Programación orientada a objetos POO

La **POO** es un paradigma de la programación es el enfoque como vamos a estructurar el código, según la necesidad que tendremos para dar una solución al proyecto. La POO es un enfoque.

**POO** (Programación orientada a objetos): Paradigma de programación que organiza el código en objetos, cada una con su propios datos y comportamientos.

Se centraliza en conceptos como la abstracción, herencia, encapsulamiento y polimorfismo para facilitar el desarrollo del software modular, reutilizable y fácil de mantener.

En **POO**, los objetos interactúan entre si mediante mensajes, lo que permite una representación mas cercana de los conceptos del mundo real y una mejor gestión de la complejidad en los programas.

Bases de la **POO**:

* Clases (Plantillas de los objetos): Piedra angular de la **POO** actuando como plantillas para la creación de objetos. Definen las propiedades (Atributos) y comportamientos (Métodos) que caracterizan a los objetos de un tipo especifico. Al encapsular datos y funcionalidades relacionadas, permite la reutilización del código, la organización modular y la abstracción.
* Métodos: características o datos que describen el estado de un objeto en un contexto determinado.
* Métodos: Son bloques de código que encapsulan una seria de acciones o comportamientos específicos asociados a un objeto. Estos permiten realizar operaciones entre los datos de un objeto, modificar su estado interno y responder a eventos dentro de un programa. Los métodos pueden acceder a los atributos de un objeto.
* Objetos: Son las instancias especificas de una clase que encapsulan datos y comportamientos relacionados. Los objetos permiten modelar entidades del mundo real de manera modular y reutilizable, facilitando la organización y el diseño de sistemas de software eficientes y mantenibles.

Características de la **POO**:

* **Abstracción**: Permite representar entidades del mundo real como objetos con características y comportamientos relevantes para el problema que se está resolviendo. Esto simplifica la complejidad del sistema al enfocarse en aspectos esenciales y proporcionar una forma de modelar conceptos del mundo real en el código.
* **Encapsulamiento**: Los detalles internos de un objeto deben estar ocultos fuera de su definición y solo deben ser accesibles atraves de una interfaz claramente definida. Esto promueve la seguridad y la integridad de los datos al prevenir accesos no autorizados y facilita el mantenimiento del código.
* **Herencia**: En programación (POO) nos permitirá reutilizar o transmitir atributos de un objeto.

Permite que una clase (subclase) herede los atributos y métodos de otra clase (superclase), lo que promueve la reutilización del código y la organización jerárquica de las clases. Permite que una clase herede propiedades y comportamientos de otra clase

Esto permite crear una clase basada en una existente, y es fundamental para reutilización de código y la creación de jerarquías de las clases.

* Clase Base (**Superclase**): La clase que de la cual se heredan los atributos y métodos. Se le conocen como superclase o clase padre.
* Clase Derivada (**Subclase**): La clase que hereda de la clase base. Se le conoce como subclase o clase hija.
* **Polimorfismo**

Poli(muchas) morfismo(formas)

Se refiere a la capacidad que tienen los objetos de diferente clase para responder a la misma acción de manera distinta.

En términos simples, una misma operación o método puede comportarse de diferentes maneras dependiendo del objeto que lo ejecute.

**Instalaciones necesarias:**