

Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación
CC4401 Ingeniería de Software



Informe de Requisitos

Aplicación Web de Apoyo Docente en Evaluaciones

Profesor de Cátedra: Jocelyn Simmonds

Profesor Auxiliar: Pablo Miranda Á.

Ayudantes: Daniela Chacón S.

María José Berger

Alumnos: Angel Huenchuleo

Bryan Riveros R.

Diego Ortego Prieto

Luis Pinochet G.

Grupo: BC & ML 007

Número: 7

Carrera: Ingeniería Civil

Especialidad: Computación

Fecha: 29 de Marzo 2019

1 Objetivos

Objetivo de Negocio

Facilitar el proceso docente de evaluación de presentaciones del curso, automatizando la mayor cantidad de pasos posible con un sistema que reemplace la actual metodología de corrección en papel.

Este objetivo está especificado explícitamente por el cliente en el documento de la sección 3.1, ” queremos que ustedes creen un sistema que permita gestionar las rúbricas de presentación de CC4401 [...] para así automatizar el cálculo de las notas de presentación ”.

Objetivo de Proyecto

Construir un sistema confiable, funcional y útil que pueda ser usado para evaluar presentaciones durante el semestre próximo.

Es decir, cumplir el deadline de terminar lo esencial del proyecto a más tardar a fines de este semestre.

Este objetivo está también especificado implícitamente en ambos documentos en las secciones 3.1 y 3.2, pues es sabido que los desarrolladores del sistema solo podrán trabajar en él durante el semestre académico actual, y, por lo tanto, debe ser terminado en ese tiempo.

2 Requisitos

2.1 Funcionales

2.1.1 Descripción

El sistema, esencialmente, debe permitir realizar las siguientes acciones:

1. Crear una única cuenta de profesor administrador.¹
2. El profesor administrador debe poder:
 - i. Crear rúbricas de evaluación, guardándolas en el sistema.²
 - ii. Editar rúbricas de evaluación guardadas.²
 - iii. Borrar rúbricas del sistema.²
 - iv. Cargar datos de alumnos al sistema en un formato Excel dado.³
 - v. Utilizar las rúbricas para evaluar a los alumnos, asignándoles una nota.²
3. El total máximo de puntos obtenible en una evaluación, detallado en las rúbricas, debe sumar siempre 6.0 puntos.⁴

El sistema debe, además, permitir los siguiente requisitos extras:

1. El profesor administrador podrá:
 - i. Crear cursos, con secciones, de alumnos.
 - ii. Modificar cursos, con secciones, de alumnos, incluyendo eliminar alumnos del curso.
 - iii. Borrar cursos, con secciones, de alumnos.
 - iv. Crear grupos de trabajo de alumnos.

¹Fuente en el documento de la sección 3.2, preguntas 17 y 32

²Fuente en el documento de la sección 3.1, "queremos que ustedes creen un sistema que permita gestionar las rúbricas de presentación de CC4401(crear, modificar y borrar rúbricas), como también aplicarlas durante las presentaciones ..."

³Fuente en el documento de la sección 3.1, " debe permitir que la profesora cree una sección del curso y cargue los datos de los alumnos (mediante un archivo Excel en un formato estándar). " El formato Excel está especificado con el siguiente [ejemplo](#).

⁴Fuente en el documento sección 3.2, pregunta 7

- v. Modificar grupos de trabajo de alumnos, desplazando estudiantes de un grupo a otro, y respetando si se eliminan alumnos del curso.
 - vi. Borrar grupos de trabajo de alumnos.
 - vii. Crear cuentas para evaluadores, sin poder administrativo.
 - viii. Asignar una rúbrica, de un grupo de alumnos por evaluar, a un grupo de evaluadores.
2. Mantener registros históricos de las modificaciones de secciones, grupos y evaluaciones.⁵

⁵Fuente en el documento de la sección [3.2](#), pregunta 21

2.1.2 Tests Propuestos

Los requisitos anteriores se pueden testear de la siguiente forma:

- 1 Para testear el requerimiento 1. se puede crear una primera cuenta de administrador con un servidor web fresco, y luego intentar crear otra. El test será exitoso si se lanza una notificación al respecto, y no se logra crear la segunda cuenta administrativa.
- 2 Para testear el requerimiento 2i. se puede crear una rúbrica simple a partir de rúbricas en papel usadas anteriormente (antes del sistema), y ver si se guarda correctamente.
- 3 Para testear el requerimiento 2ii. se puede crear de manera ficticia alguna rúbrica y luego editarla. El test será exitoso si el sistema logra preservar los cambios.
- 4 Para testear el requerimiento 2iii. se puede crear una rúbrica ficticia y luego eliminarla. El test será exitoso si no se deja rastro visible en el sistema de esa rúbrica.
- 5 Para testear el requerimiento 2iv. se puede intentar cargar datos al sistema de algún grupo de alumnos ficticio o real. El test será exitoso si los datos que se guarden en el sistema coinciden con los datos originales y perduran más que la sesión local.
- 6 Para testear el requerimiento 2v. se puede ingresar datos ficticios o anteriores al sistema a una rúbrica de prueba, simulando una evaluación, para algunos alumnos. El test será exitoso si cada alumno tiene una nota final y única.
- 7 Para testear el requerimiento 3. se puede intentar crear una rúbrica e ingresar valores tales que la suma total de puntaje sea mayor o menor, pero no igual que 6.0. Para que el test sea exitoso el sistema lanzará una notificación de advertencia sin guardar la rúbrica.

Además se debe verificar que cuando sí se ingresa una suma total de 6.0 la rúbrica se guarda.

2.2 No-Funcionales

2.2.1 Descripción

1. El sistema debe ser fácil de usar, todo se debe poder realizar en pocos clicks / taps.⁶
2. El sistema debe ser confiable, debe permitir su reusabilidad durante varios semestres.⁷

2.2.2 Métricas Propuestas

1. Como forma de medir la facilidad de uso de la aplicación se propone hacer una demostración de su funcionamiento a un grupo de personas.

Luego se observa con cuanta facilidad y velocidad son capaces de realizar algunas de las acciones que se detallan en los casos de uso de la aplicación.

Pueden medirse: tiempo de demora total en cada caso de uso y cantidad de clicks/taps realizados, además de los segundos perdidos navegando la interfaz sin ningún avance en la tarea a mano.

A fin de mejorar la calidad de la métrica se puede realizar el experimento de pedirle al grupo de personas que intente realizar lo mismo de nuevo unas semanas después, sin repasar previamente, y midiendo las mismas métricas.

La combinación de ambos experimentos puede ser un mejor indicador de la facilidad de uso de la aplicación.

2. Para medir la reusabilidad, se propone crear un grupo ficticio de alumnos.

Luego se ve qué tan fácil es guardar y borrar los datos e iniciar con un nuevo curso con nuevas rúbricas.

Pueden medirse:

Tiempo de respuesta del programa al guardar una sección y abrir otra.

Cantidad de memoria usada cuando hay varias secciones guardadas a la vez, cada una con datos de evaluaciones.

Capacidad máxima de secciones que permite el sistema.

⁶Fuente en el documento de la sección 3.1 (En general se dice que la idea es automatizar y facilitar el proceso de evaluación), y en el de la sección 3.2, pregunta 4.

⁷Fuente en el documento de la sección 3.1, "El sistema será usado durante varios meses, así que debe permitir que la profesora cree una sección del curso y cargue los datos de los alumnos ..."

Tiempo de respuesta del programa al borrar secciones.

Si efectivamente se están borrando los datos de las secciones borradas, ahorrado memoria.

Estas mediciones pueden hacerse con la ayuda de un usuario que ejecute los pasos mencionados e indique si hay pasos redundantes al ingresar un nuevo curso, mientras un administrados observa las variables del sistema, como la memoria usada.

3 Fuentes

Los objetivos y requisitos detallados anteriormente provienen de las siguientes fuentes:

3.1 Documento de Especificaciones Inicial

Este documento describe de forma general el problema y algunos de los requerimientos básicos que se deben implementar para que el proyecto se considere exitoso.

- [Link al documento.](#)

3.2 Consultas al Cliente

Este documento contiene preguntas que fueron realizadas al cliente a fin de aclarar más específicamente qué requerimientos tomar en cuenta durante el desarrollo.

- [Link al documento.](#)

Casos de uso

