstracción

El encapsulamiento permite la abstracción, donde se enfoca en las características y comportamientos esenciales de un objeto, ocultando los detalles de implementación.

Beneficios

El encapsulamiento mejora la modularidad, la reutilización de código y la mantenibilidad de los sistemas.

Beneficios del encapsulamiento

1

Abstracción

Permite ocultar los detalles de implementación y exponer solo la interfaz necesaria.

2

Modularidad

Facilita la organización del código en módulos independientes y reutilizables.

3

Mantenibilidad

Cambios en la implementación interna no afectan el uso del objeto.

4

Seguridad

Evita el acceso directo a datos internos y previene errores de uso.

Concepto de herencia (implementación)

Definición

La herencia es un mecanismo de la POO que permite crear nuevas clases (subclases) a partir de clases existentes (superclases).

Implementación

Las subclases heredan atributos y métodos de la superclase, lo que facilita la reutilización de código y la creación de jerarquías de clases.

Polimorfismo

La herencia también permite el polimorfismo, donde objetos de subclases pueden ser tratados como objetos de la superclase.

- Pilar de la POO

Herencia

Simple : Una clase solo puede heredar de una clase padre.

Múltiple: Una clase puede heredar de más de una clase padre.



Tipos de herencia en POO

1

Herencia simple

Una subclase hereda de una sola superclase.

2

Herencia múltiple

Una subclase hereda de múltiples superclases.

3

Herencia multinivel

Una subclase hereda de una super clase, que a su vez hereda de otra superclase.

Concepto de subtipos (interfaz)

Definición

Los subtipos, también conocidos como interfaces, definen un contrato de métodos que una clase debe implementar.

Abstracción

Las interfaces permiten una mayor abstracción, al definir las operaciones que una clase debe proporcionar sin especificar la implementación.

Polimorfismo

Las interfaces facilitan el polimorfismo, ya que objetos de diferentes clases pueden ser tratados de forma intercambiable si implementan la misma interfaz.

Ventajas de las interfaces



Modularidad

Las interfaces mejoran la modularidad y la flexibilidad del diseño.



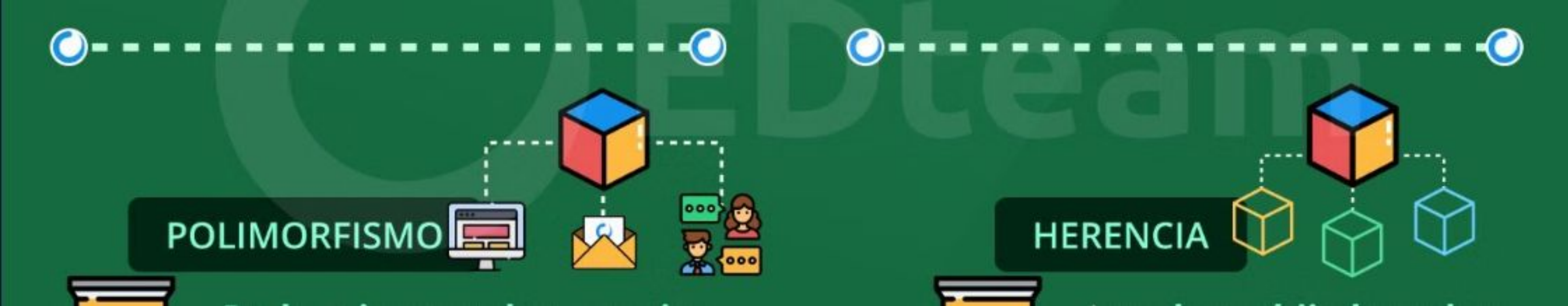
Reutilización

Facilitan la reutilización de código al desacoplar la implementación de la interfaz.



Extensibilidad

Permiten extender el sistema sin afectar el código existente.



Conclusión y resumen de los conceptos clave

1

Encapsulamiento

Ocultar los detalles internos y expone sólo la interfaz necesaria.

2

Herencia

Permite la reutilización de código y la creación de jerarquías de clases.

3

Subtipos (Interfaces)

Definen un contrato de métodos que una clase debe implementar.