

UNIVERSIDAD DE TARAPACÁ



ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL,  
INFORMÁTICA Y DE SISTEMAS

EUIIIS

Área de Ingeniería en Computación e Informática



ACTUALIZACIÓN CRSoq 2.0  
AVANCE I

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

**Curso:** Proyecto III

**Profesor:** Ricardo Valdivia Pinto

ARICA, 25 de octubre de 2017

## RESUMEN

El presente avance tiene como objeto presentar el trabajo realizado en torno al desarrollo de la actualización, migración y redocumentación del sistema CRSQ.

El documento consta de los avances con respecto a la especificación de requisitos, donde se presenta la lista de requerimientos determinados por el equipo de desarrollo en conjunto con el cliente, la migración del sistema indicando el principal problema que surgió, el traspaso a GitHub presentando información básica de estas herramientas y la propuesta de actualización, indicando el estado actual del avance de la actualización.

## INDICE

I.	INTRODUCCIÓN .....	4
II.	REQUERIMIENTOS .....	5
III.	MIGRACIÓN AICI .....	6
3.1.	El Problema .....	6
3.2.	Migración del Servidor.....	7
IV.	TRASPASO A GITHUB .....	7
4.1.	Git.....	7
4.2.	GitHub .....	7
V.	PROPUESTA DE ACTUALIZACIÓN .....	8
5.1.	Req01 .....	8
5.2.	Req02 .....	9
5.3.	Req05 .....	9
VI.	DOCUMENTACIÓN .....	10
VII.	CONCLUSIONES.....	12
VIII.	REFERENCIAS BIBLIOGRÁFICAS .....	13
IX.	ANEXOS.....	14
9.1.	Ver preguntas de un curso .....	14
9.2.	Modificación código para requisito clave Req01.....	15

## Índice de Figuras

Figura 1.	Diagrama de secuencia para el caso de uso - Ver clases de un curso. ....	8
Figura 2.	Diagrama de secuencia para el caso de uso - Ver preguntas de un curso.....	9
Figura 3.	Diagrama de casos de uso – usuario profesor .....	10
Figura 4.	Diagrama de casos de uso - usuario estudiante.....	11
Figura 5.	Diagrama de casos de uso - usuario administrador .....	11
Figura 6.	Caso de uso - ver preguntas de un curso .....	14
Figura 7.	Diagrama de secuencia para el caso de uso - Ver preguntas de un curso.....	14

## Índice de Tablas

Tabla 1.	Requerimientos funcionales.....	5
Tabla 2.	Requisitos prescindibles.....	6
Tabla 3.	Requisitos no funcionales .....	6

## I. INTRODUCCIÓN

A lo largo de la historia se han desarrollado diversas herramientas para el desarrollo interactivo de una clase, buscando involucrar cada vez más al alumno dentro del proceso educativo. CRSoq fue desarrollado con la intención de fomentar la participación del estudiante y asistir al profesor al momento de visualizar el avance de sus estudiantes, basándose en una dinámica de puntos otorgados al responder a preguntas abiertas.

Tras la utilización del sistema en un entorno real, nuevas ideas salieron a la luz para poder mejorar el sistema, dando a paso a lo que próximamente será la versión 2.0 de este mismo.

Entre las características a modificar se encuentran, la mejora de soporte en la actual implementación del sistema en la Universidad de Tarapacá, la migración del sistema a *GitHub* por medio de la herramienta de control de versiones *Git* y la actualización de funciones del sistema en base a los requerimientos del cliente para mejorar la experiencia y ajustar la utilidad de la información que este presenta.

## II. REQUERIMIENTOS

Para el desarrollo de 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. Requerimientos 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"> <li>• Participantes por pregunta: quitar.</li> <li>• Resultado de preguntas realizadas por clase: cambiar por gráfico de barras.</li> <li>• Preguntas realizadas vs participación por clase: quitar gráfico minimizado.</li> <li>• Participación por estudiante en preguntas: quitar.</li> <li>• Puntos ganados por estudiantes: quitar.</li> <li>• Participación de estudiante por clase: debe estar seleccionado solo el primer alumno o ninguno. Quitar gráfico minimizado.</li> </ul>
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"> <li>• Gráfico circular que muestra el total de respuestas correctas e incorrectas sobre el total de preguntas participadas.</li> <li>• Gráfico circular que muestra la participación sobre el total de preguntas realizadas.</li> <li>• Gráfico circular que muestra el total de preguntas seleccionadas y no seleccionadas sobre el total de preguntas participadas.</li> </ul>
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"> <li>• Total de participación de las preguntas realizadas.</li> <li>• Ganadores.</li> </ul>
Req06	Los indicadores de rendimiento por asignatura que deben mejorarse son: <ul style="list-style-type: none"> <li>• % de participación por pregunta: debe mostrar las preguntas erróneas por semestre y la cantidad de veces que se repitieron los</li> </ul>

	errores. Nota: El profesor quiere ver una muestra antes de decidir.
<b>Req07</b>	<p>Cuando el profesor recibe la lista de participantes debe poder filtrar por:</p> <ul style="list-style-type: none"> <li>• Número de preguntas correctas.</li> <li>• Número de preguntas incorrectas.</li> <li>• Número de preguntas participadas.</li> <li>• Número de preguntas no participadas.</li> <li>• Orden de llegada.</li> </ul>
<b>Req08</b>	El profesor debe poder cambiar su contraseña.

Tabla 2. Requisitos prescindibles

Requerimiento	Descripción
<b>Req09</b>	<p>La creación de una pregunta debe permitir:</p> <ul style="list-style-type: none"> <li>• Utilizar diferentes tipos de letra (fuentes).</li> <li>• Cambiar colores.</li> <li>• Adjuntar imágenes.</li> </ul>

Tabla 3. Requisitos no funcionales

Requerimiento	Descripción
<b>NF01</b>	Migrar el proyecto a un servidor del AICI
<b>NF02</b>	<p>Administrar el proyecto como un proyecto open source:</p> <ul style="list-style-type: none"> <li>• Gestionar el proyecto en las plataformas RedMine y GitHub (repositorio)</li> <li>• Manejar un control de versiones (Git, SVN, etc.)</li> </ul>

### III. MIGRACIÓN AICI

Uno de los requerimientos no funcionales es la migración del sistema a un nuevo servidor presente 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 *CRSoq* 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 3.1, 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.

Al realizar la migración del sistema, ocurrieron ciertos problemas cuyas soluciones deben ser mencionadas 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 *node* 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 modos, por ende, para solucionar este problema, es necesario desactivar el modo nombrado anteriormente.

## 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.

### 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 permite el desarrollo privado o público de un proyecto. Para el caso del sistema CRSoq, se busca que su uso sea utilizado y mejorado por la comunidad abiertamente por lo que se optó por un desarrollo público.

El sistema puede ser clonado por medio del comando:

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

## V. PROPUESTA DE ACTUALIZACIÓN

### 5.1. Req01

Tras realizar el análisis del caso de uso Ver preguntas de un curso (Anexo 9.1, Figura 6), se tomó la decisión de cambiar la forma en que un alumno obtendrá las preguntas del curso. Para realizar este cambio, se modificó el caso de uso Ver preguntas del curso, como se aprecia en la Figura 1.

<b>Nombre caso de uso</b>	Ver preguntas de un curso
<b>Código caso de uso</b>	CU12
<b>Participantes</b>	Estudiante
<b>Descripción</b>	El estudiante a través de una interfaz podrá ver la lista de preguntas del curso junto a su participación.
<b>Pre-condiciones</b>	
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón “Preguntas”.</li> <li>2. El sistema <i>carga las preguntas realizadas</i> del curso junto a la participación del estudiante y las despliega en una interfaz.</li> </ol>
<b>Flujo Alternativo</b>	-
<b>Excepciones</b>	-
<b>Post-condiciones</b>	-

Figura 1. Diagrama de secuencia para el caso de uso - Ver clases de un curso.

En esta nueva versión, se destaca el cambio de *carga las preguntas realizadas*, lo que es esencial para que el alumno solo pueda visualizar aquellas preguntas que han sido realizadas por el profesor en el curso.

Con respecto al diagrama de secuencia Ver preguntas de un curso (Anexo 9.1, Figura 7), este también sufrió de modificaciones. Cambiando el mensaje *obtenerPreguntas(curso)* por *obtenerPreguntasRealizadas(curso)*.

Sin embargo, no se eliminó el servicio *obtenerPreguntas(curso)*, ya que este también es utilizado por el profesor.

El nuevo diagrama de secuencia se muestra en la Figura 2.

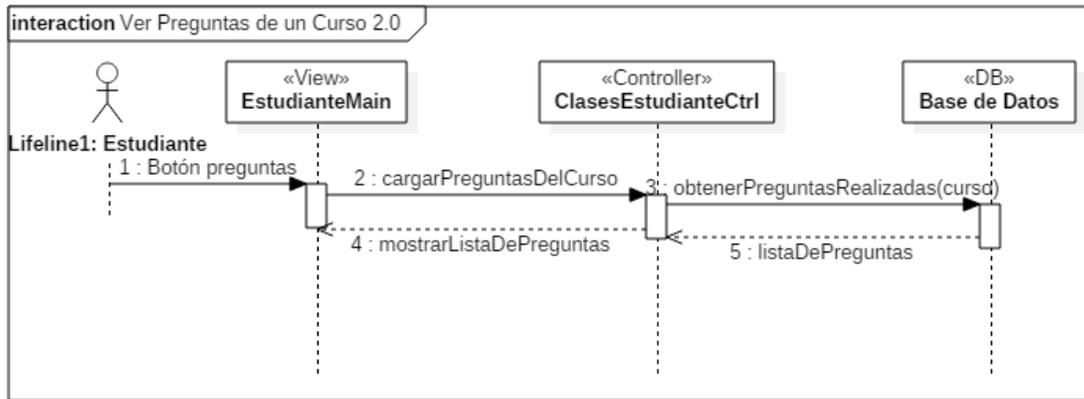


Figura 2. Diagrama de secuencia para el caso de uso - Ver preguntas de un curso.

Para realizar este cambio en términos de codificación, se añadió *obtenerPreguntasRealizadas* como un servicio, y además se modificó el controlador. Estos cambios son reflejados en el Anexo 9.2.

## 5.2. Req02

Para solucionar el problema presente en el Req02, al analizar el modelo actual es imposible utilizar módulos a la hora de filtrar preguntas, pues este subdivide los cursos, mientras que las preguntas están relacionadas a una asignatura, lo que produciría problemas de búsqueda.

De esta forma, se propone el uso de tópicos y temas, que serán configurados dentro de la asignatura, determinando que un tópico tenga varios temas para facilitar la organización de las preguntas.

## 5.3. Req05

Los gráficos propuestos para el requisito número 5 se detallan a continuación:

- a) Gráfico 1: Total de participación de las preguntas realizadas



Descripción: Este gráfico entrega la información con respecto al porcentaje de participación de los estudiantes del profesor por asignaturas (cursos).

b) Gráfico 2: Ganadores

Descripción: Este gráfico entrega la información de la cantidad de alumnos que llegaron a la meta establecida por el profesor, es decir, cuántos alumnos obtuvieron nota 7, en contraste con los alumnos que no llegaron a esa meta.



## 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*. 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 realizarán en el modelo, cuyo fin es agilizar el proceso de actualización a nivel de implementación.

Las siguientes figuras (Figura 3, Figura 4 y Figura 5) muestran los diagramas de caso de uso para los distintos usuarios del sistema.

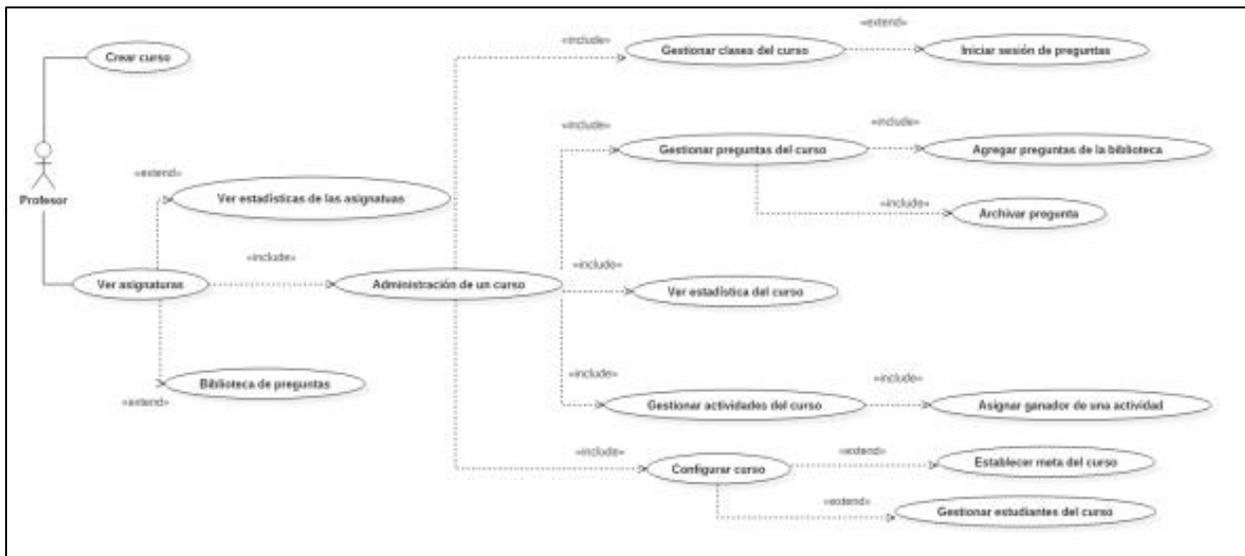


Figura 3. Diagrama de casos de uso – usuario profesor

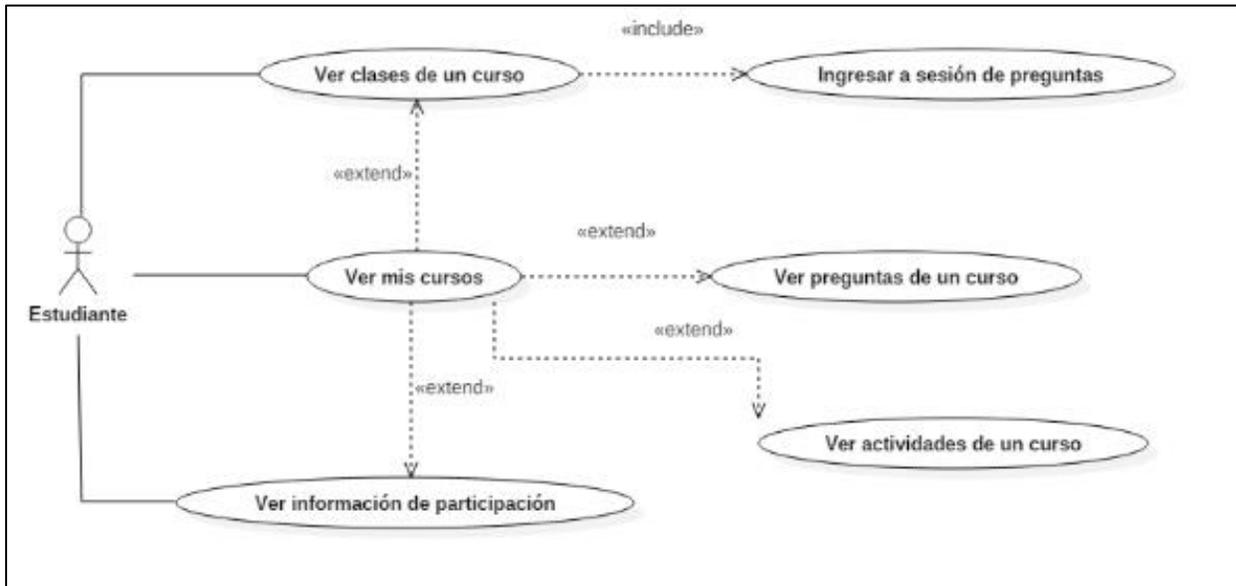


Figura 4. Diagrama de casos de uso - usuario estudiante

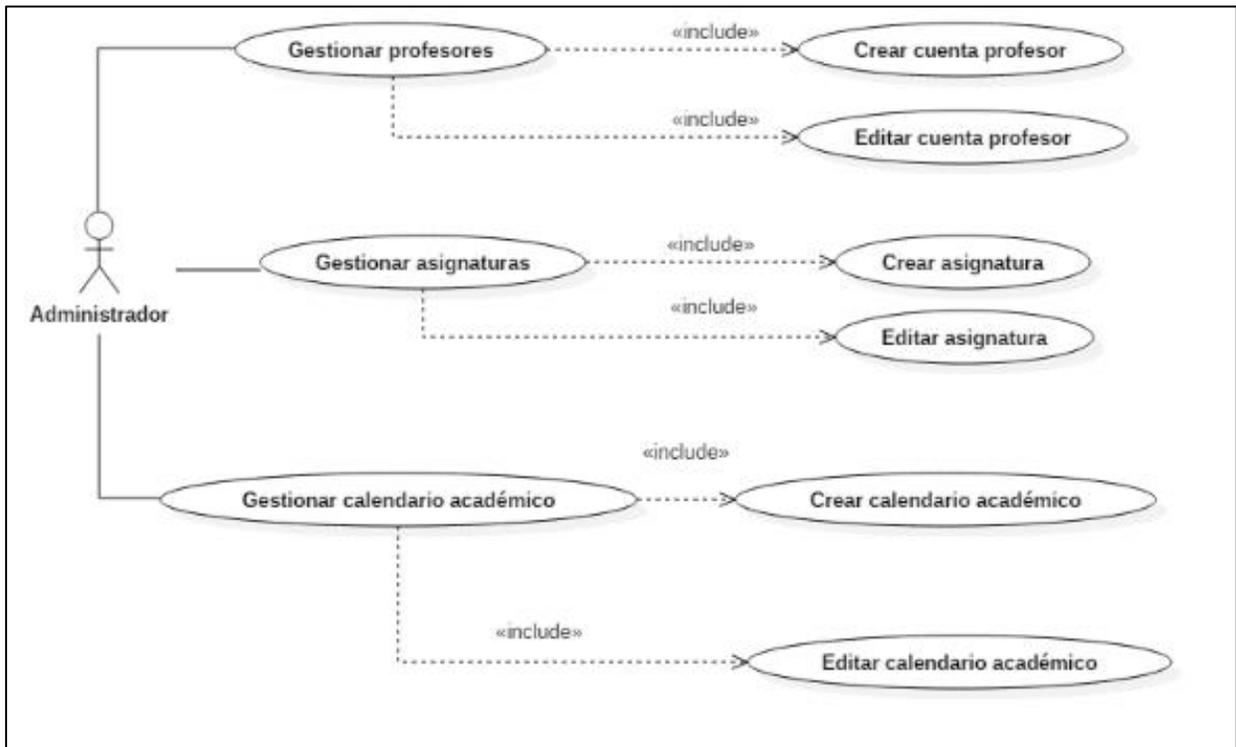


Figura 5. Diagrama de casos de uso - usuario administrador

## VII. CONCLUSIONES

Tras este avance, se determinan problemas con el progreso del proyecto, presentando retrasos en la actualización del sistema. Entre las conclusiones que se pueden rescatar con respecto al trabajo desarrollado del proyecto se encuentran:

- El retraso en la especificación de requerimientos afectó en el desarrollo de la actualización.
- La migración del sistema presentó problemas de versión de los componentes del sistema, problemas y soluciones necesarias de mencionar en la documentación del proyecto, sea la Wiki de Redmine o el documento de instalación GitHub.
- Git presenta una herramienta útil para el control de versiones y evitar el dañado del código.
- Debido a la falta de credenciales de la conexión remota actual del sistema para Git, se decidió quitar este servidor remoto.
- Actualmente el sistema se encuentra en el repositorio de uno de los miembros del equipo de desarrollo, pero se evaluará la posibilidad de moverlo eventualmente.
- Al momento de realizar la actualización del sistema, se debe tener en cuenta la replicación del código presente debido al modelo utilizado, teniendo que modificar diferentes archivos para lograr la configuración deseada.

## VIII. REFERENCIAS BIBLIOGRÁFICAS

## IX. ANEXOS

### 9.1. Ver preguntas de un curso

Para términos de este avance, solo se describirá la primera versión del Caso de Uso involucrado en la actualización realizada. Las siguientes figuras (Figura 6 y Figura 7) describen el Caso de Uso Ver preguntas de un curso y su diagrama de secuencia correspondiente.

<b>Nombre caso de uso</b>	Ver preguntas de un curso
<b>Código caso de uso</b>	CU12
<b>Participantes</b>	Estudiante
<b>Descripción</b>	El estudiante a través de una interfaz podrá ver la lista de preguntas del curso junto a su participación.
<b>Pre-condiciones</b>	-
<b>Flujo principal</b>	El usuario presiona el botón “Preguntas”. El sistema carga las preguntas del curso junto a la participación del estudiante y las despliega en una interfaz.

Figura 6. Caso de uso - ver preguntas de un curso

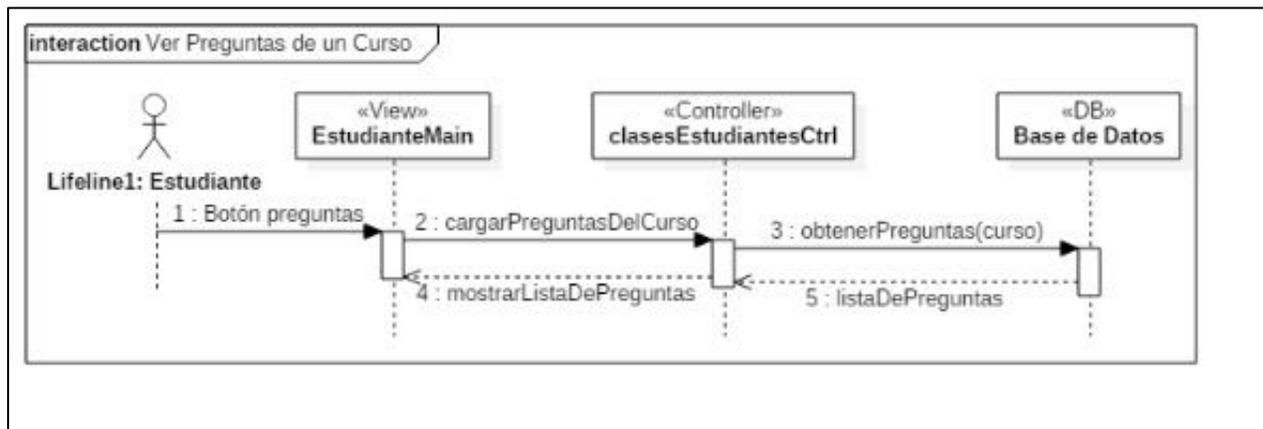


Figura 7. Diagrama de secuencia para el caso de uso - Ver preguntas de un curso

## 9.2. Modificación código para requisito clave Req01.

En la siguiente imagen, se aprecia el cambio realizado al servicio PreguntasServices. El cuadro rojo representa el código añadido.

```
crsApp.factory('PreguntasServices', function($http, $q){
  var postHelper = function(ruta, data){
    var deferred = $q.defer();
    var promise = deferred.promise;
    $http.post(ruta, data)
      .success(function (response) {
        deferred.resolve(response);
      })
      .error(function (error) {
        deferred.reject(error);
      });
    return promise;
  };
  return {
    obtenerPreguntasRealizadas: function (curso) {
      return postHelper('/preguntas/obtenerPreguntasRealizadas', curso);
    },
    obtenerPreguntasClase: function (clase) {
      return postHelper('/preguntas/obtenerPreguntasClase', clase);
    },
    obtenerPreguntaPorId: function (id_pregunta) {
      return postHelper('/preguntas/obtenerPreguntaPorId', id_pregunta);
    },
    crearPreguntaCurso: function (pregunta) {
      return postHelper('/preguntas/crearPreguntaCurso', pregunta);
    },
    obtenerPreguntasCurso: function (curso) {
      return postHelper('/preguntas/obtenerPreguntasCurso', curso);
    },
    //usar servicio de biblioteca
    obtenerPreguntasAsignatura: function (curso) {
      return postHelper('/preguntas/obtenerPreguntasAsignatura', curso);
    },
    actualizarPregunta: function (pregunta) {
      return postHelper('/preguntas/actualizarPregunta', pregunta);
    },
    actualizarEstadoPregunta: function (pregunta) {
      return postHelper('/preguntas/actualizarEstadoPregunta', pregunta);
    },
  };
});
```

En la siguiente imagen, se aprecia la función de obtenerPreguntasRealizadas. La condición para obtener las preguntas realizadas se encuentra destacada con una línea roja.

```
router.post('/obtenerPreguntasRealizadas', function (req, res) {
  if(!req.body){
    return res.sendStatus(400);
  }else{
    connection.query('SELECT id_pregunta, id_clase, id_b_pregunta, id_curso, estado_pregunta, pregunta FROM pregunta WHERE id_curso = ? AND estado_pregunta = "realizada"', [req$
    if(error){
      res.status(500);
      return res.json({'success':false, 'err':error});
    }else{
      res.status(200);
      return res.json({'success':true, 'result': rows});
    }
  }
});
```

Por último, se modificó *Preguntascontroller*, el cual es el encargado de manejar este evento. La única modificación que se realizó, es que ahora se hace una llamada a *PreguntasServices.obtenerPreguntasRealizadas*.

```
crsApp.controller("PreguntasEstudianteController", function ($scope, $stateParams, $mdDialog, toastr, PreguntasServices, CursosServices, C)
var semestres = CursosServices.obtenerCursosLocal();
var estudiante = SessionServices.getSessionData();
var semestre = _.findWhere(semestres, {'ano':Number($stateParams.ano), 'semestre':Number($stateParams.semestre)});
$scope.curso = _.findWhere(semestre.cursos, {'id_curso': Number($stateParams.id_curso)});

//preguntas curso
$scope.listaPreguntasCurso = [];
//mis preguntas
$scope.misParticipaciones = [];

$scope.promesas = [];
var promesaPreguntas = PreguntasServices.obtenerPreguntasRealizadas($scope.curso).then(function (response) {
    if(response.success){
        $scope.listaPreguntasCurso = _.cloneDeep(response.result);
        cargarClases();
        obtenerParticipacion();
    }else{
        toastr.error("No se pudieron obtener las preguntas del curso: "+response.err.code, "Error");
    }
});
$scope.promesas.push(promesaPreguntas);
var cargarClases = function () {
    _.forEach($scope.listaPreguntasCurso, function (pregunta) {
        if(!_.isNull(pregunta.id_clase)){
            ClasesServices.obtenerClasesPorID({id_clase:pregunta.id_clase}).then(function (response) {
```