

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

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

Introducción	1
1. Preliminares	2
1.1. Funcionamiento de la facultad	2
1.1.1. Asignación de docencia	2
1.1.2. Planificación de las tesis	5
1.2. Diseño de bases de datos	6
1.2.1. Metodología de diseño de bases de datos	6
1.2.2. Correcto diseño de bases de datos relacionales	8
1.3. Herramientas utilizadas para el desarrollo del software	9
1.3.1. Modelo cliente-servidor	10
1.3.2. Desarrollo del servidor	10
1.3.3. Desarrollo del cliente	11
2. Descripción de las funcionalidades que se desean	13
2.1. Asignación de docencia	13
2.2. Planificación de las tesis	15
3. Descripción del diseño de la base de datos	19
3.1. Modelación de la asignación de docencia	19
3.2. Modelación de los tribunales de tesis	22
4. Descripción las herramientas implementadas	25
4.1. Asignación de docencia	26
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

5. Extensibilidad	40
5.1. Estructura del cliente y servidor	40
5.2. Recomendaciones para agregar un nuevo proceso en el sistema de gestión	44
Conclusiones	46
Recomendaciones	47
Bibliografía	48

Índice de figuras

3.1.	23
4.1. Vista inicial del sistema de gestión	25
4.2. Vista de la planificación de las asignaturas	27
4.3. Vista del formulario para crear una planificación de asignaturas . . .	27
4.4. Vista de la planificación de las asignaturas	28
4.5. Vista de la asignación de docencia	28
4.6. Vista del formulario para realizar una asignación de docencia	29
4.7. Vista del formulario con los datos ingresados de una asignación de docencia	30
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 planificar 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

Ejemplos de código

Introducción

Capítulo 1

Preliminares

En este capítulo se presenta una introducción a los procesos que se llevan a cabo en la Facultad de Matemática y Computación de la Universidad de La Habana, así como una descripción de las herramientas que se utilizaron en este trabajo.

[ESTRUCTURA DEL CAP]

1.1. Funcionamiento de la facultad

Cada año en la Facultad de Matemática y Computación de La Universidad de La Habana se realizan procesos administrativos y de planificación relacionados con los cursos escolares. Cada semestre es necesario realizar la planificación de docencia, las evaluaciones de cada asignatura, la planificación de los exámenes finales y revalorizaciones, en el semestre final de un curso, es necesario realizar las planificaciones de las tesis, entre otros. [ARREGLAR ESTO, NO SE MUY BIEN QUE DECIR Y QUE NO]

Este trabajo tiene como objetivo el desarrollo de una herramienta informática que permita realizar dos de los procesos mencionados anteriormente: asignación de docencia y planificación de las tesis. En las siguientes secciones se describen cómo se realizan actualmente estos procesos en la facultad de MATCOM.

1.1.1. Asignación de docencia

En la facultad de Matemática y Computación de La Universidad de la Habana, se estudian las carreras de Matemática y Ciencia de la Computación. Cada carrera se rige por un plan de estudio, que es un programa estructurado donde se definen los contenidos que forman parte del currículo de los graduados. En el plan de estudio

de una carrera se debe caracterizar la profesión, describir las principales funciones de un profesional y los valores a desarrollar durante la carrera. Además determina la cantidad de horas clase de cada asignatura, así como el año y semestre en el que se deben impartir. Cada cierto tiempo el plan de estudio de una carrera universitaria cambia, principalmente para mantener la formación de los profesionales actualizada con el desarrollo de la sociedad.

Actualmente en la carrera de Ciencia de la Computación de la Facultad de Matemática y Computación de La Universidad de La Habana, existen tres planes de estudios vigentes. En el año 2018 se actualizó el plan de estudio para los profesionales de la carrera de Ciencia de la Computación, se creó 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 necesaria la modificación del plan E para la reducción de asignaturas del currículo, creándose el plan de estudio E (COVID). Por tanto los estudiantes que ingresaron en la facultad en el curso 2017-2018, que se encuentran en el quinto año de la carrera, se rigen por el plan D, los estudiantes que se encuentran en el primer, tercer y cuarto año se rigen por el plan E y los estudiantes que cursan el segundo año se rigen por el E(COVID).

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 (voz pasiva) por profesores y cuentan con un jefe de departamento. Los profesores que integran un departamento tienen un grado científico (Licenciado, Máster, Doctor) y una categoría docente (Titular, Auxiliar, Instructor, Asistente). Los departamentos de una facultad pueden brindar servicios a otras facultades, por ejemplo el departamento de Matemática Aplicada imparte en el segundo año de la carrera de Geografía la asignatura Estadística.

La dirección de la facultad realiza la planificación de la docencia, que consiste en determinar las asignaturas que se deben impartir en el semestre, así como la cantidad de grupos de conferencias, clases prácticas u otras actividades de clase. Cada departamento recibe la planificación correspondiente a las asignaturas de las cuáles se ocupa. En la tabla 1.1 se muestra un fragmento de la planificación de docencia para el departamento de Matemática Aplicada. La notación en la columna de los grupos n/m indica que existen n cantidad de grupos de conferencia y m cantidad de grupos de clases prácticas.

La asignación de docencia en un departamento consiste en distribuir a los profesores para cumplir con la planificación de docencia. En la tabla 1.2 se muestra un fragmento de una posible asignación de docencia para la planificación de la tabla 1.1.

Tabla 1.1: Fragmento de la planificación de docencia para el departamento de Matemática Aplicada.

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

Tabla 1.2: Fragmento de la asignación de docencia para el departamento de Matemática Aplicada.

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)

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 en las distintos tipos de clases como conferencias, clases prácticas, seminarios, laboratorios u otros, se maneja por el colectivo de profesores de la asignatura e incluso puede cambiar de un año a otro. En la tabla 1.2, se puede apreciar que la asignatura Optimización Matemática I tiene un total de horas a impartir de 64, que se distribuyen en 32 horas de conferencia y 32 horas de clases prácticas, y que fueron asignadas a las profesoras Aymeeé Marrero y Gemayqzel Bouza, respectivamente.

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, 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 la siguiente sección se describe cómo se realiza el proceso de planificación de tesis.

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. 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. El ejercicio se compone de dos partes, la escritura de un documento que recoja todo el proceso investigativo y, posteriormente, la exposición o defensa del mismo ante un tribunal.

La planificación de las tesis se puede separar en dos subprocesos principales: confección de los tribunales y planificación de las defensas. La organización de los tribunales consiste en definir los profesores que formarán parte de cada tribunal. Los profesores deben tener dominio acerca de los temas que se aborden en la investigación. La composición de los tribunales de tesis puede variar de un año a otro, de acuerdo a las condiciones del curso escolar, generalmente están formados por tres profesores, que asumen los roles de oponente, secretario y presidente. En el curso 2022, los estudiantes de cuarto (plan E) y quinto año (plan D) de la carrera de Ciencia de la Computación deben realizar su ejercicio de culminación de estudios. Como consecuencia, los tribunales de este curso estarán conformados por los tutores, el oponente y el presidente. Una vez se definan los tribunales, es necesario planificar cuándo y dónde se llevarán a cabo la defensas de la tesis, se debe definir para cada una: fecha, hora y local. Para realizar la planificación 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 a la vez.

Por ejemplo, en la tabla 1.3 se muestran los datos de dos tesis correspondientes al curso 2022, que se deben planificar.

Tabla 1.3

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 Camila Pérez

El primer paso, es realizar la confección de los tribunales de tesis, donde se debe definir el oponente y presidente. En la figura 1.4 se ilustran los tribunales para las dos tesis anteriores.

Tabla 1.4: Posibles tribunales de tesis

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

Una vez se definan los tribunales, se debe planificar hora y lugar donde se realizarán las defensas, como se ilustra en la figura 1.5.

Tabla 1.5: Posible horarios para las defensas de 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

Para almacenar la información relevante a los procesos de asignación de docencia y planificación de las tesis, es imprescindible el uso de una base de datos. En la siguiente sección se presentan los elementos fundamentales para el correcto diseño de la base de datos.

1.2. Diseño de bases de datos

El desarrollo de la informática y la computación ha permitido el almacenamiento de grandes cantidades de datos en espacios físicos limitados. Actualmente los sistemas de bases de datos juegan un papel fundamental en el desarrollo de todo tipo de sistemas computacionales.

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 [4].

1.2.1. Metodología de diseño de bases de datos

La complejidad de la información y la cantidad de requisitos que se deseen modelar en un sistema computacional hacen que el diseño de una base de datos no sea una tarea sencilla. Por tanto es común descomponer el proceso de diseño en tres etapas fundamentales: diseño conceptual, diseño lógico y diseño físico.

Diseño conceptual

En esta fase se obtiene como resultado una representación abstracta y de alto nivel de la realidad a partir de las especificaciones de requisitos de usuario [REF libro]. El diseño conceptual comienza con la identificación de las necesidades de los usuarios, que a menudo se pueden obtener a través de: examinar la documentación existente como formularios, observando y analizando el procesamiento de la información en el proceso que se desea modelar o mediante entrevistas a los usuarios finales [1].

El objetivo del diseño conceptual es describir el contenido de información de la base de datos, mediante un esquema conceptual de la base de datos [5], que es independiente del SGBD que se utilice para manipular la base de datos. Los programadores o diseñadores utilizan modelos conceptuales de datos para la construcción de esquemas. El modelo entidad-relación es uno de los más utilizados para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976 [REF], y está formado por un conjunto de conceptos que permiten describir la realidad mediante representaciones gráficas y lingüísticas.

La metodología utilizada para el desarrollo del modelo conceptual de datos para la modelación de los procesos de asignación de docencia y confección de tribunales de tesis fue la metodología genérica MER/XX, que se basa en un enfoque híbrido entre el modelo entidad/relacional extendido y los conceptos de la modelación orientada a objetos [4].

El modelo entidad-relación extendido describe con un alto nivel de abstracción el significado de los datos, las relaciones entre ellos y las reglas de negocio de un sistema de información. Sus dos elementos principales son las entidades y las relaciones, además existen extensiones al modelo básico como atributos de las entidades o cardinalidades de las relaciones, que aportan al modelo una mayor expresividad.

Diseño lógico

Es el proceso de transformar el esquema conceptual del dominio de la aplicación que se obtiene en la fase anterior, en un esquema para el modelo de datos subyacente a un SGBD particular. Existen diferentes modelos matemáticos utilizados para el diseño lógico de las bases de datos, tales como: el modelo de listas invertidas[REF], el modelo jerárquico[REF], el modelo de redes[REF] y el modelo relacional[REF].

El modelo relacional fue propuesto por Edgar Frank Codd en 1970. Es un modelo que se basa en la lógica de predicados y en la teoría de conjuntos, donde todos los datos se representan en términos de tuplas, agrupados en relaciones.

El modelo relacional fue el primer modelo de base de datos que se describió en términos matemáticos formales. A pesar de que existen implementaciones del modelo de base de datos relacional como lo definió originalmente Codd, no han tenido éxito popular hasta el momento. No obstante al modelo relacional se le atribuye un gran

valor por su desarrollo teórico, que ha sido fundamento de muchos de los sistemas de gestión de bases de datos relacionales que se utilizan hoy en la actualidad, como: MySQL, Oracle, SQL Server, PostgreSQL y Microsoft Access.

El resultado de esta fase consiste en una descripción de las estructuras de datos utilizadas para almacenar la base de datos[5], que se ajuste al modelo que utilice el SGBD. Esto quiere decir que, si el modelo conceptual es expresado como un modelo entidad-relación y el modelo lógico utilizado en el SGBD es el modelo relacional, entonces las entidades, relaciones y los atributos del modelo entidad-relación deben representarse como relaciones.

NOTA: ACLARAR el concepto de relación en el modelo relacional

Diseño físico

En esta etapa se transforma la estructura obtenida en la etapa del diseño lógico, con el objetivo de conseguir mayor eficiencia. Es necesario evaluar los aspectos de implementación física relacionados al SGBD que se utilice. Por ejemplo: si se trabaja con una base de datos relacional, la transformación de la estructura puede consistir en crear una nueva relación que sea la combinación de varias relaciones o separar una relación en varias relaciones o añadir algún atributo calculable a una relación.

1.2.2. Correcto diseño de bases de datos relacionales

El diseño de una base de datos mediante el enfoque relacional, es una tarea no determinista, es posible obtener distintos esquemas relacionales como propuestas de la base de datos. Un diseño incorrecto de una base de datos puede no responder apropiadamente a las exigencias del proceso que se modela, y puede conllevar a la generación de dificultades o errores en el acceso a los datos. Entre los principales errores o dificultades que se pueden generar se encuentran:

- **Redundancia en los datos**, provoca un aumento innecesario del tamaño de la base de datos, disminuyendo la eficiencia, además puede provocar inconsistencia de los datos que puede conducir a la corrupción de los mismos.
- **Violación de la integridad de los datos**, el término “integridad de los datos” se refiere a la correctitud y completitud de la información en una base de datos. Cuando los datos son modificados con sentencias INSERT, DELETE o UPDATE la integridad de los datos puede comprometerse.

Para lograr un correcto diseño de la base de datos se recomienda aplicar un método formal de análisis a cada uno de los esquemas obtenidos intuitivamente durante la fase de diseño conceptual, que se conoce como proceso de normalización.

Normalización

La normalización de una base de datos es un proceso que consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación (diseño conceptual) al modelo relacional (diseño lógico), con el objetivo de minimizar la redundancia de datos. Cada regla se denomina una “forma normal”, si se cumple la primera regla, se dice que la base de datos está en “primera forma normal”, si se cumplen las tres primeras reglas se dice que la base de datos está en “tercera forma normal”. El proceso de normalización se puede caracterizar como la transformación sucesiva de una colección de relaciones hacia una forma más restrictiva, de modo que mientras mas profundo sea el nivel de normalización menor será la redundancia que albergue la base de datos [3]. A continuación se introducen las definiciones de las reglas formales:

- **Primera Forma Normal (1FN):** una relación está en 1FN si y solo si, para cada ocurrencia de la relación, toda tupla contiene exactamente un valor del dominio subyacente en cada atributo.
- **Segunda Forma Normal (2FN):** una relación está en 2FN si, además de estar en 1FN, todos los atributos que no forman parte de alguna llave candidata constituyen información acerca de la(s) llave(s) completa(s) y no de algún subconjunto de ella(s).
- **Tercera Forma Normal (3FN):** una relación está en 3FN si, además de estar en 2FN, los atributos que no forman parte de alguna llave candidata constituyen información solo acerca de la(s) llave(s) y no acerca de otros atributos.

Aunque existen otros niveles de normalización como la “Forma Normal de Boyce-Codd (BCFN)”, la “Cuarta Forma Normal(4FN)” y la “Quinta Forma Normal (5FN)”, se considera que una base de datos en 3FN presenta niveles nulos o admisibles de redundancia en los datos[8].

1.3. 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 hace uso de Django [REF], en particular Django Rest Framework, aprovechando la versatilidad, seguridad, escalabilidad y mantenibilidad que ofrece. El SGBD escogido fue SQLite, que viene integrado de

forma predeterminada en Django. El cliente se desarrolló usando Quasar [REF], creando una *single-page application* (SPA). A continuación se profundiza en el modelo y herramientas utilizadas.

1.3.1. 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. 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.

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.

1.3.2. Desarrollo del servidor

Para la implementación de los servicios que consumen los clientes del sistema de gestión se hace uso de la biblioteca Django [REF]. Django permite el desarrollo de sitios web seguros y mantenibles [REF]. 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. Estas componentes son responsables de diferentes partes del código e incluso pueden utilizarse de forma independiente.
- 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 (APIs). [X q es relevante en este trabajo?]
- Provee un Object Relational Mapper (ORM), que permite la interacción con la base de datos de forma orientada a objetos. 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.
- Es escalable, utiliza una arquitectura de “shared-nothing” o “nada compartido”, que permite agregar hardware en cualquier nivel: servidores de bases de datos, servidores de almacenamiento en caché o servidores web (REFERENCIA).

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

1.3.3. Desarrollo del cliente

Dentro de las tecnologías de desarrollo web más utilizadas, JavaScript es una de las más populares en la actualidad [9]. JavaScript es un lenguaje de programación basado en prototipos y multiparadigma. Como JavaScript está implementado (voz pasiva) en la mayoría de los navegadores web, se utiliza para el desarrollo del cliente permitiendo el mejorar las interfaces de usuarios y la creación de páginas web dinámicas [7].

La popularidad de JavaScript para el desarrollo web ha impulsado la creación de distintas bibliotecas y herramientas, entre las más utilizadas se encuentran: Angular, React y Vuejs [6]. Angular es desarrollada y mantenida por Google, React por META

(Facebook) y Vuejs por un equipo de colaboradores a nivel mundial, con Evan Yu, su principal creador, un ex ingeniero de Google.

En los últimos años, Vue.js se ha convertido en un fuerte competidor de Angular y React. El crecimiento de Vue.js se relaciona con su simplicidad y disponibilidad de bibliotecas y materiales de aprendizaje. A pesar de que Vue.js se define en su sitio oficial como “An approachable, performant and versatile framework for building web user interfaces” o “Un marco accesible, eficaz y versátil para crear interfaces de usuario web” [10], 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 [2]. Este último fue el escogido para el desarrollo del cliente de la aplicación web.

Quasar

Es una biblioteca de código abierto basada en Vue.js, que permite crear aplicaciones con una sola base de código. 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. Entre las principales características que influyeron en que

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 para cubrir la mayoría de las necesidades de la web.
- buenas prácticas integradas tales como: compresión de código, manejo avanzado de caché, mapeo de fuentes y carga diferida.
- Quasar-CLI, una herramienta que genera de forma automática la estructura del proyecto, con los paquetes y configuraciones para empezar a desarrollar una aplicación web.
- amplia comunidad y documentación.

Capítulo 2

Descripción de las funcionalidades que se desean

El objetivo principal 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: asignación de docencia y planificación de las tesis. Se desea además que la herramienta sea lo suficientemente extensible para que permita, en un futuro, la integración de otros procesos como los que se mencionan en la sección 1.1.

En las siguientes secciones se describen las funcionalidades que se desean incorporar para cada proceso.

2.1. Asignación de docencia

Se desea implementar una mecanismo que permita realizar la asignación de docencia facilitando el acceso a toda la información que interviene en este proceso. A continuación se describen las principales funcionalidades que se desean:

- 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 la 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.

A continuación se describen como se desean utilizar las funcionalidades anteriores durante el proceso de asignación de docencia.

Al comienzo de un período de tiempo, el jefe de departamento debe realizar la planificación de docencia del semestre. Se desea que la herramienta muestre las asignaturas que se deben impartir por ese departamento en el período de tiempo dado. Por ejemplo en el período enero-julio del curso 2022, el departamento de Matemática Aplicada debe impartir las siguientes asignaturas.

Tabla 2.1: Fragmento de la planificación de docencia.

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

La primera fila de tabla indica que la asignatura Optimización Matemática I, se imparte en la facultad de MATCOM a los estudiantes de tercer año de la carrera de Matemática, con un total de horas clase de 64 y que es un solo grupo de conferencias y un solo grupo de clases prácticas.

A partir de esta información, el jefe de departamento comienza el proceso de asignación. Se desea que el sistema muestre los profesores que están disponibles para cubrir las actividades de clase de las asignaturas (conferencias, clases prácticas). En el curso 2022, se le asignaron las 32 horas de conferencia de la asignatura Optimización Matemática I, 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. En la tabla 2.2 se muestra la asignación de docencias correspondiente a la planificación que se muestra en la tabla 2.1

Tabla 2.2: Fragmento de la 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)

Para la asignación que se muestra en la tabla 2.2, el sistema de gestión debe mostrar las siguientes cargas docentes:

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

Además se quiere que el sistema muestre las asignaturas o grupos que faltan por cubrir para cumplir con la planificación de docencia. 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: confección de los tribunales de tesis y planificación de las defensas de las mismas. 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 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 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 planificación de las defensas de tesis.
- Exportar la confección de los tribunales de tesis a un documento con un formato preestablecido.
- Exportar la planificación de las defensas de tesis a un documento con un formato preestablecido.

Para la ejemplificación de las funcionalidades que se desean durante el proceso de planificación de las tesis se utilizará un ejemplo. En la tabla 2.3, se muestran dos de las tesis que se deben evaluar como parte del ejercicio de culminación de estudios del curso 2022.

Tabla 2.3: Fragmento de dos de las tesis correspondientes al curso 2022

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 Camila Pérez

El encargado de realizar la planificación de las tesis debe confeccionar un tribunal para cada una de ellas. Se desea que el sistema de gestión muestre los profesores de la facultad para la confección de los tribunales. En el curso 2022 los tribunales estarán compuestos por los tutores, un oponente y un presidente, por tanto solo se deben determinar los profesores que asumirán los roles de oponente y presidente. En la figura tal se muestran posibles tribunales para las tesis correspondientes a la tabla 2.3.

Tabla 2.4: Posibles tribunales de tesis

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

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 2.5 se muestran las participaciones de los profesores en tribunales de acuerdo a la tabla 2.4

Tabla 2.5: Cantidad de tribunales de tesis en los que participa cada profesor

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

Una vez se definan los tribunales de tesis el próximo paso es planificar un horario para la defensa de la misma. Se debe definir la fecha, hora y local en donde se realizará este ejercicio. En la tabla tal se muestran posibles planificaciones para las defensas de las tesis que se muestran en la tabla 2.3.

Tabla 2.6: Posible horarios para las defensas de 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

Una vez concluido el proceso de planificación de las tesis se desea poder exportar los datos de los tribunales y las planificaciones de 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, el sistema genere automáticamente un tribunal vacío y una planificación para la defensa de la misma, sin definir la fecha, hora y local.

Además se desea conocer para cada profesor los horarios que tiene ocupado en participaciones en tribunales de tesis, por si es necesario realizar algún cambio en las planificaciones de las defensas.

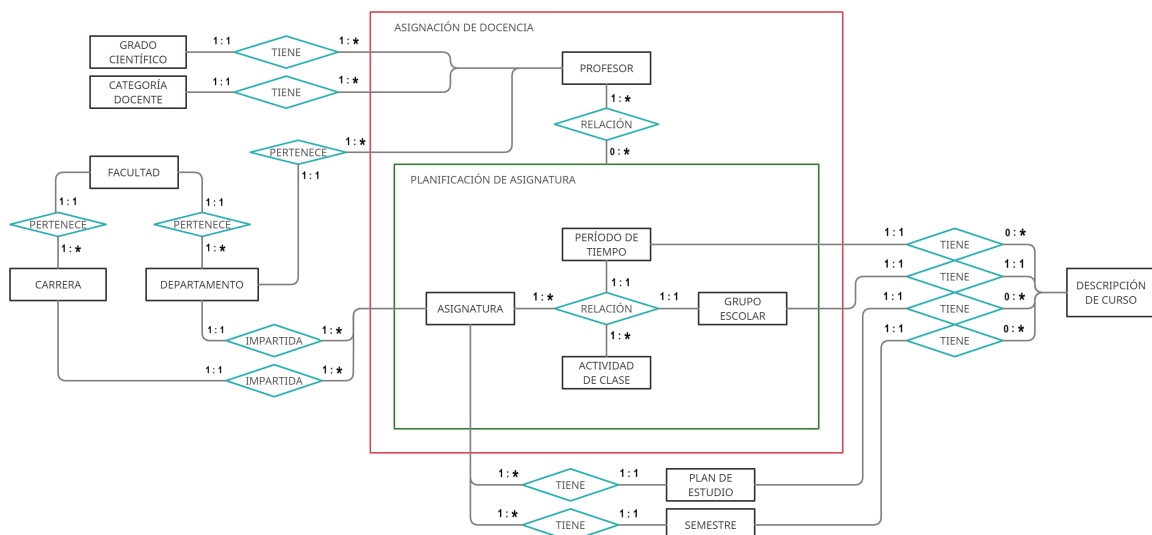
En el próximo capítulo se describe como se modelaron los datos para realizar la asignación de docencia y la planificación de las tesis.

Capítulo 3

Descripción del diseño de la base de datos

3.1. Modelación de la asignación de docencia

En la fase de diseño conceptual se modelaron las entidades fundamentales que intervienen en el proceso de la asignación de docencia, así como las interrelaciones que se establecen entre ellas y se obtuvo el siguiente esquema a partir del modelo entidad-relación extendido.



Con el objetivo de simplificar la representación del modelo entidad-relación no se agregaron a la imagen los atributos correspondientes a cada entidad, por lo que a continuación se describen las entidades en mayor profundidad.

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 asignatura 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, un departamento 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 es atendida 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 son representadas 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 imparten en el mismo semestre.

CURSO ESCOLAR: Representa los cursos escolares. El nombre del curso escolar se modela como un atributo. Pueden existir una o muchas asignaturas que se imparten en el mismo curso escolar.

PERÍODO DE TIEMPO: Representa los períodos de tiempo del año. El nombre del período de tiempo se modela como un atributo. Pueden existir una o muchas asignaturas que se imparten en el mismo período de tiempo.

GRUPO ESCOLAR: Representa los años académicos asociados a una carrera, como por ejemplo: Matemática primer año (M1) o Computación tercer año (C3). El nombre del grupo 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 período de tiempo y para un grupo escolar específico. Está compuesta por la agregación de las entidades ASIGNATURA, ACTIVIDAD DE CLASE, GRUPO ESCOLAR, CURSO 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 (poner ejemplo real de una asignatura con distintas act de clase y como queda la separación)

ASIGNACIÓN DE DOCENCIA: Representa las asignaciones de planificaciones de asignaturas a profesores. Además se agrega un atributo porciento 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.

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

Para la modelación del proceso de confección de los tribunales de tesis, se crea un esquema basado en el modelo entidad-relacional extendido como en el proceso anterior.

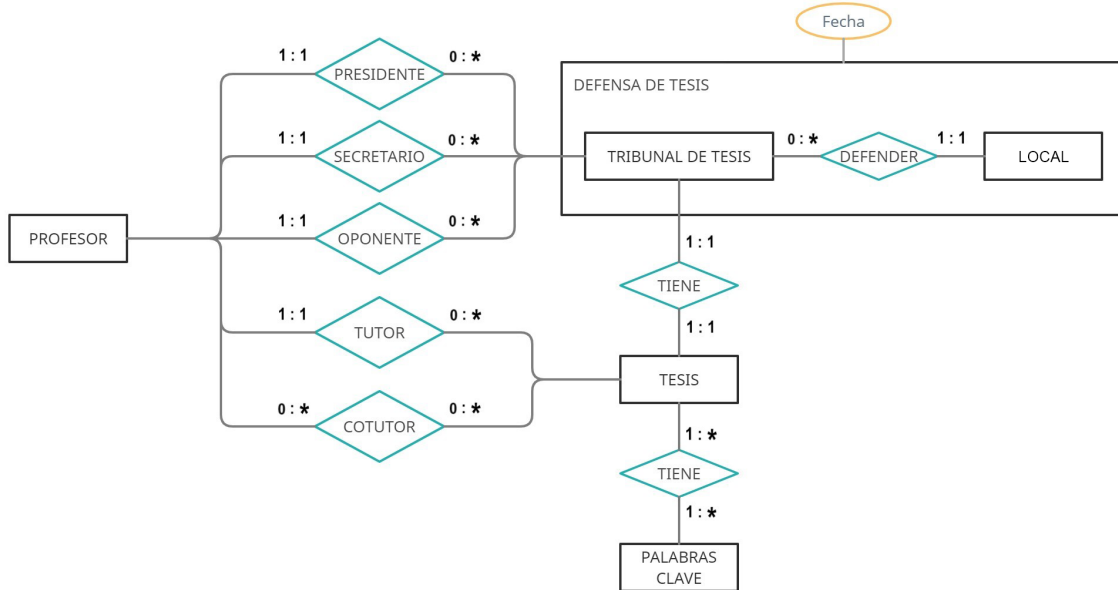


Figura 3.1

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.

Capítulo 4

Descripción las herramientas implementadas

En este capítulo se presenta la propuesta de solución para 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.

Se implementó un sistema de gestión siguiendo el modelo cliente-servidor. El desarrollo del servidor se efectuó con el uso Django, en particular Django Rest Framework y el cliente se desarrolló utilizando Quasar para la creación de las interfaces de usuarios. Para la administración de la base de datos se utilizó el SGBD SQLite. Como resultado se creó una aplicación web, cuya vista inicial se muestra en la figura 4.1.



Figura 4.1: Vista inicial del sistema de gestión

[EXPLICACION DEL FLUJO PARA REALIZAR ASIGNACION DE DOCENCIA O TESIS]

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 ejemplificación de 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 muestran las asignaturas que se deben impartir y en la 4.2 la distribución de los profesores para cubrir con la planificación.

Tabla 4.1: Fragmento de la planificación de docencia.

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

Tabla 4.2: Fragmento de la 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)

El primer paso que debe realizar el jefe de departamento es la planificación de la docencia, la interfaz de usuario implementada en el sistema de gestión para realizar este proceso se muestra en la figura 4.2

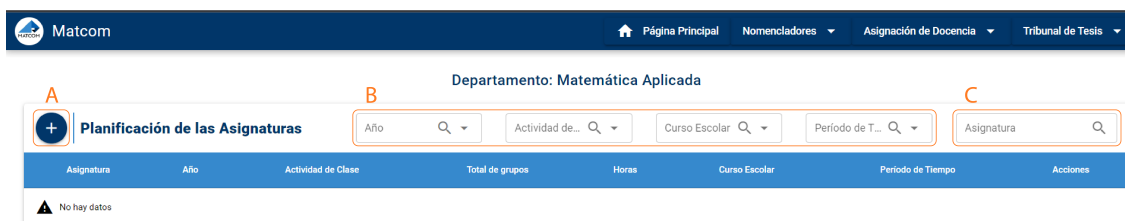


Figura 4.2: Vista de la planificación de las asignaturas

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

- A: botón para agregar una nueva planificación 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 planificación 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.

Figura 4.3: Vista del formulario para crear una planificación 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							
<div> <div>+</div> <div>Planificación de las Asignaturas</div> <div>Año <input type="text"/></div> <div>Actividad de... <input type="text"/></div> <div>Curso Escolar <input type="text"/></div> <div>Período de T... <input type="text"/></div> <div>Asignatura <input type="text"/></div> </div>							
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 planificación de las asignaturas

Por cada planificación 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 planificación. En la figura 4.5 se muestran las asignaciones de docencias creadas automáticamente por el sistema a partir de las planificaciones agregadas en la figura 4.4.

















Departamento: Matemática Aplicada								
<div> <div>+</div> <div>Asignaciones de docencia</div> <div>Profesor <input type="text"/></div> <div>Asignatura o Actividad <input type="text"/></div> </div>								
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			 
							Filas 10 1-8 de 8	

Figura 4.5: Vista de la asignación de docencia

El próximo paso es realizar la asignación de los profesores para cubrir la plani-

ficación de 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.

The screenshot shows the 'Asignaciones de docencia' (Teaching Assignments) interface. At the top, there's a header with 'Departamento: Matemática Aplicada' and a navigation bar with 'Página Principal', 'Nomencladores', 'Asignación de Docencia', and 'Tribunal de Tesis'. Below the header, there's a section titled 'Asignaciones de docencia' with a search bar for 'Profesor' and 'Asignatura o Actividad'. The main content is a table with columns: Asignatura, Año, Actividad de clase, Cantidad de horas, Total de grupos, Grupo, Profesor, Porcentaje, and Acciones. The table lists several assignments, including 'Modelos de Optimización' and 'Estadística'. A modal form titled 'Editando Asignación de docencia' is open, showing fields for 'Asignatura' (Optimización Matemática I — Conferencia — M3), 'Grupo' (1), 'Profesor', and 'Porcentaje'. The modal has 'Cancelar' and 'Editar' buttons. The table below the modal shows the following data:

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

At the bottom right of the table, there's a pagination bar showing 'Filas 10' and '1-8 de 8'.

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 4.1, 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.

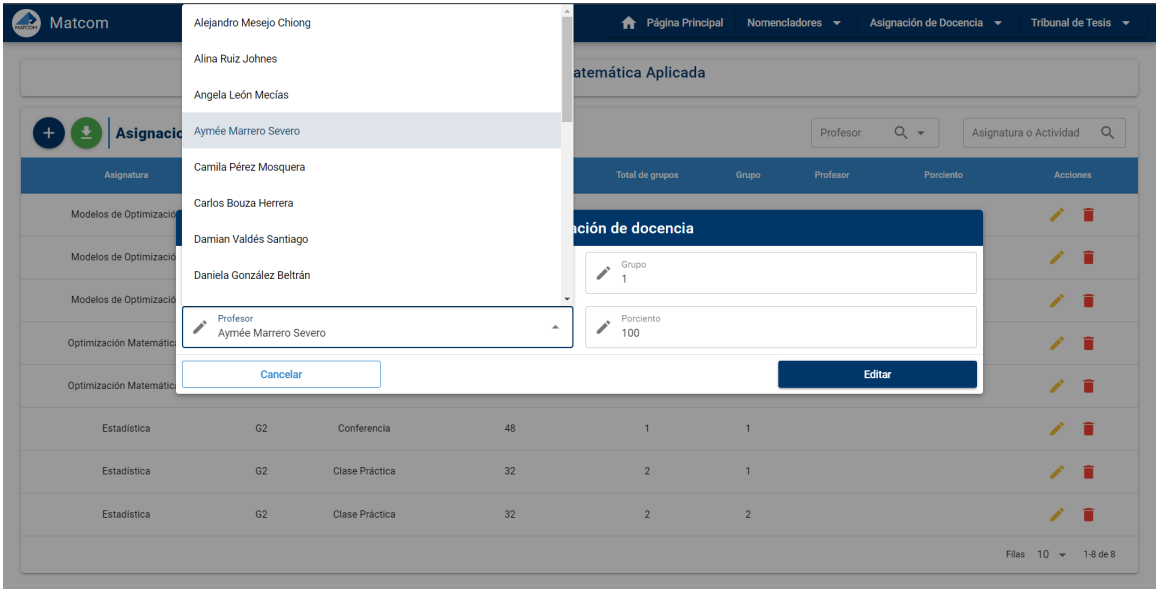


Figura 4.7: Vista del formulario con los datos ingresados de una asignación de docencia

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

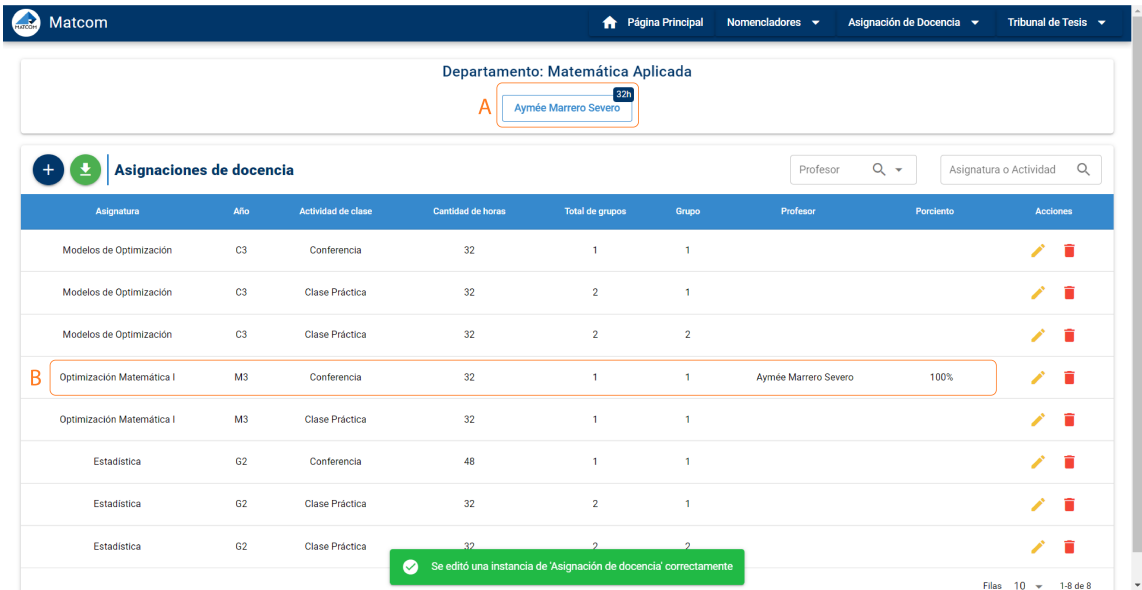
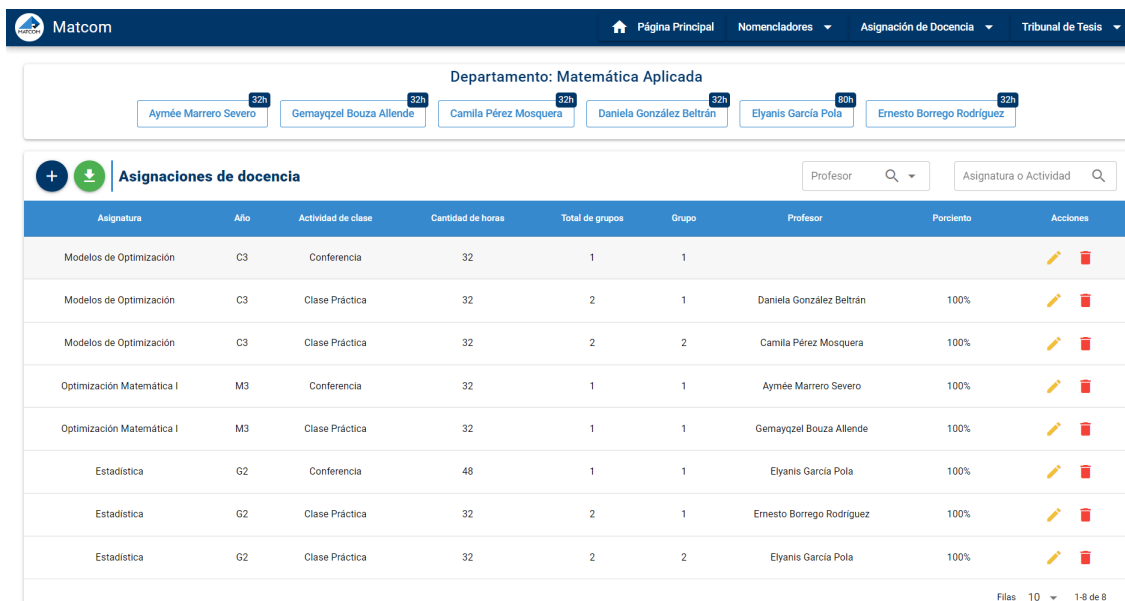


Figura 4.8: Vista resultante de realizar una asignación de docencia correctamente

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 Aymeeé 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.



















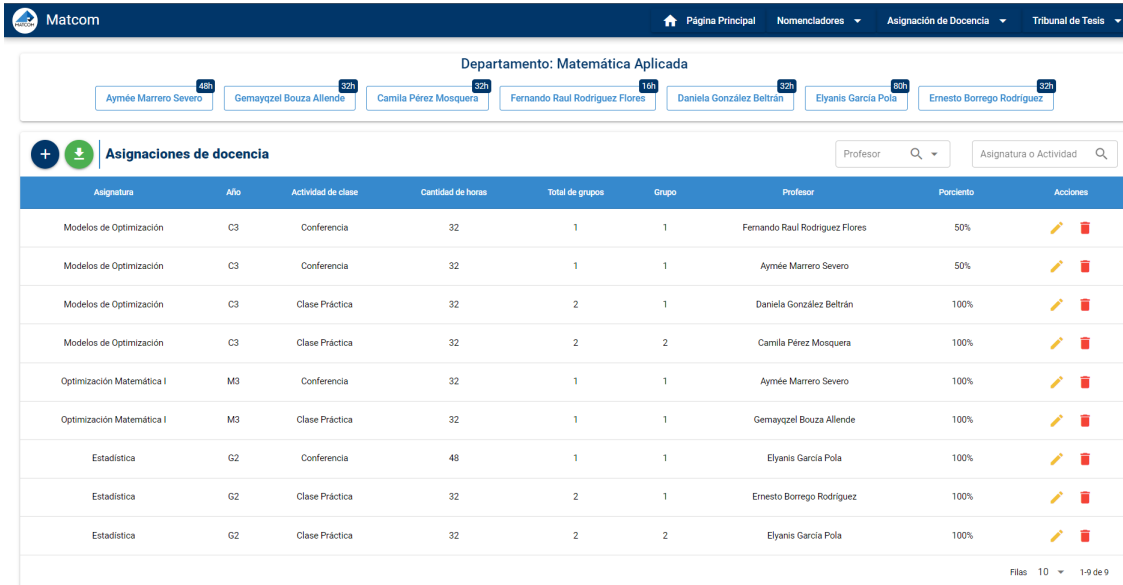
Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Modelos de Optimización	C3	Conferencia	32	1	1			 
Modelos de Optimización	C3	Clase Práctica	32	2	1	Daniela González Beltrán	100%	 
Modelos de Optimización	C3	Clase Práctica	32	2	2	Camila Pérez Mosquera	100%	 
Optimización Matemática I	M3	Conferencia	32	1	1	Aymée Marrero Severo	100%	 
Optimización Matemática I	M3	Clase Práctica	32	1	1	Gemayqzel 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 Aymeeé Marrero y Fernando Rodríguez, con 16 horas cada uno. Para modelar esta distribución en el sistema de gestión, 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 hours: Aymée Marrero Severo (48h), Gemayzel Bouza Allende (32h), Camila Pérez Mosquera (32h), Fernando Raul Rodriguez 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 Rodriguez 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 de gestión 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 asignaturas. Para poder crear la planificación de las asignaturas 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

El proceso de planificación de tesis consta de dos subprocesos principales: confección de los tribunales y planificación de la defensa. En esta sección se describen los pasos a seguir para completar el proceso de planificación de las tesis a través del sistema de gestión que se propone en este trabajo. Para ilustrar este proceso se utilizará el mismo ejemplo que se describe en la sección 2.2. En la tabla 4.3 se muestran dos de las tesis que se deben efectuar como ejercicio de culminación de estudios correspondientes al curso 2022. En la tabla 2.4, el tribunal seleccionado y en la tabla 4.5 el horario escogido para la defensa de las mismas.

Tabla 4.3: Fragmento de dos de las tesis correspondientes al curso 2022

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 Camila Pérez

Tabla 4.4: Posibles tribunales de tesis

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

Tabla 4.5: Posible horarios para las defensas de 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

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

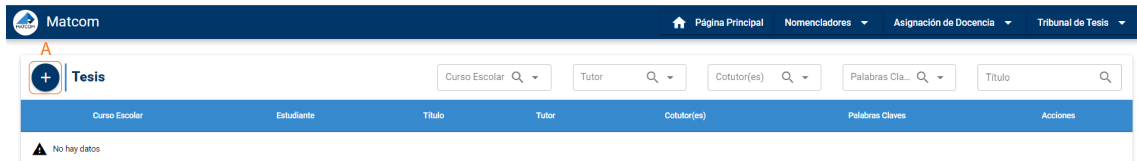


Figura 4.11: Vista de la tesis.

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 tal se muestra el resultado de agregar las dos tesis en el sistema de gestión.

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

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

Cada vez que se agregue una tesis al sistema de gestión, se crea una instancia de un tribunal de tesis con los campos de oponente y presidente sin asignar. En la figuras 4.14 se muestran los tribunales 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.

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	Gemayqzel Bouza Allende	Aymée Marrero Severo	metapoblaciones movimiento langrangiano movimiento euleriano modelos compartimentales	

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.

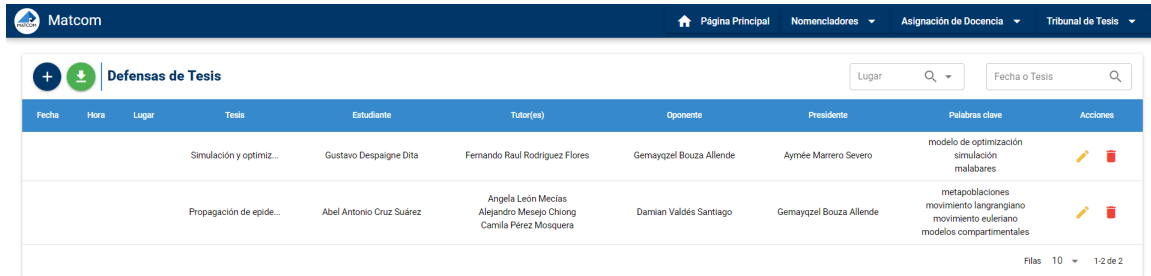
Estudiante	Título	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
Gustavo Despaigne Dita	Simulación y optimiz...	Fernando Raul Rodríguez Flores	Gemayqzel Bouza Allende	Aymée Marrero Severo	modelo de optimización simulación malabares	
Abel Antonio Cruz Suárez	Propagación de epide...	Angela León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

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

En la figura 4.17, se puede apreciar que se han actualizado las participaciones de los profesores en 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 planificación de las defensas. Cuando una tesis es agregada 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 planificación de las tesis.







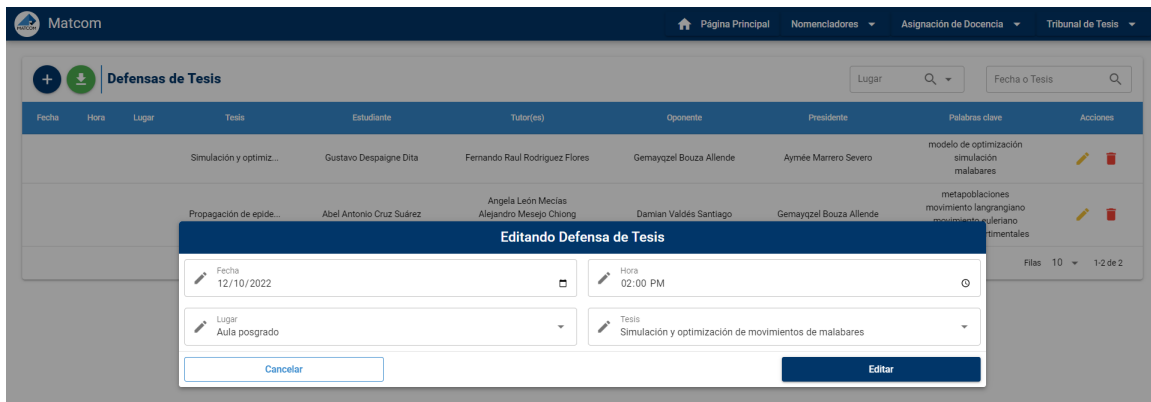
Fecha	Hora	Lugar	Tesis	Estudiante	Tutor(es)	Oponente	Presidente	Palabras clave	Acciones
			Simulación y optimiz...	Gustavo Despaigne Dita	Fernando Raul Rodriguez Flores	Gemayqzel Bouza Allende	Aymée Marrero Severo	modelo de optimización simulación malabares	 
			Propagación de epide...	Abel Antonio Cruz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

Figura 4.18: Interfaz de usuario para planificar 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

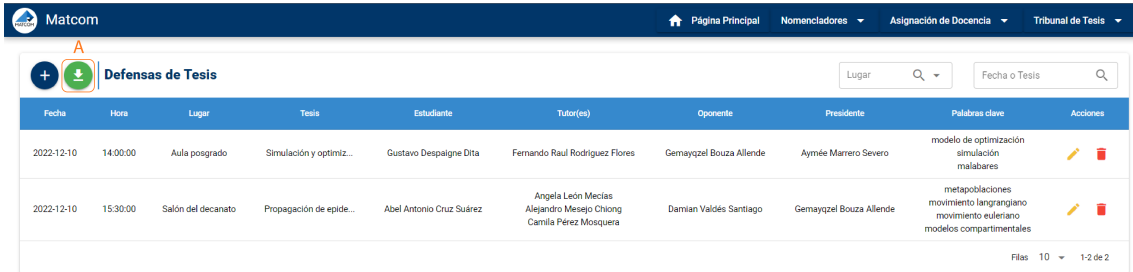
Lugar: Aula posgrado

Tesis: Simulación y optimización de movimientos de malabares

Cancelar Editar

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

Una vez se hayan realizado las planificaciones de 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 el proceso de planificación de las tesis a través del sistema de gestión 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 o cotutores de la misma hayan sido agregados previamente. 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*, *Careers*, *StudyPlans*, *CarmenTable*, *Departments*, *Subjects*, *Professors*, *SubjectDescriptions*, *TeachingAssignments*, *Places*, *Keywords*, *Thesis*, *ThesisCommittee*.

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 propone una guía de cómo agregar nuevos procesos al sistema de gestión que se implementó.

En la siguiente sección se describe cómo están estructurados los proyectos servidor y cliente.

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.

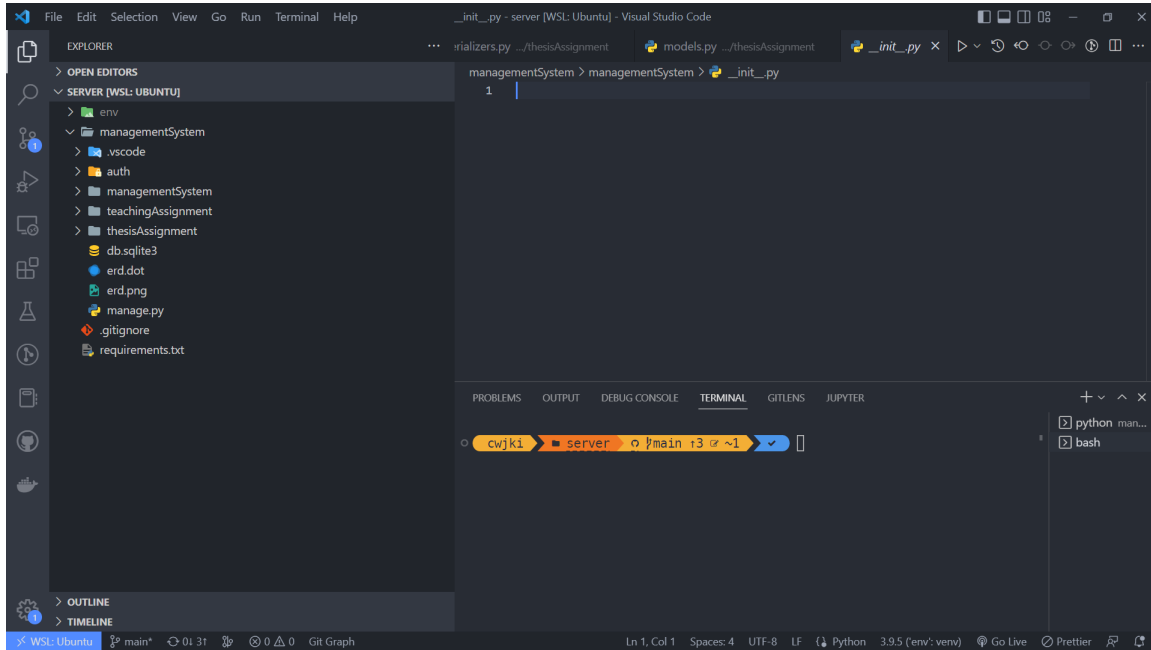


Figura 5.1: Estructura de los ficheros del proyecto servidor

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 `thesisAssignment`.
- **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.
- **teachingAssignment**: app creada para el desarrollo del proceso de planificación de las tesis. En la carpeta `excels` se implementó la funcionalidad de descargar la información de los tribunales y defensas de tesis en ficheros CSV.
- **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 inició haciendo uso de la herramienta Quasar CLI. La distribución de los ficheros quedó como se muestra a continuación.

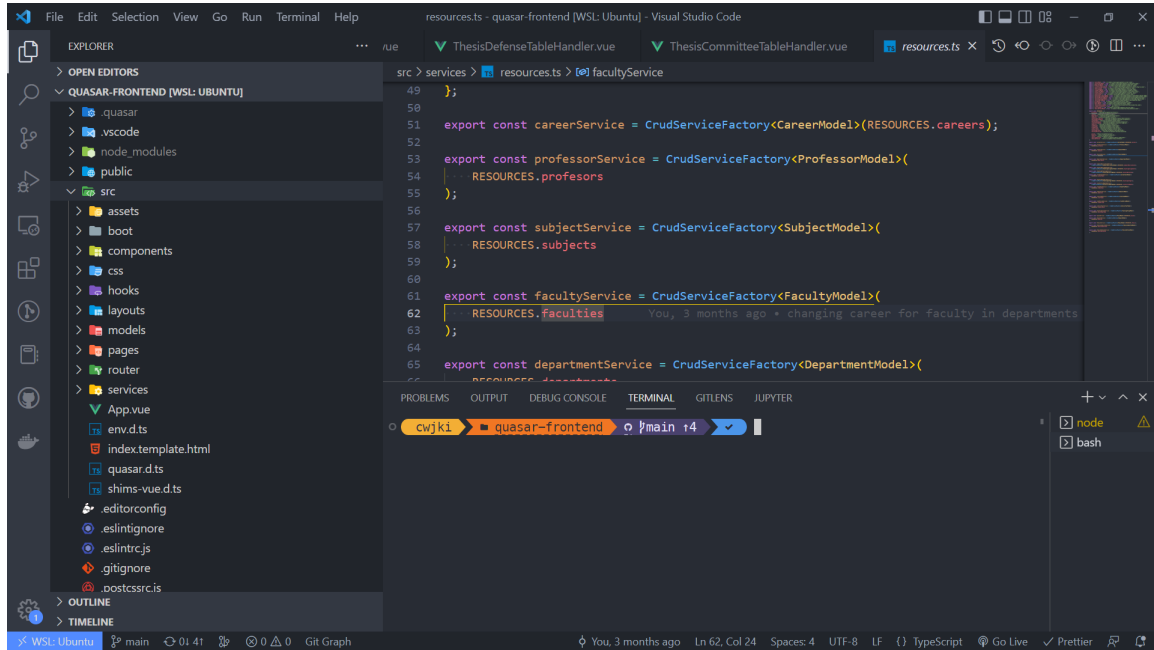


Figura 5.2: Estructura de los ficheros del proyecto servidor

A continuación se describe qué encontrar en cada una de las carpetas principales del proyecto cliente.

- **assets:** las imágenes que se utilizaron en el proyecto.
- **boot:** se configura la url base para la comunicación con el servidor.
- **components:** las componentes implementadas para el funcionamiento del proyecto. Por ejemplo se creó una componente tabla 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.
- **hooks:** la implementación de un hook para conocer qué departamento se desea administrar en cada momento y mostrar solo los datos relacionados con ese departamento. Por ejemplo, si se quiere realizar la asignación de docencia en el departamento de Matemática Aplicada, solo se deben poder asignar los profesores que pertenezcan a 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 = {
  professors: '/teaching-assignment/professors/',
  faculties: '/teaching-assignment/faculties/',
};
```

```
careers: '/teaching-assignment/careers/',
...,

};

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

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)
```

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 deben seguir para agregar un nuevo proceso en el sistema, se ejemplificará 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 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 agregar 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 `placeService`, `keywordService`, `thesisService`, `thesisCommitteeService` y `thesisDefenseService`, para realizar las peticiones al servidor.
- Se creó una carpeta `thesisCommittee` dentro de la carpeta `src/components/tables` para agrupar las componentes que se implementaron para la visualización de los datos en tablas.
- Se creó una carpeta `thesisCommittee` dentro de la carpeta `src/pages` para agrupar las vistas que 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 que permiten ir a las vistas creadas en la barra de navegación de la aplicación web.

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 de gestión. 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 crear interfaces que representen el tipo 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

Una de las funcionalidades principales que se desea es la informatización de los datos relativos a la facultad. La digitalización de datos como los planes de estudios, departamentos de la facultad, profesores, asignaturas y cualquier otra información que intervenga en los procesos que se modelen en el sistema de gestión puede ser útil tanto para el desarrollo de herramientas informativas como para el análisis y procesamiento de los datos con el objetivo de computar métricas y estadísticas (esto esta malito).

Bibliografía

- (1) Choobineh, J.; Mannino, M. y Tseng, V. en *Conceptual Modeling, Databases, and Case*; Wiley: 1992, págs. 433-445 (vid. pág. 7).
- (2) Dulanga, C. Quasar vs. Vuetify vs. Bootstrap Vue: Choosing the Right Vue.js UI Library, Accessed: 2022-04-25 (vid. pág. 12).
- (3) García Hernández, L. y Montes de Oca Richardson, M. en *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte*; Editorial Félix Varela: 2005, págs. 111-134 (vid. pág. 9).
- (4) García Hernández, L. y Montes de Oca Richardson, M. en *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte*; Editorial Félix Varela: 2005, págs. 40-64 (vid. págs. 6, 7).
- (5) García Hernández, L. y Montes de Oca Richardson, M. en *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte*; Editorial Félix Varela: 2005, págs. 65-110 (vid. págs. 7, 8).
- (6) Greif, S. Front-end Frameworks, Accessed: 2022-07-30 (vid. pág. 11).
- (7) JavaScript, Accessed: 2022-04-25 (vid. pág. 11).
- (8) Microsoft Description of the database normalization basics, Accessed: 2022-07-28 (vid. pág. 9).
- (9) Raval, N. Top 10 JavaScript Usage Statistics to Watch Out for in 2022, Accessed: 2022-04-25 (vid. pág. 11).
- (10) Vue.js The Progressive JavaScript Framework, Accessed: 2022-04-25 (vid. pág. 12).