



Integration Tests con SL

Ing. Aristóteles Aarón Ibáñez
Werthermänn





Agenda

- Revisión de Unit Tests
- Fundamentos de Integration Tests
- Demo sencilla



Revisión de Unit Tests


- Las pruebas unitarias aíslan una parte autónoma de código para **validar** su comportamiento.
- Se realiza durante la codificación, de tal manera que **garantiza** que lo implementado realmente funciona.
- Es promovido como parte del **DoD** de una **user story** en Scrum.
- Permite promover el enfoque de programación TDD.

Las 3 A's en los Unit Tests

- **Arrange (Organizar):** Define los requisitos que la prueba debe cumplir.
- **Act (Actuar):** Define la ejecución de la prueba que dará lugar al resultado a observar.
- **Assert (Afirmar):** Comprueba si los resultados obtenidos son los mismos que los resultados esperados, en caso contrario se deberá corregir el código hasta obtener la correspondencia.



Ventajas de los Unit Tests


- Demuestran que la lógica del código es la correcta.
 - Cada prueba se realiza sobre una funcionalidad atómica específica.
 - Deben contemplar tanto escenarios favorables como desfavorables.
 - Documentan técnicamente un proyecto.
 - Son parte fundamental de las buenas prácticas de CI en DevOps.
- 

Frameworks de automatización

- **XUnit:** Free open source community que automatiza las pruebas unitarias en todos los lenguajes de la plataforma .Net
 - xUnit.net trabaja con ReSharper, CodeRush, TestDriven.NET y Xamarin. Es parte del [.NET Foundation](#), y opera bajo su [código de conducta](#). Está licenciado por [Apache 2](#) (an OSI approved license).
- **NUnit:** Proviene de la evolución de JUnit y está disponible para todos los lenguajes de programación de la plataforma .Net.
 - NUnit es un software Open Source y NUnit 3 está lanzado sobre [MIT license](#). Los últimos releases usan [NUnit license](#).
 - Ambas licencias permiten el uso de NUnit en aplicaciones comerciales en forma gratuita y sin restricciones.



Integration Tests o Integration and Testing (I&T)

- Es un tipo de prueba de software que combina diferentes unidades o componentes de una aplicación de software.
 - Expone cualquier defecto que se produce durante la interacción de los componentes.
 - Verifica que cada unidad se comuniquen con otra apropiadamente garantizando la post-integración
- 

Unit Tests vs Integration Tests

Unit tests	Integration tests
Un solo componente es probado (funcionalidad aislada)	Se prueba la interacción de muchos componentes (funcionalidad compleja)
Se crea para probar escenarios aislados sobre los métodos de una sola clase.	Se crea para probar flujos de operación que comprometen distintos métodos y clases.
Se prueban escenarios favorables y desfavorables para un solo método (son sencillos y no requieren mucho nivel análisis)	Se prueban escenarios favorables y desfavorables para distintos métodos y clases (por su complejidad, requieren de un buen nivel de análisis)
Se realiza paralelamente a la codificación, en algunos casos puede ser antes (TDD) Se implementa como un paso de la CI (DevOps)	Se realiza al finalizar todas las pruebas unitarias Se implementa como un siguiente paso de la pruebas unitarias en CI (DevOps)

Pasos de un Integration Test

1. Preparar el plan de pruebas.
2. Diseñar casos de prueba, test scenarios, use cases, scripts.
3. Ejecutar luego de los Unit Tests / Integrar módulos.
4. Detectar, reportar y corregir errores.
5. Volver a ejecutar el paso 3

Tipos de Integration Test

- Big Bang
 - Involucra la integración de todos los componentes una vez y los prueba como una sola unidad.

Ventajas

- Es ideal para sistemas pequeños.
- Es fácil identificar errores, permitiendo ahorrar tiempo y esfuerzo en el desarrollo.

Deventajas

- Dificultad en localizar el origen del error.
- Consume mucho tiempo en sistemas grandes.
- Se tiene que esperar hasta terminar todo

Tipos de Integration Test

- Top down
 - Involucra las pruebas de módulos de alto nivel primero, seguido de los módulos de bajo nivel.

Ventajas

- Es fácil identificar y aislar defectos.
- Los testers revisan las unidades más importantes primero, esto ayuda a encontrar problemas de diseño.
- Es posible crear un prototipo tempranamente.

Deventajas

- La revisión de componentes de bajo nivel puede tomar mucho tiempo.
- Cuando se involucran varios componentes, el testing puede ser complicado.

Tipos de Integration Test

- Botton up
 - Involucra las pruebas de módulos de bajo nivel primero, seguido de los módulos de alto nivel.
- **Ventajas**
 - Es fácil identificar y encontrar fallas.
 - Menor tiempo de implementación, no se tiene que esperar hasta terminar todos los componentes.
- **Deventajas**
 - Probar todos los componentes puede tomar mucho tiempo y puede impactar en el release.
 - Los componentes críticos pueden ser probados al final, lo que puede llevar a la pérdida de algunos defectos.
 - Cuando hay muchos componentes de bajo nivel, se vuelve complicado.
 - No es posible crear un prototipo temprano.