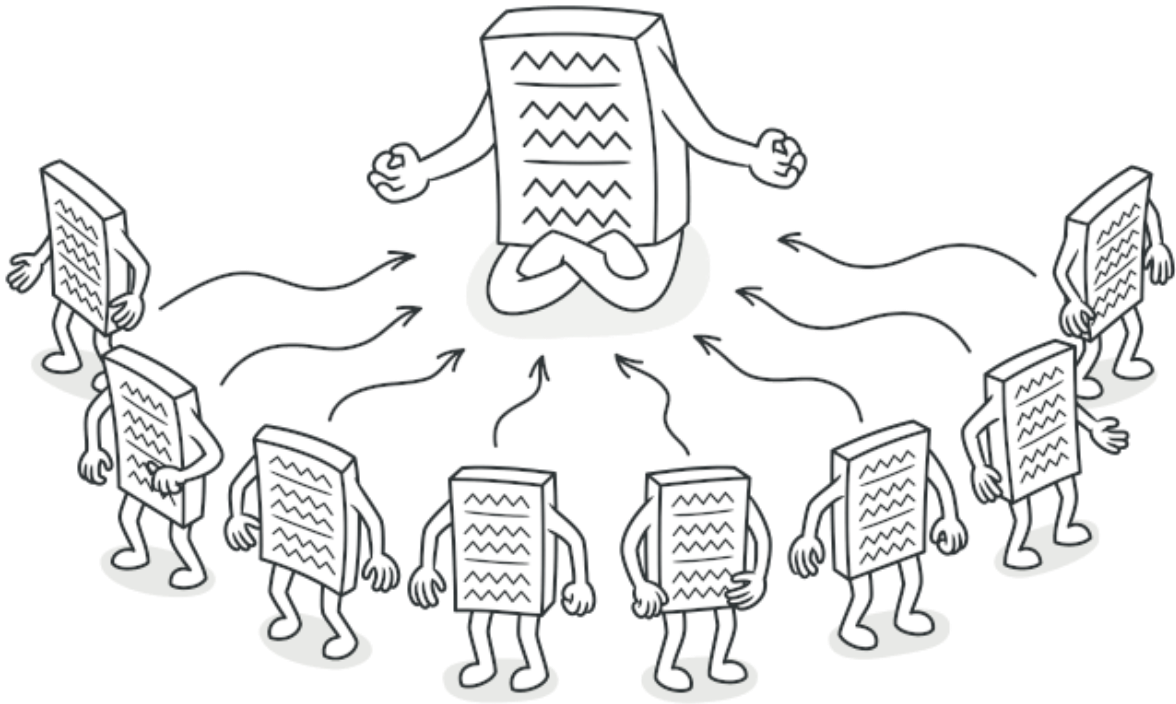


PATRON DE DISEÑO SINGLETON



El patrón Singleton es un patrón de diseño que se utiliza cuando queremos que una clase tenga una única instancia en todo el programa y evitar código repetitivo en instanciar la clase cuando se ocupe. Por ejemplo, si solo debe haber una conexión a la base de datos, o un solo log para registrar eventos, usamos Singleton.

Este patrón permite evitar que se creen varios objetos de esa clase, lo que hacemos es que el constructor sea privado, y luego creamos un método público llamado `getInstance()` que devuelve siempre la misma instancia de esa clase.

Esto nos asegura que todos los que llamen a esa clase obtendrán exactamente el mismo objeto. Es útil para ahorrar memoria, controlar el acceso a recursos compartidos y mantener consistencia en la aplicación.