

SINGLETON

SINGLETON

Patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia.

PROBLEMA

Garantizar que una clase tenga una única instancia.

Controlar el acceso a algún recurso compartido, por ejemplo, una base de datos o un archivo

Proporcionar un punto de acceso global a dicha instancia

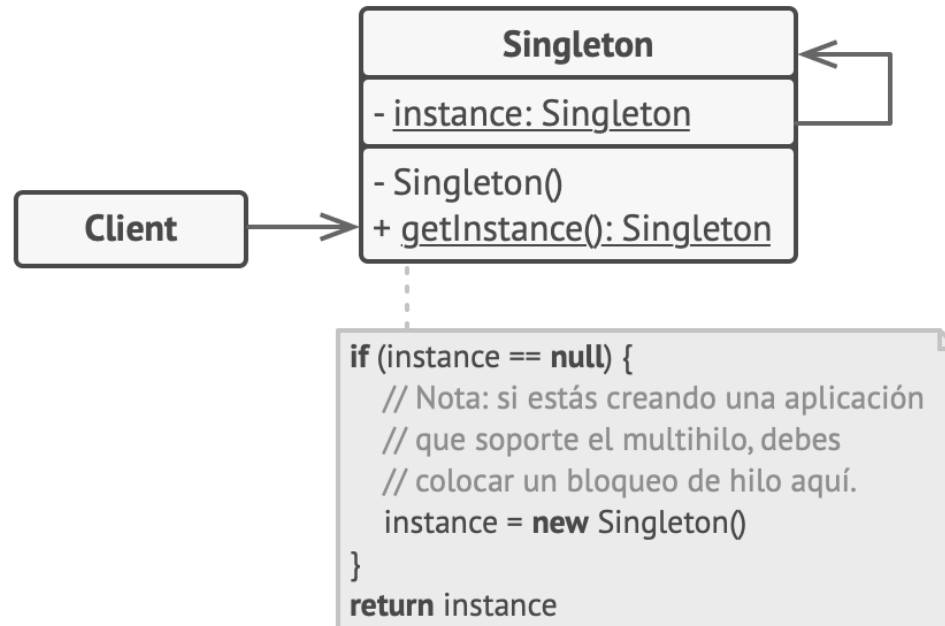
SOLUCIÓN

Hacer privado el constructor por defecto para evitar que otros objetos utilicen el operador `new` con la clase Singleton

SOLUCIÓN

Crear un método de creación estático que actúe como constructor. Tras bambalinas, este método invoca al constructor privado para crear un objeto y lo guarda en un campo estático. Las siguientes llamadas a este método devuelven el objeto almacenado en caché.

ESTRUCTURA



PROS

- Puedes tener la certeza de que una clase tiene una única instancia.
- Obtienes un punto de acceso global a dicha instancia.
- El objeto Singleton solo se inicializa cuando se requiere por primera vez.

CONTRAS

- Vulnera el Principio de responsabilidad única. El patrón resuelve dos problemas al mismo tiempo.
- Puede enmascarar un mal diseño, por ejemplo, cuando los componentes del programa saben demasiado los unos sobre los otros.
- Requiere de un tratamiento especial en un entorno con múltiples hilos de ejecución, para que varios hilos no creen un objeto Singleton varias veces.

