

Microservices 101.ATDD

Raimundo Alegría



El objetivo de este learning path es recorrer de una forma práctica algunos los aspectos de los aspectos más relevantes que influyen tener éxito a la hora de desarrollar una arquitectura de microservicios



Talleres previstos. Microservices 101

- Conceptos básicos
- **ATDD**
- TDD
- Containers. Docker
- Infraestructura. Kubernetes
- Continuous delivery
- Seguridad
- Bases de datos
- Event sourcing



¿Qué es?





De forma análoga al test driven development, Acceptance Test Driven Development (ATDD) hace que diferentes miembros del equipo con diferentes perspectivas (customer, development, testing) colaboren en escribir ejemplos de cómo debería funcionar la aplicación.

Características

- Mejoran la comunicación entre el negocio y los desarrolladores.
- Aseguran que lo que se está construyendo es lo correcto.
- Se convierten en la fuente definitiva de verdad de lo que hace el sistema.



Características

- Son especificaciones ejecutables.
- Crean una documentación viva del sistema.
- Definen qué hace el sistema no cómo lo hace.
- Generan un conjunto de test de regresión.



Microservicios y ATDD

- Generan una “Red de seguridad” para poder hacer cambios
 - Permiten hacer refactorings a nivel de arquitectura.
 - Habilitan la arquitectura evolutiva.
- Reducen la cantidad de test manuales.



No sólo funcionales

- Seguridad
 - Escenarios de abuso
- Reliability
 - Escenarios error en la infraestructura del sistema
- Contrato.
 - Especificación de qué debo proporcionar a mis clientes



Primer ejercicio

Descargar el código:

<https://github.com/rai22474/microservices-101>



¿Cómo?



Cucumber

- Cucumber es el framework más extendido para automatizar pruebas de aceptación.
- Su primera implementación la realizó la comunidad ruby.
- Ahora hay versiones en muchos lenguajes como java, javascript, etc.



Gherkin

- Gherkin está escrito en texto plano en inglés (u otro de los más de 60 lenguajes en los que está traducido) con un poco de estructura extra.
- Está diseñado para ser fácil de aprender por los no programadores.
- Tiene la suficiente estructura para permitir una descripción concisa de los ejemplos que permite describir reglas de negocio en la mayoría de los dominios.



Feature: Customer registration

As a customer,
I need to register on the application to create my own account and have a mechanism to log into my own account

Scenario: Register a new person successfully

Given a system without any registered users

When a person is registered in the system with the following data:

id	name	lastName	idCard	email	terms	mobilePhone
43r34t	Bart	Simpson	50861048K	lmb@gmail.com	true	636589713

And therefore, it exists a customer, identified by "43r34t" with the data:

name	lastName	idCard	email
Bart	Simpson	50861048K	lmb@gmail.com

And the customer "50861048K" has the following limits into its bucks:

rechargeThisPeriod	rechargeLast	rechargeRemaining	rechargeMax
0.0 EUR	0.0 EUR	2500.0 EUR	2500.0 EUR

Feature

Describe una característica del sistema o un aspecto particular de una característica. Es una manera de proporcionar una descripción de alto nivel del sistema así como una manera de agrupar un grupo de escenarios.



Feature

Una feature tiene tres elementos básicos:

- La palabra clave Feature
- Un nombre (en la misma línea)
- Una descripción opcional (pero muy recomendable) que puede extenderse varias líneas.



Feature

Feature: Customer registration

As a customer,
I need to register on the application to create my
own account and have a mechanism to log in



Escenario

Un escenario es un ejemplo concreto que ilustra una regla de negocio. Consiste en una lista de pasos.

- Puedes tener tantos pasos como quieras, pero es recomendable mantener el número entre 3-5 por escenario.
- Si se hace más largo pierde expresividad como especificación y documentación.



Escenario. Pasos

- Given: Describe el contexto inicial del sistema.
- When: Es usado para describir un evento, o una acción.
- Then: Indican cual es el resultado o salida esperada del sistema.



Segundo ejercicio

- Implementar una feature en el proyecto de ejemplo.





Muchas gracias

Referencias (Libros)

- The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends (Facets of Ruby) by David Chelimsky, Dave Astels ...
- Specification by Example: How Successful Teams Deliver the Right Software. Gojko Adzic
- Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing Paperback. Gojko Adzic.



Referencias (Libros)

- The Cucumber Book: Behaviour-Driven Development for Testers and Developers. Matt Wynne, Aslak Helleoy, Steve Tooke.
- The Cucumber for Java Book: Behaviour-Driven Development for Testers and Developers. Seb Rose, Matt Wynne, Aslak Helleoy.

