

## Patrón Singleton

Se utiliza para asegurar que solo exista una instancia de una clase durante la ejecución del programa. En este proyecto se utilizó en la clase Calculadora, ya que solo necesitamos una calculadora para realizar todas las operaciones.

Una ventaja del patrón Singleton es que evita crear múltiples objetos innecesarios y permite tener un punto central de acceso a la clase. También facilita el control de la instancia, ya que el constructor es privado y solo se puede acceder mediante el método `getInstance()`.

Sin embargo, una desventaja es que su funcionamiento es parecido a una variable global. Esto puede hacer que el código sea más difícil de probar o modificar en programas grandes, ya que muchas partes del sistema pueden depender directamente de esa única instancia.

En este programa considero que el uso de Singleton sí es adecuado, porque solo se necesita una calculadora para convertir y evaluar expresiones. No se requiere tener varias calculadoras funcionando al mismo tiempo, por lo que el patrón cumple su propósito sin complicar el diseño.