



# Módulo 2

Sesión N° 5



## ACTIVIDAD:



## Herencia y Excepciones en Python

Objetivo: Aplicar herencia y manejo de excepciones en Python mediante la creación de clases relacionadas, comprendiendo cómo funcionan las subclases, el polimorfismo y las excepciones personalizadas en situaciones prácticas.



## Contexto

Durante esta actividad, el estudiante desarrollará un sistema de clases representando una jerarquía de usuarios en una plataforma. Además de la implementación de herencia, polimorfismo y sobrecarga, se integrará el manejo de excepciones en operaciones como login, permisos o eliminación de usuarios.





## Requerimientos:

---

1. Exploración inicial (como comentarios en el script):
  - Explica brevemente qué es la herencia y por qué es útil.
  - Describe qué es una excepción y cómo ayuda a manejar errores en los programas.
  - Incluye una lista de los dunder methods más comunes que conoces.
2. Definición de clases y jerarquía:
  - Crea una clase Usuario con los atributos nombre, correo, activo=True.
  - Crea un método presentarse() que imprima el nombre del usuario.
  - Crea una subclase Administrador que herede de Usuario, añadiendo:
    - Atributo adicional: permisos (lista de strings).
    - Método sobrescrito presentarse() que también muestre sus permisos.
    - Método agregar\_permiso().
  - Crea una clase SuperUsuario que herede de Administrador y otra clase llamada Soporte, simulando herencia múltiple.
    - Incluye un ejemplo que imprima \_\_mro\_\_.
3. Uso de polimorfismo y sobrecarga:
  - Crea una función mostrar\_usuario(usuario) que funcione con cualquier tipo de objeto (Usuario, Administrador, etc.).
  - Implementa una función con parámetros por defecto o \*args para simular sobrecarga.
4. Implementación de excepciones:
  - Crea una excepción personalizada PermisoInsuficienteError.
  - En el método agregar\_permiso, lanza esta excepción si el usuario no tiene rol "administrador".
  - Implementa bloques try-except para capturar ValueError y tu excepción personalizada.
  - Usa finally para imprimir un mensaje final.
5. Entrega:
  - Formato de entrega: comprimido (.zip, .rar) con Script, capturas de pantalla y documento explicativo.
  - Tiempo estimado de desarrollo: 90 minutos.
  - Formato de ejecución: individual.

