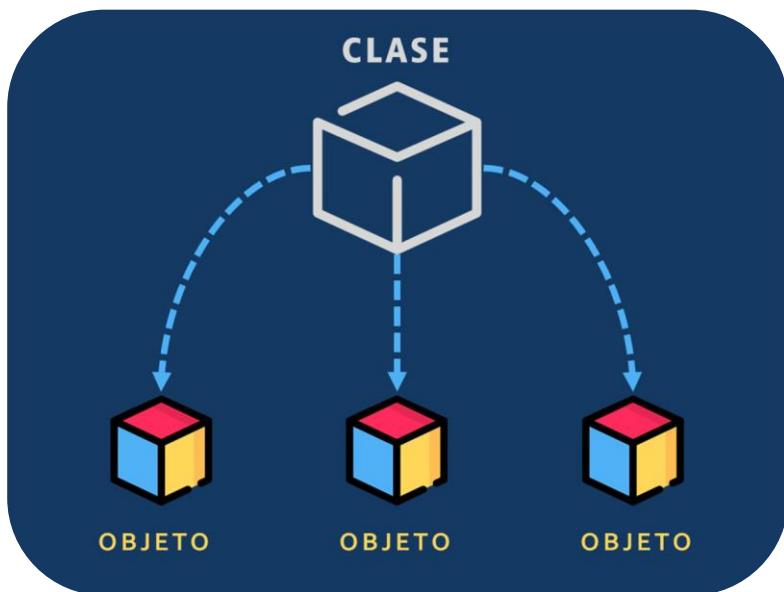


"PROGRAMACIÓN ORIENTADA A OBJETOS (POO)"

César Gustavo Chan Tun

Es un paradigma de programación, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.



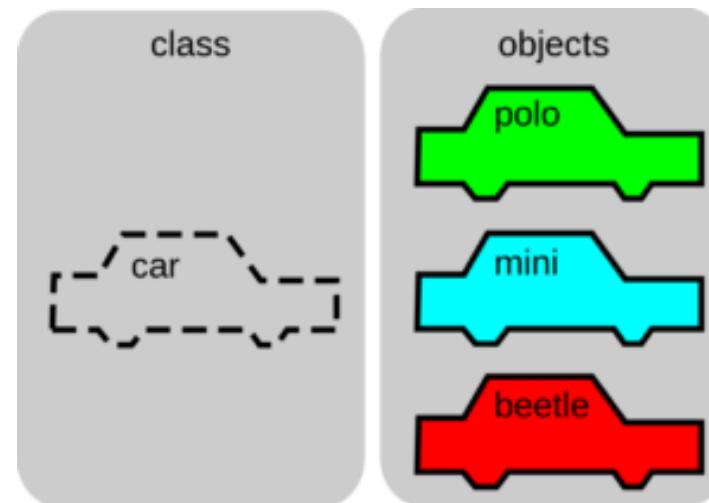
¿Qué es la Programación Orientada a Objetos?

Clase en POO

Una clase es una plantilla o molde para crear objetos. Define las propiedades (atributos) y comportamientos (métodos) que tendrán los objetos.

Atributo: Característica o propiedad de un objeto.

Método: Acción o comportamiento que un objeto puede realizar.



Objetos en POO

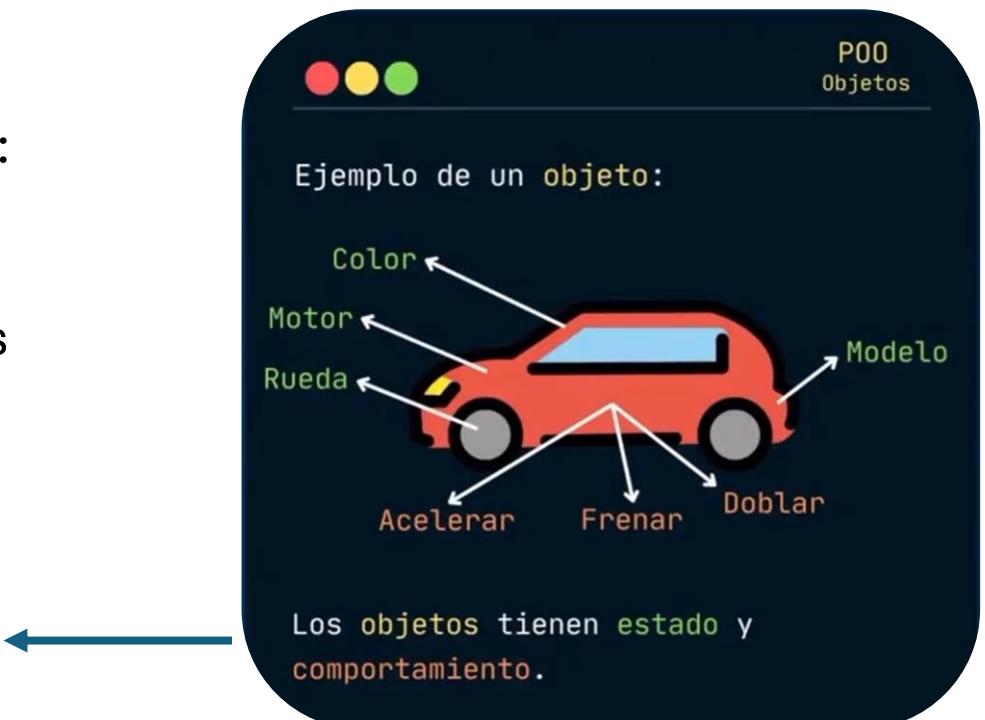
Un objeto es una instancia de una clase.

Es la representación de una entidad.

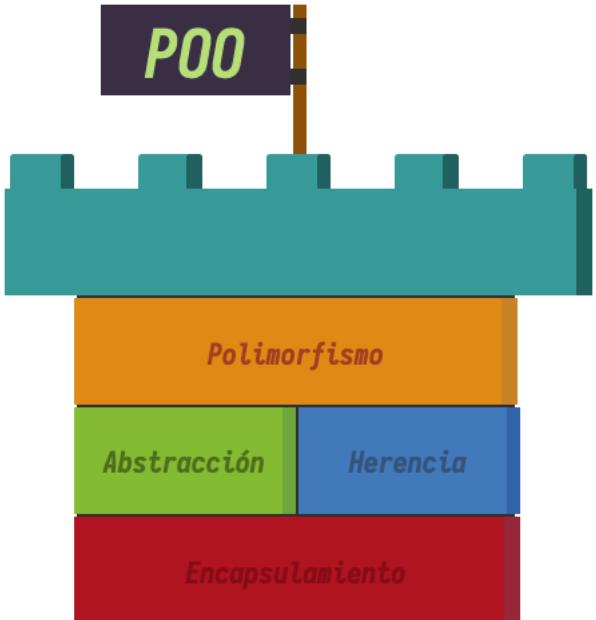
Contiene toda la información necesaria para abstraerlo:

- Datos que describen sus atributos.
- Operaciones que pueden realizarse sobre los mismos (métodos).

El estado de un objeto se define por los valores de sus atributos un momento determinado.



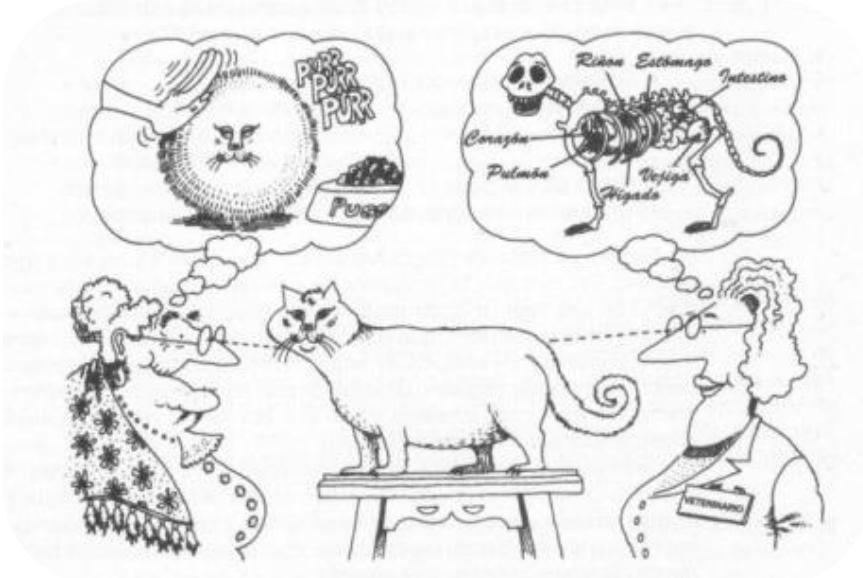
Los 4 pilares de la POO:



- 1 Abstracción.**
- 2 Encapsulamiento.**
- 3 Herencia.**
- 4 Polimorfismo.**

Estos principios definen la manera en que se estructura y organiza el código en este paradigma, permitiendo escribir programas más modulares, reutilizables y escalables.

Proceso de ocultar detalles innecesarios y mostrar solo lo esencial.

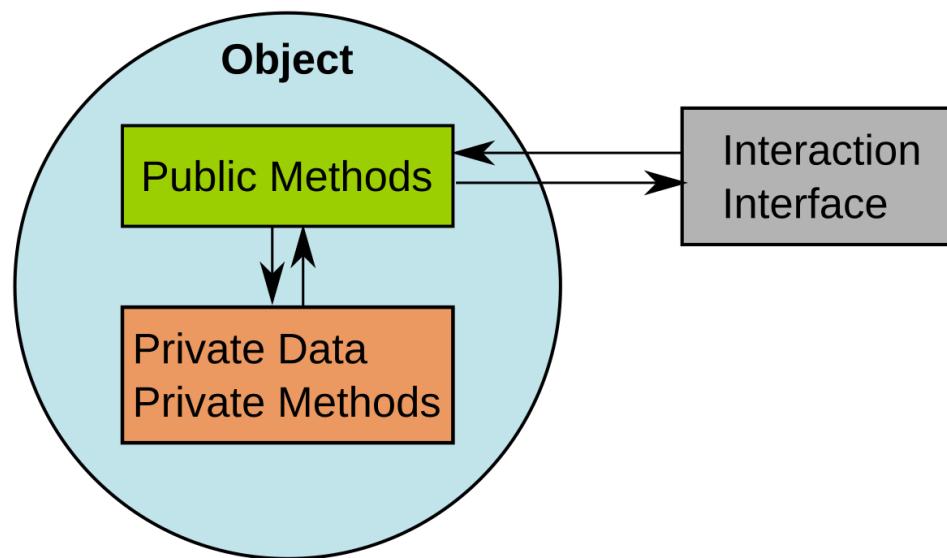


Se centra en las características esenciales de un objeto, en relación con la perspectiva del espectador, ocultando los detalles innecesarios y mostrando solo lo relevante para su uso.

Abstracción

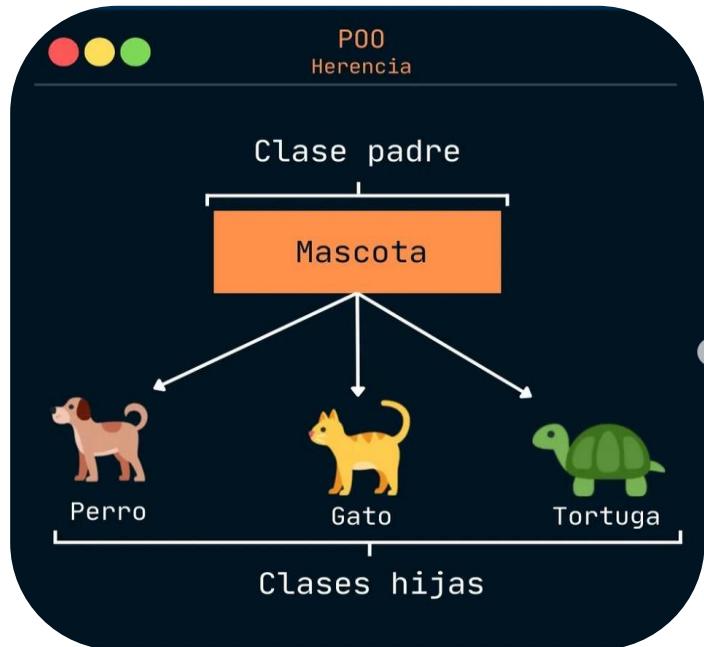
Encapsulamiento

Técnica de proteger los datos de un objeto evitando el acceso directo.



Se logra mediante atributos privados y métodos de acceso, garantizando seguridad e integridad en el programa.

Mecanismo que permite que una clase (subclase o hija) herede atributos y métodos de otra clase (superclase o padre).



Permite reutilizar código, agregar nuevos métodos o atributos en la subclase y también se puede modificar métodos heredados.

Permite que una clase hija amplíe el comportamiento sin modificar la clase padre.

Herencia

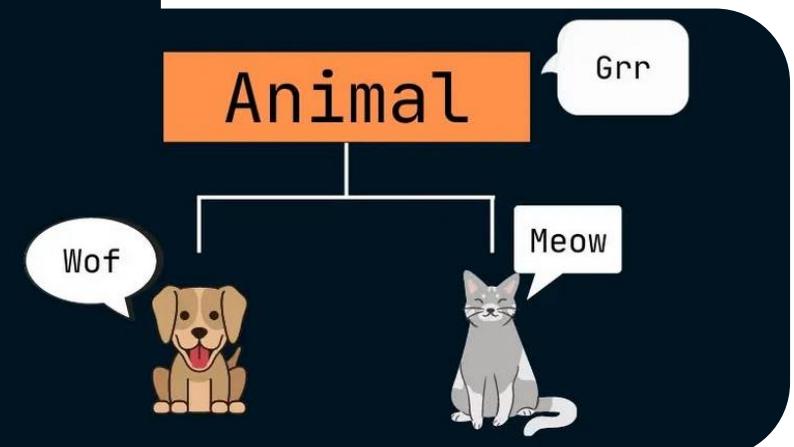
Polimorfismo

Es la capacidad de usar el mismo método en diferentes clases, pero con diferentes implementaciones.

POO
Polimorfismo

```
class Animal {  
    public void hacerSonido() {  
        System.out.println("Grr");  
    }  
  
class Perro extends Animal {    class Gato extends Animal {  
    public void hacerSonido() {        public void hacerSonido() {  
        System.out.println("Wof");            System.out.println("Meow");  
    }                                }  
}  
  
public static void main(String[ ] args) {  
    Animal a = new Perro();  
    Animal b = new Gato();  
  
    a.hacerSonido();  
    //Muestra "Wof"  
    b.hacerSonido();  
    //Muestra "Meow"  
}
```

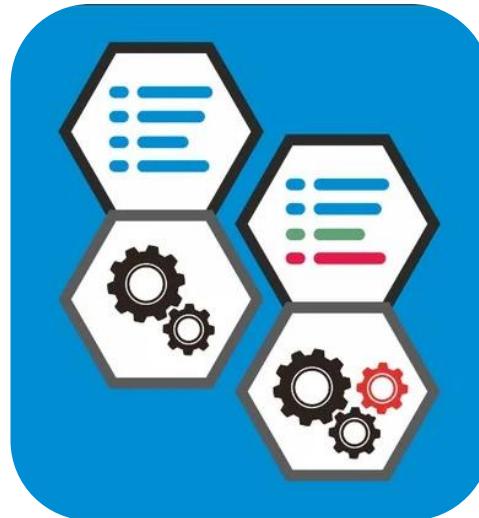
Que una misma acción se realice de diferentes maneras, dependiendo de quién la realice.



Beneficios y Aplicaciones de la POO

Beneficios

- Reutilización de código.
- Modularidad y escalabilidad.
- Facilidad de mantenimiento.
- Facilita el trabajo en equipo.



Aplicaciones

- Desarrollo de software empresarial.
- Aplicaciones móviles.
- Videojuegos.
- Aplicaciones web.
- Automatización y robótica.

La POO es un paradigma ampliamente utilizado porque permite crear sistemas más eficientes, escalables y mantenibles en múltiples industrias.