

Practica 5. Singleton patrones de diseño

Castillo Dominguez Sonia Vanessa

date: "19/09/25"

codigo

```
class logger:
    #atributo de la clase para guardar la unica instancia
    _instancia = None
#metodo __new__: controla la creacion del objeto antes de la int. se asegura que solo existe 1
    def __new__(cls, *args, **kwargs):
        if cls._instancia is None:
            cls._instancia = super().__new__(cls)
            #Abrimos una archivo de logs en modo "append"
            cls._instancia.archivo = open ("app.log", "a")
            return cls._instancia #Devolver siempre a la misma instancia

        def registro(self, mensaje):
            self.archivo.write(mensaje + "\n")
            self.archivo.flush()# Forza al archivo para guardar el archivo

registro1 = logger() #Creamos la unica instancia SINGLETON
registro2 = logger() #Devuelve la misma instancia , sin crear una nueva

registro1.registro("Inicia de sesion en la aplicacion")
registro2.registro("El usuario se autentico")

print(registro1 is registro2)#True o False
#Si me registro true: Es el mismo Objeto
```

True

1. ¿Qué pasaría si eliminamos la verificación `if cls.instancia is None` en el método `__new__`?

cada vez que se cree un objeto de la clase `logger` se generará una nueva instancia distinta, rompiendo el principio del patrón Singleton.

2. En la línea `print(registro1 is registro2)`. ¿Qué significa que devuelva `True` en el contexto del Singleton?

Que ambas variables (`registro1` y `registro2`) apuntan exactamente al mismo objeto en memoria. Esto confirma que el patrón Singleton está funcionando correctamente

3. ¿Crees que siempre es buena idea usar Singleton para todo lo que es global?

No, no siempre es recomendable. El patrón Singleton es útil para recursos que deben tener una única instancia. Ejemplo: Una clase `Usuario` en un sistema. Cada usuario que se conecta debería tener su propia instancia con datos distintos. Si fuera Singleton, todos los usuarios compartirían la misma instancia, lo cual es incorrecto y peligroso.