Testing Automation

Curso de Ingeniería de Software

Primer Semestre 2022

UNIVERSIDAD DE
COSTA RICA

SR-CIE

Carrera de
Informática Empresarial
Sedes Regionales

Contenidos

- ¿Qué es el testing?
- Qué es Testing Automation?
- TDD - Test Driven Development
- Demo

¿Qué es el testing?

¿Qué es el testing?

- “Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder.”

Tipos de Testing (los más comunes)

1. **Unit tests**
   - Son usados para testear pequeñas partes del código, relativamente aisladas. En general una clase o un método.

2. **Integration tests**
   - Testean la forma en que diferentes partes del código trabajan juntas y se pasan mensajes.

3. **Application tests**
   - También llamados Acceptance Tests o Functional Tests. Prueban toda la aplicación. En general usan un driver que emula el uso del navegador.

Qué es Testing Automation?

Qué es Testing Automation?

- “Consiste en el uso de software especial (casi siempre separado del software que se prueba) para controlar la ejecución de pruebas y la comparación entre los resultados obtenidos y los resultados esperados”

Para qué sirve?

- Asegurarnos que los cambios que hacemos en nuestra aplicación no impacten en cosas que ya hicimos
- Asegurar la calidad de la aplicación
- Enfocar el uso del tiempo del equipo de desarrollo en tareas no repetitivas
- Simplifica el mantenimiento
- Describe comportamientos y define expectativas

Ventajas

- Evitar tareas repetitivas y vicios por repetición
- Permite enfocarse en el testing manual de nuevas funcionalidades que son menos estables
- Fuerza a pensar en los casos de test y documentarlos
- Permite correr los mismos tests en distintos entornos en tiempo mínimo
- Se puede incorporar a un modelo de Integración Continua (coming soon)
- Se pueden escalar y re-utilizar para hacer test de stress

Desventajas

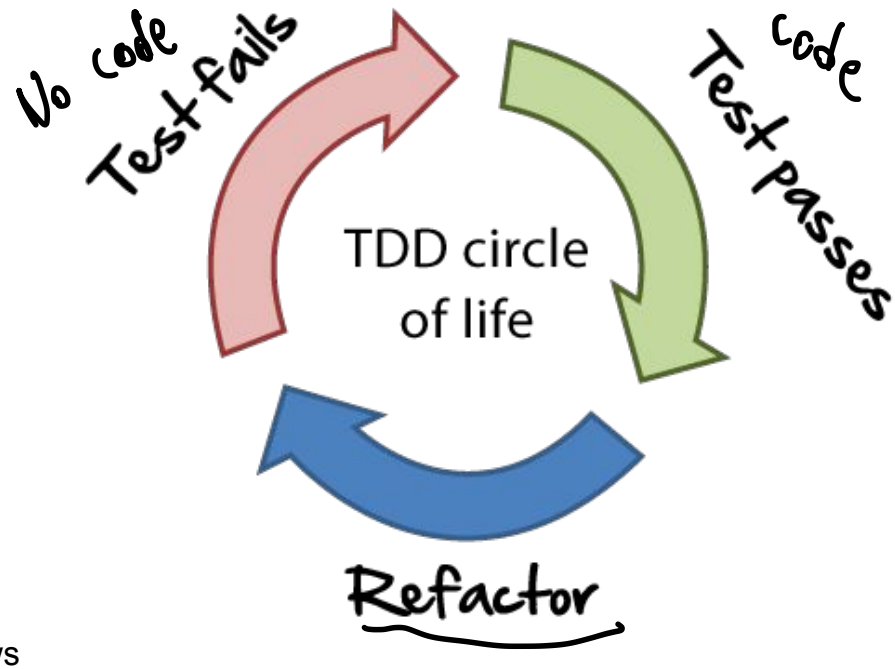
- Período inicial de aprendizaje y adaptación del equipo de desarrollo
- Hay que mantener la suite de tests
- Los tests se ven impactados por los cambios en la UI (existen tácticas y patrones para tolerar esto)
- El perfil de tester puede requerir habilidades técnicas avanzadas (desarrollo, diseño)

TDD - Test Driven Development

¿Qué es Test Driven Development?

- TDD es una práctica de programación, dividida en 3 partes:

1) – Escribir las pruebas (tests)
2) – Escribir el código fuente
3) – Refactorizar el código escrito

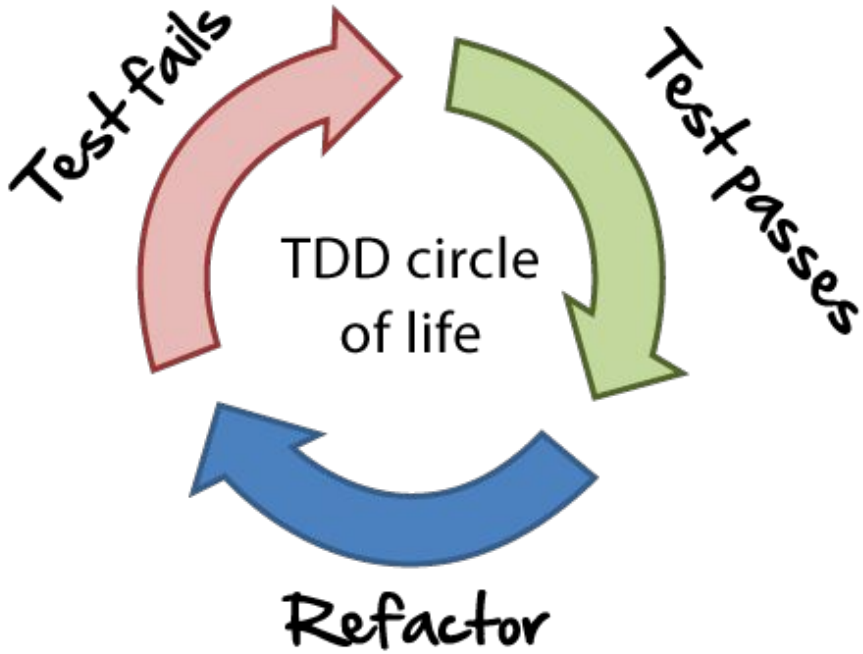


Ventajas

- **Código más robusto**
  - Nos aseguramos que todo el código que escribimos esta probado y lo podemos seguir probando cuantas veces necesitemos
- **Más seguro**
  - Si ejecutamos los tests, estamos seguros que nada dejó de funcionar con cualquier cambio que hagamos
- Más mantenible
  - Teniendo una suite de tests, podemos refactorizar el código, sabiendo que no tenemos que probar todo a mano, sino que ya tenemos código que prueba lo que cambiamos

Mantra - Red, Green, Refactor

- Escribo un test que falle, porque aun no tengo el código hecho
- Escribo el mínimo código para que el test pase
- Mejoró la solución, refactorizando el código



Demo

Nuevo Requerimiento

- El cliente quiere que se agregue el campo:
  - **last name**

- Decidimos entonces hacer los cambios requeridos con TDD

Conclusiones

Conclusiones

- El testing nos ayuda a ganar velocidad y asegurarnos que ciertas cosas funcionan
- Es muy difícil tener una cobertura total de la app
- Hacer testing no asegura calidad de código
- Teniendo tests de browsers, nos aseguramos que la funcionalidad testeada, en ese caso, funciona
- El usuario siempre puede probar una combinación que no cubri con los tests y romper mi app

Code coverage

- Los test hacen que se ejecute una parte del código de la aplicación
- Cada línea fue (o no) ejecutada
- Las líneas que no se ejecutan no fueron testeadas
- Se llama “code coverage” al % de líneas que nuestros test hacen ejecutar

Referencias

https://dev.to/jackmarchant/no-excuses-write-unit-tests
https://github.com/laravel/dusk
https://laravel.com/docs/master/dusk