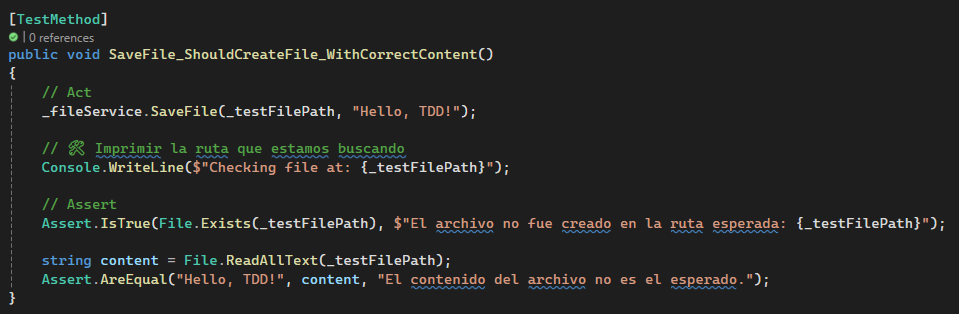
**Test-Driven Development (TDD)**

Test-Driven Development (TDD) es una metodología de desarrollo de software en la que las pruebas unitarias se escriben antes del código de implementación. Este enfoque permite diseñar software más robusto, modular y fácil de mantener, asegurando que el código cumpla con los requisitos desde el inicio.

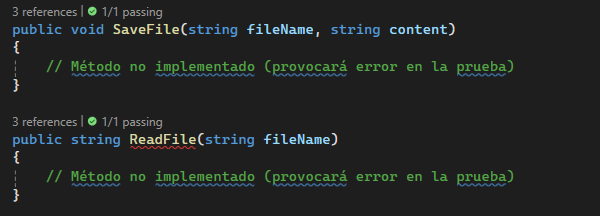
* **Fases:**
* **Red (Fallar primero)**: Se escribe una prueba automatizada que define una nueva funcionalidad, pero al ejecutarla, falla porque la funcionalidad aún no está implementada.
* **Green (Hacerla pasar)**: Se implementa el código mínimo necesario para que la prueba pase exitosamente.
* **Refactor (Mejorar el código)**: Se optimiza el código asegurándose de que sigue pasando las pruebas, sin modificar su comportamiento.
* **Beneficios de Clean Architecture:**
* **Mayor calidad del código**: Reduce la cantidad de errores y defectos en producción.
* **Mejor diseño del software**: Fomenta la modularidad y el bajo acoplamiento.
* **Facilita el mantenimiento**: Las pruebas proporcionan documentación viva del código.
* **Acelera la detección de errores**: Se identifican fallos en las primeras etapas del desarrollo.
* **Mejora la confianza en el código**: Los cambios futuros pueden realizarse sin temor a romper funcionalidades existentes.
* **Ejemplo implementado en el Proyecto:**

En este proyecto, aplicamos TDD en la implementación de FileService, asegurando que su funcionalidad esté validada antes de escribir el código final.

### **Etapa 1: prueba fallida**

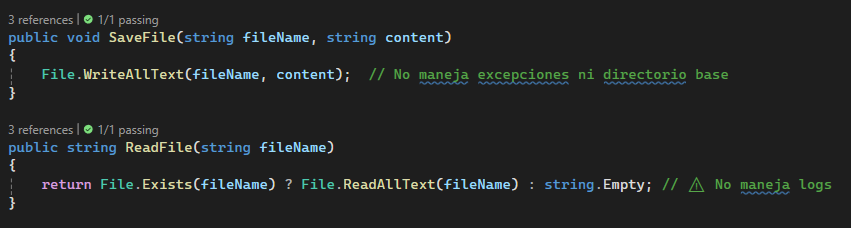


Falla porque SaveFile no está implementado:

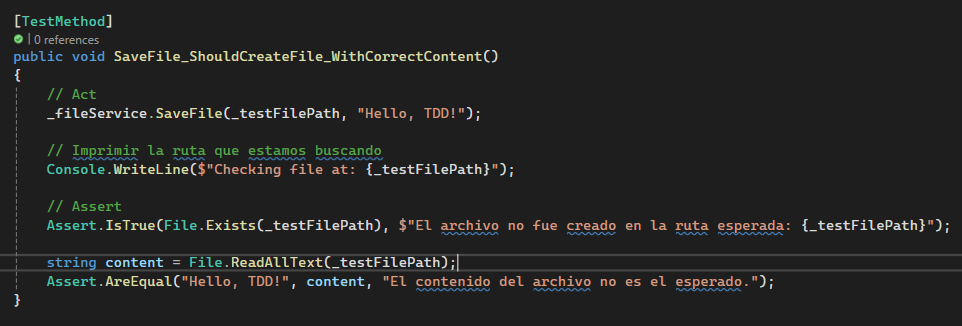


### **Etapa 2: implementación mínima**

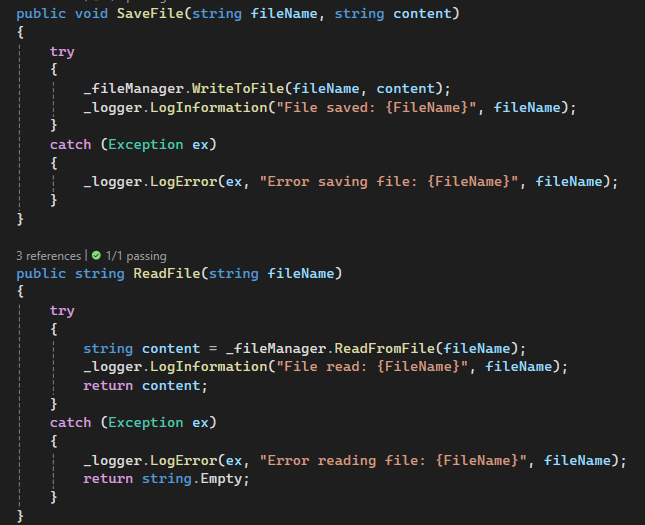


Pasa la prueba, pero sin validaciones ni logs:

### **Etapa 3: refactorización y optimización**



Ahora el código es más robusto y confiable, manteniendo la prueba aprobada



# **Conclusión**

El desarrollo guiado por pruebas (TDD) es una metodología poderosa que mejora la calidad del software, facilita su mantenimiento y reduce los errores en producción. Su integración con Clean Architecture y otros principios de diseño refuerza la creación de sistemas bien estructurados, flexibles y confiables.