

Ejercicio repaso Python:

1. Encapsulación:

- Define una clase llamada **Empleado** con atributos encapsulados como `id`, `nombre`, correo y `salario`. Añade métodos para acceder y modificar estos atributos de manera controlada.

2. Herencia:

- Crea una clase derivada llamada **Programador** que herede de **Empleado**. Agrega un nuevo atributo llamado `lenguaje` que represente el lenguaje de programación preferido del programador.

3. Métodos Mágicos:

- Implementa los métodos mágicos `__str__` y `__repr__` en ambas clases (**Empleado** y **Programador**) para que al imprimir una instancia, se muestre de manera legible.

4. Acceso a Internet (requests):

- Utiliza la biblioteca **requests** para realizar una solicitud GET a `"https://jsonplaceholder.typicode.com/users"` (JSONPlaceholder). Almacena la respuesta en un diccionario y muestra algunos datos relevantes.

5. Manejo de Ficheros:

- Crea un archivo llamado `"empleados.txt"`. Guarda información ficticia sobre empleados en este archivo, utilizando el formato que prefieras (puede ser JSON, CSV, etc.).

6. Excepciones Propias:

- Define una excepción personalizada llamada **SalarioInvalidoError**. Lanza esta excepción si se intenta asignar un salario negativo a un empleado.

7. Gestión de Errores:

- Al leer el archivo `"empleados.txt"`, maneja las posibles excepciones que puedan ocurrir, como la falta de archivo o un formato incorrecto.

8. Uso de Listas/Diccionarios/Tuplas:

- Utiliza una lista o diccionario para almacenar instancias de empleados

y programadores. Puedes crear instancias ficticias o cargarlas desde el archivo "empleados.txt".

Notas:

- Modularización
- Código legible
- Uso de docstrings