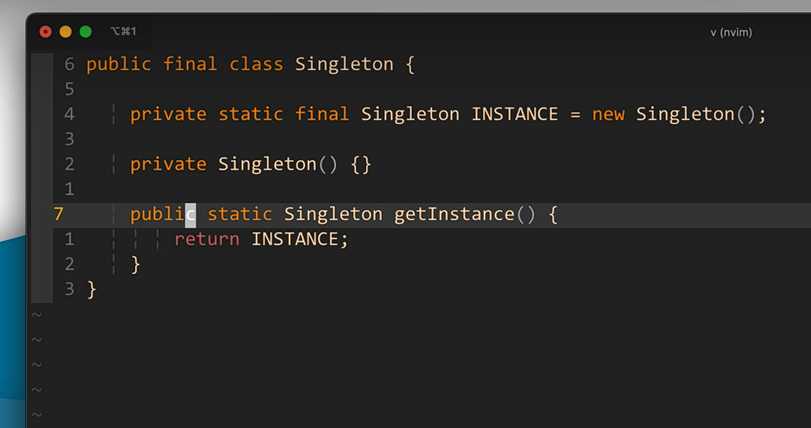
**Patrones de diseño**

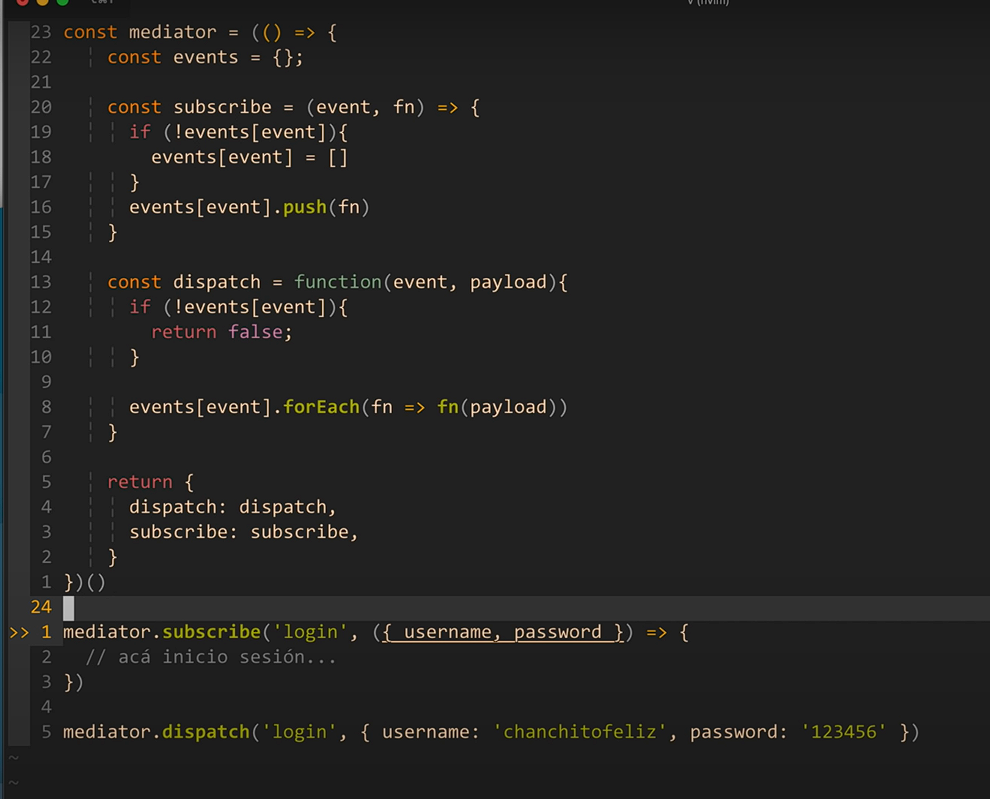
**SINGELTON**: restringe la instanciación de una clase a solo una única instancia esto es útil cuando solo necesitas solo una instancia de una clase para coordinar acciones entre distintos sistemas



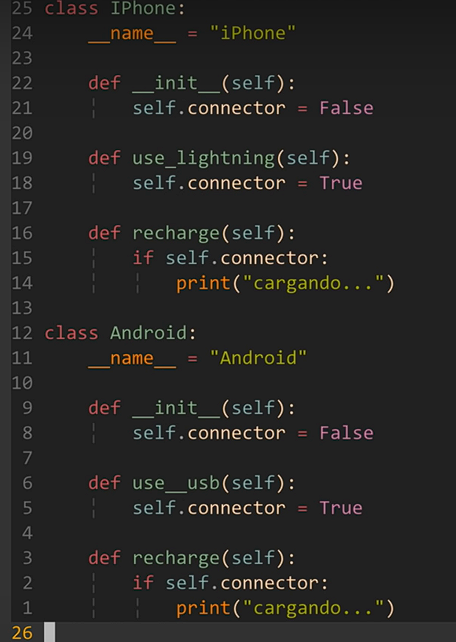
**DECORATOR**: es un patrón que permite agregar funcionalidades a un objeto sin alterar el comportamiento de otras instancias de objectos de la misma clase soluciona el problema de agregar o quitar responsabilidades a un objeto en el RUN TIMA y es una alternativa flexible a las subclases

Texto

Descripción generada automáticamente

**MEDIATOR**: define un objeto el cual encapsula como otros objectos van a interactuar este patrón se considera un patrón de comportamiento de que puede alterar como se ejecuta la aplicación un patrón mediador

**APDAPTER**: permite que un objeto que no está diseñado para funcionar con una interfaz sea modificado para que este si pueda funcionar con este



**MVC(Model View Cotroller):** se utiliza comúnmente se utiliza para desarrollar interfaces de usuario dividiendo la lógica en tres elementos interconectados el modelo el cual contiene la lógica que contiene de la aplicación, se encargara de llamar a los distintos datos y contiene las reglas de negocio la vista es la representación visual de los datos que se recolectan esto es lo que vera el usuario y el controlador se encarga de recibir los datos transformarlos en algo útil para el modelo recibe los datos del modelo y lo construye la respuesta para el usuario.

