Universidad Católica "Nuestra Señora de la Asunción"

DEPARTAMENTO de Electrónica e Informática

FACULTAD de Ciencias y Tecnología

MoFQA: Una propuesta para la generación automática de tests a partir de modelos siguiendo el proceso TDD

Ingeniería Informática

Proyecto Final de Carrera

Primera Presentación

Mayo 2016

Alumna: Linda Riquelme

Tutora: Ing. Magalí González
Co-Tutor: Dr. Luca Cernuzzi
Colaboradora: MSc. Nathalie Aquino

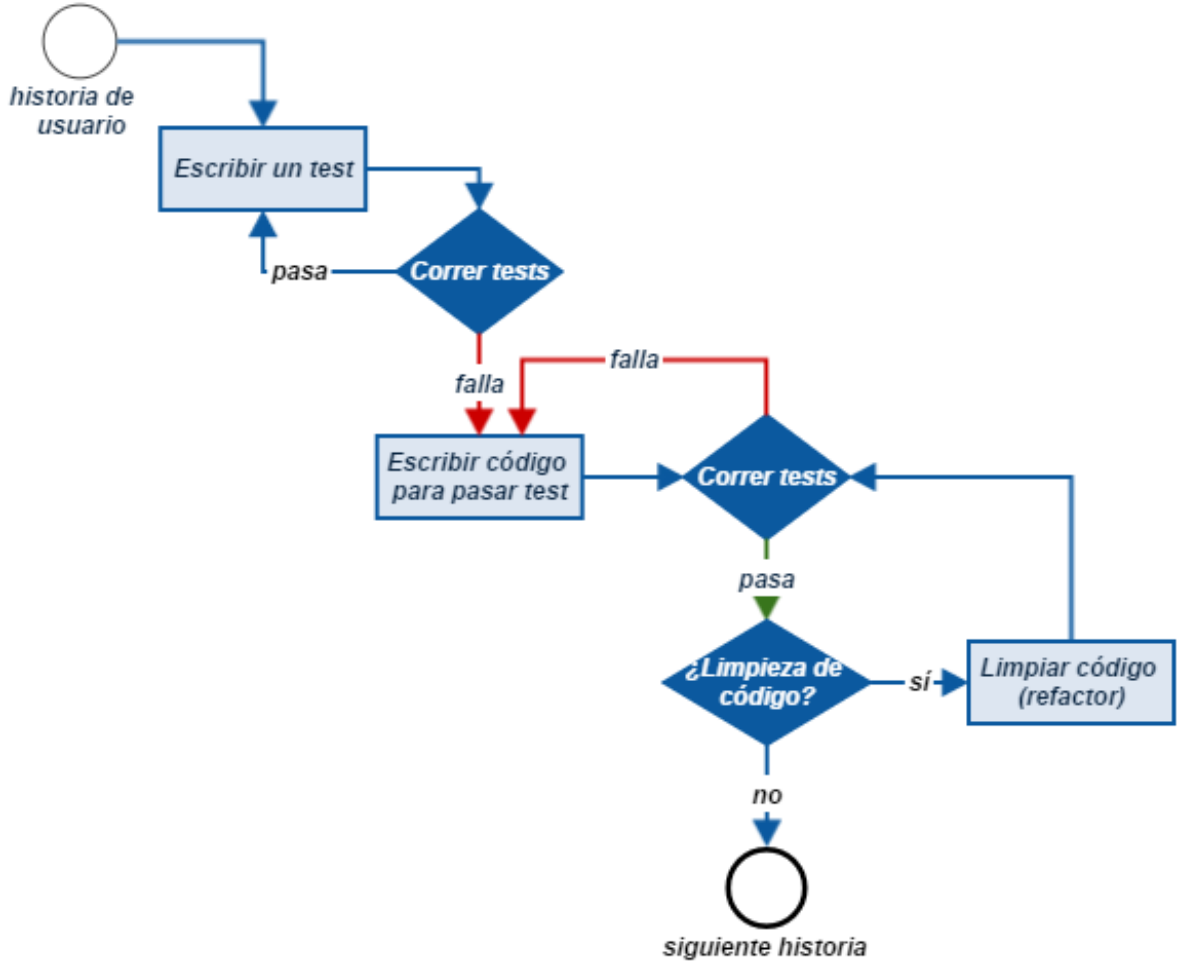
Este trabajo está siendo desarrollado en el marco del proyecto "Mejorando el proceso del desarrollo de *software*: Una propuesta basada en MDD" (14-INV-056), financiado por el Consejo Nacional de Ciencias y Tecnología (CONACYT).

Errores en el software

Explosión del cohete Ariane (1996)

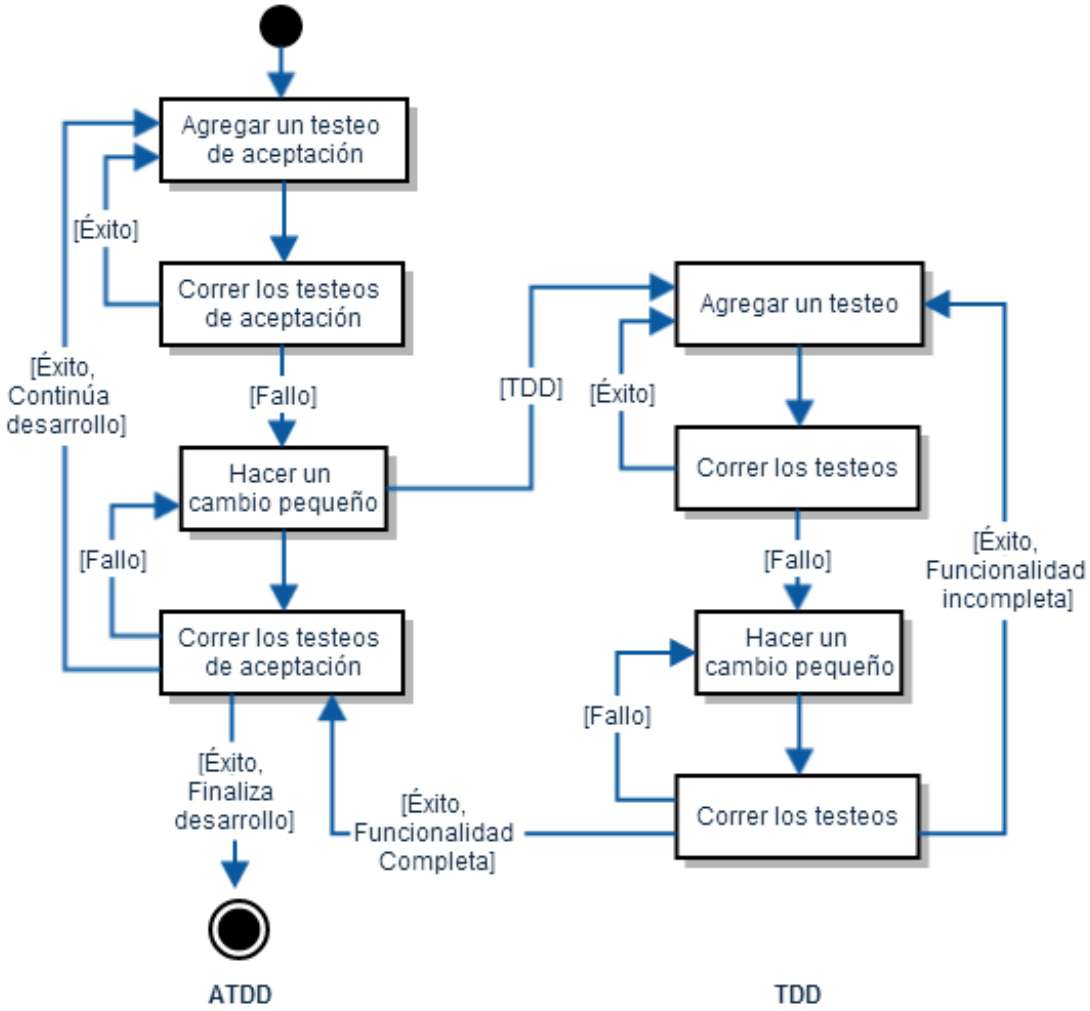
Pérdidas de Knight Capital Group Inc. (2012)

Test-Driven Development (TDD)



Proceso TDD

Acceptance TDD (ATDD)



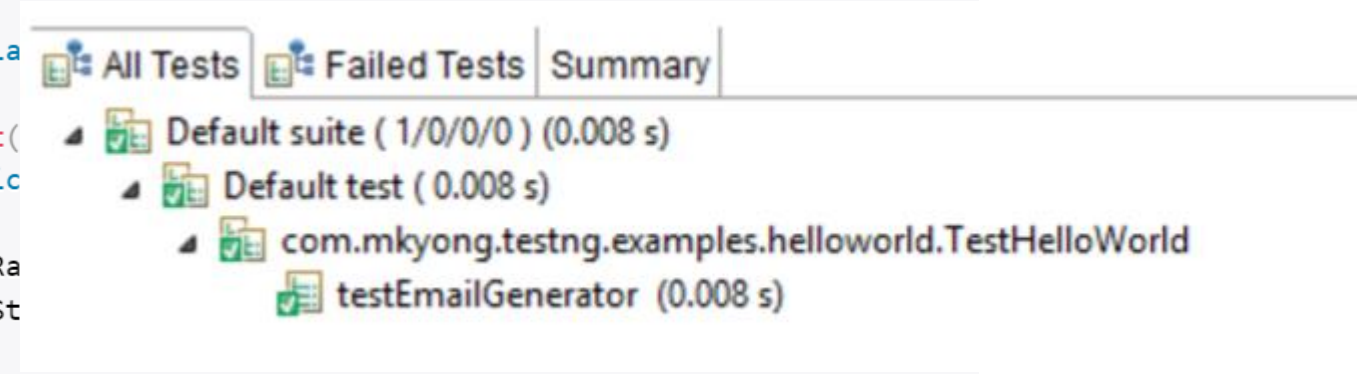
Implementación de TDD y ATDD en conjunto
Ambler, 2006

Herramientas de Testing Automatizado

- TestNG

TestHelloWorld.java

```
package com.mkyong.testng.examples.helloworld;

import org.testng.Assert;
import org.testng.annotations.Test;
import com.mkyong.testng.project.service.email.RandomEmailGenerator;

public cla

    @Test(
    public

        Ra
        St

        Assert.assertNotNull(email);
        Assert.assertEquals(email, "feedback@yoursite.com");

    }

}
```

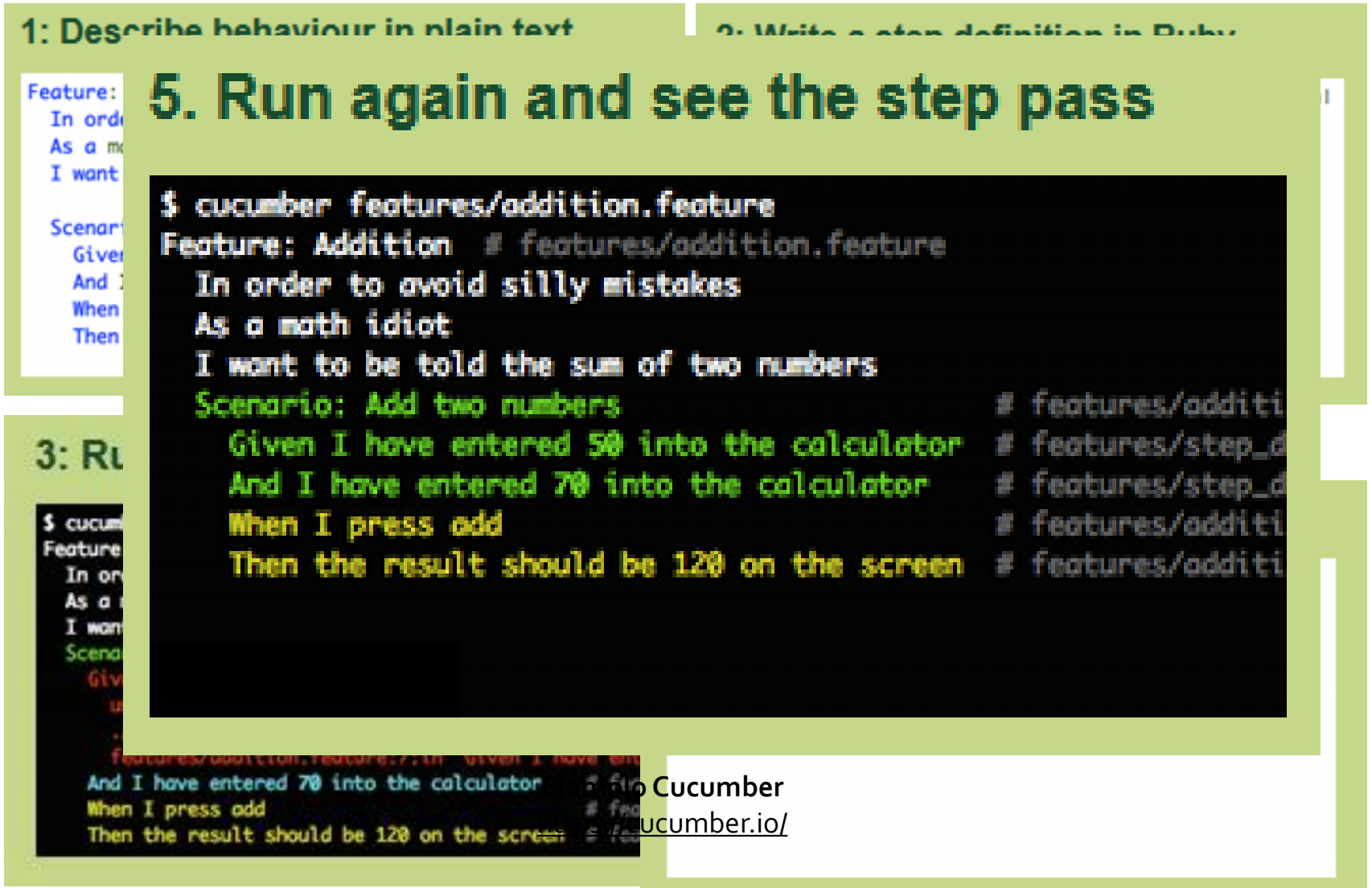


Ejemplo TestNG

http://www.mkyong.com/tutorials/testng-tutorials/

Herramientas de Testing Automatizado

- Cucumber: Una herramienta ATDD

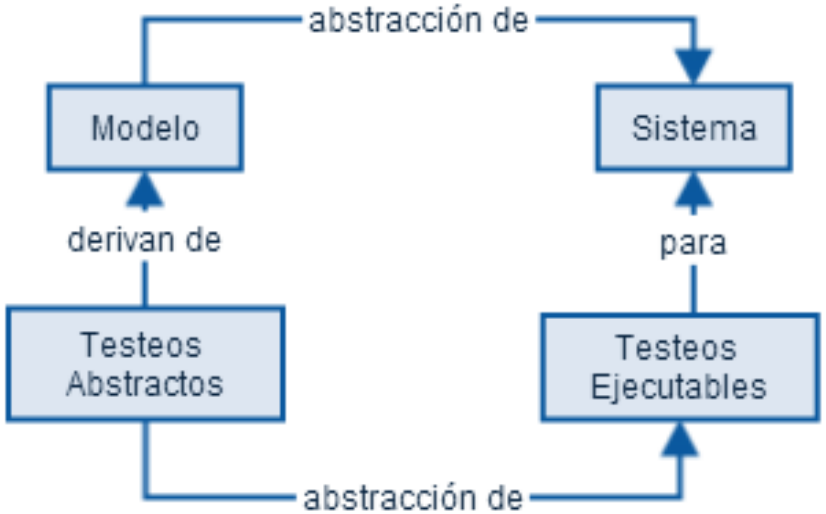


Cucumber
ucumber.io/

Test-Driven Development (TDD)

Test-Driven Development (TDD)

Model-Based Testing (MBT)

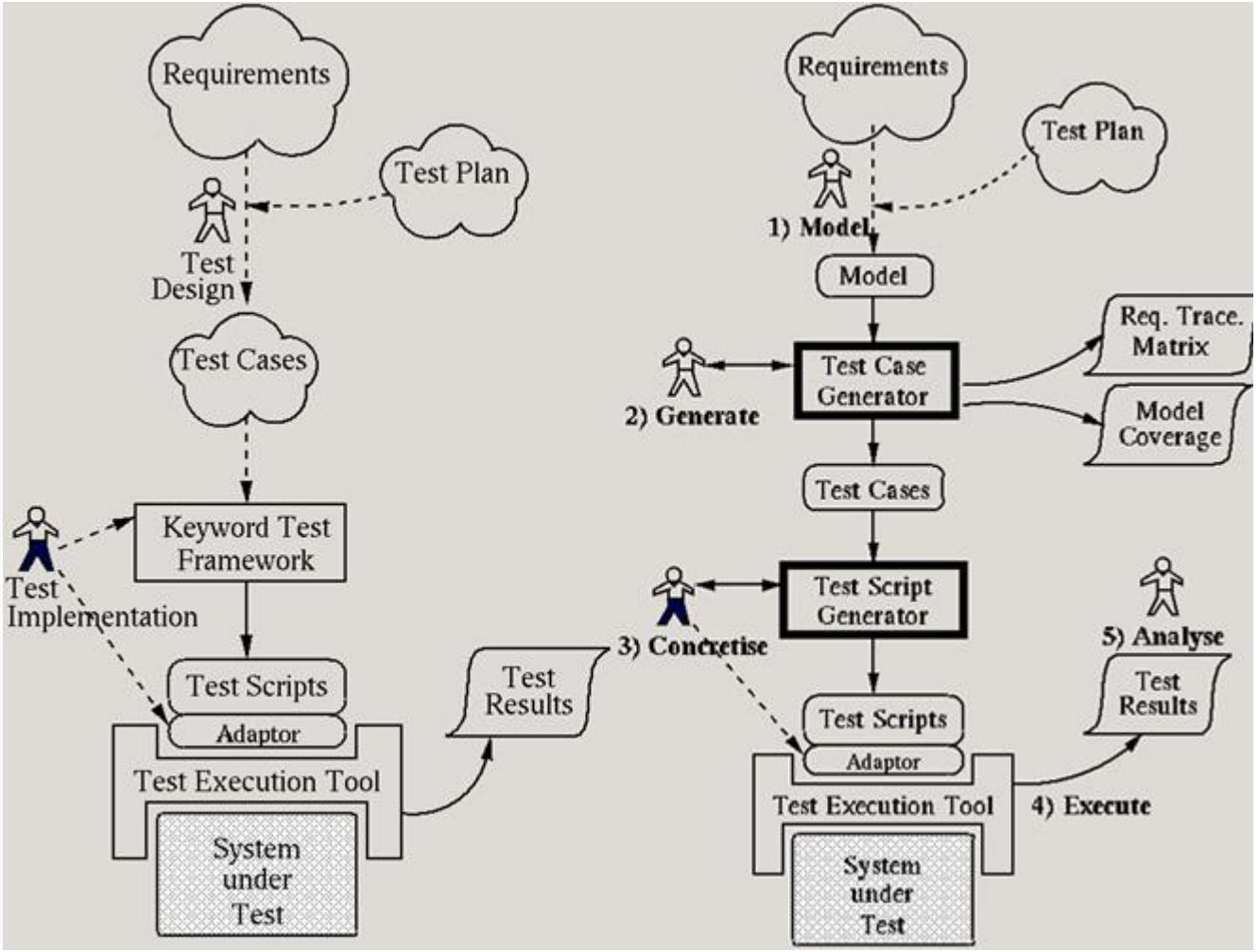


Model-Based Testing
Brambilla et al., 2012

Model-Based Testing (MBT)

Testing dirigido por palabras claves

Model-Based Testing



Generaciones de Automatización de Testing

Utting & Legeard, 2010

Herramientas MBT

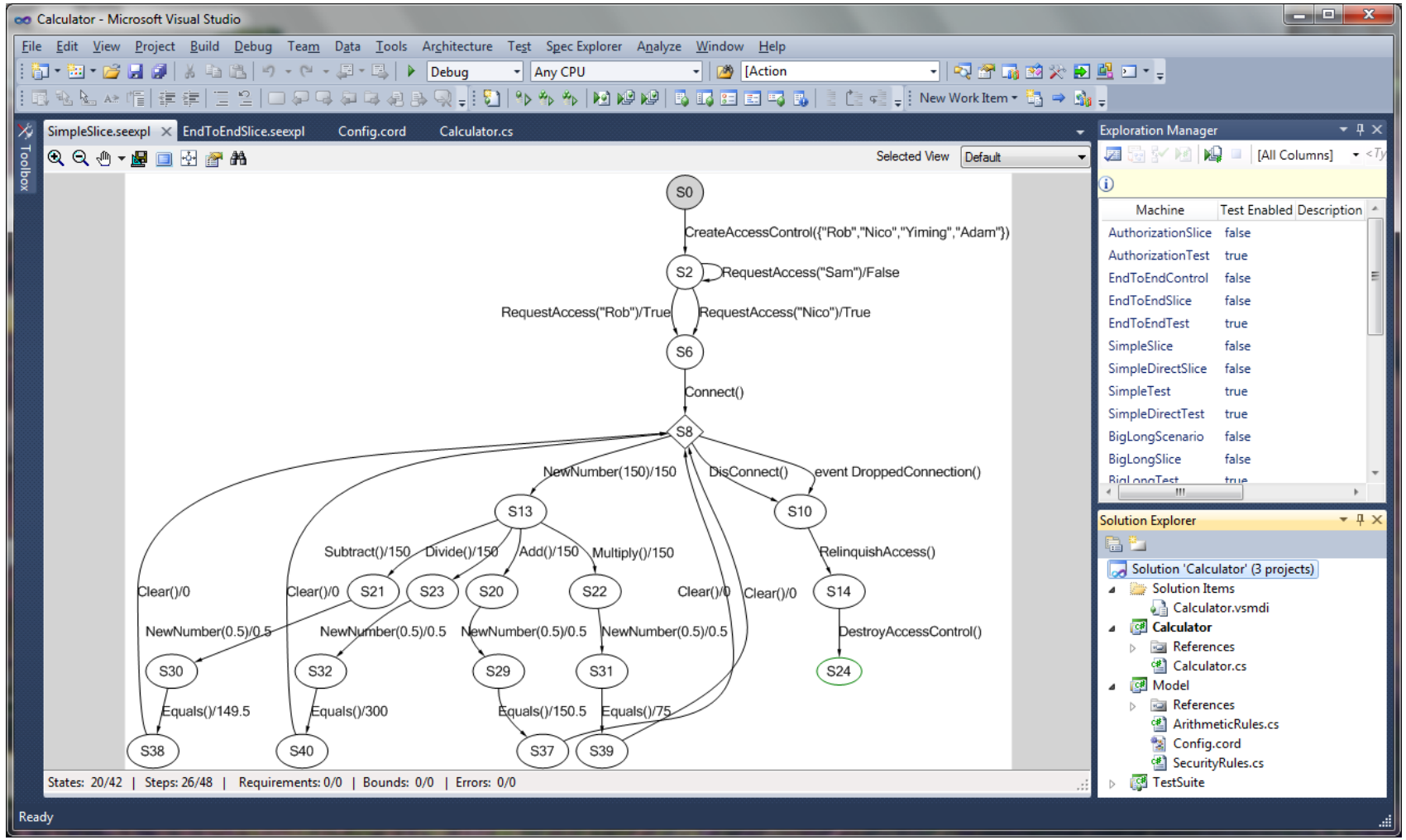
Nombre	Descripción
JUMBL	Java Usage Model Builder Library, soporta generación automática de tests estadísticos basados en modelos.
Spec Explorer	Herramienta para Visual Studio, genera test suites a partir de programas C#.
Torx	Testing para software reactivos, basado en su especificación formal.
Uppaal TRON	Entorno integrado para la validación y verificación a partir de modelos de sistemas de tiempo real.
Nmodel	Genera tests a partir de programas en C#, es posible añadir estrategias de generación. Precursor de Spec Explorer.
TopCased	Soporta métodos formales SysML, UML, OCL para el desarrollo y testing de sistemas embebidos.
ATS4 AppModel	Especificación de aplicaciones y tests a partir de modelos, simple interfaz para el manejo de modelos complejos para testing de aplicaciones móviles.
Autofocus 3	Soporta el modelado y análisis de la estructura y comportamiento de sistemas distribuidos reactivos y de tiempo real.
Fastest	Genera tests a partir de modelos expresados en notación Z, soporta testing unitario.
Jtorx	Derivado de Torx. Los tests derivados verifican la relación entre el sistema bajo test y un modelo de transiciones expresado en Aldeberan o GraphML.
GUITAR	A partir de un modelo genera un grafo de flujo de eventos (representa el modelo de test para GUI). Genera tests ejecutables.
MISTA	Modelado de tests usando redes de Petri. Genera códigos de test para Java, C, C++, C#, VB, HTML, Junit, Nunit, Selenium.
TEMA	Herramienta y metodología para la generación en línea de tests para testing GUI para smartphones.
EvoMT	Utiliza algoritmos genéticos para generar oráculos de tests.
Nmodel Remote Stepper	Extensión de Nmodel, MBT para sistemas en Java u otros lenguajes.
OSMO Tester	Modelado del comportamiento del software con máquinas de estado extendidas.
Parteg	Generador de tests de particiones.

Herramientas Open-Source para MBT

http://robertvbinder.com/open-source-tools-for-model-based-testing/

Herramientas MBT

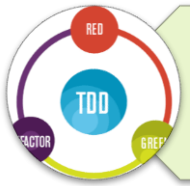
- Spec Explorer



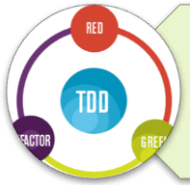
Ejemplo Spec Explorer para Visual Studio 2010

https://visualstudiogallery.msdn.microsoft.com/271d0904-f178-4ce9-956b-d9bfa4902745

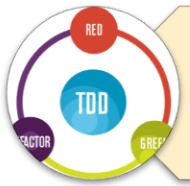
Integración de MBT y TDD



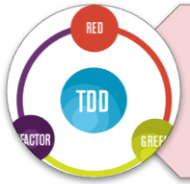
Desarrollo dirigido por tests.



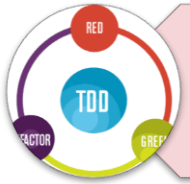
Automatización en la ejecución de tests.



Definición de tests por medio de código y palabras claves.



Muchas líneas de código para tests.



Cambio de requerimientos introduce problemas de mantenimiento.

MBT

Mayor abstracción de tests.

MBT

Seguimiento de los requerimientos del sistema.

MBT

Generación automática o semi-automática de tests.

MBT

Reducción de costos de mantenimiento.

MBT

Independencia de plataforma.

Estado del Arte

- Criterios para el análisis:
  - Adaptación del desarrollo de software a los pasos definidos por TDD
  - Relación de tests con requerimientos del software
  - Soporte de herramientas
  - Complejidad en la definición de tests
  - Niveles de automatización
  - Independencia entre modelos de test e implementación
  - Adecuadas para web testing

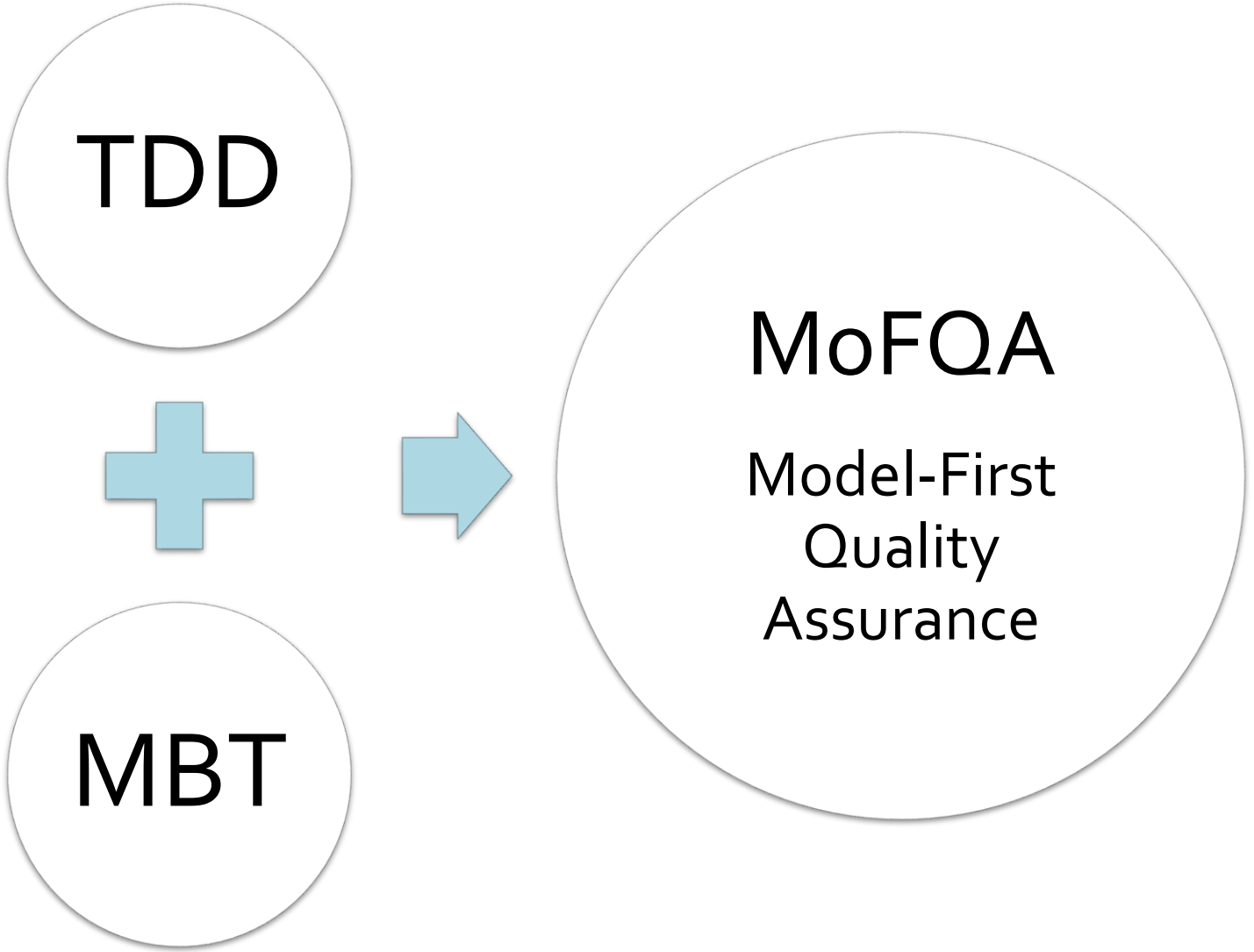
Estado del Arte

Título	Tipo de Informe	Año	Tipo de Estudio	Objeto de Estudio
Test First Model-Driven Development	Tesis de Msc. en Informática	2012	Presentación de una herramienta y validación mediante casos de estudio.	Se presenta la herramienta TFMDD que permite generar tests y software a partir de un modelado único, basado en restricciones a partir de la definición de pre y post condiciones.
Alignment of requirements specification and testing: A systematic mapping study	Artículo de Conferencia	2011	Mapeo sistemático, incluye el análisis de 35 trabajos seleccionados.	Analiza la relación que guarda el proceso de testing con respecto a las especificaciones de requerimientos funcionales y no funcionales del software.
A survey on model-based testing approaches: a systematic review	Artículo de Workshop	2007	Revisión sistemática de técnicas MBT, se analizan 78 trabajos seleccionados.	Se realiza comparaciones con respecto a aspectos como: modelos utilizados, soporte de herramientas, criterios de cobertura que soportan, niveles de automatización de pasos de testing, complejidad. Sus resultados permiten conocer qué tipos de tests están mayormente cubiertos y cuáles son algunas limitaciones pendientes en el área.
Industrial-strength model-based testing-state of the art and current challenges	Artículo de Workshop	2013	Estudio primario. Utilizando una herramienta existente (RT-Tester) como referencia, se describen aspectos de MBT aplicados en la industria.	Utilizando la herramienta RT-Tester, se ilustran aspectos de MBT en la práctica y métodos que dieron buenos resultados en testing aplicado a problemas del mundo real. El campo de aplicación considerado es: sistemas embebidos de tiempo real aplicados a la aviación, industria automotriz y vías férreas. Las técnicas y métodos presentados pueden ser utilizados como benchmarks para este campo de aplicación.
Towards Testing Future Web Applications	Artículo de Conferencia	2011	Se presenta una metodología que implementa nuevos paradigmas para el testing de webs futuras.	En el marco del proyecto FITTEST se realiza un análisis de las limitaciones de las técnicas de testing actuales para su aplicación sobre sistemas de plataforma web. Se presenta nuevos métodos y paradigmas que pueden ayudar a hacer frente a los problemas mencionados.
Integrating Model-Based Testing in Model-Driven Web Engineering	Artículo de Conferencia	2011	Estudio primario. Se presenta una técnica práctica para aplicar MBT, reutilizando modelos de desarrollo para la generación de tests.	Discusión sobre las ventajas y desventajas de la reutilización de modelos para generación de tests. Se presentan además detalles de web testing. Finalmente, se implementa una técnica para la aplicación de MBT mediante un ejemplo práctico.
Model-Based Testing of Community-Driven Open-Source GUI Applications	Artículo de Conferencia	2006	Estudio primario. Descripción de un proceso ideado para llevar a cabo el testing de aplicaciones elaboradas por una comunidad de desarrollo.	Se describen los problemas de testing en aplicaciones GUI, especialmente en comunidades de desarrollo. Se propone una técnica para llevar a cabo el testing de comunidades Open Source de desarrolladores, interconectados por la WWW.
Model Based Testing in Web Applications	Artículo de Journal	2014	Documento informativo, recopila varios trabajos y se analiza los modelos que utilizan para generar tests de aplicaciones web.	Discusión de conceptos principales y modelos utilizados para generar tests para aplicaciones web, se acompaña la discusión con un caso de estudio ilustrativo.

Estado del Arte

- Se destacan las limitaciones:
  - Las implementaciones existentes de MBT no toman ventaja completa de TDD.
  - Como un obstáculo para la implementación de MBT en la práctica, aparece la falta de herramientas integradas que faciliten los pasos envueltos en el proceso.
  - Se requiere minimizar los conocimientos necesarios para la implementación de técnicas actuales de MBT.
  - Es necesario generar tests más acordes a los requerimientos del sistema y no solo en base al funcionamiento del mismo.

Propuesta: MoFQA



Objetivos

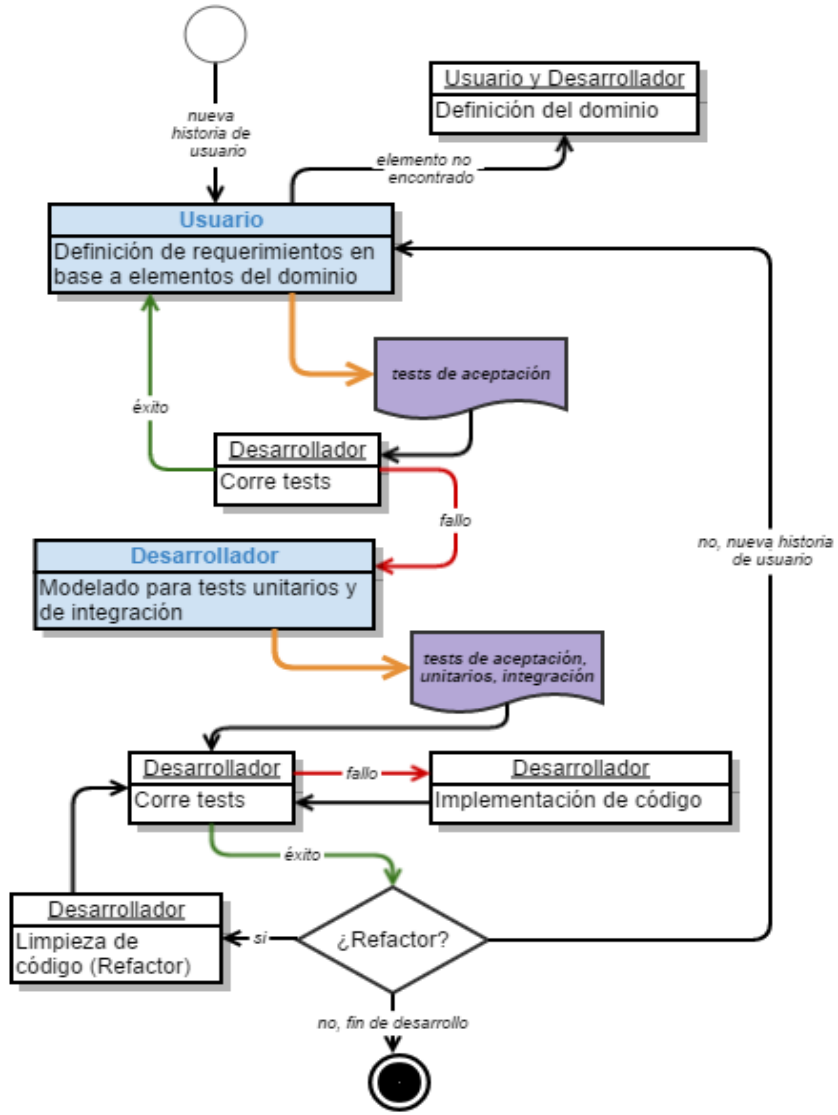
General

- Definición de un método de desarrollo de software basado en la definición de tests a partir de modelos definidos por el usuario y el desarrollador, siguiendo los pasos y prácticas descritos en TDD.

Específicos

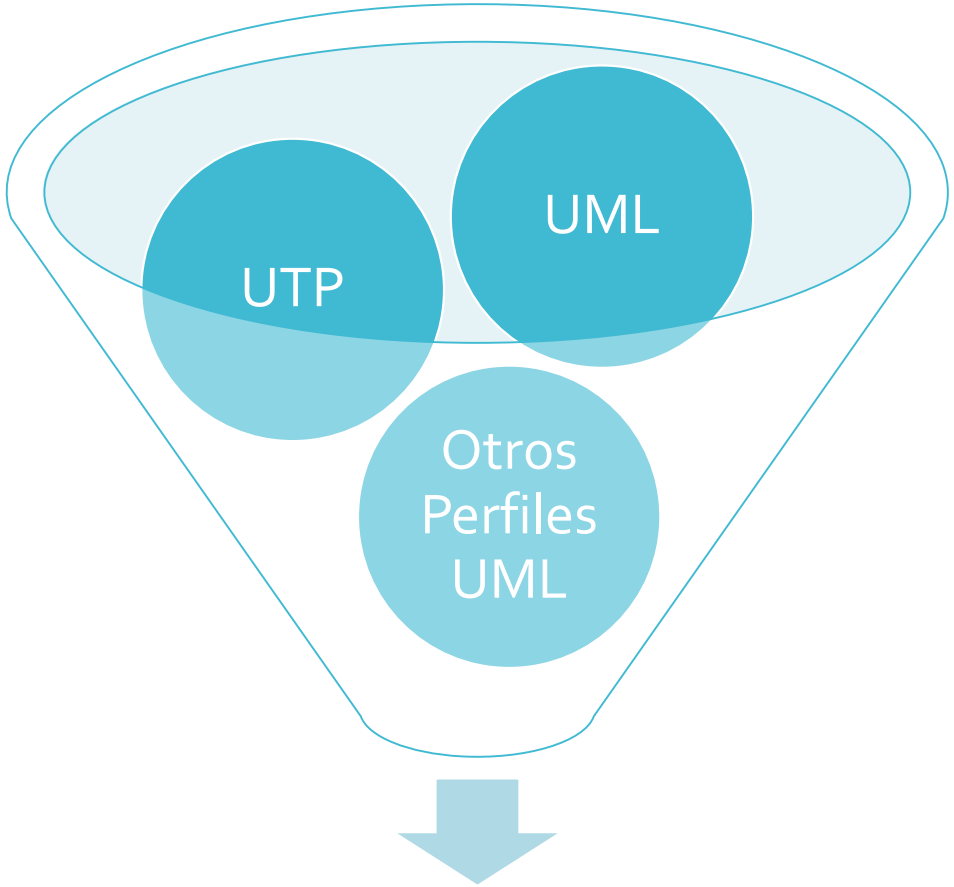
- Definición de un método de trabajo que integre los pasos de TDD, siguiendo las prácticas definidas por MBT.
- Enriquecimiento de perfiles UML para la definición de tests aplicables a sistemas de plataforma Web.
- Desarrollo de una herramienta integrada para el modelado, generación y ejecución de tests orientada a la plataforma definida para los propósitos de este trabajo.
- Validación de resultados.

Propuesta: MoFQA



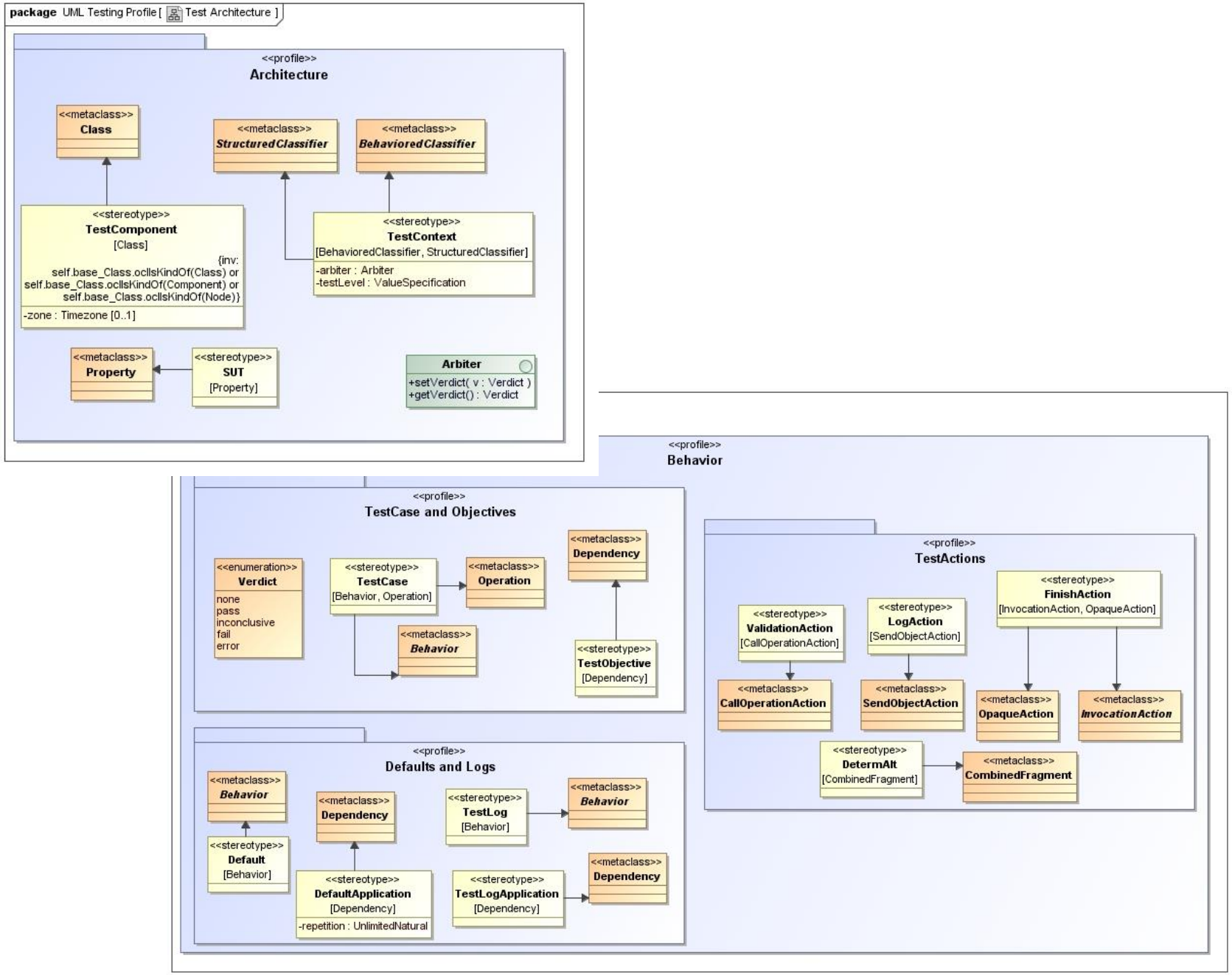
Flujo del desarrollo de software en MoFQA

Propuesta: MoFQA

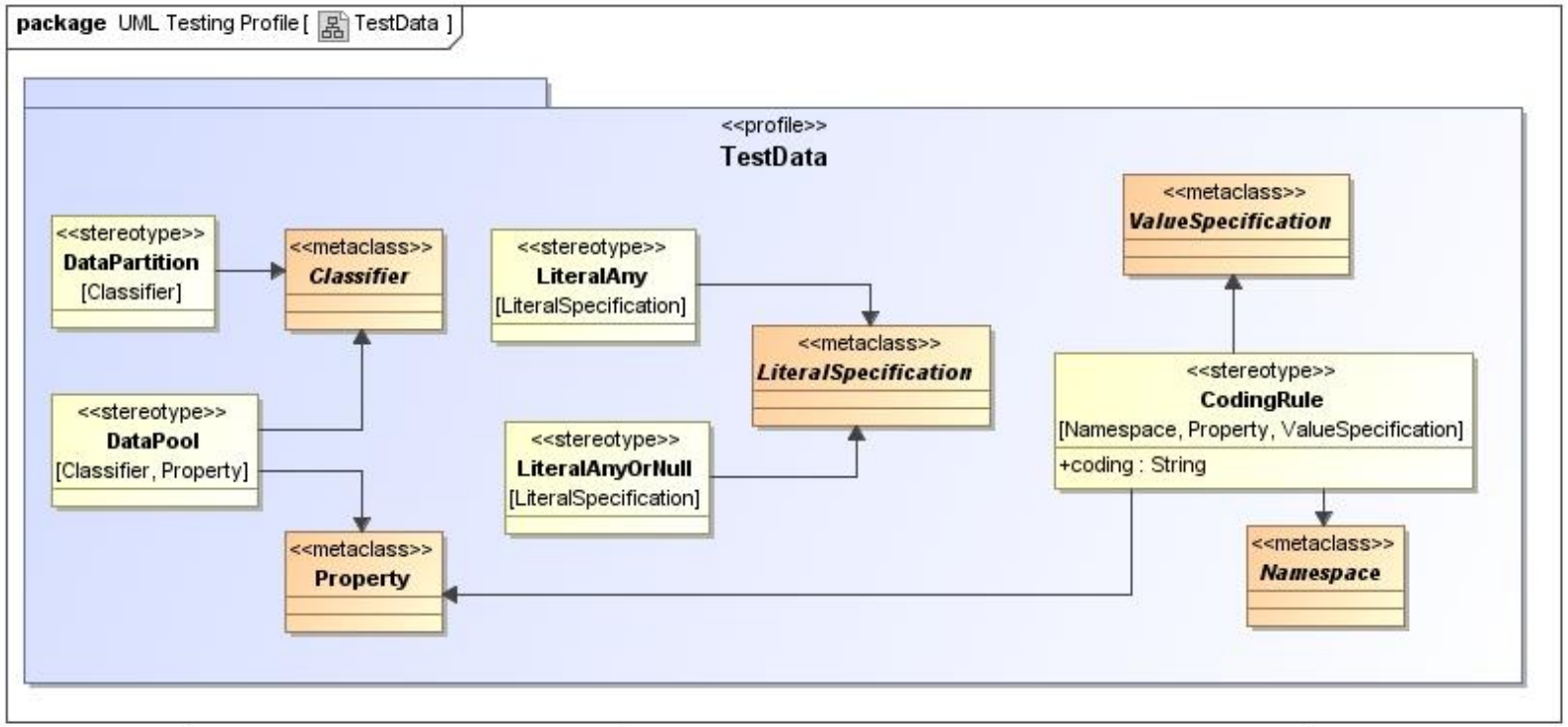


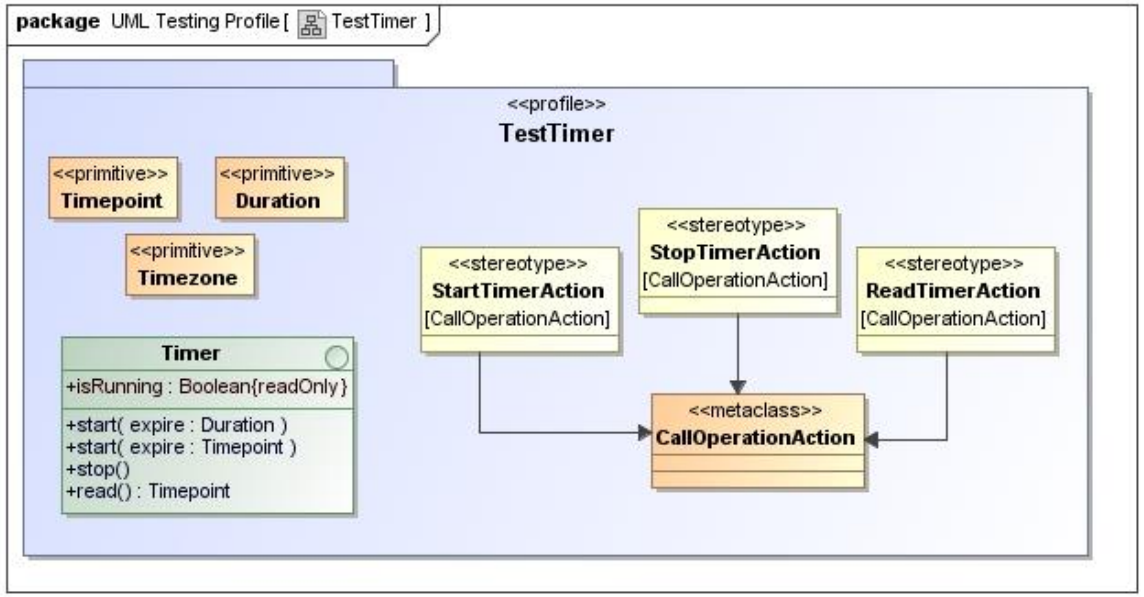
Modelos en MoFQA

UTP (UML Testing Profile)



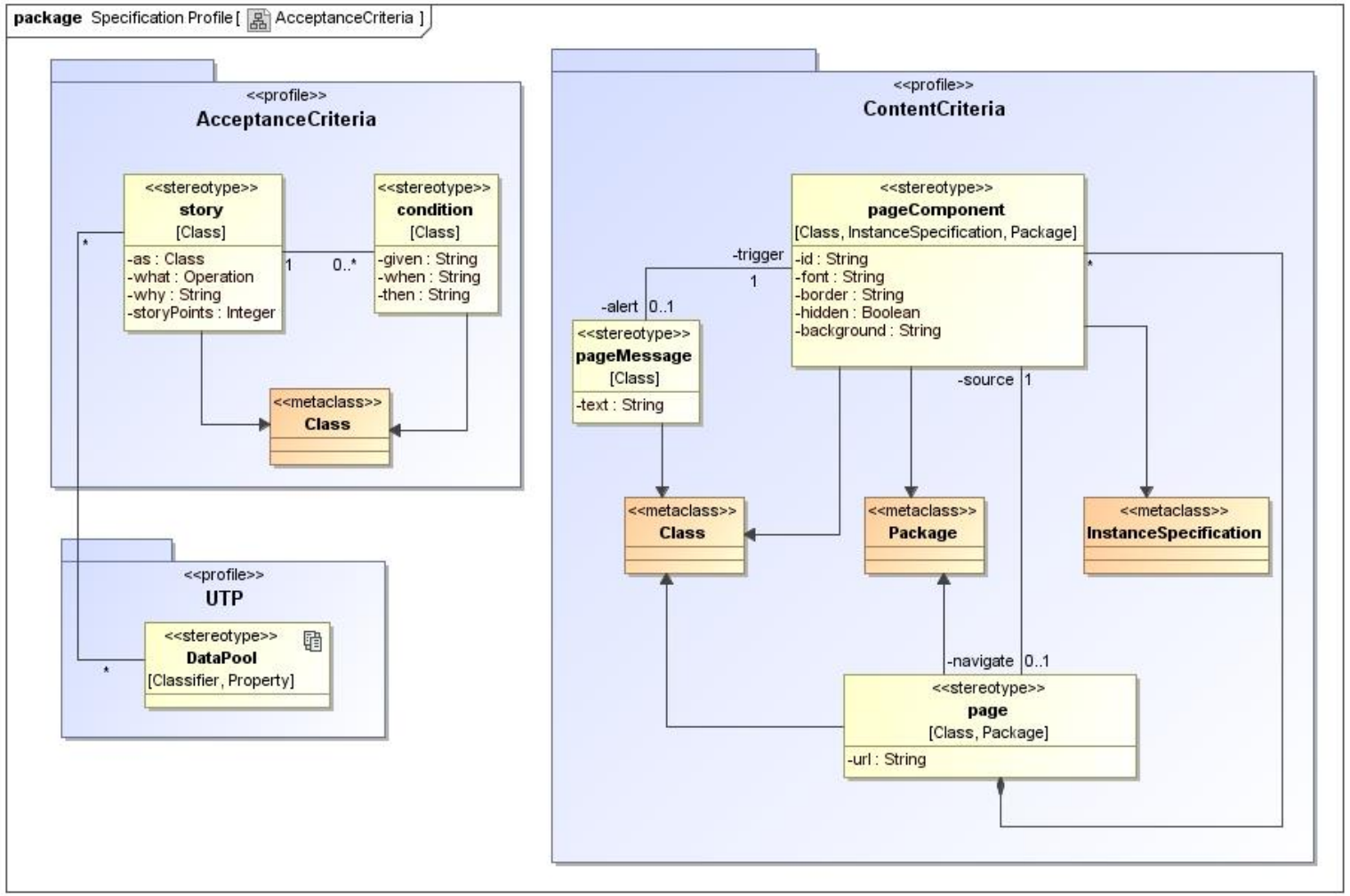
UTP (UML Testing Profile)





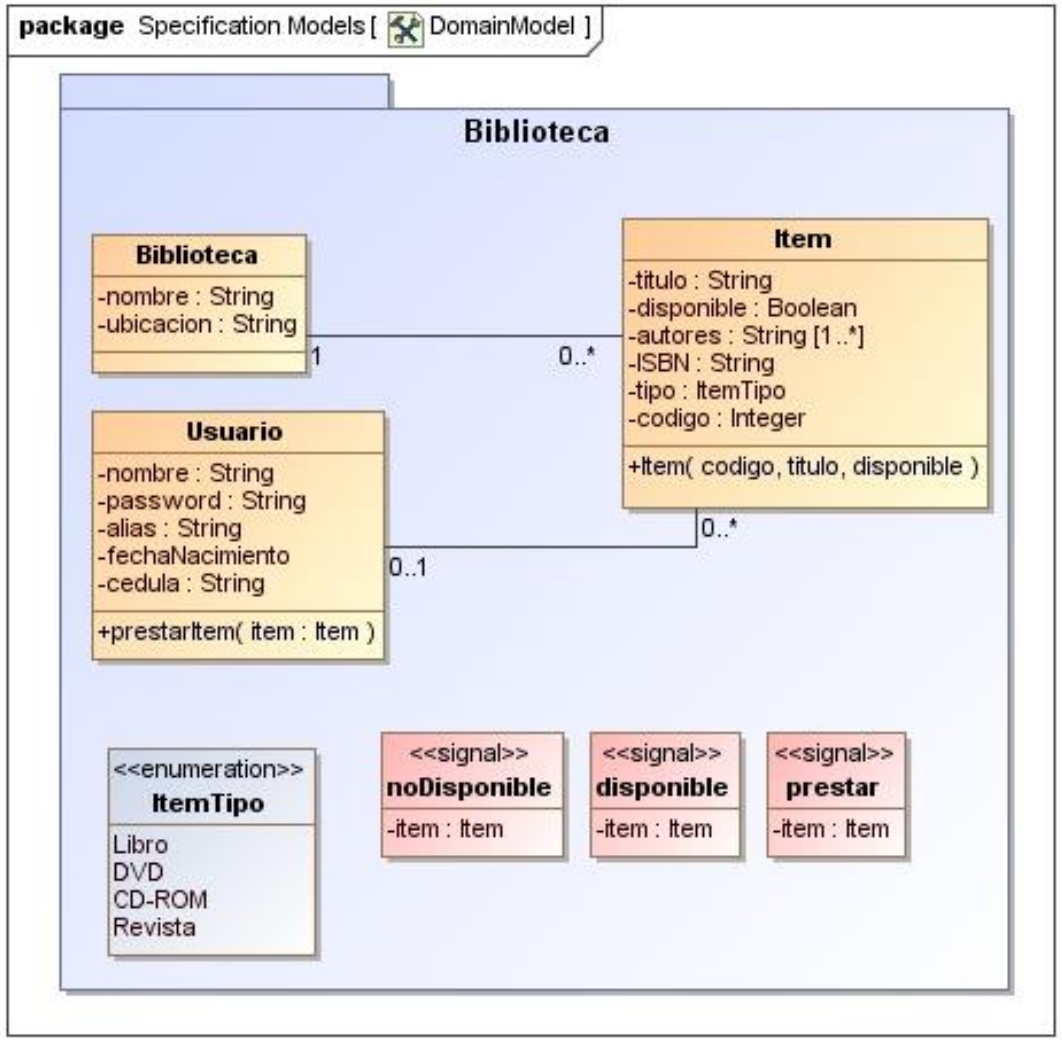
Propuesta MoFQA: Perfiles UML

Versión Inicial



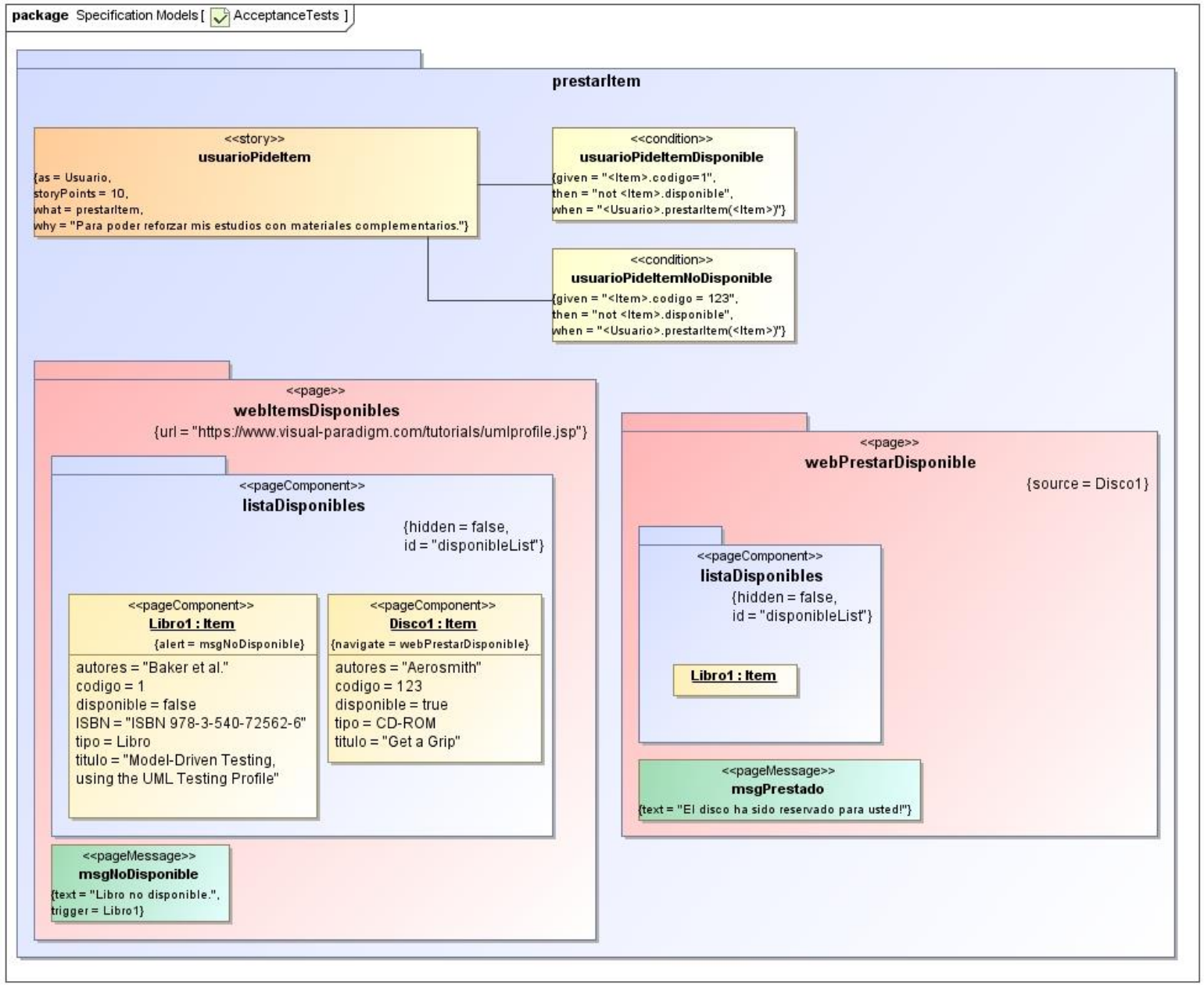
Ejemplo de Aplicación

Biblioteca Online



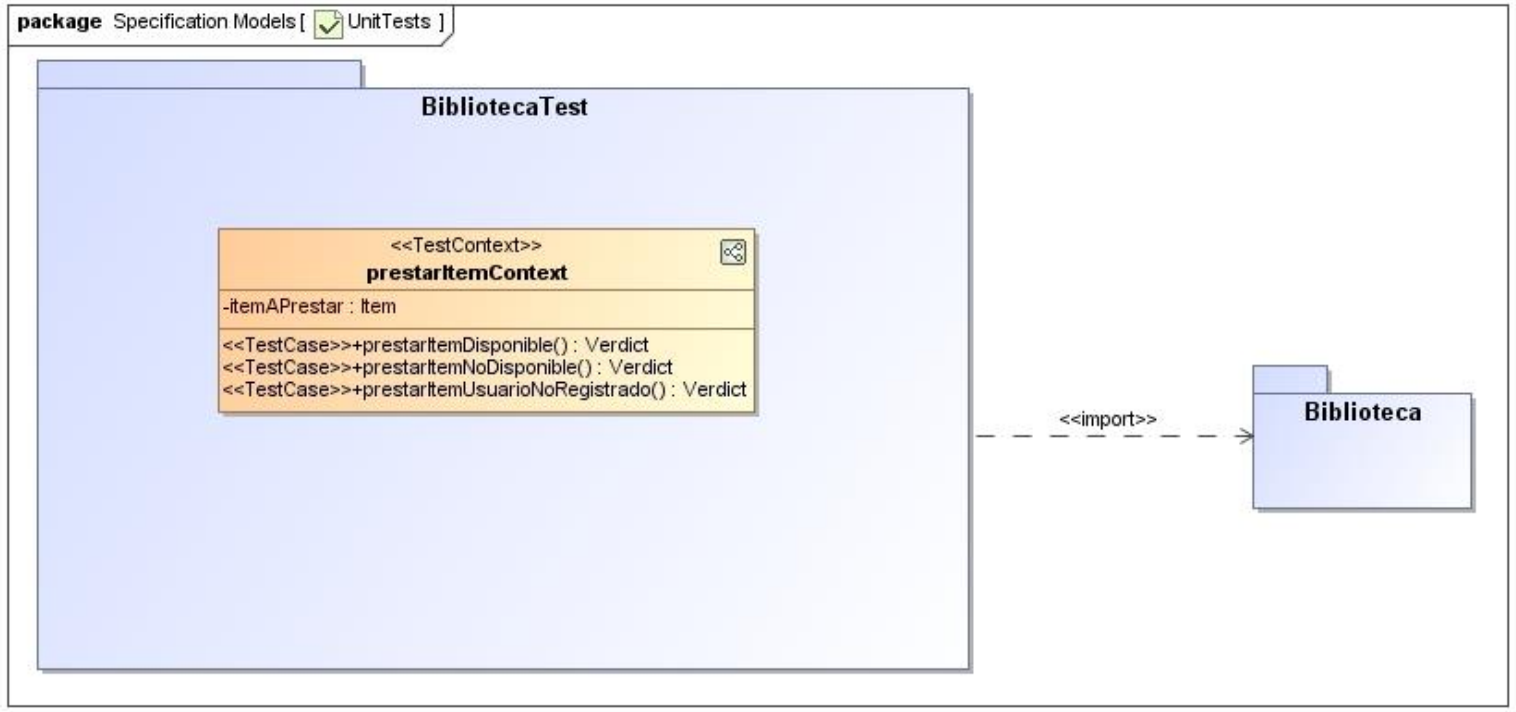
Ejemplo de Aplicación

Biblioteca Online

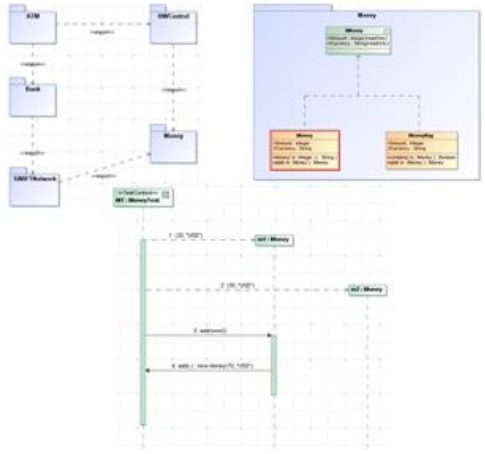


Ejemplo de Aplicación

Biblioteca Online



Prototipo



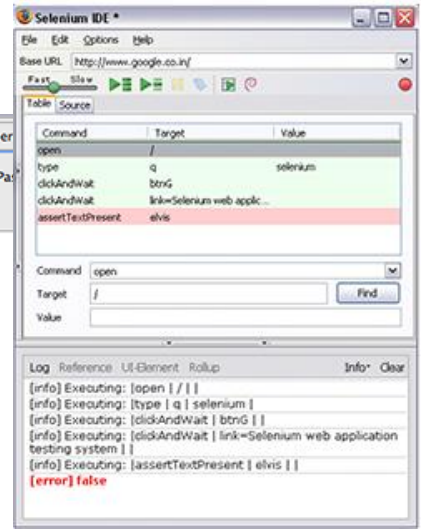
Transformación M2T

Acceleo

```
package example1;

import org.testng.annotations.*;

public class SimpleTest {

  @BeforeClass
  public void setUp() {
    // code that will be invok
  }

  @Test(groups = { "fast" })
  public void aFastTest() {
    System.out.println("Fast t
  }

  @Test(groups = { "slow" })
  public void aSlowTest() {
    System.out.println("Slow
  }
```

```
@Test(description="Launches the WordPress site")
public void launchSite(){
  selenium.open("");
  selenium.waitForPageToLoad("10000");
  assertEquals(selenium.getTitle(), "Demo | Just another WordPress

@Test(description="Navigates to the admin page")
  public void openAdminPage() {
  selenium.open("wp-admin");
  selenium.waitForPageToLoad("30000");
  assertEquals(selenium.getTitle(), "Demo â€º Log In");
  }

@Test(description="Enters valid login data")
  public void loginAsAdmin() {
  selenium.type("user_login", "admin");
  selenium.type("user_pass", "demo123");
  selenium.click("wp-submit");
  selenium.waitForPageToLoad("30000");
  assertTrue(selenium.isTextPresent("Howdy, admin"));
  }

@Test(description="Navigates to the New Post screen")
public void navigateNewPost() {
  selenium.click("//a[contains(text(),'Posts')]/following::a[conta
New"]][1]");
  selenium.waitForPageToLoad("30000");
  assertTrue(selenium.isTextPresent("Add New Post"));
  }
```

TestNG



Avances

Actividad	Estado
Recopilación de trabajos relacionados / Análisis del Estado del Arte	
Estudio de notaciones de modelado utilizadas para MBT	
Definición de perfiles UML	
Definición del Método MoFQA	
Estudio de las herramientas a utilizar para la implementación	
Definición de reglas de transformación	
Desarrollo de herramienta	
Validaciones	

¡Gracias por su atención!

¿Preguntas?

Linda Riquelme
linriquelme@gmail.com