Módulo 2

Sesión N° 4

ACTIVIDAD:

Introducción a la Programación Orientada a Objeto

Objetivo: Aprender los conceptos básicos de programación orientada a objetos en Python creando clases con atributos, métodos, herencia y decoradores, aplicándolos en un modelo basado en un caso real.

Contexto

Esta actividad tiene como propósito aplicar los conceptos básicos de POO mediante el desarrollo de un sistema simple basado en clases. Se definirá una jerarquía de clases con atributos, métodos, visibilidad, encapsulamiento y decoradores para getters y setters. El código resultante simulará un sistema de gestión de usuarios y administradores, utilizando los fundamentos clave de la orientación a objetos.

Requerimientos:

1. Exploración Inicial:
   - Escribe como comentario al inicio del script:
     - ¿Qué es la Programación Orientada a Objetos?
     - ¿Cuáles son sus cuatro principios fundamentales?
     - ¿Qué ventajas aporta la POO frente a la programación estructurada?
2. Definición de Clases:
   - Crea una clase Usuario con:
     - Atributos: nombre, correo, _contraseña (privado).
     - Métodos: saludar(), mostrar_info().
     - Constructor que inicialice todos los atributos.
     - Getters y setters para la contraseña usando decoradores.
3. Encapsulamiento y Visibilidad:
   - Asegúrate de que el atributo contraseña sea privado (__contraseña) y se acceda solo mediante métodos get_contraseña() y set_contraseña() con decoradores @property y @<nombre>.setter.
4. Herencia y Polimorfismo:
   - Crea una clase Administrador que herede de Usuario.
     - Agrega un atributo adicional permisos (lista de strings).
     - Redefine el método mostrar_info() para mostrar también los permisos.
     - Añade un método agregar_permiso().
5. Instanciación y Prueba:
   - Crea al menos dos objetos: uno de Usuario y otro de Administrador.
   - Llama a sus métodos y muestra sus comportamientos diferenciados.
   - Realiza pruebas para leer y modificar la contraseña usando getters y setters.

6. Entrega:
   - Formato de entrega: comprimido (.zip, .rar) con Script, capturas de pantalla y documento explicativo.
   - Tiempo estimado de desarrollo: 75 minutos.
   - Formato de ejecución: grupal.
   - Extra: incorporar txt o doc con breve explicación de los principios aplicados (encapsulamiento, herencia,etc.), qué hace cada clase y qué rol tienen los decoradores en el control de acceso a atributos.