

Universidad de La Habana  
Facultad de Matemática y Computación



# **Sistema de gestión para el funcionamiento de un departamento docente en la facultad de Matemática y Computación de la UH**

Autor:

**Juan Carlos Casteleiro Wong**

Tutores:

**Msc. Fernando Rodriguez Flores**

**Lic. Alain Cartaya Salabarría**

Trabajo de Diploma  
presentado en opción al título de  
Licenciado en Ciencia de la Computación

Fecha

[github.com/username/repo](https://github.com/username/repo)

Dedicación

# Agradecimientos

Agradecimientos

# Opinión del tutor

Opiniones de los tutores

# Resumen

Resumen en español

# Abstract

Resumen en inglés

# Índice general

<b>Introducción</b>	<b>1</b>
<b>1. Preliminares</b>	<b>2</b>
1.1. Funcionamiento de la facultad . . . . .	2
1.1.1. Asignación de docencia . . . . .	2
1.1.2. Planificación de las tesis . . . . .	5
1.2. Herramientas utilizadas para el desarrollo del software . . . . .	7
1.2.1. Modelo cliente-servidor . . . . .	7
1.2.2. SQLite . . . . .	8
1.2.3. Desarrollo del servidor . . . . .	9
1.2.4. Desarrollo del cliente . . . . .	11
<b>2. Descripción de las funcionalidades que se desean</b>	<b>12</b>
2.1. Asignación de docencia . . . . .	12
2.2. Planificación de las tesis . . . . .	15
<b>3. Descripción del diseño de la base de datos</b>	<b>18</b>
3.1. Modelación de la asignación de docencia . . . . .	18
3.2. Modelación de los tribunales de tesis . . . . .	22
<b>4. Descripción las herramientas implementadas</b>	<b>24</b>
4.1. Asignación de docencia . . . . .	25
4.2. Planificación de las tesis . . . . .	32
4.3. Exportar e importar datos de la base de datos . . . . .	38
4.3.1. Exportar los datos de la base de datos . . . . .	38
4.3.2. Importar datos a la base de datos . . . . .	39
<b>5. Extensibilidad</b>	<b>40</b>
5.1. Estructura del cliente y servidor . . . . .	40
5.2. Recomendaciones para agregar un nuevo proceso en el sistema de gestión	47

Conclusiones	49
Recomendaciones	50
Bibliografía	51



# Índice de figuras

3.1. MER para la asignación de docencia . . . . .	19
3.2. MER para la planificación de las tesis . . . . .	22
4.1. Vista inicial del sistema de gestión . . . . .	25
4.2. Vista de la planificación de las cargas de las asignaturas . . . . .	26
4.3. Vista del formulario para crear una carga de asignaturas . . . . .	27
4.4. Vista de la carga de las asignaturas . . . . .	27
4.5. Vista de la asignación de docencia . . . . .	28
4.6. Vista del formulario para realizar una asignación de docencia . . . . .	28
4.7. Vista del formulario con los datos ingresados de una asignación de docencia . . . . .	29
4.8. Vista resultante de realizar una asignación de docencia correctamente	30
4.9. Vista con varias asignaciones de docencia . . . . .	31
4.10. Vista resultado de la asignaciones de docencia . . . . .	32
4.11. Vista de la tesis. . . . .	33
4.12. Vista del formulario para agregar una tesis. . . . .	34
4.13. Vista de la interfaz de usuario tras agregar las dos tesis del ejemplo. .	34
4.14. Vista de la interfaz de usuario para la confección de los tribunales. . .	35
4.15. Vista del formulario para la confección de los tribunales. . . . .	35
4.16. Vista resultante de confeccionar el tribunal de la primera tesis de la tabla. . . . .	36
4.17. Vista del formulario para la confección de los tribunales. . . . .	36
4.18. Interfaz de usuario para programar las defensas de tesis . . . . .	37
4.19. Vista del formulario para la planificación de las tesis. . . . .	37
4.20. Vista resultado de las planificaciones de tesis. . . . .	38
5.1. Estructura de los ficheros del proyecto servidor . . . . .	41
5.2. Estructura de los ficheros del proyecto servidor . . . . .	42
5.3. Tabla para los departamentos . . . . .	47

## Ejemplos de código

# Introducción

# Capítulo 1

## Preliminares

El objetivo de este trabajo es diseñar e implementar un sistema que permita informatizar dos de los procesos que se llevan a cabo en la Facultad de Matemática y Computación de la Universidad de La Habana (MATCOM): la asignación de docencia y las planificaciones de las tesis.

En este capítulo se describe cómo se realizan estos procesos actualmente y se presentan los elementos fundamentales de las herramientas que se utilizaron en el trabajo.

En la sección 1.1 se describe el funcionamiento de la facultad, en particular, cómo se llevan a cabo la asignación de docencia y la planificación de las tesis. En la sección 1.2 se presentan las características principales de las herramientas que se utilizaron para la implementación del sistema.

### 1.1. Funcionamiento de la facultad

En MATCOM, todos los semestres se realizan un conjunto de procesos administrativos y de planificación relacionados con los curso escolares, tales como: la asignación de docencia, la planificación de las tesis de pregrado, la planificación de los exámenes finales y revalorizaciones, la gestión de los alumnos ayudantes, la planificación de los cursos optativos, entre otros. En este trabajo nos interesa informatizar los dos primeros.

En las siguientes secciones se describe cómo se efectúan los procesos de asignación de docencia y planificación de las tesis actualmente en la facultad.

#### 1.1.1. Asignación de docencia

La asignación de docencia, consiste en determinar qué profesor debe impartir cada asignatura. Para conocer cuáles asignaturas se deben impartir en cada semestre, se

debe tener en cuenta el plan de estudio por el que se rige cada año.

El plan de estudio es un programa estructurado donde se definen los contenidos que forman parte del currículo de los graduados. En el plan de estudio se determina la cantidad de horas clase de cada asignatura, así como el año y semestre en el que se deben impartir. Con el objetivo de mantener la formación de los profesionales actualizada con el desarrollo de la sociedad, cada cierto tiempo el plan de estudio de una carrera universitaria se modifica.

En el curso 2022, en la carrera de Ciencia de la Computación, existen tres planes de estudio vigentes. En el año 2018 entró en vigor el plan de estudio E, donde se redujo la cantidad de años escolares de 5 a 4. Además, como consecuencia de la situación epidemiológica que provocó la COVID-19, fue necesario modificar el plan E, creándose un plan de estudio E modificado, que en este documento se le referencia como E (COVID). De forma que, los estudiantes que ingresaron en la facultad en el curso 2017-2018 y que se encuentran en el quinto año de la carrera, se rigen por el plan D, los estudiantes que se encuentran en primero, tercero y cuarto año, se rigen por el plan E y los estudiantes que cursan el segundo año, se rigen por el plan E (COVID).

Esta variedad de planes de estudios vigentes, provoca que para realizar la asignación de docencia correspondiente al primer semestre del curso 2022, se deban contemplar las asignaturas que se imparten en quinto año según el plan D; en primero, tercero y cuarto según el plan E; y en segundo según el plan E (COVID). En caso de que solo haya un plan de estudio vigente, el proceso se reduce y solo se deben impartir las asignaturas establecidas en ese plan.

Cada asignatura se imparte por un departamento, en MATCOM existen 4 departamentos: Matemática, Matemática Aplicada, Programación y Sistemas de Información e Inteligencia Artificial y Sistemas Computacionales. En un departamento se realizan actividades de coordinación de docencia, investigación y tutoría de estudiantes. Los departamentos están formados por profesores y cuentan con un jefe de departamento. Los profesores tienen un grado científico (Licenciado, Máster, Doctor) y una categoría docente (Titular, Auxiliar, Instructor, Asistente). Los departamentos de una facultad pueden impartir asignaturas en otras facultades, a estas asignaturas se le llama docencia externa. Por ejemplo, el departamento de Matemática Aplicada imparte en el segundo año de la carrera de Geografía la asignatura Estadística. Las asignaturas que se imparten en otras facultades, a los efectos del proceso de asignación de docencia, tienen la misma relevancia que las que se imparten en la facultad.

En la tabla 1.1 se muestran las asignaturas que debe impartir el departamento de Matemática Aplicada en el período enero-julio del curso 2022.

Asignatura	Año	Facultad	Horas	Grupos
Programación y Algoritmos	M1	MATCOM	64	1/3
Seminarios de Problemas	M1	MATCOM	48	1/3
Programación y Algoritmos	M2	MATCOM	32	1/1
Inferencia Estadística	M3	MATCOM	80	1/1
Matemática Numérica	M3	MATCOM	48	1/1
Optimización Matemática I	M3	MATCOM	64	1/1
Estadística	C3	MATCOM	64	1/2
Modelos de Optimización I	C3	MATCOM	64	1/2
Estadística	G2	GEOGRAFÍA	80	1/2
Estadística	S2	SOCIOLOGÍA	80	1

Tabla 1.1: Asignaturas que debe impartir el departamento de Matemática Aplicada en el período enero–julio del curso 2022.

En las columnas de la tabla 1.1 se muestran: el nombre de la asignatura, el año escolar en el cual se imparte, la facultad a la que pertenece el año escolar, el total de horas clase y la cantidad de grupos de conferencias y clases prácticas. Por ejemplo, la primera fila indica que la asignatura Programación y Algoritmos que se imparte en el primer año de la carrera de Matemática, de la facultad MATCOM, tiene un total de 64 horas clase y cuenta con un grupo de conferencias y tres de clases prácticas.

Cada asignatura tiene un total de horas que se deben impartir de acuerdo al plan de estudio al que pertenezcan. La distribución del total de horas clase en conferencias, clases prácticas u otras actividades de clase, se decide por el colectivo de profesores de la asignatura y puede cambiar de un año a otro.

Para realizar la asignación de la docencia en un departamento, se deben distribuir los profesores, de forma que cubran la cantidad de horas de conferencias y clases prácticas de cada asignatura. En la tabla 1.2 se muestra un fragmento de una posible asignación de docencia.

Asignatura	Horas	Grupos	Profesores
Optimización Matemática I	64	1/1	C: Aymeeé Marrero (32) CP: Gemayqzel Bouza(32)
Modelos de Optimización I	64	1/2	C: Aymeeé Marrero(16) C: Fernando Rodríguez (16) CP: Daniela González(32) CP: Camila Pérez(32)
Estadística	80	1/2	C: Elianys García (48) CP: Ernesto Borrego(32) CP: Elianys García(32)

Tabla 1.2: Fragmento de una posible asignación de docencia en el departamento de Matemática Aplicada.

En las columnas de la tabla 1.2 se muestran: el nombre de la asignatura, el total de horas clase, la cantidad de grupos de conferencia (C) y clases prácticas (CP), y los profesores que se asignaron para cubrir cada actividad de clase, especificando el total de horas que imparte cada uno. Por ejemplo, en la primera fila se indica que la asignatura Optimización Matemática I tiene un total de 64 horas clase, con un grupo de conferencia y uno de clases prácticas, y que la profesora Aymeeé Marrero imparte las 32 horas de conferencia y la profesora Gemayqzel Bouza imparte las 32 horas de clases prácticas.

Existen asignaturas donde la cantidad de horas de las actividades de clase se comparten entre varios profesores, por ejemplo, en la asignatura de Modelos de Optimización I, que se muestra en la segunda fila de la tabla 1.2, se deben impartir 32 horas de conferencia, de las cuáles 16, fueron asignadas a la profesora Aymeeé Marrero y las restantes 16, al profesor Fernando Rodríguez.

En conclusión, el proceso de asignación docente se realiza en cada departamento, al comienzo de cada semestre. A partir de los planes de estudios vigentes, se puede conocer las asignaturas que se deben impartir en cada período de tiempo. El jefe del departamento debe determinar qué profesor imparte cada actividad de clase de cada asignatura, de forma que se cubran todos los grupos y horas clase.

En la siguiente sección se describe cómo se realiza el proceso de planificación de tesis en la facultad de MATCOM.

### 1.1.2. Planificación de las tesis

El ejercicio de culminación de estudios de pregrado consiste en la realización de un trabajo de diploma o tesis. Las tesis se deben discutir frente a un tribunal en una fecha especificada.

El proceso de planificación de las tesis se puede separar en dos subprocesos principales: la confección de los tribunales y la programación de las defensas. La confección de los tribunales consiste en determinar qué profesor formará parte de cada tribunal y la programación de las defensas consiste en definir, en qué día, hora y lugar se realizará cada acto de defensa.

La tesis es un trabajo de investigación que realizan los estudiantes bajo la tutela de uno o varios profesores (tutores y cotutores) sobre un tema que tribute a alguna de las áreas que abarca la carrera. Para la confección de los tribunales se debe tener en cuenta que los miembros estén relacionados con los temas que se aborden en la investigación. La composición de los tribunales puede variar de un año a otro de acuerdo a las condiciones del curso escolar. Es usual que los tribunales estén formados por tres profesores, y que además participe un oponente y el tutor. Sin embargo, en el curso 2021 los tribunales estuvieron compuestos por un presidente, un oponente y un vocal.

Una vez que se definan los tribunales, es necesario determinar cuándo y dónde se llevarán a cabo las defensas de la tesis. Para ello, se debe definir la hora, fecha y local. Para realizar la programación de las defensas se debe contemplar la disponibilidad de los locales de la facultad y que cada profesor solo tenga una defensa a la que asistir en cada instante de tiempo.

A continuación se ilustra el proceso de planificación para dos tesis. Se asume que los tribunales se conformarán con dos profesores, un oponente y un presidente.

ID	Tesis	Estudiante	Tutores
1	Simulación y optimización de movimientos de malabares	Gustavo Despaigne	Fernando Rodríguez
2	Propagación de epidemias mediante modelos basados en metapoblaciones	Abel Antonio Cruz	Angela M. León José A. Mesejo

Tabla 1.3: Datos de las tesis

En la tabla 1.3 se muestra los datos asociados a las tesis, como el título, el nombre del autor y los tutores, que pueden ser más de uno. Además se agregó un identificador (ID), para referenciar una tesis durante la explicación del proceso.

El primer paso es realizar la confección de los tribunales de tesis, donde se debe definir el oponente y el presidente. Para ello, se deben tener en cuenta todos los posibles profesores que por su actividad investigativa, estén relacionados con el contenido de la tesis, y de ellos escoger los que formarán el tribunal. En la tabla 1.4 se ilustran posibles tribunales para las dos tesis anteriores.



ID Tesis	Tutores	Oponente	Presidente
1	Fernando Rodríguez	Gemayqzel Bouza	Aymeeé Marrero
2	Angela M. León José A. Mesejo	Damian Valdés	Gemayqzel Bouza

Tabla 1.4: Posibles tribunales de las tesis

Una vez que se definan los tribunales, se debe determinar la fecha, hora y lugar donde se realizarán los actos de defensas, un ejemplo hipotético se muestra en la tabla 1.5.

ID Tesis	Fecha	Hora	Local
1	10/12/2022	2:00 PM	Aula Posgrado
2	10/12/2022	3:30 PM	Salón del Decanato

Tabla 1.5: Posible horario para las defensas de las tesis

Si el acto de defensa de la tesis 2, se hubiera planificado a la misma hora que el de la tesis 1, provocaría un conflicto, ya que un miembro del tribunal (Gemayqzel Bouza) debe estar presente en dos lugares a la vez. Este es uno de los tipos de situación que se desea evitar con el desarrollo de este trabajo.

En la siguiente sección se describen las herramientas y tecnologías que se utilizaron para la informatización de los procesos de asignación de docencia y planificación de las tesis.

## 1.2. Herramientas utilizadas para el desarrollo del software

Para la informatización de los procesos de asignación de docencia y planificación de las tesis se implementó una aplicación web, utilizando un modelo cliente-servidor. Para el desarrollo en el lado del servidor se hizo uso de Django, aprovechando la versatilidad, seguridad, escalabilidad y mantenibilidad que ofrece. El cliente se desarrolló usando Quasar, creando una *single-page application* (SPA). El sistema gestor de base de datos (SGBD) escogido fue SQLite. A continuación se profundiza en el modelo y herramientas utilizadas.

### 1.2.1. Modelo cliente-servidor

El modelo cliente-servidor se define como una arquitectura de software compuesta por los proveedores de un recurso o servicio, denominados servidores, y los solicitantes

del servicio, denominados clientes [1]. En este modelo se realiza una comunicación de procesos que implica el intercambio de datos tanto por parte del cliente como del servidor, realizando cada uno funciones diferentes. La comunicación se realiza frecuentemente a través de una red informática, pero tanto el cliente como el servidor pueden residir en un mismo sistema. Entre las principales ventajas que ofrece este modelo se encuentran:

- centralización de los recursos, ya que los accesos y la integridad de los datos se controlan en el lado del servidor, de forma que un programa cliente defectuoso o no autorizado no puede dañar el sistema.
- división del procesamiento de la aplicación en varios dispositivos.
- permite la escalabilidad del sistema, en cualquier momento se puede mejorar la capacidad de procesamiento aumentando la cantidad de servidores sin que el funcionamiento de la red se vea afectado.
- la separación en cliente y servidor permite el uso de tecnologías enfocadas en las labores específicas que realiza cada uno.

Tanto la planificación de las tesis como la asignación de la docencia, requieren de un conjunto de datos e informaciones que es necesario almacenar. Para ello, se utiliza una base de datos. En la siguiente sección se presentan las características principales de SQLite, que fue el SGBD que se utilizó en el trabajo.

### 1.2.2. SQLite

SQLite es una biblioteca escrita en C, que implementa un motor de base de datos SQL. Se caracteriza por su portabilidad, confiabilidad [2] y rendimiento [3]. Sus transacciones satisfacen las propiedades: atomicidad, consistencia, aislamiento y durabilidad (ACID) [4]. A continuación se describen las principales ventajas y desventajas de SQLite.

#### **Ventajas:**

- No es necesario instalar ninguna dependencia externa para utilizar SQLite [5].
- SQLite se describe como una base de datos de “configuración cero” [6], que está lista para usar de inmediato. No se ejecuta como un proceso servidor, por lo que no es necesario detenerlo, iniciarlo o reiniciarlo.
- Almacena todos los datos en un solo archivo, permitiendo la portabilidad de la base de datos.

**Desventajas:**

- Aunque varios procesos pueden acceder y consultar una base de datos SQLite simultáneamente, solo uno puede realizar cambios en la base de datos en un momento dado. Los SGBD que implementan un modelo cliente-servidor como MySQL o PostgreSQL permiten una mayor concurrencia.
- Los SGBD que implementan un servidor pueden proporcionar una mejor protección contra errores en la aplicación cliente que una base de datos sin servidor como SQLite.
- El sitio web de SQLite recomienda que las bases de datos que se acerquen a 1 TB se alojen en una base de datos cliente-servidor centralizada [7].

Tras analizar las principales ventajas y desventajas de SQLite, y realizar una comparación con otros de los SGBD como MySQL [8] y PostgreSQL [9], se decidió utilizar SQLite por las siguientes razones:

- No se necesitan realizar cambios en la base de datos de forma concurrente durante los procesos de asignación docente y planificación de las tesis.
- El tamaño de la base de datos, con la información necesaria para el funcionamiento de estos procesos no debe exceder de 50 MB.
- La fácil integración de SQLite en un proyecto y la portabilidad que ofrece.

En la siguiente sección se describen las características principales de la tecnología que se utilizó en el lado del servidor para el desarrollo de la aplicación web.

**1.2.3. Desarrollo del servidor**

Para el desarrollo en el lado del servidor se hizo uso de la biblioteca Django, que permite la creación de sitios web seguros y mantenibles [10]. Entre sus principales características se encuentran:

- Sigue la filosofía de diseño DRY (Don't Repeat Yourself), brindando un conjunto de funcionalidades implementadas por los desarrolladores para evitar la repetición de código en procesos comunes en el desarrollo web.
- Implementa el patrón de diseño Model-View-Template (MVT), que consta de tres componentes esenciales Modelo, Vista, Plantilla. La separación del código siguiendo este patrón de diseño permite desarrollar aplicaciones modulares, escalables y mantenibles [11].

- Está implementado en Python, por lo que cuenta con un extenso conjunto de bibliotecas para resolver distintas tareas. Entre las bibliotecas más utilizadas resalta Django REST framework, desarrollada para la creación de interfaces de programación de aplicaciones (API).
- Provee un Object Relational Mapper (ORM), que permite la interacción con la base de datos de forma orientada a objetos [12]. Brinda la posibilidad de crear tablas, insertar, editar, borrar y extraer datos sin escribir consultas SQL, acelerando el proceso de desarrollo web.
- Sigue también la filosofía de “Batteries Included” o “Baterías Incluidas” proporcionando una biblioteca estándar rica y versátil con herramientas para crear sistemas complejos sin la necesidad de instalar paquetes separados. Algunas de estas “baterías” son “Django Admin”, “Django ORM”, “Authentication” y “HTTP”.
- Cuenta con extensa documentación, desde la documentación oficial hasta contenido en forma de artículos, tutoriales y cursos en línea.

Las principales razones que hicieron que Django fuera la herramienta escogida se presentan a continuación:

- La biblioteca Django REST Framework brinda un conjunto de funcionalidades que permiten que la creación de la API sea una tarea simple en comparación con empezar desde cero. En este trabajo se desea implementar una API RESTful que brinde toda la información que interviene en los procesos de asignación de docencia y planificación de las tesis.
- El ORM de Django que permite la creación de tablas y consultar los datos sin la necesidad de escribir consultas SQL.
- El panel de administración que viene instalado por defecto en Django. En el cual se pueden realizar operaciones CRUD a la base de datos durante la fase de desarrollo, sin la necesidad de implementar interfaces de usuarios.

Aunque Django incluye herramientas para el desarrollo en el lado del cliente con el uso de Plantillas, en este trabajo se decidió utilizar Quasar, los motivos se describen en la próxima sección.

### 1.2.4. Desarrollo del cliente

Para el desarrollo de la aplicación web en el lado del cliente se utilizó Quasar [13]. Quasar es una biblioteca de código abierto que se basa en Vue.js [14].

A pesar de que Vue.js se define en su sitio oficial como “Un marco accesible, eficaz y versátil para crear interfaces de usuario web” [14], no proporciona elementos o componentes de interfaz de usuarios. Es por esto que surgen otras bibliotecas encima de Vue.js, como una capa más de abstracción, que brindan un conjunto de componentes reutilizables y con estilos personalizables. Entre las más populares se encuentran: Vuetify, Bootstrap Vue y Quasar [15].

Con Quasar se pueden desarrollar aplicaciones/sitios web de una sola página (SPA), aplicaciones/sitios web renderizados del lado del servidor (SSR), aplicaciones web progresivas (PWA), incluso aplicaciones móviles y de escritorio, con la misma base de código [13]. Entre las principales características que influyeron en la selección de esta biblioteca para el desarrollo en el lado del cliente se encuentran:

- más de 70 componentes web personalizables y de alto rendimiento. En la aplicación web que se implementó en este trabajo se utilizaron varias de estas componentes, que permitieron que el desarrollo de las interfaces de usuarios fuera más rápido y sencillo.
- buenas prácticas integradas tales como: compresión de código, manejo avanzado de caché, mapeo de fuentes y carga diferida. Estas buenas prácticas permiten que la aplicación web se actualice más rápido, disminuyendo el tiempo de espera para los usuarios finales.
- Quasar-CLI, una herramienta que genera de forma automática la estructura del proyecto, con los paquetes y configuraciones necesarias para el funcionamiento inicial de la aplicación web. Al crear el proyecto con esta herramienta, se ahorra tiempo de desarrollo destinado a realizar las configuraciones iniciales.
- amplia comunidad y documentación.

En este trabajo se implementó una aplicación web para la informatización de los procesos de asignación de docencia y planificación de las tesis. Para el desarrollo en el lado del servidor se utilizó Django y en el lado del cliente Quasar. Los datos se almacenaron en una base de datos SQLite.

En el próximo capítulo se describen el flujo que se debe seguir para la realización de los procesos de asignación de docencia y planificación de las tesis, así como las principales funcionalidades que se desean incorporar en el sistema durante la ejecución de estos procesos.

## Capítulo 2

# Descripción de las funcionalidades que se desean

El objetivo de este trabajo es la creación de una herramienta que permita informatizar dos de los procesos que se realizan en la Facultad de Matemática y Computación de la Universidad de La Habana (MATCOM): la asignación de docencia y la planificación de las tesis.

En este capítulo se describen las funcionalidades que se desean incorporar en el sistema para cada proceso. Para ello, se ilustra el flujo de trabajo actual para realizar la asignación de docencia y la planificación de las tesis, resaltando cómo y cuándo utilizar las funcionalidades deseadas. La sección 2.1 está destinada a la asignación de la docencia y la sección 2.2 a la planificación de las tesis.

### 2.1. Asignación de docencia

Como parte del proceso de informatización de la asignación docente, se desea incorporar las siguientes funcionalidades al sistema:

- Conocer las asignaturas que se deben impartir en un período de tiempo dado, por ejemplo enero-julio 2021.
- Conocer durante el proceso de asignación, la carga docente que se le ha asignado a cada profesor.
- Exportar la asignación de docencia a un documento con un formato preestablecido, que es el que se utiliza actualmente en la facultad.

Para ilustrar el flujo del proceso de asignación de docencia y cómo se desean utilizar las funcionalidades anteriores, se utilizará el siguiente ejemplo. Supongamos

que se desea realizar la asignación docente del departamento de Matemática Aplicada en el período enero–julio del curso 2022. A continuación se muestran las asignaturas que se deben impartir.

Asignatura	Año	Facultad	Horas
Programación y Algoritmos	M1	MATCOM	64
Seminarios de Problemas	M1	MATCOM	48
Programación y Algoritmos	M2	MATCOM	32
Inferencia Estadística	M3	MATCOM	80
Matemática Numérica	M3	MATCOM	48
Optimización Matemática I	M3	MATCOM	64
Estadística	C3	MATCOM	64
Modelos de Optimización I	C3	MATCOM	64
Estadística	G2	GEOGRAFÍA	80
Estadística	S2	SOCIOLOGÍA	80

Tabla 2.1: Asignaturas que debe impartir el departamento de Matemática Aplicada en el período enero–julio 2022.

En la tabla 2.1 se muestra los datos asociados a las asignaturas, como el nombre, el año escolar en el que se imparte, la facultad a la que pertenece el año escolar y la cantidad de horas totales.

El primer requerimiento para el sistema, es que la herramienta muestren las asignaturas que se deben impartir por un departamento en un período de tiempo dado, por lo tanto, en este caso, se deberían mostrar todas las asignaturas que aparecen en la tabla 2.1.

Para simplificar la descripción del proceso, vamos a asumir que las únicas asignaturas que se deben impartir son Optimización Matemática I, Modelos de Optimización I y Estadística (GEOGRAFÍA).

A partir de las asignaturas que se deben impartir en el semestre, se debe determinar la cantidad de grupos de conferencias, clases prácticas o cualquier otra actividad de clase, así como la distribución del total de horas a impartir en cada una de estas actividades. Esta tarea se referencia en el documento como “planificar la carga de las asignaturas”. En la tabla 2.2 se muestra la planificación de la carga de las asignaturas que se deben impartir en este ejemplo.

Asignatura	Año	Facultad	Horas	Grupos	Distribución de horas
Optimización Matemática I	M3	MATCOM	64	1/1	32/32
Modelos de Optimización I	C3	MATCOM	64	1/2	32/32
Estadística	G2	GEOGRAFÍA	80	1/2	48/32

Tabla 2.2: Carga de las asignaturas

En las columnas de la tabla 2.2 se muestran: el nombre de la asignatura, el año escolar en el cual se imparte, la facultad a la que pertenece el año escolar, la cantidad total de horas clase, la cantidad de grupos de conferencias y de clases prácticas, y la distribución de horas clase en conferencias y clases prácticas. Por ejemplo, la primera fila de la tabla indica que la asignatura Optimización Matemática I, que se imparte a los estudiantes de tercer año de la carrera Matemática, tiene un total de 64 horas clase, que se distribuyen en 32 horas de conferencias y 32 horas de clases prácticas, y que solo existe un grupo para las conferencias y para las clases prácticas.

A partir de esta información, el jefe del departamento comienza el proceso de asignación. Se desea que el sistema muestre los profesores que están disponibles para cubrir la carga de las asignaturas. En la tabla 2.3 se muestra una posible asignación docente para este ejemplo.

Asignatura	Horas	Grupos	Profesores
Optimización Matemática I	64	1/1	C: Aymeeé Marrero (32) CP: Gemayqzel Bouza(32)
Modelos de Optimización I	64	1/2	C: Aymeeé Marrero(16) C: Fernando Rodríguez (16) CP: Daniela González(32) CP: Camila Pérez(32)
Estadística	80	1/2	C: Elianys García (48) CP: Ernesto Borrego(32) CP: Elianys García(32)

Tabla 2.3: Posible asignación de la docencia.

En la primera fila de la tabla 2.3, se indica que las 32 horas de conferencia de la asignatura Optimización Matemática I, se le asignaron a la profesora Aymeeé Marrero y las 32 horas de clases prácticas a la profesora Gemayqzel Bouza.

Cada vez que el jefe de departamento realice una asignación, se desea que se actualice la carga docente que tendrá en el semestre el profesor asignado, de forma que se conozca la carga de todos los profesores durante el proceso de asignación. Para



la asignación que se muestra en la tabla 2.3, el sistema de gestión debe mostrar las siguientes cargas docentes:

- |                                |                             |
|--------------------------------|-----------------------------|
| ■ Aymeeé Marrero: 48 horas     | ■ Camila Pérez: 32 horas    |
| ■ Gemayqzel Bouza: 32 horas    | ■ Elianys García: 80 horas  |
| ■ Fernando Rodríguez: 16 horas | ■ Ernesto Borrego: 32 horas |
| ■ Daniela González: 32 horas   |                             |

En este caso se puede apreciar que la carga docente de la profesora Elianys García es mayor que la del resto de los profesores, lo cual se quisiera evitar con la implementación de esta funcionalidad.

Por otra parte, se quiere también que el sistema muestre las asignaturas o grupos que faltan por cubrir para cumplir con la planificación de la docencia. Además, una vez concluida la asignación, se desea poder exportar la información de la docencia a un documento con el formato que se utiliza actualmente en la facultad.

En la próxima sección se describen las funcionalidades que se desean incorporar para el proceso de planificación de las tesis.

## 2.2. Planificación de las tesis

El proceso de planificación de las tesis se puede descomponer en dos subprocesos principales: la confección de los tribunales y la programación de las defensas. Para la planificación de las tesis, se desea disponer de las siguientes funcionalidades.

- Cada vez que se agregue una tesis en el sistema, crear un tribunal sin definir para la misma.
- Conocer, durante el proceso de confección de los tribunales de tesis, la cantidad de tribunales en los que participa cada profesor.
- Una vez se confeccione el tribunal de una tesis, crear un horario sin definir para la defensa de la misma.
- Una vez estén definidos los horarios, se desea conocer en cuáles, un determinado profesor no está libre (por participaciones en defensas). Con el objetivo de facilitar cambios necesarios en la programación de los actos de defensa.
- Exportar la confección de los tribunales de tesis a un documento con un formato preestablecido.

- Exportar la programación de las defensas de tesis a un documento con un formato preestablecido.

Para ilustrar las funcionalidades que se desean durante el proceso de planificación de las tesis se utilizará un ejemplo. En la tabla 2.4, se muestran dos tesis.

ID	Tesis	Estudiante	Tutores
1	Simulación y optimización de movimientos de malabares	Gustavo Despaigne	Fernando Rodríguez
2	Propagación de epidemias mediante modelos basados en metapoblaciones	Abel Antonio Cruz	Angela M. León José A. Mesejo

Tabla 2.4: Ejemplo de dos tesis

Las columnas de la tabla 2.4 muestran: un identificador, el nombre de la tesis, el nombre del autor y los nombres de los tutores

El primer paso es la confección de los tribunales para cada tesis. Se debe definir qué profesor formará parte de cada tribunal y con qué rol. En este ejemplo se asume que los tribunales solo se conformarán por un oponente y un presidente. A continuación se muestran posibles tribunales para las tesis correspondientes a la tabla 2.4.

ID Tesis	Tutores	Oponente	Presidente
1	Fernando Rodríguez	Gemayqzel Bouza	Aymeeé Marrero
2	Angela M. León José A. Mesejo	Damian Valdés	Gemayqzel Bouza

Tabla 2.5: Posibles tribunales de tesis

Con el objetivo de realizar una distribución equitativa de los profesores en los tribunales de tesis, se desea que durante el proceso de confección de los tribunales, se pueda conocer la cantidad en los que participa cada profesor. En la tabla 2.6 se muestran las participaciones de los profesores en tribunales de acuerdo a la tabla 2.5

Profesor	Oponente	Presidente
Gemayqzel Bouza	1	1
Aymeeé Marrero	0	1
Damian Valdés	1	0

Tabla 2.6: Cantidad de tribunales de tesis en los que participa cada profesor

Una vez que se definan los tribunales de tesis, el próximo paso es relizar la programación de las defensas. Se debe definir la fecha, hora y local en donde se realizará el acto de defensa de cada tesis. En la tabla 2.7 se muestran posibles horarios para las defensas de las tesis que se muestran en la tabla 2.4.

ID Tesis	Fecha	Hora	Local
1	10/12/2022	2:00 PM	Aula Posgrado
2	10/12/2022	3:30 PM	Salón del Decanato

Tabla 2.7: Posible horarios para las defensas de tesis

Una vez concluido el proceso de planificación de las tesis, se desea poder exportar los datos de los tribunales y la programación de las defensas a documentos con un formato preestablecido.

Con el objetivo de evitar que una tesis se quede sin planificar, se desea que cada vez que una tesis se agregue al sistema, se genere automáticamente un tribunal sin definir y una programación para el acto de defensa, sin definir la fecha, la hora ni el local.

Además se desea conocer para cada profesor los horarios que tiene ocupado a partir de la programación de las defensas, por si es necesario realizar algún cambio en la planificación.

En este capítulo se describieron las funcionalidades que se desean incorporar en el sistema como parte de la informatización de los procesos de asignación de docencia y planificación de las tesis. En el próximo capítulo se describe como se modeló y diseñó la base de datos para representar la información que interviene en estos procesos.

## Capítulo 3

# Descripción del diseño de la base de datos

El diseño de base de datos es un proceso fundamental a la hora de modelar el conjunto de datos y las operaciones que se deseen realizar sobre ellos. Un correcto diseño de la base de datos es esencial para garantizar la consistencia de la información, eliminar datos redundantes, ejecutar consultas de manera eficiente y mejorar el rendimiento de la base de datos [16].

En este capítulo se presenta el diseño de la base de datos que se obtuvo siguiendo el enfoque relacional. En las secciones 3.1 y 3.2 se describen las modelaciones que se obtuvieron para los procesos de asignación de la docencia y planificación de las tesis, respectivamente.

### 3.1. Modelación de la asignación de docencia

Para la informatización del proceso de asignación de docencia, fue necesario almacenar toda la información que interviene en este proceso. Para ello, se modelaron las entidades fundamentales, así como las interrelaciones que se establecen entre ellas. En la figura 3.1 se muestra el esquema que se obtuvo a partir del modelo entidad-relación.

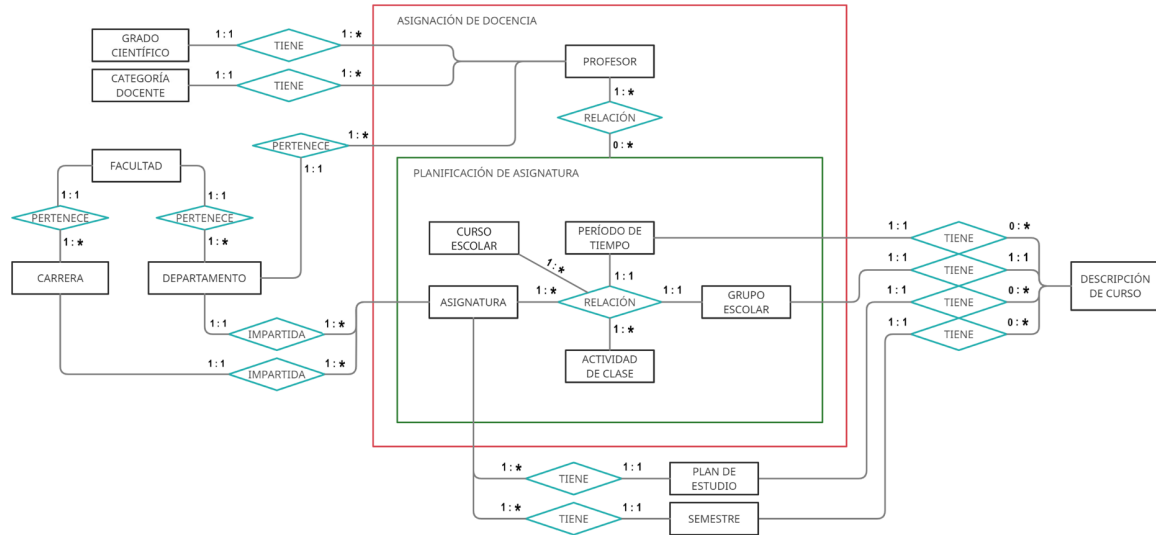


Figura 3.1: MER para la asignación de docencia

Con el objetivo de simplificar la representación del modelo entidad-relación no se agregaron en la imagen 3.1 los atributos correspondientes a cada entidad, a continuación se describen los datalles de cada entidad.

**FACULTAD:** Representa las facultades de la Universidad de La Habana. El nombre de la facultad se modela como un atributo, mientras que las carreras que se estudian en la facultad y los departamentos que pertenecen a ella, se modelan como relaciones con las entidades CARRERA y DEPARTAMENTO, respectivamente. Una facultad tiene uno o muchos departamentos, y en una facultad se pueden estudiar una o más carreras.

**CARRERA:** Representa las carreras que se estudian en la Universidad de La Habana. El nombre de la carrera se modela como un atributo, mientras que la facultad a la que pertenece la carrera y las asignaturas que se imparten en ella, se modelan como relaciones con las entidades FACULTAD y ASIGNATURA, respectivamente. Una carrera pertenece a una única facultad y en una carrera se imparten una o muchas asignaturas.

**DEPARTAMENTO:** Representa los departamentos de una facultad de la Universidad de La Habana. El nombre del departamento se modela como un atributo, mientras que los profesores que pertenecen a un departamento, las asignaturas que son atendidas por el departamento y la facultad a la que pertenece el departamento,

se modelan como relaciones con las entidades PROFESOR, FACULTAD y ASIGNATURA, respectivamente. Un departamento pertenece a una única facultad, atiende una o muchas asignaturas y a un departamento pertenecen uno o muchos profesores.

**PROFESOR:** Agrupa los datos asociados a los profesores. Los campos nombre y apellidos se modelan como atributos, mientras que la categoría docente, el grado científico de un profesor y el departamento al que pertenece, se modelan como relaciones con las entidades CATEGORÍA DOCENTE, GRADO CIENTÍFICO y DEPARTAMENTO, respectivamente. Un profesor pertenece a un único departamento y puede tener solo un grado científico y una categoría docente.

**CATEGORÍA DOCENTE:** Representa la categoría docente que tienen los profesores. El nombre de la categoría docente se modela como un atributo. Pueden existir uno o muchos profesores con una misma categoría docente.

**GRADO CIENTÍFICO:** Representa el grado científico que tienen los profesores. El nombre del grado científico se modela como un atributo. Pueden existir uno o muchos profesores con un mismo grado científico.

**ASIGNATURA:** Agrupa los datos asociados a las asignaturas. Los campos nombre de la asignatura y cantidad de horas totales a impartir, se modelan como atributos, mientras que el plan de estudio asociado a la asignatura, el semestre en el que se imparte, el departamento responsable de la asignatura y la carrera a la que pertenece, se modelan como relaciones con las entidades PLAN DE ESTUDIO, SEMESTRE, DEPARTAMENTO y CARRERA respectivamente. Una asignatura se atiende por un único departamento, pertenece a una única carrera, se imparte en un único semestre y tiene un único plan de estudio. Las asignaturas que se imparten de forma anual se deben representar como dos asignaturas independientes.

**PLAN DE ESTUDIO:** Representa el plan de estudio por el que se rige una asignatura. El nombre del plan de estudio se modela como un atributo. Pueden existir una o muchas asignaturas con el mismo plan de estudio.

**SEMESTRE:** Representa los semestres asociados a una carrera. El nombre del semestre se modela como un atributo. Pueden existir una o muchas asignaturas que se impartan en el mismo semestre.

[NO NECESITO ESTO CREO]

**PERÍODO DE TIEMPO:** Representa los períodos de tiempo del año, por ejemplo, enero–julio. El nombre del período de tiempo se modela como un atributo. Pueden existir una o muchas asignaturas que se impartan en el mismo período de tiempo.

**CURSO ESCOLAR:** Representa los curso escolares, por ejemplo: 2020-2021. El nombre del curso de escolar se modela como un atributo. Además cuenta con un campo booleano que indica si el curso es el vigente.

**AÑO ESCOLAR:** Representa los años académicos asociados a una carrera, como por ejemplo: Matemática primer año (M1) o Computación tercer año (C3). El nombre del año escolar se modela como un atributo.

**ACTIVIDAD DE CLASE:** Representa los tipos de actividades que se imparten en una asignatura, tales como: conferencias, clases prácticas, seminarios, laboratorios, otros.

**PLANIFICACIÓN DE ASIGNATURA:** Se crea con el objetivo de modelar la separación de una asignatura según las actividades de clases a impartir, en un curso escolar, en un período de tiempo y para un grupo escolar específico. Está compuesta por la agregación de las entidades ASIGNATURA, AÑO ESCOLAR, ACTIVIDAD DE CLASE, GRUPO ESCOLAR Y PERÍODO DE TIEMPO. Se agregan los atributos cantidad de horas para el tipo de actividad a realizar y la cantidad de grupos. Por ejemplo, para la asignatura Optimización Matemática I, que se imparte en el tercer año de la carrera de Matemática, con un total de 64 horas clase, se deben crear dos instancia de PLANIFICACIÓN DE ASIGNATURA, una con 32 horas de conferencias y otra con 32 horas de clases prácticas, para que posteriormente se puedan asignar los profesores que se ocuparán de estas planificaciones. [REVISAR LA CANT DE GRUPOS Y EL AÑO ESCOLAR]

**ASIGNACIÓN DE DOCENCIA:** Representa las asignaciones de profesores a planificaciones de asignaturas. Además se agrega un atributo porcentaje que indica el porcentaje del total de horas a impartir, que asume el profesor asignado. Un profesor puede tener asignada cero o muchas planificaciones de asignaturas y una planificación de asignatura puede estar asignada a uno o muchos profesores. [Hay que añadir el año escolar o la tabla de carmen]

**DESCRIPCIÓN DE CURSO:** Representa el grupo escolar vigente en el curso actual. Está compuesta por la agregación de las entidades GRUPO ESCOLAR,

PERÍODO DE TIEMPO, PLAN DE ESTUDIO y SEMESTRE. Por ejemplo, una DESCRIPCIÓN DE CURSO puede ser que el grupo escolar C4 (Computación de cuarto año), con plan de estudio E, se encuentra en el semestre 8, en el período de tiempo septiembre-diciembre.

### 3.2. Modelación de los tribunales de tesis

Se modelaron las entidades fundamentales que intervienen en el proceso de planificación de las tesis, así como las interrelaciones que se establecen entre ellas. En la figura 3.2 se muestra el esquema que se obtuvo a partir del modelo entidad-relación extendido.

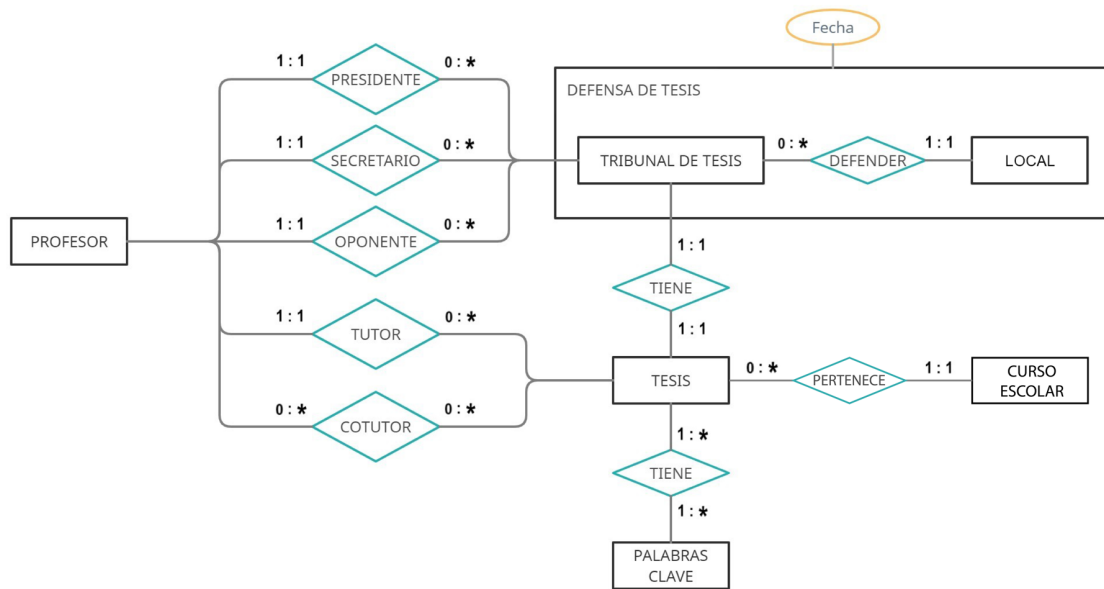


Figura 3.2: MER para la planificación de las tesis

**TESIS:** Agrupa los datos asociados a una tesis. El título de la tesis y el autor se modelan como atributos, mientras que el tutor, los cotutores, curso escolar y las palabras claves se modelan como relaciones con las entidades PROFESOR (tutor y cotutores), CURSO ESCOLAR y PALABRAS CLAVES. Una tesis tiene un único tutor, puede tener cero o muchos cotutores, pertenece a un único curso escolar y tiene una o muchas palabras claves.

**PALABRAS CLAVES:** Representa las palabras claves que describen el contenido de una tesis. El nombre o texto de las palabras claves se modela como un



atributo. Una palabra clave puede aparecer en una o muchas tesis.

**TRIBUNAL DE TESIS:** Representa el tribunal de la defensa de una tesis. Está compuesto por los miembros del tribunal: presidente, secretario y oponente, y la tesis, campos que se modelan como relaciones con las entidades PROFESOR y TESIS, respectivamente. Un tribunal de tesis tiene una única tesis, un único presidente, un único secretario y un único oponente.

**DEFENSA DE TESIS:** Representa el acto de defensa de la tesis. Está compuesta por un tribunal de tesis, en un local y una fecha determinada. La fecha se modela como un atributo mientras que el tribunal y el local se modelan como relaciones con las entidades TRIBUNAL DE TESIS y LOCAL, respectivamente. Una defensa de tesis tiene un único tribunal de tesis, una única fecha y un único local.

**LOCAL:** Representa el local donde se lleva a cabo la defensa de las tesis. El nombre del local se modela como un atributo. En un local se puede realizar la defensa de cero o muchas tesis.

**PROFESOR:** Se utiliza la misma entidad creada en el proceso de asignación de docencia. Un profesor puede tener el rol de presidente, secretario u oponente en cero o muchos tribunales de tesis. Un profesor puede ser tutor o cotutor de cero o muchas tesis.

En el próximo capítulo se presenta el sistema que se implementó en el trabajo para la informatización de los procesos de asignación de docencia y planificación de las tesis, así como las principales funcionalidades que se incorporaron para facilitar la realización la planificación de estos procesos.

## Capítulo 4

# Descripción las herramientas implementadas

El objetivo de este trabajo es la informatización de los procesos de asignación de docencia y planificación de las tesis que se llevan a cabo en la Facultad de Matemática y Computación de la Universidad de La Habana (MATCOM).

En este capítulo se presenta la aplicación web que se implementó como propuesta de solución para la informatización de estos procesos. En las secciones 4.1 y 4.2 se describen los pasos que se deben seguir para realizar a través del sistema, la asignación de docencia y la planificación de las tesis, respectivamente. En la sección 4.3 se describe una funcionalidad que se implementó en el lado del servidor para salvar el estado de la base datos en documentos CSV y poblar la base de datos a partir de los mismos.

En la figura 4.1 se ilustra la vista inicial de la aplicación web que se implementó. En la parte superior se encuentra una barra de navegación, desde la cual se puede acceder a todas las vistas de la aplicación. Se crearon dos paneles principales, uno para la administración de los departamentos y otro para la planificación de las tesis. En el panel para la administración de los departamentos se muestran los cuatro departamentos de la facultad MATCOM, con cuatro botones cada uno que llevan a las vistas para administrar los profesores, las asignaturas, las cargas de las asignaturas y las asignaciones de docencia correspondientes al departamento. En el panel para la planificación de las tesis se agregó un selector para definir qué curso escolar se desea planificar y tres botones que llevan a las vistas para administrar las tesis, los tribunales y las defensas, correspondientes al curso escolar seleccionado.

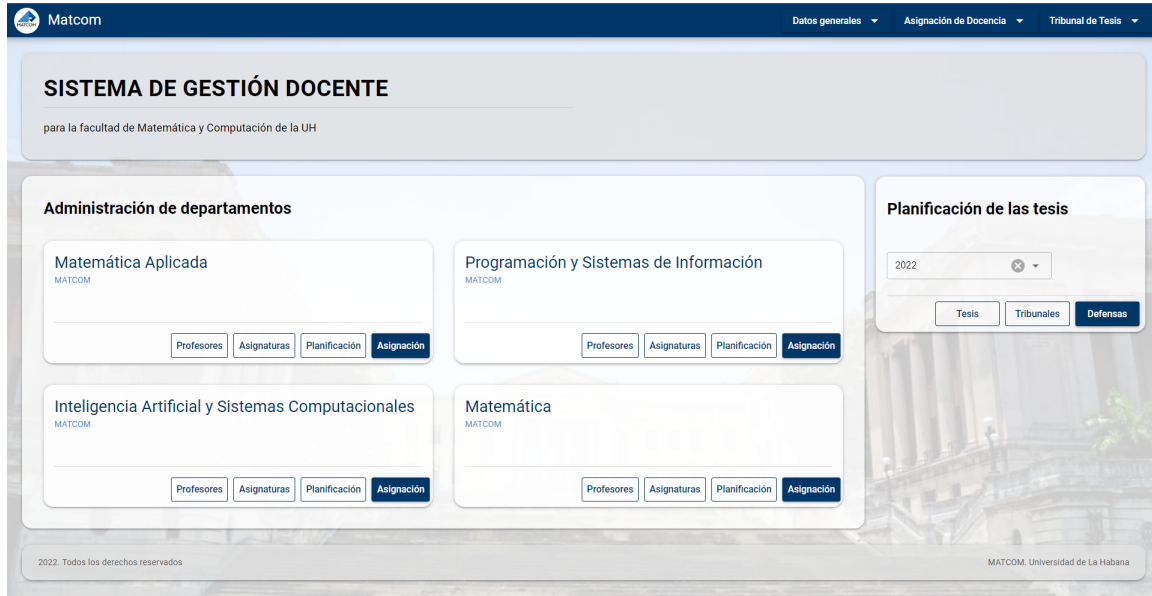


Figura 4.1: Vista inicial del sistema de gestión

## 4.1. Asignación de docencia

En esta sección se describen los pasos que se deben seguir para realizar el proceso de asignación de docencia en un departamento a través del sistema de gestión que se propone en este trabajo. Para la ilustrar este proceso se utilizará el mismo fragmento de la docencia correspondiente al departamento de Matemática Aplicada en el período de tiempo enero–julio del curso 2022, que se describe en la sección 2.1. En la tabla 4.1 se muestra la carga de las asignaturas que se deben impartir y en la 4.1 la distribución de los profesores para cubrir con la docencia.

Asignatura	Año	Facultad	Horas	Grupos	Distribución de horas
Optimización Matemática I	M3	MATCOM	64	1/1	32/32
Modelos de Optimización I	C3	MATCOM	64	1/2	32/32
Estadística	G2	GEOGRAFÍA	80	1/2	48/32

Tabla 4.1: Carga de las asignaturas

Asignatura	Horas	Grupos	Profesores
Optimización Matemática I	64	1/1	C: Aymeeé Marrero (32) CP: Gemayqzel Bouza(32)
Modelos de Optimización I	64	1/2	C: Aymeeé Marrero(16) C: Fernando Rodríguez (16) CP: Daniela González(32) CP: Camila Pérez(32)
Estadística	80	1/2	C: Elianys García (48) CP: Ernesto Borrego(32) CP: Elianys García(32)

Tabla 4.2: Fragmento de la asignación de docencia.

El primer paso que se debe realizar es planificar la carga de las asignaturas que se deben impartir, la interfaz de usuario que se implementó en el sistema para realizar este proceso se muestra en la figura 4.2.

Figura 4.2: Vista de la planificación de las cargas de las asignaturas

En la figura 4.2, se indican con recuadros tres regiones cuyo significado se explica a continuación.

- A: botón para agregar una nueva carga de asignatura.
- B: filtros por: Año, Actividad de clase, Curso escolar, Período de tiempo.
- C: barra de búsqueda por texto para el nombre de las asignaturas.

Para agregar una nueva carga de asignatura se debe pulsar en el botón que se indica en el recuadro A y completar los campos del formulario. En la figura 4.3 se muestra como agregar la planificación correspondiente a las conferencias de la asignatura Optimización Matemática I para los estudiantes de tercer año de la carrera Matemática.

Departamento: Matemática Aplicada

**Planificación de las Asignaturas**

Año:  Actividad de...:  Curso Escolar:  Período de T...:  Asignatura:

**Creando Planificación**

Asignatura: Optimización Matemática I -- plan E Año: M3

Actividad de Clase: Conferencia Total de grupos: 1

Horas: 32 Curso Escolar: 2022

Período de Tiempo: enero - julio

Figura 4.3: Vista del formulario para crear una carga de asignaturas

La figura 4.4 muestra el resultado de agregar las planificaciones de las asignaturas correspondientes a la tabla 4.1.

Departamento: Matemática Aplicada

**Planificación de las Asignaturas**

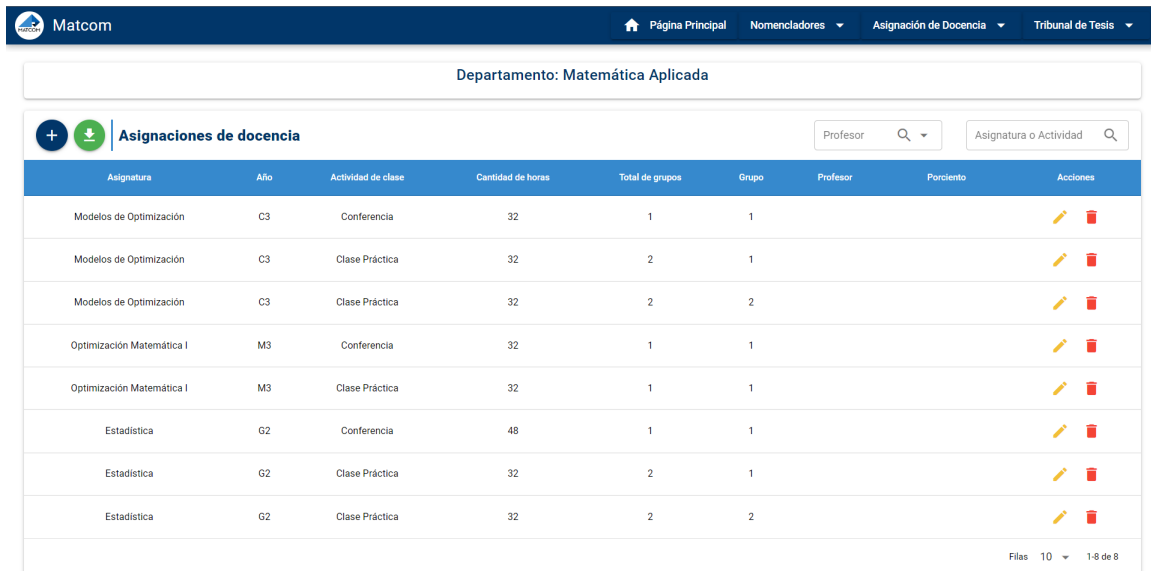
Año:  Actividad de...:  Curso Escolar:  Período de T...:  Asignatura:

Asignatura	Año	Actividad de Clase	Total de grupos	Horas	Curso Escolar	Período de Tiempo	Acciones
Estadística	G2	Conferencia	1	48	2022	enero - julio	
Estadística	G2	Clase Práctica	2	32	2022	enero - julio	
Modelos de Optimización	C3	Conferencia	1	32	2022	enero - julio	
Modelos de Optimización	C3	Clase Práctica	2	32	2022	enero - julio	
Optimización Matemática I	M3	Conferencia	1	32	2022	enero - julio	
Optimización Matemática I	M3	Clase Práctica	1	32	2022	enero - julio	

Filas 10 1-6 de 6

Figura 4.4: Vista de la carga de las asignaturas

Por cada carga de asignatura que se agregue en el sistema se crean  $n$  asignaciones de docencia sin asignar, donde  $n$  es la cantidad de grupos que tiene la carga de la asignatura. En la figura 4.5 se muestran las asignaciones de docencias creadas automáticamente por el sistema a partir de las cargas agregadas en la figura 4.4.



















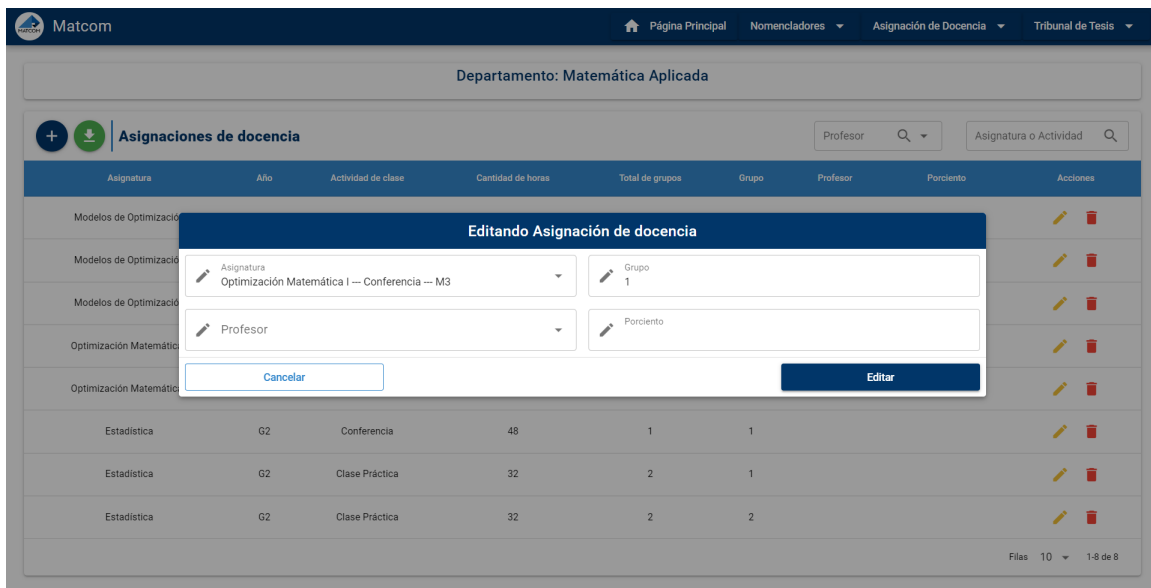

Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Modelos de Optimización	C3	Conferencia	32	1	1			 
Modelos de Optimización	C3	Clase Práctica	32	2	1			 
Modelos de Optimización	C3	Clase Práctica	32	2	2			 
Optimización Matemática I	M3	Conferencia	32	1	1			 
Optimización Matemática I	M3	Clase Práctica	32	1	1			 
Estadística	G2	Conferencia	48	1	1			 
Estadística	G2	Clase Práctica	32	2	1			 
Estadística	G2	Clase Práctica	32	2	2			 


Figura 4.5: Vista de la asignación de docencia


El próximo paso es realizar la asignación de los profesores para cubrir la docencia del semestre. Para asignar un profesor se debe pulsar sobre el botón de editar de una fila. En la figura 4.6 se muestra el resultado de pulsar sobre el botón de editar de la cuarta fila, correspondiente a las conferencias de la asignatura de Optimización Matemática I.




### Editando Asignación de docencia


Asignatura
Optimización Matemática I — Conferencia — M3


Grupo
1


Profesor


Porcentaje

Cancelar

Editar

Figura 4.6: Vista del formulario para realizar una asignación de docencia

Se deben completar los campos del formulario y pulsar sobre el botón editar. Como se muestra en la tabla 4.2, la profesora Aymeeé Marrero es la encargada de impartir las 32 horas de conferencia de la asignatura Optimización Matemática I, por tanto se debe especificar que la profesora imparte el 100 por ciento de las horas como se muestra en la figura 4.7.

The screenshot shows the 'Matcom' web application interface. A modal form titled 'Asignación de Docencia' is open, allowing for the editing of a teaching assignment. The form contains two input fields: 'Grupo' with the value '1' and 'Porcentaje' with the value '100'. A dropdown menu for 'Profesor' is open, displaying a list of names: Alejandro Mesejo Chiong, Alina Ruiz Johnes, Angela León Mecías, Aymée Marrero Severo (highlighted), Camila Pérez Mosquera, Carlos Bouza Herrera, Damian Valdés Santiago, and Daniela González Beltrán. Below the list is a 'Cancelar' button. To the right of the form is an 'Editar' button. In the background, a table lists assignments with columns: Asignatura, Grupo, Tipo, Horas, Grupo, Profesor, and Porcentaje. The table shows three rows of data for 'Estadística' and 'Optimización Matemática I'.

Asignatura	Grupo	Tipo	Horas	Grupo	Profesor	Porcentaje
Estadística	G2	Conferencia	48	1	1	
Estadística	G2	Clase Práctica	32	2	1	
Estadística	G2	Clase Práctica	32	2	2	

Figura 4.7: Vista del formulario con los datos ingresados de una asignación de docencia

En la figura 4.8 se muestra el resultado de realizar la asignación.

Departamento: Matemática Aplicada

Aymée Marrero Severo 32h

**Asignaciones de docencia**

Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Modelos de Optimización	C3	Conferencia	32	1	1			
Modelos de Optimización	C3	Clase Práctica	32	2	1			
Modelos de Optimización	C3	Clase Práctica	32	2	2			
B Optimización Matemática I	M3	Conferencia	32	1	1	Aymée Marrero Severo	100%	
Optimización Matemática I	M3	Clase Práctica	32	1	1			
Estadística	G2	Conferencia	48	1	1			
Estadística	G2	Clase Práctica	32	2	1			
Estadística	G2	Clase Práctica	32	2	2			

Se editó una instancia de 'Asignación de docencia' correctamente

Filas 10 1-8 de 8

Figura 4.8: Vista resultante de realizar una asignación de docencia correctamente

Cada vez que se realicen cambios en la tabla, ya sean por agregar, editar o eliminar asignaciones de docencia, la carga docente de los profesores se actualiza como se indica en el recuadro A de la figura 4.8. El recuadro B refleja como quedó la fila tras la asignación de la profesora Aymée Marrero para que imparta las 32 horas de conferencia de la asignatura Optimización Matemática I a los estudiantes que cursan el tercer año de la carrera Matemática.

En la figura 4.9 se muestra el resultado de cubrir la docencia de las asignaturas de Optimización Matemática I, Estadística y las clases prácticas de Modelos de Optimización, de acuerdo a la tabla 4.2.



The screenshot shows the Matcom web application interface. At the top, there is a navigation bar with the Matcom logo and links to 'Página Principal', 'Nomencladores', 'Asignación de Docencia', and 'Tribunal de Tesis'. Below the navigation bar, the 'Departamento: Matemática Aplicada' is displayed. A row of buttons shows the names of professors and their associated hours: Aymée Marrero Severo (32h), Gemayzel Bouza Allende (32h), Camila Pérez Mosquera (32h), Daniela González Beltrán (32h), Elyanis García Pola (80h), and Ernesto Borrego Rodríguez (32h). The main section is titled 'Asignaciones de docencia' and features a table with the following columns: Asignatura, Año, Actividad de clase, Cantidad de horas, Total de grupos, Grupo, Profesor, Porcentaje, and Acciones. The table contains 8 rows of data. At the bottom right, there is a pagination control showing 'Filas 10' and '1-8 de 8'.

















Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Modelos de Optimización	C3	Conferencia	32	1	1			 
Modelos de Optimización	C3	Clase Práctica	32	2	1	Daniela González Beltrán	100%	 
Modelos de Optimización	C3	Clase Práctica	32	2	2	Camila Pérez Mosquera	100%	 
Optimización Matemática I	M3	Conferencia	32	1	1	Aymée Marrero Severo	100%	 
Optimización Matemática I	M3	Clase Práctica	32	1	1	Gemayzel Bouza Allende	100%	 
Estadística	G2	Conferencia	48	1	1	Elyanis García Pola	100%	 
Estadística	G2	Clase Práctica	32	2	1	Ernesto Borrego Rodríguez	100%	 
Estadística	G2	Clase Práctica	32	2	2	Elyanis García Pola	100%	 

Figura 4.9: Vista con varias asignaciones de docencia

Las conferencias de la asignatura Modelos de Optimización I, se imparten por los profesores Aymée Marrero y Fernando Rodríguez, con 16 horas cada uno. Para modelar esta distribución en el sistema, es necesario agregar una nueva asignación de docencia y especificar que cada profesor va a impartir el 50 por ciento de las horas totales de conferencia. Para agregar una nueva asignación se debe pulsar sobre el botón de agregar y llenar los campos necesarios. En la figura 4.10, se muestra como queda la asignación final de la docencia.

The screenshot shows the Matcom web application interface. At the top, there's a navigation bar with 'Matcom' logo and links to 'Página Principal', 'Nomencladores', 'Asignación de Docencia', and 'Tribunal de Tesis'. Below this, a header for 'Departamento: Matemática Aplicada' lists several professors with their respective hour counts: Aymée Marrero Severo (48h), Gemayzel Bouza Allende (32h), Camila Pérez Mosquera (32h), Fernando Raul Rodríguez Flores (16h), Daniela González Beltrán (32h), Elyanis García Pola (80h), and Ernesto Borrego Rodríguez (32h).

The main section is titled 'Asignaciones de docencia' and contains a table with the following columns: Asignatura, Año, Actividad de clase, Cantidad de horas, Total de grupos, Grupo, Profesor, Porcentaje, and Acciones. The table lists 9 teaching assignments for 'Modelos de Optimización' and 'Estadística' across different years and activities.

Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Modelos de Optimización	C3	Conferencia	32	1	1	Fernando Raul Rodríguez Flores	50%	
Modelos de Optimización	C3	Conferencia	32	1	1	Aymée Marrero Severo	50%	
Modelos de Optimización	C3	Clase Práctica	32	2	1	Daniela González Beltrán	100%	
Modelos de Optimización	C3	Clase Práctica	32	2	2	Camila Pérez Mosquera	100%	
Optimización Matemática I	M3	Conferencia	32	1	1	Aymée Marrero Severo	100%	
Optimización Matemática I	M3	Clase Práctica	32	1	1	Gemayzel Bouza Allende	100%	
Estadística	G2	Conferencia	48	1	1	Elyanis García Pola	100%	
Estadística	G2	Clase Práctica	32	2	1	Ernesto Borrego Rodríguez	100%	
Estadística	G2	Clase Práctica	32	2	2	Elyanis García Pola	100%	

At the bottom right of the table, there's a pagination control showing 'Filas 10' and '1-9 de 9'.

Figura 4.10: Vista resultado de la asignaciones de docencia

Para poder llevar a cabo el proceso de asignación de docencia a través del sistema que se propone, es necesario que en la base de datos se ingresen las informaciones que intervienen en este proceso. Por ejemplo, para realizar una asignación de docencia se necesitan los profesores del departamento y las planificaciones de las cargas de las asignaturas. Para poder crear las cargas se necesitan las asignaturas, los años escolares, las actividades de clase y los períodos de tiempo. Por tanto el primer paso para realizar la asignación de docencia es ingresar todos los datos que se describen en la sección 3.1.

En la siguiente sección se describen los pasos para realizar la confección de los tribunales y la planificación de las defensas de tesis.

## 4.2. Planificación de las tesis

La planificación de las tesis consta de dos subprocesos principales: la confección de los tribunales y la programación de las defensas. En esta sección se describen los pasos a seguir para completar la planificación de las tesis a través del sistema. Para ilustrar este proceso se utilizará el mismo ejemplo que se describe en la sección 2.2. En la tabla 4.3 se muestran los datos de dos tesis, en la tabla 2.5 el tribunal seleccionado y en la tabla 4.5 el horario escogido para la defensa de las mismas.

ID	Tesis	Estudiante	Tutores
1	Simulación y optimización de movimientos de malabares	Gustavo Despaigne	Fernando Rodríguez
2	Propagación de epidemias mediante modelos basados en metapoblaciones	Abel Antonio Cruz	Angela M. León José A. Mesejo

Tabla 4.3: Datos de las tesis

ID Tesis	Tutores	Oponente	Presidente
1	Fernando Rodríguez	Gemayqzel Bouza	Aymeeé Marrero
2	Angela M. León José A. Mesejo	Damian Valdés	Gemayqzel Bouza

Tabla 4.4: Posibles tribunales para las tesis tesis

ID Tesis	Fecha	Hora	Local
1	10/12/2022	2:00 PM	Aula Posgrado
2	10/12/2022	3:30 PM	Salón del Decanato

Tabla 4.5: Posible horarios para los actos de defensa

El primer paso es ingresar las tesis en el sistema, en la figura 4.11 se muestra la interfaz de usuario que permite realizar esta tarea.

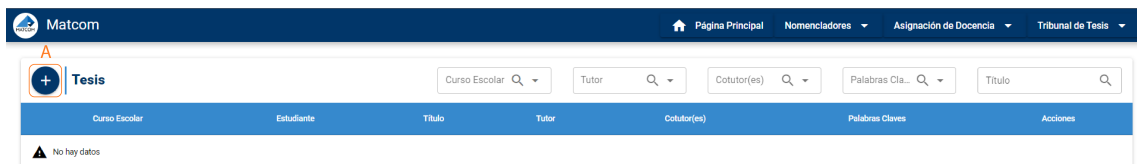


Figura 4.11: Vista de la tesis.

Para agregar una nueva tesis en el sistema, se debe pulsar sobre el botón de agregar que se indica en el recuadro A de la figura 4.11. Luego se deben llenar los campos necesarios relacionados con las tesis. En la figura 4.12 se muestra el resultado de pulsar sobre el botón de agregar y llenar los campos correspondientes a la primera tesis que se muestra en la tabla 4.3.

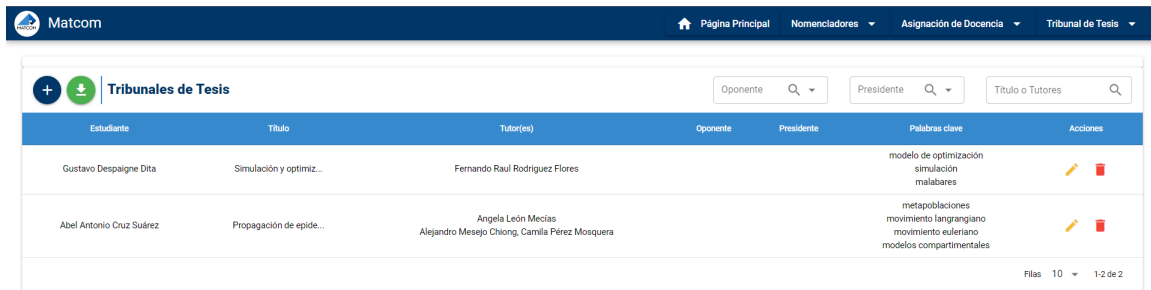
Figura 4.12: Vista del formulario para agregar una tesis.

Repitiendo el procedimiento anterior se puede agregar la segunda tesis que se muestra en la tabla 4.3. En la figura 4.13 tal se muestra el resultado de agregar las dos tesis en el sistema.

Curso Escolar	Estudiante	Título	Tutor	Cotutor(es)	Palabras Claves	Acciones
2022	Gustavo Despaigne Dita	Simulación y optimización de movimientos de malabares	Fernando Raul Rodriguez Flores		modelo de optimización simulación malabares	
2022	Abel Antonio Cruz Suárez	Propagación de epidemia mediante modelos basados en metapoblaciones	Angela León Mecías	Alejandro Mesejo Chiong Camila Pérez Mosquera	metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	

Figura 4.13: Vista de la interfaz de usuario tras agregar las dos tesis del ejemplo.

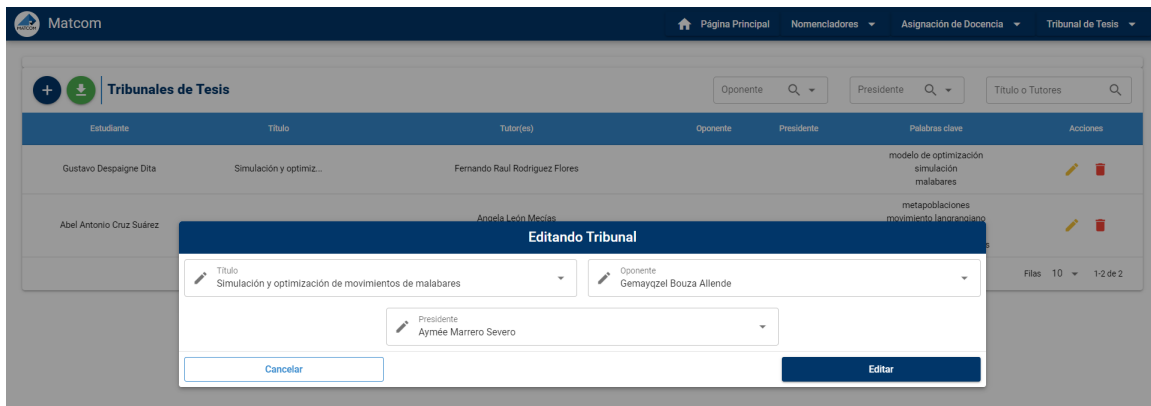
Cada vez que se agregue una tesis al sistema, se crea una instancia de un tribunal de tesis con los campos de oponente y presidente sin asignar. En la figura 4.14 se muestran los tribunales incompletos creados automáticamente a partir de las tesis que se ingresaron en el sistema.



Estudiante	Título	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
Gustavo Despaigne Dita	Simulación y optimiz...	Fernando Raul Rodriguez Flores			modelo de optimización simulación malabares	
Abel Antonio Cruz Suárez	Propagación de epide...	Angela León Mecías Alejandro Mesejo Chiong, Camila Pérez Mosquera			metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	

Figura 4.14: Vista de la interfaz de usuario para la confección de los tribunales.

El próximo paso es asignar los profesores que conformarán los tribunales de tesis. El encargado de realizar este proceso debe pulsar sobre el botón de editar de la fila correspondiente al tribunal que desea definir. En la figura 4.15 se muestra el resultado de pulsar sobre el botón de la primera fila y completar el tribunal con los profesores que se muestran en la tabla 4.4.



**Editando Tribunal**

Título:

Oponente:

Presidente:

Figura 4.15: Vista del formulario para la confección de los tribunales.

Cada vez que un tribunal de tesis se modifique se actualiza la participación de los profesores en tribunales, para cada profesor se muestra la cantidad de tribunales en los que participa como oponente y como presidente. En la figura 4.16 se muestra el resultado de confeccionar el tribunal para la tesis del estudiante Gustavo Despaigne Dita.

Matcom

Página Principal Nomencladores Asignación de Docencia Tribunal de Tesis

**Tribunales de Tesis**

Oponente Presidente Título o Tutores

Estudiante	Título	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
Gustavo Despaigne Dita	Simulación y optimiz...	Fernando Raul Rodríguez Flores	Gemayqzel Bouza Allende	Aymée Marrero Severo	modelos de optimización simulación malabares	
Abel Antonio Cruz Suárez	Propagación de epide...	Angela León Mecías Alejandro Mesejo Chiong, Camila Pérez Mosquera			metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	

Filas 10 1-2 de 2

Figura 4.16: Vista resultante de confeccionar el tribunal de la primera tesis de la tabla.

En el recuadro A que se indica en la figura 4.16, se muestran las participaciones de los profesores en los tribunales que se hayan asignado. En este ejemplo, la profesora Aymeeé Marrero participa en 0 tribunales como oponente y en 1 tribunal como presidente y la profesora Gemayqzel Bouza participa en 1 tribunal como oponente y en 0 tribunales como presidente. El recuadro B muestra una descripción que aparece al colocar el cursor sobre uno de los profesores, en este caso sobre el recuadro que representa la cantidad de tribunales en los que participa la profesora Gemayqzel. La figura 4.17 muestra el resultado de confeccionar el tribunal para la tesis del estudiante Abel Antonio Cruz Suárez.

Matcom

Página Principal Nomencladores Asignación de Docencia Tribunal de Tesis

**Tribunales de Tesis**

Oponente Presidente

Oponente Presidente Título o Tutores

Estudiante	Título	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
Gustavo Despaigne Dita	Simulación y optimiz...	Fernando Raul Rodríguez Flores	Gemayqzel Bouza Allende	Aymée Marrero Severo	modelo de optimización simulación malabares	
Abel Antonio Cruz Suárez	Propagación de epide...	Angela León Mecías Alejandro Mesejo Chiong, Camila Pérez Mosquera	Damian Valdés Santiago	Gemayqzel Bouza Allende	metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	

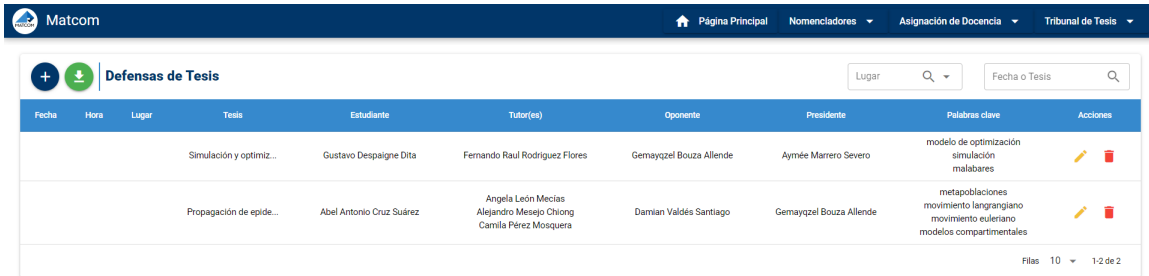
Filas 10 1-2 de 2





Figura 4.17: Vista del formulario para la confección de los tribunales.

En la figura 4.17 se puede apreciar que se actualizaron las participaciones de los profesores en los tribunales. Se agregó la participación del profesor Damian Valdés y se actualizó la de la profesora Gemayqzel Bouza, aumentando en 1 la cantidad de participaciones en tribunales como presidente.

Cuando se hayan confeccionado los tribunales de tesis, el siguiente paso es realizar la programación de las defensas. Cuando una tesis se agrega al sistema, se crea una instancia de una defensa de tesis sin definir los campos fecha, hora y local. En la figura

4.18 se muestra la interfaz de usuario donde se realiza el proceso de programación de las defensas.

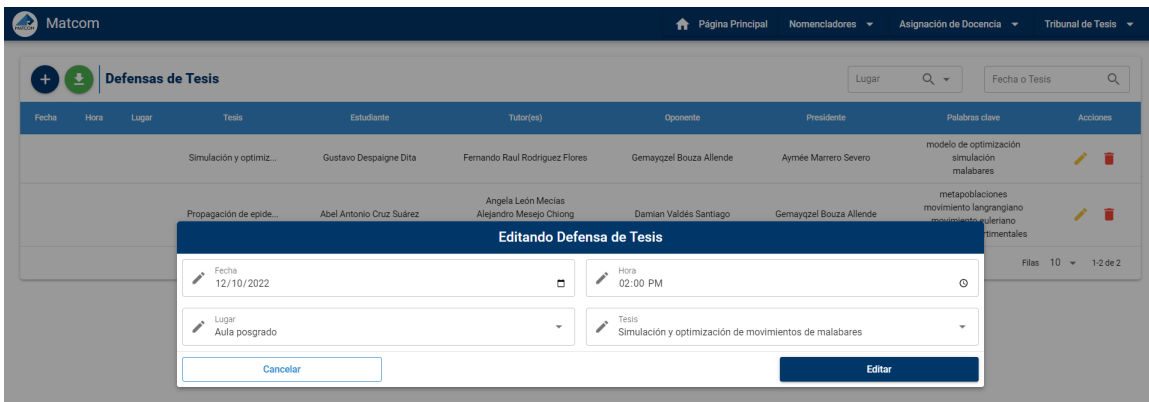


Fecha	Hora	Lugar	Tesis	Estudiante	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
			Simulación y optimiz...	Gustavo Despaigne Dita	Fernando Raul Rodriguez Flores	Gemayqzel Bouza Allende	Aymée Marrero Severo	modelo de optimización simulación malabares	 
			Propagación de epide...	Abel Antonio Cruz Suárez	Angela León Mecías Alejandro Mesejo Chiong Camila Pérez Mosquera	Damian Valdés Santiago	Gemayqzel Bouza Allende	metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	 

Filas 10 1-2 de 2

Figura 4.18: Interfaz de usuario para programar las defensas de tesis

Similar al proceso de confección de los tribunales, para definir la fecha, hora y lugar donde se realizará la defensa de una tesis, se debe pulsar sobre el botón de editar correspondiente a la fila de la defensa que se desea planificar. En la figura 4.19 se muestra el resultado de pulsar sobre el botón de editar de la primera fila y completar los datos de la planificación correspondiente a la tabla 4.5.



**Editando Defensa de Tesis**

Fecha: 12/10/2022

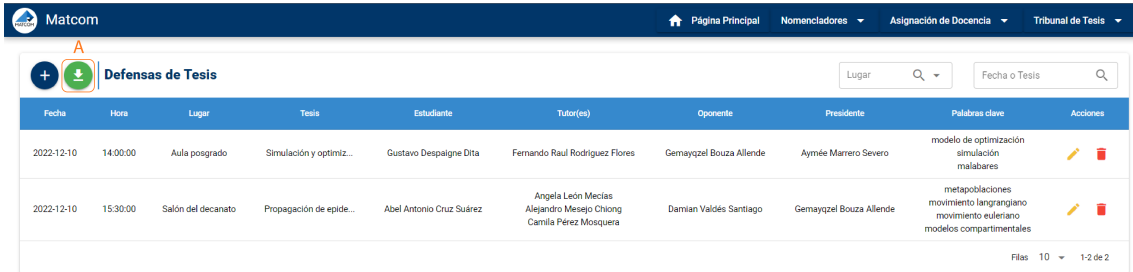
Hora: 02:00 PM

Lugar: Aula posgrado

Tesis: Simulación y optimización de movimientos de malabares

Figura 4.19: Vista del formulario para la planificación de las tesis.

Una vez que se hayan programado las defensas de las tesis, se puede exportar la información de estas a un archivo CSV pulsando sobre el recuadro A que se muestra en la figura 4.20.







Fecha	Hora	Lugar	Tesis	Estudiante	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
2022-12-10	14:00:00	Aula posgrado	Simulación y optimiz...	Gustavo Despaigne Dita	Fernando Raul Rodriguez Flores	Gemayqzel Bouza Allende	Aymée Marrero Severo	modelo de optimización simulación malabares	 
2022-12-10	15:30:00	Salón del decanato	Propagación de epide...	Abel Antonio Cruz Suárez	Angela León Mecías Alejandro Mesejo Chiong Camila Pérez Mosquera	Damian Valdés Santiago	Gemayqzel Bouza Allende	metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	 

Figura 4.20: Vista resultado de las planificaciones de tesis.

Para poder llevar a cabo la planificación de las tesis a través del sistema que se propone, es necesario que en la base de datos se ingresen las informaciones que intervienen en este proceso. Por ejemplo para poder confeccionar un tribunal es necesario que la tesis haya sido agregada anteriormente, pero para poder agregar la tesis es necesario que los tutores y cotutores se encuentren en el sistema. Por tanto el primer paso para realizar la planificación de las tesis, es ingresar todos los datos que se describen en la sección 3.2

En la siguiente sección se describen otras funcionalidades implementadas con el objetivo de agregar utilidad al sistema que se propone.

### 4.3. Exportar e importar datos de la base de datos

En el servidor se implementó un módulo que permite salvar el estado de la base de datos en documentos csv y poblar la base de datos a partir de los mismos. Se crearon dos comandos para ejecutar en la terminal con el fin de realizar las tareas mencionadas previamente:

#### 4.3.1. Exportar los datos de la base de datos

Se implemetó el comando `save_database` que permite exportar una o todas las tablas de la base de datos. La información de cada tabla se almacena en un archivo CSV. Para exportar la información de una única tabla se debe especificar su nombre precedido del argumento `-m`. Por ejemplo para salvar la tabla que contiene la información de los profesores, se debe ejecutar el siguiente comando.

```
python manage.py save_database -m Professors
```

Por otra parte, si lo que se desea es exportar la información de todas las tablas de la base de datos a documentos CSV, se debe ejecutar el siguiente comando.

```
python manage.py save_database
```



### 4.3.2. Importar datos a la base de datos

Se implementó el comando `fill_database` que permite poblar la base de datos a partir de los archivos CSV generados con el comando `save_database`. Con este comando se puede poblar una o todas las tablas de la base de datos. Para poblar la información de una única tabla se debe especificar su nombre precedido del argumento `-m`. Por ejemplo para llenar la tabla que contiene la información de las asignaturas, se debe ejecutar el siguiente comando.

```
python manage.py fill_database -m Subjects
```

Por otra parte, si lo que se desea es importar la información de todos los archivos CSV para la base de datos, se debe ejecutar el siguiente comando.

```
python manage.py fill_database
```

Los posibles nombres de entidades a utilizar con el parámetro `-m` tanto para el comando `save_database` como `fill_database` son: `ClassTypes`, `Faculties`, `ScientificDegrees`, `TeachingCategories`, `Semesters`, `TeachingGroups`, `TimePeriods`, `ScholarYears`, `Careers`, `StudyPlans`, `CarmenTable`, `Departments`, `Subjects`, `Professors`, `SubjectDescriptions`, `TeachingAssignments`, `Places`, `Keywords`, `Thesis`, `ThesisCommittee`.

En este capítulo se describieron los pasos a seguir para realizar la asignación de docencia y la planificación de las tesis a través del sistema que se propone. Se ilustraron las principales funcionalidades que se deseaban incorporar como: conocer la carga docente de los profesores durante el proceso de asignación, conocer la cantidad de participaciones de los profesores en los tribunales y la exportación de la información de estos procesos a un documento CSV.

En el próximo capítulo se describe la estructura del proyecto y algunos detalles de implementación que pueden servir para incorporar nuevos procesos en el sistema de gestión implementado.

# Capítulo 5

## Extensibilidad

Uno de los objetivos de este trabajo es la creación de una herramienta que permita la posterior integración de procesos de administración y planificación que se llevan a cabo en la Facultad de Matemática y Computación de la Universidad de La Habana. En este capítulo se presenta la estructura del proyecto, se explican algunos detalles de implementación y se propone una guía de cómo agregar nuevos procesos al sistema.

En la sección 5.1 se describen las estructuras de los proyectos cliente y servidor, así como algunas de las herramientas implementadas que se pueden reutilizar en otros procesos que se deseen agregar al sistema. En la sección 5.2 se proponen una serie de pasos a seguir para incorporar nuevos procesos en el sistema.

### 5.1. Estructura del cliente y servidor

El desarrollo en el lado del servidor se llevó a cabo con el uso de la biblioteca Django, en particular Django Rest Framework. Para el desarrollo de los procesos de asignación de docencia y planificaciones de las tesis se crearon dos *apps* independientes de Django, con el objetivo de agrupar los ficheros necesarios para la modelación de cada proceso. En la figura 5.1 se muestra cómo está estructurado el servidor.

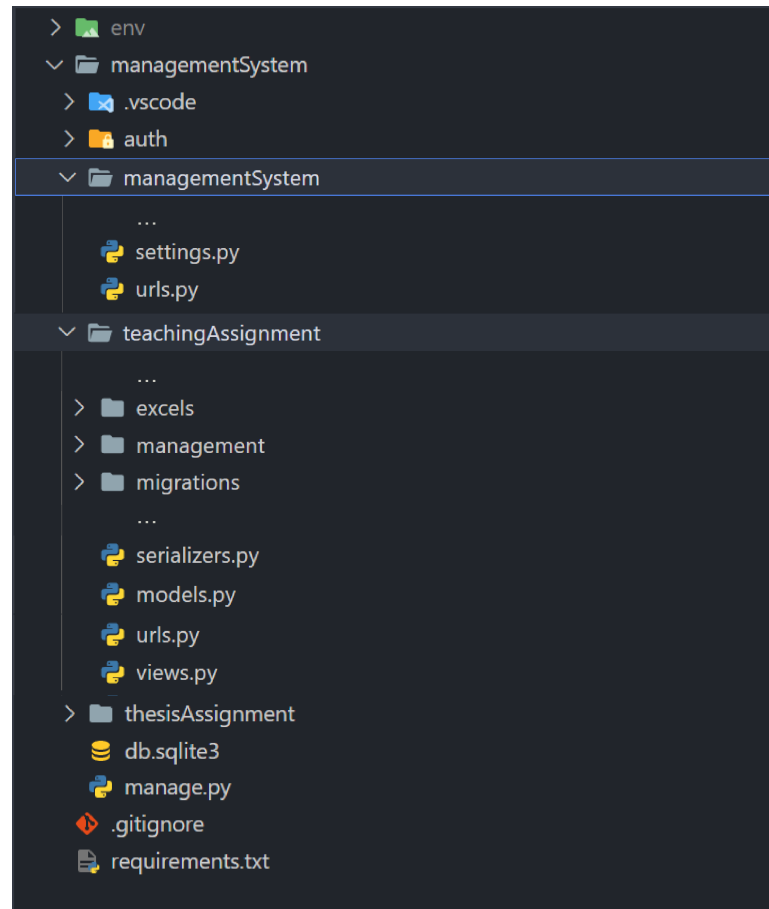


Figura 5.1: Estructura de los ficheros del proyecto servidor

A continuación se describen las carpetas y ficheros principales del proyecto servidor.

- **managementSystem**: es la carpeta principal del proyecto, en el fichero `settings.py` se encuentran todas las configuraciones. En el fichero `url.py` se deben agregar las *urls* de las *apps* creadas en el proyecto. En este caso, se incluyen las *urls* de las *apps* `teachingAssignment` y `teachingAssignment`.
- **teachingAssignment**: *app* creada para el desarrollo del proceso de asignación de docencia. En la carpeta `excels` se implementó la funcionalidad de descargar un fichero CSV con la información de la asignación de docencia.
- **thesisAssignment**: *app* creada para el desarrollo del proceso de planificación de las tesis. En la carpeta `excels` se implementó la funcionalidad de descargar en ficheros CSV la información de los tribunales y defensas de tesis.

- **teachingAssignment/management**: en esta carpeta se encuentran las implementaciones de los comandos `save_database` y `fill_database`, con los que se permite salvar y poblar la base de datos, respectivamente.

Para el desarrollo del cliente se utilizó la biblioteca Quasar. El proyecto cliente se creó con el uso de la herramienta Quasar CLI. La distribución de los ficheros quedó como se muestra a continuación.

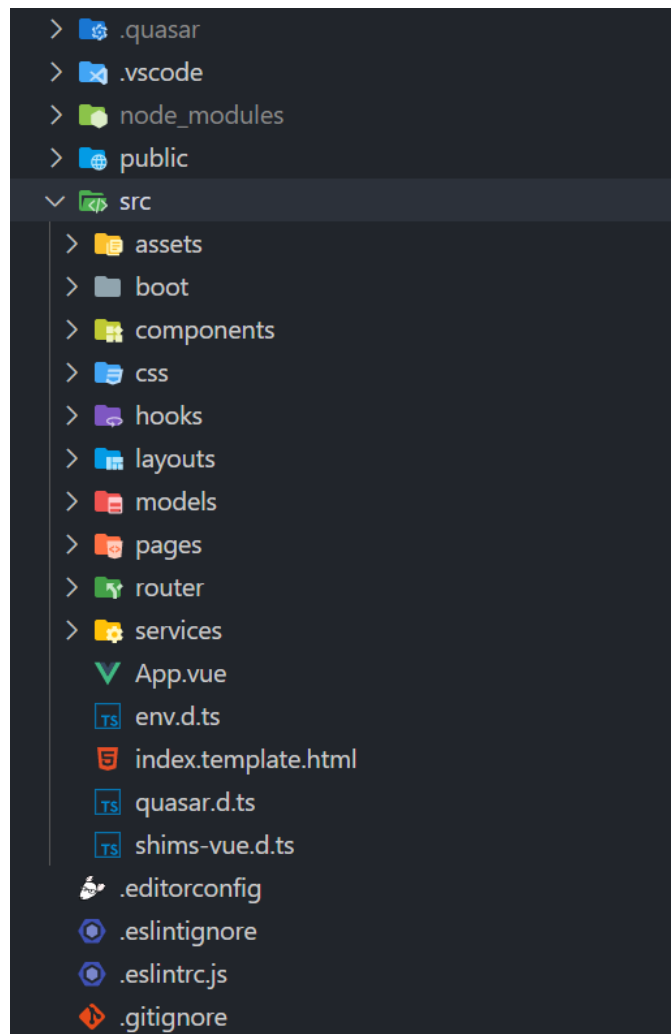


Figura 5.2: Estructura de los ficheros del proyecto servidor

Todo el desarrollo del cliente se encuentra dentro de la carpeta **src**. A continuación se describe qué encontrar en cada una de las carpetas.

- **assets**: las imágenes que se utilizaron en el proyecto.

- **boot**: configuraciones que se desean inyectar antes de que se cree una instancia de la componente raíz de la aplicación. Por ejemplo, configurar la *url* base para la comunicación con el servidor.
- **components**: las componentes implementadas para el funcionamiento del proyecto. Por ejemplo, en este trabajo se implementó una componente genérica para la visualización de cada una de las entidades que intervienen en los procesos de asignación de docencia y planificación de las tesis.
- **css**: los ficheros relacionados con el estilo utilizado. Por ejemplo, se redefinieron los colores **primary** y **secondary** por los colores que aparecen en el logo de MATCOM.
- **hooks**: los *hooks* que se implementaron para el funcionamiento del proyecto. Por ejemplo, se creó un *hook* para conocer qué departamento se desea administrar en cada momento y mostrar solo los datos relacionados con ese departamento.
- **layouts**: la implementación de la barra de navegación principal.
- **models**: las definiciones de los tipos utilizados. Por ejemplo, se creó una interfaz **ProfessorModel** donde se definen los campos que tiene un profesor, como: **id**, **name**, **lastName**, **department**, **scientificDegree** y **teachingCategory**.
- **pages**: la implementación de las páginas de la aplicación web.
- **router**: el enrutamiento de la aplicación web.
- **services**: la implementación de una capa de abstracción para la comunicación con el servidor.

Se definió una capa de abstracción para la comunicación con el servidor, que permite a partir de una *url*, realizar las operaciones CRUD: *create*, *read*, *update*, *delete*. A continuación se muestra su definición.

```
export const CrudServiceFactory = <T = any>(url: string) => {
  return {
    url: url,
    fullUrl: baseUrl + url,
    create(obj: T) {
      return api.post<T>(url, obj);
    },
    list(query: Dictionary = {}) {
      return api.get<ListResult<T>>(url + buildQuery(query));
    }
  };
}
```

```

    },
    update(id: string, obj: Dictionary) {
        return api.patch<T>(url + id + '/', obj);
    },
    delete(id: string) {
        return api.delete<T>(url + id);
    },
};
};
};

```

En el fichero `resources.ts`, que se encuentra dentro de la carpeta `services`, se debe definir un servicio para cada uno de los *endpoints* con los que se quiera realizar la comunicación. Por ejemplo, para realizar las peticiones relacionadas a los profesores se creó el servicio `professorService`, cuyo definición se muestra a continuación.

```

export const RESOURCES = {
    profesores: '/teaching-assignment/professors/',
    faculties: '/teaching-assignment/faculties/',
    careers: '/teaching-assignment/careers/',
    ...,
};

export const professorService = CrudServiceFactory<ProfessorModel>(
    RESOURCES.profesores
);

```

De esta forma, desde cualquier fichero que se importe el servicio `professorService`, se pueden realizar peticiones al servidor del estilo:

```

professorService.create(obj: T)
professorService.list(query: Dictionary)
professorService.update(id: string, obj: Dictionary)
professorService.delete(id)

```

Desde la aplicación web cliente se pueden visualizar y modificar todos los datos que intervienen en los procesos que se informatizaron. Para ello, se creó una página para administrar los datos de cada una de las entidades que se modelaron en la base de datos y se implementó una componente genérica que a partir de una configuración transforma los datos que se reciben desde el servidor y los muestra en una tabla. En esta tabla genérica se implementaron las operaciones `create`, `delete` y `update`. A continuación se muestra la interfaz que define una configuración para esta tabla.

```

export interface GenericCrudTableConfig {
  name: string;
  singularLabel: string;
  searchLabel?: string;
  actions?: {
    create?: boolean;
    update?: boolean;
    delete?: boolean;
    external?: {
      color: string;
      icon: string;
      func: () => void;
    }[];
  };
  service: CRUDService;
  query?: Dictionary;
  fields: FieldModel[];
  defaultValues?: Dictionary;
}

```

Para permitir las operaciones de *create*, *delete* o *update* en una tabla se debe indicar en la configuración con el campo **actions**, en el campo **services** se debe indicar el servicio a partir del cual se va a poblar la tabla, por ejemplo, el **professorService** que se describió anteriormente y en el campo **fields** se deben definir los datos que se deseen mostrar a partir de los objetos que se reciben con el servicio que se especifique.

A continuación se ilustra un ejemplo de cómo configurar una tabla para administrar los departamentos.

```

<template>
  <generic-crud-data-table :config="config" />
</template>

<script lang="ts">
import { facultyService, departmentService } from 'src/services';

export default defineComponent({
  components: { GenericCrudDataTable },
  name: 'departmentHandler',
  props: {},
  emits: [],
  setup(props, { emit }) {

```

```
const config = ref<GenericCrudTableConfig>({
  name: 'Departamentos',
  singularLabel: 'Departamento',
  searchLabel: 'Departamento o Facultad',
  service: departmentService,
  fields: [
    {
      name: 'name',
      label: 'Departamento',
      type: 'text',
    },
    {
      name: 'faculty',
      label: 'Facultad',
      column: {
        transform(row) {
          return `${row.faculty.name}`;
        },
      },
      type: 'select',
      selectOptions: {
        list: facultyService.list,
        value: 'id',
        label: 'name',
      },
      rules: ['required'],
    },
  ],
  actions: {
    create: true,
    update: true,
    delete: true,
  },
});

return { config };
});
</script>
```



En este caso se define que el servicio a partir del cual se va a poblar la tabla es `departmentService`, en `fields` se especifica que los campos que se desean mostrar son el nombre del departamento y la facultad a la que pertenece, y en `actions` que se desean habilitar las operaciones de create, update y delete en la tabla. En la figura 5.3 se muestra el resultado de esta configuración.

Departamento	Facultad	Acciones
Matemática Aplicada	MATCOM	
Programación y Sistemas de Información	MATCOM	
Inteligencia Artificial y Sistemas Computacionales	MATCOM	
Matemática	MATCOM	

Filas 10 1-4 de 4

Figura 5.3: Tabla para los departamentos

Si se desea extender este concepto para administrar los datos que intervienen en los procesos que se agregan en el sistema, solo se debe utilizar la componente `genericCrudDataTable` y definir la configuración que tendrá la tabla.

En la siguiente sección, se recomiendan las pautas a seguir para la incorporación de nuevos procesos en el sistema de gestión.

## 5.2. Recomendaciones para agregar un nuevo proceso en el sistema de gestión

Para describir los pasos que se recomiendan seguir para agregar un nuevo proceso en el sistema, se ilustrará con el proceso de planificación de las tesis. Supongamos que en el sistema de gestión solo está informatizado el proceso de asignación de docencia y se quiere agregar el de planificación de las tesis.

En el lado del servidor, se recomienda crear una `app` de Django para encapsular los ficheros necesarios que intervienen en la modelación del proceso de planificación de las tesis. Por tanto el primer paso sería ejecutar el siguiente comando.

```
python manage.py startapp thesisAssignment
```

Dentro de la carpeta `thesisAssignment` se deben definir los modelos, serializadores, vistas y urls necesarias. Para el proceso de planificación de las tesis se crearon nuevas entidades como `Place`, `Keyword`, `Thesis`, `ThesisCommittee` y `ThesisDefense`. Se modelaron sus relaciones, incluso con otras entidades que se habían definido durante el proceso de asignación de docencia. Por ejemplo, entre las entidades `Thesis`

y **Professor** se definieron las relaciones de tutor y cotutor.

En el lado del cliente, se siguieron los siguientes pasos para agregar el proceso de planificación de las tesis.

- Se creó la carpeta **thesisCommittee**, dentro de la carpeta **src/models**, para definir los tipos de las entidades que interviene en este proceso. Se agregaron las interfaces **KeywordModel**, **PlaceModel**, **ThesisModel**, **ThesisCommitteeModel** y **ThesisDefenseModel**.
- En el fichero **resources.ts**, que se encuentra en la carpeta **services**, se crearon los servicios para realizar las peticiones al servidor: **placeService**, **keywordService**, **thesisService**, **thesisCommitteeService** y **thesisDefenseService**.
- Se creó una carpeta **thesisCommittee** dentro de la carpeta **src/components/tables** para agrupar las componentes que se definen las configuraciones de las tablas correspondientes a este proceso.
- Se creó una carpeta **thesisCommittee** dentro de la carpeta **src/pages** para agrupar las vistas necesarias para la modelación de este proceso: **Keywords.vue**, **Place.vue**, **Thesis.vue**, **ThesisCommittee.vue** y **ThesisDefense.vue**.
- En el fichero **routes.ts** que se encuentra en la carpeta **src/routes**, se agregaron las rutas que llevan a las vistas definidas en el paso anterior.
- Se agregaron los botones a la barra de navegación que permiten ir a las vistas que se crearon en este proceso.

En resumen, en el lado del servidor, se recomienda la creación de una nueva *app* de Django que agrupe los ficheros necesarios para la modelación del proceso que se desea incorporar en el sistema. Se deben definir los modelos, cómo se serializan los datos y a través de cuáles *urls* se exponen. En el lado del cliente se recomienda definir interfaces para representen los tipos de las entidades que intervienen en el proceso, definir los servicios para realizar las peticiones al servidor, agrupar las componentes y las vistas necesarias en carpetas que indiquen el proceso que se está informatizando y agregar un mecanismo para acceder a esas vistas.

# Conclusiones

Conclusiones

# Recomendaciones

- Agregar temas de investigacion de los profesores para confeccionar mejores tribunales
- Si se agregan mas proyectos y se necesita un SGDB mas potente cambiar a PostgreSQL o MySQL.

# Bibliografía

- [1] S. Sulyman, «Client-Server Model,» *IOSR Journal of Computer Engineering*, vol. 16, págs. 57-71, ene. de 2014. DOI: 10.9790/0661-16195771 (vid. pág. 8).
- [2] *Sitio oficial de SQLite. High Reliability*, Accessed: 2022-04-18. dirección: <https://www.sqlite.org/hirely.html> (vid. pág. 8).
- [3] *Sitio oficial de SQLite. 35 % Faster Than The Filesystem*, Accessed: 2022-04-18. dirección: <https://www.sqlite.org/fasterthanfs.html> (vid. pág. 8).
- [4] T. Härder y A. Reuter, «Principles of Transaction-Oriented Database Recovery,» *ACM Computer Surveys*, vol. 15, págs. 287-, ene. de 1983 (vid. pág. 8).
- [5] *Sitio oficial de SQLite. SQLite is a Self Contained System*, Accessed: 2022-04-18. dirección: <https://www.sqlite.org/selfcontained.html> (vid. pág. 8).
- [6] *Sitio oficial de SQLite. About SQLite*, Accessed: 2022-04-18. dirección: <https://www.sqlite.org/about.html> (vid. pág. 8).
- [7] *Sitio oficial de SQLite. Appropriate Uses For SQLite*, Accessed: 2022-04-18. dirección: <https://www.sqlite.org/footprint.html> (vid. pág. 9).
- [8] *Sitio oficial de MySQL*, Accessed: 2022-04-18. dirección: <https://www.mysql.com/> (vid. pág. 9).
- [9] *Sitio oficial de PostgreSQL*, Accessed: 2022-04-18. dirección: <https://www.postgresql.org/> (vid. pág. 9).
- [10] *Django*, Accessed: 2022-04-24. dirección: <https://www.djangoproject.com> (vid. pág. 9).
- [11] D.-P. Pop y A. Altar Samuel, «Designing an MVC Model for Rapid Web Application Development,» *Procedia Engineering*, vol. 69, dic. de 2014. DOI: 10.1016/j.proeng.2014.03.106 (vid. pág. 9).
- [12] *Documentación oficial de Django. Making queries*, Accessed: 2022-04-18. dirección: <https://docs.djangoproject.com/en/4.1/topics/db/queries/> (vid. pág. 10).

- [13] *QUASAR BEYOND THE FRAMEWORK*, Accessed: 2022-04-25. dirección: <https://quasar.dev> (vid. pág. 11).
- [14] *Vue.js The Progressive JavaScript Framework*, Accessed: 2022-04-25. dirección: <https://vuejs.org/> (vid. pág. 11).
- [15] C. Dulanga, *Quasar vs. Vuetify vs. Bootstrap Vue: Choosing the Right Vue.js UI Library*, Accessed: 2022-04-25. dirección: <https://vuejs.org/> (vid. pág. 11).
- [16] L. García Hernández y M. Montes de Oca Richardson, «DISEÑO DE LA BASE DE DATOS,» en *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte*. Editorial Félix Varela, 2005, págs. 40-64, ISBN: 959-258-881-3 (vid. pág. 18).