

UNIVERSIDAD DE TARAPACÁ



ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL,
INFORMÁTICA Y DE SISTEMAS

EUIIIS

Área de Ingeniería en Computación e Informática



ACTUALIZACIÓN CRSoq 2.0

Autores: Jorge Bernal Navarrete
Douglas Gómez Jiménez
Rodrigo Salazar Ortega
Alejandra Sobarzo Taucare
Felipe Yáñez Pérez

Curso: Proyecto III

Profesor: Ricardo Valdivia Pinto

ARICA, 15 de diciembre de 2017

RESUMEN

El presente informe tiene como objetivo presentar el trabajo realizado en torno al desarrollo de la actualización, migración y documentación del sistema CRSoq.

En este se presenta el estado final al que llegó el proyecto luego del trabajo realizado, aquí se detalla cómo se cumplió cada uno de los requerimientos o por qué algunos no se cumplieron, además se presenta el principal problema que sufrió el desarrollo para futuras referencias. Finalmente se incluyen ideas a futuro que surgieron durante el proceso.

Con respecto a la especificación de requisitos, se presenta la lista de requerimientos determinados por el equipo de desarrollo en conjunto con el cliente, son estos los requerimientos que se detallan a través de todo el documento.

Si bien algunos de los requerimientos no fueron cumplidos, el principal problema fue el retraso del proyecto al resolver problemas o analizar el código, aún con esto en cuenta, se considera que las actualizaciones realizadas fueron las más importantes, pero sin lograr realizar las pruebas pertinentes lo que requerirá de un futuro análisis.

Contenido

RESUMEN	2
I. INTRODUCCIÓN	5
II. REQUERIMIENTOS.....	6
III. MIGRACIÓN AICI.....	8
3.1. El Problema.....	8
3.2. Migración del Servidor	8
3.3. Problemas de Versión de MySQL	8
IV. TRASPASO A GITHUB	9
4.1. Git.....	9
4.2. GitHub.....	9
V. ACTUALIZACIÓN.....	10
5.1. Estado Final Requisitos Funcionales.....	10
5.2. Descripción de Resolución de Requisitos	11
5.2.1. Req01	11
5.2.2. Req02	12
5.2.3. Req03	16
5.2.4. Req04	17
5.2.5. Req05	18
5.2.6. Req06	20
5.2.7. Req07	20
5.2.8. Req08	21
5.2.9. Req09	21
5.2.10. Req10	23
VI. DOCUMENTACIÓN.....	23
VII. PROBLEMAS CON EL DESARROLLO	24
VIII. REQUERIMIENTOS NO IMPLEMENTADOS.....	25
8.1. Cambio de Bower	25
8.2. Manejo de Versiones de Redmine.....	25

8.3. Cambio de Mecánica de las Preguntas	25
IX. IDEAS A FUTURO	25
9.1. Teoría de Colores	25
9.2. Dificultad de Preguntas.....	25
9.3. Problemas de Gráfico	25
X. CONCLUSIONES.....	27

Índice de Ilustraciones

Ilustración 1. Vista de configuración de asignatura	15
Ilustración 2. Filtrado de preguntas	15
Ilustración 3. Gráfico resultado de preguntas realizadas por clase.....	16
Ilustración 4. Resumen de estudiante.....	18
Ilustración 5. Indicadores de estadística general	20
Ilustración 6. Encabezado para el ordenamiento de participantes	21
Ilustración 7. Vista de alumnos del administrador.....	23
Ilustración 8. Modelo de datos	24

Índice de Tablas

Tabla 1. Requisitos funcionales	6
Tabla 2. Requisitos prescindibles	7
Tabla 3. Requisitos no funcionales	7
Tabla 4. Estado de final de los requisitos	10
Tabla 5. Consultas creadas Req02	13

I. INTRODUCCIÓN

La tecnología sigue avanzando y con ella todos los sistemas deben avanzar también, este avance se ve reflejado en las actualizaciones que sufren los sistemas, no solo con respecto a la incorporación de nuevas tecnologías sino también con respecto a mejoras o la solución de problemas.

El sistema CRSOq es un sistema de apoyo al profesor en la implementación de la dinámica CRSOq que busca que el estudiante sea participe del proceso educativo, entre ello con el fin de generar un desarrollo o fortalecimiento de habilidades blandas, como por ejemplo la capacidad de reflexión y crítica, o sus habilidades comunicativas.

Para esto se proponía el uso de preguntas cerradas para que el estudiante debiese expresarse a la clase y al profesor, valiéndose de sus propios medios para desarrollar una respuesta coherente y fundamentada a las preguntas que pudiesen surgir.

Esto no era posible lograr debido a la poca motivación de los estudiantes, por lo que la propuesta consta de otorgar puntos a los estudiantes por las preguntas respondidas correctamente hasta un máximo establecido por el profesor, reflejado en una calificación para la asignatura.

El sistema CRSOq ha sufrido una serie de cambios que lo han llevado a la versión 2.0, estos cambios fueron realizados por el equipo conformado por estudiantes de carrera de Ingeniería en Informática de la Universidad de Tarapacá para el curso de Proyecto III.

La actualización de un sistema y su re-documentación es un tema bastante complicado para un equipo diferente del de su desarrollo inicial, no solo al no estar familiarizado con el código, sino también porque los estándares que maneja cada alumno tienden a diferir lo que provoca que este se tenga que adaptar a la nueva metodología.

En este caso, los cambios realizados en la documentación realizada para satisfacer los requisitos de actualización se vieron reflejados a nivel de código, entre las que se encuentran el añadido de nuevas clases al modelo de datos, para permitir el filtrado de preguntas, el ordenamiento de participantes, el añadido y removido de gráficos de información, entre otros.

II. REQUERIMIENTOS

Para la actualización del sistema, se determinaron los siguientes requisitos funcionales (Tabla 1), prescindibles (Tabla 2) y no funcionales (Tabla 3), estos requisitos fueron aprobados por el cliente para su implementación en el sistema.

Tabla 1. Requisitos funcionales

Requerimiento	Descripción
Req01	Las preguntas no deben ser vistas por los alumnos antes de ser realizadas.
Req02	Organización en las preguntas. Debe existir un filtro para agrupar las preguntas por categorías y también por módulo/tema.
Req03	Los indicadores de rendimiento por curso que deben mejorarse son: <ul style="list-style-type: none"> • Participantes por pregunta: quitar. • Resultado de preguntas realizadas por clase: cambiar por gráfico de barras. • Preguntas realizadas vs participación por clase: quitar gráfico minimizado. • Participación por estudiante en preguntas: quitar. • Puntos ganados por estudiantes: quitar. • Participación de estudiante por clase: debe estar seleccionado solo el primer alumno o ninguno. Quitar gráfico minimizado.
Req04	Se creará un nuevo indicador llamado “estadísticas de alumno”, el cual incluirá un filtro por alumno y contendrá los siguientes gráficos: <ul style="list-style-type: none"> • Gráfico circular que muestra el total de respuestas correctas e incorrectas sobre el total de preguntas participadas. • Gráfico circular que muestra la participación sobre el total de preguntas realizadas. • Gráfico circular que muestra el total de preguntas seleccionadas y no seleccionadas sobre el total de preguntas participadas.
Req05	Se creará un nuevo indicador de rendimiento por asignatura llamado “estadísticas generales”, el cual contendrá los siguientes gráficos: <ul style="list-style-type: none"> • Total de participación de las preguntas realizadas. • Ganadores.

Req06	Los indicadores de rendimiento por asignatura que deben mejorarse son: <ul style="list-style-type: none"> • % de participación por pregunta: debe mostrar las preguntas erróneas por semestre y la cantidad de veces que se repitieron los errores. Nota: El profesor quiere ver una muestra antes de decidir.
Req07	Cuando el profesor recibe la lista de participantes debe poder filtrar por: <ul style="list-style-type: none"> • Número de preguntas correctas. • Número de preguntas incorrectas. • Número de preguntas participadas. • Número de preguntas no participadas. • Orden de llegada.
Req08	El profesor debe poder cambiar su contraseña.
Req09	El administrador debe contar con una nueva vista, para poder gestionar a los alumnos del sistema.
Req10	Actualizar gráfico “preguntas realizadas vs participación por clase”

Tabla 2. Requisitos prescindibles

Requerimiento	Descripción
Req11	La creación de una pregunta debe permitir: <ul style="list-style-type: none"> • Utilizar diferentes tipos de letra (fuentes). • Cambiar colores. • Adjuntar imágenes.

Tabla 3. Requisitos no funcionales

Requerimiento	Descripción
NF01	Migrar el proyecto a un servidor del AICI
NF02	Administrar el proyecto como un proyecto open source: <ul style="list-style-type: none"> • Gestionar el proyecto en las plataformas RedMine y GitHub (repositorio) • Manejar un control de versiones (Git, SVN, etc.)
NF03	El sistema debe prescindir del uso de Bower como herramienta de instalación de dependencias.

III. MIGRACIÓN AICI

Uno de los requerimientos no funcionales era la migración del sistema a uno de los nuevos servidores presentes en el Departamento de Computación e Informática de la Universidad de Tarapacá, lugar en el cual fue implementado el sistema para su uso en diversas asignaturas.

3.1. El Problema

Anteriormente el sistema CRS_{oq} funcionaba en un computador que cumplía la función de servidor no dedicado, esto generaba incertidumbre al momento de saber cuándo iba a fallar o no el sistema, resultando en una característica necesaria de mejorar.

3.2. Migración del Servidor

Para solucionar los problemas y preocupaciones referentes a la situación presentada en el punto anterior, el sistema fue migrado a un servidor manteniendo la configuración del puerto de conexión, siendo 146.83.109.226:3000 la nueva dirección de acceso al sistema.

Se procuró mantener los datos del sistema, para de esta forma mantener información importante como las preguntas, las cuentas de los alumnos y los profesores y también datos estadísticos para mostrar en la información.

Debido al cambio realizado en la biblioteca de preguntas, estas fueron quitadas, manteniendo las preguntas en los cursos para que pudiesen ser archivadas posteriormente de ser necesario y agregar así el tema de cada pregunta.

3.3. Problemas de Versión de MySQL

Al realizar la migración del sistema, ocurrieron ciertos problemas cuyas soluciones cabe mencionar debido a su posible necesidad al momento de implementar el sistema en otros servidores.

El principal problema corresponde al fallo de truncado de valores de tipo *date*, estos valores son manejados por NodeJS en un formato que difiere del utilizado por MySQL, lo que produce una excepción al momento de insertar filas que contengan atributos de este tipo. Este error solo existe en versiones actuales de MySQL pues desde la versión 5.7.5 en adelante, el modo *STRICT_TRANS_TABLES* está presente por defecto en la lista de *sql_modes*.

Para solucionar este problema fue necesaria la modificación de la consulta referente a la creación de nuevas clases donde se utilizaba la fecha. La modificación realizada consta del uso de la función *STR_TO_DATE()*, que funciona

de forma similar a la función *TO_DATE()* de Oracle, tomando el formato de fecha manejado por Angular (%Y-%m-%dT%H:%i:%s.%fZ) y aplicándolo a la función para no necesitar de la conversión automática.

Otro problema surge de la misma actualización de MySQL, en este caso, con el modo *ONLY_FULL_GROUP_BY*, que afecta a consultas referentes a los gráficos de información, evitando la obtención de datos. Para resolver este problema es necesario desactivar este modo.

IV. TRASPASO A GITHUB

Otro de los requerimientos no funcionales es el traspaso del sistema a la plataforma de administración de proyectos GitHub, aplicando la herramienta de manejo de versiones Git.

4.1. Git

Git es un sistema de control de versiones de código abierto que permite la administración de proyectos de todo tipo, este proporciona funciones como el uso de ramas de desarrollo que permiten la modificación del código del sistema sin modificar el sistema principal permitiendo así poder regresar en cualquier momento a un estado seguro del sistema.

El sistema ya poseía los archivos de la herramienta *git* debido a su utilización por parte del anterior encargado del sistema, pero debido a la falta de contraseñas para la utilización de BitBucket como repositorio remoto, se eliminó de los archivos de Git por medio del comando “`git remote rm`”. Tras esto se agregó el repositorio de GitHub del proyecto CRSoq.

4.2. GitHub

GitHub es una plataforma de desarrollo que permite la administración de proyectos en conjunto con la comunidad, permitiendo la interacción entre desarrolladores. Esta plataforma tiene la opción de un desarrollo privado o público para el proyecto. En el caso del sistema CRSoq, se busca que este sea utilizado y mejorado por la comunidad abiertamente por lo que se optó por un desarrollo público.

El código del sistema puede ser obtenido a través del comando:

```
git clone https://github.com/CRSoq/CRSoq.git
```

Entre los archivos del sistema, se encuentra el script de SQL para la creación de la base de datos.

V. ACTUALIZACIÓN

En el proceso de actualización se trataron de resolver los requisitos funcionales que solicitaba el cliente.

5.1. Estado Final Requisitos Funcionales

Tras la realización del proyecto se logró cumplir con la mayoría de los requerimientos solicitados por el cliente, detallándose en la Tabla 4 el estado final de los mismos.

Tabla 4. Estado de final de los requisitos

Requerimiento	Descripción	Estado
Req01	Las preguntas no deben ser vistas por los alumnos antes de ser realizadas.	Realizado
Req02	Organización en las preguntas. Debe existir un filtro para agrupar las preguntas por categorías y también por módulo/tema.	Realizado
Req03	Los indicadores de rendimiento por curso que deben mejorarse son: <ol style="list-style-type: none"> 1. <u>Participantes por pregunta</u>: quitar. 2. <u>Resultado de preguntas realizadas por clase</u>: cambiar por gráfico de barras. 3. <u>Preguntas realizadas vs participación por clase</u>: quitar gráfico minimizado. 4. <u>Participación por estudiante en preguntas</u>: quitar. 5. <u>Puntos ganados por estudiantes</u>: quitar. 6. <u>Participación de estudiante por clase</u>: debe estar seleccionado solo el primer alumno o ninguno. Quitar gráfico minimizado. 	1. Realizado 2. Realizado 3. No posible 4. Realizado 5. Realizado 6. No posible
Req04	Se creará un nuevo indicador llamado “estadísticas de alumno”, el cual incluirá un filtro por alumno y contendrá los siguientes gráficos: <ul style="list-style-type: none"> • <u>Gráfico circular</u> que muestra el total de respuestas correctas e incorrectas sobre el total de preguntas participadas. • <u>Gráfico circular</u> que muestra la participación sobre el total de preguntas realizadas. • <u>Gráfico circular</u> que muestra el total de preguntas seleccionadas y no seleccionadas sobre el total de preguntas participadas. 	Realizado

Req05	Se creará un nuevo indicador de rendimiento por asignatura llamado “estadísticas generales”, el cual contendrá los siguientes gráficos: <ul style="list-style-type: none"> • Total de participación de las preguntas realizadas. • Ganadores. 	Realizado
Req06	Los indicadores de rendimiento por asignatura que deben mejorarse son: <ul style="list-style-type: none"> • <u>% de participación por pregunta</u>: debe mostrar las preguntas erróneas por semestre y la cantidad de veces que se repitieron los errores. <p>Nota: El profesor quiere ver una muestra antes de decidir.</p>	No realizado - No posible
Req07	Cuando el profesor recibe la lista de participantes debe poder filtrar por: <ul style="list-style-type: none"> • Número de preguntas correctas. • Número de preguntas incorrectas. • Número de preguntas participadas. • Número de preguntas no participadas. • Orden de llegada. 	Realizado
Req08	El profesor debe poder cambiar su contraseña.	Realizado
Req09	El administrador debe contar con una nueva vista, para poder gestionar a los alumnos del sistema.	Realizado
Req10	Actualizar gráfico “preguntas realizadas versus participación por clase”	No realizado - No posible

Muchos de los requisitos que no fueron realizados fue debido a la necesidad de modificar los archivos por defecto de Angular lo que podría producir problemas en caso de que otro equipo de desarrollo deseara actualizar el sistema.

5.2. Descripción de Resolución de Requisitos

A continuación, se describe como han sido implementados los requerimientos en el sistema, abordando los cambios más significativos a nivel de código.

5.2.1. Req01

Para satisfacer este requerimiento, se realizaron los siguientes cambios:

1. Se añadió una nueva consulta llamada “*obtenerPreguntasRealizadas*”, cuya finalidad es entregar todas aquellas preguntas que han sido realizadas en el curso.
2. Aplicar esta nueva consulta en el controlador que rige la vista del estudiante.

De esta forma el estudiante no podrá ver preguntas no realizadas.

5.2.2. Req02

Para satisfacer este requerimiento, se realizaron los siguientes cambios:

1. Se añadieron dos tablas al modelo de datos: Tópicos y Temas, las cuales se describen a continuación:
 - a. Topico¹:

topico	Atributo	Tipo	Restricciones
	id_topico	Integer(11)	Clave Primaria.
	Nombre	Varchar(250)	No nulo.
	Id_asignatura	Integer(11)	Clave foránea de Asignatura.

- a. Tema:

tema	Atributo	Tipo	Restricciones
	id_tema	Integer(11)	Clave Primaria.
	Nombre	Varchar(250)	No nulo.
	Id_topico	Integer(11)	Clave foránea de Topico.

El tópico y el tema presentan 2 niveles de filtrado que sirven para lograr el filtrado de las preguntas al agregarlas a una clase o sesión. El tema es agregado a las preguntas que se encuentran en la base de datos pues son estas las preguntas que es necesario filtrar.

2. Se crearon nuevas consultas, las cuales se detallan en la Tabla 5.

¹ Escrito “Topico” no Tópico debido a que representa el nombre que posee la tabla.

- *TopicosController*: Este controlador se encarga de cargar el listado de tópicos en la vista de configuración de la asignatura para el usuario profesor. Además, provee las siguientes funciones:
 - *nuevoTopico*: Esta función se encarga de hacer el llamado de la consulta *crearTopico*.
 - *editarTopico*: Esta función se encarga de hacer el llamado de la consulta *editarTopico*.
 - *eliminarTopico*: Esta función se encarga de hacer el llamado de la consulta *eliminarTopico*.
- *TemasController*: Este controlador se encarga de cargar el listado de temas en la vista de configuración de la asignatura para el usuario profesor. Además, provee las siguientes funciones:
 - *nuevoTema*: Esta función se encarga de hacer el llamado de la consulta *crearTema*.
 - *editarTema*: Esta función se encarga de hacer el llamado de la consulta *editarTema*.
 - *eliminarTema*: Esta función se encarga de hacer el llamado de la consulta *eliminarTema*.
- *ModalAgregarTopicosController*: Este controlador es el encargado de capturar los datos ingresados por el usuario profesor cuando este crea o edita un tópico.
- *ModalAgregarTemasController*: Este controlador es el encargado de capturar los datos ingresados por el usuario profesor cuando este crea o edita un tema.

Con todo lo anterior, se puede apreciar la nueva pestaña de configuración de la asignatura, donde se muestran los tópicos y temas correspondientes a la asignatura (ver Ilustración 1).

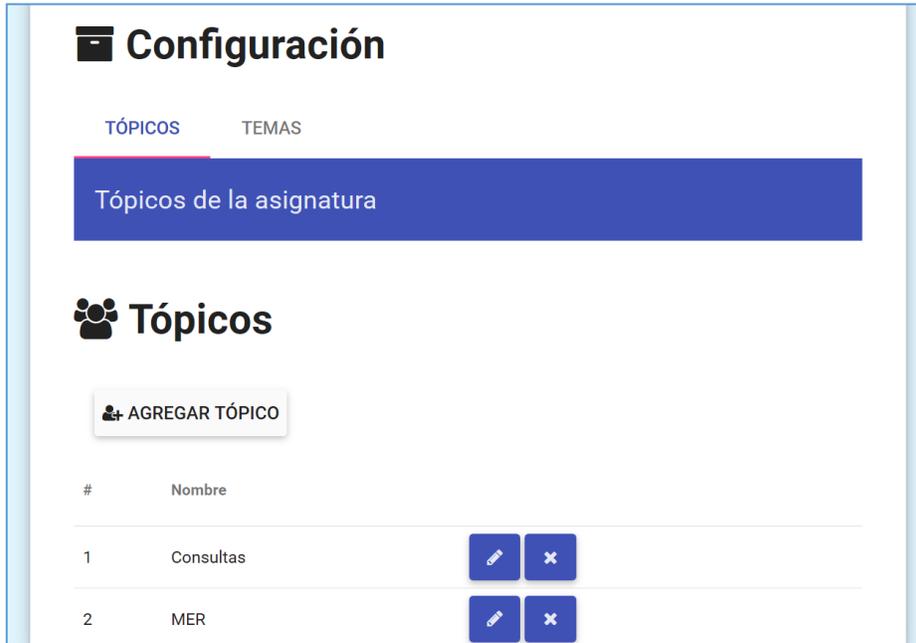


Ilustración 1. Vista de configuración de asignatura

Para el filtrado de preguntas, solo es necesario seleccionar Agregar Pregunta ya sea en la opción de preguntas del curso o en la clase y seleccionar un tópico de la lista y un tema (ver Ilustración 2).

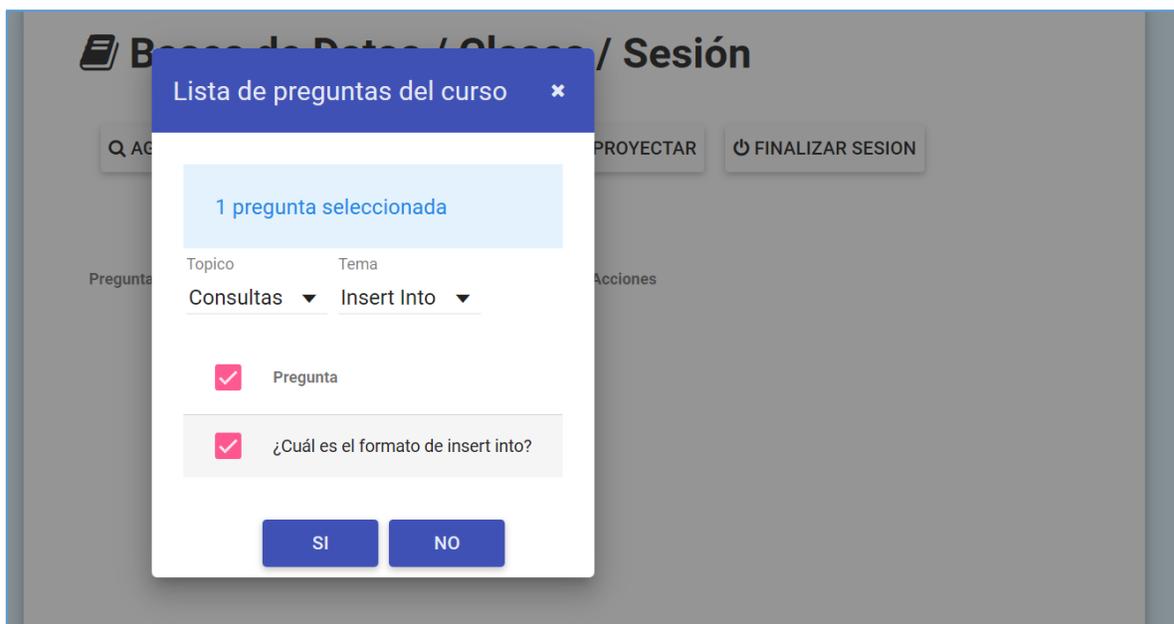


Ilustración 2. Filtrado de preguntas

5.2.3. Req03

Para satisfacer este requerimiento, se realizaron los siguientes cambios:

1. Participantes por pregunta: Se eliminó el gráfico.
2. Resultado de preguntas realizadas por clase: Se modificó el antiguo gráfico por un gráfico de barras llamado *MultiBarChar*. Este gráfico puede apilarse o agruparse (Ilustración 3).

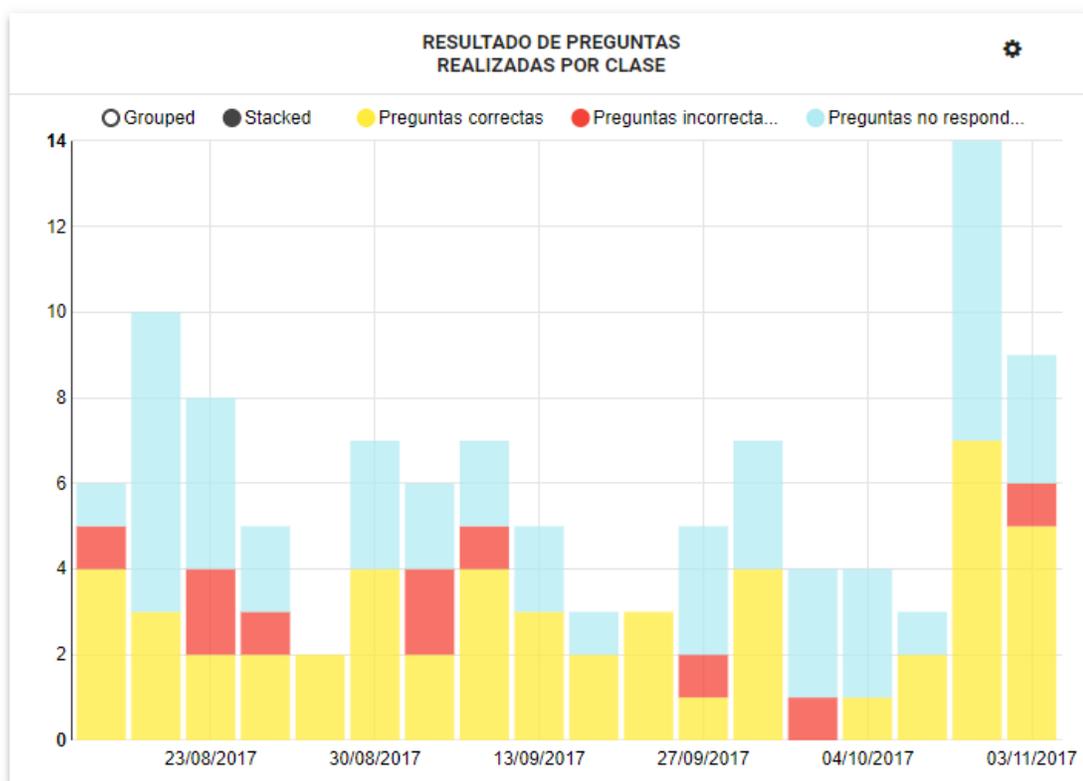


Ilustración 3. Gráfico resultado de preguntas realizadas por clase

3. Preguntas realizadas vs participación por clase: No fue posible quitar gráfico minimizado, ya que esta opción viene por defecto en Angular.
4. Participación por estudiante en preguntas: Se eliminó el gráfico.
5. Puntos ganados por estudiantes: Se eliminó el gráfico.
6. Participación de estudiante por clase: No fue posible seleccionar solo un estudiante, ya que la selección viene por defecto en Angular.

5.2.4. Req04

Para satisfacer este requerimiento, se realizaron los siguientes cambios:

Se crearon las siguientes funciones:

- *respuestas()*: Esta función permitirá mostrar los datos correspondiente por medio de un gráfico, las respuestas correctas e incorrectas del alumno seleccionado. Para obtener los datos, se crearon las siguientes consultas:
 - *incorrectaAlumno*: Entrega la cantidad de respuestas incorrectas del alumno que participa para responder.
 - *correctaAlumno*: Entrega la cantidad de respuestas correctas del alumno que participa para responder.
- *participacion()*: Esta función permitirá mostrar los datos correspondiente por medio de un gráfico, la participación y no participación del alumno del total de las preguntas realizadas. Para obtener los datos, se creó las siguientes consultas:
 - *participacionAlumno*: Entrega la cantidad de participación del alumno.
 - *preguntasRealizadas*: Entrega la cantidad de preguntas realizadas del alumno.
- *seleccion()*: Esta función permitirá mostrar los datos correspondiente por medio de un gráfico, la selección y no selección del alumno para responder las preguntas realizadas. Para obtener los datos, se creó las siguientes consultas:
 - *preguntasSeleccionada*: entrega la cantidad de preguntas seleccionas del alumno.
 - *preguntasNoSeleccionada*: entrega la cantidad de preguntas no seleccionas del alumno.

Según lo anterior, en la Ilustración 4, se muestran los gráficos correspondientes a cada uno de los puntos solicitados en el Req04.

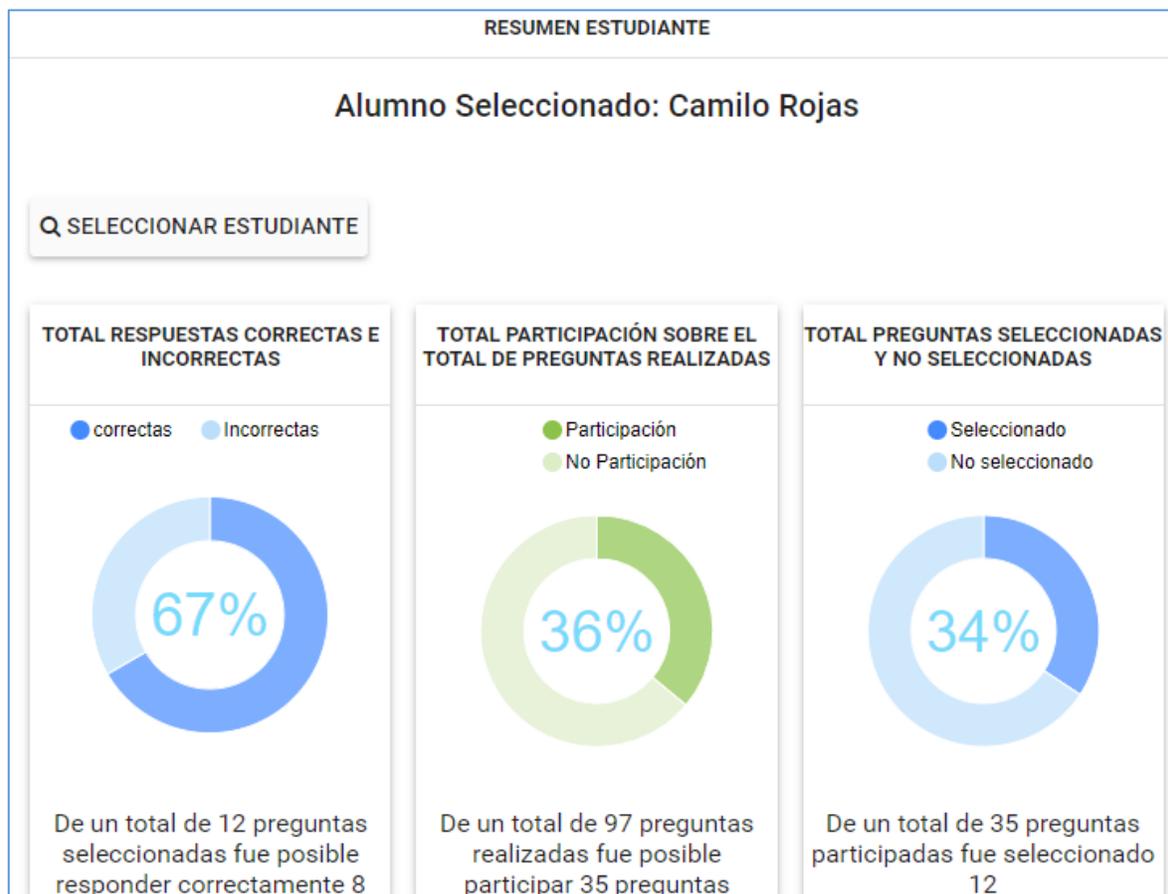


Ilustración 4. Resumen de estudiante

5.2.5. Req05

Para satisfacer este requerimiento, se realizaron los siguientes cambios:

1. Se añadió un nuevo atributo al curso, correspondiente a la cantidad de puntos que necesita un alumno. Esta cantidad puede modificarse en la configuración del curso.
2. Se añadió una nueva consulta llamada “*obtenerCantidadNoGanadores*”, cuya finalidad es entregar la cantidad de alumnos que no llegan a la meta de 12 preguntas.

```
SELECT count(*) AS obtenerCantidad
FROM pertenece p
WHERE p.id_curso = ?
AND (SELECT meta_alumno FROM curso WHERE id_curso = ?) >
(SELECT count(*)
```

```
FROM participan_por_responder pa, pregunta pre
WHERE p.id_user = pa.id_user
AND p.id_curso = pre.id_curso
AND pa.id_pregunta = pre.id_pregunta
AND estado_part_preg = "ganador") +
(SELECT count(*)
FROM participa par, actividad act
WHERE p.id_user = par.id_user
AND p.id_curso = act.id_curso
AND par.id_actividad = act.id_actividad
AND estado_part_act = "ganador")
GROUP BY p.id_curso;
```

3. Se añadieron dos nuevas funciones en el controlador *InformacionAsignaturaController*, las cuales se detallan a continuación:

- *Ganadores*: Esta función obtiene los datos de los estudiantes que llegaron a la meta y carga los datos para que puedan ser utilizados por el gráfico 'Ganadores', en la información general de la asignatura.
- *TotalParticipacion*: Esta función obtiene los datos del total de participación de los alumnos de la asignatura y además obtiene las posibles participaciones de los alumnos de la asignatura. Por último, carga estos datos para que puedan ser utilizados por el gráfico '*Total de participación de las preguntas realizadas*', en la información general de la asignatura.

Los gráficos resultantes se observan en la Ilustración 5.

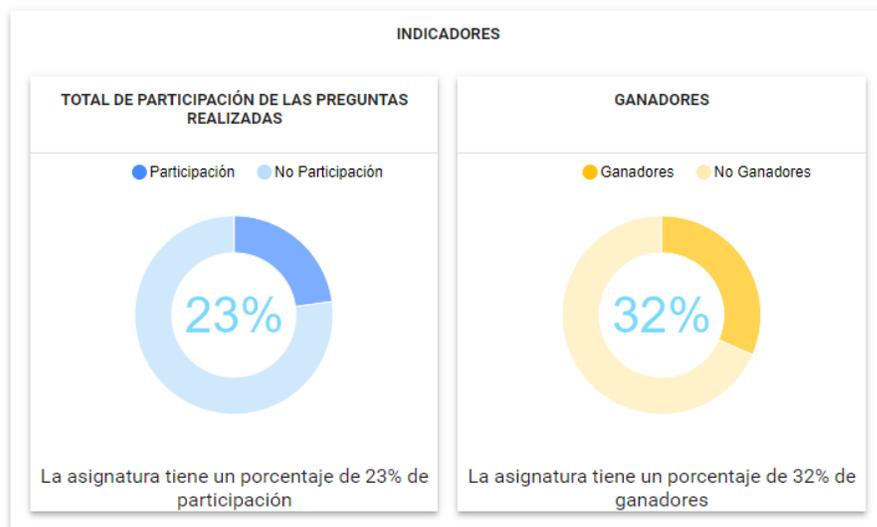


Ilustración 5. Indicadores de estadística general

5.2.6. Req06

Este requerimiento no se pudo satisfacer debido a que era necesario cambiar las opciones por defecto de Angular y dado que este gráfico no presenta información al existir un gran número de preguntas, se optó por eliminarlo.

5.2.7. Req07

Para satisfacer este requerimiento, se realizaron los siguientes cambios:

1. Se modificó el archivo *preguntaSesionProfesor.html*, donde se cambiaron los encabezados de la tabla de participantes para que fueran seleccionables y permitir el ordenamiento.
2. Se modificó el controlador *preguntasController*, añadiendo el código necesario para actualizar las variables necesarias para el ordenamiento.

El nuevo encabezado se muestra en la Ilustración 6, en este, se encuentran por defecto seleccionados el ordenamiento por llegada y la opción por sesión, las opciones sesión y semestre afectan al *Total Preguntas*, *Participa*, *No Participa*, *Correctas*, *Incorrectas* y *No Seleccionado*, para aplicar el ordenamiento dependiendo de si se desea utilizar los datos por sesión o semestre; el orden de llegada es independiente.



Ilustración 6. Encabezado para el ordenamiento de participantes

5.2.8. Req08

Este requerimiento sufrió un cambio de último momento, al realizar la migración del servidor, el cambio de contraseña por parte del profesor era posible sin problema, pero al agregar los datos de la base de datos anterior, el error volvió a aparecer lo que hace suponer un problema con respecto a la base de datos.

5.2.9. Req09

Para satisfacer este requerimiento, se realizaron los siguientes cambios

1. Se añadió una nueva vista para el usuario administrador llamada *adminAlumno.html*.
2. Se crearon dos nuevas consultas, llamadas *obtenerAlumnosSistema*, cuya función es entregar una lista de todos los alumnos del sistema, y *eliminarAlumno*, cuya función es eliminar al estudiante del sistema.
3. Se añadieron tres nuevas funciones al controlador *AdministradorController*, las cuales se detallan a continuación:
 1. *nuevoAlumno*: Esta función se encarga de hacer el llamado a la consulta *crearEstudiante*.
 2. *editarAlumno*: Esta función se encarga de hacer el llamado a la consulta *editarEstudiante*.
 3. *eliminarAlumno*: Esta función se encarga de hacer el llamado a la consulta *eliminarEstudiante*.
4. Se realizaron cambios en tres tablas, para permitir eliminar un alumno del sistema, las cuales se detallan a continuación:

1. Tabla pertenece:
 - ALTER TABLE pertenece
 - ADD CONSTRAINT fk_pertenece
 - FOREIGN KEY (id_curso) REFERENCES curso (id_curso)
 - ON DELETE CASCADE
2. Tabla participan por responder:
 - ADD CONSTRAINT fk_participan_por_responder
 - FOREIGN KEY (id_user) REFERENCES estudiante (id_user)
 - ON DELETE CASCADE
3. Tabla participa
 - ALTER TABLE pertenece
 - ADD CONSTRAINT fk_pertenece
 - FOREIGN KEY (id_user) REFERENCES estudiante (id_user)
 - ON DELETE CASCADA

La vista del administrador es bastante similar a la del profesor (ver Ilustración 7) la diferencia está en que, al eliminar un alumno, lo elimina completamente del sistema.

Administración / Alumnos

Alumnos

+ AGREGAR ALUMNO

Rut	Nombre	Apellido	Usuario	Clave	Acciones
18.371.430-9	Ivan22	Cardemil	ICardemil	183714309	 
19.355.257-9	Bastián	Cuadros	BCuadros	193552579	 
19.045.930-6	José Luis	Mamani	JMamani	190459306	 
18.313.026-9	Llosect	Moscoso	LMoscoso	183130269	 
18.607.629-K	Camilo	Rojas	CRojas	18607629	 

Ilustración 7. Vista de alumnos del administrador

5.2.10. Req10

Este requerimiento no se pudo satisfacer debido a que era necesario cambiar las opciones por defecto de angular.

VI. DOCUMENTACIÓN

Tomando en cuenta que, uno de los principales objetivos es actualizar el sistema, como grupo se optó por traspasar toda la documentación existente de la memoria de Yordan José Vera Castillo a la herramienta StarUml [1]. Esta es una herramienta conocida por los miembros del equipo, que permite realizar diagramas *UML*. En este sentido, ayudará en gran medida a gestionar los cambios que se realizaron en el modelo, cuyo fin es agilizar el proceso de actualización a nivel de implementación.

Además, la documentación estará disponible en la plataforma Redmine [2] para que en un futuro esta información esté disponible para aquellos interesados en el sistema.

Uno de los principales cambios en la documentación fue el modelo de datos donde se agregaron el tópico y el tema (Ilustración 8).

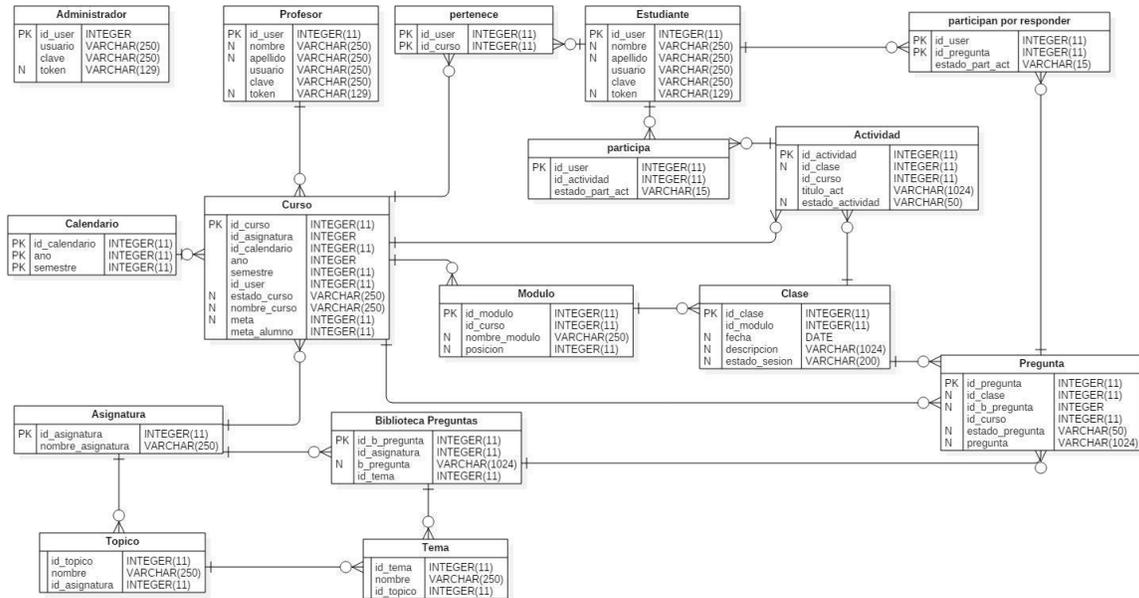


Ilustración 8. Modelo de datos

Finalmente, se realizó un manual de usuario [3] para todo aquel que utilice el sistema, ya sea el administrador, el profesor o el alumno, detallando instrucciones con respecto al uso del sistema y la interpretación de la información entregada. Este manual estará disponible en la plataforma Redmine.

VII. PROBLEMAS CON EL DESARROLLO

Durante el desarrollo el principal problema fue la falta de tiempo, debido a los diversos errores que se produjeron y el largo tiempo que tomó resolverlos. Esto se vio agravado por la falta de conocimiento acerca de Angular por parte del equipo.

El problema de tiempo provocó que las pruebas a realizar sobre el sistema no lograran desarrollarse, esto es un deja al sistema en un estado de seguridad, en cuanto a funcionalidad y detalles, sin confirmar.

VIII. REQUERIMIENTOS NO IMPLEMENTADOS

Entre los requerimientos no implementados, se encuentran los requerimientos funcionales mencionados anteriormente y algunos de los requerimientos no funcionales como los que se describen a continuación.

8.1. Cambio de Bower

Uno de los requisitos no funcionales que no se abordaron fue el cambio del uso de Bower por parte del sistema para la descarga de dependencias, debido al retraso que sufrió el sistema.

8.2. Manejo de Versiones de Redmine

Otro requerimiento que se vio afectado por el lento avance del proyecto fue el uso de manejo de versiones por medio de Redmine, que puede conectarse con el repositorio de GitHub.

8.3. Cambio de Mecánica de las Preguntas

Entre los requerimientos prescindibles se encontraba la modificación de preguntas, permitiendo la modificación de fuentes y añadido de imágenes. Este cambio tampoco llegó a realizarse.

IX. IDEAS A FUTURO

Durante el desarrollo de la actualización, surgieron nuevas ideas para implementar en un futuro, algunas de estas presentadas por profesores interesados en el uso del sistema.

9.1. Teoría de Colores

Entre los temas propuestos, surgió la idea de modificar los colores de los gráficos del sistema por nombres más representativos, sería necesario un análisis de los importante que sería esta idea para el sistema.

9.2. Dificultad de Preguntas

Otra idea es la incorporación de un atributo de dificultad a las preguntas, esto serviría para determinar el nivel de dificultad que los alumnos son capaces de manejar.

9.3. Problemas de Gráfico

Al finalizar el proyecto, se descubrió un error con el que cuenta el sistema. El gráfico "*Estadística preguntas*" no muestra la información la mayoría de las veces, la posible razón es que, debido a la naturaleza concurrente de las promesas y su uso para obtener la información, el gráfico se carga antes de que la información

se encuentre disponible. Una solución a esto sería la implementación de un *TimeOut*, de modo que el gráfico solo se cargue una vez se ha cargado la información, durante las pruebas realizadas se trató con un tiempo de espera de 100 segundos sin resultados positivos por lo que se espera que esta operación podría tomar mucho tiempo.

X. CONCLUSIONES

Tras el desarrollo del proyecto se obtuvieron las siguientes conclusiones:

- El establecimiento de una lista de requerimientos de forma temprana es importante para asegurar el trabajo a realizar y evitar el retraso del proyecto.
- La migración de un sistema, puede traer consigo problemas de compatibilidad, lo que produce problemas en caso de que un nuevo equipo de desarrollo tome las riendas del proyecto al no saber los detalles con que cuenta el sistema.
- La actualización de MySQL trajo consigo una serie de problemas para el sistema, pues este no fue planificado para los nuevos estándares de seguridad que presenta MySQL desde la versión 5.7.5.
- Git y GitHub son herramientas importantes para el manejo de versiones o para la distribución de sistemas.
- La actualización tomó en cuenta los requerimientos más importantes debido al retraso del proyecto.
- La documentación presenta todos los cambios realizados en el sistema, tratando de especificar las rutas que se siguen al momento de realizar operaciones, debido al modelo del sistema MVW (Model View Whatever), es complicada la actualización de archivos que interactúan entre sí, esto se veía incrementado pues la documentación utilizaba nombres desactualizados, por eso fue necesaria la modificación de estos documentos.
- Para que futuros equipos no sufran problemas de desarrollo, hay que tener en cuenta los posibles problemas que puede tener sistema, se recomienda comprobar las versiones que se utilizan y su lista de cambios, y capacitar de la manera debida al equipo con las tecnologías con que cuenta el sistema.
- Si bien el sistema sufrió de una actualización, aún quedan temas que pueden mejorar, el uso del sistema otorgará nueva información sobre qué aspectos es necesario modificar, agregar o eliminar.

Referencias

- [1] MKLab, «StarUML,» [En línea]. Available: <http://staruml.io/>. [Último acceso: 15 12 2017].
- [2] «CRSoq - Redmine,» [En línea]. Available: <http://146.83.109.227/redmine/projects/crsoq>. [Último acceso: 15 12 2017].
- [3] Equipo de Desarrollo CRSoq 2.0, «Manual de Usuario - CRSoq,» Arica, 2017.