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

MÓDULO DE ASIGNACIÓN DE COMISIONES EXAMINADOREAS EN EL SISTEMA DE TITULACIÓN DCC

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

VICENTE ESTEBAN OLIVARES GÓMEZ

PROFESOR GUÍA:
MARÍA CECILIA BASTARRICA PINEYRO

MIEMBROS DE LA COMISIÓN:
CÉSAR RAMÓN GUERRERO SALDIVIA
FRANCISCO JAVIER GUTIÉRREZ FIGUEROA

SANTIAGO DE CHILE
2025

Resumen

La asignación de comisiones examinadoras para los trabajos de título es un proceso con importancia, pues son estas las que evalúan los trabajos de título y las defensas de los mismos, siendo un factor clave en el éxito de la titulación de un estudiante. Además, al realizar la asignación se debe tener en cuenta que los integrantes de las comisiones examinadoras cumplan con tener conocimiento en el área del tema del trabajo de título y que tengan tiempo disponible para evaluarlo. Por lo tanto, la elección de integrantes de las comisiones examinadoras es una tarea de complejidad no menor. Actualmente, esta tarea es realizada con una planilla Excel, lo que resulta en un proceso tedioso, propenso a errores y poco eficiente, más aún con el constante aumento de estudiantes en el Departamento de Ciencias de la Computación (DCC).

En este contexto, se propone desarrollar y desplegar una herramienta que permita asignar las comisiones examinadoras de manera interactiva y eficiente, y que sea capaz de integrarse con los sistemas existentes relacionados con el proceso de titulación del DCC, como el Sistema de Titulación y el Sistema de Seguimiento de Memorias (SSM). La solución fue implementada usando Django como framework web y PostgreSQL como base de datos. Esta corresponde a una extensión del sistema de titulación del DCC, que agrega un módulo para asignar comisiones examinadoras a los trabajos de título, tomando como base un sistema piloto desarrollado por un equipo del curso CC5401 Ingeniería de Software II en el semestre de otoño de 2025. Al piloto se le agregaron funcionalidades como validaciones al momento de asignar una comisión, un gráfico de carga de los evaluadores para distribuir a los evaluadores de manera equitativa y que pueda exportar las comisiones asignadas en formato CSV y también de forma directa al SSM. Para que el SSM pueda recibir las comisiones asignadas de forma directa, se implementó una API que permite recibir las comisiones asignadas en formato JSON. Tanto la extensión del sistema de titulación como la adaptación del SSM fueron desplagadas primero en un servidor de test para evaluar su funcionalidad, para luego ser desplegadas en el servidor de producción del DCC.

Para evaluar la solucion implentada, se utilizó el módulo de comisiones examinadoras para asignar las comisiones examinadoras a los trabajos de título del semestre de primavera 2025 del DCC. La evaluación mostró que aunque la herramienta requería de algunas mejoras, como optimizaciones en controladores, asignar las comisiones con ella significó menos trabajo y tomó menos tiempo que con la planilla Excel. El proyecto fue desplegado en el servidor de producción del DCC y permitirá que la asignación de comisiones examinadoras no signifique un gasto de tiempo y esfuerzo innecesario para la coordinación de titulación.

Una dedicatoria corta.

Agradecimientos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Tabla de Contenido

Índice de Ilustraciones

Listings

Capítulo 1

Introducción

1.1. Contexto

El hito final de una carrera en la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, como lo es Ingeniería Civil en Computación, es el Trabajo de Titulación. Este es un proceso que se divide en tres etapas: el curso de Introducción al Trabajo de Título, el curso de Trabajo de Título y el Examen de Título. La coordinación de titulación del departamento debe asignar una comisión examinadora a cada estudiante cursando Trabajo de Título y enviarlas a la jefatura de estudios del departamento con plazo máximo la semana número 12 de cada semestre. Cada comisión tiene el rol de evaluar el informe redactado por el estudiante en el curso de Trabajo de Título y la defensa en el examen de título. Cada comisión está compuesta por el profesor guía, el profesor coguía, en caso de tener, y al menos dos integrantes más. Estos integrantes adicionales pueden ser académicos/as de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile (FCFM) o profesores expertos externos, con la condición de que al menos uno debe ser académico/a de la FCFM con jerarquía de profesor/a. [6, p. 17]

Los/as estudiantes requieren aprobar la defensa de su trabajo de título para obtener el título [6, pp. 15-16], por lo que es crucial que la comisión examinadora sea capaz de evaluar el trabajo de título y su defensa de manera adecuada. Esto involucra factores como que los/as profesores/as tengan conocimiento en el área del tema del trabajo de título y que tengan tiempo disponible para evaluarlo. Por lo tanto, la elección de integrantes de las comisiones examinadoras es una tarea de complejidad no menor.

En el Departamento de Ciencias de la Computación de la Universidad de Chile (DCC), la selección de integrantes de las comisiones examinadoras es realizada de forma manual y el registro de estos con hojas de cálculo tipo Excel por el coordinador de titulación. Debido al gran aumento de estudiantes en el DCC en los últimos años, el número de memoristas también ha crecido, haciendo que la asignación manual de comisiones sea una tarea tediosa, propensa a errores y poco eficiente. A modo de ejemplo del aumento de estudiantes, en el semestre de primavera de 2024 había un total de 60 inscritos en los cursos de Trabajo de Título y en el semestre de otoño de 2025 aumentó a 103, según el catálogo de cursos de

UCampus de la FCFM [4, 5].

Es por lo anterior que aparece la necesidad de una herramienta que permita facilitar la asignación de comisiones examinadoras, aportando información sobre las áreas de conocimiento de los/as profesores/as y la cantidad de comisiones que han sido asignadas a cada profesor/a, y que se encuentre integrada con las herramientas ya existentes, como el sistema de titulación.

1.2. Objetivos

1.2.1. Objetivo General

El objetivo de esta memoria es desarrollar y desplegar una herramienta que permita que la asignación de comisiones examinadoras para las memorias sea una tarea eficiente. Esto incluye que la herramienta se comunique con los sistemas existentes relacionados con el proceso de titulación, como el Sistema de Titulación y el Sistema de Seguimiento de Memorias (SSM).

1.2.2. Objetivos Específicos

Para cumplir con el objetivo, se plantean los siguientes objetivos específicos:

- Desarrollar una herramienta que permita asignar comisiones de forma interactiva.
- Integrar la herramienta con los sistemas existentes relacionados con el proceso de titulación, en particular con el Sistema de Titulación y el Sistema de Seguimiento de Memorias (SSM).
- Dejar la herramienta desarrollada en producción.

1.3. Solución Propuesta

A modo general, la solución propuesta consiste en cuatro partes. La primera parte corresponde a la adición de funcionalidades en el módulo de asignación de comisiones examinadoras, la segunda es la integración del módulo con el Sistema de Titulación, la tercera es la integración con el Sistema de Seguimiento de Memorias (SSM) y la cuarta es el despliegue de la solución en los servidores del DCC.

1.3.1. Desarrollo del módulo de asignación de comisiones examinadoras

La primera funcionalidad que se agregará al módulo de asignación de comisiones examinadoras es la validación de las comisiones asignadas. Se debe verificar que las comisiones tengan al menos dos integrantes además de los guías. De estos integrantes, al menos uno debe tener jerarquía de profesor, es decir, que sean académicos de jornada completa (AJC) o académicos de jornada parcial (AJP). También se debe verificar que no se repitan académicos en la misma comisión.

Luego, se implementará un filtro en la interfaz principal del módulo que permita mostrar todas las memorias, solo las memorias con comisión completamente asignada o solo las memorias con comisión incompleta. Este filtro permitirá navegar con mayor facilidad entre las memorias al momento de asignar comisiones examinadoras.

Además, en la misma interfaz se agregará un gráfico que muestre la carga de los académicos. Específicamente, por cada académico se mostrará la cantidad de comisiones que guía y la cantidad de comisiones que integra. El gráfico será de barras y mostrará a los académicos ordenados de forma decreciente por la carga que tienen. De esta forma, se podrá identificar académicos con carga excesiva y, por lo tanto, evitar que se les asigne a más comisiones.

1.3.2. Integración en el Sistema de Titulación

Actualmente el sistema de titulación solo cuenta con un listado de estudiantes de Introducción al Trabajo de Título, por lo que se debe agregar a los estudiantes de Trabajo de Título para integrar correctamente el módulo de asignación de comisiones examinadoras. Para esto, se agregará una pestaña con un listado de integrantes de Trabajo de Título en los distintos periodos académicos. Para mantener este listado actualizado, se implementará un cronjob que se encargue de obtener periódicamente desde UCampus a los estudiantes que están cursando Trabajo de Título.

Luego, se agregará la posibilidad de exportar las comisiones examinadoras asignadas, que se podrá realizar de dos formas. Por un lado, se podrá descargar un archivo CSV con las comisiones, y por otro lado, estas podrán ser exportadas directamente hacia el Sistema de Seguimiento de Memorias, mediante a través de su API.

1.3.3. Integración con el Sistema de Seguimiento de Memorias

El Sistema de Seguimiento de Memorias (SSM) es un software utilizado por la coordinación de estudios para monitorear las fechas de entrega de los informes de Trabajo de Título y coordinar la corrección de estos documentos con los integrantes de la comisión examinadora respectiva. Antes de esta memoria, para que el SSM pudiera obtener las memorias y las comisiones asignadas a cada una, se debía subir manualmente un archivo CSV.

Para hacer la exportación de las comisiones examinadoras al Sistema de Seguimiento

de Memorias más fácilmente, se implementará un endpoint en el sistema de titulación para obtener las comisiones examinadoras asignadas por una API REST. Además, desde el Sistema de Seguimiento de Memorias se agregará la opción de importar las comisiones desde esta API.

1.3.4. Despliegue de la solución

Por último, la solución desarrollada se desplegará en los servidores del DCC para que pueda ser utilizada por la coordinación de titulación. Para este despliegue, se utilizarán contenedores de Docker, uno para la aplicación de Django y otro para la base de datos de PostgreSQL.

Capítulo 2

Situación Actual

Actualmente existen varios proyectos y sistemas relacionados con el proceso de titulación en el DCC. Dentro de estos se encuentran el sistema de titulación del DCC, el Sistema de Monitoreo de Memorias [3], un sistema de recomendación de comisiones [2] y un sistema de asignación de comisiones desarrollado por un grupo del curso CC5401 Ingeniería de Software II. A continuación se describen las características y las limitaciones de estos sistemas, relacionadas con la asignación de comisiones examinadoras.

2.1. Sistema de Titulación DCC

El sistema de titulación del DCC es un sistema web en producción que ofrece distintas funcionalidades dependiendo de si se es estudiante, profesor o coordinador de titulación. Como estudiante, se pueden ver un listado de temas para trabajos de título, solicitar la inscripción en un tema, subir propuestas su memoria y su informe final del curso Introducción al Trabajo de Título. Además se puede ver su ficha personal, que incluye la etapa actual del trabajo de título, el tema del trabajo de título junto a su guía. Además, se puede descargar la propuesta de memoria y el informe final de Introducción al Trabajo de Título, en caso de haber subido.

Si se es profesor, se pueden publicar temas de trabajo de título y ver solicitudes de inscripción de estudiantes y ver a los memoristas que se está guiando junto a sus respectivas fichas. Si se es coordinador de titulación, se puede ver el listado de todos los memoristas, sumado a lo que puede ver un profesor.

Figura 2.1: Listado de temas de trabajo de título en el sistema de titulación del DCC.

Figura 2.2: Ficha de un estudiante en el sistema de titulación del DCC.

Actualmente, las funcionalidades de este sistema están enfocadas en el ramo Introducción al Trabajo de Título, por lo que no cuenta con las funcionalidades de asignar miembros de comisiones examinadoras ni de visualizar las comisiones asignadas a cada estudiante, que corresponden a Trabajo de Título.

2.2. Sistema de Seguimiento de Memorias

El Sistema de Seguimiento de Memorias (SSM), es un sistema desarrollado por Matías Rivas Aguilera en 2024 [3] y extendido por Diego Orellana Vidal en 2025 como sus respectivas memorias para optar al título de Ingeniero Civil en Computación [1]. En la figura 2.3 se puede

apreciar la vista principal del SSM. Este sistema se encuentra en producción desde julio de 2025 pero no ha tenido uso hasta la fecha de redacción de esta memoria. Este sistema permite a la jefatura de estudios monitorear los plazos de entrega de los informes finales de Trabajo de Título y gestionar la corrección de los mismos. Para realizar esta tarea, el sistema requiere saber cuales son las comisiones examinadoras asignadas a cada trabajo de título y el método que se utiliza para obtenerlos es que el usuario los ingrese, por lo que el sistema permite agregar, eliminar y modificar memorias, estudiantes, profesores y miembros de comisiones examinadoras. El ingreso de esta información es realizado por la jefatura de estudios, una vez la coordinación de titulación realice la asignación de las comisiones. Esto sucede a más tardar en la semana académica número 12 de cada semestre.

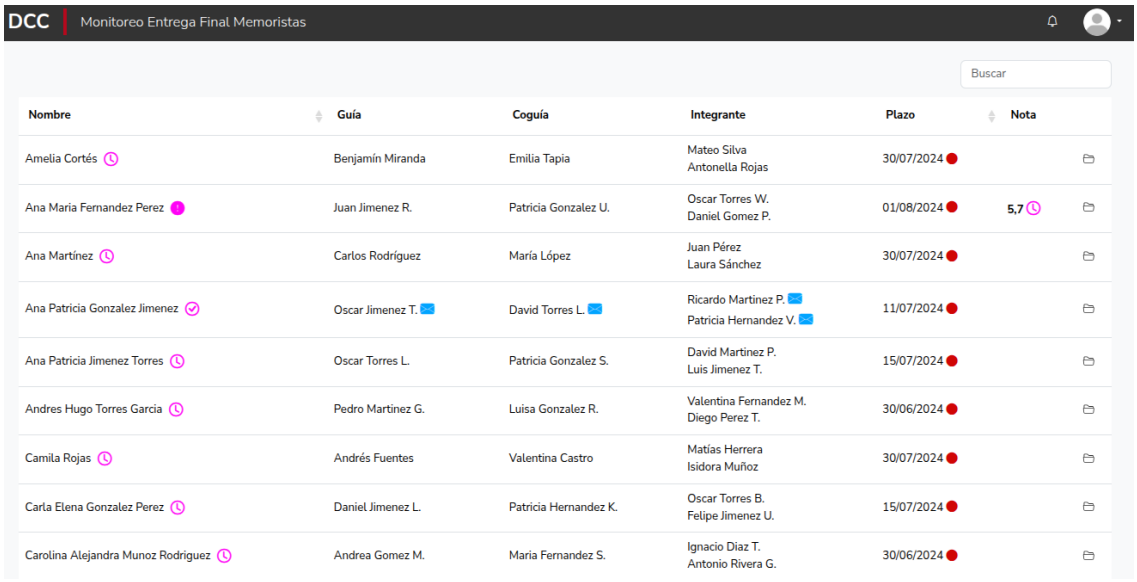


Figura 2.3: Vista principal del SSM que muestra un listado de memoristas.

El SSM ofrece dos formas de ingresar miembros de comisiones examinadoras. La primera es de forma directa, llenando un formulario que permite asignar un/a profesor/a a la vez a una memoria a la que se asigna. La segunda es mediante la subida de un archivo CSV que sirve para agregar varias memorias simultáneamente, como se muestra en la figura 2.4. Cada línea del archivo representa una memoria y debe contener las siguientes columnas:

- Estudiante
- Correo Estudiante
- Tema
- Guías
- Correos Guías
- Coguías
- Correos Coguías
- Integrantes
- Correos Integrantes

La principal limitación del SSM respecto a la asignación de comisiones examinadoras es que fue desarrollado para que sea utilizado por la jefatura de estudios [3, p. 3]. Como la coordinación de titulación es la encargada de asignar las comisiones, el sistema fue diseñado para facilitar el registro de comisiones ya definidas y no la asignación de estas.

Este enfoque se puede ver en la asignación de integrantes a una comisión mediante la subida de un archivo CSV, que permite agregar varias memorias simultáneamente, haciendo

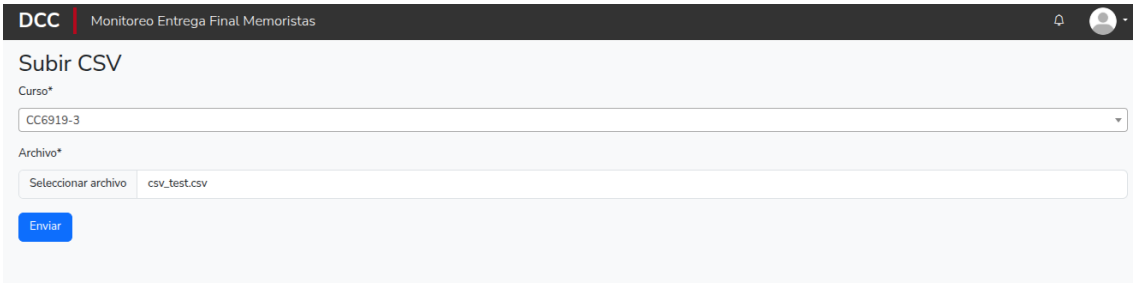


Figura 2.4: Formulario de subida de archivo CSV del SSM.

bastante sencillo el registro de comisiones pero no ofrece ayuda alguna para generar el archivo CSV. Además, el sistema no ofrece ningún tipo de validación sobre las restricciones de la asignación de comisiones examinadoras, como que cada comisión debe tener al menos un/a profesor/a con jerarquía de profesor/a. [6, p. 17]

En el caso de la asignación de integrantes a una comisión mediante un formulario, también se puede notar que el SSM no está diseñado para asignar integrantes a comisiones examinadoras, ya que el formulario no permite asignar varios integrantes a una comisión a la vez como se puede apreciar en la figura 2.5. Como por cada comisión se requieren al menos 2 integrantes además de los guías, se debe llenar al menos tres veces el formulario por cada comisión, lo que resulta ineficiente y tedioso.

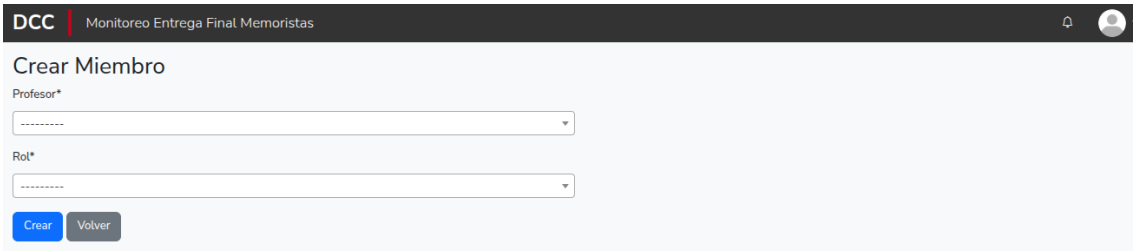


Figura 2.5: Formulario del SSM para agregar a un integrante a la comisión de una memoria, seleccionando el profesor deseado y el rol que tendrá.

2.3. Sistema de recomendación de comisiones

El Sistema de recomendación de comisiones [2] es un sistema que no se encuentra en producción y fue desarrollado por Rodrigo Oportot González en 2024 como su memoria para optar al título de Ingeniero Civil en Computación. Este sistema propone 7 profesores/as candidatos/as para la comisión examinadora de una memoria, basándose en el área de conocimiento de los/as profesores/as y el tema del trabajo de título. Para esto, utiliza procesamiento de lenguaje natural y Machine Learning.

Este sistema es conveniente para la asignación de comisiones examinadoras, ya que ofrece candidatos/as según su área de conocimiento, permitiendo tener comisiones con mayor conocimiento en el tema del trabajo de título. No obstante, estas propuestas no toman en cuenta la cantidad de comisiones que han sido asignadas a cada profesor/a, lo que puede resultar

en profesores/as con mucha carga y que no tengan el tiempo necesario para examinar las memorias. Además, cae en la misma falta que el SSM, ya que tampoco se tiene en cuenta restricciones sobre la conformación de comisiones examinadoras, como que cada comisión debe tener al menos un/a profesor/a con jerarquía de profesor/a. [6, p. 17]

2.4. Sistema de Asignación de Comisiones

El Sistema de Asignación de Comisiones es un sistema piloto que fue desarrollado por un equipo del curso CC5401 Ingeniería de Software II del DCC durante el semestre de otoño de 2025. Los integrantes de este equipo fueron Daniel Sarazua Y., Ignacio Silva Ghisolfo, J. Andreu Díaz P., Manuel Saavedra Soto, Monserrat Montero y Ricardo Fernández Reyes. El piloto trata de una primera versión de una extensión del sistema de titulación del DCC que agrega un módulo para asignar comisiones examinadoras para los trabajos de título. La interfaz principal de este módulo lista los estudiantes que están cursando el ramo Trabajo de Título del periodo académico seleccionado. Cada fila indica el título de una memoria, el nombre del estudiante, el nombre del guía, el nombre del coguía si es que se tiene y los miembros de la comisión examinadora si estos fueron asignados. Al final de cada fila se encuentra un botón que permite agregar una comisión examinadora en caso de que no se le haya asignado una. Esta interfaz se puede apreciar en la figura 2.6. Sobre la tabla, a la izquierda se encuentra una barra de búsqueda que permite buscar filas por el contenido de cualquiera de sus campos.

Figura 2.6: Interfaz principal del Sistema de Asignación de Comisiones.

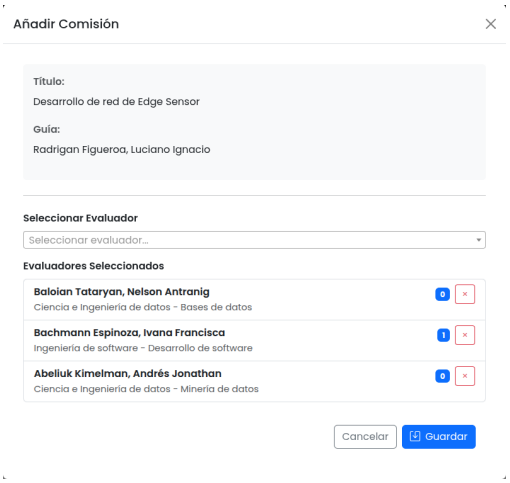
Sobre la barra de búsqueda se encuentran los botones de *Sincronizar* y *Exportar*. El botón *Sincronizar* actualiza el listado de estudiantes que cursaron o se encuentran cursando el ramo Trabajo de Título en el periodo académico seleccionado. Para lograr esto, primero se obtienen a los integrantes del ramo desde la API de UCampus. Luego, se obtiene el tema de cada estudiante desde el sistema de titulación, buscando el último registro de aprobación de Introducción al Trabajo de Título asociado al estudiante. La funcionalidad de este botón es

útil, pero debe hacerse manualmente. Además, fuera de este botón, no hay otro mecanismo en el sistema de titulación para obtener a los integrantes del ramo Trabajo de Título. El botón *Exportar* permite exportar la lista de comisiones asignadas a un archivo CSV. Este tiene las siguientes columnas:

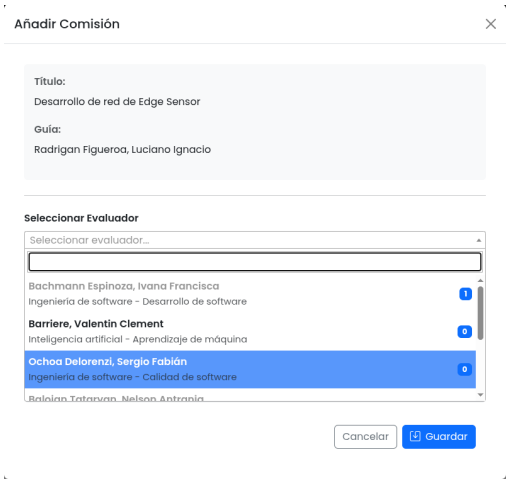
- Título
- Periodo
- Estudiante
- Guía
- Coguía
- Estado
- Evaluadores
- Correos

El archivo exportado tiene el potencial de ser utilizado para importar comisiones examinadoras asignadas al SSM, ya que contiene toda la información requerida por este sistema e incluso más, sin embargo, las columnas tienen distintos nombres y la información se encuentra en otro formato. Por ejemplo, el archivo CSV generado tiene el caracter ; (punto y coma) como separador, mientras que el SSM requiere que ese caracter sea utilizado para separar los nombres y los correos de los integrantes de la comisión examinadora en las columnas *Integrantes* y *Correos Integrantes*, respectivamente.

Este sistema fue diseñado para ser utilizado por la coordinación de titulación del DCC, por lo que al agregar o editar comisiones toma en cuenta elementos como la cantidad de comisiones a las que ha sido asignada cada profesor/a y que se puedan agregar varios miembros a una comisión en un mismo formulario. Además, en el formulario se permite buscar profesores por nombre y área de conocimiento, como se puede ver en la figura 2.7b.



(a) Formulario que permite agregar varios miembros a una comisión examinadora.



(b) Selector que permite buscar profesores por nombre y área de conocimiento.

Figura 2.7: Formulario de asignación de comisiones examinadoras en el Sistema de Asignación de Comisiones.

Como se mencionó anteriormente, el Sistema de Asignación de Comisiones es un sistema piloto, por lo que no tiene todas las características deseadas. En primer lugar, faltan validaciones al momento de asignar miembros a una comisión examinadora, haciendo posible asignar a los profesores guía y coguía como integrantes de la misma comisión examinadora, quedando

registrados dos veces en la comisión. En segundo lugar, no se encuentra bien integrado con el sistema de titulación, sobre el cual fue desarrollado. Esto se debe a que el sistema permite asignar comisiones examinadoras a memoristas y exportarlas, pero las comisiones asignadas no son visibles en otros módulos del sistema de titulación, en los que sería deseable verlas, como por ejemplo las fichas de los estudiantes. En tercer lugar, el sistema no se encuentra directamente integrado con el SSM.

Capítulo 3

Diseño

En este capítulo se abordarán la arquitectura, el modelo de datos y el diseño de interfaces del sistema de titulación. En la sección 3.1 se describirá la arquitectura del sistema de titulación y como se relaciona con otros sistemas. Luego, en la sección 3.2 se describirá el modelo de datos del sistema de titulación y los cambios por los que este pasó. Finalmente, en la sección 3.3 se describirá el diseño de interfaces del sistema de titulación.

3.1. Arquitectura

A continuación se describirá la arquitectura del sistema de titulación. Primero se describirá la arquitectura interna del sistema, es decir, los módulos que componen el sistema. Luego, se describirá la arquitectura externa del sistema, es decir, cómo se relaciona con otros sistemas.

El sistema de titulación es un sistema web hecho con el framework Django y utiliza una base de datos PostgreSQL. Internamente se encuentra formado por varios módulos como se puede ver en la figura 3.1, cada uno sigue la arquitectura Modelo-Vista-Controlador (MVC), puesto que el framework Django ya implementa esta arquitectura. Los módulos son los siguientes:

- Core
- Servicios
- Comisiones
- Titulación
- Trabajo de Título
- Docencia
- Evaluación
- Departamento
- Investigación

El módulo Core actúa como base transversal, conteniendo la configuración y funcionalidades utilizadas por todo el sistema. Por su parte, el módulo de servicios gestiona las notificaciones, mientras que el de docencia se ocupa de los periodos académicos y los cursos. Asimismo, el módulo de investigación administra las áreas de investigación, y el de departamento es responsable de la gestión de estudiantes, funcionarios y sus respectivos roles. De igual manera, el módulo de evaluación se encarga de gestionar a los evaluadores de las memorias en ambos ramos de titulación. El módulo de titulación abarca la lógica del ramo Introducción al

Trabajo de Título, incluyendo temas, solicitudes y memorias; a su vez, el módulo de trabajo de título se enfoca en el ramo homónimo. Finalmente, el módulo de comisiones examinadoras maneja lo referente a la asignación y exportación de dichas comisiones.

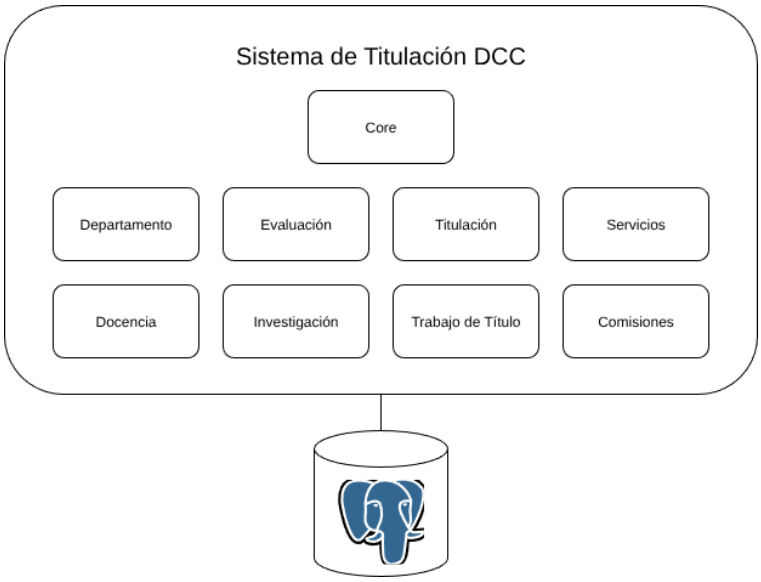


Figura 3.1: Arquitectura interna del sistema de titulación.

El sistema de titulación interactúa con otros sistemas, ya sea consumiendo sus APIs o exportando datos, como se puede ver en la figura 3.2. Específicamente, el sistema de titulación se relaciona con el Sistema de Seguimiento de Memorias (SSM), la API de UCampus llamada Mufasa y el Portal DCC.

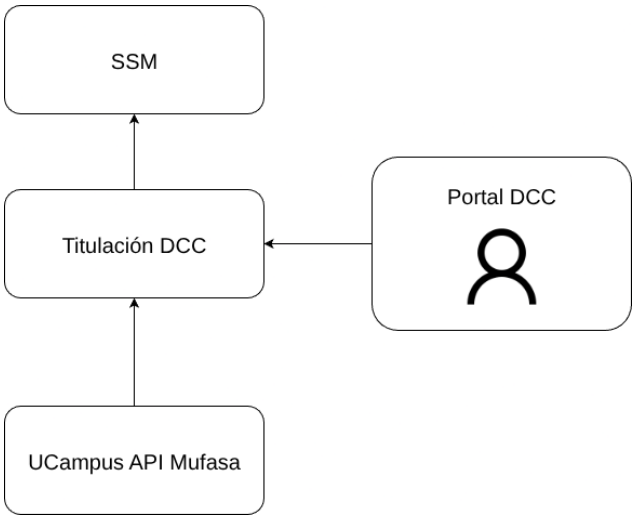


Figura 3.2: Arquitectura externa del sistema de titulación.

El sistema depende del Portal DCC para el inicio de sesión de los usuarios y la obtención de los datos de estos, como roles y permisos. Por otro lado, el sistema utiliza la API de UCampus para obtener los periodos académicos, los ramos relacionados con el proceso de titulación, sus respectivas secciones y estudiantes inscritos en estas secciones. Finalmente, el sistema exporta datos sobre las memorias y las comisiones examinadoras asignadas al Sistema

de Seguimiento de Memorias (SSM), mediante una petición POST autenticada con un token generado por el SSM.

3.2. Modelo de Datos

En esta sección se describirá el modelo de datos del sistema de titulación y cómo evolucionó durante este trabajo de título. Primero se describirá el modelo de datos en su estado inicial en la subsección 3.2.1, es decir, como estaba antes de realizar este trabajo de título. Luego, en la subsección 3.2.2 se describirán los cambios que se planificaron inicialmente. Por último, se describirán los cambios que surgieron durante el desarrollo de esta memoria.

3.2.1. Modelo de Datos Inicial

El repositorio del sistema de titulación tiene varios módulos y cada uno tiene sus propias entidades que pueden relacionarse con las de otros módulos. No todas las entidades son relevantes para la asignación de comisiones examinadoras, por lo que solo se describirán las entidades que se relacionan con el módulo de comisiones examinadoras y el módulo de titulación. Es importante mencionar que el modelo de datos que se describirá a continuación es el modelo en su estado previo a este trabajo de título.

El módulo principal es el módulo de titulación, que se puede ver en la figura 3.3. Este módulo contiene las entidades Tema, Solicitud, Propuesta y Documento. Estas representan el tema de un trabajo de título, una solicitud de un estudiante a un tema, un trabajo de título de un estudiante en el contexto del ramo de Introducción al Trabajo de Título y los documentos con los informes que deben entregar los estudiantes, respectivamente.

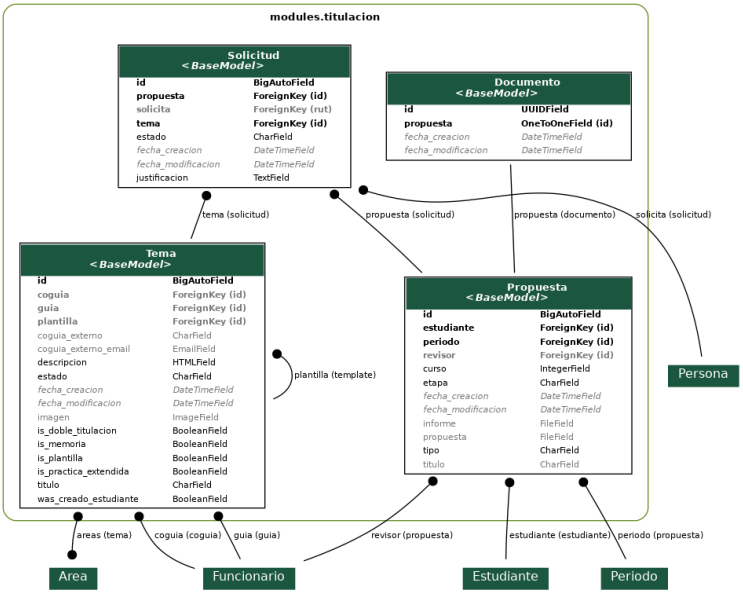


Figura 3.3: Diagrama del modelo de datos del módulo de titulación.

El módulo de comisiones examinadoras contiene a las entidades Comision y AlumnoCursandoMemoria, como muestra la figura 3.4. Comisión representa una comisión examinadora de un trabajo de título. Contiene una relación con la entidad Tema, que representa el tema del trabajo de título, y una relación con la entidad Solicitud, que representa la solicitud del tema. Además, tiene una relación de n a n con la entidad Evaluador del módulo de evaluación, que representa los integrantes de la comisión examinadora.

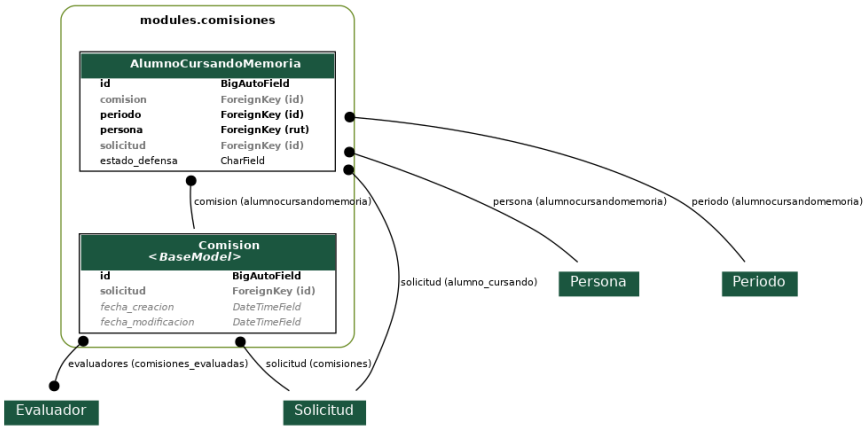


Figura 3.4: Diagrama del modelo de datos del módulo de comisiones examinadoras.

AlumnoCursandoMemoria representa a un estudiante que está cursando Trabajo de título. Contiene al estudiante a través de una relación con la entidad Persona del módulo kernel y se asocia a una memoria a través de la solicitud del tema. Además, contiene el periodo académico en el que el estudiante cursa el ramo Trabajo de título y tiene una relación con la entidad Comision, que representa la comisión examinadora que se le asigna al estudiante. Por último, tiene el estado de la defensa, que indica si la defensa ha sido aprobada, reprobada o si la defensa aún no ha sido realizada. Los posibles estados son *pendiente*, *aprobado* y *reprobado*.

El módulo de evaluación contiene a las entidades Evaluador, que corresponden a personas que evalúan, como los integrantes de una comisión examinadora. El módulo de departamento proporciona las entidades Funcionario y Estudiante. Funcionario representa a funcionarios del departamento y en el caso de titulación, representa a académicos del departamento que son guías o coguías de trabajos de título. Estudiantes representa a estudiantes del DCC. Las entidades de ambos módulos se pueden ver en la figura 3.5.

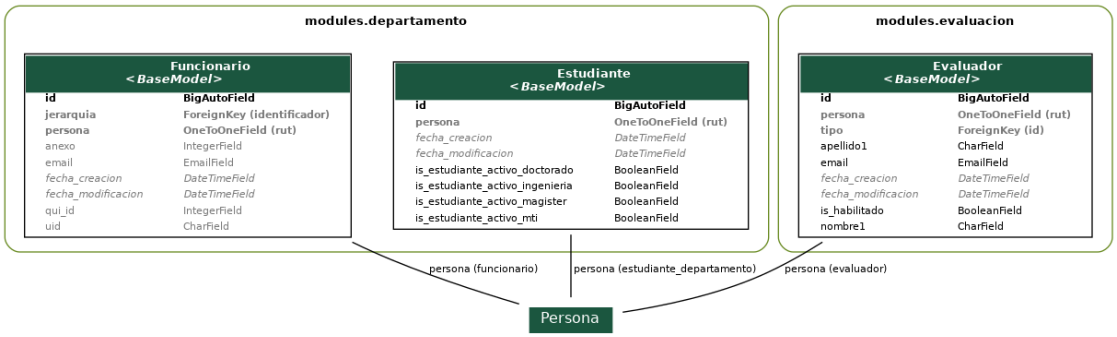


Figura 3.5: Diagrama del modelo de datos de los módulos departamento y evaluación.

El módulo de docencia proporciona los periodos académicos y el módulo de investigación proporciona Area, que representa áreas de conocimiento dentro de computación y permite asignar áreas de conocimiento tanto a los académicos como a los temas de trabajo de título. Por último, el módulo kernel proporciona la entidad Persona, que representa a cualquier usuario de la plataforma.

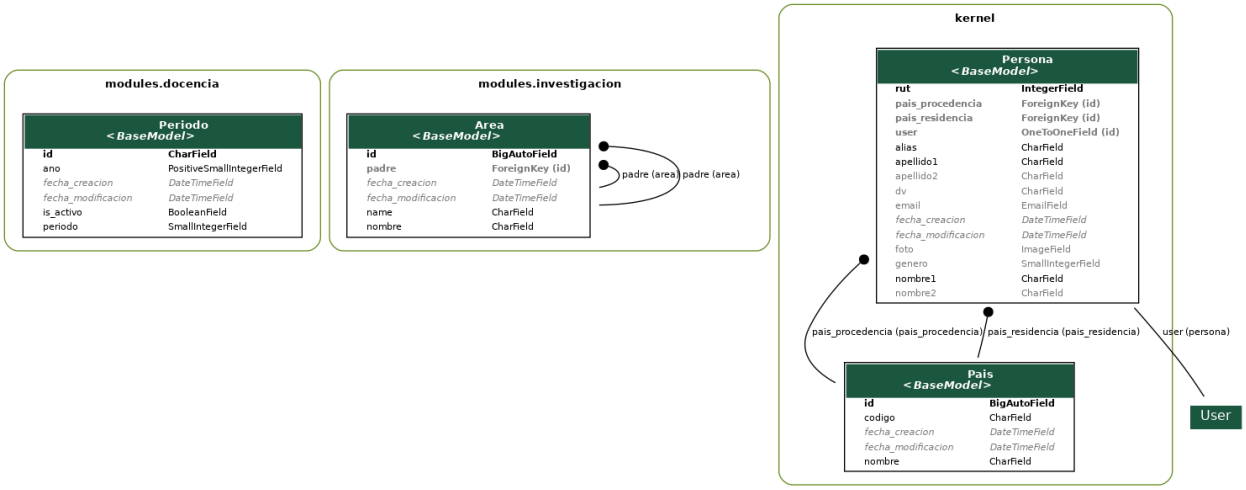


Figura 3.6: Diagrama del modelo de datos de los módulos docencia, investigación y kernel.

3.2.2. Cambios Iniciales al Modelo de Datos

Tomando en cuenta el modelo de datos actual y las funcionalidades que se desea agregar a la herramienta, el modelo de datos no sufrirá muchos cambios, ya que la mayoría de las funcionalidades trabajan con datos que ya se encuentran en el modelo. La única funcionalidad que requiere un cambio es la de publicar las comisiones examinadoras, puesto que se necesita diferenciar entre comisiones que han sido publicadas y aquellas que no lo han sido. Para lograr esto, se agregará la entidad ComisionBorrador que tendrá los mismos atributos que Comision, pero que se diferenciará por el hecho de que no ha sido publicada. De esta forma, la publicación de comisiones examinadoras por API se realizará con los datos de la entidad Comision.

3.2.3. Cambios al Modelo de Datos Durante el Desarrollo

A continuación se detallará por cada módulo las modificaciones realizadas al modelo de datos que no fueron planificadas inicialmente, y la razón por la que se realizaron.

En primer lugar, en el módulo de titulación se agregó el atributo tema a la entidad Propuesta, que corresponde al tema de la memoria. Este se agregó, puesto que anteriormente solo se podía acceder al tema de una memoria a través de la entidad Solicitud. Esto no era práctico y tampoco hace sentido, ya que Propuesta representa una memoria en el ramo de Introducción al Trabajo de Título y toda memoria debe tener un tema. Esto se puede ver en la figura 3.7.

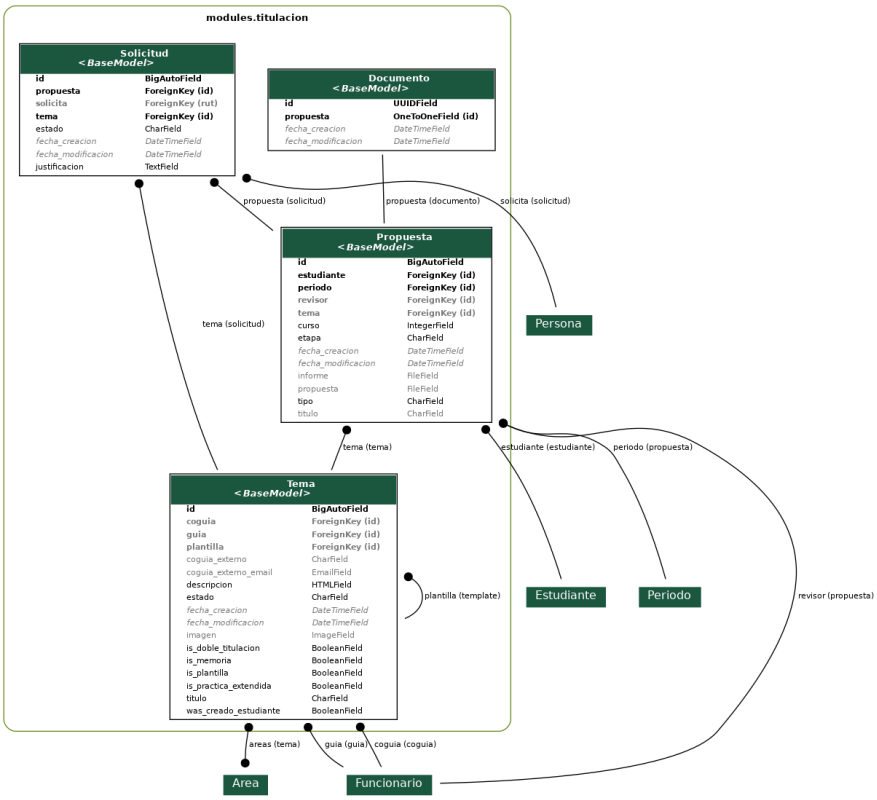


Figura 3.7: Diagrama del modelo de datos del módulo de titulación.

Luego, en el módulo de docencia, se agregó la entidad Curso, que representa una sección de un ramo en un periodo académico específico, lo cual se ve reflejado en la figura 3.8. Esta entidad tiene los atributos periodo, número de sección, código del ramo, nombre del ramo, id del ramo e id del curso. Estos dos últimos corresponden a identificadores dentro de la API de UCampus. Se agregó esta entidad para identificar el curso al que pertenece una memoria y así poder exportar las comisiones examinadoras asignadas al SSM que separa las memorias por curso.

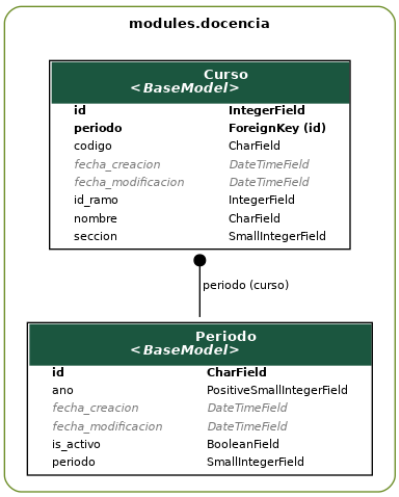


Figura 3.8: Diagrama del modelo de datos del módulo de docencia.

En el módulo de comisiones examinadoras se eliminó el atributo solicitud de la entidad Comision, puesto que no es necesario que desde una comisión se acceda a la solicitud al tema de la memoria, ya que la solicitud solo es relevante al comienzo del ramo Introducción al Trabajo de Título y las comisiones examinadoras aparecen recién en el ramo Trabajo de Título. También se eliminó la entidad ComisionBorrador, puesto que no es necesario tener una entidad para comisiones examinadoras que no han sido publicadas, en su lugar, se agregó el atributo booleano publicada a la entidad Comision, que indica si la comisión ha sido exportada al SSM. Además, se eliminó la entidad AlumnoCursandoMemoria, ya que la entidad representaba a un estudiante que cursaba una memoria en el ramo Trabajo de Título, lo cual no era consistente con el modelo de datos, pues ya existe la entidad Estudiante y su equivalente en el ramo de Introducción al Trabajo de Título, es decir Propuesta, representa a la memoria en el ramo de Introducción al Trabajo de Título y no al estudiante que cursaba la memoria. La entidad fue reemplazada por la entidad MemoriaEnF, que será descrita a continuación.

Por último, se creó el módulo de Trabajo de Título que contiene la entidad MemoriaEnF. Esta entidad representa una memoria en el ramo de Trabajo de Título y tiene como atributos al estudiante que realiza la memoria, la comisión examinadora, el tema de la memoria, el periodo académico y curso en el que se realiza la memoria, el tipo de memoria, el estado de defensa y una referencia a la entidad Propuesta, que corresponde a la misma memoria, pero en el ramo de Introducción al Trabajo de Título. Tanto la entidad MemoriaEnF como la entidad Comision se pueden ver en la figura 3.9.

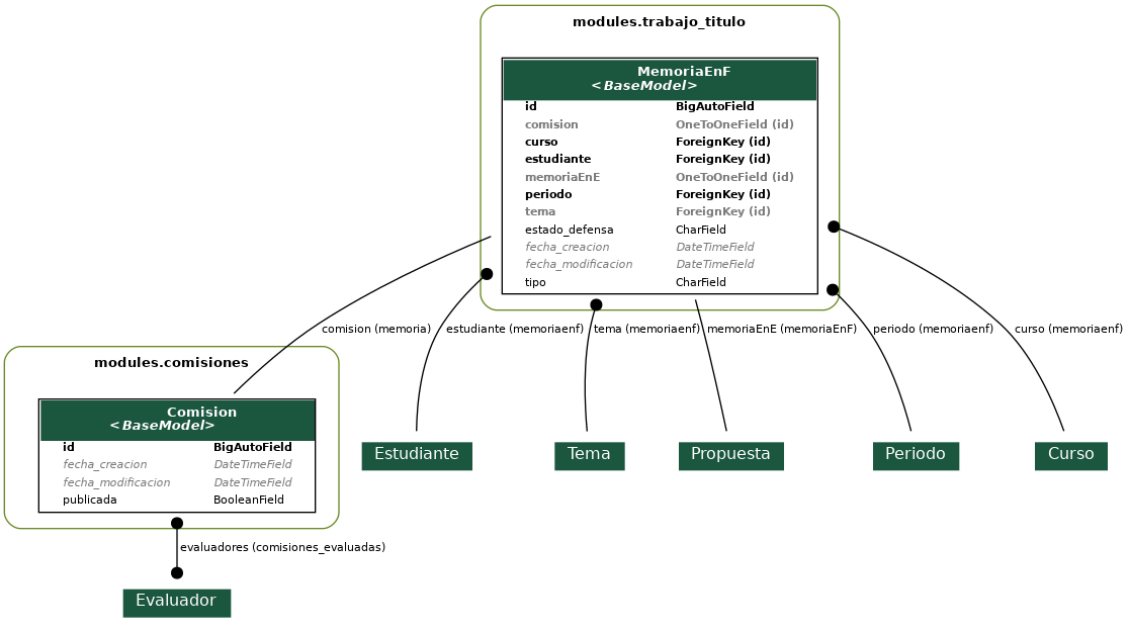


Figura 3.9: Diagrama del modelo de datos de los módulos de comisiones examinadoras y trabajo de título.

3.3. Diseño de Mockups

Es importante recalcar que la herramienta es una extensión del trabajo realizado por un equipo de Ingeniería de Software II y del sistema de titulación, por lo tanto, en la mayoría de los mockups se utilizan las interfaces ya existentes como base.

3.3.1. Interfaz principal del módulo de comisiones examinadoras

El primer diseño en ser realizado fue el de la interfaz principal del módulo de asignación de comisiones examinadoras. La interfaz original se puede ver en la figura 2.6 y mientras que la propuesta se puede ver en la figura 3.10.

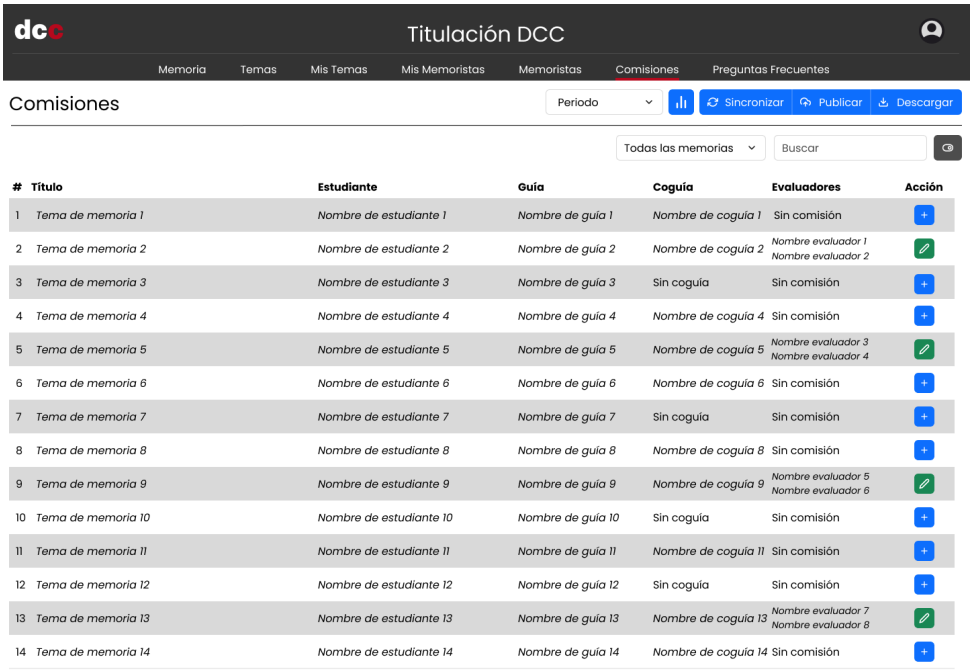


Figura 3.10: Mockup de la interfaz principal del módulo de comisiones examinadoras.

Ambas interfaces son similares, ya que la tabla mantiene la misma estructura. Comenzando por la parte superior, se mantiene el selector de periodo académico y el botón de *Sincronizar*. El botón *Descargar* es equivalente al botón *Exportar* de la interfaz original, solo cambia el texto y el ícono. Se agregó un botón con ícono de gráfico que permite mostrar y esconder un gráfico con la cantidad de comisiones examinadoras y memorias guíadas por cada profesor en el periodo académico seleccionado, que se muestra en la figura 3.11, las columnas están ordenadas de forma descendiente por la cantidad de carga de cada profesor. Como se trata de varios profesores, se decidió que el gráfico no tenga etiquetas en el eje X, pues no serían visibles y saturaría el gráfico. En su lugar, se muestra el nombre del profesor y la cantidad de comisiones examinadoras y memorias guíadas al pasar el mouse por encima de una barra del gráfico.

También se agregó el botón *Publicar* que permite publicar las comisiones examinadoras

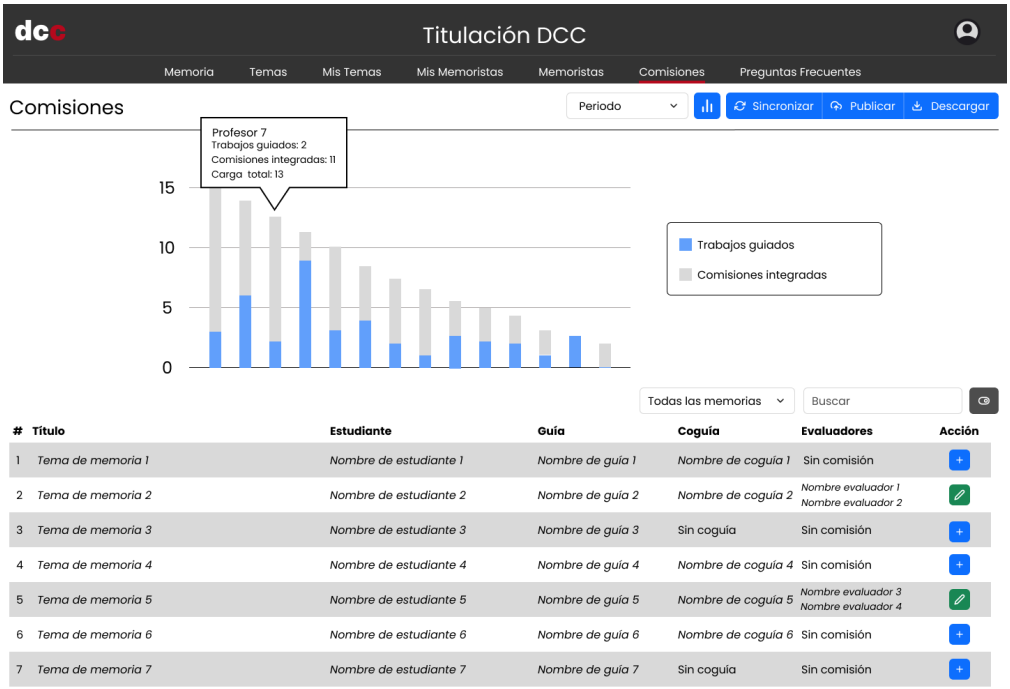


Figura 3.11: Mockup del gráfico de comisiones examinadoras.

en el sistema de titulación, permitiendo que las comisiones asignadas se muestren en la ficha del estudiante y estén disponibles para exportar. Al presionar *Publicar*, aparece un modal con un mensaje de confirmación, indicando si es que quedan memorias sin comisión asignada y cuántas son, en caso de que haya. Esto se puede ver en la figura 3.12.

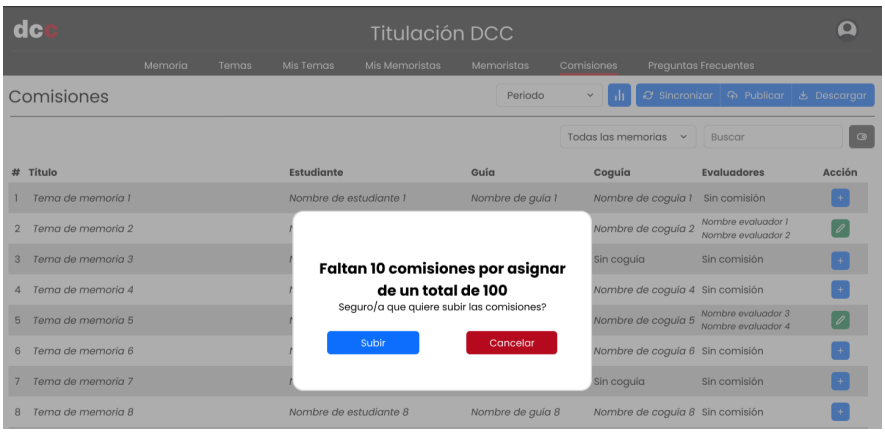


Figura 3.12: Mockup de la confirmación de la publicación de comisiones examinadoras.

Más abajo, al costado del buscador, se agregó un filtro que permite ajustar si en la tabla se muestran todas las memorias, solo las memorias con comisión incompleta o solo las memorias con comisión completa. Este filtro se puede ver en la figura 3.13.

Por último, en la tabla se decidió que en los botones de acción, es decir, *Agregar* y *Editar*, se mostrarán solo los íconos de + y lápiz, respectivamente, porque se consideró que sin el texto se mantendría la claridad de la acción que realizan y se disminuiría la densidad de texto

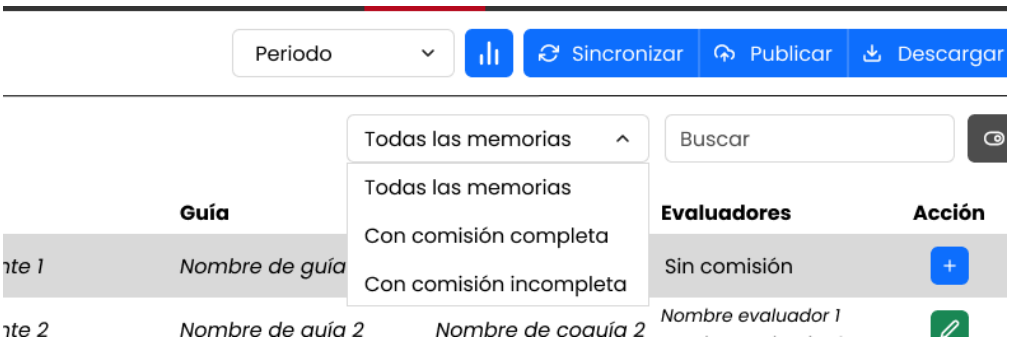


Figura 3.13: Mockup del filtro de la interfaz principal del módulo de comisiones examinadoras.

en la tabla.

3.3.2. Ficha de un estudiante con comisión asignada

Después, se diseñó la interfaz de la ficha de un estudiante desde la vista de un estudiante con la comisión examinadora que le ha sido asignada, que se puede ver en la figura 3.14. No se cambió nada de la interfaz original, solo se agregó la sección de la comisión examinadora. Esta sección se ubica bajo la sección de Tema en una tabla con el mismo formato visual que las otras tablas de la ficha del estudiante. Cada fila corresponde a un integrante de la comisión examinadora, indicando el rol dentro de la comisión, una imagen del integrante, su nombre y correo electrónico.

Figura 3.14: Mockup de la ficha de un estudiante desde la vista de un estudiante con la comisión examinadora asignada.

Capítulo 4

Implementación

En este capítulo se describirá la implementación realizada durante la memoria. Esta se separa puede separar en tres partes, la primera es la implementación relacionada con los estudiantes del ramo Trabajo de Título, también llamado F. La segunda parte corresponde a la asignación de comisiones examinadoras. La tercera parte corresponde a la exportación de comisiones examinadoras.

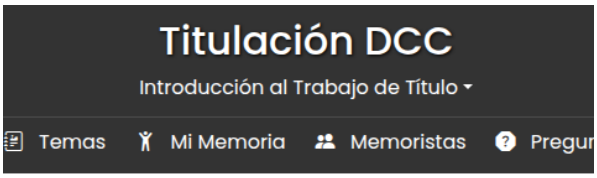
4.1. Estudiantes de Trabajo de Título

4.1.1. Módulo de Trabajo de Título

Para implementar funcionalidades relacionadas con el ramo Trabajo de Título y separarlas de las funcionalidades del ramo Introducción al Trabajo de Título, se creó el módulo de Trabajo de Título. Esta separación también evita que la barra de navegación se llene de pestañas. Para acceder al módulo, se creó un selector en la barra de navegación, que se ubica en la parte superior de la página, como se puede ver en la figura 4.1. Como por el momento las únicas funcionalidades asociadas al módulo de Trabajo de Título son ver los estudiantes de Trabajo de Título y la asignación de comisiones examinadoras, y ambas requieren permisos de coordinación de titulación, el selector solo es visible para coordinadores, que está manejado por la variable booleana `show_module_selector`, como se muestra en la línea 11 del código 4.1. Al seleccionar el módulo Introducción al Trabajo de Título, el usuario es redirigido a la raíz de la aplicación. Mientras que al seleccionar el módulo Trabajo de Título, el usuario es redirigido a la vista principal de Trabajo de Título, que es el listado de estudiantes de Trabajo de Título.

Como anteriormente todo el sistema estaba dedicado al ramo Introducción al Trabajo de Título, el título de la barra de navegación era el nombre del módulo. Por lo tanto, se creó el módulo Trabajo de Título para poder separarlo del módulo de Introducción al Trabajo de Título. Además, el título de la barra de navegación se cambió a un título configurable mediante la variable `titulo` y si esta variable no se especifica, se muestra el nombre del módulo. Se decidió dejar este caso, ya que el header es el mismo para todos los módulos y el

resto de módulos no requieren un título personalizado.



(a) Captura de pantalla del selector cerrado en la barra de navegación en el módulo de Introducción al Trabajo de Título.

(b) Captura de pantalla del selector abierto en la barra de navegación en el módulo de Introducción al Trabajo de Título.

(c) Captura de pantalla del selector cerrado en la barra de navegación en el módulo de Trabajo de Título.

(d) Captura de pantalla del selector abierto en la barra de navegación en el módulo de Trabajo de Título.

Figura 4.1: Capturas de pantalla del selector en la barra de navegación.

Listing 4.1: Código de la parte central de la barra de navegación

```
<div class="header">
  <a href="{% url 'servicios:index' %}" class="logo"><span class="dcc">dc<span>c</span></span></a>
  <div class="d-flex flex-column flex-grow-1 align-items-center pt-2">
    <h1 class="modulo mb-0">
      {% if titulo %}
        {{ titulo }}
      {% else %}
        {{ modulo.nombre }}
      {% endif %}
    </h1>
    {% if show_module_selector %}
      <div class="dropdown-center">
        <button class="btn btn-dark dropdown-toggle" type="button" data-bs-toggle="dropdown">
          {{ modulo.nombre }}
        </button>
        <ul class="dropdown-menu">
          {% for mod in modules %}
            <li>
              <a class="dropdown-item {% if mod.nombre == modulo.nombre %} disabled {% endif %}"
                 href="{% if mod.nombre == modulo.nombre %}#{% else %}{{ mod.url }}{% endif %}">
                {{ mod.nombre }}
              </a>
            </li>
          {% endfor %}
        </ul>
      </div>
    {% endif %}
  </div>
</div>
```

4.1.2. Listado de estudiantes de Trabajo de Título

La asignación de comisiones examinadoras a los estudiantes de Trabajo de Título, requiere tener un registro de los estudiantes que están cursando Trabajo de Título junto con sus respectivos temas de memoria. Por lo tanto, dentro del módulo de Trabajo de Título se creó

una vista que lista los estudiantes que están cursando Trabajo de Título, que se puede ver en la figura 4.2.

En esta vista se muestra una tabla que por cada estudiante muestra su nombre, el título de su memoria, su guía, su coguía si tiene y un botón que redirige a su ficha. Sobre la tabla al costado derecho se encuentra una barra de búsqueda que permite buscar por nombre de estudiante, título de memoria, nombre de guía y nombre de coguía. Sobre esta barra, se encuentra un filtro que permite elegir el semestre académico de los estudiantes que se mostrarán en la tabla. Por defecto se mostrarán los estudiantes del semestre académico activo. Por último, al costado derecho de este filtro está el botón de *Sincronizar*, que permite actualizar la tabla con los estudiantes.

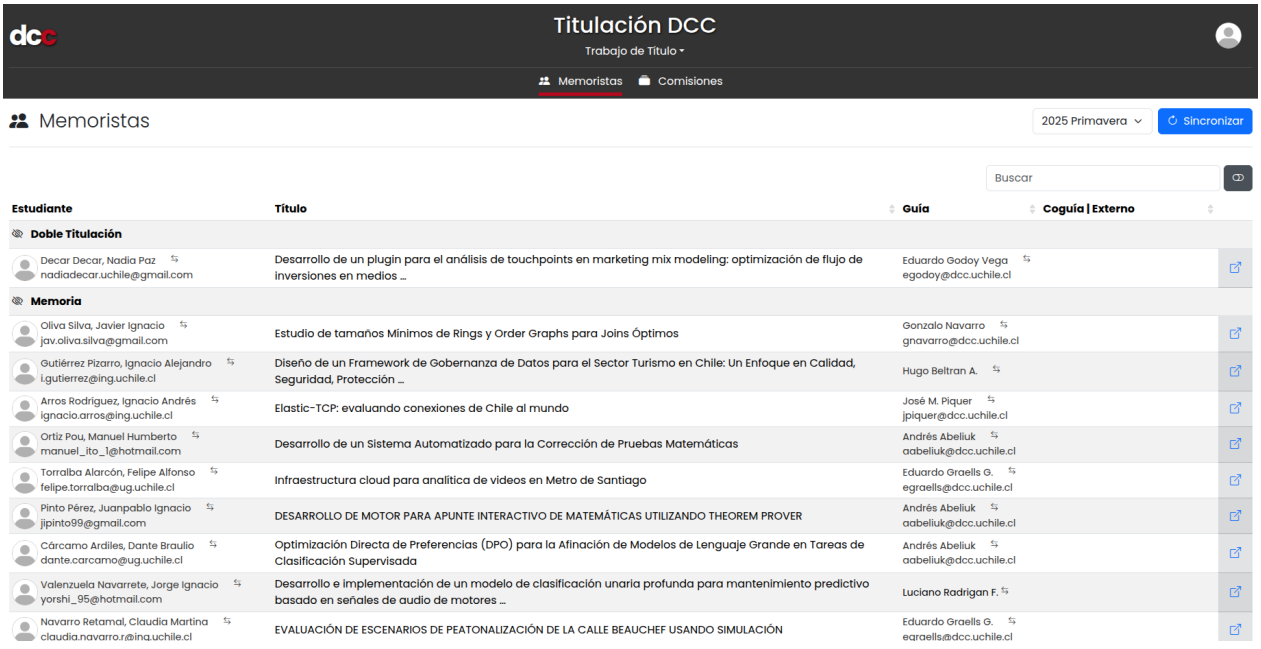


Figura 4.2: Captura de pantalla de la vista del listado de estudiantes de Trabajo de Título.

Para la actualización del registro se tomó en cuenta que los estudiantes deben estar cursando alguna sección de los ramos de Trabajo de Título, CC6909 y CC6919, por lo tanto se puede obtener los nombres de los estudiantes desde la API de UCampus. Además, los estudiantes deben haber aprobado Introducción al Trabajo de Título, por lo que se asume que los temas de los estudiantes ya se encuentran registrados en el sistema de titulación y que también hay una instancia del modelo Propuesta para cada estudiante asociada a su tema y con estado aprobado, haciendo posible asociar a los estudiantes que cursan Trabajo de Título con sus temas.

Dada la asunción anterior, para actualizar el registro de estudiantes de Trabajo de Título dentro del sistema de titulación se creó un comando de administración `upd_estudiantes_f`, que es llamado por el botón *Sincronizar*. Este obtiene desde la API de UCampus todas las secciones de los ramos de Trabajo de Título y para después obtener los integrantes de cada sección. Luego, por cada integrante se busca la instancia del modelo Propuesta con estado aprobado más reciente asociada al estudiante, pues puede haber varias instancias asociadas al mismo estudiante. Si se encuentra una instancia, se obtiene el tema asociado y se crea una instancia del modelo MemoriaEnF con el estudiante y su tema. En caso contrario, se registra

en el log que para ese estudiante no se encontró una instancia, se agrega al estudiante a la lista de estudiantes sin una instancia y se continua con el siguiente integrante. Al finalizar el proceso se muestra un mensaje indicando cuántos estudiantes no tuvieron una instancia, como se ve en la figura 4.3. Una vez que termina la ejecución del comando, se recarga la página y se muestra la lista actualizada de estudiantes de Trabajo de Título.

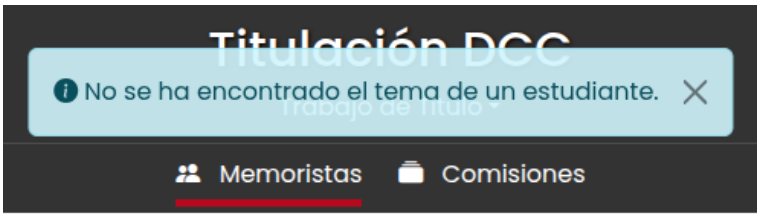


Figura 4.3: Captura de pantalla de la notificación de que un estudiante no tiene un tema registrado.

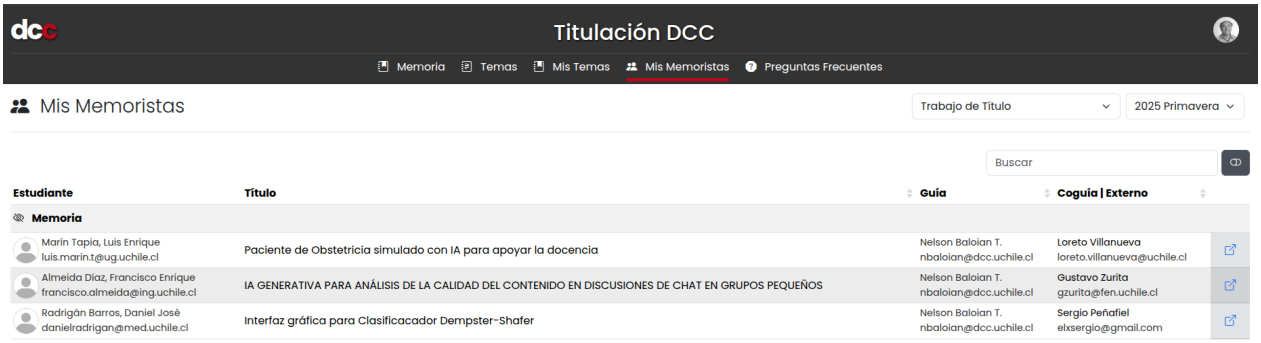
Como se mencionó en la descripción del comando, puede suceder que haya estudiantes que no tengan una instancia de Tema y Propuesta con estado Aprobado. Esto sucede por lo que la asunción puede no ser cierta para estudiantes que hayan solicitado la vía rápida, pues actualmente ese proceso no se encuentra implementado en el sistema de titulación y se realiza directamente con la coordinación de titulación. El caso de la vía rápida no se manejará en esta memoria, ya que al momento del desarrollo de este trabajo de título, había otro tema de memoría sobre la incorporación de un módulo de vía rápida. Mientras que tanto, basta con crear una instancia de Tema y Propuesta con estado Aprobado para cada estudiante de vía rápida mediante el administrador de Django o por consola para que puedan ser incluidos en la lista de estudiantes de Trabajo de Título.

4.1.3. Vista de *Mis Memoristas* para académicos

Habiendo incorporado a los estudiantes de Trabajo de Título al sistema de titulación, se extendió la vista de *Mis Memoristas* que es visible para académicos, para que pudieran ver una lista de sus memoristas en Trabajo de Título. Para ello se agregó un selector de ramo de titulación que permite elegir entre Introducción al Trabajo de Título y Trabajo de Título. Este se encuentra bajo la barra de navegación de la vista, a la izquierda del filtro de periodo académico. En la figura 4.4a se muestra el selector en la vista con memoristas de Introducción al Trabajo de Título, mientras que en la figura 4.4b se muestra el selector en la vista con memoristas de Trabajo de Título.

Al seleccionar Trabajo de Título, se muestra la lista de los estudiantes de Trabajo de Título que son memoristas del usuario. Por cada estudiante se muestra su nombre, su título de memoria, su guía, su coguía si tiene y un botón a la ficha del estudiante. El resto de la vista se mantiene igual que la vista para Introducción al Trabajo de Título.

(a) Captura de pantalla de la vista de *Mis Memoristas* para Introducción al Trabajo de Título.



(b) Captura de pantalla de la vista de *Mis Memoristas* para Trabajo de Título.

Figura 4.4: Vista de *Mis Memoristas* para académicos con selector de ramo de titulación.

4.2. Asignación de Comisiones

En esta sección se describe la implementación de funcionalidades relacionadas con la asignación de comisiones examinadoras. Primero se mencionarán las funcionalidades que se implementaron para la evaluación de la herramienta, y luego se describirán las modificaciones que se surgieron a partir de la retroalimentación recibida en la evaluación.

El sistema piloto desarrollado por el equipo de ingeniería de software II ya contaba con la funcionalidad de asignación de comisiones examinadoras, tal como se menciona en la sección 2.4. Sin embargo, esta funcionaba con el modelo AlumnoCursandoMemoria, por lo que se adaptó para que funcione con el nuevo modelo MemoriaEnF, manteniendo la tabla de comisiones examinadoras que muestra una lista de memorias con sus respectivos estudiantes, temas, guías, coguías e integrantes de la comisión. Al final de cada fila se encuentra un botón que permite asignar evaluadores a la comisión. La figura 4.5 muestra la lista de memorias con sus comisiones examinadoras .

4.2.1. Filtro de Comisiones Examinadoras

Una vez adaptada la implementación al nuevo modelo, se le agregaron funcionalidades. En primer lugar, se agregó un filtro de comisiones examinadoras en la interfaz principal del módulo de comisiones examinadoras. Este filtro permite configurar los trabajos de título que se muestran dependiendo de la cantidad de evaluadores asignados a la comisión examinadora. Si se selecciona “Todas las memorias”, se muestran todos los trabajos de título, si se selec-

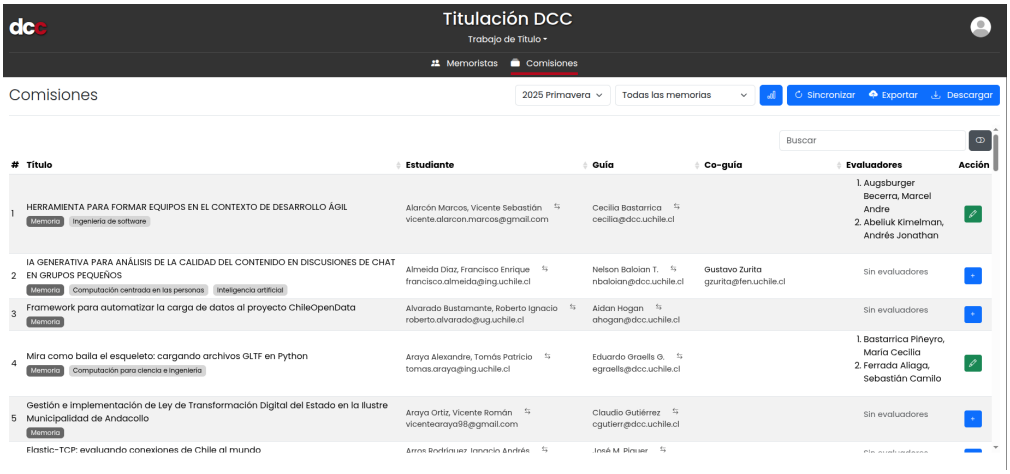


Figura 4.5: Captura de pantalla de la lista de memorias con sus comisiones examinadoras.

ciona “Con comisión completa”, se muestran solo aquellas que tengan dos o más evaluadores asignados, y si se selecciona “Con comisión incompleta”, se muestran aquellas que tengan un evaluador o ninguno. En la figura 4.6 se muestra el filtro configurado para mostrar solo las memorias con comisión completa. Internamente, el filtro refresca la página y agrega un query parameter a la URL que indica el filtro seleccionado. Desde el backend se filtran las comisiones según el query parameter, como se muestra en el código 4.2.

Figura 4.6: Captura de pantalla de la lista de comisiones examinadoras filtrada por comisiones completas.

Listing 4.2: Filtrado de comisiones examinadoras.

```
memorias = MemoriaEnF.objects.filter(periodo=periodo).select_related(
    "tema__guia__persona",
    "tema__coguia__persona",
    "estudiante__persona",
).prefetch_related(
    "comision__evaluadores"
).annotate(
    Count("comision__evaluadores")
)
# Filtrado por comision completa, incompleta o todas
if comision_filter == "completas":
    memorias_filtradas = memorias.filter(comision__evaluadores__count__gte=2)
elif comision_filter == "incompletas":
    memorias_filtradas = memorias.filter(comision__evaluadores__count__lt=2)
else:
    memorias_filtradas = memorias

# Ordenar: primero con tema (True > False)
memorias_filtradas = sorted(
    memorias_filtradas,
    key= lambda memoria: (
        memoria.tema is None, # False (tienen tema) va antes
        memoria.estudiante.persona.apellido1.lower() if memoria.estudiante.persona.apellido1 else "",
        memoria.estudiante.persona.nombre1.lower() if memoria.estudiante.persona.nombre1 else "",
    )
)
context["memorias"] = memorias_filtradas
```

4.2.2. Gráfico de Carga de Evaluadores

El mayor cambio visual en la interfaz del módulo de comisiones proviene de la incorporación del gráfico que muestra la carga de evaluadores. Este se ubica sobre la lista de comisiones examinadoras, como se muestra en la figura 4.7. Cada columna representa la carga de un evaluador, que está dividida entre la carga proveniente de las memorias guíadas o coguiadas y la carga proveniente de las comisiones examinadoras que integra. Es importante destacar que integrar una comisión examinadora equivale a un punto de carga, mientras que guiar o coguir una memoria equivale a dos puntos de carga. Esta ponderación fue recomendada por la coordinadora de titulación. Al posicionar el cursor sobre una columna, se muestra un tooltip que indica la cantidad de comisiones y la cantidad de memorias.

El gráfico está ordenado de mayor a menor función de la carga total de cada evaluador, es decir, por la siguiente fórmula: $carga\_total = 2 * memorias\_guiadas + comisiones\_integradas$. Este orden ayuda a que sea fácil identificar a los evaluadores con mayor y menor carga. Además, el gráfico se actualiza cada vez que se asignan integrantes a una comisión examinadora y no se ve afectado por el filtro de comisiones. Por último, el gráfico puede ocultarse y mostrarse mediante un botón que se encuentra bajo la barra de navegación, entre el filtro de comisiones y el botón de sincronización.

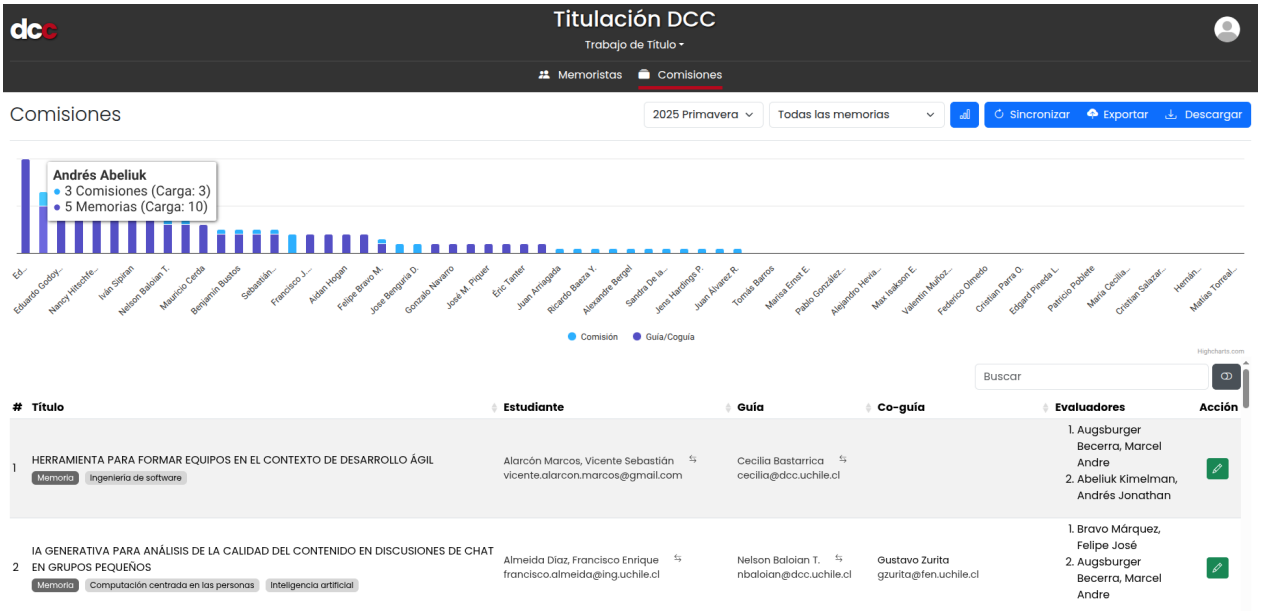


Figura 4.7: Captura de pantalla del gráfico de carga de evaluadores.

4.2.3. Modal de asignación de comisiones

Al presionar el botón para asignar integrantes a una comisión examinadora, se abre un modal para realizar la asignación. Este modal también se encontraba en la versión piloto del sistema, pero al igual que la tabla de comisiones examinadoras, se adaptó para que use el modelo MemoriaEnF en vez de AlumnoCursandoMemoria. Además, en el modal se agregó el nombre del estudiante. En la figura 4.8 se muestra el modal con los cambios realizados.

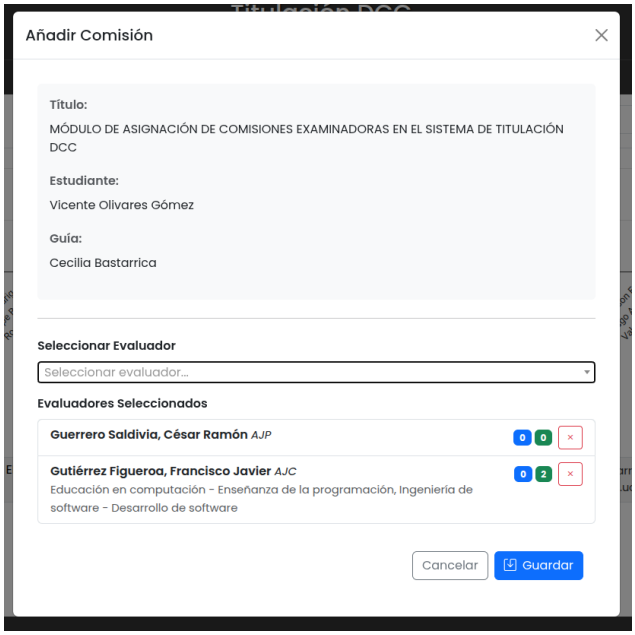
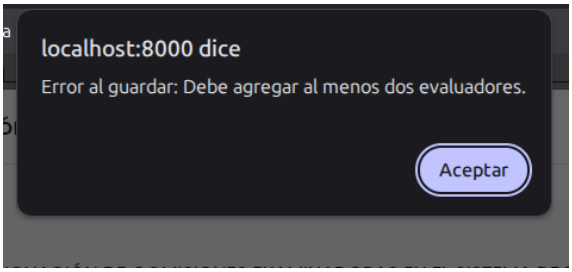


Figura 4.8: Captura de pantalla del modal de asignación de comisiones.

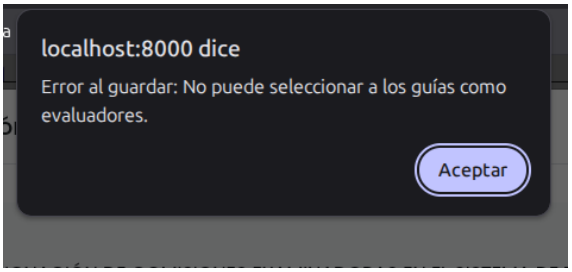
Un cambio importante es que se agregaron validaciones en la asignación de comisiones. En particular, se valida que al guardar una comisión examinadora, esta tenga al menos dos integrantes, que ninguno de los integrantes sea guía o coguía de la memoria y que al menos uno de los integrantes tenga jerarquía de Académico de Jornada Completa (AJC) o Académico de Jornada Parcial (AJP). Si se intenta guardar una comisión examinadora que no cumpla con estas validaciones, se muestra un mensaje de error. En la figura En el código 4.3 se muestra la implementación de las validaciones, mientras que en la figura 4.9 se muestran los mensajes de error por validación.

Listing 4.3: Validación de comisión examinadora.

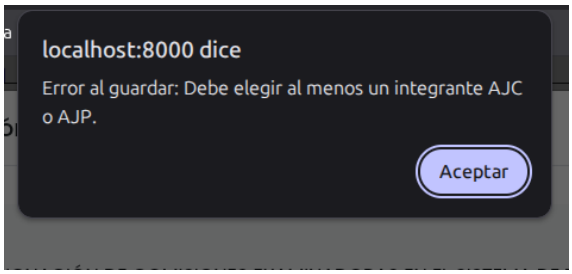
```
# Validación de comisión
if len(evaluadores) < 2:
    return JsonResponse({"success": False, "message": "Debe agregar al menos dos evaluadores."})

rut_guia = comision.get_guia.persona.rut if comision.memoria.tema.guia else None
rut_coguia = comision.get_coguia.persona.rut if comision.memoria.tema.coguia else None

has_non_pex = False
for evaluador in evaluadores:
  if evaluador.persona.rut in (rut_guia, rut_coguia):
    return JsonResponse({"success": False, "message": "No puede seleccionar a los guías como evaluadores."
        })

  if evaluador.tipo.nombre != "PEX":
    has_non_pex = True

if not has_non_pex:
  return JsonResponse({"success": False, "message": "Debe elegir al menos un integrante AJC o AJP."})

comision.evaluadores.add(*evaluadores)
```



(a) Mensaje de error indicando que se debe agregar al menos dos evaluadores.



(b) Mensaje de error indicando que un integrante de la comisión es guía o coguía.



(c) Mensaje de error indicando que al menos un integrante debe ser ajc o ajp.

Figura 4.9: Mensajes de error por validaciones de comisión examinadora.

4.2.4. Cambios por Retroalimentación

Luego de realizar la evaluación de la herramienta (capítulo 5), se recibió una retroalimentación que indicaba que se debían hacer ciertos cambios en el sistema. El punto de mayor importancia en la retroalimentación era que al presionar el botón para asignar la comisión examinadora de una memoria demoraba mucho tiempo en aparecer el modal, pues podía demorar hasta 15 segundos. Esta cantidad de tiempo es inaceptable para la experiencia del usuario, en especial porque es un botón que se debe presionar por cada memoria. Por lo tanto, se revisó el código del modal para ver posibles problemas y optimizaciones.

El problema se encontraba en el controlador del modal, que es llamado cada vez que se presiona el botón para abrir el modal y responde con el HTML del modal. Que se llame cada vez no es un problema, pero el trabajo que realizaba el controlador en cada llamada era innecesariamente excesivo, ya que por cada llamada se obtenían todos los evaluadores habílitados y a cada uno se le generaba un string con sus áreas de conocimiento con sus respectivas subareas. Este proceso se realizaba para poblar el selector de evaluadores en el modal. Para optimizar el proceso de poblar el selector de evaluadores, se tomaron dos medidas: Precalcular el string con las áreas de conocimiento y obtener una única vez los evaluadores.

Para realizar la primera medida, se agregó el atributo `areas_string` al modelo Evaluador, que almacena el string con las áreas de conocimiento de un evaluador. Además, se agregó un método para calcular el string con las áreas de conocimiento de un evaluador y se añadió una señal que se llama al método cada vez que haya un cambio en las áreas del evaluador.

Esto permite que el string solo se calcule cuando es necesario y no cada vez que se llama el controlador del modal. En los códigos 4.4 y 4.5 se muestra el código de la señal y el método respectivamente.

Listing 4.4: Código de la señal que actualiza el string con las áreas de conocimiento de un evaluador

```python
@receiver(m2m_changed, sender=Evaluador.areas.through)
def update_areas_string(sender, instance, action, **kwargs):
    """
    Actualiza el campo areas_string de un evaluador cada vez que cambian sus áreas.
    """
    if action in {"post_add", "post_remove", "post_clear"}:
        instance.generate_areas_string()
```

Listing 4.5: Código del método para generar el string con las áreas de conocimiento de un evaluador

```python
    def generate_areas_string(self):
        """
        Genera un string con las áreas del evaluador y lo guarda en el atributo areas_string.
        El string tiene el siguiente formato: Padre - Hijo - Hijo - Hijo, Padre, Padre - Hijo
        """
        areas_by_parent = {}
        for area in self.areas.all():
            # Obtenemos el área padre (puede ser None)
            parent = area.padre
            parent_name = parent.nombre if parent else None
            child_name = area.nombre

            if parent_name:
                areas_by_parent.setdefault(parent_name, set()).add(child_name)
            else:
                if child_name not in areas_by_parent:
                    areas_by_parent[child_name] = set()

            # Construimos el string final
        result = []
        for parent, children in sorted(areas_by_parent.items()):
            if children:
                hijos_ordenados = sorted(children)
                result.append(f"{parent} - " + " - ".join(hijos_ordenados))
            else:
                result.append(parent)

        # Unimos todas las areas padre separadas por comas
        # Padre - Hijo - Hijo - Hijo, Padre, Padre - Hijo
        self.areas_string = ", ".join(result)
        self.save()
```

Para la segunda medida, se dejó de obtener a los evaluadores en el controlador del modal y pasaron a obtenerse una vez al cargar la página del módulo de comisiones. Esto no significa un mayor costo, ya que los evaluadores ya eran obtenidos al cargar la página, dado que el gráfico de carga de evaluadores ya los obtenía. Estos evaluadores se almacenan en un arreglo de objeto, almacenando su id, nombre, áreas de conocimiento, memorias guíadas, comisiones integradas y jerarquía, para que puedan ser utilizados en el modal y el gráfico..

Un segundo punto a mejorar indicado en la retroalimentación fue al guardar una comisión, se refrescaba la página, lo cual hacía que el usuario tuviera que bajar nuevamente hasta la memoria que estaba viendo, lo cual es trabajo innecesario. Para resolver este problema, se evitó refrescar la página y que al guardar una comisión el listado de integrantes se actualice mediante JavaScript, se actualice el contenido del arreglo que se utiliza para poblar el listado de integrantes y se actualice el gráfico de carga de integrantes.

4.3. Exportación de Comisiones

Para exportar las comisiones examinadoras, el sistema ofrece dos opciones: descargar un archivo CSV y enviarlas a través de una petición POST la API del SSM.

4.3.1. Archivo CSV

El piloto desarrollado por el equipo de ingeniería de software II ya contaba con la funcionalidad de exportar las comisiones examinadoras a un archivo CSV. Sin embargo, el formato del archivo no seguía el formato requerido por el SSM, como se mencionó en las secciones 2.2 y 2.4. Por lo tanto, se cambiaron las columnas del archivo para que cumplan con los requisitos del SSM, se cambió el separador al carácter coma y se agregaron las columnas que faltaban. Por último, se agregó la condición de que solo las memorias con comisión completa sean exportadas

4.3.2. Exportación directa al SSM

Se implementó una exportación directa al SSM a través de una petición POST al SSM. Esta implementación consta de dos partes. La primera es hacer que el SSM sea capaz de recibir una petición desde otro sistema y procesarla. Mientras que la segunda es que el sistema de titulación envíe una petición POST al SSM con los datos de las comisiones examinadoras. Esta funcionalidad fue implementada luego de la evaluación por falta de tiempo.

Adaptación del SSM

Para adaptar el SSM a recibir una petición POST desde otro sistema, se creó una view de Django en el endpoint `/api/comisiones/` que recibe la petición POST con un JSON en el cuerpo de la petición. Debido a que en el SSM las memorias se separan por curso, es decir, por ramo y por sección, se decidió que en el JSON recibido, las comisiones deben estar separadas por cursos. Además, es necesario que este contenga la información necesaria para crear los cursos en la base de datos, en caso de que alguno no exista. Esta corresponde al código del ramo, el número de la sección, el año y el semestre, siendo "O" para el semestre de otoño y "P" para el semestre de primavera, ya que así lo tiene definido el SSM. Por último, por cada comisión se debe tener el tema de la memoria,nombre y correo del estudiante, del guía y del coguía, además de la lista de integrantes. El campo coguía puede ser nulo si no hay co-guía. Un ejemplo de JSON recibido con un solo curso y una sola comisión se puede ver en el código 4.6.

Listing 4.6: Ejemplo de JSON recibido con un solo curso y una sola comisión.

```
{
  "cursos": [{
    "codigo": "CC6919",
    "seccion": 1,
    "semestre": "P",
    "anno": 2025,
    "memorias": [{
      "estudiante": {
        "nombre": "Vicente Olivares",
        "correo": "vicente.olivares@ug.uchile.cl"
      },
      "tema": "Tema de ejemplo",
      "guia": {
        "nombre": "Nombre de ejemplo",
        "correo": "guia@example.com"
      },
      "coguia": null,
      "integrantes": [{
        "nombre": "Integrante de ejemplo 1",
        "correo": "integrante1@example.com"
      }, {
        "nombre": "Integrante de ejemplo 2",
        "correo": "integrante2@example.com"
      }]
    }]
  }]
}
```

Al recibir el JSON, primero se valida el formato de este. Se verifica que tenga la clave `cursos` y que su valor no sea nulo, y se valida que las memorias de cada curso contengan los campos `estudiante`, `tema`, `guia` y `integrantes`. En segundo lugar, se verifica que el contenido no tenga inconsistencias. Para esto se corrobora que no haya dos comisiones con el mismo estudiante y que dentro de cada comisión, el guía, coguía e integrantes sean diferentes. Si cualquiera de estas validaciones falla, se retorna un error 400 BadRequest con un mensaje indicando qué validación falló.

En tercer lugar, se guardan los datos en la base de datos. Para esto, primero por cada curso se obtiene la instancia correspondiente del modelo Curso, si no existe se crea. Luego, por cada comisión se obtiene o crea al estudiante del modelo Estudiante y al guía, coguía e integrantes del modelo Profesor. Al igual que con los cursos, se registran en la base de datos si estos no existían previamente. Después, se obtiene o se crea la instancia de la memoria en el modelo Memoria. Si esta se crea, se agregan al guía, coguía e integrantes como miembros de la comisión con su respectivo rol a través del Modelo Miembro. Mientras que si ya existía, primero se eliminan los miembros anteriores y luego se agregan los miembros que vienen en el JSON. Este comportamiento se implementó así, para que se puedan corregir los miembros de la comisión a través de la API. Habiendo terminado lo anterior, se retorna un mensaje de éxito 201 Created. El código 4.7 muestra el código del guardado de las comisiones sin las validaciones.

Listing 4.7: Guardado de datos en la base de datos.

```
    # Guardado de datos
    for memoria in memorias:
        try:
            estudiante, _ = Estudiante.objects.get_or_create(
                nombre=nombre_normalizado(memoria.get("estudiante").get("nombre")),
                defaults={
                    "correo": memoria.get("estudiante").get("correo"),
                },
            )
            memoria_instance, memoria_created = Memoria.objects.get_or_create(curso=curso, estudiante=
                estudiante, tema=memoria.get("tema"))

            if not memoria_created:
                Miembro.objects.filter(memoria=memoria_instance).delete()

            if "guia" in memoria and memoria.get("guia") is not None:
                profesor, _ = Profesor.objects.get_or_create(
                    nombre=nombre_normalizado(memoria.get("guia").get("nombre")),
                    defaults={
                        "correo": memoria.get("guia").get("correo"),
                    },
                )
                Miembro.objects.create(memoria=memoria_instance, profesor=profesor, rol=rol_guia)
            if "coguia" in memoria and memoria.get("coguia") is not None:
                profesor, _ = Profesor.objects.get_or_create(
                    nombre=nombre_normalizado(memoria.get("coguia").get("nombre")),
                    defaults={
                        "correo": memoria.get("coguia").get("correo"),
                    },
                )
                Miembro.objects.create(memoria=memoria_instance, profesor=profesor, rol=rol_coguia)
            for integrante in memoria.get("integrantes"):
                profesor, _ = Profesor.objects.get_or_create(
                    nombre=nombre_normalizado(integrante.get("nombre")),
                    defaults={
                        "correo": integrante.get("correo"),
                    },
                )
                Miembro.objects.create(memoria=memoria_instance, profesor=profesor, rol=rol_integrante)

        except ValidationError as e:
            return Response(
                {"error": f"Error al guardar la memoria de '{memoria.get('estudiante').get('nombre')}': {e}
                    "},
                status=status.HTTP_400_BAD_REQUEST,
            )
        except Exception as e:
            return Response(
                {"error": f"Error al guardar la memoria de '{memoria.get('estudiante').get('nombre')}': {e}
                    "},
                status=status.HTTP_500_INTERNAL_SERVER_ERROR,
            )

return Response({"success": "Comisiones importadas correctamente."}, status=status.HTTP_201_CREATED)
```

Es importante mencionar que tanto en el modelo Estudiante como el modelo Profesor, se debe buscar por nombre, pues este se define como único. Además, al guardar una instancia de Profesor o Estudiante, se normalizan sus nombres, es decir, que todas las palabras del string se transforman para que comiencen con mayúscula y el resto de caracteres estén en minúscula, como se muestra en el código 4.8. Este comportamiento estaba definido previamente en el SSM. Debido a lo anterior, al realizar la búsqueda por nombre, se debe buscar con el nombre normalizado, ya que de lo contrario puede que no se encuentren nombres con apellidos como “De la Fuente” o nombres como “Juan-Bastián”, pues al normalizarlos se convierten en “De La Fuente” (la pasa a mayúscula) y “Juan-bastián” (Bastián pasa a minúscula) respectivamente. Por lo tanto, se implementó la función `nombre_normalizado`, visible en el código 4.9, para normalizar los nombres. Se hizo una función a parte para realizar la normalización, ya que este comportamiento era requerido en más de una parte del código, en particular en el controlador de importación de comisiones por API y en el procesamiento de archivos CSV subidos por el usuario. Se implementó esta función a modo de parche en vez de modificar los modelos Profesor y Estudiante, ya que no se conocía el impacto que podría tener la modificación en el SSM, pues al no ser el sistema principal de esta memoria, no se conocía en profundidad y se prefirió cambiar lo menos posible.

Listing 4.8: Modelo Estudiante en el SSM.

```
class Estudiante(models.Model):
    nombre = models.CharField(max_length=300, unique=True)
    correo = models.EmailField(max_length=200, null=True, blank=True)

    def save(self, *args, **kwargs):
        self.nombre = " ".join(word.capitalize() for word in self.nombre.split())
        super().save(*args, **kwargs)
```

Listing 4.9: Función para normalizar nombres.

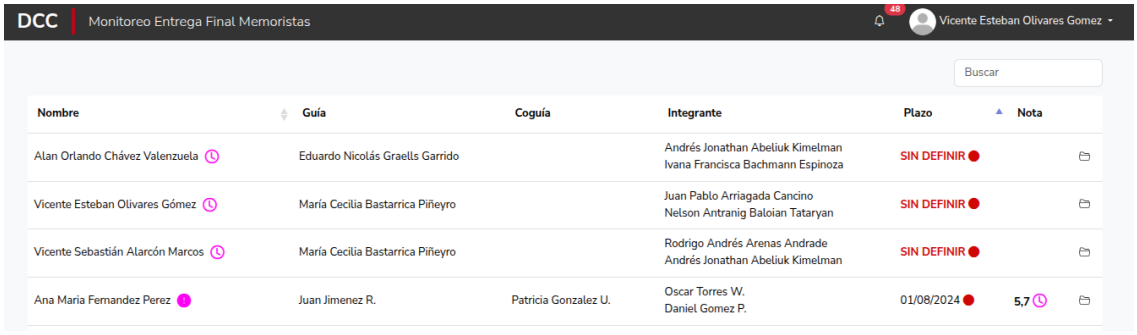
```
def nombre_normalizado(nombre: str):
    """
    Retorna un nombre normalizado, es decir, con la primera letra de cada palabra en mayúscula.
    """
    return " ".join(word.capitalize() for word in nombre.strip().split())
```

Además de la implementación del procesamiento del JSON, se restringió el acceso a la API, para que no cualquier persona que haga un POST pueda enviar datos al SSM. Con este objetivo, se agregó autenticación con tokens al endpoint de la API y para ello, se agregó la extensión de Django llamada Django Rest Framework (DRF) al proyecto, que proporciona herramientas para crear APIs Restful y extiende las posibilidades de autenticación para endpoints. Entonces, se creó un usuario para el sistema de titulación, de nombre `sistema_titulacion`, al cual se le generó un token de autenticación. Con este token el sistema de titulación podrá autenticarse y hacer peticiones POST al endpoint de la API. Para la autenticación es necesario enviar en el header `Authorization` de la petición el token con el prefijo "Token ". Cómo se puede ver en el código 4.10, para proteger el endpoint como se mencionó anteriormente, se utilizaron decoradores de DRF sobre la view de Django que define el endpoint. El decorador `@permission_classes` es para restringir el acceso al endpoint y el decorador `@authentication_classes` se utiliza para especificar el tipo de autenticación que se utilizará.

Listing 4.10: Decoradores utilizados en la API.

```
@api_view(["POST"])
@permission_classes([IsAuthenticated])
@authentication_classes([TokenAuthentication])
def import_comisiones(request):
    data = request.data
    if not data or data.get("cursos") is None:
        return Response({"error": "No se encontraron datos para importar."}, status=status.
            HTTP_400_BAD_REQUEST)
```

Es relevante señalar que el modelo Curso del SSM incluye dentro de sus atributos la fecha límite para la entrega del informe final de Trabajo de Título `fecha_entrega_informe_final` y cuando se crea un curso mediante la importación de comisiones por la API, este campo se inicializa en `None`, dado que dicha información no la maneja el sistema de titulación. El modelo permite crear cursos sin fecha de entrega y es posible asignarle una fecha a cada curso en el SSM más tarde a través de la interfaz gráfica. Sin embargo, la aplicación no estaba diseñada para manejar cursos sin fecha de entrega, ya que varías veces se intentaba calcular una diferencia entre fechas, causando que el sistema arrojara errores. Entonces, se adaptó el SSM para que los casos en los que la fecha de entrega sea `None` no cause errores y en la interfaz aparezca `SIN DEFINIR` en rojo y negrita que sea notorio que se requiere asignar una fecha de entrega, como se puede ver en la figura 4.10a. Además, se aprovechó el sistema de notificaciones del SSM para enviar una notificación al usuario por cada curso sin fecha de entrega definida, como se puede ver en la figura 4.10b.



(a) Comisiones de un curso sin fecha de entrega definida

(b) Notificación de fecha de entrega

Figura 4.10: Cambios en la interfaz del SSM para manejar cursos sin fecha de entrega

Exportación de Comisiones desde el Sistema de Titulación

La exportación de comisiones desde el Sistema de Titulación ejecuta al presionar el botón `Exportar` en la vista del listado de comisiones. Este botón realiza una llamada al endpoint `/comisiones/exportar/<periodo_id>`, donde `<periodo_id>` es el id del periodo académico que se desea exportar. El controlador de este endpoint se puede apreciar en el código 4.11, y se encarga de llamar al comando `export_comisiones_ssm`, que tiene la tarea de realizar la exportación de comisiones, y luego redirigir al usuario al listado de comisiones. Si el comando arroja un error, se envía un mensaje de error al usuario. Por otro lado, si todo sale bien, se envía un mensaje de éxito al usuario. Cabe destacar que es necesario redirigir al usuario para que el mensaje de éxito o error se muestre en la interfaz gráfica, puesto que así está diseñado el sistema de mensajes en el sistema de titulación.

Listing 4.11: Código del controlador que llama al comando de exportación de comisiones.

```
def get(self, request, *args, **kwargs):
  periodo = self.get_periodo()

  try:
    resultado_comando = call_command("export_comisiones_ssm", "--periodo_id", str(periodo.id))
    if resultado_comando == settings.FALLO_EN_COMANDO:
      self.msg_error("Se ha producido un error al exportar la lista de comisiones.")
      return HttpResponseRedirect(reverse_lazy("comisiones:comisiones_list", kwargs={"periodo_id": periodo.
          id}))
    else:
      self.msg_success("Listado de comisiones exportado correctamente.")
      return HttpResponseRedirect(reverse_lazy("comisiones:comisiones_list", kwargs={"periodo_id": periodo.
          id}))
  except CommandError as e:
    self.msg_error(str(e))
    return HttpResponseRedirect(reverse_lazy("comisiones:comisiones_list", kwargs={"periodo_id": periodo.id
        }))
```

```
  except Exception:
    self.msg_error("Se ha producido un error al exportar la lista de comisiones.")
    return HttpResponseRedirect(reverse_lazy("comisiones:comisiones_list", kwargs={"periodo_id": periodo.id
        }))
```

El comando `export_comisiones_ssm`, que se puede apreciar en el código 4.12, recibe como parámetro el periodo académico de las comisiones que se debe exportar y en caso de no recibir ninguno, se toma el periodo académico activo. Luego, se envían las memorias y comisiones de los estudiantes separadas por curso en un JSON a la API del SSM. El formato del JSON es el indicado en la sección 4.3.2. Solo se envían las comisiones que tengan al menos dos evaluadores asignados. En caso de que la respuesta de la API no sea 200 ok o 201 created, se registra el mensaje de error en el log y se arroja un `CommandError` con el mensaje de error. Por último, para incorporar las credenciales de la API del SSM, se agregaron las variables de entorno `SSM_URL` y `SSM_TOKEN` que corresponden a la URL y el token de la API del SSM, respectivamente.

Listing 4.12: Comando de exportación de comisiones.

```
def handle(self, *args, **options):
  periodo_id = options.get("periodo_id")
  if periodo_id is None:
    periodo = Periodo.objects.filter(is_activo=True).first()
  else:
    try:
      periodo = Periodo.objects.get(id=periodo_id)
    except Periodo.DoesNotExist:
      self.stdout.write(self.style.ERROR("Periodo no encontrado."))
      return settings.FALLO_EN_COMANDO

  cursos = Curso.objects.select_related("periodo").filter(periodo=periodo)
  memorias = MemoriaEnF.objects.select_related(
    "estudiante__persona",
    "tema__guia__persona",
    "tema__coguia__persona"
  ).prefetch_related(
    "comision__evaluadores__persona"
  ).filter(periodo=periodo)

  comisiones_por_curso = {
    "cursos": [
      {
        "codigo": curso.codigo,
        "seccion": curso.seccion,
        "semestre": "O" if curso.periodo.periodo == 1 else "P",
        "anno": curso.periodo.ano,
        "memorias": [
          memoria.to_ssm_dict() for memoria in memorias.filter(curso=curso) if memoria.comision.evaluadores
              .count() >= 2
        ]
      } for curso in cursos
    ]

  }

  response = requests.post(
    settings.SSM_URL + "/api/comisiones/",
    json=comisiones_por_curso,
    headers={"Authorization": f"Token {settings.SSM_TOKEN}"},
  )

  if response.status_code not in (requests.codes.ok, requests.codes.created):
    self.stdout.write(self.style.ERROR(f"Error al exportar comisiones: {response.text}"))
    raise CommandError(response.json()["error"])

  # Establece la comisión de un estudiante como publicada y se podr\'a ver en la ficha del estudiante
  for memoria in memorias:
    memoria.comision.publicada = True
    memoria.comision.save()

  self.stdout.write(self.style.SUCCESS("Comisiones exportadas correctamente"))

  return 0
```

4.4. Despliegue

El último de los objetivos específicos definidos en la sección 1.2 es el despliegue del sistema en el servidor de producción y en esta sección se describen los pasos que se siguieron para llevar a cabo el despliegue tanto del sistema de titulación como del SSM con los cambios realizados.

Como se mencionó en la sección 1.3.4, el despliegue se realizará en los servidores del DCC para que pueda ser utilizada por la coordinación de titulación. Para ello, durante todo el desarrollo del sistema se utilizaron contenedores de Docker con Docker Compose para coordinar la creación de los contenedores de la base de datos Postgres y el servidor web Django, facilitando el despliegue. El uso de Docker Compose no es algo incorporado por esta memoria en el sistema de titulación y el SSM, ambos sistemas ya lo utilizaban.

El primer despliegue del sistema de titulación se realizó en un servidor de test del DCC, con el objetivo de realizar la evaluación. En esa instancia no se desplegó el SSM, ya que no se habían realizado cambios en el SSM hasta ese momento. Luego de la evaluación, se aplicaron los cambios sugeridos en la retroalimentación dada por la coordinadora de titulación y se implementó la exportación directa de comisiones al SSM descrita en la sección 4.3.2. Con estos cambios, se procedió a realizar un segundo despliegue en los servidores de test. Esta vez, se incluyó tanto el sistema de titulación como el SSM con los cambios realizados. Se utilizó este despliegue para mostrarle los cambios del SSM a la jefa docente Jocelyn Simmonds, quien está a cargo de este sistema. Luego de probarlo, ella pidió que se subieran los cambios al servidor de producción para utilizarlo en la gestión de memorias del semestre de primavera 2025.

Capítulo 5

Validación

En este capítulo se describe la evaluación de la herramienta de asignación de comisiones examinadoras. Primero, se explica el método utilizado para la evaluación y luego se comentan los resultados obtenidos.

5.1. Evaluación

Para establecer el método de evaluación, primero es necesario repasar el objetivo definido en la sección 1.2. Este es desarrollar y desplegar una herramienta que permita asignar comisiones examinadoras de estudiantes de Trabajo de Título de manera eficiente. Esto incluye que la herramienta se comunique con los sistemas existentes relacionados con el proceso de titulación. Por lo tanto, para evaluar la herramienta se requiere poner a prueba que la asignación de comisiones examinadoras con el software implementado sea más eficiente que el proceso actual, es decir, asignar comisiones examinadoras usando una planilla de Excel.

Teniendo en cuenta el objetivo mencionado, se decidió que el método para evaluar la herramienta sea que la coordinadora de titulación asigne las comisiones examinadoras de los estudiantes de Trabajo de Título del semestre primavera 2025 con el software desarrollado desplegado en un servidor de test del DCC, ya que esto permite evaluar la herramienta en un entorno cercano al proceso real, con datos reales, con la diferencia de que el software se encuentra desplegado en un servidor de test del DCC, en vez de uno de producción. Los datos utilizados de estudiantes cursando Trabajo de Título en el semestre primavera 2025 fueron obtenidos desde la API de UCampus al momento de la evaluación, mientras que el resto de datos, como los profesores, las memorias de Introducción al Trabajo de Título y los temas, fueron obtenidos desde un backup al sistema de titulación realizado una semana antes de la evaluación. Respecto a las personas que participarán de la evaluación, se consideró solo a la coordinadora de titulación, ya que al ser la única encargada de asignar comisiones examinadoras y única usuaria del módulo de asignación de comisiones examinadoras, es la única persona que puede validar la herramienta y percibir su eficiencia. Por lo tanto, los resultados obtenidos serán más realistas, tanto en la eficiencia percibida por la coordinadora, como en los posibles errores que puedan ocurrir.

La asignación de comisiones examinadoras realizada en el servidor de test puede ser utilizada como asignación oficial de comisiones examinadoras, dado que las comisiones asignadas en la evaluación pueden ser exportadas a través de la descarga de un archivo CSV y ser revisadas por la coordinadora de titulación antes de comunicarlas oficialmente a la jefatura de estudios.

Dos puntos importantes a mencionar son que la evaluación se realizó una semana antes de la fecha límite de asignación, para que la coordinadora de titulación tuviera tiempo para realizar la asignación de comisiones examinadoras con el método anterior, en caso de que hubiera sido inviable asignar las comisiones con el software implementado. Segundo, al momento de la evaluación, no se había implementado la exportación de comisiones al SSM por medio de la API, por lo que esto no se evaluó.

5.2. Resultados

La asignación de comisiones examinadoras con el software implementado fue realizada con éxito, ya que la coordinadora fue capaz de asignar las comisiones y exportarlas a un archivo CSV, siendo este archivo utilizado oficialmente para comunicar las comisiones examinadoras a la jefatura de estudios. La coordinadora dejó comentarios sobre los aspectos positivos y los que creía que se debían mejorar.

En cuanto a los aspectos positivos, la coordinadora mencionó que el proceso fue más rápido que el proceso manual, y por lo tanto, más eficiente. También mencionó que el gráfico de carga fue de mucha ayuda para visualizar la carga de los evaluadores y así poder distribuir a los evaluadores en las comisiones sin sobrecargarlos. Por último, mencionó que fue útil que la barra de búsqueda permitiera buscar evaluadores por nombre y por área de conocimiento.

En cuanto a los aspectos que creía que se debían mejorar, la coordinadora mencionó que en la lista de alumnos a quienes se debe asignar comisión no se muestra coguía ni coguía externo. También mencionó que después de asignar una comisión, la página se recarga y el listado de estudiantes se muestra desde el principio, por lo que cada vez tuvo que hacer scroll para proceder con la siguiente asignación. Otra observación fue que al seleccionar un alumno para asignarle una comisión, el modal tarda alrededor de 13 segundos en desplegarse. Por último, indicó que el gráfico de carga tiene mucho margen superior, perdiendo espacio que podría usarse para ampliar el gráfico.

5.3. Discusión de resultados

Los comentarios positivos de la coordinadora de titulación indican que las funcionalidades implementadas como el gráfico de carga y las barras de búsqueda fueron útiles para realizar la asignación de comisiones examinadoras de manera más eficiente, ya que dan acceso rápido al usuario a información relevante para la asignación de comisiones. Por otro lado, a excepción de la falta de coguía y coguía externo, los aspectos que se deben mejorar no son problemas graves que impidan la utilización de la herramienta, sino aspectos que pueden mejorar la

experiencia del usuario y la eficiencia de la asignación de comisiones examinadoras.

En general, la evaluación del módulo de asignación de comisiones examinadoras se puede considerar exitosa, ya que cumple con el objetivo de implementar una herramienta que permita asignar comisiones examinadoras de manera eficiente, puesto que realizar la asignación con la implementación realizada permite completar el proceso más rápido que con el método manual. Incluso teniendo en cuenta la demora de aparición del modal de asignación de comisión y el scroll que se debe hacer para proceder con la siguiente asignación, el proceso es más rápido que el método anterior.

5.4. Correcciones

Dados los comentarios de la coordinadora de titulación, se realizaron correcciones a la herramienta para mejorar la experiencia del usuario y la eficiencia de la asignación de comisiones examinadoras. Primero, se corrigió la falta de coguía y coguía externo de forma inmediata, ya que la falta de estos se debía a un error de tipografía en el template de la tabla de asignación de comisiones examinadoras. Posteriormente, se redujo el margen superior del gráfico de carga, lo que permitió ampliar el gráfico verticalmente.

Respecto a la demora de aparición del modal de asignación de comisión, se optimizó el código del controlador asociado al modal, precalculando la cadena de texto con las áreas de conocimiento de cada evaluador y evitando consultar por todos los evaluadores cada vez que se abre el modal. Por último, se eliminó la recarga de la página al asignar una comisión, lo que permitió quitar el scroll que se debe hacer para proceder con la siguiente asignación. El razonamiento y la justificación de la corrección de los últimos dos aspectos puede encontrarse en la sección 4.2.4.

Capítulo 6

Conclusiones

El desarrollo del módulo de asignación de comisiones examinadoras en el sistema de titulación del DCC tuvo como objetivo general desarrollar y desplegar una herramienta que permita asignar comisiones examinadoras a los estudiantes de Trabajo de Título de forma eficiente y que se comunique con los sistemas existentes relacionados con el proceso de titulación. Este objetivo fue logrado con éxito, ya que se implementó una extensión del sistema de titulación del DCC que permite a la coordinadora de titulación asignar comisiones examinadoras a los estudiantes de Trabajo de Título en menor tiempo que el método anterior de planillas Excel y que es capaz de exportar las comisiones asignadas al Sistema de Seguimiento de Memorias por dos métodos: la descarga de un archivo CSV y mediante una consulta POST a la API del Sistema de Seguimiento de Memorias. Además, se desplegó la herramienta en el servidor de producción del DCC, dejándola lista para su uso.

Dentro de los aprendizajes obtenidos, se destaca la importancia de tener reuniones periódicas con la profesora guía para mantener la constancia en el trabajo, ya que inicialmente no se tenían y en cuanto se fijó un horario para tener reuniones semanales, el ritmo de trabajo aumentó. Las reuniones, tanto con la profesora guía como con el equipo de desarrollo del DCC, requirieron de saber expresar lo que se estaba haciendo y los problemas que surgían, para así obtener ayuda o sugerencias que permitieran resolver las dificultades que se tuvieron.

Uno de los principales desafíos de la memoria fue trabajar sobre código desarrollado por otras personas, en particular trabajar sobre el código del piloto de asignación de comisiones examinadoras, ya que no se tenía una documentación detallada del código y al ser una versión piloto, funcionaba pero tenía errores que se debían corregir y optimizar. Gran parte del tiempo dedicado al desarrollo se centró en corregir errores que surgían de casos borde no considerados en el piloto u optimizar el código para que fuera más eficiente.

En cuanto a aspectos a mejorar y trabajos futuros en la asignación de comisiones examinadoras, primero se deben realizar mejoras en la usabilidad de la herramienta, por ejemplo, mientras se obtienen los estudiantes de Trabajo de Título o mientras se exportan las comisiones al SSM no se muestra ningún indicador que permita al usuario saber que la operación está en curso. Otro aspecto a mejorar son dos comportamientos no deseados presentes en el sistema que no hacen que la herramienta se caiga. El primero aparece cuando se obtienen los estudiantes de Trabajo de Título con el botón Sincronizar en el listado de comisiones. Al

completar la operación, se redirige al usuario al listado de estudiantes de Trabajo de Título en vez de simplemente actualizar la página. El segundo aparece al guardar una comisión examinadora y se tiene algún error de validación. En este caso puede que la alerta de error se muestre más de una vez. Un último trabajo futuro sería incorporar el procesamiento de solicitudes de vía rápida a través del sistema de titulación. Esto afecta al módulo de Trabajo de Título, ya que se deben haber ingresado al sistema de titulación los temas de los estudiantes que desean solicitar vía rápida, para así poder identificarlos correctamente como estudiantes de Trabajo de Título y poder asignarles comisiones examinadoras.

A modo de resumen, el desarrollo del módulo de asignación de comisiones examinadoras en el sistema de titulación del DCC tuvo éxito en cumplir su objetivo general. Este trabajo aporta a que el Departamento de Ciencias de la Computación tenga herramientas más adecuadas ante el aumento de estudiantes que ingresan a la carrera de Ingeniería Civil en Computación, como lo son el sistema de titulación y el sistema de seguimiento de memorias. Además, permite que el sistema de titulación del DCC no solo cubra el ramo de Introducción al Trabajo de Título, sino que también cubra parte del ramo Trabajo de Título. Esto abre puertas para agregar más funcionalidades al sistema de titulación relacionadas con Trabajo de Título.

Bibliografía

[1] D. Orellana. Extensiones de la plataforma web de monitoreo de memoristas del departamento de ciencias de la computación. Trabajo de titulación, Universidad de Chile, 2025.

[2] Oportot R. Sistema de recomendación para evaluar memorias en el dcc. Trabajo de titulación, Universidad de Chile, 2024. URL: https://repositorio.uchile.cl/handle/2250/201449.

[3] M. Rivas. Plataforma web de monitoreo de memoristas del departamento de ciencias de la computación. Trabajo de titulación, Universidad de Chile, 2024. URL: https://repositorio.uchile.cl/handle/2250/204040.

[4] UCampus. Catálogo de cursos de ucampus de la fcfm, 2024. URL: https://ucampus.uchile.cl/m/fcfm_catalogo/?semestre=20242&depto=5&force=0.

[5] UCampus. Catálogo de cursos de ucampus de la fcfm, 2025. URL: https://ucampus.uchile.cl/m/fcfm_catalogo/?semestre=20251&depto=5&force=0.

[6] Universidad de Chile. Reglamento general de los estudios de pregrado impartidos por la facultad de ciencias físicas y matemáticas de la universidad de chile, 2018. URL: https://ingenieria.uchile.cl/dam/jcr:4e3a6140-e743-4844-a196-0cc67e43a60a/CCI30012019.pdf.