

Explicación de las Pruebas Unitarias

Introducción

Las pruebas unitarias nos permiten asegurarnos de que cada parte del código funciona correctamente antes de integrarla en el sistema. Para nuestro proyecto, hemos realizado pruebas en tres patrones de diseño: **Factory Method, Facade y Observer**.

1. Prueba para el Patrón Factory Method

Objetivo: Verificar que la fábrica de tareas cree correctamente los objetos esperados.

- **Prueba 1:** Se crea una tarea de tipo "simple" y se verifica que la instancia no es nula y corresponde a TareaSimple.
- **Prueba 2:** Se intenta crear una tarea con un tipo inválido y se verifica que se lanza una excepción.

Importancia: Estas pruebas aseguran que el sistema no genere objetos erróneos y maneje correctamente los casos no previstos.

2. Prueba para el Patrón Facade

Objetivo: Comprobar que la clase ProyectoFacade permite la creación de proyectos y la validación de usuarios de manera correcta.

- **Prueba:** Se inicia un proyecto y se captura la salida para confirmar que tanto la validación del usuario como la creación del proyecto se ejecutan correctamente.

Importancia: Garantiza que los servicios internos trabajen juntos sin errores y que el acceso simplificado a la funcionalidad esté bien implementado.

3. Prueba para el Patrón Observer

Objetivo: Evaluar si los observadores (usuarios) reciben correctamente las notificaciones cuando una tarea cambia de estado.

- **Prueba:** Se agregan dos usuarios a una tarea y se verifica que ambos reciben una notificación cuando la tarea cambia de estado.

Importancia: Nos asegura que el sistema de notificaciones funcione correctamente, permitiendo una comunicación fluida entre usuarios.

Conclusión

Estas pruebas ayudan a detectar errores antes de que lleguen a producción, reduciendo problemas y mejorando la calidad del software.

