



PROGRAMACIÓN ORIENTADO A OBJETOS



La programación orientada a objetos (Object Oriented Programming, OOP) es un modelo de programación informática que organiza el diseño de software en torno a datos u objetos, en lugar de funciones y lógica. Un objeto se puede definir como un campo de datos que tiene atributos y comportamiento únicos.

La programación orientada a objetos se centra en los objetos que los desarrolladores quieren manipular en lugar de enfocarse en la lógica necesaria para manipularlos. Este enfoque de programación es adecuado para programas que son grandes, complejos y se actualizan o mantienen activamente.

La organización de un programa orientado a objetos también hace que el método sea beneficioso para el desarrollo colaborativo, donde los proyectos se dividen en grupos.

Los beneficios adicionales de la programación orientada a objetos incluyen la reutilización, la escalabilidad y la eficiencia del código. Incluso cuando se



utilizan microservicios, los desarrolladores deben seguir aplicando los principios de la programación orientada a objetos.

El primer paso en OOP es recopilar todos los objetos que un programador desea manipular e identificar cómo se relacionan entre sí, un ejercicio que a menudo se conoce como modelado de datos.

Los ejemplos de un objeto pueden variar desde entidades físicas, como un ser humano que se describe por propiedades como nombre y dirección, hasta pequeños programas informáticos, como widgets.

Una vez que se conoce un objeto, se etiqueta con una clase de objetos que define el tipo de datos que contiene y cualquier secuencia lógica que pueda manipularlo. Cada secuencia lógica distinta se conoce como método. Los objetos pueden comunicarse con interfaces bien definidas llamadas mensaje

Principios de OOP

La programación orientada a objetos se basa en los siguientes principios:

- **Encapsulación.** La implementación y el estado de cada objeto se mantienen de forma privada dentro de un límite definido o clase. Otros objetos no tienen acceso a esta clase o la autoridad para realizar cambios, pero pueden llamar a una lista de funciones o métodos públicos. Esta característica de ocultación de datos proporciona una mayor seguridad al programa y evita la corrupción de datos no intencionada.
- **Abstracción.** Los objetos solo revelan mecanismos internos que son relevantes para el uso de otros objetos, ocultando cualquier código de implementación innecesario. Este concepto ayuda a los desarrolladores a realizar cambios y adiciones más fácilmente a lo largo del tiempo.
- **Herencia.** Se pueden asignar relaciones y subclases entre objetos, lo que permite a los desarrolladores reutilizar una lógica común sin dejar de mantener una jerarquía única. Esta propiedad de OOP obliga a un análisis de datos más completo, reduce el tiempo de desarrollo y asegura un mayor nivel de precisión.
- **Polimorfismo.** Los objetos pueden adoptar más de una forma según el contexto. El programa determinará qué significado o uso es necesario para cada ejecución de ese objeto, reduciendo la necesidad de duplicar código.



Lenguajes de programación orientados a objetos

Si bien Simula se acredita como el primer lenguaje de programación orientado a objetos, los lenguajes de programación orientada a objetos más populares son:

- Java
- JavaScript
- Python
- C++
- Visual Basic .NET
- Ruby
- Scala
- PHP

