

IIC 3745 - Testing

Entrega 1

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.

Tests realizados

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

Creando usuarios

Se realizaron 6 *tests*, de los cuales 5 fueron pasados.

1. Un profesor puede crear una cuenta y va a estar automáticamente ingresado: *test* pasado.
2. Un alumno puede crear una cuenta y va a estar automáticamente ingresado: *test* pasado.
3. No es posible crear un usuario con un *e-mail* ya utilizado: *test* pasado.
4. No es posible crear un usuario si la contraseña no coincide con la confirmación de contraseña: *test* pasado.
5. No es posible crear un usuario si no se ingresan todos los campos: *test* pasado.
6. No se puede crear un usuario con una contraseña inválida: *test* fallado.

Ingresando a la aplicación

Se realizaron 6 *tests* y se pasaron 5.

1. Un alumno puede ingresar con las credenciales correctas: *test* pasado.
2. Un alumno no puede ingresar como un profesor: *test* pasado.
3. Un profesor puede ingresar con las credenciales correctas: *test* pasado.
4. Un profesor no puede ingresar como un alumno: *test* pasado.
5. Un usuario no puede ingresar si no existe: *test* pasado.
6. Un usuario no puede ingresar con credenciales incorrectas: *test* fallado.

Editar usuarios

Se realizaron 8 *tests* y se pasaron 4.

1. Un alumno no puedo editar su información a vacío: *test* fallado.
2. Un alumno no puede editar su *e-mail* a uno inválido: *test* fallado.
3. Un alumno puede cancelar la edición de su información: *test* pasado.
4. Un alumno puede editar la información de su perfil: *test* pasado.
5. Un profesor no puedo editar su información a vacío: *test* fallado.
6. Un profesor no puede editar su *e-mail* a uno inválido: *test* fallado.
7. Un profesor puede cancelar la edición de su información: *test* pasado.
8. Un profesor puede editar la información de su perfil: *test* pasado.

Cursos

Se realizaron 7 *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 no puede agregar una actividad sin nombre.
4. Un profesor no puede agregar una actividad sin preguntas.
5. Un profesor puede crear una planilla a través de una actividad.
6. Un profesor puede crear una actividad utilizando una planilla.
7. Un alumno puede suscribirse a un curso

Planillas

Se realizaron 10 *tests* y se pasaron todos.

1. Permitir que una planilla tenga más de un tipo de pregunta.
2. No poder crear planillas sin nombre.
3. Crear una planilla con todos los tipos de preguntas.
4. Eliminar una planilla.
5. Editar una planilla.
6. No poder crear preguntas de respuesta larga sin nombre.
7. No poder crear preguntas de alternativas con nombre o alternativas vacías.
8. No poder crear preguntas numéricas sin nombre.
9. No poder crear preguntas de respuesta corta sin nombre.
10. No poder crear preguntas de verdadero y falso con nombre o alternativas vacías.

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 37 que se realizaron se pasaron 32 y fallaron 5, con un éxito de un 86,5%. 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 desee.
- 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.

