<!--SOLID-->

Que es la programación orientada a objetos? {

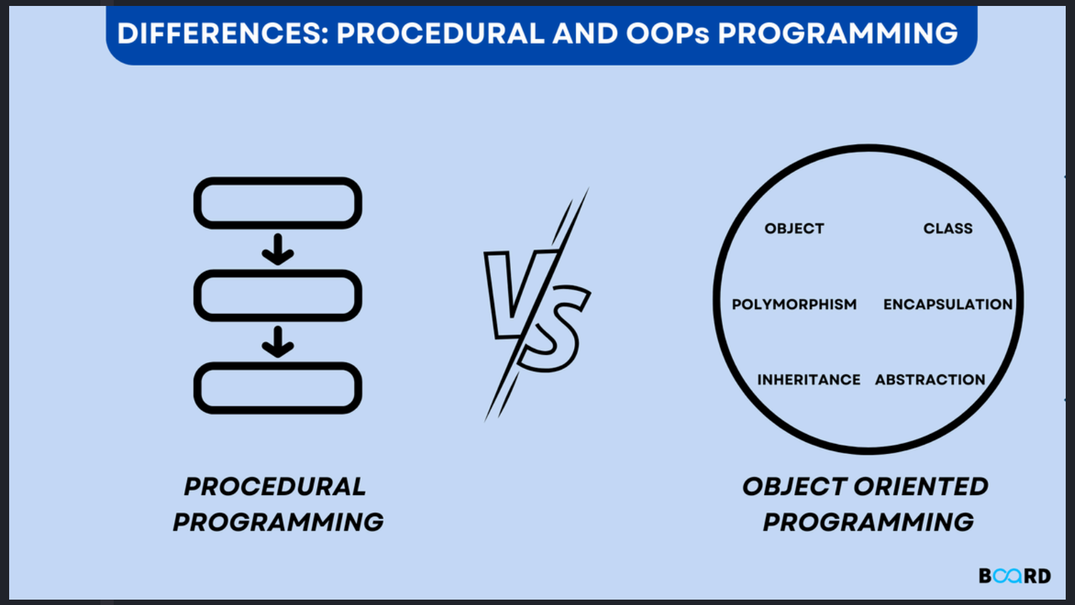
<Por="Raquel Martinez"/>

}

Contenidos

Paradigma {

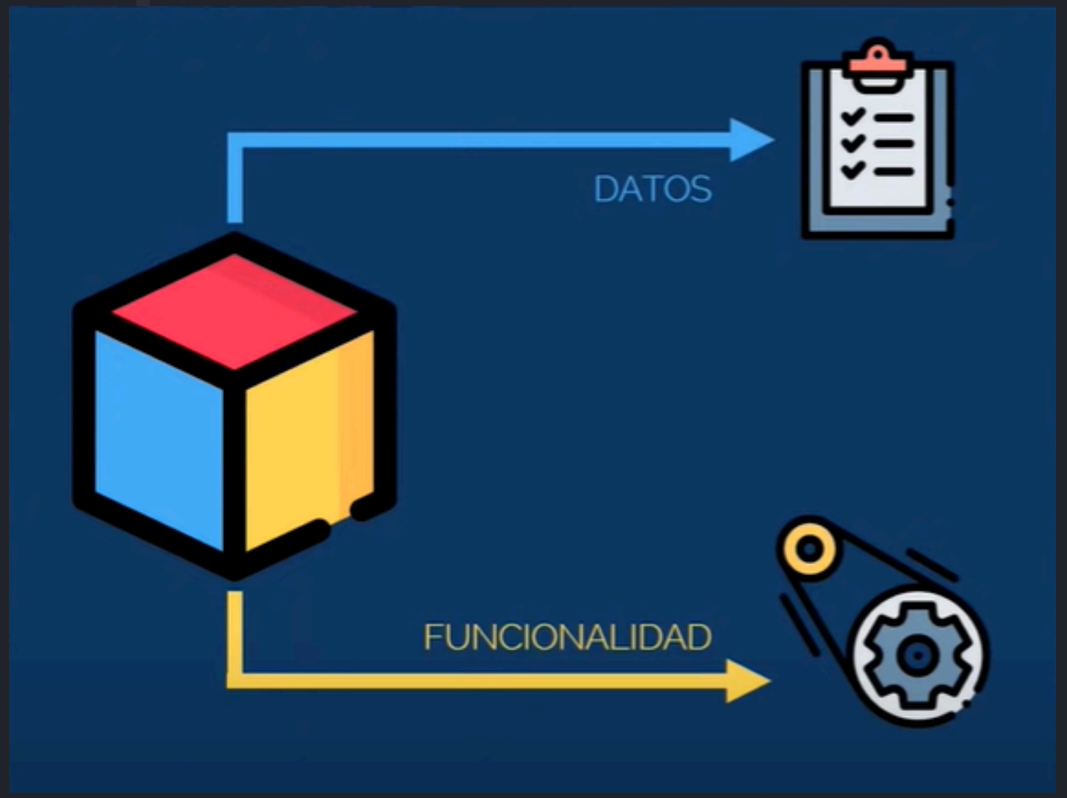
- Basado en el concepto de “objetos”
- Objetos de la vida real
- Objeto -> datos y comportamientos



}

Paradigma {

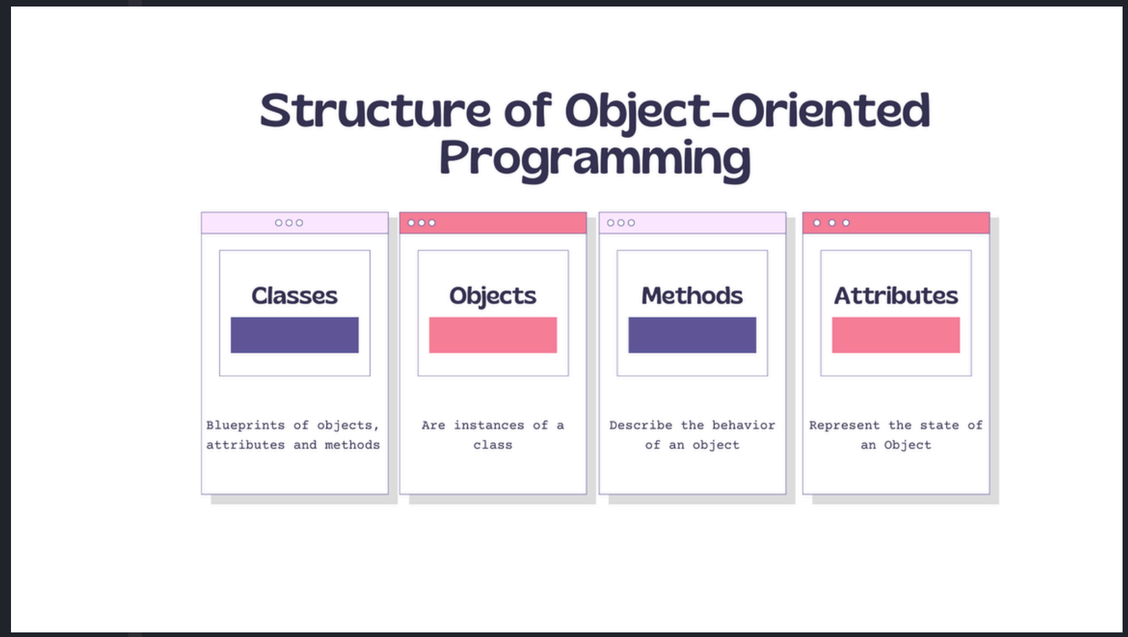
- Basado en el concepto de “objetos”
- Objetos de la vida real
- Objeto -> datos y comportamientos



}

Clases{

- Atributos
- Métodos de instancia
- Métodos de clase
- Métodos estáticos



}

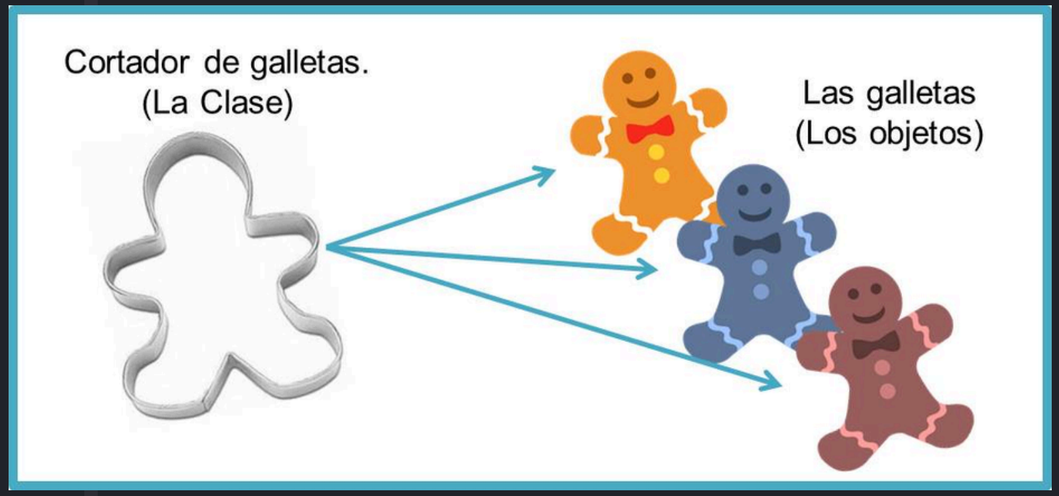
Instancia y objeto {

Objeto:

- Un objeto es una entidad que representa una instancia concreta de una clase. Métodos de instancia

Instancia:

- Una instancia es cada objeto individual creado a partir de una clase.



}

<!--SOLID-->

Codifiquemos! {

<Ejemplo/>

}

Abstracción {

Captar características

Separar caracteristicas

Una abstracción se enfoca en la visión externa de un objeto, separa el comportamiento específico de un objeto

}

Encapsulación {

Atributos publicos

Accesibles por instancias

Atributos privados

Los atributos privados se definen con un doble guión bajo (__) al comienzo de su nombre.

Atributos protegidos

Indica que el atributo no debe ser accedido directamente desde fuera de la clase, pero no impide su acceso.

Principio de ocultar los detalles internos de un objeto y controlar el acceso a sus atributos y métodos

}

Herencia

Reutilización de código

Promueve la reutilización del código y evitando la duplicación de funcionalidades.

Estructura jerárquica

Clases más específicas (subclases) pueden heredar características de clases más generales (superclases).

Abstracción de comportamiento común

La herencia facilita la creación de clases que encapsulan comportamientos comunes

La herencia permite que una clase hija herede atributos y métodos de una clase padre. Esto promueve la reutilización de código y facilita la creación de jerarquías de clases.

}

Polimorfismo

Flexibilidad en el código

El polimorfismo permite escribir código más flexible y genérico al tratar objetos de diferentes clases de manera uniforme si comparten una interfaz común.

Facilita la extensión del software

nuevas clases pueden implementar la misma interfaz que las clases existentes y ser utilizadas en el mismo contexto.

El polimorfismo es un concepto de la programación orientada a objetos que se refiere a la capacidad de diferentes objetos de responder de manera diferente a la misma invocación de método.

}

<!--SOLID-->

Gracias {

<Por="Raquel Martinez/>

}