

IIC 3745 - Testing

Entrega 2

2do semestre 2016

Testing FeedMe

Integrantes:

- Nicolás Campos
- Antonio Fontaine
- Agustín Gómez
- Fernando Lizana
- Diego Steinsapir

Índice

Software elegido.....	2
Tests realizados.....	2
Creando usuarios.....	2
Ingresando a la aplicación.....	3
Editar usuarios.....	3
Cursos.....	3
Planillas.....	3
Conclusiones.....	4

Software elegido

Para este proyecto se escogió realizar *testing* al *software* FeedMe utilizando Robot Framework. Este *software* fue realizado por tres de los cinco integrantes del grupo en el ramo Desarrollo de Software. FeedMe consiste en una aplicación web donde los profesores publican encuestas y los alumnos las responden con el objetivo de obtener *feedback* sobre las clases o actividades que se realizan en la sala.

Las principales funcionalidades de la aplicación son las siguientes:

- Alumnos y profesores pueden crear usuarios e ingresar a la aplicación web.
- Alumnos y profesores pueden editar sus perfiles luego de haberlos creados.
- Un profesor puede crear cursos.
- Un profesor puede crear planillas de preguntas.
- Un profesor puede enviar actividades (encuestas) a los alumnos inscritos en el curso.
- Un alumno puede buscar los cursos existentes, y enviar solicitudes para ser aceptado en algunos.
- Un profesor puede aceptar o rechazar las solicitudes de los alumnos.
- Un profesor puede visualizar las respuestas de las actividades, puede ver un informe general de las respuestas de todos los alumnos o ver las respuestas de un alumno específico.

Se realizó *testing* a varias de estas funcionalidades buscando cubrir de la mejor manera posible cada una de estas funcionalidades.

División de *tests*

Los tests se dividieron en 5 categorías: creando usuarios, ingresar a la aplicación, editar usuarios, planillas y cursos. Para las primeras tres categorías se utilizó cobertura de caso base y para las últimas dos cobertura de aristas.

Tests realizados

Estos fueron los *tests* que se desarrollaron para la aplicación, divididos en las categorías que les corresponden. Los resultados de estos *tests* también se pueden ver en el reporte autogenerado por Robot Framework.

Creando usuarios

Se utilizó una cobertura de caso base sobre el dominio de entrada. Para esto se identificaron las posibles entradas para crear usuarios que son las siguientes:

nombre, apellido, colegio, email, contraseña, y confirmación de contraseña. En la siguiente tabla se presenta la partición que se realizó para cada categoría, los valores destacados corresponden a los utilizados en el caso base.

Nombre	Apellido	Colegio	E-mail	Contraseña	Confirmación de contraseña
String	String	String	String con @ y punto	String válido	Igual a contraseña
Vacío	Vacío	Vacío	String sin @	String inválido	Distinto a contraseña
			String sin punto		
			Vacío		

Tabla 1: partición del dominio de entrada para crear usuarios.

Dada esta partición se realizaron los siguientes 9 *tests* de los cuales 7 pasaron.

1. Caso base: *test* pasado
2. Nombre vacío: *test* pasado
3. Apellido vacío: *test* pasado
4. Colegio vacío: *test* pasado
5. E-mail sin @: *test* pasado
6. E-mail sin punto: *test* fallado
7. E-mail vacío: *test* pasado
8. Contraseña inválida: *test* fallado
9. Confirmación no calza con contraseña: *test* pasado

Ingresando a la aplicación

Se utilizó una cobertura de caso base sobre el dominio de entrada. Para esto se identificaron las posibles entradas para ingresar a la aplicación que son las siguientes: e-mail y contraseña. En la siguiente tabla se presenta la partición que se realizó para cada categoría, los valores destacados corresponden a los utilizados en el caso base.

E-mail	Contraseña
Existe como usuario correcto (profesor si se está ingresando como profesor)	Correcta

Existe como usuario incorrecto (alumno si se está ingresando como profesor)	Incorrecta
No existe	

Tabla 2: partición del dominio de entrada para ingresar a la aplicación.

Dada esta partición se realizaron los siguientes 4 *tests* de los cuales todos pasaron.

1. Caso base
2. Usuario incorrecto (profesor trata de ingresar como alumno)
3. E-mail no existente
4. Contraseña incorrecta

Editar usuarios

Se utilizó una cobertura de caso base sobre el dominio de entrada. Para esto se identificaron las posibles entradas para editar usuarios que son las siguientes: nombre, apellido, colegio, email. En la siguiente tabla se presenta la partición que se realizó para cada categoría, los valores destacados corresponden a los utilizados en el caso base.

Nombre	Apellido	Colegio	E-mail
String	String	String	String con @ y punto
Vacío	Vacío	Vacío	String sin @
			String sin punto
			Vacío

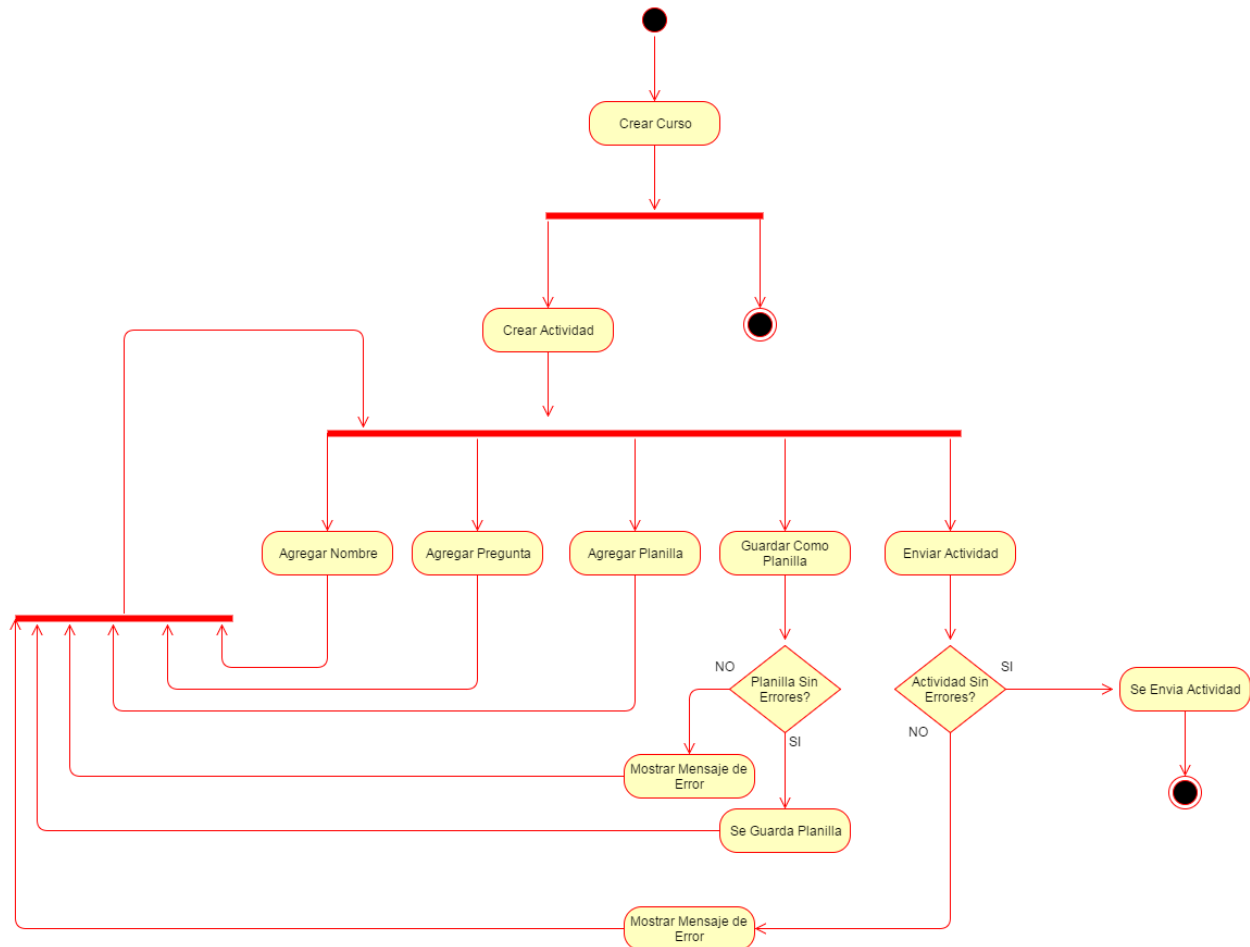
Tabla 3: partición del dominio de entrada para editar usuarios.

Se realizaron 7 *tests* y se pasó solamente 1.

1. Caso base, editar a información correcta: *test* pasado.
2. Editar a nombre vacío: *test* fallado.
3. Editar a apellido vacío: *test* fallado.
4. Editar a colegio vacío: *test* fallado.
5. Editar a e-mail sin @: *test* fallado.
6. Editar a e-mail sin punto: *test* fallado.
7. Editar a e-mail vacío: *test* fallado.

Cursos y Actividades

Para la sección de Cursos y Actividades se hizo cobertura de aristas. Para lograr esto se generó el siguiente diagrama:



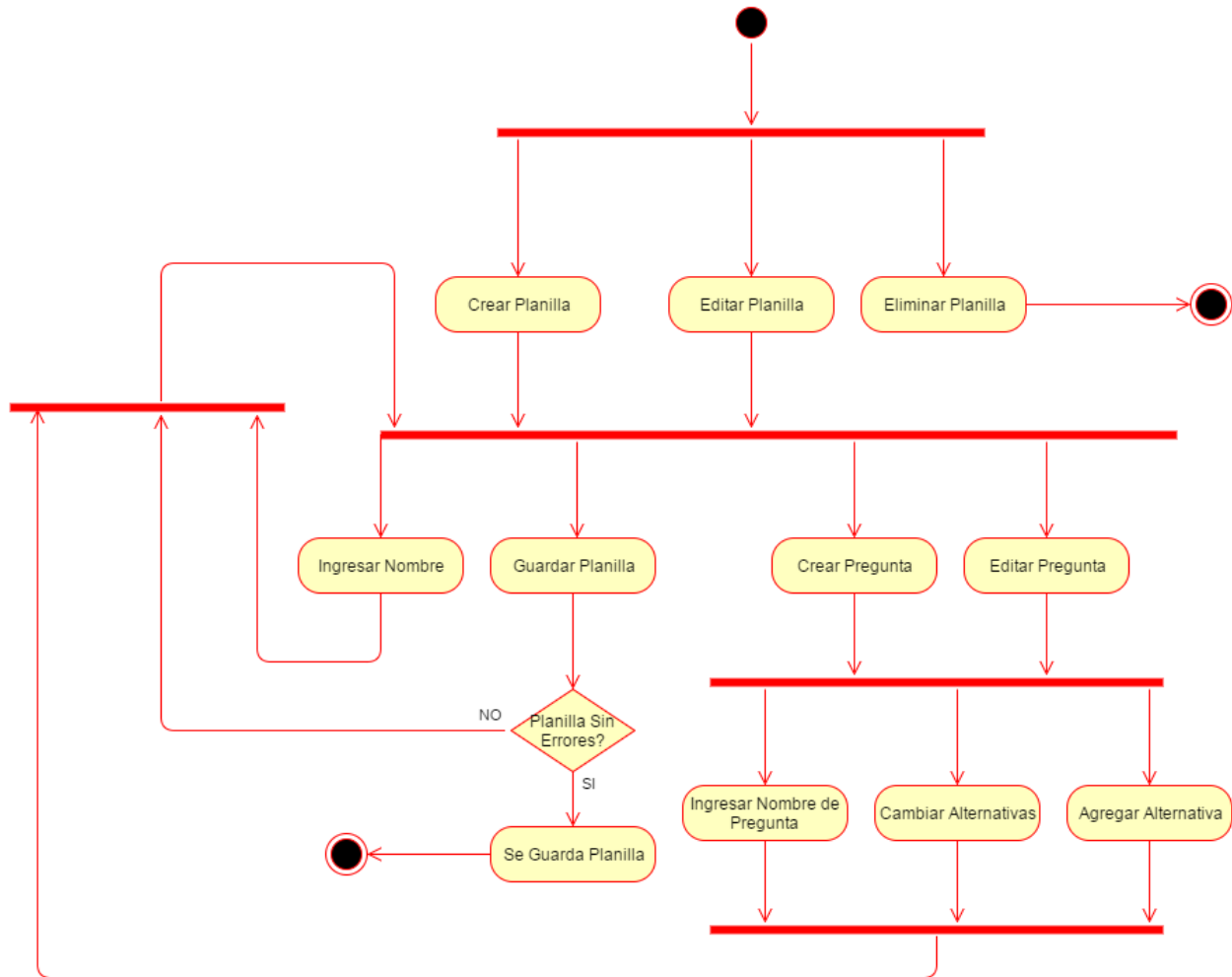
Se realizaron 12 *tests* y se pasaron todos.

1. Un profesor puede crear un curso.
2. Un profesor puede agregar una actividad a un curso.
3. Un profesor puede agregarle una pregunta de selección múltiple a la actividad creada.
4. Un profesor puede agregarle una pregunta de verdadero o falso a la actividad creada.
5. Un profesor puede agregarle una pregunta numérica a la actividad creada.
6. Un profesor puede agregarle una pregunta de respuesta corta a la actividad creada.
7. Un profesor puede agregarle una pregunta de repuesta larga a la actividad creada.
8. Un profesor puede agregar una planilla a la actividad creada.

9. Un profesor puede guardar una planilla desde la página de actividad.
10. Un profesor no puede guardar una planilla con errores.
11. Un profesor no puede enviar una actividad sin preguntas.
12. Un profesor no puede enviar una actividad sin título.

Planillas

Para planillas se hizo cobertura de aristas. Para lograr esto se generó el siguiente diagrama:



Se realizaron 10 *tests* y se pasaron todos.

1. Crear una planilla con todos los tipos de preguntas.
2. No poder crear preguntas de respuesta larga sin nombre.
3. No poder crear preguntas de alternativas con nombre o alternativas vacías.
4. No poder crear preguntas numéricas sin nombre.
5. No poder crear preguntas de respuesta corta sin nombre.

6. No poder crear preguntas de verdadero y falso con nombre o alternativas vacías.
7. No poder crear planillas sin nombre.
8. No poder crear planillas sin preguntas.
9. Crear una planilla luego de agregar una opción más.
10. Crear una planilla luego de editar alguna pregunta.
11. Eliminar una planilla.
12. Editar una planilla generando cambios al título y a alguna pregunta, editar nombre y opciones.

Conclusiones

A través de las pruebas realizadas se encontraron varios errores en la aplicación los cuales deberían ser solucionados por los desarrolladores. También se generó una cantidad de *tests* razonables que lograrían probar varias de las principales funcionalidades del *software* en caso de que se realicen cambios para asegurar que estas funcionalidades sigan cumpliendo su función de manera correcta.

De los 44 *tests* que se realizaron se pasaron 36 y fallaron 8, con un éxito de un 81,8%. Se detectaron los siguientes problemas con el código:

- Se debe validar la contraseña que ingresa el usuario, para certificar que esta tenga un largo mínimo y alguna otra característica que se deseé.
- Se debe aplicar validación al editar los perfiles de profesores y alumnos ya que como está el código se permiten cambios que no debiesen ser válidos.
- Se debe agregar la opción de cambiar la contraseña de los profesores y alumnos, y generar los *tests* para esto.

Para corregir estos errores, se debe utilizar una metodología de *Test Driven Development (TDD)* al mantener el código, escribiendo *tests* críticos previo a la modificación del *software*. En general, se le debe dar énfasis a la sección de editar perfiles, dado que la mayor cantidad de *tests* fallados provienen de ahí. Sin embargo, el *software* presenta una buena calidad en todas las otras áreas probadas.