

TDD y BDD (incluye refactorización de código asistida por IA)

Temario

1. Fundamentos y Metodologías Ágiles
 - 1.1. Introducción a las metodologías ágiles
 - 1.2. Prácticas clave en XP (Extreme Programming)

2. Pruebas de Software
 - 2.1. Tipos de pruebas: caja negra y caja blanca
 - 2.2. Pruebas de funcionamiento y cualidades del software
 - 2.3. Imposibilidad de probar todo el software
 - 2.4. Impacto de los requerimientos y casos de uso en las pruebas

3. Principios de Diseño de Software
 - 3.1. Diseño simple y principio SOLID
 - 3.2. Principios de diseño: "Dile, no preguntes" y "Divide y vencerás"
 - 3.3. Calidad del software: ¿Por qué no se puede codificar sin errores?

4. Gestión y Ejecución de Pruebas
 - 4.1. Roles y responsabilidades en pruebas
 - 4.2. Métodos y métricas clave: MITs (Most Important Test), MIMs (Most Important Metrics)
 - 4.3. Análisis de datos y valores frontera en pruebas
 - 4.4. Técnicas de reducción de datos y análisis de rutas

5. Metodología TDD (Test Driven Development)
 - 5.1. Ciclo de desarrollo basado en pruebas
 - 5.2. Ventajas de TDD y herramientas: JUnit, Hamcrest, Mockito
 - 5.3. Técnicas avanzadas: pruebas paramétricas, mocks, fakes y stubs
 - 5.4. Cobertura de pruebas (Code Coverage) y herramientas como Eclemma

6. Refactorización de Código
 - 6.1. Principios y técnicas de refactorización
 - 6.2. Identificación y resolución de "malos olores" en el código
 - 6.3. Reutilización y simplificación de expresiones condicionales
 - 6.4. Herramientas de refactorización: Eclipse
 - 6.5. Refactorización de código con IA

7. Pruebas de Aceptación (ATDD/BDD)
 - 7.1. Definición de requisitos mediante historias de usuario
 - 7.2. Desarrollo de escenarios de prueba con un lenguaje accesible
 - 7.3. Herramientas BDD: Cucumber