

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 . . . . .	3
1.1.2. Tribunales de 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
<b>2. Descripción de las funcionalidades que se desean</b>	<b>13</b>
2.1. Asignación de docencia . . . . .	13
2.2. Confección de los tribunales de tesis y planificación de las defensas de tesis . . . . .	14
<b>3. Descripción del diseño de la base de datos</b>	<b>15</b>
3.1. Modelación de la asignación de docencia . . . . .	15
3.2. Modelación de los tribunales de tesis . . . . .	18
<b>4. Descripción las herramientas implementadas</b>	<b>20</b>
4.1. Asignación de docencia . . . . .	20
4.2. Confección de los tribunales de tesis y planificación de las defensas de tesis . . . . .	25
4.3. Exportar e importar datos de la base de datos . . . . .	25
4.3.1. Exportar los datos de la base de datos . . . . .	25
4.3.2. Importar datos a la base de datos . . . . .	26

<b>5. Extensibilidad</b>	<b>27</b>
5.1. Estructura del cliente y servidor . . . . .	27
5.2. Pasos para agregar un nuevo proceso en el sistema de gestión . . . . .	27
<b>Conclusiones</b>	<b>29</b>
<b>Recomendaciones</b>	<b>30</b>
<b>Bibliografía</b>	<b>31</b>



# Índice de figuras

3.1.	. . . . .	18
4.1.	Vista de la asignación de docencia . . . . .	21
4.2.	Vista de la asignación de docencia, ordenada por profesores para que aparezcan primero las asignaturas que aún no han sido asignadas . .	22
4.3.	Vista del formulario para editar una asignación de docencia . . . . .	22
4.4.	Vista del formulario para editar una asignación de docencia, mostrando los profesores posibles para la asignación . . . . .	23
4.5.	Vista de la asignación tras realizar una asignación correctamente . . .	24

## 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 utilizadas en este trabajo.

En la sección 1.1 se describe el funcionamiento de la facultad y se y en la mas cual.

### 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. A continuación se describen algunos de estos procesos.

[SI SE MANTIENE ESTO, REVISAR LA REDACCION DE LOS PROCESOS.]

**Asignación de docencia:** Consiste en dadas las asignaturas a impartir en el semestre actual, realizar una distribución de los profesores del departamento, tal que se cumplan ciertas restricciones. Por ejemplo, es necesario impartir cierta cantidad de horas de conferencia, clases prácticas u otro tipo de actividad relacionada con cada asignatura, de acuerdo al plan de estudio al que pertenezca.

**Planificación de las evaluaciones en el semestre:** Por cada asignatura que se imparte en el semestre es necesario realizar un planificación de las evaluaciones del curso. Por ejemplo, se deben definir la cantidad de evaluaciones, el tipo de evaluación: trabajos de control (TC), intrasemestral, preguntas escritas, proyectos, seminarios, entre otros y las fechas en las que se efectuarán estas evaluaciones.

**Planificación de los exámenes finales y las revalorizaciones:** Por cada asignatura es necesario realizar la planificación de los exámenes finales y revalorizaciones. Este es un proceso complejo, se debe tener en cuenta varios factores como la cantidad de locales, con sus respectivas capacidades, todas las asignaturas no tienen las mismas características, por ejemplo la asignatura de Programación requiere de dos días completos consecutivos en el laboratorio para la realización de sus exámenes. En el caso de las revalorizaciones se debe tener en cuenta las asignaturas que tengan estudiantes comunes para no programar los exámenes en los mismos horarios. Además se pueden valorar otros factores como el orden en el que se programen los exámenes, el nivel de complejidad de las asignaturas.

**Confección de los tribunales de tesis:** Como parte del proceso de graduación cada año es necesaria la planificación de los tribunales de tesis para realizar las defensas de los trabajos de grado. Para la organización de esta tarea se necesita la coordinación de los profesores que formarán parte de los tribunales de acuerdo a los horarios asignados, además estos profesores deben tener conocimientos sobre los temas que se aborden en las tesis en que participen.

Este trabajo tiene como objetivo el desarrollo de una herramienta informática que permita la integración de los procesos mencionados anteriormente, pero se centra en resolver primeramente la asignación de docencia y la confección de tribunales de tesis. A continuación se describen ...

### 1.1.1. Asignación de docencia

En la facultad de Matemática y Computación de La Universidad de la Habana (MATCOM) 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 epi-

demográfica 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 y tercer 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 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 cuales 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.

Tabla 1.1: 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 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 1.2: Fragmento de la planificación de docencia.

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

[CONSULTAR QUE PROFESORES PONER EN LA TABLA DE ASIGNACION, y si poner la categoria docente en la tabla.] Explicar que significa C y CP y las horas.

[EXPLICAR que hay asignaturas donde las conferencias para un mismo grupo se imparten por dos profesores distintos.]

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. Por ejemplo, para la asignatura Modelos de Optimización I que se rige por el plan de estudio D para la carrera de Ciencia de la Computación, que se imparte en el tercer año, se tiene un total de horas a impartir de 64, de esas, 32 horas son de conferencia y 32 horas son de clases prácticas.

### 1.1.2. Tribunales de 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.

El tribunal de tesis debe conformarse por  $n$  profesores, que asumen los roles de secretario, presidente y oponente. La cantidad de miembros de un tribunal y los roles que deben cumplir puede variar de un año a otro de acuerdo a las condiciones del

curso escolar. Los profesores deben tener dominio acerca de los temas que se aborden en la investigación. Este año los tribunales de tesis se formarán por  $n$  profesores con  $m$  roles.

El proceso de confección de los tribunales de tesis consiste en determinar cuáles profesores conformarán el tribunal de cada una de las tesis que se defienden cada año como parte del ejercicio de culminación de estudio. Una vez estén definidos (voz pasiva) los tribunales es necesario planificar la fecha, hora y lugar donde se efectuará la defensa de la tesis.

[Debo hablar de que ser oponente es una carga mucho mayor]. [Debo hablar sobre los principales problemas de realizar esa tarea de forma manual].

Para mantener toda la información relevante a los procesos de asignación de docencia y confección de los tribunales de tesis es imprescindible el uso de bases 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 [García Hernández y Montes de Oca Richardson 2005b].

### 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 [Choobineh y col. 1992].

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 [García Hernández y Montes de Oca Richardson 2005c], 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 [García Hernández y Montes de Oca Richardson 2005b].

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[García Hernández y Montes de Oca Richardson 2005c], 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 [García Hernández y Montes de Oca Richardson 2005a]. 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[ s.f.].

### 1.3. Herramientas utilizadas para el desarrollo del software

Para la informatización de los procesos de asignación de docencia y confección de los tribunales de tesis se implementa un sistema de gestión a través de una aplicación web, utilizando un modelo cliente-servidor [REF]. 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 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

[VOLVER A REDACTAR ESTE PARRAFO] El modelo cliente-servidor es uno de los más populares en el mundo de la informática actual [REF], se utiliza en el desarrollo de todo tipo de aplicaciones, incluso muchos de los protocolos utilizados a diario implementan este modelo, tales como File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) y Hypertext Transfer Protocol (HTTP).

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 de la tecnología que se utilizó en el desarrollo de la aplicación web, en el lado del servidor.

### 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 [REF]. 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. 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 Vue [REF]. 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” 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 [REF]. Este último fue el escogido para el desarrollo del cliente de la aplicación web.

## Quasar

Es una biblioteca de código abierto desarrollado para Vue.js, que permite el desarrollo 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, reutilizando el mismo código fuente. Entre los principales beneficios que Quasar ofrece se encuentran:

[ENFOCAR ESTO COMO X Q ES UTIL LO DE ABAJO PARA MI.]

- 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 en cuestión.
- 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 la informatización de los procesos de asignación de docencia, confección de los tribunales de tesis y planificación de las defensas de las tesis, que se llevan a cabo en la Facultad de Matemática y Computación de la Universidad de La Habana. Se desea además que la herramienta permita la posterior integración de los procesos que se abordan en la sección 1.1.

En las siguientes secciones se describen las funcionalidades que se desean 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:

- Realizar la asignación de docencia en un departamento, mostrando solo los datos relevantes para el mismo. Por ejemplo, cuando se realice la asignación de docencia para el departamento de Matemática Aplicada, solo se mostrarán las asignaturas y profesores del mismo.
- Conocer la carga docente de los profesores durante el proceso de asignación.
- Conocer las asignaturas que se deben impartir en un semestre a partir de un curso escolar y un período de tiempo. Por ejemplo, se desea saber las asignaturas a impartir en el curso 2021-2022 en el período de tiempo enero-julio.

- Realizar filtrados o búsquedas a partir de los datos que intervienen en la asignación de docencia. Por ejemplo, saber las asignaturas que debe impartir un profesor en un semestre dado.
- Exportar la asignación de docencia a un documento.

## 2.2. Confección de los tribunales de tesis y planificación de las defensas de tesis

Se desea implementar un mecanismo que permita la confección de los tribunales de tesis y la planificación de las defensas de las tesis. Además se desean algunas funcionalidades relacionadas con estos procesos, a continuación se describen.

- Realizar la confección de los tribunales de tesis a partir de las tesis que se deban realizar como ejercicio de culminación en un curso escolar.
- Conocer la cantidad de tribunales de los que forman parte los profesores durante el proceso de confección de los tribunales de tesis.
- Realizar filtrados o búsquedas a partir de los datos que intervienen en la confección de tribunales de tesis. Por ejemplo, saber en cuáles tribunales participa un profesor y con qué rol (secretario, oponente, presidente).
- Realizar la planificación de las defensas de tesis a partir de la confección de los tribunales de tesis.
- Realizar filtrados o búsquedas a partir de los datos que intervienen en la planificación de las defensas de tesis. Por ejemplo, saber dado una fecha cuáles tesis se defienden en la misma o el local en donde se llevará a cabo el ejercicio.
- Exportar la confección de los tribunales de tesis a un documento.
- Exportar la planificación de las defensas de tesis a un documento.

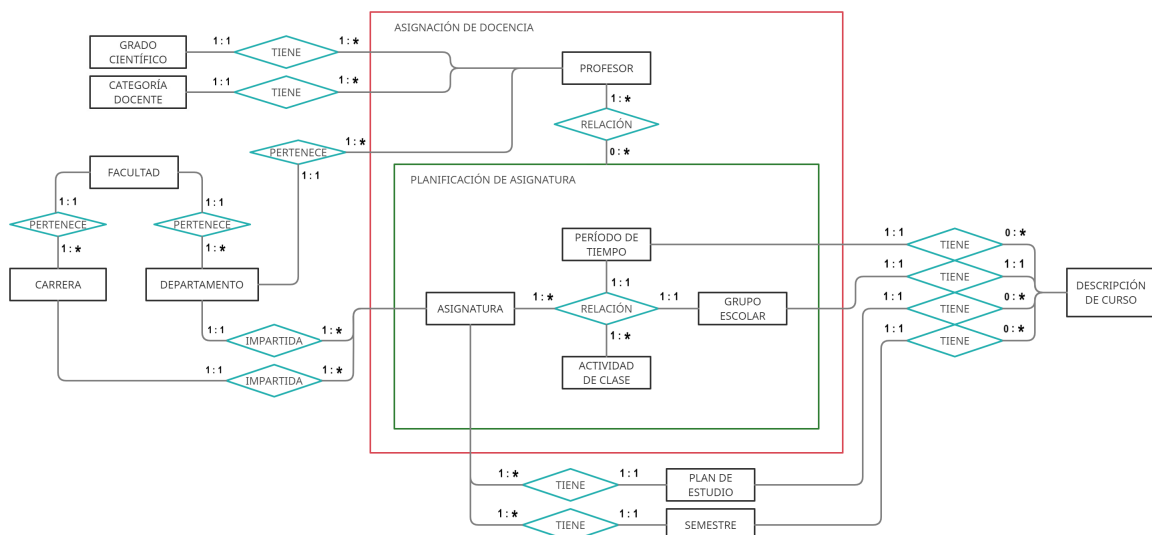


## Capítulo 3

# 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.

**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 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.

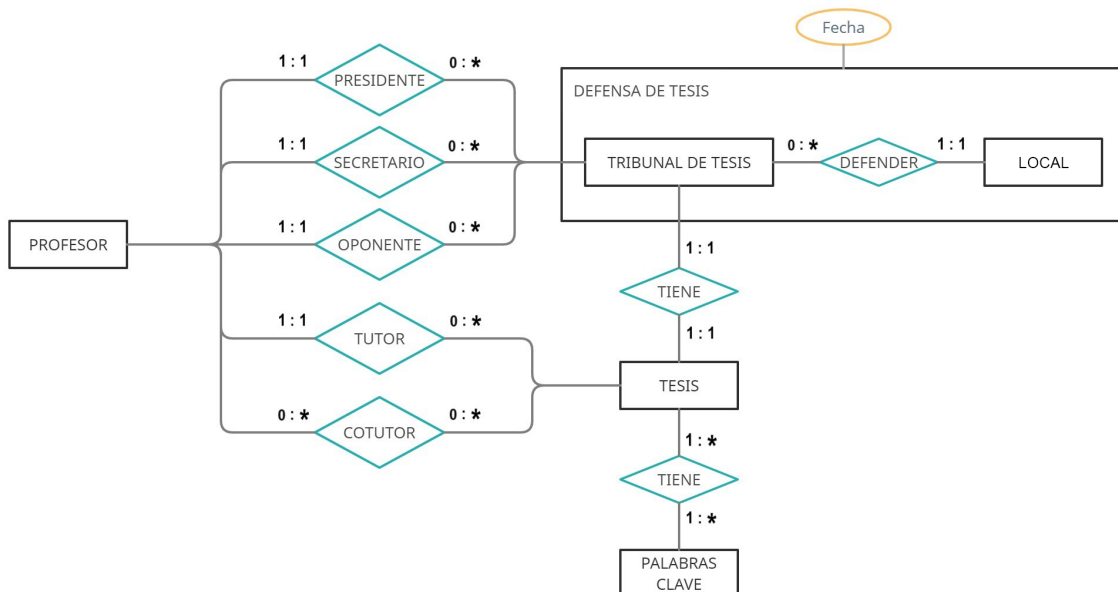


Figura 3.1

**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 y las palabras claves se modelan como relaciones con las entidades PROFESOR (tutor y cotutores) y PALABRAS CLAVES. Una tesis tiene un único tutor, puede tener cero o muchos cotutores 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.

DESPUES DE LA MODELACION DE CADA PROCESO CREO Q DEBO HABLAR UN POCO DEL PROCESO DE NORMALIZACION PARA ESOS ESQUEMAS, Y REVISAR SI ESTAN EN 3FN...

NO SE QUE MAS MENCIONAR EN ESTE CAPITULO.

## Capítulo 4

# Descripción las herramientas implementadas

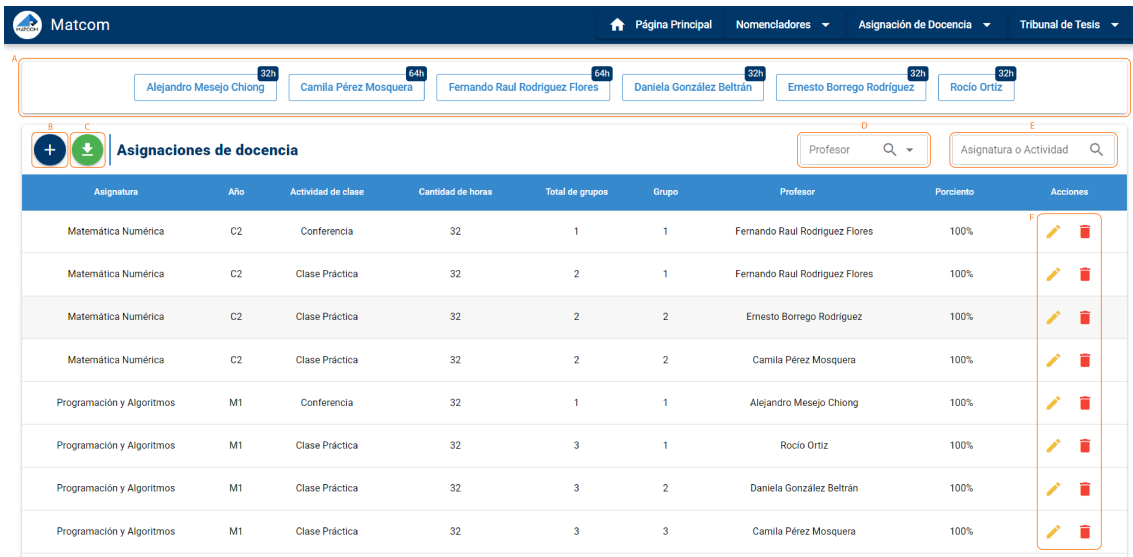
En este capítulo se presenta la propuesta de solución para la informatización de los procesos de asignación de docencia, confección de los tribunales de tesis y planificación de las defensas de tesis que se llevan a cabo en la Facultad de Matemática y Computación de la Universidad de La Habana.

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, cuyo funcionamiento se describe a continuación.

FOTO DE LA VISTA HOME ALGO COMO AQUI SE DECIDE QUE PROCESOS SE QUIEREN REALIZAR

### 4.1. Asignación de docencia

El proceso de asignación de docencia en un departamento se realiza en la interfaz que se muestra en la figura 4.1. El usuario de la aplicación puede ver la carga docente de los profesores durante el proceso de asignación, realizar filtrados por profesor para ver las asignaturas que tiene asignada y descargar un documento CSV que contiene la información de la docencia una vez haya terminado la asignación.



**Asignaciones de docencia**

Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Matemática Numérica	C2	Conferencia	32	1	1	Fernando Raul Rodriguez Flores	100%	[editar] [borrar]
Matemática Numérica	C2	Clase Práctica	32	2	1	Fernando Raul Rodriguez Flores	100%	[editar] [borrar]
Matemática Numérica	C2	Clase Práctica	32	2	2	Ernesto Borrego Rodriguez	100%	[editar] [borrar]
Matemática Numérica	C2	Clase Práctica	32	2	2	Camila Pérez Mosquera	100%	[editar] [borrar]
Programación y Algoritmos	M1	Conferencia	32	1	1	Alejandro Mesejo Chiong	100%	[editar] [borrar]
Programación y Algoritmos	M1	Clase Práctica	32	3	1	Rocio Ortiz	100%	[editar] [borrar]
Programación y Algoritmos	M1	Clase Práctica	32	3	2	Daniela González Beltrán	100%	[editar] [borrar]
Programación y Algoritmos	M1	Clase Práctica	32	3	3	Camila Pérez Mosquera	100%	[editar] [borrar]

Figura 4.1: Vista de la asignación de docencia

En la figura 4.1, se indican con recuadros cuatro regiones cuyo significado se explica a continuación.

- A: muestra la carga docente de los profesores dada la asignación actual.
- B: botón para agregar una nueva planificación de docencia.
- C: botón para descargar la asignación de docencia actual.
- D: realizar filtrado por los profesores para ver las asignaturas que tiene asignadas.
- E: realizar búsquedas por las asignaturas o actividades de clase.
- F: botones para editar o borrar una fila de la tabla.

En la figura 4.2 se muestra como quedaría ordenada la tabla a partir de los profesores, con el objetivo de mostrar primero aquellas asignaturas que no se han asignado aún.

Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Inferencia Estadística	M3	Conferencia	48	1	1			[Edit] [Delete]
Inferencia Estadística	M3	Clase Práctica	32	1	1			[Edit] [Delete]
Programación y Algoritmos	M1	Conferencia	32	1	1	Alejandro Mesejo Chiong	100%	[Edit] [Delete]
Matemática Numérica	C2	Clase Práctica	32	2	2	Camila Pérez Mosquera	100%	[Edit] [Delete]
Programación y Algoritmos	M1	Clase Práctica	32	3	3	Camila Pérez Mosquera	100%	[Edit] [Delete]
Matemática Numérica	C2	Conferencia	32	1	1	Fernando Raul Rodriguez Flores	100%	[Edit] [Delete]
Matemática Numérica	C2	Clase Práctica	32	2	1	Fernando Raul Rodriguez Flores	100%	[Edit] [Delete]
Programación y Algoritmos	M1	Clase Práctica	32	3	2	Daniela González Beltrán	100%	[Edit] [Delete]

Figura 4.2: Vista de la asignación de docencia, ordenada por profesores para que aparezcan primero las asignaturas que aún no han sido asignadas

Para ordenar la tabla a partir de los profesores se debe pulsar sobre el elemento que se indica en con el recuadro A. Para realizar una asignación se debe pulsar el botón de editar que se indica con con el recuadro B. La figura 4.3 muestra la vista después de pulsar el botón de editar.

**Editando Asignación de docencia**

Asignatura: Inferencia Estadística -- Conferencia -- M3

Grupo: 1

Profesor:

Porcentaje:

Cancelar Editar

Figura 4.3: Vista del formulario para editar una asignación de docencia



Cuando se edita una asignación de docencia es necesario definir la planificación de la asignatura (Inferencia Estadística—Conferencia—M3), grupo, profesor que se desea asignar y el porcentaje total de horas que impartirá el profesor. En la figura 4.4 se muestra como asignar a la profesora Vivian Sistaschs con un 100 porcentaje de las horas que se deben impartir en la planificación de la asignatura Inferencia Estadística.

The screenshot displays the 'Asignación de docencia' (Teaching Assignment) interface in the Matcom system. A modal window is open for editing an assignment. The dropdown menu for selecting a professor is active, showing a list of names including 'Manuel Valiente', 'Marcos Torres', 'Marelys Crespo Navas', 'Marta Lourdes Baguer Díaz-Romañach', 'Rocio Ortiz', 'Sheyla Artilles', 'Sira Allende Alonso', and 'Vivian Sistaschs'. The 'Porcentaje' (Percentage) field is set to 100. Below the form, a table lists existing assignments with columns for subject, group, activity, hours, and professor.

Asignatura	Grupo	Actividad	Horas	Profesor	Porcentaje	Acciones
Programación y Algoritmos	M1	Clase Práctica	32	Camila Pérez Mosquera	100%	[Edit] [Delete]
Matemática Numérica	C2	Conferencia	32	Fernando Raul Rodriguez Flores	100%	[Edit] [Delete]
Matemática Numérica	C2	Clase Práctica	32	Fernando Raul Rodriguez Flores	100%	[Edit] [Delete]
Programación y Algoritmos	M1	Clase Práctica	32	Daniela González Beltrán	100%	[Edit] [Delete]

Figura 4.4: Vista del formulario para editar una asignación de docencia, mostrando los profesores posibles para la asignación

La figura 4.5 muestra el resultado de completar los pasos que se muestran en la figuras 4.2, 4.3 y 4.4.

Asignatura	Año	Actividad de clase	Cantidad de horas	Total de grupos	Grupo	Profesor	Porcentaje	Acciones
Inferencia Estadística	M3	Clase Práctica	32	1	1			
Programación y Algoritmos	M1	Conferencia	32	1	1	Alejandro Mesejo Chiong	100%	
Inferencia Estadística	M3	Conferencia	48	1	1	Vivian Sistaschs	100%	
Matemática Numérica	C2	Clase Práctica	32	2	2	Camila Pérez Mosquera	100%	
Programación y Algoritmos	M1	Clase Práctica	32	3	3	Camila Pérez Mosquera	100%	
Matemática Numérica	C2	Conferencia	32	1	1	Fernando Raul Rodriguez Flores	100%	
Matemática Numérica	C2	Clase Práctica	32	2	1	Fernando Raul Rodriguez Flores	100%	
Programación y Algoritmos	M1	Clase Práctica				Daniela González Beltrán	100%	

Figura 4.5: Vista de la asignación tras realizar una asignación 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. El recuadro B refleja como quedó la fila tras la asignación de la profesora Vivian Sistaschs para que imparta las 48 horas de conferencia de la asignatura Inferencia Matemática a los estudiantes que cursan el tercer año de la carrera Matemática.

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 las 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.

[PONER OTRAS IMG DEL SITIO COMO LA VISTA DE LOS PROFESORES O PLANIFICACIÓN DE ASIGNATURAS O ESO VA PA LOS ANEXOS]

En la siguiente sección se explica el flujo para la confección de los tribunales de tesis y la planificación de las defensas de tesis.

## 4.2. Confección de los tribunales de tesis y planificación de las defensas de tesis

Para la confección de los tribunales de tesis y la planificación de las defensas de tesis, se debe partir de las tesis que se vayan a defender en un curso dado. En la figura tal se muestra la interfaz de usuario para la creación de las tesis.

[INSERTAR VISTA DE LAS TESIS]

[EXPLICAR EL FLUJO DE CREACION DE LAS TESIS]

Cada vez que se agregue una tesis al sistema de gestión, se crean dos instancias, una de TRIBUNAL DE TESIS con los campos de oponente y presidente sin asignar y otra de DEFENSA DE TESIS con los campos fecha, hora y local sin asignar. A continuación se describe el flujo para completar los procesos de confección de los tribunales de tesis y planificación de las defensas de tesis.

[IMGS PARA EXPLICAR COMO ASIGNAR EL OPONENTE Y PRESIDENTE]

[LUEGO IMG PARA EXPLICAR COMO DEFINIR FECHA HORA Y LUGAR]

## 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.

### 5.1. Estructura del cliente y servidor

El desarrollo en el lado del servidor se llevó a cabo haciendo uso de la biblioteca Django, en particular Django Rest Framework. Por tanto la creación del proyecto que contiene el servidor se hizo con el [comando de django para crear el proyecto], como resultado se obtuvo la siguiente estructura para los ficheros del proyecto.

[DISTRIBUCION DE CARPETAS DEL SERVER]

Para la modelación de los procesos se crearon aplicaciones para separar el desarrollo de cada proceso, con el [comando de crear app de django rest framework].

Luego se crearon las apps `teachingAssignment` y `thesisCommittee` para la

[AQUI EXPLICAR LA DISTRUBUCION DE LAS CARPETAS ECT]

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

[SUPONER QUE VAMOS A AGREGAR UN NUEVO PROCESO, EXPLCIAR DONDE SE DEBEN CREAR LAS COSAS EN EL BACKEND Y EN EL FRONT]

Supongamos que se quiere agregar al sistema de gestión el proceso de planificación de las evaluaciones de un semestre. A continuación se describen los pasos principales que se sugieren para la integración de este proceso.

- Crear una nueva app de django con el comando tal.
- Dentro de la carpeta creada en el paso anterior definir los modelos, serializadores, vistas y urls necesarios en ficheros nombrados `models.py`, `serializers.py`, `views.py`, `urls.py` respectivamente.
- Crear una carpeta con el nombre del proceso que se está modelando dentro de las carpetas `pages` y `components`. En la carpeta `pages` se encuentran las vistas principales y dentro de la carpeta `components` definir las componentes necesarias.
- En la carpeta `services` se implementó una api para la comunicación con el servidor. Se creó un `CRUDServiceFactory` que permite realizar pedidos de crear, listar, actualizar y borrar. Solo se necesita agregar la url y el tipo de dato en el fichero `resources`.

# 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

(s.f.). (Vid. pág. 9).

Choobineh, J., Mannino, M. & Tseng, V. (1992). The role of form analysis in computer-aided software engineering. En *Conceptual Modeling, Databases, and Case* (pp. 433-445). Wiley. (Vid. pág. 7).

García Hernández, L. & Montes de Oca Richardson, M. (2005a). DISEÑO DE BASES DE DATOS RELACIONALES. En *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte* (pp. 111-134). Editorial Félix Varela. (Vid. pág. 9).

García Hernández, L. & Montes de Oca Richardson, M. (2005b). DISEÑO DE LA BASE DE DATOS. En *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte* (pp. 40-64). Editorial Félix Varela. (Vid. págs. 6, 7).

García Hernández, L. & Montes de Oca Richardson, M. (2005c). MODELOS MATEMÁTICOS DE DATOS. En *SISTEMAS DE BASES DE DATOS: Modelación y Diseño Primera Parte* (pp. 65-110). Editorial Félix Varela. (Vid. págs. 7, 8).