



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DESARROLLO Y ADAPTACIÓN DE SISTEMA DE COEVALUACIÓN DE EQUIPOS  
PARA CURSOS DE TRABAJO COLABORATIVO

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERA CIVIL EN COMPUTACIÓN

VALENTINA AYLÉN PINTO DE LAS RIBERAS

PROFESORA GUÍA:  
JOCELYN SIMMONDS WAGEMANN

MIEMBROS DE LA COMISIÓN:  
JOSE BENGURIA DONOSO  
ÉRIC TANTER

SANTIAGO DE CHILE  
2023

# Resumen

El trabajo colaborativo considera un grupo de personas que trabajan conjuntamente en una actividad. Esta situación se da tanto en el diario vivir como en el ámbito académico. A lo largo de nuestra vida, son múltiples las ocasiones en que debemos enfrentarnos a este escenario, que forma parte fundamental de nuestro desarrollo propio y comunitario. Es por esto que un grupo de investigadores de la Universidad de Chile pretenden estudiar el trabajo colaborativo. Para lo cual, nace la necesidad de contar con un sistema en que ellos puedan administrar coevaluaciones y donde los alumnos puedan responderlas.

En la malla de la de la carrera Ingeniería Civil en Computación se encuentran asignaturas que requieren trabajo colaborativo para el desarrollo de software, por lo cual existe un sistema de coevaluaciones de dicho departamento. Es debido a esto que nace la idea de utilizar como referencia esta aplicación para implementar un nuevo sistema que responda a las necesidades de los investigadores en trabajo colaborativo.

Durante este trabajo de título se desarrolló un nuevo sistema de gestión de coevaluaciones, el cual es robusto y fácil de utilizar. Este toma como referencia el sistema del departamento, sin embargo, posee un modelo de datos, funcionalidades y estructura propios, debido a que el propósito que tiene es diferente.

En el proceso de análisis y diseño se definieron historias de usuario que encapsularan las funcionalidades necesarias claves del sistema. Estas fueron fundamentales para la posterior validación del sistema, donde se revisaron las interfaces creadas verificando que estas se correspondieran con las historias de usuario. Además, se realizó una validación de usabilidad con alumnos, que presentó resultados positivos y que también entregó opciones de mejora del sistema.

*A mi gente.*

# Agradecimientos

Agradezco a mi familia, por permitirme ser quién soy.

A mis amigos, por su incondicionalidad.

A los múltiples profesores que encontré en mi camino, por su inspiración y guía.

Y a la profesora Jocelyn Simmonds, por permitirme ser su memorista, por su paciencia, consideración y guía.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Solución propuesta . . . . .	3
1.4. Metodología . . . . .	4
1.5. Estructura de la memoria . . . . .	4
<b>2. Estado del arte</b>	<b>5</b>
2.1. Sistema de Coevaluaciones relacionado . . . . .	5
2.2. Diseño de Interfaces Móviles . . . . .	7
2.3. Django . . . . .	11
2.4. Otras herramientas complementarias . . . . .	12
2.5. Resumen . . . . .	13
<b>3. Análisis y diseño de la solución</b>	<b>14</b>
3.1. Definición del problema . . . . .	14
3.1.1. Historias de usuarios . . . . .	14
3.1.2. Requisitos . . . . .	15
3.1.3. Flujo del sistema . . . . .	16
3.2. Solución propuesta . . . . .	17
3.2.1. Arquitectura y modelo de datos . . . . .	17
3.2.2. Interfaz de usuario . . . . .	24

3.2.3. Diseño de componentes . . . . .	30
3.3. Resumen . . . . .	32
<b>4. Implementacion</b>	<b>33</b>
4.1. Implementación de la aplicación web . . . . .	33
4.1.1. Back-end . . . . .	33
4.1.2. Front-end . . . . .	36
4.2. Interfaces . . . . .	37
4.2.1. Gestión de evaluaciones . . . . .	37
4.2.2. Formulario de coevaluación de alumnos . . . . .	43
4.2.3. Alumnos pendientes . . . . .	45
4.2.4. Resultados . . . . .	46
4.3. Funcionalidades importantes . . . . .	49
4.3.1. Envío de notificaciones a alumnos pendientes . . . . .	49
4.3.2. Importación de alumnos . . . . .	50
4.3.3. Exportación de resultados . . . . .	51
4.4. Resumen . . . . .	51
<b>5. Validación</b>	<b>52</b>
5.1. Validación de interfaz web . . . . .	52
5.1.1. Vista de ingreso de código . . . . .	52
5.1.2. Vista de respuesta de coevaluación para alumnos . . . . .	53
5.1.3. Vista de toma de asistencia para profesores . . . . .	53
5.1.4. Vista de creación de curso . . . . .	53
5.1.5. Vista de creación de evaluación . . . . .	54
5.1.6. Vista de envío de recordatorio . . . . .	55
5.1.7. Vistas de ficha de evaluación y resultados . . . . .	55
5.2. Validación del sistema con alumnos . . . . .	56

5.2.1. Metodología . . . . .	56
5.2.2. Resultados y análisis . . . . .	56
5.3. Resumen . . . . .	57
<b>6. Conclusión</b>	<b>58</b>
6.1. Discusión final . . . . .	58
6.2. Trabajo a futuro . . . . .	59
<b>Bibliografía</b>	<b>61</b>
<b>ANEXOS</b>	<b>62</b>

# Índice de Tablas

4.1. Ejemplo de archivo de alumnos. . . . .	50
---	----

# Índice de Ilustraciones

2.1. Home page profesor. . . . .	6
2.2. Parte de formulario de respuesta coevaluación. . . . .	6
2.3. Formulario para agregar una coevaluación. . . . .	6
2.4. Resultados de coevaluación desde perspectiva profesor. . . . .	7
2.5. Interfaz de introducción del código en aplicación Kahoot. . . . .	8
2.6. Interfaz de introducción del código en aplicación Mentimeter. . . . .	9
2.7. Interfaz de pregunta de formulario en Kahoot. . . . .	10
2.8. Interfaz de pregunta de formulario en Mentimeter. . . . .	10
2.9. Interfaz de pregunta de formulario en Slido. . . . .	11
3.1. Flujo de evaluaciones. . . . .	17
3.2. Arquitectura física del sistema. . . . .	18
3.3. Arquitectura lógica del sistema. . . . .	19
3.4. Diagrama del modelo de datos. . . . .	20
3.5. Diagrama del modelo de datos, sección relacionada con instrumentos y preguntas. . . . .	21
3.6. Diagrama del modelo de datos, sección relacionada con evaluaciones de alumnos. . . . .	23
3.7. Diagrama del modelo de datos, sección relacionada con respuestas. . . . .	24
3.8. Vista de ingreso de código de evaluación. . . . .	25
3.9. Vista de ingreso de código de evaluación (versión móvil). . . . .	26
3.10. Vista de formulario de evaluación. . . . .	26
3.11. Vista de formulario de evaluación (versión móvil). . . . .	27

3.12. Vista de toma de asistencia. . . . .	28
3.13. Vista de toma de asistencia (versión móvil). . . . .	28
3.14. Vista de ficha de una evaluación. . . . .	30
3.15. Vista de estado de una evaluación (versión móvil). . . . .	31
4.1. Interfaz de gestión de evaluaciones. . . . .	38
4.2. Vista de gestión de evaluación, sección información. Estado evaluación: creada. . . . .	39
4.3. Vista de gestión de evaluación, sección información. Estado evaluación: finalizada. . . . .	39
4.4. Vista de asistencia. . . . .	41
4.5. Vista de gestión de evaluación, sección grupos. . . . .	42
4.6. Vista de generación de grupos. Asistencia previamente ingresada. . . . .	42
4.7. Vista de generación de grupos. Asistencia no ingresada . . . . .	43
4.8. Interfaz de formulario de coevaluación de alumnos. . . . .	44
4.9. Interfaz de formulario de coevaluación de alumnos (versión móvil). . . . .	45
4.10. Interfaz de alumnos pendientes. . . . .	46
4.11. Interfaz de resultados de evaluación. . . . .	47
4.12. Tabla de resultados por grupo. Interfaz de resultados de evaluación. . . . .	48
4.13. Tabla de resultados por alumno. Interfaz de resultados de evaluación. . . . .	49
4.14. Botones de exportación de tabla de resultados. . . . .	51
5.1. Vista de ingreso de código. . . . .	53
5.2. Vista de creación de evaluación. . . . .	54
5.3. Vista de creación de evaluación. . . . .	55
1. Mockup de creación de evaluación. . . . .	62
2. Mockup de creación de curso. . . . .	63
3. Diagrama del modelo de datos. . . . .	64
4. Vista de importación de archivo de alumnos. . . . .	65
5. Vista de ingreso de código para alumnos (versión móvil). . . . .	66

6.	Tabla de evaluaciones. . . . .	67
7.	Tabla de cursos. . . . .	67
8.	Tabla de profesores. . . . .	67
9.	Tabla de alumnos. . . . .	68
10.	Ejemplo de email de recordatorio a alumno. . . . .	69

# Capítulo 1

## Introducción

### 1.1. Antecedentes

El trabajo colaborativo se define como personas que trabajan conjuntamente en una actividad o proyecto para lograr un objetivo común. Especialmente cuando esto implica la cooperación entre varios equipos, departamentos y/o diferentes organizaciones para lograr un objetivo acordado [1].

El 2014 un equipo de Stanford realizó diversos estudios sobre el trabajo colaborativo y la motivación. En ellos descubrieron que los participantes en la investigación preparados para actuar colaborativamente se mantuvieron en su tarea un 64 % más que sus compañeros solitarios, además mantuvieron niveles más altos de participación, niveles más bajos de fatiga y una mayor tasa de éxito. “Los resultados mostraron que el simple hecho de sentirse parte de un equipo de personas que trabajan en una tarea hace que las personas estén más motivadas a medida que asumen desafíos” mencionan Priyanka Carr y Gregory Walton [11].

Dentro de la malla de la carrera Ingeniería Civil en Computación existen ramos que requieren trabajo colaborativo para el desarrollo de software: Ingeniería de Software I, Ingeniería de Software II y Proyecto de Software. En ellos se forman equipos que trabajan en conjunto para desarrollar un proyecto. Al momento de la evaluación, se encuentra una sección de coevaluación en la que los integrantes deben calificarse entre ellos, en base a diversos factores predeterminados.

Para realizar las coevaluaciones de estos ramos existe un Sistema de Coevaluaciones [15], desarrollado por Riquelme [28] y Sánchez [29]. El sistema original presentaba falencias, por esto y debido a nuevos requisitos del sistema, es que este fue modificado el año 2020 por Carolina Contreras, estudiante que presentó esta actualización como su memoria de título [14]. Estos cambios fueron enfocados en los ramos de la carrera, e involucraron un dashboard de resultados y una función de seguimiento de estudiantes. Este proyecto aún no está en producción.

Actualmente, se necesita modificar el sistema para adaptarse a nuevas funcionalidades. Los cambios deben permitir que el proyecto se adecue a asignaturas de otras instituciones distintas

de la Universidad de Chile. Esto significa que se necesitarán nuevos enfoques, cambios al modelo de datos y nuevas interfaces.

La solicitud de cambio nace desde los psicólogos Patricio Cabello y Mauricio López, los cuales investigan el trabajo colaborativo, por lo que el sistema también cambiaría de usuario final. Para la parte administrativa del sistema los usuarios serían los investigadores, y en este caso los alumnos serían estudiantes de diversas carreras, no necesariamente ligadas a la computación, como se da en la situación actual del sistema.

En el sistema de Contreras, se pueden crear coevaluaciones asociadas a cursos. Todas las evaluaciones contienen las mismas preguntas que están predefinidas en la aplicación. También permite el manejo de grupos, importados por archivos o creados en el sistema. En este proyecto cada alumno evalúa a cada uno de sus compañeros del grupo. Finalmente, los resultados contemplan una interfaz de seguimiento de desempeño asociado a un estudiante particular.

Por otro lado, el sistema implementado en este trabajo permite crear instrumentos, los cuales poseen un conjunto de preguntas, para luego asociar evaluaciones a un curso e instrumento particular. Esto permite que, a diferencia del sistema anterior, las evaluaciones contengan un distinto conjunto de preguntas. Además, permite el manejo de usuarios, los cuales son profesores, administradores y alumnos. En el caso de alumnos, estos pueden ser creados individualmente o importados a través de un archivo.

Las nuevas funciones se basan en un cambio en el objetivo del sistema. En este momento, el enfoque del sistema del Departamento de Ciencias de la Computación de la Universidad de Chile (de siglas DCC) desarrollado por Contreras, es el de dar seguimiento a cada estudiante dentro del curso. La modificación considera enfocarse en el estudio del trabajo colaborativo, por lo que se busca que el sistema permita que los usuarios evalúen a su grupo de forma global, y no particular como en la aplicación anterior. El nuevo propósito también requiere de nuevas formas de presentar los resultados, que permitan a los investigadores conocer las conclusiones de los trabajos y tener una vista integral de cada curso.

## **1.2. Objetivos**

### **Objetivo General**

El objetivo general de esta memoria es crear un nuevo sistema de coevaluaciones, tomando como referencia el sistema actual de coevaluaciones del Departamento de Ciencias de la Computación de la Universidad de Chile, que fue específicamente diseñado para la carrera de Ingeniería Civil en Computación, para que se adapte a otros cursos de trabajo colaborativo. Esto a través del diseño y la implementación de un sistema que robusto y escalable para gestionar coevaluaciones.

## Objetivos Específicos

Con fin de cumplir el objetivo general, se establecen los siguientes objetivos específicos:

1. Analizar y conocer el sistema actual de coevaluaciones, entendiendo su modelo de datos y comprendiendo el código realizado.
2. Diseñar los cambios necesarios para implementar los nuevos requisitos, considerando que se podrían realizar cambios en el modelo de datos existente. Además de un diseño de las interfaces de usuario.
3. Implementar funcionalidades del sistema, para permitir que los administradores gestionen evaluaciones y los usuarios respondan coevaluaciones grupales.
4. Validar el nuevo sistema a través de métodos de evaluación de usabilidad con los diversos agentes involucrados, cómo estudiantes, profesores y psicólogos. Esto para conocer la aceptación de los usuarios y sus propuestas de mejora.

### 1.3. Solución propuesta

Para abordar los objetivos definidos se propuso una versión inicial de solución para el problema. La cual consistía en utilizar el sistema del DCC, extendiendo su modelo y modificando sus interfaces. No obstante, esto cambio debido a que durante el proceso de desarrollo fueron más visibles la diferencia entre ambos sistemas.

Es así, que la solución propuesta consiste en la creación de una aplicación de manejo de evaluaciones, desarrollada con el framework Django. Para esto se hará primero un análisis y diseño de la solución. Durante esta etapa, se definirá el problema, especificando sus requisitos. Posteriormente, se generará un modelo de datos que responda a las necesidades del proyecto, y se diseñarán interfaces de usuario que definan la estructura de navegación en el sistema.

Luego de la etapa de análisis y diseño, se realizará la implementación de las funcionalidades. Las cuales consideraran la gestión de cursos, grupos y evaluaciones para los administradores y profesores. Además de la gestión de usuarios e instrumentos que corresponderá solo para usuarios administradores.

Por otro lado, se considerará la función de respuesta de un formulario para usuarios alumnos que no poseen cuenta en el sistema, siendo esta una funcionalidad fundamental del proyecto.

Asimismo, se desarrollará el modulo de notificaciones que permita a los administradores enviar email de recordatorio a los alumnos pendientes de responder coevaluaciones, y el modulo de descarga de resultados, que permita a estos usuarios descargar las respuestas de los alumnos en cada coevaluación.

El sistema debe ser fácil de usar, escalable y robusto. Las funcionalidades deben también pasar pruebas funcionales y existirá un proceso de validación con usuarios.

## 1.4. Metodología

Con el fin de desarrollar el trabajo de memoria, se dividió el trabajo en distintas etapas. En la primera fase, se analizó con mayor detalle el sistema de coevaluaciones de Contreras, esto corresponde al objetivo específico 1 del proyecto.

Luego, para responder al objetivo específico 2, que se relaciona con el diseño de la solución, se especificaron historias de usuario y requisitos que definieran el problema. Además de el diseño de las interfaces de usuario.

En la siguiente etapa se realizaron modificaciones al sistema existente. Es durante esta fase que se decidió separar el nuevo sistema y crear un modelo de datos propio que no se extendiera del sistema del DCC. Después de esto, se implementaron las funcionalidades del sistema, al mismo tiempo que se realizaron pruebas de que estas funcionaran correctamente, creando datos de prueba y verificando que los resultados fueran los esperados.

Posteriormente, se realizó una validación del sistema con usuarios y en la última fase se realizó la escritura del informe.

## 1.5. Estructura de la memoria

En los capítulos siguientes se presentan los resultados conseguidos durante esta memoria. Estos están organizados de la manera que se detallará a continuación.

- El capítulo 2 presenta el marco teórico de la memoria, donde se presentan los conocimientos que se necesitan para entender el documento. Dentro de estos se presenta el sistema de coevaluaciones que se utiliza como referencia, aplicaciones relacionadas y herramientas utilizadas en el proyecto.
- En el capítulo 3 se encuentra el análisis y diseño de la solución. Aquí, se muestran las arquitecturas del sistema, el modelo de datos de este y los mockups de interfaces.
- Luego, en el capítulo 4 se describe la implementación del sistema. En el comienzo se muestra como el proyecto está estructurado y posteriormente se detalla el desarrollo del proyecto.
- El capítulo 5 muestra las validaciones que se realizaron durante el trabajo. Las metodologías que se utilizaron en estas validaciones y los resultados obtenidos.
- Para finalizar, el capítulo 6 aborda las conclusiones de la memoria, y muestra posibles trabajos futuros para mejorar el sistema.

# Capítulo 2

## Estado del arte

En este capítulo se presenta la aplicación que sirvió de inspiración para el trabajo realizado, además de otras aplicaciones cuyas interfaces fueron utilizadas como base y modelos a seguir para el desarrollo posterior. Finalmente, se muestran herramientas complementarias, las cuales fueron estudiadas para luego ser incluidas en el proyecto y de tal forma satisfacer todas las funciones que este considera.

En la sección 2.1 se presenta el sistema de coevaluaciones de Contreras, luego en la sección 2.2 se habla sobre el diseño de interfaces móviles mostrando algunas aplicaciones investigadas que sirven como modelos para el trabajo realizado. En la sección 2.3 se muestra Django, framework en que se desarrolla el sistema. Para terminar, la sección 2.4 presenta otras tecnologías consideradas para la implementación del proyecto.

### 2.1. Sistema de Coevaluaciones relacionado

El Sistema de Coevaluaciones actualizado por Carolina Contreras [14], se basa en los requisitos de los ramos de la carrera. Es por esto que el principal enfoque es el de realizar seguimiento a los grupos en las asignaturas de trabajo colaborativo. Lo que significa que el profesor utiliza esta herramienta para ver cuál es la situación de cada alumno dentro de cada grupo, en sus distintos cursos. La página inicial del profesor se puede ver en la Figura 2.1. En esta interfaz se presenta al lado izquierdo la gestión de coevaluaciones y al derecho la gestión de cursos. En ella además el profesor puede dirigirse al seguimiento de desempeño de alumnos.

En este contexto es necesario que cada estudiante evalúe a cada uno de sus compañeros, como se muestra en la Figura 2.2 que representa la interfaz de un usuario estudiante que evalúa a una participante de su grupo en este caso de ejemplo llamada Tamara Parra. La coevaluaciones que está siendo respondida es una de las que se muestra en la lista de Coevaluaciones de la figura 2.1. De esta forma se obtienen resultados asociados a un estudiante particular y así permiten al profesor ver la situación de sus alumnos y tomar medidas al respecto.

Sistema de Coevaluaciones DCC Carmen Uribe Fernández [Cerrar Sesión](#)

**Coevaluaciones** Seguimiento de Desempeño Agregar coevaluación

Fecha inicio	Nombre	Curso	Semestre	Fecha límite	Estado
12/04/2021 00:00	Sprint 2	Ingeniería de Software CC4401-2	2021-1	26/04/2021 03:59	<span style="background-color: #28a745; color: white; padding: 2px 5px; border-radius: 3px;">Publicada</span>
16/06/2021 00:00	Sprint 3	Ingeniería de Software CC4401-2	2021-1	30/06/2021 23:59	<span style="background-color: #6c757d; color: white; padding: 2px 5px; border-radius: 3px;">Creada</span>
10/03/2021 00:00	Sprint 1	Ingeniería de Software CC4401-2	2021-1	24/03/2021 23:59	<span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Respuestas publicadas</span>

**Tus Cursos** Agregar curso

Curso	Semestre
Ingeniería de Software CC4401-2	2021-1
Ingeniería de Software CC4401-1	2020-2

Figura 2.1: Home page profesor.

### Responder coevaluación

*Estás respondiendo para:*  
**Tamara Parra Olivares**

1. Demuestra compromiso con el proyecto.  
 1     2     3     4     5
2. Cumple de manera adecuada con las tareas que le son asignadas.  
 1     2     3     4     5

Figura 2.2: Parte de formulario de respuesta coevaluación.

El sistema se encuentra diseñado para que los profesores puedan crear nuevas coevaluaciones, especificando el curso asociado, y las fechas de inicio y de término de la misma (como se muestra en la Figura 2.3). Es importante mencionar que en esta sección el profesor no puede especificar el formulario (las preguntas de la coevaluación) ni las ponderaciones, ya que las preguntas están previamente definidas y son específicas al caso particular de los ramos de la carrera.

### Agregar Coevaluación

**Título de la coevaluación**

**Curso**

Seleccione un curso...
v

**Fecha de inicio**

**Hora de inicio**

**Fecha límite**

**Hora límite**

Autoevaluación

Confirmar
Cancelar

Figura 2.3: Formulario para agregar una coevaluación.

En la sección de reportes el profesor puede ver los resultados de las coevaluaciones asociadas a un alumno particular (Figura 2.4). Cabe destacar que no existen reportes en base a grupos o cursos completos en el sistema existente, ya que no aportan al propósito particular para el que fue diseñado.

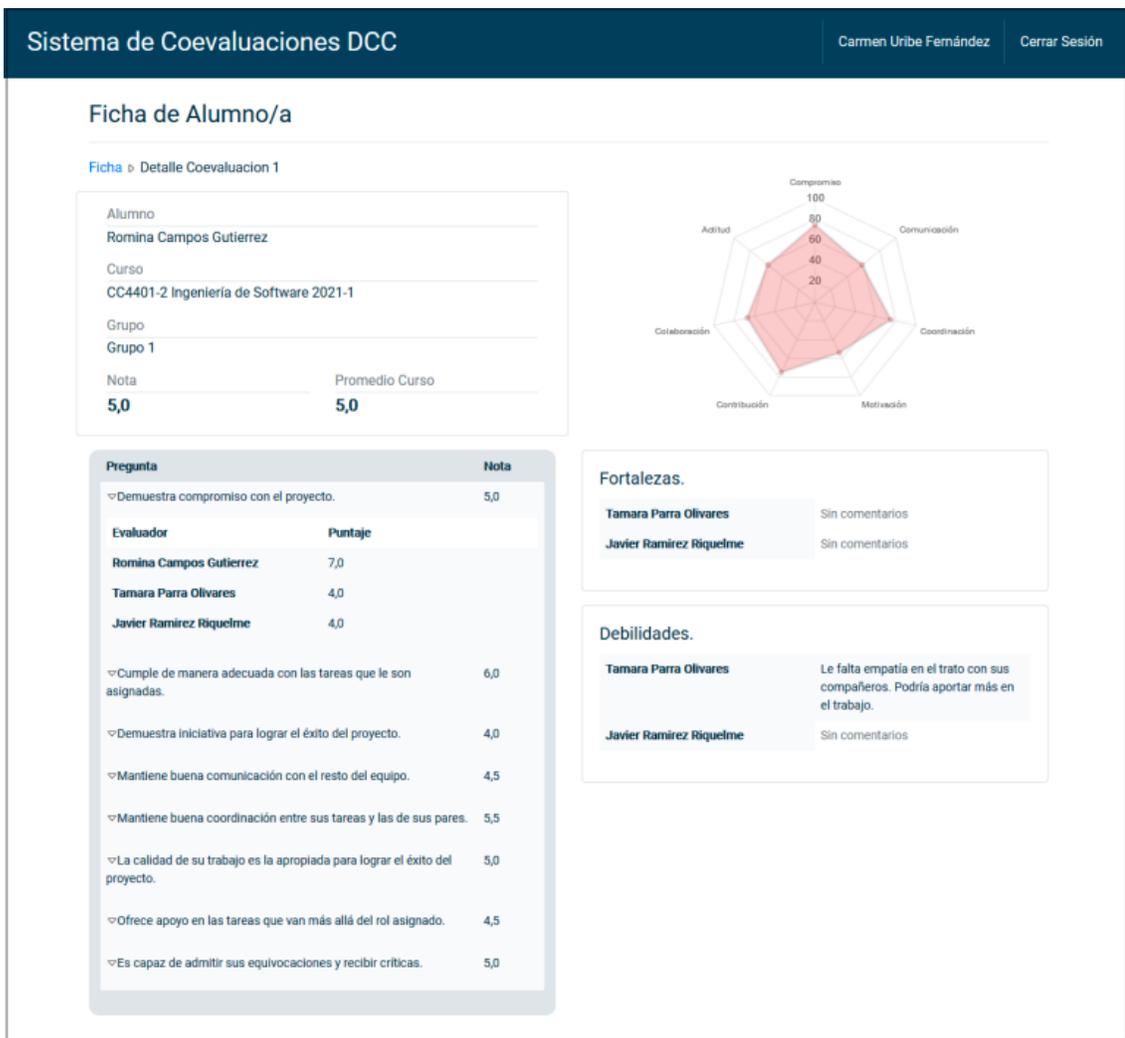


Figura 2.4: Resultados de coevaluación desde perspectiva profesor.

## 2.2. Diseño de Interfaces Móviles

Actualmente el sistema de coevaluaciones está conformado por interfaces diseñadas para computadores. El diseño se realizó pensando en los estudiantes de la carrera ingeniería civil en computación, por lo que se tenía una noción más certera de que estos tuvieran acceso a un computador para acceder a la coevaluación. En el nuevo contexto, en que las evaluaciones serán respondidas por alumnos de diversas carreras de institutos profesionales, por lo que se presenta un mayor flujo de usuarios. Es así que nace la necesidad de estudiar el diseño para dispositivos móviles, y de esta forma poder facilitarle el proceso de respuesta del formulario a los alumnos.

Los teléfonos móviles son los dispositivos más usados para acceder a la web en Chile, con un 57.8% del tráfico web [8]. Tomando esto en consideración, y que estos dispositivos son más portátiles para asistir a clases, es que se decide crear nuevas interfaces del sistema para dispositivos móviles.

Previo al diseño de las nuevas interfaces que formará parte del trabajo, se estudian aplicaciones que actualmente tengan como propósito responder formularios en dispositivos móviles. Entre ellas se encuentra Kahoot [3], Mentimeter [4] y Slido [6].

Kahoot es una aplicación diseñada para ayudar al estudio en conjunto, donde se pueden crear cuestionarios de manera rápida y compartirla con otras personas para participar de ellas. Existen diversas maneras de compartir la encuesta, una de ellas es la que se ve en la figura 2.5, donde cada persona debe ingresar un código en su celular para ingresar.

En el contexto del sistema de evaluación de trabajo colaborativo resulta útil el método mencionado, ya que la persona encargada de la evaluación podría compartir este código con los estudiantes presentes. Sin embargo, la desventaja de este método es que no se puede controlar quién responde, por lo que queda en manos del alumno ingresar su identificación, que en este caso sería su correo electrónico. Lo anterior puede contribuir a errores, por lo que se debe considerar esta opción de ingreso y la manera de manejar potenciales errores.

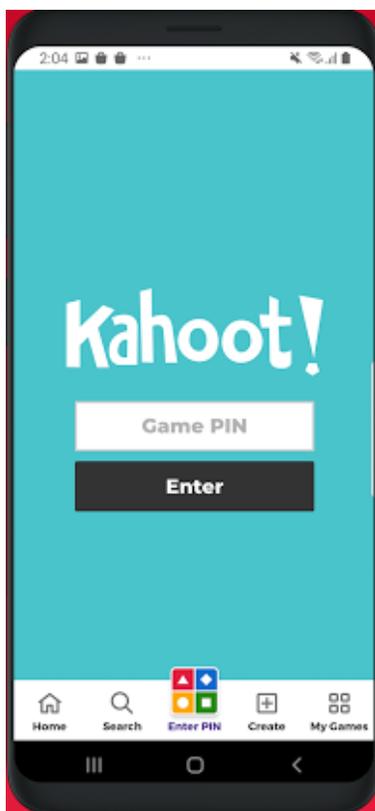


Figura 2.5: Interfaz de introducción del código en aplicación Kahoot.

Otra aplicación que posee la opción de introducir un código para ingresar a la evaluación es Mentimeter, sistema dedicado a crear presentaciones y reuniones interactivas. En la figura 2.6 se muestra la interfaz para ingresar el código, en esta aplicación este es único, es decir, se genera uno que puede ser utilizado por cualquier participante que quiera entrar a la presentación. Aunque, para los objetivos del trabajo se podría utilizar esta interfaz, pero se deberían emitir distintos códigos (uno por cada usuario) que sean enviados al email de los alumnos presentes.

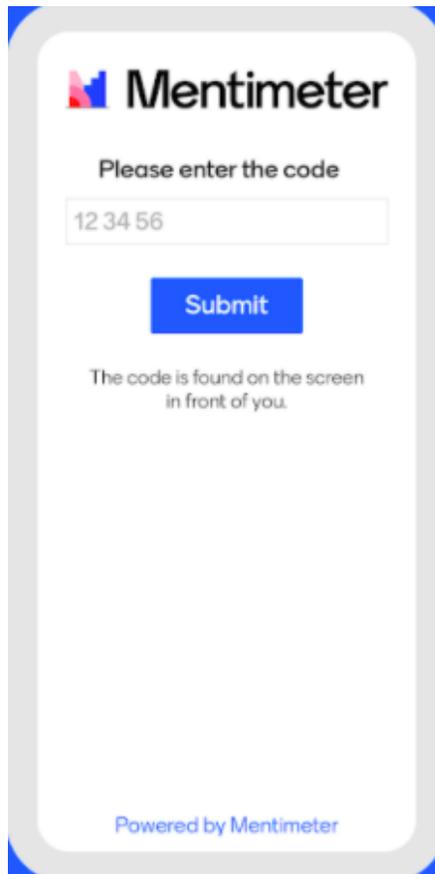


Figura 2.6: Interfaz de introducción del código en aplicación Mentimeter.

Al momento de mostrar las preguntas estas aplicaciones cuentan con interfaces que muestran una pregunta a la vez, al responder marcando una de las opciones de respuestas, se dirige a la siguiente pregunta. Esto hace que el usuario vea una pregunta a la vez y por lo tanto exista una mayor atención hacia la pregunta que responden, disminuyendo los potenciales errores de ubicación en el formulario.

En Kahoot las opciones aparecen en cuatro bloques, donde cada bloque representa una respuesta. Estos se encuentran en filas donde cada fila posee dos bloques (figura 2.7). El diseño de esta interfaz no se adapta al sistema en cuestión, ya que las opciones de respuesta de los instrumentos presentados se encuentran en una escala que posee orden (para nada, un poco, parcialmente, en gran medida, totalmente). Entonces, visualmente se adecua más que estas opciones se presenten por el orden que contienen. Para esto se adecúa mejor la interfaz de pregunta que presenta Mentimeter (figura 2.8) y Slido (figura 2.9).

Una de las principales diferencias entre las interfaces de pregunta de formulario de Mentimeter y Slido es el control de input que posee el segundo, es decir, el círculo que se rellena al seleccionar la opción. Otra disparidad que tienen es que Slido tiene un botón de envío, lo que significa que en este caso se requiere que el usuario tenga una interacción extra cuando se encuentre listo con su respuesta.



Figura 2.7: Interfaz de pregunta de formulario en Kahoot.

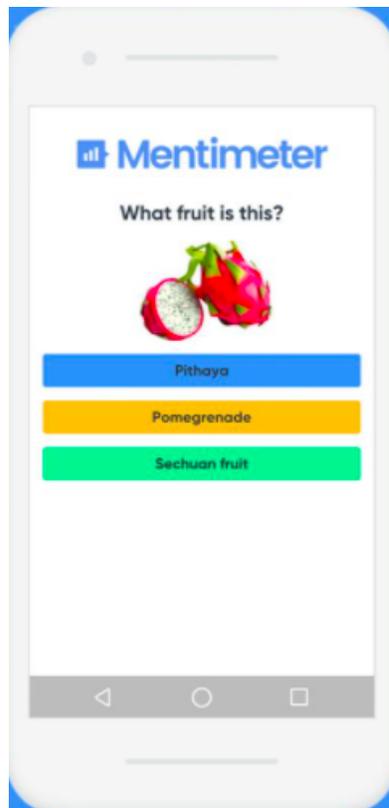


Figura 2.8: Interfaz de pregunta de formulario en Mentimeter.

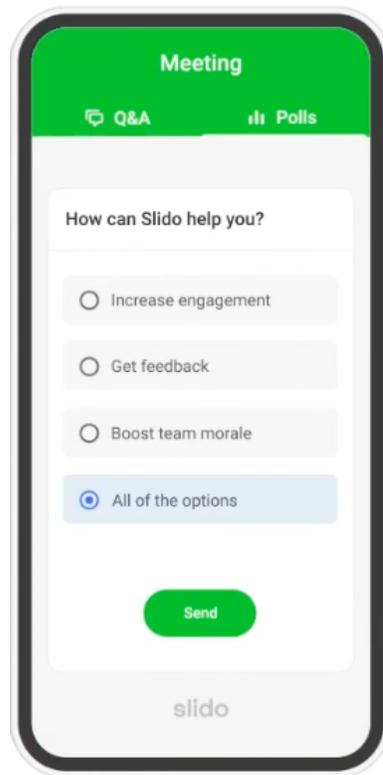


Figura 2.9: Interfaz de pregunta de formulario en Slido.

Luego de conocer el sistema actual y las funcionalidades que este integra para la gestión de coevaluaciones y el seguimiento de desempeño de estudiantes, se tiene como finalidad extender estas funciones para poder adaptarse a otros cursos, distintos a los de la carrera de Ingeniería Civil en Computación. Con respecto a las interfaces actuales, estas fueron diseñadas pensando en el acceso desde computadoras y no pensando en un diseño adaptable a otros dispositivos, por lo que otro propósito del trabajo consiste en diseñar e implementar interfaces de diseño responsivo, teniendo en consideración los diseños presentados por las aplicaciones Kahoot y Mentimeter.

## 2.3. Django

Django es un framework de alto nivel escrito en python que se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto, tiene una comunidad activa y una gran documentación [22].

Los frameworks de lado servidor, es decir, frameworks de aplicaciones web, tales como Django, son frameworks software que hacen más fácil escribir, mantener y escalar aplicaciones web. Proporcionan herramientas y bibliotecas que simplifican tareas comunes de desarrollo web, incluyendo enrutado de URLs a los manejadores apropiados, interacción con bases

de datos, soporte de sesiones y autorizaciones de usuario [21].

Se podría decir que django es un framework con arquitectura MVT (Modelo-Vista-Template)[18]. Esta es similar a la arquitectura MVC (Modelo-Vista-Controlador). MVC es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización, la que proporciona una mejor división del trabajo y una mejora de mantenimiento [23].

Las partes de MVC son: **Modelo** para manejar los datos, **Vista**, que se encarga del diseño, o front-end y **Controlador**, que es el intermediario entre los comandos del usuario y del modelo de datos. De manera similar, MVT considera 3 principales componentes lógicos, siendo estos **Modelo**, **Vista** y **Templates**.

La principal diferencia entre MVC y MVT es que en el primero debemos escribir código específico del controlador, pero en el MVT el framework se encarga de este código. En MVC, cuando un usuario realiza una acción o realiza una solicitud, se llama a una acción en el controlador. Luego, el controlador le dice al modelo que realice cambios y actualice la vista o devuelve una vista basada en un modelo. Sin embargo, en MVT la vista funciona de manera diferente que en MVC, es esta capa la que acepta las solicitudes de los usuarios, realiza una consulta en el Modelo y recopila el conjunto de resultados del Modelo. La Vista luego llena el resultado en una plantilla (componente Template) y lo envía al usuario [12].

La organización de MVT en django funciona en base a los siguientes componentes:

- **Modelos:** objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar y consultar registros.
- **Vista:** una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Estas acceden a los datos necesarios para responder a las peticiones por medio de modelos, y delegan el formateo de la respuesta a los templates o plantillas.
- **Plantillas (Templates):** fichero de texto que define la estructura de otro fichero (tal como una página HTML), con marcadores de posición que se utilizan para representar el contenido real. Las vistas pueden crear una página de forma dinámica utilizando una plantilla y luego rellenándola con datos de un modelo.
- **URLs:** es una nueva componente que se encarga de recibir peticiones y redirigirlas a la vista apropiada según la URL de la petición. El mapeador URL puede también emparejar patrones de cadenas o dígitos específicos que aparecen en una URL y los pasan a la función de visualización como datos [22].

## 2.4. Otras herramientas complementarias

Para lograr implementar todas las funciones del proyecto, se consideraron herramientas complementarias al framework ya mencionado: Django, y a los lenguajes de programación

Python y JavaScript. En esta sección se detallan las herramientas consideradas y sus usos en la aplicación.

Primero, se utilizó **JQuery**, una librería JavaScript que simplifica cosas como la manipulación de documentos HTML, el manejo de eventos y Ajax [25].

Otra librería utilizada fue **Bootstrap 5**. Bootstrap es un framework para crear sitios adaptables para dispositivos móviles [10]. Bootstrap 5 es la versión más nueva del framework. Este fue utilizado debido a que su popularidad hace que sus componentes sean conocidos por los usuarios. Además, se usó para asegurarse que el sitio sea adaptable para dispositivos móviles.

La librería **django-bootstrap5** fue añadida para juntar Django con Bootstrap 5, su objetivo es combinar a la perfección Django y Bootstrap 5 [9], su principal uso para el proyecto es en los formularios.

A pesar de que con Bootstrap las vistas se adaptarían al dispositivo que ingresa al sistema, existen interfaces, tales como la de evaluación del alumno, que debiesen tener otra interfaz de usuario. Para esto, se utiliza **user-agents**, una librería de python que provee una forma fácil de identificar dispositivos como teléfonos móviles, tablets y computadores y sus capacidades, al analizar strings de agentes de usuarios (de navegadores web) [26].

Para agregar funciones avanzadas a las tablas de las vistas, se incluyó **DataTables**, que es un plug-in para JQuery, altamente flexible, que agrega funciones avanzadas a cualquier tabla HTML [30].

Otra librería considerada es **psycopg2**, que se usa para conectar Python con las bases de datos PostgreSQL.

Para los modelos se considera la librería **django-polymorphic**, que simplifica el uso de modelos heredados en proyectos de Django [13]. Esta librería tiene utilidad para definir subclases en el modelo de datos, como por ejemplo para definir los distintos tipos de integrantes del sistema.

## 2.5. Resumen

En esta sección se presentó el sistema de coevaluaciones relacionado con el proyecto. Además se presentaron distintos diseños de interfaces móviles que se tienen como referencia para el diseño de interfaces. También se presentó el framework Django y sus componentes. Finalmente, se presentaron otras herramientas utilizadas en la aplicación.

# Capítulo 3

## Análisis y diseño de la solución

Como se muestra en los capítulos anteriores, el objetivo de este trabajo es implementar una plataforma que sirva para manejar evaluaciones colaborativas. Para poder definir y detallar este problema se muestran en la sección 3.1 las historias de usuario que el sistema debe realizar, además de los requisitos que se deben cumplir. Luego, en la sección 3.2 se presenta la solución propuesta para resolver el problema declarado.

Con el fin de crear e implementar el proyecto, es que se vuelve necesaria una arquitectura física en la cual se pueda correr la aplicación, además de una arquitectura lógica que permita planear el diseño del sistema antes de su implementación. Por otro lado, para poder organizar como los datos serán almacenados en la arquitectura física, se desarrolla un modelo de datos.

Tanto las arquitecturas como el modelo de datos se muestran en este capítulo dentro de la sección de solución propuesta. Además, la solución presentada también considera el diseño de interfaces con las que el usuario interactuará con la aplicación, estas se encuentran en la sección 3.2.2.

### 3.1. Definición del problema

El problema fue definido primeramente por las historias de usuario que detallan lo que los usuarios esperan de las distintas funcionalidades del sistema. Luego, se especificaron los requerimientos que el sistema debe cumplir. Finalmente se presenta el flujo del sistema, considerando los procesos más fundamentales de este.

#### 3.1.1. Historias de usuarios

Los usuarios de estas historias corresponden a alumnos que ingresan al sistema para responder coevaluaciones, además de profesores y administradores que tienen acceso a la administración de coevaluaciones, teniendo estos últimos además acceso a funcionalidades exclusivas.

**HU01:** Un usuario quiere crear un curso, agregando alumnos a este.

**HU02:** Un usuario profesor quiere iniciar una evaluación, asociándola a un curso y un instrumento.

**HU03:** Un usuario profesor quiere registrar la asistencia de una evaluación para generar grupos aleatorios en base a ella.

**HU04:** Un usuario alumno quiere ingresar al formulario para responder la coevaluación.

**HU05:** Un usuario alumno quiere completar el formulario y enviar sus respuestas al sistema.

**HU06:** Un usuario administrador quiere enviar un email de recordatorio a los alumnos pendientes que aún no responden la evaluación.

**HU07:** Un usuario administrador quiere exportar los resultados de una evaluación finalizada.

### 3.1.2. Requisitos

En base a lo especificado por los clientes y las historias de usuario definidas, se definen los siguientes requisitos que el sistema debe asegurarse de cumplir.

**R01:** Un alumno sin cuenta debe poder identificarse en el sistema con un token de acceso asociado a su correo electrónico.

**R02:** Un alumno sin cuenta debe poder ver y responder una evaluación.

**R03:** Un profesor debe poder iniciar y finalizar evaluaciones.

**R04:** Un profesor puede editar los alumnos de un curso.

**R05:** Un profesor debe poder registrar asistencia de una evaluación.

**R06:** Un profesor debe poder crear grupos aleatorios en base a la asistencia de la evaluación.

**R07:** Un profesor puede editar grupos, agregando y eliminando alumnos de este.

**R08:** Los administradores pueden ingresar, editar y eliminar nuevos instrumentos.

**R09:** Se deben poder crear evaluaciones, asociadas a un instrumento y a un curso.

**R10:** Los administradores pueden gestionar evaluaciones.

**R11:** Los administradores pueden descargar los resultados de las evaluaciones para cada grupo.

**R12:** Los profesores y administradores deben poder ver el estado de las evaluaciones y la cantidad de alumnos pendientes por responderlas, pudiendo enviar una notificación.

**R13:** La aplicación debe garantizar la adecuada visualización en múltiples computadores personales, y teléfonos inteligentes.

**R14:** La aplicación debe ser escalable. Debe ser posible escalar la carga de usuarios a 2.000 usuarios concurrentes sin que el sistema presente caídas.

El requisito R01, que especifica la forma en que un usuario sin cuenta puede identificarse en el sistema, responde a la historia de usuario HU04, en la que el usuario quiere ingresar al formulario para responder la coevaluación. De esta misma forma, el requisito R02 se corresponde con la historia HU05, los cuales responden a la necesidad de un alumno de completar un formulario y enviar sus respuestas. Del mismo modo, el requisito R03 y R10 y la historia HU02 consideran la función de crear evaluaciones e iniciar evaluaciones.

Por otro lado, la historia HU01 menciona la función de un usuario de crear un curso agregándole alumnos, esto se considera en el requisito R04. Así mismo, en HU03 se considera la necesidad de un profesor de registrar la asistencia de una evaluación, contemplado en R05, para luego generar grupos aleatorios en base a ella, considerado en R06.

Para cubrir la historia HU06, en que un administrador quiere enviar un email de recordatorio, existe el requisito R12, y para la historia sobre exportación de resultados, HU07, se encuentra el requisito R11.

Finalmente los requisitos R13 y R14, corresponden a requisitos no funcionales que se preocupan sobre la correcta visualización de la aplicación en distintos dispositivos y la escalabilidad del sistema.

### **3.1.3. Flujo del sistema**

El proyecto está compuesto por diversos procesos que detallan el funcionamiento del sistema. Uno de los flujos principales es el de las evaluaciones, el que se detalla en esta sección.

#### **Flujo de evaluaciones**

Para mostrar este flujo se presenta el diagrama de la figura 3.1. En el primer paso se verifica que exista un instrumento, el cual define un conjunto de preguntas, para asociar a la evaluación. De no estar presente, este es creado. Luego, se puede ver el proceso de una evaluación desde que es creada, pasando por diversas etapas hasta que los resultados se encuentran disponibles.

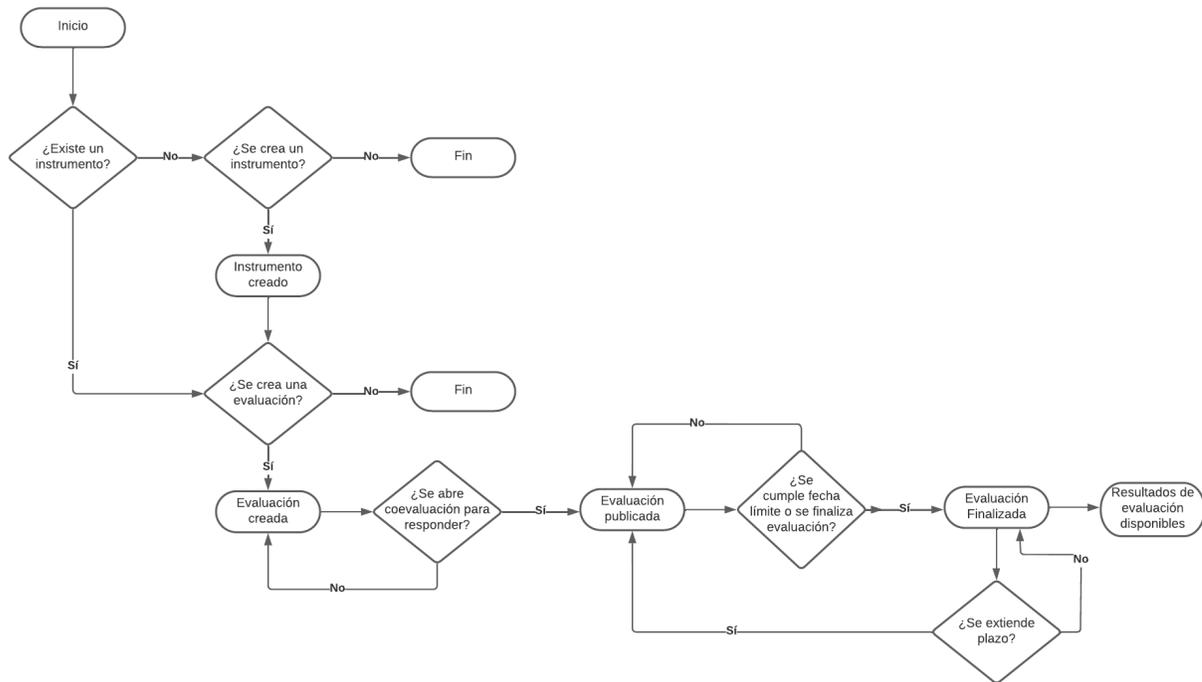


Figura 3.1: Flujo de evaluaciones.

## 3.2. Solución propuesta

### 3.2.1. Arquitectura y modelo de datos

En esta sección se expondrán las arquitecturas físicas y lógicas del sistema. Asimismo, se presentará el modelo de datos del proyecto.

Para realizar el proyecto, se estableció la arquitectura física presentada en la figura 3.2. El backend considera a Django 4.0.6 como framework, debido a que el proyecto de Coevaluaciones[14] de referencia para este sistema lo utiliza, además de los conocimientos que se tienen de él, ya que se enseña durante la carrera de Ingeniería Cívil en Computación en la FCFM. Este framework se monta en un servidor web con un ambiente virtual de Python 3.8.5.

Django utiliza de forma predeterminada el sistema de gestión de bases de datos relacional SQLite, esto ya que es la opción más fácil para usuarios nuevos en bases de datos [7]. Sin embargo, este fue reemplazado por PostgreSQL, también soportado por Django. Esta elección se basa en que es de código abierto y presenta buenos resultados frente a grande volúmenes de datos, lo que es importante para facilitar la escalabilidad del sistema.

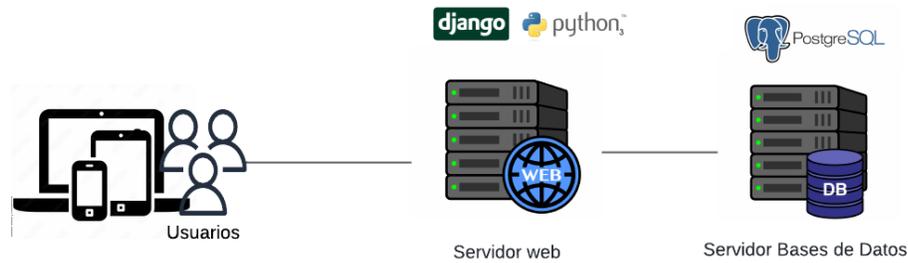


Figura 3.2: Arquitectura física del sistema.

La arquitectura lógica presentada en la figura 3.3, se basa en el patrón de arquitectura Modelo Template Vista o MTV, que es el usado por Django [2]. Podemos ver que la capa que se conecta con la bases de datos es la de DAO (objeto de acceso de datos), que contiene la lógica de acceso a datos, es decir, presenta los métodos necesarios para manejar la información en la bases de datos.

En esta arquitectura podemos ver los módulos dentro de los controladores encargados de administrar la gestión de coevaluaciones, integrantes, profesores, alumnos, grupos y respuestas.

También existe el módulo de autorización para manejar los permisos de los usuarios y el de asistencia para poder manejar la información de alumnos presentes en cada evaluación.

El módulo de notificaciones es el encargado del envío de recordatorios por correo electrónicos a los alumnos pendientes que no han respondido la coevaluación a la fecha.

Por otro lado, el módulo de dashboard cuenta de tablas que presentan los datos de las evaluaciones al usuario.

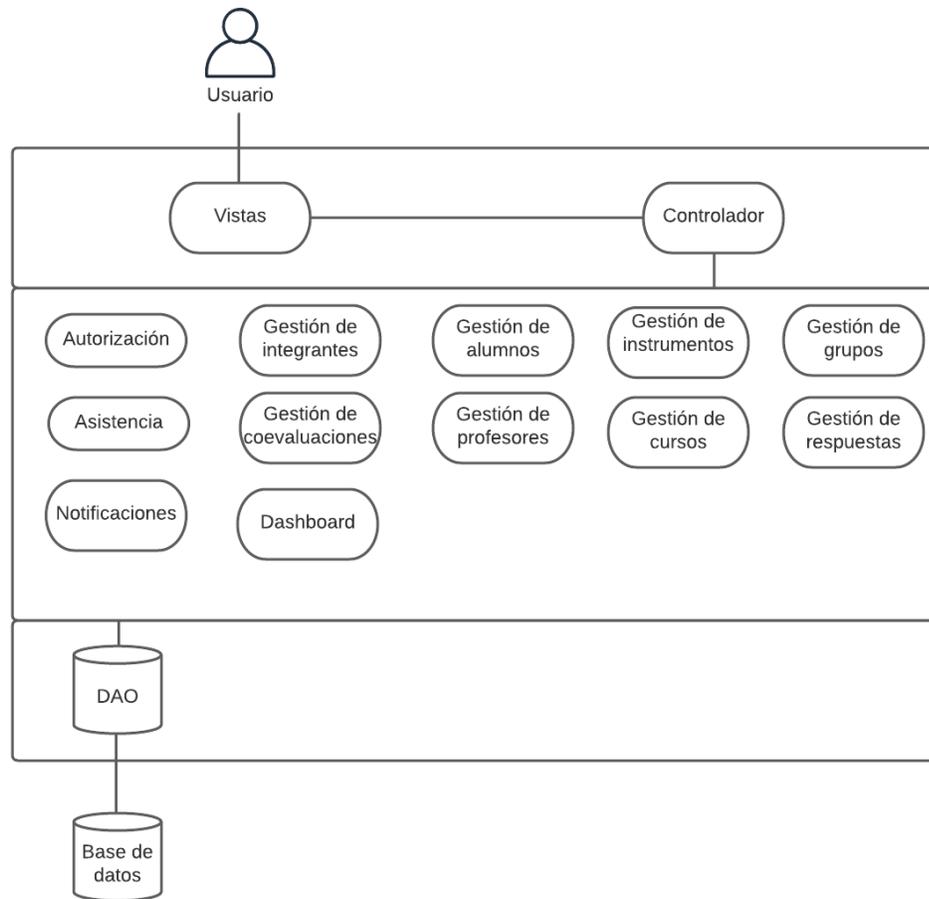


Figura 3.3: Arquitectura lógica del sistema.

Al comenzar el trabajo, se decidió extender el modelo de datos perteneciente al sistema de coevaluaciones utilizado como referencia para el proyecto (sección 2.1). Esto debido a que ambos sistemas consideran el manejo de evaluaciones, por lo que tienen elementos en común, tales como evaluaciones, preguntas, respuestas, cursos, entre otros. Sin embargo, a pesar de sus similitudes, los sistemas tienen diferentes procesos y fines, por lo que a medida que se avanzaba en el trabajo los cambios realizados al antiguo modelo crecían y se decidió crear un nuevo modelo de datos específico para esta aplicación.

El nuevo modelo de datos, que por temas de tamaño este se encuentra completo en el anexo 3, considera 22 clases, además de la clase **Usuario** que extiende la clase **User** nativa de Django. Esta extensión se realiza para agregar el campo **rol**, cuyas opciones son **Profesor** y **Administrador**, lo que sirve para manejar los permisos dentro de la aplicación.

Existen subconjuntos de clases del modelo, dentro de los cuales las clases se relacionan entre sí. Estos serán separados en distintas figuras para facilitar la lectura del modelo. En la figura 3.4 se puede ver una versión reducida del modelo de datos, en la cual este no cuenta con todas sus clases. Las clases que no están aquí, estarán presente en otras figuras posteriores y se mencionarán cuando se presente tal fragmento del modelo.

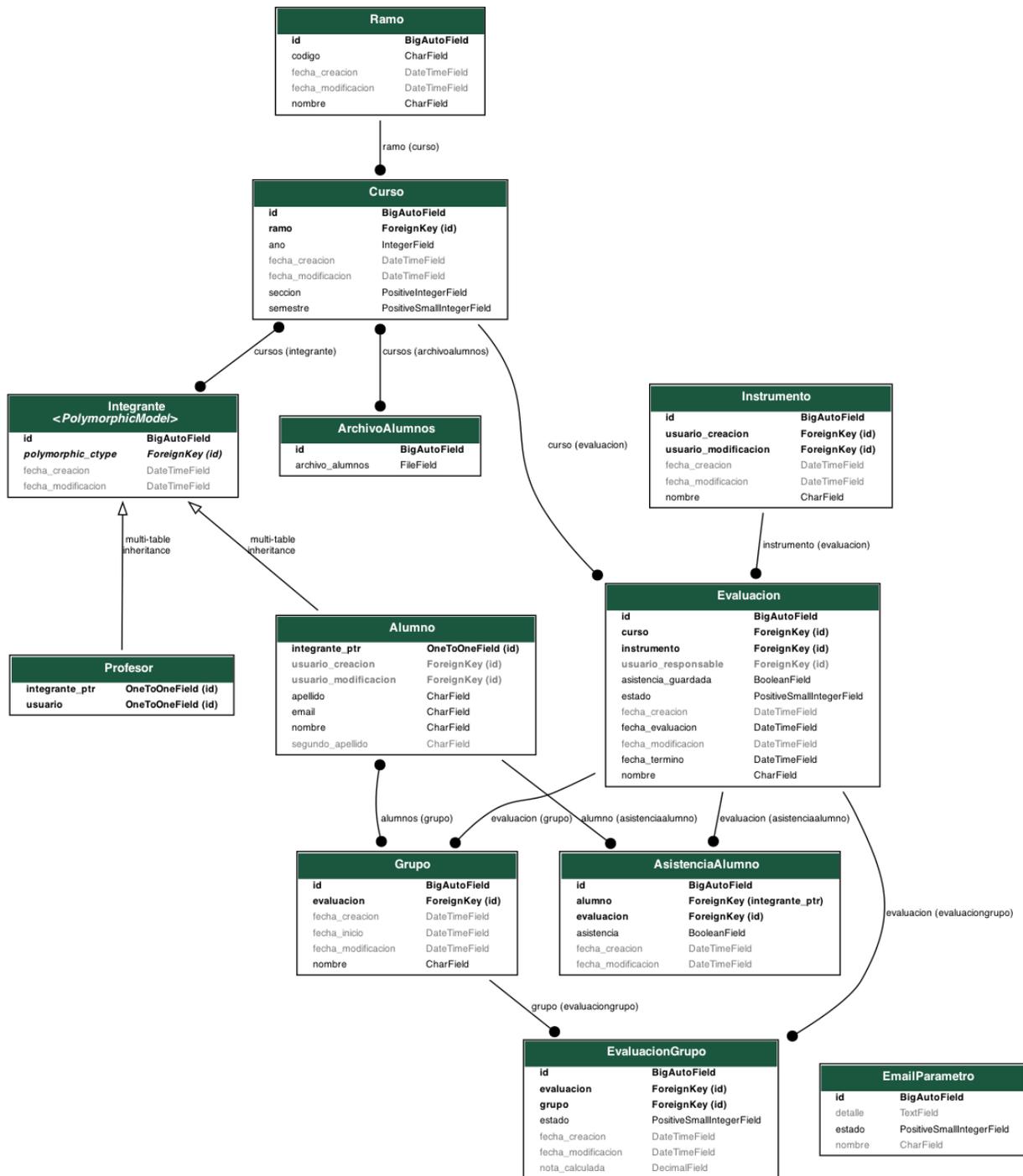


Figura 3.4: Diagrama del modelo de datos.

Como se ve en la figura 3.4, dentro del modelo se encuentra la clase **Ramo**, que asocia un código con un nombre. La clase **Curso** está constituida por un **Ramo**, además de los campos de año, semestre y sección. Un **Curso** podría ser visto como un detalle de un **Ramo**.

Por su parte, la clase **Integrante** está compuesta de cursos (**Curso**). Existen dos subclases de esta, las cuales son **Profesor** y **Alumno**. La diferencia radica en que los profesores tienen asociado un **Usuario**, ya que estos integrantes poseerán cuenta en el sistema para ingresar

a los módulos de gestión, y por otro lado los alumnos no tendrán cuenta en el sistema. Estos últimos tienen los campos de **nombre**, **apellido** y **segundo apellido** para mantener la información de alumnos, y también el campo de **email**, el cual tiene una restricción de unicidad (unique constraint), lo que significa que es el email el que identifica cada alumno en el sistema.

La clase **ArchivoAlumnos** fue agregada debido a la función de importar alumnos, la cual permite que profesores o administradores suban un archivo que contenga una lista de alumnos, para que estos sean agregados a la aplicación. Esta clase contiene el archivo subido y lo asocia con cursos(**Curso**).

Por otro lado, está la clase **Instrumento**, la cual se muestra en la figura 3.5, que representa un conjunto de preguntas (**Pregunta**), las cuales pueden ser de dos tipos, preguntas de texto, definidas en la subclase **PreguntaTexto** y preguntas con opciones, definidas en la subclase **PreguntaOpcion**. Las últimas contienen un conjunto de opciones (**ConjuntoOpcion**), el que agrupa distintas opciones (**Opcion**). Las opciones definen un texto, que representa la opción, y un valor, que se usa para calcular la nota o puntuación de las evaluaciones.

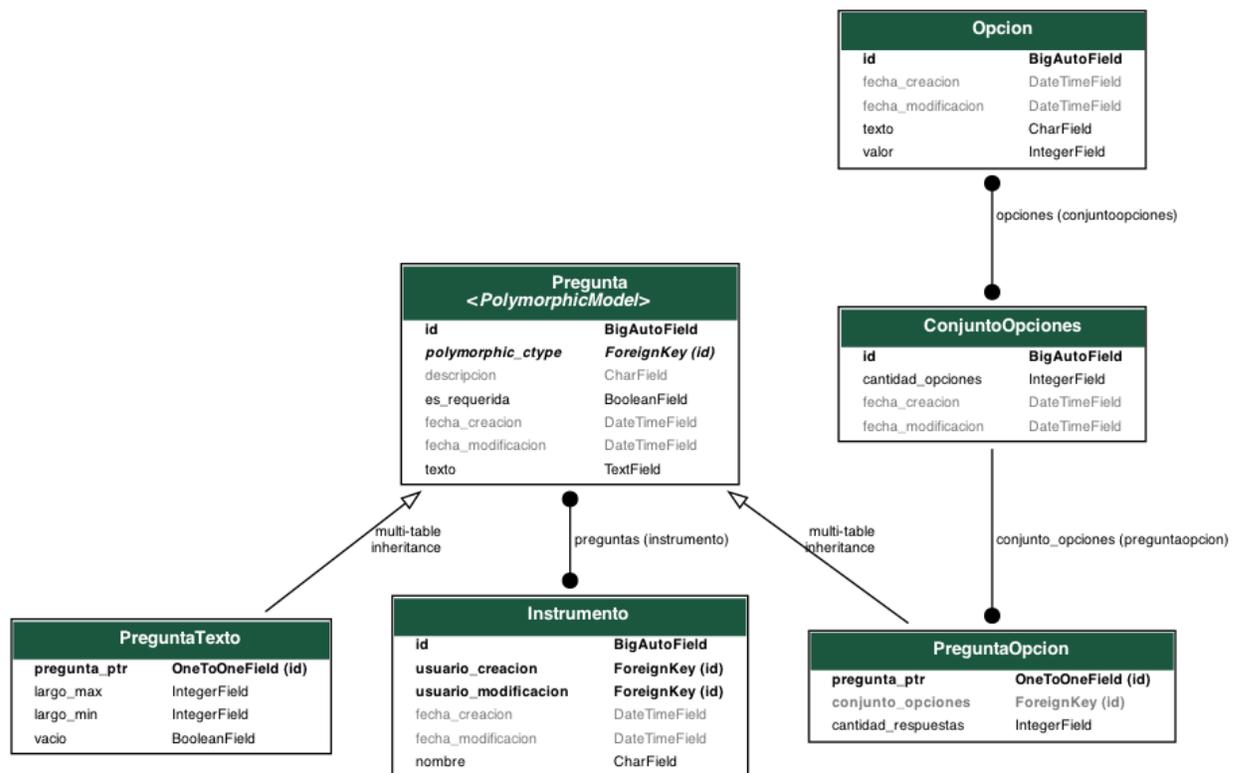


Figura 3.5: Diagrama del modelo de datos, sección relacionada con instrumentos y preguntas.

Una **Evaluacion** asocia un **Instrumento** con un **Curso** en una fecha determinada. Además, tiene un **estado** que indica en que parte del proceso se encuentra la evaluación. Hay 3 estados disponibles: Creada, Publicada y Finalizada, estos fueron fijados en base al flujo definido en la sección 3.1.3.

La clase **Grupo** posee un conjunto de alumnos (**Alumno**) y además se asocia a una

**Evaluación.** Cabe destacar que en el sistema de coevaluaciones[14] usado como referencia para este proyecto, los grupos estaban asociados a un curso, y estos se mantenían durante la permanencia del curso. Sin embargo, en este contexto lo que importa es saber en qué evaluación participó este grupo, y estos pueden variar su conformación (conjunto de alumnos) en distintas evaluaciones.

También se encuentra la clase **AsistenciaAlumno**, en esta se define si el **Alumno** se encuentra presente o no (campo **asistencia**) en el momento de la **Evaluación**.

La coevaluación se encuentra definida en la asociación de **Evaluación** con **Grupo** en la clase **EvaluacionGrupo**, que contiene la nota final que el grupo posee luego de que sus integrantes envíen sus respuestas. Además, esta considera un **estado** que tiene 3 posibles valores: **Creada**, para el momento en que se inicia una evaluación de grupo, **Parcial**, para evaluaciones de grupo en que al menos un alumno ha respondido la evaluación, pero aún falta al menos uno que responda, y finalmente **Finalizada**, para las cuales todos los alumnos del grupo ya enviaron sus respuestas. Cabe mencionar, que si la **Evaluación** es finalizada, entonces todas las evaluaciones de grupo son finalizadas con los resultados que tengan en ese momento, independiente de si todos los integrantes respondieron o no.

Como se ve en la figura 3.6, la clase **EvaluacionAlumno** tiene una **EvaluacionGrupo** y contiene la nota final que un **Alumno** le asignó a una **EvaluacionGrupo**, es decir, la nota que un alumno le puso a un grupo en una evaluación particular (coevaluación). Esta también posee 3 posibles estados: **Creada**, para las que han sido iniciadas, **Guardada**, para las que tienen respuestas guardadas pero aún falta al menos una pregunta por responder, y **Finalizada**, para las que se han enviado todas sus respuestas.

El **Codigoevaluacion** tiene una **EvaluacionAlumno** y un **codigo** de acceso que el alumno utiliza para acceder al formulario de la coevaluación en el sistema. Este también tiene un campo que define el **estado** en que se encuentra, el cual puede ser: **Activo** o **Desactivo**.

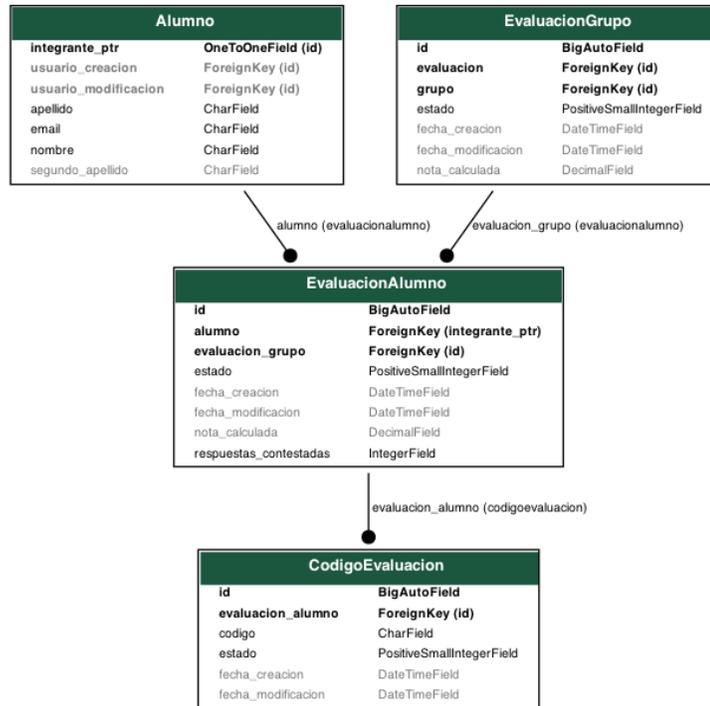


Figura 3.6: Diagrama del modelo de datos, sección relacionada con evaluaciones de alumnos.

La lógica de las respuestas se muestra en el diagrama de la figura 3.7. Para las respuestas existe la clase (`RespuestaEvaluacion`). De esta clase existen dos tipos: `RespuestaEvaluacionTexto`, donde la `respuesta` que se guarda es un texto. Esta fue creada para las respuestas de las preguntas de tipo texto (`PreguntaTexto`). Y `RespuestaEvaluacionOpcion`, donde la respuesta es una `Opcion`. Este tipo es para las preguntas de tipo opción.

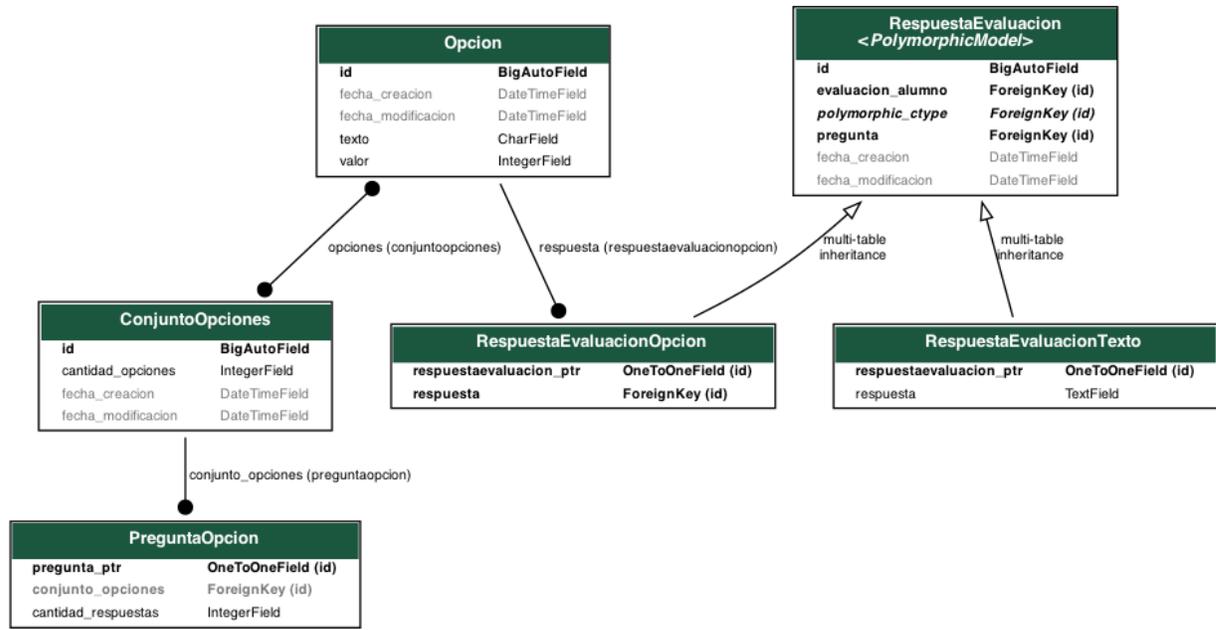


Figura 3.7: Diagrama del modelo de datos, sección relacionada con respuestas.

Finalmente, la clase `EmailParametro` define parámetros para los textos que contienen los emails que se envían a los alumnos. Esto sirve para mantener textos guardados para el asunto y mensaje de los correos a enviar, y de esta forma evitar que el administrador reescriba estos textos cada vez que quiere enviar un email.

Para la utilización de esta base de datos se deben tener en cuenta las siguientes consideraciones:

- Los usuarios de tipo administradores no forman parte de las subclases de `Integrante` debido a que estos tienen acceso a todos los modelos y no presentan relación con un curso o conjunto de cursos en particular.
- Existe una versión de `Instrumento` que debe ser precargada en la bases de datos, esta se basa en un documento entregado por los clientes, que define las preguntas (`Preguntas`) y las opciones (`Opcion`) de ellas. Sin embargo, se pueden crear nuevos instrumentos en el sistema.

### 3.2.2. Interfaz de usuario

Para diseñar la interfaz de usuario se realizaron mockups que permitieron visualizar ideas y definir la estructura de la navegación en el sistema. Los diseños fueron realizados previos a la implementación y obtuvieron cambios posteriormente, esto se verá reflejado en la sección siguiente. La herramienta utilizada para su desarrollo fue Justinmind. Cabe mencionar, que también se realizó un mockup de versión móvil equivalente a cada una de las vistas mencionadas. Esto debido a que el flujo de usuarios será mayor que el del sistema de base, y

los teléfonos móviles son los dispositivos más usados para acceder a la web en Chile, esto fue previamente mencionado en la sección 2.2.

### Vistas de respuesta de coevaluación para alumnos

En la figura 3.8 se muestra la vista en que los alumnos realizan el **ingreso de código** para acceder a la coevaluación. La versión equivalente en dispositivos móviles se encuentra en la figura 3.9 y está inspirada en los diseños de aplicaciones mencionadas en la sección 2.2.

Por otro lado, la figura 3.10 representa la vista de **formularios de evaluación** que deben responder los alumnos. En este caso, la versión móvil, en la figura 3.11, tiene otra distribución, en la que se muestra una pregunta del formulario a la vez. Esta interfaz tuvo de inspiración los diseños mencionados anteriormente, al analizar aplicaciones móviles en que los usuarios respondieran cuestionarios 2.2.

En la versión móvil del formulario (imagen a la izquierda de la figura 3.11) se encuentran en un comienzo los datos de la coevaluación que el usuario responderá, el alumno deberá apretar un botón que lo redirigirá a las preguntas.



Figura 3.8: Vista de ingreso de código de evaluación.

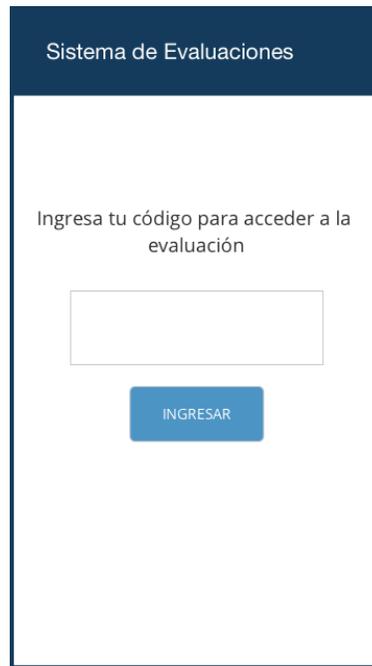


Figura 3.9: Vista de ingreso de código de evaluación (versión móvil).

The image shows a mobile application interface for an evaluation form. At the top, there is a dark blue header with the text 'Sistema de Evaluaciones' in white. Below the header, the main content area is white and contains the following information:  
**Ficha evaluación**  
**Información**  
Título: Evaluación 1  
Curso: MA3502 - Matemáticas 1  
Fecha de inicio: 00:00 4/3/2022  
Fecha de término: 00:00 5/4/2022  
Estado: Publicada  
**Información personal**  
Nombre: Marta Ramírez  
Email: mramirez@gmail.com  
**Responder Evaluación**  
1. Dedicamos tiempo a compartir los puntos de vista de cada una/o sobre el problema planteado.  
 Para nada     Un Poco     Parcialmente     En gran medida     Totalmente  
2. Dedicamos tiempo a identificar las habilidades de cada una/o para la solución del problema planteado.  
 Para nada     Un Poco     Parcialmente     En gran medida     Totalmente

Figura 3.10: Vista de formulario de evaluación.



Figura 3.11: Vista de formulario de evaluación (versión móvil).

### Vistas para profesor

Las primeras interfaces para profesores, fueron las diseñadas para la toma de **asistencia de alumnos** a evaluaciones. En las figuras 3.12 y 3.13 se muestran las versiones de escritorio y móvil respectivamente. En la parte superior de ellas, se encuentra la selección de evaluación a la que el profesor tomará asistencia. Además, se incluye la fecha de la misma. Posteriormente, se encuentran los estudiantes en una lista en la que el encargado debe seleccionar a los presentes.

Sistema de Evaluaciones msanchez | Cerrar sesión

### Asistencia

Evaluación:  Fecha:

Curso: MA3502 - Matemáticas 1

**A**

Alvarez Rodriguez, Javiera Almendra	✓
Alarcón Pérez, Julián Alfredo	✓
Aguilera Maldonado, Andrea Catalina	

**B**

Briceño Soto, Alberto Javier	✓
------------------------------	---

**C**

Cárrazco Gómez, Felipe Ignacio	✓
Castro Vásquez, Fernanda Camila	

Figura 3.12: Vista de toma de asistencia.

Sistema de Evaluaciones ☰

### Asistencia

Evaluación:

Curso: MA3502 - Matemáticas 1

Fecha:

**A**

Alvarez Rodriguez, Javiera Almendra	✓
Alarcón Pérez, Julián Alfredo	✓
Aguilera Maldonado, Andrea Catalina	

**B**

Briceño Soto, Alberto Javier	✓
------------------------------	---

Figura 3.13: Vista de toma de asistencia (versión móvil).

Los profesores pueden **gestionar una evaluación**, el diseño de la interfaz para esta funcionalidad considera 3 componentes (figura 3.14 y 3.15 para la versión móvil):

- La información de la evaluación, que especifica su nombre, curso, fechas de inicio y término y el estado en que se encuentra. Dependiendo de este estado es que se presentan distintos botones que representan las acciones que puede realizar el usuario.
- Los alumnos pendientes, donde se presenta la lista de alumnos que aún no responden la evaluación, además de un botón que permite enviar un email de recordatorio a estos estudiantes.
- La asistencia y grupos, la primera presenta la cantidad de alumnos presentes el día de la evaluación, y la sección de grupos muestra una lista especificando los integrantes de cada uno de ellos. Además, cada grupo tiene un botón de editar que permite cambiar la composición de él.

### Ficha evaluación

#### Información

Título  
Evaluación 1

Curso  
MA3502 - Matemáticas 1

Fecha de inicio  
00:00 4/3/2022

Fecha de término  
00:00 5/4/2022

Estado  
Publicada

[publicar evaluación](#)

#### Asistencia

Total de asistentes  
60 / 65

#### Grupos

**1. Grupo 1** 

- Javiera Medina
- Joaquín Sandoval
- Emilio Palma
- Agustina Henríquez
- Jorge Orellana

**2. Grupo 2** 

- Adolfo Roa
- Benjamin Godoy
- Francisco Vidal
- Isidora Mellado
- Adela Llao

**3. Grupo 3** 

- Roberto Alvarado
- Lucas Saez
- Valeria Santana
- Constanza Loyola
- Fernando Poblete

**4. Grupo 4** 

- Roberto Alvarado
- Lucas Saez
- Valeria Santana
- Constanza Loyola
- Fernando Poblete

**5. Grupo 5** 

- Alejandra Bustos
- Tomás Vera
- Gaspar Nuñez
- Ximena Alarcón
- Maximiliano Soto

**6. Grupo 6** 

- Alejandro Carreño
- Marcela Aguirre
- Juan Vásquez
- Emilia Soto

**Pendientes** 34 / 60

Buscar:

Isidora Mellado
Jorge Orellana
Francisca Vidal
Fernando Poblete
Emilio Palma
Tomás Vera

[Enviar correo de recordatorio](#)

[Anterior](#) [1](#) [2](#) [Siguiente](#)

Figura 3.14: Vista de ficha de una evaluación.

### 3.2.3. Diseño de componentes

En términos de la paleta de colores del sistema, se dejaron los colores predeterminados por Bootstrap, ya que estos son conocidos por los usuarios. Se utilizaron las clases de bootstrap para personalizar los tamaños de los componentes.



(a) Primera parte: información. (b) Segunda parte: alumnos pendientes.



(c) Tercera parte: asistencia y grupos.

Figura 3.15: Vista de estado de una evaluación (versión móvil).

### 3.3. Resumen

Recapitulando, en la primera sección de definición del problema se presentaron las historias de usuario y requisitos que delimitan el proyecto, además se presentó el flujo de evaluaciones que especifica el ciclo de vida de ellas.

En base a esta definición del problema, se muestra una solución propuesta, la que considera las arquitecturas física y lógica del sistema y su modelo de datos. También se exhibieron las interfaces que permiten definir como será la interacción del usuario con el sistema.

En los próximos capítulos se muestran los análisis y decisiones que se tomaron durante la implementación de la solución propuesta en este capítulo, y luego la validación del sistema en base a los objetivos específicos del trabajo.

# Capítulo 4

## Implementacion

Considerando el diseño de solución desarrollado en el capítulo 3, se llevo a cabo la implementación del sistema, la cual es detallada en este capítulo. Es aquí, que se profundizará sobre el back-end y el front-end de la plataforma web. También, se presentarán las interfaces resultantes y, finalmente, una descripción de las funcionalidades importantes.

### 4.1. Implementación de la aplicación web

La implementación de la aplicación web se puede separar en dos categorías, el **back-end** y el **front-end**. El back-end maneja la lógica de acceso y administración de los datos. Esta capa se preocupa de que la lógica del sitio funcione de manera correcta y que el desempeño de la plataforma sea el adecuado. Por otro lado, el front-end opera la parte visual y las interacciones con usuarios. En este capítulo se mostrarán desarrollos del proyecto en cada una de estas áreas.

#### 4.1.1. Back-end

En un comienzo se trabajó extendiendo el sistema de coevaluaciones del DCC, creando una nueva aplicación que reutilizaba algunas interfaces y partes del modelo. No obstante, a medida que se avanzaba en el desarrollo se presentaban cada vez más diferencias, por lo que se decidió crear una nueva aplicación separada a la anterior.

Se creó una aplicación en Django, la cual fue llamada **Evaluaciones**. Es en ella que fue implementada la gestión de instrumentos, cursos, grupos y evaluaciones, considerando las vistas para crear, editar y eliminar cada uno de ellos.

Cuando una evaluación es creada, se debe tomar la asistencia de alumnos presentes en ella. En base a esta asistencia se generan grupos de forma aleatoria. La funcionalidad en torno a esto se encuentra en el archivo `asistencia_y_grupos.py` del back-end. En él se encuentra la función `generar_grupos_aleatorios_evaluacion` que genera los grupos utilizando la librería

*random* de python, la cual sirve para generar números aleatorios y contiene la función `shuffle` que toma una secuencia y la reordena, la cual es utilizada en la generación de grupos.

Cuando una evaluación es publicada, se deben crear códigos de acceso para los estudiantes participantes de ella. El archivo `codigo_evaluacion.py` se encarga de generar códigos aleatorios utilizando la librería `random` de python.

El back-end también se hace cargo de realizar validaciones cuando los usuarios ingresan archivos al sistema para importar alumnos. Para esto se utiliza la librería `csv` nativa de python.

Para el modulo de notificaciones se creó el archivo `email_handler.py` que se encarga del envío de emails a alumnos, reemplazando los datos de cada uno de ellos en el contenido del correo a enviar. Este archivo utiliza el módulo de envío de emails `django.core.email`, propio de Django [20].

En las siguientes secciones se comentará la conexión al sistema de gestión de bases de datos PostgreSQL, la organización de los archivos y la gestión de usuarios en el sistema.

## Conexión a PostgreSQL

Se utilizó el adaptador de bases de datos `psycopg2` para conectar Django a PostgreSQL, esto se realizó en las configuraciones de Django de la forma que se ve en el código 4.1.

```
1 DATABASES = {
2     'default': {
3         'ENGINE': 'django.db.backends.postgresql_psycopg2',
4         'NAME': 'sistema_evaluaciones_db',
5         'USER': <usuario_postgres>,
6         'PASSWORD': <contraseña_usuario_postgres>,
7         'HOST': localhost<url_host>,
8         'PORT': '5432',
9     }
10 }
```

Código 4.1: Conexión a PostgreSQL

## Archivos importantes

La aplicación de evaluaciones cuenta de 4 archivos fundamentales para su funcionamiento. El primero es el de `models.py` donde se define el modelo de datos de la aplicación de Evaluaciones. El segundo es `views.py` donde se define la lógica necesaria para devolver una respuesta a un usuario, estas vistas reciben solicitudes web y devuelven una respuesta web. La respuesta retornada puede ser de distintos tipos, particularmente en este proyecto se retorna el contenido HTML de una página web, una redirección o un error.

Debido a que la aplicación necesita recibir inputs de los usuarios, se vuelve necesario crear

formularios que reciban y manejen esta información. Estos pueden ser definidos directamente en los archivos HTML, sin embargo Django provee una serie de herramientas que ayudan a preparar datos para renderizar, crear los formularios HTML, y recibir la información ingresada por los usuarios. Es por esto, que los formularios se definen como clases en el sistema, estos se encuentran en el archivo `forms.py`.

Finalmente, para mapear las direcciones URL a sus respectivas vistas se tiene el archivo `urls.py`. La estructura de URL de un sitio debe ser lo más simple posible, estas deben construirse de forma lógica y ser comprensibles para seres humanos. Para lograr esto en el esquema se utilizaron las recomendaciones de la central de la búsqueda de Google [16].

## Gestión de usuarios

Los usuarios del sistema pueden ser separados en dos categorías. La primera considera a los usuarios que gestionan el proceso de evaluaciones y poseen una cuenta en el sistema, y la segunda de usuarios que no poseen cuenta y responden las coevaluaciones. Para los usuarios con cuenta se utilizó el sistema de autenticación de Django.

Para los usuarios que se autentican en la aplicación se creó la clase `Usuario`, definida en 4.2. Esta es una subclase de `AbstractUser`, clase de Django que se utiliza para extender su usuario predeterminado `User`. La razón de por qué no se utilizó directamente la opción predeterminada, es debido al campo `rol` que se agregó para diferenciar los tipos de usuarios entre profesores y administradores. Esta distinción es necesaria para manejar los permisos y acceso a información que cada grupo tiene.

Además del rol, se agregó un campo opcional que guarda el segundo nombre de los usuarios. Cabe mencionar que esta extensión también permite añadir otros campos de ser necesarios en un futuro.

```
11 class Usuario(AbstractUser):
12     PROFESOR = 1
13     ADMIN = 2
14     ROL = (
15         (PROFESOR, _('Profesor')),
16         (ADMIN, _('Administrador')),
17     )
18     rol = models.PositiveSmallIntegerField(
19         choices=ROL,
20         default=PROFESOR,
21     )
22     second_last_name = models.CharField(max_length=150, blank=
True, null=True)
```

Código 4.2: Clase Usuario

Por otro lado, para gestionar los usuarios que participan de diversos cursos, se implementó la clase `Integrante`, que según muestra el código 4.3 considera una relación *Many-to-Many* de integrantes con cursos (`Curso`). Esta relación fue considerada debido a que los alumnos

y profesores pueden tener más de un curso asociado.

`PolymorphicModel` es la clase de de la librería `django-polymorphic` que facilita el uso de modelos heredados. La clase `Integrante` se extiende de ella, debido a que esta tiene dos subclases que representan a los profesores (`Profesor`) y alumnos (`Alumno`) de un curso. La separación de ambas es necesaria debido a que representan a distintos entes en la lógica del sistema y la información que se debe mantener de cada una de ellas difiere.

Por un lado, los profesores tienen asociado un `Usuario` ya que pueden ingresar al sistema y gestionar evaluaciones. Es ahí donde se encuentra su información como correo electrónico y nombres. En cambio, los alumnos deben mantener en su clase esta información. Un campo fundamental de esta clase es el de `email`, el que permite diferenciar a los alumnos ingresados. Para asegurar esta diferenciación se utiliza la restricción de unicidad (*unique constraint*).

```
23 class Integrante(PolymorphicModel):
24     cursos = models.ManyToManyField(Curso)
25
26 class Profesor(Integrante):
27     usuario = models.OneToOneField(Usuario, on_delete=models.
    CASCADE)
28
29 class Alumno(Integrante):
30     email = models.CharField(max_length=100, unique=True)
31     nombre = models.CharField(max_length=255)
32     apellido = models.CharField(max_length=255)
33     segundo_apellido = models.CharField(max_length=255, blank=
    True, null=True)
```

Código 4.3: Clase Usuario

### 4.1.2. Front-end

Para el front-end se utilizó la funcionalidad de *templates* de Django, con los cuales se pueden generar documentos HTML de forma dinámica.

Se crearon archivos HTML dentro del directorio `templates`, los cuales luego son utilizados por las vistas del sistema, que se encarga de entregarlos los valores necesarios para desplegar la información solicitada por el usuario.

Además, se hizo uso de bootstrap para crear interfaces limpias y con un diseño responsivo. Esto para facilitar la navegación de los usuarios que utilizarán dispositivos móviles para interactuar con la aplicación.

También, se utilizó la herramienta datatables para el diseño de las tablas de la aplicación. Esta permite que todas tengan un diseño común y las mismas funcionalidades, tales como búsqueda de datos y paginamiento.

Para facilitar la integración entre Django y bootstrap se utilizó la librería `django-`

`bootstrap5`, la cual contiene etiquetas como la de `bootstrap_form`, que se utilizó en el proyecto para generar formularios bootstrap en las interfaces.

## 4.2. Interfaces

En esta sección se presentarán vistas importantes del sistema. Para los profesores se mostrará la vista de gestión de evaluaciones y la de alumnos pendientes por responder la coevaluación. Por otro lado, para alumnos se mostrará su interfaz de formulario de coevaluación a un grupo, la cual es parte fundamental del proceso. Cabe destacar que todos los datos que poseen estas interfaces fueron inventados para la validación del sistema.

### 4.2.1. Gestión de evaluaciones

La interfaz diseñada para gestionar evaluaciones considera tres partes importantes, las cuales pueden verse en la figura 4.1. En la primera parte se encuentra su información, como nombre, curso, fechas de inicio y término y estado en que se encuentra. En la segunda sección se encuentra una lista de alumnos pendientes que aún no responden la coevaluación. Finalmente, en la tercera parte se encuentra la asistencia y los grupos de la evaluación en cuestión.

Al acceder a esta interfaz el sistema verifica que el usuario se encuentre logueado. Esta verificación sucede en todas las interfaces para profesores y administradores. En la parte de información de esta vista se encuentran los botones para editar y eliminar la evaluación. Luego de esto, se presentan los datos importantes, tales como el instrumento y curso asociados, fecha de inicio y término y estado actual de esta.

Dependiendo del estado, los cuales fueron definidos en la sección 3.2.1, el sistema ofrecerá las acciones adecuadas para dicho estado, a través de botones que se encuentran en la parte inferior de la sección. Como ejemplo, se puede ver en la imagen 4.1 una evaluación en estado publicada, por lo tanto los botones consideran las acciones de finalización de evaluación o visualización de resultados parciales.

Por otro lado, si la evaluación se encuentra en estado creada, como se muestra en la figura 4.2, la acción disponible es la de publicar la evaluación, y si está en estado finalizada (figura 4.3) se pueden ver los resultados finales.

## Ficha de Evaluación

### Información

Nombre  
Evaluación 1 de Prueba

Instrumento  
Escala de evaluación de actividades centradas en la resolución colaborativa de problemas

Curso  
Matemáticas MA01-1 2022

Fecha de inicio  
14 de Junio de 2022 a las 12:00

Fecha límite  
30 de Junio de 2022 a las 23:59

Estado  
Publicada

[finalizar evaluación](#) [ver resultados parciales](#)

### Asistencia

Total de asistentes  
21 / 25

### Grupos (total 5)

<p>1. G1 </p> <ol style="list-style-type: none"> <li>1. Joaquín Sandoval</li> <li>2. Emilio Palma</li> <li>3. Agustina Henríquez</li> <li>4. Jorge Orellana</li> <li>5. Rodrigo Espinoza</li> <li>6. Javiera Medina</li> </ol>	<p>2. G2 </p> <ol style="list-style-type: none"> <li>1. Alejandra Bustos</li> <li>2. Tomás Vera</li> <li>3. Gaspar Nuñez</li> <li>4. Ximena Alarcón</li> <li>5. Maximiliano Soto</li> </ol>
<p>3. G3 </p> <ol style="list-style-type: none"> <li>1. Roberto Alvarado</li> <li>2. Lucas Saez</li> <li>3. Valeria Santana</li> <li>4. Constanza Loyola</li> <li>5. Fernando Poblete</li> </ol>	<p>4. G4 </p> <ol style="list-style-type: none"> <li>1. Adolfo Roa</li> <li>2. Benjamín Godoy</li> <li>3. Francisco Vidal</li> <li>4. Isidora Mellado</li> <li>5. Adela Llao</li> </ol>
<p>5. Grupo extra </p> <ol style="list-style-type: none"> <li>1. Benjamín Godoy</li> <li>2. Ximena Alarcón</li> <li>3. Maximiliano Soto</li> <li>4. Agustina Henríquez</li> </ol>	

### Pendientes

Mostrar  registros Buscar:

- Adela Llao Llao
- Adolfo Roa Martínez
- Agustina Henríquez Salazar
- Alejandra Bustos Quezada
- Benjamín Godoy Lagos
- Constanza Loyola Carrasco
- Emilio Palma Ruiz
- Fernando Poblete Carvajal
- Francisco Vidal Moreno
- Gaspar Nuñez Cortés

Mostrando 1 a 10 de 21 registros

[Anterior](#)
**1**
2
3
[Siguiente](#)

[enviar recordatorio por correo](#)

Figura 4.1: Interfaz de gestión de evaluaciones.

## Ficha de Evaluación

---

**Información**  

Nombre  
Evaluación 2 de prueba

Instrumento  
Escala de evaluación de actividades centradas en la resolución colaborativa de problemas

Curso  
Matemáticas MA01-1 2022

Fecha de inicio  
7 de Julio de 2022 a las 20:06

Fecha límite  
21 de Julio de 2022 a las 20:06

Estado

**Creada**

[publicar evaluación a alumnos](#)

Figura 4.2: Vista de gestión de evaluación, sección información. Estado evaluación: creada.

## Ficha de Evaluación

---

**Información**  

Nombre  
Evaluación 2 de prueba

Instrumento  
Escala de evaluación de actividades centradas en la resolución colaborativa de problemas

Curso  
Matemáticas MA01-1 2022

Fecha de inicio  
7 de Julio de 2022 a las 20:06

Fecha límite  
21 de Julio de 2022 a las 20:06

Estado

**Finalizada**

[ver resultados](#)

Figura 4.3: Vista de gestión de evaluación, sección información. Estado evaluación: finalizada.

En la segunda sección de la interfaz, se encuentran los alumnos pendientes, es decir, estudiantes que aún no completan el formulario de coevaluación. Aquí, se puede buscar por el nombre del alumno, y se puede enviar un recordatorio por correo. El botón de envío de recordatorio redirige a otra vista que se detallará próximamente.

Finalmente, la tercera sección considera la asistencia de la evaluación. Esta debe ser tomada luego de que la evaluación sea creada, este formulario puede verse en 4.4. El formulario contiene los nombres de los alumnos del curso asociado a la evaluación. Cada uno de estos nombres, está acompañado de un elemento input de tipo checkbox, el cual el profesor deberá seleccionar para los alumnos presentes.

Luego de enviar el formulario el total de asistentes será mostrado en la sección de Asistencia de la interfaz de ficha de evaluación mencionada y presente en la figura 4.1. No obstante, luego de que este dato se encuentre disponible, la asistencia puede seguir siendo editada utilizando el botón de edición que se encuentra junto al título Asistencia.

## Asistencia

### Evaluación

Nombre

Evaluación de prueba 3

Curso

Matemáticas MA01-1 2022

Fecha de inicio

22 de Julio de 2022 a las 12:00

Fecha límite

30 de Julio de 2022 a las 12:00

### Alumnos

Martín Álvarez :	<input type="checkbox"/>
Adolfo Roa Martínez:	<input checked="" type="checkbox"/>
Roberto Alvarado Muñoz:	<input checked="" type="checkbox"/>
Alejandra Bustos Quezada:	<input checked="" type="checkbox"/>
Benjamín Godoy Lagos:	<input type="checkbox"/>
Lucas Saez Guzmán:	<input type="checkbox"/>
Tomás Vera Ortiz:	<input checked="" type="checkbox"/>
Andrea Rivera Valdivia:	<input checked="" type="checkbox"/>
Gaspar Nuñez Cortés:	<input checked="" type="checkbox"/>
Ximena Alarcón Flores:	<input checked="" type="checkbox"/>
Maximiliano Soto Pérez:	<input type="checkbox"/>
Sofía Hernández Silva:	<input checked="" type="checkbox"/>
Valeria Santana Orellana:	<input checked="" type="checkbox"/>
Joaquín Sandoval Molina:	<input checked="" type="checkbox"/>
Francisco Vidal Moreno:	<input checked="" type="checkbox"/>
Denisse Navarrete Ortega:	<input checked="" type="checkbox"/>
Isidora Mellado Acuña:	<input checked="" type="checkbox"/>
Emilio Palma Ruiz:	<input checked="" type="checkbox"/>
Agustina Henríquez Salazar:	<input checked="" type="checkbox"/>
Jorge Orellana Bravo:	<input type="checkbox"/>
Constanza Loyola Carrasco:	<input checked="" type="checkbox"/>
Rodrigo Espinoza Herrera:	<input checked="" type="checkbox"/>
Adela Liao Pino:	<input checked="" type="checkbox"/>
Fernando Poblete Carvajal:	<input type="checkbox"/>
Javiera Medina Guerrero:	<input checked="" type="checkbox"/>

[Guardar asistencia](#)

Figura 4.4: Vista de asistencia.

Después de que la asistencia es tomada, se habilita la opción de generar grupos aleatorios en base a ella, para esto existe el botón que se muestra en la figura 4.5, el cual redirige a la interfaz que luce como la de la imagen 4.6. Esta interfaz contiene información de la evaluación en la cual se generarán estos grupos, y del total de asistentes, este dato se muestra para que el profesor tenga conocimiento antes de elegir la cantidad de grupos que se harán. Luego de que el profesor entregue este dato y presione el botón para generar grupos, será redirigido a la interfaz de ficha de una evaluación (figura 4.1), donde podrá ver los grupos resultantes.



Figura 4.5: Vista de gestión de evaluación, sección grupos.

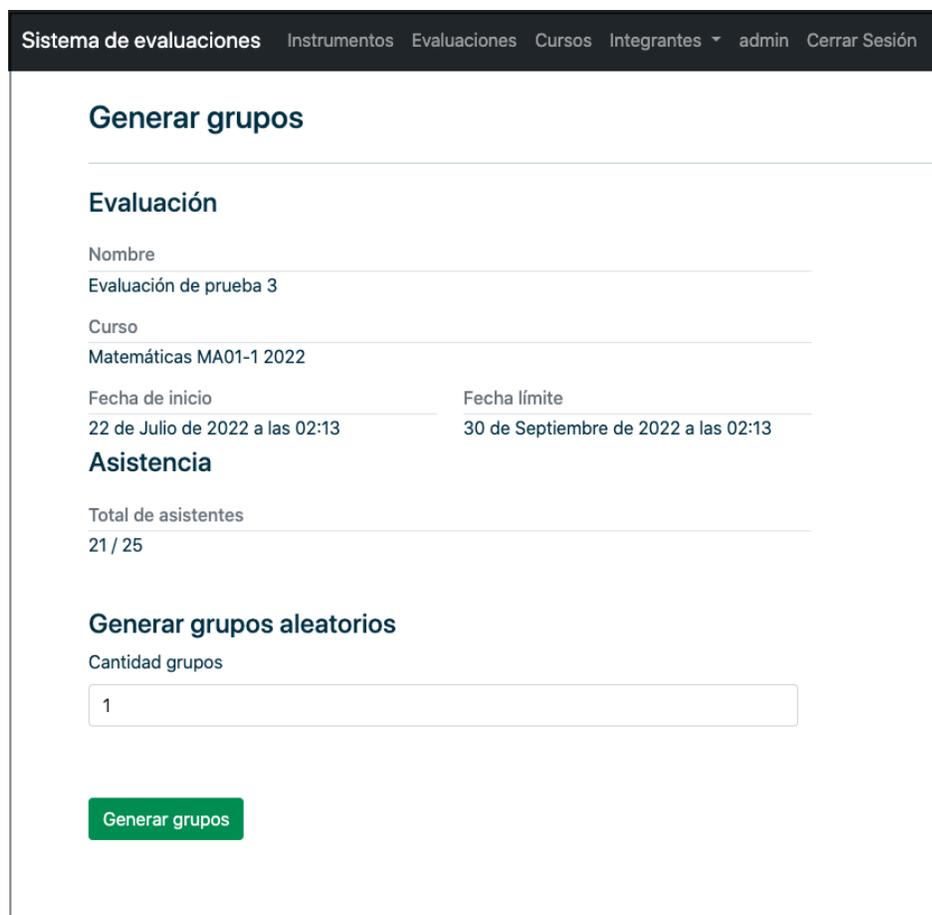


Figura 4.6: Vista de generación de grupos. Asistencia previamente ingresada.

Si se intenta generar estos grupos sin tener la asistencia entonces se muestra el mensaje presente en la figura 4.7, y la opción de tomar asistencia, presentada en el botón de la parte posterior, el cual redirige al usuario a la interfaz de toma de asistencia de la figura 4.4.

## Generar grupos

---

### Evaluación

Nombre  
Evaluación de prueba 3

Curso  
Matemáticas MA01-1 2022

Fecha de inicio  
22 de Julio de 2022 a las 02:13

Fecha límite  
30 de Septiembre de 2022 a las 02:13

**Asistencia**  
La asistencia no ha sido tomada aún. Debe tomar la asistencia para luego generar grupos con los alumnos presentes.

[tomar asistencia](#)

Figura 4.7: Vista de generación de grupos. Asistencia no ingresada

#### 4.2.2. Formulario de coevaluación de alumnos

Esta vista considera una interfaz para escritorio y otra para dispositivos móviles. En la primera, como se puede ver en la figura 4.8, se encuentra una sección en el comienzo que informa el grupo al que se está evaluando, y luego se presenta el formulario con las preguntas ordenadas verticalmente y las opciones de respuestas ordenadas de forma horizontal.

## Responder evaluación

Estás respondiendo para:

Grupo 6 - G6

1. Dedicamos tiempo a identificar las habilidades de cada una/o para la solución del problema planteado.

Para nada	Un poco	Parcialmente	En gran medida	Totalmente
-----------	---------	--------------	----------------	------------

2. Dedicamos tiempo a compartir los puntos de vista de cada una/o sobre el problema planteado.

Para nada	Un poco	Parcialmente	En gran medida	Totalmente
-----------	---------	--------------	----------------	------------

3. Encontramos una forma de trabajo en la que todos contribuimos a la solución del problema planteado.

Para nada	Un poco	Parcialmente	En gran medida	Totalmente
-----------	---------	--------------	----------------	------------

4. Exploramos distintas posibilidades para organizar el trabajo.

Para nada	Un poco	Parcialmente	En gran medida	Totalmente
-----------	---------	--------------	----------------	------------

5. La distribución de tareas la realizamos mediante la discusión grupal.

Para nada	Un poco	Parcialmente	En gran medida	Totalmente
-----------	---------	--------------	----------------	------------

Figura 4.8: Interfaz de formulario de coevaluación de alumnos.

Del mismo modo, la versión móvil muestra en un principio la información del grupo que se está evaluando y posteriormente las preguntas de la coevaluación que el alumno debe contestar, lo que es visible en la figura 4.9. No obstante, en este caso el formulario se presenta por etapas, de tal forma que cada etapa exhibe una pregunta a la vez. Para volver a la pregunta previa y pasar a la siguiente existen los botones de *anterior* y *siguiente* respectivamente. Como se vio en la sección 3.2.2, este diseño está inspirado en las aplicaciones revisadas durante el trabajo que manejan formularios para dispositivos móviles.

The image shows a mobile application interface for an evaluation system. At the top, there is a dark header with the text "Sistema de evaluaciones" and a hamburger menu icon. Below the header, the main content area is white and contains the following elements: a title "Responder evaluación", a subtitle "Estás respondiendo para:", the group name "Grupo 6 - G6", and a paragraph of text: "Dedicamos tiempo a compartir los puntos de vista de cada una/o sobre el problema planteado." Below this text is a vertical stack of five buttons representing rating options: "Para nada", "Un poco", "Parcialmente", "En gran medida", and "Totalmente". The "En gran medida" button is highlighted with a blue background. At the bottom of the form, there are two blue buttons labeled "Anterior" and "Siguiente".

Figura 4.9: Interfaz de formulario de coevaluación de alumnos (versión móvil).

### 4.2.3. Alumnos pendientes

Dentro de las interfaces importantes también se encuentra la de alumnos pendientes. En esta, como se ve en la figura 4.10, se presenta la lista de los alumnos que aún no responden la coevaluación. Asimismo, se presenta un apartado de envío de recordatorio donde se puede enviar un correo electrónico a estos alumnos. Además, previo al envío se puede editar el asunto y mensaje del correo. Los valores predeterminados de estos campos se rescatan desde la base de datos, estos se encuentran en el modelo de parámetros de email (**EmailParametro**).

### Alumnos pendientes y recordatorios

#### Pendientes

Mostrar 10 registros      Buscar:

- Adela Llao Pino
- Adolfo Roa Martínez
- Agustina Henríquez Salazar
- Alejandra Bustos Quezada
- Benjamín Godoy Lagos
- Constanza Loyola Carrasco
- Emilio Palma Ruiz
- Fernando Poblete Carvajal
- Francisco Vidal Moreno
- Gaspar Nuñez Cortés

Mostrando 1 a 10 de 21 registros      [Anterior](#) [1](#) [2](#) [3](#) [Siguiente](#)

#### Enviar recordatorio

Asunto

Mensaje  

```
<p>Estimado/a {alumno_nombre} {alumno_apellido}, <br> </p>
<p>Se le recuerda contestar la coevaluación <i>"{evaluacion_nombre}"</i> del
curso <i>{curso_nombre}</i> para el grupo <i>{grupo_nombre}</i>.</p>
<br>
<p>Debe dirigirse a {url_ingresar_codigo} para ingresar su código.</p>
<br>
<p>Su código de acceso: <b>{codigo}</b>.</p>
```

Figura 4.10: Interfaz de alumnos pendientes.

#### 4.2.4. Resultados

La interfaz de resultados es donde se muestran las respuestas que los alumnos han ingresado y las notas calculadas que tiene cada grupo. Esta vista, que se muestra en la imagen 4.11, contiene dos tablas. La primera, presente en la figura 4.12 contiene los resultados por grupo y la segunda, en la figura 4.13, los resultados por alumno.

## Resultados de coevaluación

### Información de evaluación

Nombre

Evaluación trabajo colaborativo

Curso

Matemáticas MA01-1 2022

Resultados por grupo

Resultados por alumno

Copiar

CSV

Excel

PDF

Imprimir

Buscar:

email de alumno	nombre de alumno	ID grupo	nombre grupo	Dedicamos tiempo a identificar las habilidades de cada una/o para la solución del problema planteado.	Dedicamos tiempo a compartir los puntos de vista de cada una/o sobre el problema planteado.	Encontramos una forma de trabajo en la que todos contribuimos a la solución del problema planteado.	Exploramos distintas posibilidades para organizar el trabajo.	La distribución de tareas la realizamos mediante la discusión grupal.	Cada uno comprendió las tareas que le correspondían.	Logramos acordar una definición compartida del problema.
adolfo@gmail.com	Adolfo Roa Martínez	11	G11	100	100	75	100	50	50	100
arivera_78@gmail.com	Andrea	10	G10	75	50	100	100	100	75	100
bgodoy@gmail.com	Benjamín Godoy Lagos	12	G12	50	75	75	100	100	100	75
cloyola@gmail.com	Constanza Loyola Carrasco	10	G10	100	100	75	100	100	100	100
dnavarrete@gmail.com	Denisse Navarrete Ortega	11	G11	75	50	100	100	100	75	100
epalma2@gmail.com	Emilio Palma Ruiz	13	G13	75	75	100	50	100	75	100
fpoblete@gmail.com	Fernando Poblete Carvajal	12	G12	75	50	100	100	100	75	100
fvidal34@gmail.com	Francisco Vidal Moreno	12	G12	75	50	100	100	100	75	100
gnunez@example.com	Gaspar Nuñez Cortés	10	G10	50	100	100	75	100	50	100
imellado@gmail.com	Isidora Mellado Acuña	11	G11	75	50	100	100	75	100	100

Mostrando 1 a 10 de 21 registros

Anterior **1** 2 3 Siguiente

Figura 4.11: Interfaz de resultados de evaluación.

Resultados por grupo    Resultados por alumno

Copiar   CSV   Excel   PDF   Imprimir    Buscar:

ID Grupo	Nombre de grupo	Dedicamos tiempo a identificar las habilidades de cada una/o para la solución del problema planteado.	Dedicamos tiempo a compartir los puntos de vista de cada una/o sobre el problema planteado.	Encontramos una forma de trabajo en la que todos contribuimos a la solución del problema planteado.	Exploramos distintas posibilidades para organizar el trabajo.	La distribución de tareas la realizamos mediante la discusión grupal.	Cada uno comprendió las tareas que le correspondían.	Logramos acordar una definición compartida del problema.	Discutimos diferentes posibilidades antes de encontrar una definición compartida del problema.	Logramos identificar las tareas necesarias para resolver el problema.
10	G10	81.25	87.5	87.5	68.75	93.75	68.75	87.5	75.0	75.0
11	G11	75.0	75.0	93.75	93.75	81.25	68.75	100.0	68.75	68.75
12	G12	62.5	62.5	87.5	87.5	100.0	87.5	93.75	81.25	81.25
13	G13	56.25	56.25	100.0	87.5	81.25	93.75	81.25	87.5	87.5

Mostrando 1 a 4 de 4 registros

Anterior   **1**   Siguiente

Figura 4.12: Tabla de resultados por grupo. Interfaz de resultados de evaluación.

Resultados por grupo		Resultados por alumno										
Copiar		CSV	Excel	PDF	Imprimir						Buscar: <input type="text"/>	
email de alumno	nombre de alumno	ID grupo	nombre grupo	Dedicamos tiempo a identificar las habilidades de cada uno/a para la solución del problema planteado.	Dedicamos tiempo a compartir los puntos de vista de cada uno/a sobre el problema planteado.	Encontramos una forma de trabajo en la que todos contribuimos a la solución del problema planteado.	Exploramos distintas posibilidades para organizar el trabajo.	La distribución de tareas la realizamos mediante la discusión grupal.	Cada uno comprendió las tareas que le correspondían.	Logramos acordar una definición compartida del problema.		
adolfo@gmail.com	Adolfo Roa Martínez	11	G11	100	100	75	100	50	50	100		
arivera_78@gmail.com	Andrea Rivera Valdivia	10	G10	75	50	100	100	100	75	100		
bgodoy@gmail.com	Benjamín Godoy Lagos	12	G12	50	75	75	100	100	100	75		
cloyola@gmail.com	Constanza Loyola Carrasco	10	G10	100	100	75	100	100	100	100		
dnavarrete@gmail.com	Denisse Navarrete Ortega	11	G11	75	50	100	100	100	75	100		
epalma2@gmail.com	Emilio Palma Ruiz	13	G13	75	75	100	50	100	75	100		
fpoblete@gmail.com	Fernando Poblete Carvajal	12	G12	75	50	100	100	100	75	100		
fvidal34@gmail.com	Francisco Vidal Moreno	12	G12	75	50	100	100	100	75	100		
gnunez@example.com	Gaspar Nuñez Cortés	10	G10	50	100	100	75	100	50	100		
imellado@gmail.com	Isidora Mellado Acuña	11	G11	75	50	100	100	75	100	100		

Mostrando 1 a 10 de 21 registros

Anterior **1** 2 3 Siguiente

Figura 4.13: Tabla de resultados por alumno. Interfaz de resultados de evaluación.

### 4.3. Funcionalidades importantes

A continuación se detallarán funcionalidades claves que forman parte de este sistema. Dentro de las que se encuentran el envío de notificaciones a alumnos, la importación de alumnos a través de archivos y la exportación de resultados.

#### 4.3.1. Envío de notificaciones a alumnos pendientes

El sistema también permite enviar correos electrónicos de recordatorio a los alumnos pendientes, utilizando la vista presentada en 4.10. La interfaz muestra valores predeterminados del asunto y cuerpo del correo, los que rescata de la base de datos. Estos pueden ser edi-

tados por el usuario. Además, estos campos contienen variables entre los paréntesis {}, las cuales son reemplazadas por la información correspondiente a cada usuario al que se envía la notificación.

Al enviar el recordatorio a alumnos pendientes, por cada alumno se llama a la función de envío de email, la cual primero reemplaza las variables con la información de la evaluación y alumno correspondiente. Un ejemplo de email de recordatorio enviado, con estas variables reemplazadas, puede verse en 10.

### 4.3.2. Importación de alumnos

Existen dos formas de agregar alumnos a un curso, la primera es agregar cada alumno de forma individual y la segunda, que se detalla en esta sección, que consiste en importar alumnos desde un archivo con extensión *csv*, *xls* o *xlsx*.

Para realizar esta importación el usuario debe completar el formulario presente en 4, donde debe ingresar un archivo. Esto manda una petición post a la aplicación. Los encabezados deben contener las columnas *nombre*, *primer apellido*, *segundo apellido* y *email*.

Como ejemplo se tiene el archivo cuyos datos son detallados en la tabla 4.1. Luego de que el usuario suba este archivo, este se maneja utilizando la librería *pandas* y la librería *csv* de python. Antes de guardar la información, el archivo pasa por una función de validación. La función verifica que el archivo contenga las columnas necesarias, además de realizar una validación por fila. En cada fila se comprueba que exista el nombre, primer apellido y email del alumno, y que el email tenga un formato correcto. De no cumplir con alguno de estos requisitos el archivo es rechazado mostrando un mensaje de error que detalle el problema ocasionado.

Nombre	Primer apellido	Segundo apellido	Email
Adela	Llao	Pino	allao@gmail.com
Adolfo	Roa	Martínez	adolfo@gmail.com
Roberto	Alvarado	Muñoz	ralvarado@hotmail.com
Javiera	Medina	Guerrero	jmedina@outlook.com
Fernando	Poblete	Carvajal	fpoblete@gmail.com
Alejandra	Bustos	Quezada	abustos46@outlook.es
Benjamín	Godoy	Lagos	blagos@gmail.com

Tabla 4.1: Ejemplo de archivo de alumnos.

Luego de las validaciones, la función guarda la información del archivo en una lista de diccionarios, presente en el código 4.4. En ella, cada elemento representa una fila del archivo en forma de diccionario, donde las llaves especifican los nombres de la columnas, y los valores la información de las celdas respectivas.

```

34 [{ 'nombre': 'Adela', 'primer apellido': 'Llao', 'segundo apellido
    ': 'Pino', 'email': 'allao@gmail.com' },
35 { 'nombre': 'Adolfo', 'primer apellido': 'Roa', 'segundo apellido'
    ': 'Martínez', 'email': 'adolfo@gmail.com' },
36 { 'nombre': 'Roberto', 'primer apellido': 'Alvarado', 'segundo
    apellido': 'Muñoz', 'email': 'ralvarado@hotmail.com' },
37 ...
38 { 'nombre': 'Benjamín', 'primer apellido': 'Godoy', 'segundo
    apellido': 'Lagos', 'email': 'blagos@gmail.com' } ]

```

Código 4.4: Lista resultante de función de importación de alumnos.

### 4.3.3. Exportación de resultados

La interfaz de resultados, presentada en la sección 4.2.4, contiene dos tablas con la información de las respuestas. Ambas presentan la posibilidad de descarga de los datos, como se puede ver en la figura 4.14, entre las opciones se encuentran los formato csv y excel, lo cual fue solicitado por los clientes, y otras posibilidades como la de copiar los datos. Para esto se utilizó la librería *datatables*.



Figura 4.14: Botones de exportación de tabla de resultados.

## 4.4. Resumen

En este capítulo se pudo ver como se realizó la implementación de la solución, partiendo por señalar como se realizó el back-end y front-end del proyecto. Luego, se mostraron las interfaces fundamentales de la aplicación, las cuales consideraban las vistas de gestión de evaluaciones, de formulario de coevaluación para alumnos, la vista de alumnos pendientes y la de resultados.

Asimismo, se presentaron las funcionalidades importantes del sistema. Primero, se mostró la funcionalidad de envío de notificaciones por correo a los alumnos pendientes. Luego, el funcionamiento de la importación de alumnos a través de archivos y la exportación de resultados finales.

Después de descrita esta implementación, es necesaria la validación del sistema. En el siguiente capítulo se da a conocer la metodología y resultados de esta validación.

# Capítulo 5

## Validación

En este capítulo se pretende demostrar que el trabajo realizado presenta una solución efectiva al problema planteado. Es por esto, que en la sección 5.1 se compararán los resultados obtenidos en las vistas con los mockups e historias de usuario definidos en el capítulo 3.

### 5.1. Validación de interfaz web

#### 5.1.1. Vista de ingreso de código

La interfaz de ingreso de código presente en la figura 5.1 se corresponde con el mockup 3.8 y tiene como funcionalidad proporcionar una forma de ingreso a los alumnos, sin necesidad de que ellos posean cuenta. Esto responde a la historia de usuario HU04.

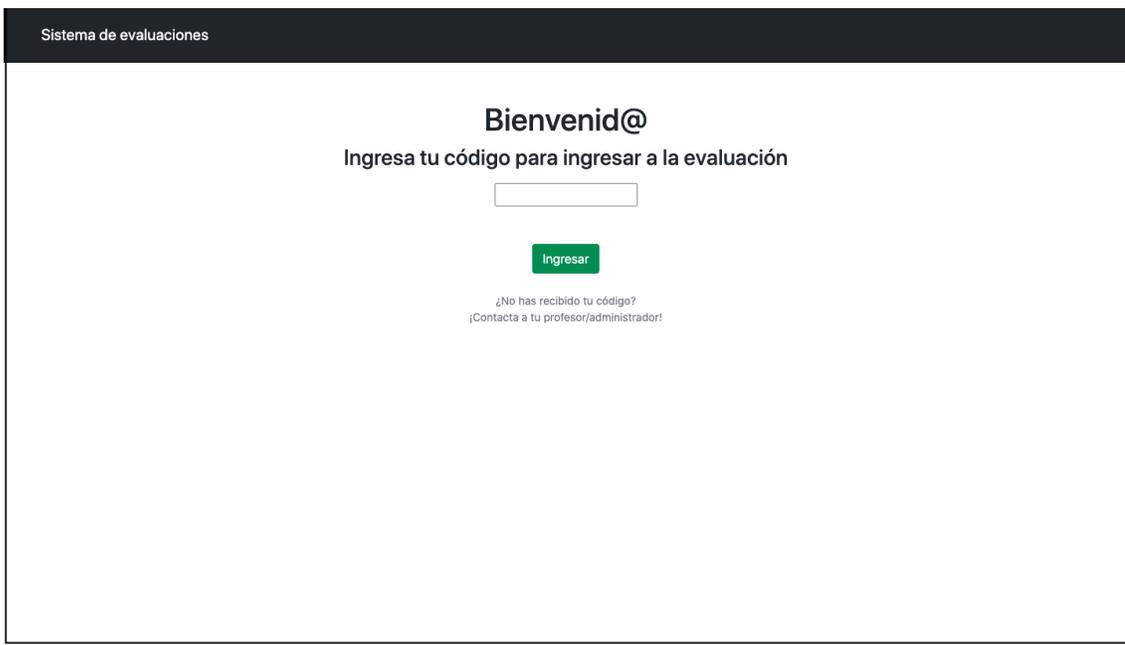


Figura 5.1: Vista de ingreso de código.

### 5.1.2. Vista de respuesta de coevaluación para alumnos

La vista que se encuentra en las figuras 4.8 y 4.9, y que corresponde a los mockups de las figuras 3.10 y 3.11 respectivamente, se encargan de la respuesta de coevaluación por parte de los alumnos. Esto responde a la historia de usuario HU05 donde un usuario busca completar el formulario y enviar sus respuestas al sistema.

### 5.1.3. Vista de toma de asistencia para profesores

La interfaz presente en 4.4, corresponde al mockup presente en la figura 3.12. En esta, se presenta la funcionalidad de toma de asistencia para los usuarios profesores. Particularmente, esta vista responde a la historia de usuario HU03 donde un profesor quiere registrar la asistencia de una evaluación, lo que se lleva a cabo con el formulario de esta interfaz.

### 5.1.4. Vista de creación de curso

En la figura 5.2 se presenta la interfaz de creación de una evaluación, que se corresponde con el mockup presentado en la figura 2. Esta interfaz responde a la historia de usuario HU01 donde un profesor quiere crear un curso. Además, como se muestra en la figura 4 el profesor puede importar un archivo con alumnos, o de otro modo, crear alumnos asociados al curso.

Sistema de evaluaciones Instrumentos Evaluaciones Cursos Integrantes admin Cerrar Sesión

## Nuevo curso

Ramo  
MA01 Matemáticas

Semestre  Anual  Otoño  Primavera  Verano

Año  
2022

Sección  
1

Figura 5.2: Vista de creación de evaluación.

### 5.1.5. Vista de creación de evaluación

En la figura 5.3 se presenta la interfaz de creación de una evaluación, que se corresponde con el mockup presentado en la figura 1. Esta interfaz responde a la historia de usuario HU02 donde un profesor quiere iniciar una evaluación, asociándola a un curso y un instrumento, tal como se puede hacer en esta vista.

Sistema de evaluaciones Instrumentos Evaluaciones Cursos Integrantes admin Cerrar Sesión

## Nueva evaluación

Nombre  
Evaluación 1 de trabajo colaborativo

Curso  
Curso: MA01-1 Matemáticas Primavera 2022

Instrumento  
Instrumento: 1 - Escala de evaluación de actividades centradas en la resolución c

Fecha de inicio  
20/06/2022 12:00

Fecha de término  
24/08/2022 12:00  
Fecha final en que los alumnos pueden responder la evaluación

Estado  Creada  Publicada  Finalizada

Crear evaluación

Figura 5.3: Vista de creación de evaluación.

### 5.1.6. Vista de envío de recordatorio

En la historia de usuario HU06 un administrador quiere enviar un recordatorio a través de un correo electrónico a los alumnos pendientes. Esta funcionalidad está implementada en la interfaz de la figura 4.10.

### 5.1.7. Vistas de ficha de evaluación y resultados

En la figura 4.1 se presenta la ficha de gestión de una evaluación, que se corresponde con el mockup presentado en la figura 3.14. En esta interfaz, luego de publicada la evaluación, se pueden ver los resultados de esta. Además, luego de dirigirse a la vista de resultados de la figura 4.11, estos pueden ser descargados, lo que corresponde a lo descrito en la historia de usuario HU07.

## 5.2. Validación del sistema con alumnos

Una parte fundamental de la aplicación es la de respuesta de coevaluación por parte de alumnos. Por lo mismo, se decidió realizar un estudio de usabilidad que se enfocara en esta funcionalidad. La metodología y resultados obtenidos se detallan a continuación.

### 5.2.1. Metodología

Para realizar esta validación se necesitó reclutar participantes, para lo cual se contactó a individuos que actualmente estudian en dos distintos institutos profesionales, siendo estos Duoc y Arcos. Las personas en cuestión realizaron un llamado dentro de sus cursos a participar. De esta manera se contó con cinco estudiantes dispuestos a colaborar.

La validación final consistió en un experimento realizado por cinco alumnos. La técnica utilizada fue la de *thinking aloud*, que consiste en pedirle a los participantes que utilicen el sistema mientras piensan en voz alta continuamente, es decir, que verbalicen sus pensamientos mientras navegan por la interfaz. Esta técnica fue elegida debido a su facilidad de llevar a cabo y que permite obtener información de la navegación de los usuarios sin tener que realizar preguntas puntuales. Además, los usuarios pueden entregar ideas de como solucionar el problema o que preferirían en cada caso. Finalmente, este es un método flexible que puede realizarse en cualquier etapa de desarrollo, en este caso se realizó al final de este [24].

Los alumnos asistieron de forma presencial a un punto de encuentro. Al encontrarse todos en la misma sala, la aplicación se dejó funcionando en un servidor propio. Para entregar acceso a la aplicación en los dispositivos móviles de los usuarios, se utilizó **ngrok**, que genera direcciones URL públicas seguras para endpoints de *localhost*.

### 5.2.2. Resultados y análisis

Los objetivos que los usuarios debían realizar eran ingresar un código entregado para entrar a la coevaluación, responder cada pregunta del formulario y enviar las respuestas al sistema. Luego de terminado el experimento, se determinó que los participantes pudieron lograr los objetivos fijados de forma correcta. Sin embargo, hubieron observaciones en cuanto a su desarrollo.

Durante el experimento se sugirió fijar los botones de anterior y siguiente de la interfaz, debido a que estos se encontraban en distintas posiciones dependiendo del largo de la pregunta. Además, una observación importante fue que, si el usuario no respondía todas las preguntas, el formulario pasaba de mostrar una pregunta por etapa a mostrar todas las preguntas en la misma vista, sin mencionar cuales eran las preguntas faltantes.

Finalmente, un usuario mencionó que el tamaño del texto de cada pregunta era demasiado grande y prefería que cada etapa contuviera más preguntas, no obstante, los otros usuarios no presentaron este problema.

Como conclusión, se ve que el experimento tuvo buenos resultados, ya que a pesar de las observaciones, todos los usuarios pudieron terminar la tarea asignada en un tiempo acotado de tiempo, mencionando que el proceso fue fácil de realizar.

Debido al tiempo acotado que la memoria posee y a la priorización del cumplimiento de otros objetivos, no se pudieron realizar validaciones con otros usuarios, del tipo profesor y administrador.

### **5.3. Resumen**

Con lo que se expone en este capítulo, se puede ver que se satisfacen las historias de usuario definidas en la sección 3.1.1 con las interfaces definidas en la sección 4.2 y 5.1. También, se pudo validar la usabilidad del sistema con un experimento realizado por alumnos. Además, se puede ver que se cumplió el objetivo principal, ya que se implementó el sistema conforme al diseño planteado. Por otro lado, también se cumplieron los objetivos específicos, con la excepción del objetivo específico 4 que consideraba una validación a través de métodos de usabilidad con profesores y psicólogos.

# Capítulo 6

## Conclusión

### 6.1. Discusión final

Para crear un sistema de coevaluaciones de grupos que ayude a la investigación de trabajo colaborativo, se realizó el trabajo que se detalla en esta memoria. Este toma como inspiración el sistema de coevaluaciones del departamento de ciencias de la computación de la universidad. En un comienzo, el sistema se realizó como una extensión del sistema del departamento, luego de avanzar en el proceso se decidió separarlos debido a que presentaba diferencias importantes en cuando al propósito de cada uno. Esto generó dificultades y retraso en el trabajo.

Como se muestra en el documento, se consideran cumplidos los objetivos generales y la mayoría de los objetivos específicos, con única excepción la de validación con profesores y psicólogos. Además, aún existen opciones de mejora en términos de usabilidad para las interfaces de alumnos.

Por otra parte, las validaciones muestran que el sistema cumple con los requisitos definidos. Además, estas señalan que el sistema presenta un buen funcionamiento para alumnos que respondan coevaluaciones, aunque aún se presentan posibilidades de mejorar este proceso. También, cabe mencionar que aún es necesario realizar estudios de usabilidad con otros usuarios, teniendo en cuenta otras interfaces de la aplicación.

Terminando el proceso, se tiene un sistema funcional que permite crear a los administradores y profesores gestionar evaluaciones, además de publicarlas a alumnos. En términos de usuarios alumnos, a estos se les permite ingresar al sistema sin poseer una cuenta y responder coevaluaciones en él. Además, se tienen funcionalidades extras como la de envío de recordatorios y generación de grupos aleatorios.

Contar con un sistema de referencia al comenzar con el desarrollo, el cual además pensaba ser extendido, tuvo sus ventajas. Una de ellas fue la de tener un modelo de datos que ya considerara como organizar la información necesaria para manejar coevaluaciones. Al mismo tiempo, otra ventaja fue la de poder visitar el código de las interfaces que tuvieran funcionalidades y elementos en común con las del nuevo sistema.

Sin embargo, a pesar de las ventajas que entregó el sistema de coevaluaciones mencionado, una gran desventaja fue la de intentar extender el sistema, con la recurrente duda de si sería mejor opción separarlos y empezar el nuevo sistema desde cero. Esto tomó tiempo y alargó el proceso.

Además, aunque contar con la implementación de funciones similares fuera de ayuda, también se debe considerar el tiempo para conocer y entender el código y la forma en que otro desarrollador implementa las funcionalidades. En ocasiones es más corto el proceso de implementar aquellas funcionalidades desde cero con ideas propias.

Otra conclusión es que durante la implementación existen momentos en que surgen dudas de diseño en las que el desarrollador o la desarrolladora puede tomar una decisión, pero esta puede ser deficiente en comparación a la decisión que podría tomarse si hubiera mayor interacción con los usuarios. Por más vueltas que se dé a un asunto, el usuario tiene características y conocimientos particulares que difieren de las del desarrollador o desarrolladora. Para este proyecto faltó interacción con los usuarios, de volver a realizar un proyecto de estas características, este sería uno de los principales ámbitos que intentaría mejorar.

## 6.2. Trabajo a futuro

Como trabajo a futuro es fundamental realizar una evaluación de usabilidad con los usuarios de tipo profesor y administrador. Además de llevar a cabo una evaluación distinta a la de entrevista con los alumnos.

En términos de usabilidad, también se presentaron ideas en la etapa de validación con alumnos, los cuales incluyen mejorar la disposición de los botones en la interfaz y especificar de mejor forma los errores en caso de intentar enviar el formulario de manera incompleta.

También, se hace necesario subir esta aplicación y realizar pruebas de carga que consideren una carga de alumnos respondiendo y enviando la coevaluación simultáneamente.

Por otro lado, se podría mejorar la interfaz de resultados, agregando gráficos de dispersión con las respuestas obtenidas. Los cuales pueden ser de utilidad para los usuarios administradores.

Además, sería interesante agregar la funcionalidad de importar grupos ya creados en una evaluación del sistema.

En el área de notificaciones, nace la idea de poder seleccionar los alumnos a los cuales se enviará el correo, además de mostrar en la vista la cantidad de veces que se han enviado recordatorios y la última fecha de envío.

# Bibliografía

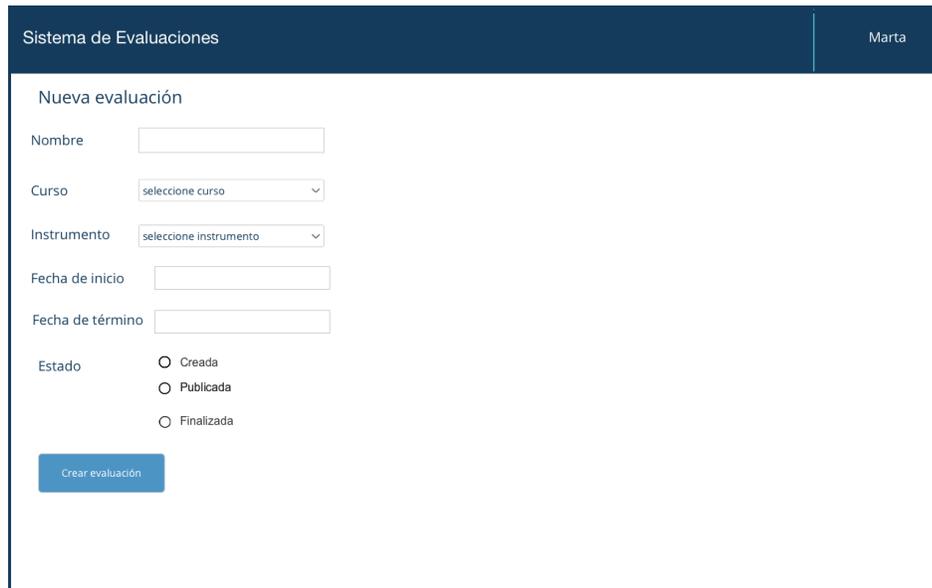
- [1] What is collaborative working? <https://instituteformcollaborativeworking.com/About-ICW/What-is-Collaborative-Working>.
- [2] Glosario. <https://docs.djangoproject.com/es/4.0/glossary/>, 2022.
- [3] Kahoot app. <https://kahoot.com/home/mobile-app/>, 2022.
- [4] Mentimeter. <https://www.mentimeter.com/>, 2022.
- [5] Ngrok. <https://ngrok.com/>, 2022.
- [6] Slido. <https://www.slido.com/>, 2022.
- [7] Writing your first django app, part 2, database setup. <https://docs.djangoproject.com/en/4.0/intro/tutorial02/>, 2022.
- [8] Clay Alvino. Estadísticas de la situación digital de Chile en el 2020-2021. <https://branch.com.co/marketing-digital/estadisticas-de-la-situacion-digital-de-chile-en-el-2020-2021/>, 2021.
- [9] Andre Bar'yudin. django-bootstrap v5. <https://github.com/zelenij/django-bootstrap-v5>, 2022.
- [10] Bootstrap. Introducción. <https://getbootstrap.esdocu.com/docs/5.1/getting-started/introduction/>, 2022.
- [11] Priyanka Carr and Gregory Walton. Cues of working together fuel intrinsic motivation. *Journal of Experimental Social Psychology*, 53, 07 2014.
- [12] Tejaswi Chaudhari. Mvc vs mvt architectural pattern. <https://medium.com/dsc-umit/mvc-vs-mvt-architectural-pattern-d306a56dce55>, 2021.
- [13] Bert Constantin. Polymorphic models for django. <https://github.com/django-polymorphic/django-polymorphic>, 2022.
- [14] Carolina Contreras. Desarrollo y perfeccionamiento de sistema de coevaluación de equipos para cursos del dcc. Memoria de ingeniería civil en computación, departamento de ciencias de la computación, fcm, Universidad de Chile, 2020.
- [15] Carolina Contreras. Sistema coevaluaciones. <https://github.com/ccrecao/SistemaCoevaluacion>, 2020.

- [16] Central de la búsqueda de Google. Mantén una estructura de url simple. <https://developers.google.com/search/docs/advanced/guidelines/url-structure>, 2022.
- [17] Deloitte Ireland LLP. Milliseconds make millions, 2020.
- [18] Django. Faq: General. <https://docs.djangoproject.com/es/4.0/faq/general/#django-appears-to-be-a-mvc-framework-but-you-call-the-controller-the-view-and-the-template>, 2022.
- [19] Django. Meet django. <https://www.djangoproject.com/>, 2022.
- [20] Django. Sending email. <https://docs.djangoproject.com/en/4.1/topics/email/>, 2022.
- [21] MDN Web Docs. Frameworks web de lado servidor. [https://developer.mozilla.org/es/docs/Learn/Server-side/First\\_steps/Web\\_frameworks](https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Web_frameworks), 2022.
- [22] MDN Web Docs. Introducción a django. <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>, 2022.
- [23] MDN Web Docs. Mvc. <https://developer.mozilla.org/es/docs/Glossary/MVC>, 2022.
- [24] Nielsen Norman Group. Thinking aloud: The 1 usability tool. <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool>, 2022.
- [25] JQuery. What is jquery? <https://jquery.com/>, 2022.
- [26] Selwin Ong. Python user agents. <https://github.com/selwin/python-user-agents>, 2022.
- [27] psychopg. psychopg2 - python-postgresql database adapter. <https://github.com/psychopg/psychopg2>, 2022.
- [28] Roberto Riquelme. Sistema de evaluación del desempeño de los miembros de un equipo de desarrollo de software. Memoria de ingeniería civil en computación, departamento de ciencias de la computación, fcm, Universidad de Chile, 2014.
- [29] Sebastián Sánchez. Extensión de un sistema de coevaluación de miembros de equipos de desarrollo de software. Memoria de ingeniería civil en computación, departamento de ciencias de la computación, fcm, Universidad de Chile, 2016.
- [30] Cloud Tables. Datatables — table plug-in for jquery. <https://datatables.net/>, 2022.

# ANEXOS

## Anexo A

A continuación se presentan los mockups de interfaces del sistema. En las figuras 1 y 2 se muestra los mockups de creación de una evaluación y un curso respectivamente.



El mockup muestra una interfaz de usuario para crear una nueva evaluación. El encabezado del sistema es 'Sistema de Evaluaciones' y el nombre de usuario es 'Marta'. El formulario 'Nueva evaluación' incluye los siguientes campos:

- Nombre: campo de texto.
- Curso: menú desplegable con el texto 'seleccione curso'.
- Instrumento: menú desplegable con el texto 'seleccione instrumento'.
- Fecha de inicio: campo de texto.
- Fecha de término: campo de texto.
- Estado: tres botones de radio con las opciones 'Creada', 'Publicada' y 'Finalizada'.

Debajo de los campos hay un botón azul que dice 'Crear evaluación'.

Figura 1: Mockup de creación de evaluación.

The image shows a web application interface for creating a new course. At the top, there is a dark blue header with the text 'Sistema de Evaluaciones' on the left and 'Marta Sánchez' on the right. Below the header, the main content area is titled 'Nuevo curso'. It contains the following form elements: a dropdown menu for 'Ramo' with the placeholder text 'seleccione ramo'; a text input field for 'Año' containing the value '2022'; a text input field for 'Sección' containing the value '1'; and two radio button options for 'Semestre', labeled 'Otoño' and 'Primavera'. At the bottom left of the form area, there is a blue button labeled 'Crear curso'.

Figura 2: Mockup de creación de curso.

## Anexo B

En este capítulo se presenta el modelo de datos completo de la aplicación, el cual se encuentra en la figura 3.

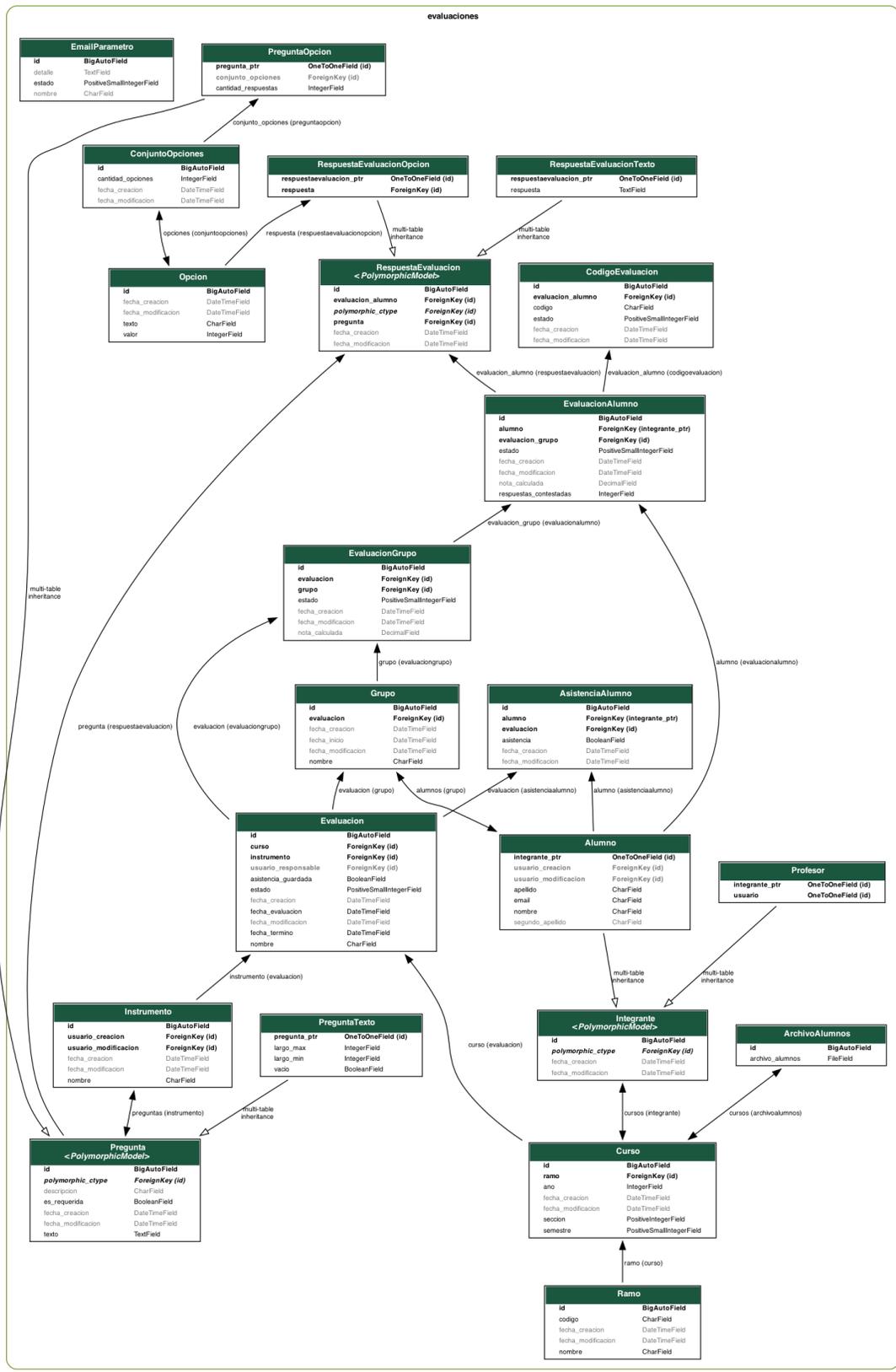


Figura 3: Diagrama del modelo de datos.

## Anexo C

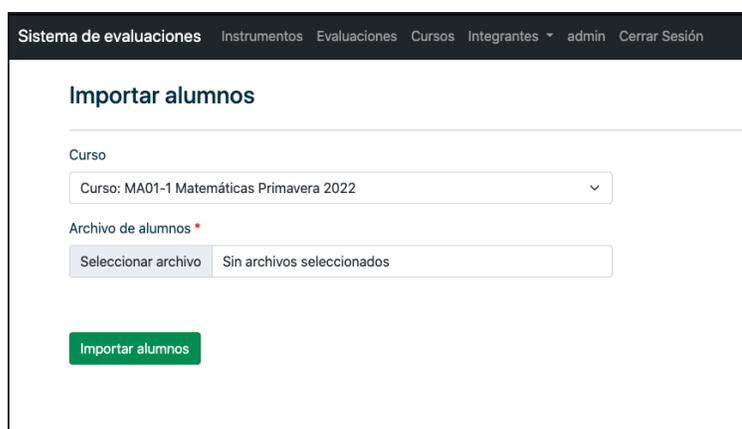
En este capítulo se muestran las interfaces de la aplicación, los datos que presenta cada vista fueron inventados para la validación del proyecto.

En la figura 4.5 se puede ver la sección de grupos cuando aún no se han generado grupos en la evaluación.

También, la 4 muestra la vista de importación de alumnos.

Por otro lado, las figura 5 representan a la interfaz de ingreso de código en versión móvil.

Finalmente, en las figurass 6, 7, 8 y 9 se encuentran las tablas de evaluaciones, cursos, profesores y alumnos respectivamente.



The screenshot shows the 'Importar alumnos' (Import students) interface within the 'Sistema de evaluaciones' (Evaluation System) application. The top navigation bar includes 'Instrumentos', 'Evaluaciones', 'Cursos', 'Integrantes', 'admin', and 'Cerrar Sesión'. The main content area features a dropdown menu for 'Curso' set to 'Curso: MA01-1 Matemáticas Primavera 2022'. Below this is a section for 'Archivo de alumnos' (Students file) with a 'Seleccionar archivo' button and a text field displaying 'Sin archivos seleccionados'. At the bottom, there is a green 'Importar alumnos' button.

Figura 4: Vista de importación de archivo de alumnos.



Figura 5: Vista de ingreso de código para alumnos (versión móvil).

Sistema de evaluaciones Instrumentos Evaluaciones Cursos Integrantes admin Cerrar Sesión

### Evaluaciones [+ agregar evaluación](#)

Mostrar 10 registros Buscar:

ID	Nombre	Curso	Instrumento	Fecha	Fecha límite	Estado	Acciones
1	Evaluación 1 de Prueba	Matemáticas MA01-1	Escala de evaluación de actividades centradas en la resolución colaborativa de problemas	14 de Junio de 2022 a las 12:00	30 de Junio de 2022 a las 23:59	Publicada	
2	Evaluación 2 de prueba	Matemáticas MA01-1	Escala de evaluación de actividades centradas en la resolución colaborativa de problemas	7 de Julio de 2022 a las 20:06	21 de Julio de 2022 a las 20:06	Creada	
3	Evaluación de prueba 3	Matemáticas MA01-1	Escala de evaluación de actividades centradas en la resolución colaborativa de problemas	22 de Julio de 2022 a las 12:00	30 de Julio de 2022 a las 12:00	Creada	
4	Evaluación trabajo colaborativo	Matemáticas MA01-1	Escala de evaluación de actividades centradas en la resolución colaborativa de problemas	27 de Junio de 2022 a las 12:00	27 de Julio de 2022 a las 12:00	Publicada	
5	Evaluación de trabajo colaborativo 5	Lenguaje LE01-2	Escala de evaluación de actividades centradas en la resolución colaborativa de problemas	4 de Mayo de 2022 a las 14:30	18 de Mayo de 2022 a las 23:59	Finalizada	

Mostrando 1 a 5 de 5 registros Anterior **1** Siguiente

Figura 6: Tabla de evaluaciones.

Sistema de evaluaciones Instrumentos Evaluaciones Cursos Integrantes admin Cerrar Sesión

### Cursos [+ agregar curso](#)

Mostrar 10 registros Buscar:

ID	Ramo	Semestre	Año	Sección	Fecha modificación	Acciones
1	MA01 Matemáticas	Primavera	2022	1	12 de Septiembre de 2022 a las 21:19	
2	LE01 Lenguaje	Anual	2022	2	12 de Septiembre de 2022 a las 23:06	
3	IN Inglés	Primavera	2022	1	26 de Septiembre de 2022 a las 13:34	
4	IN Inglés	Otoño	2021	1	26 de Septiembre de 2022 a las 13:34	

Mostrando 1 a 4 de 4 registros Anterior **1** Siguiente

Figura 7: Tabla de cursos.

Sistema de evaluaciones Instrumentos Evaluaciones Cursos Integrantes admin Cerrar Sesión

### Profesores [+ agregar profesor\(a\)](#)

Mostrar 10 registros Buscar:

ID	Nombre de usuario	Nombre Completo	Email	Fecha último inicio de sesión	Acciones
26	mlara	Marta Lara	mlara@gmail.com	15 de Julio de 2022 a las 20:18	
27	galvardo@gmail.com	Guillermo Alvarado	galvardo@gmail.com		
30	osanchez	Osvaldo Sánchez	osanchezzz_56@hotmail.com		
31	jdiaz	Javiera Diaz	jdiaz23@gmail.com	26 de Mayo de 2022 a las 11:40	

Mostrando 1 a 4 de 4 registros Anterior **1** Siguiente

Figura 8: Tabla de profesores.

Sistema de evaluaciones Instrumentos Evaluaciones Cursos Integrantes admin Cerrar Sesión

**Alumnos** [+ agregar alumno\(a\)](#) [importar alumnos](#)

Mostrar 10 registros Buscar:

ID	Email	Nombre	Apellido	Segundo Apellido	Cursos	Cantidad de Cursos	Acciones
1	mjm@hola.com	Martín	Álvarez		MA01 Matemáticas 2022 Primavera-1, LE01 Lenguaje 2022 Anual-2	2	 
2	valentinanana24@gmail.com	Adela	Llao	Pino	MA01 Matemáticas 2022 Primavera-1	1	 
3	adolfo@gmail.com	Adolfo	Roa	Martínez	LE01 Lenguaje 2022 Anual-2, IN Inglés 2022 Primavera-1	2	 
4	ralvarado@hotmail.com	Roberto	Alvarado	Muñoz	MA01 Matemáticas 2022 Primavera-1, LE01 Lenguaje 2022 Anual-2, IN Inglés 2022 Primavera-1	3	 
5	jmedina@outlook.com	Javiera	Medina	Guerrero	MA01 Matemáticas 2022 Primavera-1, LE01 Lenguaje 2022 Anual-2	2	 
6	fpoblete@gmail.com	Fernando	Poblete	Carvajal	MA01 Matemáticas 2022 Primavera-1	1	 
7	abustos46@outlook.es	Alejandra	Bustos	Quezada	MA01 Matemáticas 2022 Primavera-1, LE01 Lenguaje 2022 Anual-2	2	 
8	bgodoy@gmail.com	Benjamín	Godoy	Lagos	LE01 Lenguaje 2022 Anual-2, IN Inglés 2022 Primavera-1	2	 
9	lsaez@gmail.com	Lucas	Saez	Guzmán	IN Inglés 2022 Primavera-1	1	 
10	tvera.011@gmail.com	Tomás	Vera	Ortiz	MA01 Matemáticas 2022 Primavera-1, LE01 Lenguaje 2022 Anual-2	2	 

Mostrando 1 a 10 de 27 registros Anterior **1** 2 3 Siguiente

Figura 9: Tabla de alumnos.

## Anexo D

A continuación se presentan notificaciones de ejemplo que la aplicación permite enviar a alumnos. En la figura 10 se ve un ejemplo de email de recordatorio de coevaluación a alumno.

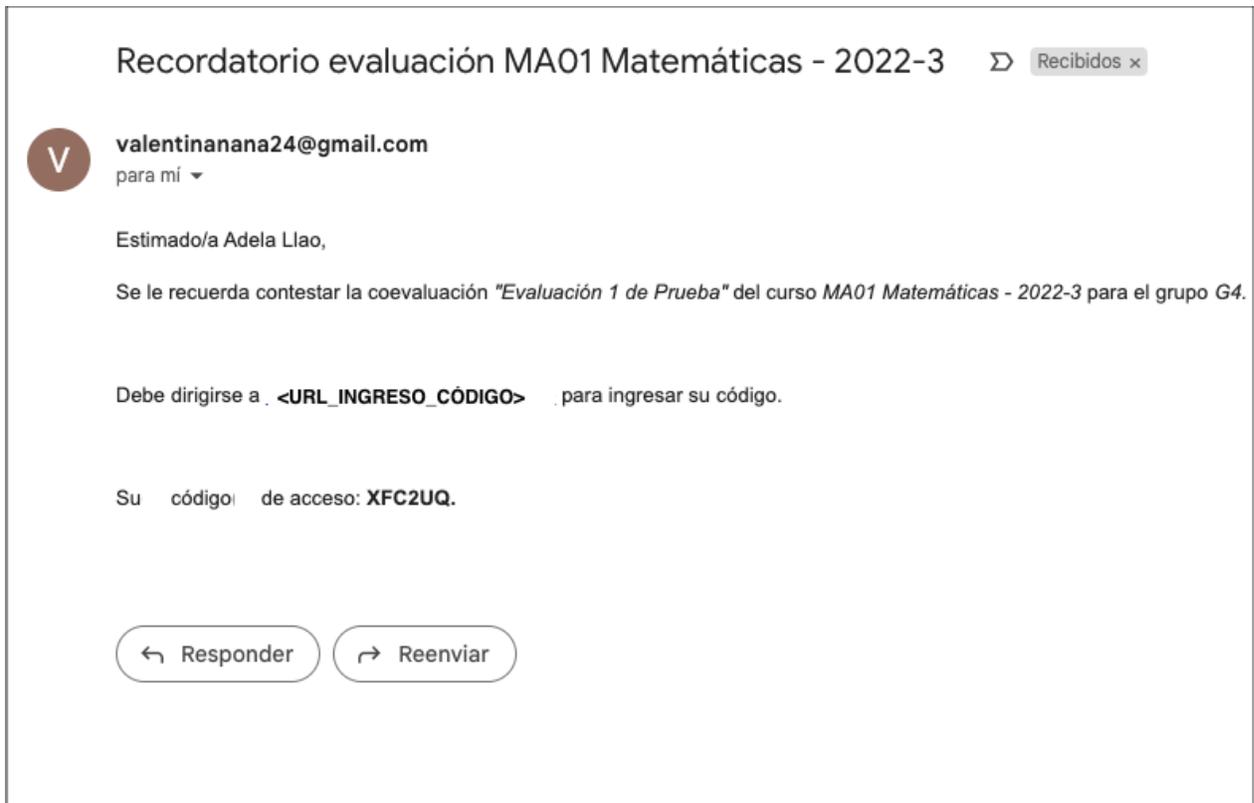


Figura 10: Ejemplo de email de recordatorio a alumno.

## Anexo E

En este capítulo se especifican las instrucciones para levantar el sistema.

- Instalar requerimientos

```
39 pip install -r requirements.txt
```

Código 1: Comando para instalar requerimientos

- Hacer migraciones de Django

```
40 python manage.py makemigrations  
41 python manage.py migrate --run-syncdb
```

Código 2: Comando para hacer migraciones

- Cargar base de datos con fixtures

```
42 python manage.py loaddata Instrumento_initial.json --app
    evaluaciones.Instrumento
43 python manage.py loaddata Opcion_initial.json --app evaluaciones.
    Opcion
44 python manage.py loaddata Pregunta_initial.json --app
    evaluaciones.Pregunta
45 python manage.py loaddata PreguntaOpcion_initial.json --app
    evaluaciones.PreguntaOpcion
```

Código 3: Comando para cargar datos a la base de datos