

FractionalCTO.es

Logalty – Intro Unit Tests con IA

25 Septiembre 2025

¿Qué son los Unit Tests?

Pruebas **automatizadas** que validan **pequeñas unidades** de código de forma **aislada**.

Características de los Unit Tests

- Rápidos (del orden de ms) -> Se ejecutan muchas veces sin casi penalización
- Sencillos -> Fáciles de entender y mantener
- Aislados -> Validan únicamente la unidad sin dependencias
- Deterministas -> Si no hay cambios su resultado nunca debería cambiar

¿Qué aportan los Unit Tests?

- **Validación en tiempo de codificación** -> Feedback loops más rápidos -> Mayor calidad + Desarrollo más rápido (Shift Left)
- **Mejor diseño del código** (Para que el código sea testeable requiere que sea modular y con responsabilidades separadas)
- **Código más robusto (Obligan a pensar como el "cliente" de la unidad)** -> Ayudan a diseñar el código y hacerlo más robusto (Sobre todo con TDD)
- **Menos incertidumbre para hacer cambios gracias a la batería de tests de no regresión** -> Detección automática y rápida de errores -> Menos regresiones
- **Documentación viva y actualizada (Los tests documentan el uso y comportamiento de las clases y están siempre actualizados)** -> Mejor documentación

Buenas prácticas para hacer tests unitarios

- **Patrón AAA:** Arrange / Act / Assert
- **Nomenclatura:** Ser descriptivos en los nombres de los Test Cases
- **Mockear dependencias externas**
- **Usar TDD (Test Driven Development):** Se comienza escribiendo el test y se ejecuta (el test falla), luego se codifica hasta que todos los tests pasen con éxito.
- **Cobertura inteligente:** Evitar excesivos tests con poco aporte, ni demasiado pocos
- **Incluirlos en el pipeline de CI/CD,** como chequeo previo antes de integrar nuevo código

FractionalCTO.es

Demo

Java con VSCode y Copilot

Uso de la IA para Unit Tests

Casos de uso que podemos pedirle a la IA

- Generar el fichero de reglas/instrucciones para Copilot con buenas prácticas
- Incluir los tests unitarios al generar el código
- Hacer una clase “legacy” testeable
- Crear un test plan, que nosotros podemos codificar
- Hacer TDD
- Revisar el código de los tests
- Automatizar la inclusión de los tests en el pipeline de CI/CD

Resumen

La creación de tests unitarios es una práctica muy extendida que ayuda a hacer *Shift Left* de la calidad, aumentando la calidad del código desde que se escribe por primera vez.

Hoy en día gracias a la IA, el coste de crear tests unitarios ha disminuido significativamente, al ser el código de los tests generalmente sencillo es buen candidato para usar IA.

Recomendación: Si estás empezando con los tests unitarios, empieza por hacerlos tú sin IA para entenderlos y luego delegarlos a la IA, así habrás adquirido la experiencia necesaria para revisar y corregir a la IA.

Repositorio con el código: https://github.com/gerodp/intro_unit_tests_with_ai



Gracias

Web

<https://fractionalcto.es>