



Módulo 2

Sesión N° 3



ACTIVIDAD:



Construcción de un Sistema de Gestión de Vehículos Utilizando POO en Python

- **Objetivo:** Demostrar la comprensión de los conceptos teóricos y prácticos de la POO en Python, aplicando los fundamentos básicos y los principios de diseño en la implementación de un sistema sencillo.



Instrucciones:

1. Diseño del Sistema:

- **Funcionalidad:**

Desarrollar un programa en Python que simule la gestión de vehículos. El sistema debe permitir:

- Registrar vehículos con atributos básicos: marca, modelo y año.
- Definir una clase base Vehiculo y, a partir de ella, crear clases derivadas (por ejemplo, Auto y Moto).

- **Métodos Obligatorios:**

- La clase base debe incluir un método `acelerar()` que imprima un mensaje genérico.
- Las clases hijas deben sobrescribir (override) el método `acelerar()` para mostrar un comportamiento específico (por ejemplo, un mensaje distintivo que indique el tipo de vehículo).
- Cada clase derivada debe incluir al menos un método propio que no exista en la clase base (por ejemplo, `abrir_maletero()` para Auto y `hacer_caballito()` para Moto).



2. Aplicación de Principios POO:

- Utilizar encapsulación para definir y proteger los atributos (usando, por ejemplo, convenciones como `self.__atributo` para variables internas).
- Emplear herencia y polimorfismo para demostrar la reutilización de código y la respuesta diferenciada a métodos comunes.





- Integrar al menos uno de los principios SOLID (por ejemplo, el Principio de Responsabilidad Única, SRP) en la organización de las clases, de forma que cada clase tenga una sola responsabilidad.

3. Documentación y Presentación:

- El código debe estar bien comentado, explicando la funcionalidad de cada clase y método.

4. Actividades Complementarias (Opcionales):

- Crear un diagrama de clases que represente la relación entre Vehículo, Auto y Moto.

5. Entrega:

- Formato de ejecución: en parejas.
- Formato de entrega: comprimido (.zip, .rar)
- Tempo estimado: 1 hora 30 minutos.

