

M9 Desarrollo de aplicaciones orientadas a objetos FIN A

Actividad 1

|  |  |
| --- | --- |
| **Tutor:** | **Karla Ramos García** |
| **Estudiante:** | **José Ramón Ibáñez Posadas** |
| **Matricula:** | **BNL098377** |

|  |  |
| --- | --- |
| Monterrey, Nuevo León | sábado, 13 de septiembre de 2025 |

Introducción

La Programación Orientada a Objetos (POO) es un modelo de programación que utiliza el concepto de objetos para representar elementos de la vida real. A diferencia de otros estilos de programación que se centran en algoritmos, la POO usa objetos como bloques lógicos de construcción.

Un programa diseñado con POO es un conjunto de objetos vinculados o relacionados entre sí. Para que una programación sea considerada orientada a objetos, debe cumplir con tres principios fundamentales:

Utilizar objetos, no algoritmos, como los bloques lógicos de construcción. Cada objeto debe ser una instancia de una clase. Las clases se relacionan entre sí por medio de la herencia.

Una de las principales ventajas de la POO es que simplifica los datos complejos al hacer que los objetos sean unidades autocontenidas.

También agiliza el desarrollo de software y la construcción de prototipos. Aunque tiene algunas desventajas, como una ejecución más lenta en algunos programas, la POO ofrece herramientas y bases para un desarrollo de software más organizado y fácil de manejar.

Desarrollo

**Mapa mental**

**A diagram of a diagram

AI-generated content may be incorrect.**

<https://lucid.app/lucidspark/7aa70502-bb17-4000-8b1a-76c9fb1b04cc/edit?viewport_loc=-2165%2C-971%2C5243%2C3085%2C0_0&invitationId=inv_45853102-d548-4433-8e44-e92a1cc93b59>

**Cuadro sinóptico**

|  |  |  |
| --- | --- | --- |
| Nombre de los métodos de los objetos | Definición | Ejemplo de la vida real |
| Abstracción | Permite usar solo las características de un objeto que son de interés, descartando las no utilizadas. | En un sistema de empleados, solo te interesarían datos como nombre, edad y dirección, no el color de cabello o la filiación política. |
| Encapsulamiento | Combina datos (atributos) y métodos (comportamientos) en un solo elemento (un objeto), ocultando los detalles de implementación. | Al usar un televisor, no necesitas saber cómo funciona el sintonizador de canales o el sistema eléctrico. Todos esos mecanismos están ocultos, o encapsulados, para el usuario. |
| Polimorfismo | La capacidad de un objeto para elegir la implementación de un método que mejor se adapte a sus características, permitiendo que un mismo mensaje sea interpretado de forma diferente por objetos de distintas clases. | El mensaje "obtener área" se interpreta de forma distinta si se envía a un objeto de la clase "triángulos" o a uno de la clase "rectángulos", ya que cada uno usa una fórmula diferente. |
| Persistencia | Se refiere a la permanencia de un objeto, o al tiempo que dura en la memoria para ser utilizado. Los objetos almacenados de forma permanente se denominan persistentes. | En un sistema de cuentas, la persistencia es el tiempo en que se usa una operación para obtener un resultado. Una vez que se obtiene, el espacio en memoria se libera. |

Conclusión

La POO te ayuda a pensar en problemas complejos de una forma más intuitiva, similar a como los resolverías en la vida real. La relación entre clases, objetos y mensajes es fundamental para crear programas organizados y eficientes.

Una clase es como un molde, el objeto es la pieza creada, y el mensaje es la instrucción que le das a esa pieza para que actúe.

El polimorfismo, la abstracción y el encapsulamiento son conceptos que se pueden ver a diario y que simplifican la complejidad de los sistemas.

Por ejemplo, al usar un cajero automático, solo interactúas con la pantalla (abstracción) y sus funciones de retirar dinero (polimorfismo), sin necesidad de entender su funcionamiento interno (encapsulamiento).

**Bibliografía**

**A screenshot of a computer

AI-generated content may be incorrect.**