

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías Emergentes

Paradigmas de la programación

Práctica 2

“Paradigma Orientado a Objetos”

ALUMNO: Cesar Alejandro Velazquez Mercado

MATRÍCULA: 372329

GRUPO: 941

PROFESOR: Carlos Gallegos

28 de mayo del 2024

## Introducción al Paradigma Orientado a Objetos

El paradigma orientado a objetos (POO) es un enfoque de programación que utiliza "objetos" para representar datos y métodos asociados. Esta metodología busca modelar el mundo real de manera más intuitiva y estructurada, facilitando el desarrollo, mantenimiento y escalabilidad de los programas.

### ¿Qué es el Paradigma Orientado a Objetos?

El paradigma orientado a objetos es una forma de programación que organiza el software en unidades llamadas objetos. Cada objeto contiene datos y métodos que operan sobre esos datos, encapsulando la funcionalidad dentro de estructuras claramente definidas.

- **Modularidad:** Los objetos permiten descomponer un sistema complejo en partes más manejables, facilitando el desarrollo y mantenimiento.
- **Reutilización:** La herencia y la creación de clases permiten reutilizar código existente, reduciendo la redundancia.
- **Flexibilidad y Escalabilidad:** Los sistemas orientados a objetos pueden adaptarse y crecer más fácilmente gracias a su estructura modular y reutilizable.
- **Mantenimiento:** El encapsulamiento y la abstracción simplifican la localización y corrección de errores, así como la actualización de sistemas sin afectar otras partes.

### Conceptos Clave del Paradigma Orientado a Objetos

1. **Objeto:** Un objeto es una instancia de una clase que encapsula datos y comportamientos. Los datos se representan mediante atributos y los comportamientos mediante métodos.
2. **Clase:** Una clase es una plantilla o un modelo que define los atributos y métodos que sus objetos asociados tendrán. Las clases permiten crear múltiples objetos con las mismas características.
3. **Encapsulamiento:** Es la propiedad que permite agrupar datos y métodos que operan sobre esos datos dentro de una sola unidad llamada objeto. También incluye la idea de ocultar la implementación interna y exponer sólo lo necesario a través de una interfaz pública.
4. **Herencia:** Es un mecanismo que permite crear una nueva clase a partir de una clase existente. La nueva clase hereda los atributos y métodos de la clase original, permitiendo la reutilización del código y la extensión de funcionalidades.
5. **Polimorfismo:** Es la capacidad de diferentes clases de ser tratadas como instancias de una misma clase a través de una interfaz común. El

polimorfismo permite a un solo método funcionar de manera diferente en diferentes contextos.

6. **Abstracción:** Es el proceso de simplificar un sistema complejo mediante la modelación de clases que representan conceptos clave, ignorando los detalles irrelevantes. La abstracción facilita el enfoque en los aspectos importantes de un problema.

## Referencias

López, M. (2023, 19 diciembre). Aplicaciones de la programación orientada a objetos - IMMUNE. *Immune Technology Institute*.

<https://immune.institute/blog/programacion-orientada-a-objetos/#:~:text=La%20POO%20es%20un%20paradigma,y%20el%20mantenimiento%20del%20software.>

Alarcón, J. M. (s. f.). *Los conceptos fundamentales sobre Programación Orientada Objetos explicados de manera simple* - campusMVP.es. campusMVP.es.

<https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>

*Qué es la Programación Orientada a Objetos.* (s. f.). Intelequia.

<https://intelequia.com/es/blog/post/qu%C3%A9-es-la-programaci%C3%B3n-orientada-a-objetos>