

Práctica 5 - Singleton

Ochoa Lucero Angel Martin

Ejemplo de Patrón de Diseño Singleton

Sistema de registro de logs en Python

Este ejemplo muestra cómo implementar el patrón **Singleton** en Python.

La idea es garantizar que solo exista **una instancia** del objeto **logger** en toda la aplicación.

#practica 5. Singleton #ejemplo de patron de diseño Singleton - sistema de registro de logs.

```
class logger: #atributo para guardar la unica instancia _instancia = None
```

```
#metodo __new__ controla la creacion del objeto antes de init
```

```
#y se asegura que solo exista una unica instancia de logger.
```

```
def __new__(cls, *args, **kwargs):
```

```
    if cls._instancia is None:
```

```
        cls._instancia = super().__new__(cls)
```

```
        cls._instancia.archivo = open("app.log", "a")
```

```
    return cls._instancia #devuelve siempre la misma instancia.
```

```
def registro(self, mensaje):
```

```
    self.archivo.write(mensaje + "\n")
```

```
    self.archivo.flush() #fuerza a que el archivo se guarde en disco.
```

```
#creamos la unica instancia Singleton registro1 = logger() registro2 = logger() #devuelve la misma instancia sin crear una nueva
```

```
registro1.registro("inicio de sesion en la aplicacion") registro2.registro("el usuario se autentifico")
```

```
print(registro1 is registro2) # True o False
```