



# Pruebas de integración REST

Erik Giron  
@kreig  
[linkedin.com/in/erikgiron](https://linkedin.com/in/erikgiron)



# Programa

- Testing
- Tipos de pruebas
- Metodologías
- Unit Testing
- Mocking
- Integration Testing
- REST mocking
- Wiremock

# Testing

¿Para qué?

- Validación
- Verificación
- Entrega Continua



# Tipos de pruebas

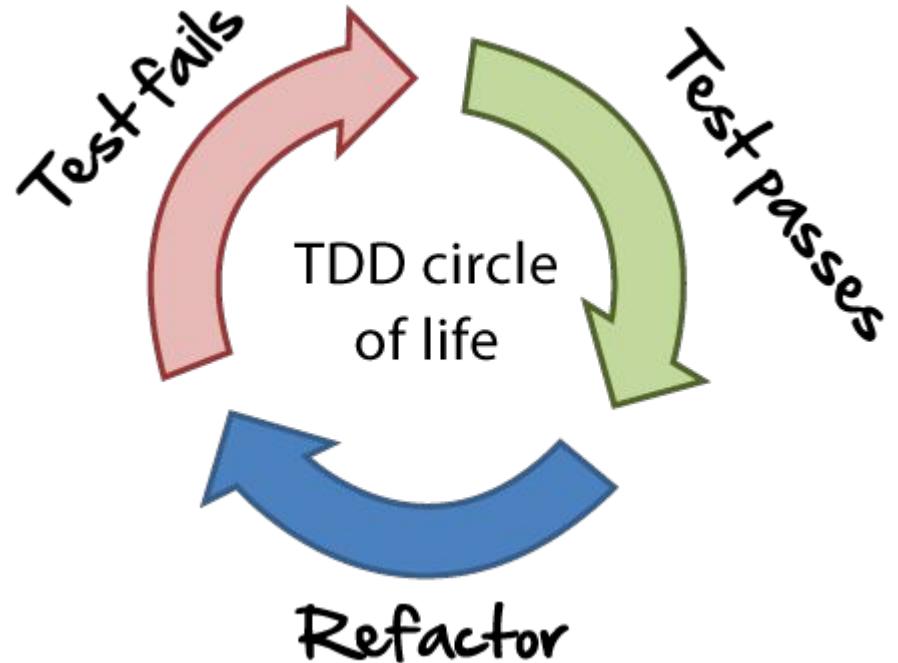
- Pruebas unitarias
- Pruebas de integración
- ... y las demás que no implican código.
  - Aceptacion
  - UI
  - Rendimiento
  - ...



ILLUSTRATED BY SEGUE TECHNOLOGIES

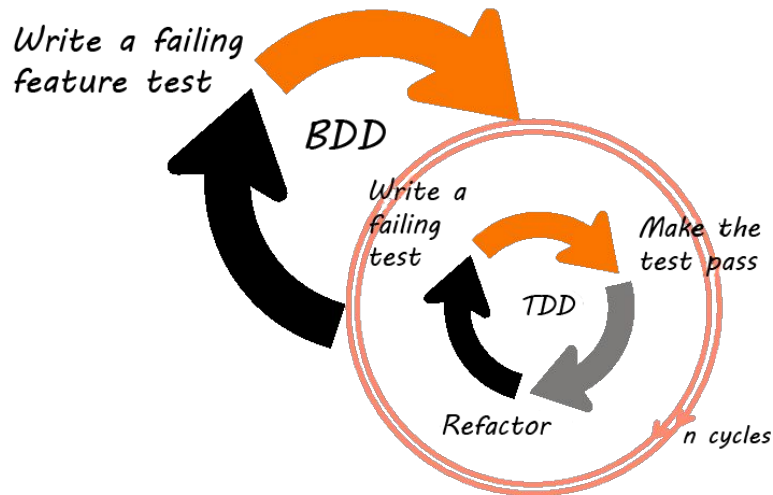
# Metodologías orientadas a pruebas - TDD

- TDD
  - Fail
  - Pass
  - Refactor



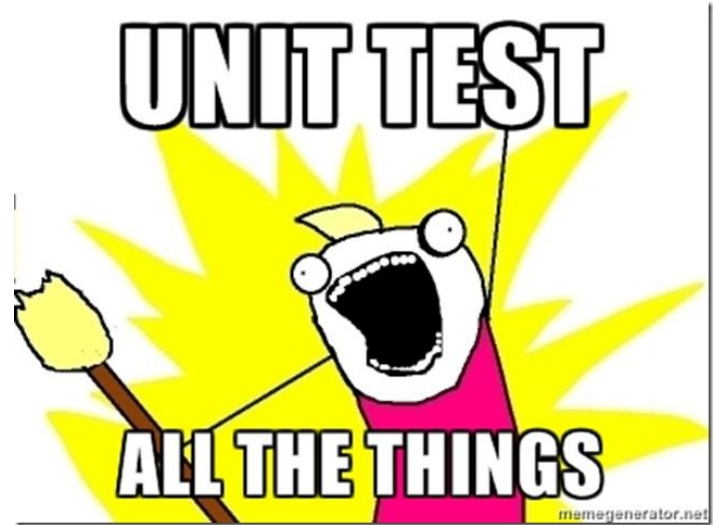
# Metodologías orientadas a pruebas - BDD

- Test de comportamiento unitario.
- DDD y Lenguaje ubicuo
  - Lenguaje específico de dominio (DSL) en casos de prueba
    - *given*
    - *when*
    - *then*



# Pruebas unitarias

- Aislar una unidad de funcionalidad a la vez...
  - metodos
  - funciones
  - procedimientos
- ... y verificar que cada una funciona correctamente
- Se debe cubrir todos los casos de comportamiento
  - Un caso por condicional o ruta alterna de comportamiento
  - Code coverage 100%



# Frameworks de Pruebas Unitarias

- JUnit
  - El mas antiguo
  - Desarrollo activo
  - Test Parametrizados
    - Theories
  - Aserciones
    - nativos
    - hamcrest





# Frameworks de Pruebas Unitarias

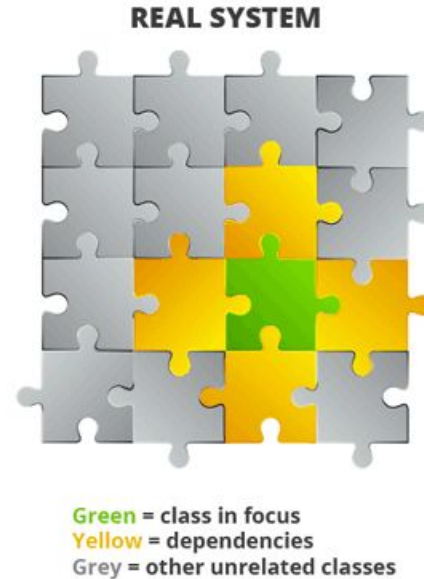
- TestNG
  - Basado en JUnit
  - Test grouping
  - Assertion
    - hamcrest
  - verbosidad menor
    - Multithreading
    - Test parametrizados
    - Suite Testing

TestNG

# Ejemplo prueba unitaria con JUnit

# Mocks

- Objetos que simulan una interacción
  - Colaboradores a nivel clase (Unit Testing)
    - Proveedores
    - Servicios
  - Sistemas (Integration Testing) invocados desde colaboradores
    - Repositorios
      - SQL
    - Channels
      - AMQ
    - Endpoints
      - REST
      - SOAP
    - Configuraciones externas
      - Property bundles, ZK, Memcached, redis, etc



# Mock frameworks

## JMock

- Combina simulación y verificación de llamadas
- Menos expresivo
- Más estricto.

## Mockito

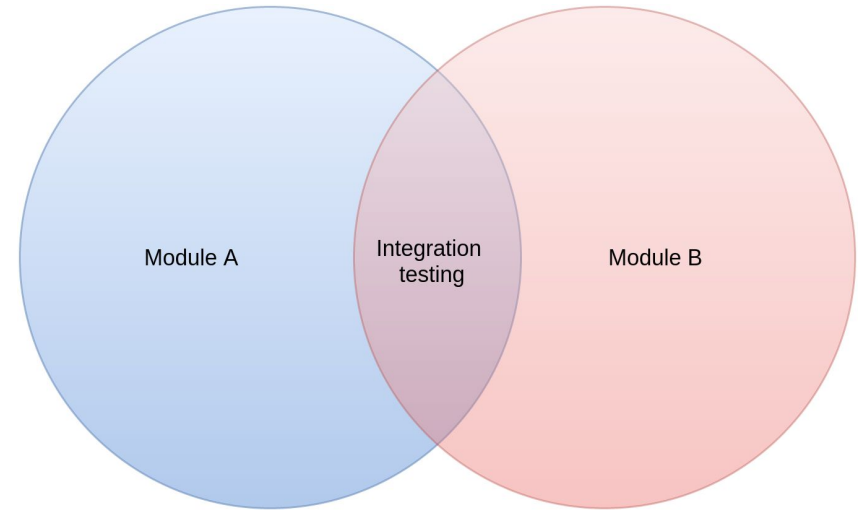
- DSL parecido a BDD
- Expresivo
- Verificación de llamadas a métodos simulados debe hacerse por separado.

# Ejemplo de Mocking frameworks

Código disponible en <https://github.com/kreigiron/javaday-17>

# Integration Testing

- Interacciones entre sistemas.
- Se hacen por cada interfaz hacia o desde el interior del sistema.
  - Gateways
  - Repositories
  - Controllers (MVC).
  - Channels (Queue).
  - REST endpoints.
  - REST clients.
- Sistemas externos **DEBEN SER simulados**



# REST mocking

- Rest assured
  - Biblioteca
    - DSL mas expresivo para contenido Json y XML
  - Standalone
    - No disponible

## Wiremock

- Biblioteca
  - DSL tipo
- Standalone de simulacion
  - Archivo de mapeo en json

# Wiremock como biblioteca

## Requerimientos:

- JDK Reciente (1.6+ 1.8+ dependiendo la version de wiremock)
- Maven reciente con la dependencia de wiremock
- Plugin de maven Failsafe para pruebas de integracion

O

- IDE que soporte correr pruebas unitarias
  - Netbeans
  - Eclipse
  - IDEA

```
<dependency>
```

```
  <groupId>com.github.tomakehurst</groupId>
```

```
  <artifactId>wiremock</artifactId>
```

```
  <version>2.10.1</version>
```

```
</dependency>
```



# Ejemplo wiremock en pruebas de integración

Código disponible en <https://github.com/kreigiron/javaday-17>

# Wiremock como aplicacion de simulacion

Requisitos:

- Java runtime reciente
- Wiremock Standalone jar (disponible en el sitio o en maven central)

# Ejemplo de wiremock como aplicación

Codigo disponible en <https://github.com/kreigiron/javaday-17>

¿Dudas?