

PRUEBAS FUNCIONALES Y ATDD

Pruebas Funcionales y ATDD

- Tipos de tests
- Piramide de Mike Cohn
- Nightwatch-cucumber
- Gherkin syntax
- Feature Files
- Step Definitions
- ATDD

TIPOS DE TESTS

TESTS UNITARIOS

- Tratan de asegurar el funcionamiento esperado de componentes individuales (funciones, métodos, rutinas, lógica, etc.)
- Más rápidas, menor coste, menor tiempo de desarrollo.

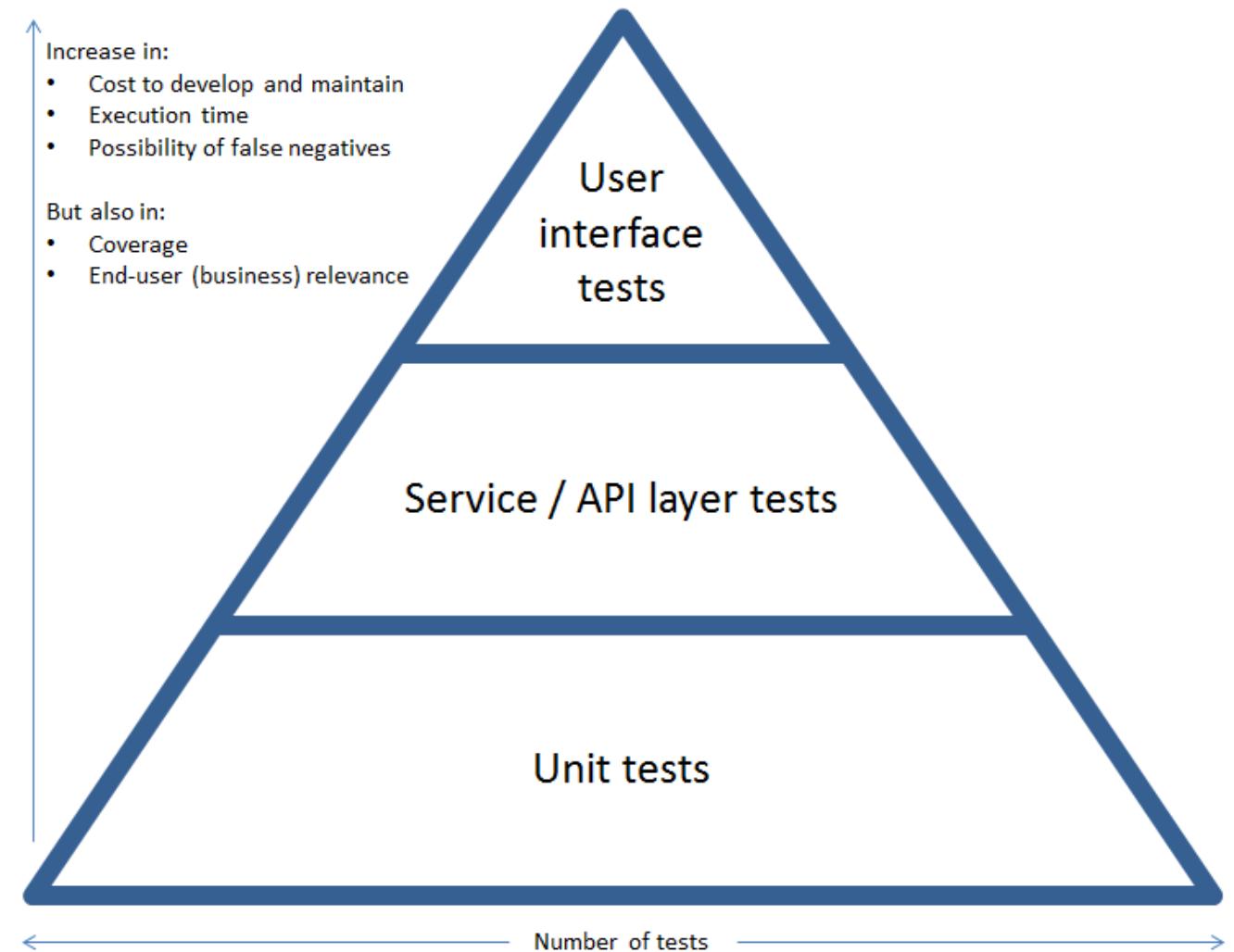
TESTS DE INTEGRACIÓN

- Tratan de garantizar el funcionamiento esperado entre componentes del sistema (API, registros en base de datos, envío de emails, logs, etc.)
- Rápidas, coste medio, tiempo de desarrollo medio.

TESTS DE FUNCIONALIDAD

- Enfocadas en garantizar que la aplicación se comporte como se espera, desde la perspectiva de los usuarios.
- Más lentas, mayor coste, mayor tiempo de desarrollo.

- Los TEST UNITARIOS, ayudan a construir el producto de manera correcta.
- Los TEST DE ACEPTACIÓN, ayudan a construir el producto correcto.



Nightwatch- Cucumber

npm install --save-dev nightwatch

npm install --save-dev cucumber

npm install --save-dev nightwatch-cucumber

npm install --save-dev selenium-server

npm install --save-dev chromedriver

Crear nightwatch.conf.js

Features

Step_definitions

Google feature

*features/google.feature

*features/step_definitions/google.js

- **Feature:** Google Search
- More text and description
- **Scenario:** Searching Google
- **Given** I open Google's search page
- **Then** the title is "Google"
- **And** the Google search form exists

Nightwatch- Cucumber Pretty

npm install –save-dev cucumber-pretty

nigthwatchOuput: true|false

'--format',
'node_modules/cucumber-pretty'

Nightwatch- Cucumber Reportes

```
npm install --save-dev  
cucumber-html-reporter
```

```
'--format',  
'json:reports/report.json'
```

Crear archivo para generar
reporte “create-html-report.js

Nightwatch- Cucumber Screenshots

Activar capturas en
'nightwatch.conf.js'

- screenshots : {
 - enabled : true,
 - on_failure : true,
 - path: 'screenshots/default'
- }

Activar capturas en los reportes en
'create-html-report.js'

- screenshotsDirectory: 'screenshots/>,
- storeScreenshots: true

Gherkin Syntax

- Domain Specific Language (DSL)
- Especificación dedicada a resolver un problema en particular
 - Comunicación más natural sin entrar detalles de implementación.
- 5 sentencias principales:
 - Feature:
 - Scenario:
 - Given
 - When
 - Then

Feature Files

- Archivos que contienen nuestras historias para ser validadas
- Terminados en .feature
- Contiene una sentencia Feature y tantos escenarios como se desee
 - Feature: Manage Customers
 - Scenario: Create Customer
 - Scenario: Update Customer
 - Scenario: Send Email to Customer
 - -
 - -
 - -

Step Definitions

Las definiciones son escritas en Javascript

Generalmente dentro del directorio features/step_definitions

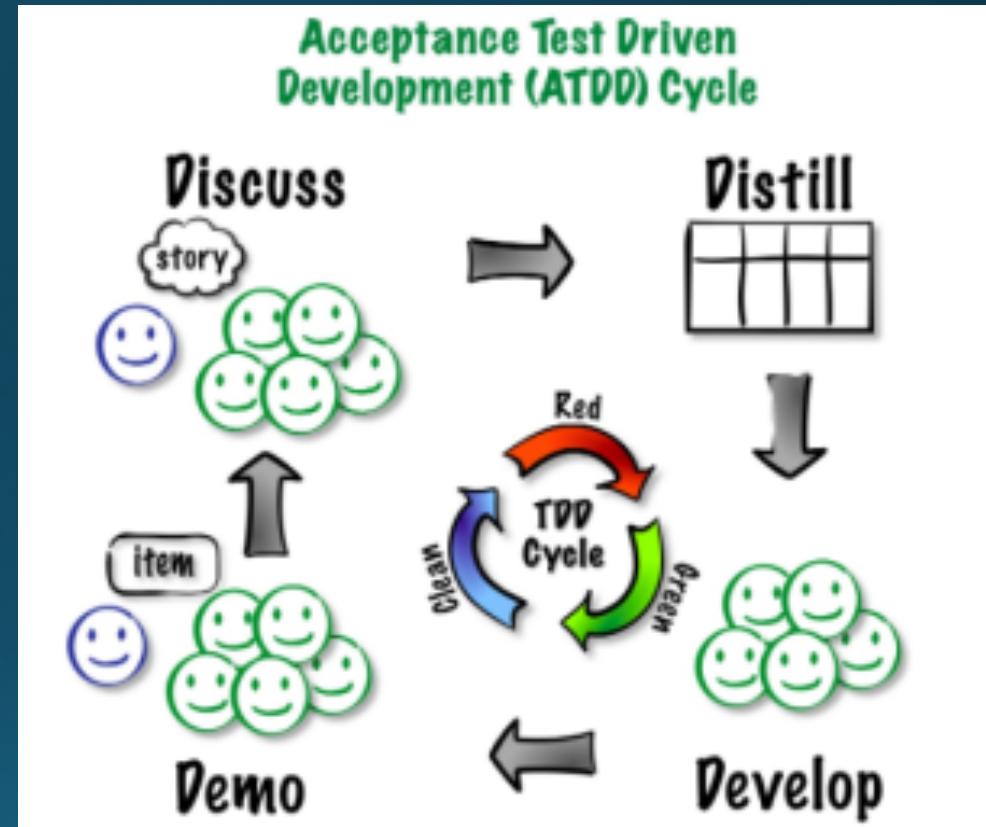
Utilizan regex para hacer match con las sentencias en los archivos features

Pueden retornar una función callback o una promesa

Toman o más argumentos

ATDD (Acceptance Test Driven Development)

Los test de aceptación aseguran que se ha desarrollado la funcionalidad esperada.



ATDD

Todo el equipo involucrado.

Analisis conjunto de los criterios de
aceptación.

Comunicación y visión compartida.

Ejemplos significativos.

Introducir la calidad en el proceso.

Criterios de aceptación en una lista

- As an user, I want to be able to register to enter to site
 - It should have an input to name
 - It should have a input to lastname
 - It should have a input to email
 - It should have a input to password
 - All fields must be validated.

Basados en una historia de usuario.

Criterios de aceptación escritos en forma de historia.

Utilizando *Given*, *When*, *Then* y *And*.

Diseñada para proporcionar más detalles a la historia de usuario.

Historia de
aceptación

Historia de aceptación

- As an user, I want to be able to register to enter to the site
 - Given I'm on the home screen,
 - And I'm not registered,
 - Then I see the 'Register' button,
 - When I press the 'Register' button,
 - Then I should see the 'Register form',
 - Given I fill name input with "Carlos",
 - And lastname input with "Gutierrez",
 - And email input with "carlos.gutierrez@gigigo.com.mx"
 - And pass input with "123456"
 - And I press the submit button
 - Then I'm back on the home screen,
 - And the site shows Welcome Carlos.

Automatizar historia

- Crear archivo register.feature
- Crear archivo register.js en `step_definitions`
- Implementar pasos
- Red, Green, Refactor
- Documentación viva

¿Preguntas?