



# Σavante

T7E1 -

Principales  
características  
de la POO

Diego Ramos  
Fernández

## Índice

<b>1. Características principales de la programación orientada a objetos:</b>	<b>2</b>
Abstracción:	2
Encapsulamiento:	2
Herencia:	2
Polimorfismo:	2
Clase y Objeto:	3
Clase:	3
Objeto:	3
Ventajas e inconvenientes:	3
Ventajas:	3
Inconvenientes:	3
Relaciones entre clases:	3

# 1. Características principales de la programación orientada a objetos:

La programación orientada a objetos (POO) es un estilo de programación que busca imitar cómo funciona el mundo real, usando objetos que tienen propiedades y pueden hacer acciones. Es un enfoque diferente al de la programación estructurada.

## Abstracción:

La abstracción consiste en identificar sólo las características importantes de un objeto y ocultar los detalles irrelevantes. Por ejemplo, si tu modelo es una persona, sólo incluyes nombre, edad y dirección, no incluyes información innecesaria. Cada objeto es una especie de agente independiente dentro del programa.

## Encapsulamiento:

El encapsulamiento es el principio de proteger los datos que tiene un objeto, haciendo que sólo sean accesibles a través de los métodos de la clase. Esto ayuda a mantener la seguridad y el orden del código. Por ejemplo, si tienes una clase Perro, sus atributos como nombre o raza se manejan desde dentro de la clase, y sólo se pueden cambiar usando funciones preparadas para eso.

## Herencia:

La herencia permite que una clase “hija” reciba todas las características de una clase “padre”. Por ejemplo, si tienes una clase Animal con atributos como “nombre” y “edad”, la clase Perro puede heredar de Animal, y así no tienes que escribir todo de nuevo.

## Polimorfismo:

El polimorfismo es la capacidad de que un mismo método o acción funcione de diferentes maneras según la clase que lo implemente. Imagina el método hacerSonido(), que puede hacer “guau” si el objeto es Perro, o “miau” si es Gato. El polimorfismo permite crear programas más flexibles y reutilizables.

## Clase y Objeto:

### Clase:

- Es un modelo, una plantilla que define los atributos y los métodos comunes a todos los objetos que pertenezcan a esa clase.

### Objeto:

- Es una instancia concreta de la clase, con valores propios para sus atributos (por ejemplo, la clase Perro y el objeto miPerro con nombre "Rocky").

## Ventajas e inconvenientes:

### Ventajas:

- Reutilización del código
- Más organización y mantenimiento sencillo
- Facilita construir programas complejos
- Protección de información (encapsulamiento)

### Inconvenientes:

- Mayor dificultad de aprendizaje
- Más análisis previo y diseño antes de programar
- Puede ser más lento en casos concretos por el uso del polimorfismo

## Relaciones entre clases:

Las clases pueden tener relaciones:

- Asociación: Una clase tiene objetos de otras.
- Dependencia: Una clase usa temporalmente otra.
- Generalización (herencia): Una clase hereda de otra.

Comprender estas relaciones es esencial para estructurar bien los proyectos.