

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**

**Lic. Daniela González Beltrán**

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 . . . . .	3
1.2. Descripción del software . . . . .	3
1.2.1. Modelo cliente-servidor . . . . .	4
1.2.2. Desarrollo del servidor . . . . .	4
1.2.3. Desarrollo del cliente . . . . .	5
<b>2. Descripción de las funcionalidades que se desean</b>	<b>8</b>
2.1. Informatización de la datos . . . . .	8
2.2. Asignación de docencia . . . . .	9
2.3. Confección de los tribunales de tesis . . . . .	9
<b>3. Descripción del diseño de la base de datos</b>	<b>10</b>
3.1. Modelación de la asignación de docencia . . . . .	10
3.2. Modelación de los tribunales de tesis . . . . .	11
<b>4. Descripción las herramientas implementadas</b>	<b>12</b>
4.1. Salvar y llenar datos de la BD . . . . .	12
4.1.1. save database . . . . .	12
4.1.2. fill database . . . . .	12
4.2. Generar csv . . . . .	13
4.3. Generar asignaciones de docencia . . . . .	13
<b>5. Extensibilidad</b>	<b>14</b>
<b>Conclusiones</b>	<b>15</b>





# Índice de figuras

## Ejemplos de código

# Introducción

# Capítulo 1

## Preliminares

Este capítulo provee una breve introducción a los distintos procesos que se llevan a cabo en un departamento de la Facultad de Matemática y Computación de La Universidad de La Habana, así como una descripción de las herramientas utilizadas en la solución brindada.

### 1.1. Funcionamiento de la facultad

Cada año en la Facultad de Matemática y Computación de La Universidad de La Habana se realizan un conjunto de procesos de planificación y administración relacionados con los cursos escolares. La ejecución de los procesos es delegada a los distintos departamentos de la facultad y en su mayoría, si no en su totalidad, se realizan actualmente de manera manual y no informatizada, por ejemplo:

- Asignación de docencia
- Confección de los tribunales de tesis
- Asignación de cursos optativos
- Asignación de alumnos ayudantes
- Planificación de las evaluaciones del semestre

A continuación se describen los procesos de asignación de docencia y confección de tribunales de tesis que son llevados a cabo en la Facultad de Matemática y Computación de La Universidad de La Habana:

### 1.1.1. Asignación de docencia

El proceso de asignación de docencia en un departamento consiste en, dadas las asignaturas a impartir en un semestre y el conjunto de profesores del mismo, realizar una distribución que satisfaga los requerimientos de dichas asignaturas, contemplando la experiencia y las preferencias de los profesores.

Cada asignatura tiene sus particularidades, como la cantidad de horas a impartir en conferencias, clases prácticas, seminarios, laboratorios u otras modalidades, y la cantidad de grupos de clase que reciben dicha actividad. Estas particularidades varían en dependencia de la matrícula del curso y del plan de estudio al que la asignatura pertenece.

Por ejemplo para la asignatura Modelos de Optimización I, perteneciente al 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, además se tiene que las conferencias se imparten a un solo grupo mientras que las clases prácticas a dos.

(No se si agregar algo sobre esto) Actualmente este proceso es ejecutado por una persona 'ciclano' y debe ponerse de acuerdo con los profes, muchas veces.

### 1.1.2. Tribunales de tesis

Para realizar la confección de un tribunal de tesis se necesitan definir tres incógnitas fundamentales: el local, la fecha y el tribunal. El tribunal debe estar compuesto por tres profesores que asuman los roles de secretario, presidente y oponente. Los profesores que integren el tribunal no pueden ser tutores o cotutores de la tesis en cuestión, y deben tener dominio sobre el tema que será presentado. Para la elección de la fecha y el local se debe tener en cuenta que la defensa de una tesis dura aproximadamente 45 minutos.

(No se si agregar algo sobre esto) lo mismo que en el proceso anterior, mencionar las desventajas de hacer este proceso no informatizado

## 1.2. Descripción del software

Para la informatización de estos procesos se implementa un sistema de gestión a través de una aplicación web, utilizando un modelo cliente-servidor. Para el desarrollo en el lado del servidor se hace uso del framework Django, en particular Django Rest Framework, aprovechando la versatilidad, seguridad, escalabilidad y mantenibilidad que el framework ofrece. Mientras que en el lado del cliente se hace uso del framework Quasar, que se basa en Vue.js, creando una single-page application (SPA) o aplicación de página única. A continuación se profundiza en el modelo y herramientas utilizadas.

### 1.2.1. Modelo cliente-servidor

El modelo cliente-servidor es uno de los más populares en el mundo de la informática actual, es utilizado para 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 puede ser definido 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, los accesos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda 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.

### 1.2.2. Desarrollo del servidor

Para la implementación de los servicios que consumen los clientes del sistema de gestión se hace uso del framework Django. Django es un framework web de alto nivel, gratuito y de código abierto que permite el desarrollo rápido de sitios web seguros y mantenibles. Entre las principales características del framework se encuentran:

- Django 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.
- Django 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. Al ser MVT una “loosely coupled architecture” o una “arquitectura débilmente acoplada”, permite de manera sencilla la integración de componentes desarrolladas en paralelo.

- Django 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).
- Django proporciona un Object Relational Mapper (ORM) o mapeador relacional de objetos, permitiendo 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.
- Django 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”.
- Django cuenta con extensa documentación, desde la documentación oficial hasta contenido en forma de artículos, tutoriales y cursos en línea.
- Django 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).

### 1.2.3. Desarrollo del cliente

Dentro de las tecnologías de desarrollo web más utilizadas, sin duda, JavaScript es una de las más populares en la actualidad. JavaScript es un lenguaje de programación ligero, interpretado o compilado “just-in-time” o “justo-a-tiempo” con funciones de primera clase. Es un lenguaje de programación basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa. Se usa principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejorar las interfaces de usuarios y la creación de páginas web dinámicas[REFERENCIA WIKI].

La popularidad de JavaScript para el desarrollo web ha impulsado la creación de distintos frameworks, entre los más utilizados se encuentran:

**Angular** Lanzado en 2010, Google está a cargo de desarrollar y mantener Angular. La principal filosofía tras Angular es tener todo lo que el desarrollador necesita, es un framework con muchas características integradas, tales como: validaciones de formularios, envíos de solicitudes http, enrutamiento y manejo de estados, cuenta además con herramientas adicionales como: línea de comandos, interfaces para el manejo y creación de proyectos. Debido a esto muchos han llamado a Angular una plataforma, más que un framework. Google y Wix se encuentran entre las empresas más populares que utilizan Angular.

**React** Lanzado en 2013, Facebook desarrolla y mantiene React. A diferencia de Angular, React se enfoca en ser lo más minimalista posible, especializada en el desarrollo de interfaces de usuario, describiéndose a sí misma como una biblioteca. Cuenta con muchas menos funciones integradas que Angular por lo que se terminan utilizando muchos paquetes o dependencias de terceros al desarrollar un proyecto, ya sea para enrutamiento, manejo de estados o envío de solicitudes http. Whatsapp, Instagram, Paypal, Glassdoor y BBC son algunas de las compañías populares que usan React.

**Vue.js** Lanzado en 2014, su desarrollo y mantenimiento es llevado a cabo por un equipo de colaboradores a nivel mundial, siendo Evan Yu, su principal creador, un ex ingeniero de Google. Se puede decir que Vue se encuentra entre Angular y React en cuanto a la cantidad de funciones integradas que ofrece, más que React, pero menos que Angular. Sitios web como GitLab y Alibaba están usando Vue.

COMPARACION CON POPULARIDAD Y DESARROLLO  
Graficos y datos sobre estos frameworks?

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 la simplicidad del framework, la disponibilidad de bibliotecas, componentes y materiales de aprendizaje. A pesar de que Vue.js es un framework para construir interfaces de usuario no proporciona elementos o componentes de interfaz de usuarios. Es por esto que surgen otros frameworks encima de Vue.js como una capa más de abstracción, que brindan un conjunto de componentes reutilizables y con estilo. Entre las más populares se encuentran: Vuetify, Bootstrap Vue y Quasar. Este último fue el escogido para el desarrollo del front-end del sistema de gestión.

**Quasar** Es un framework 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:

- 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.
- soporte para varios idiomas, incluyendo compatibilidad para idiomas RLT (right to left), tanto para los componentes de Quasar como para el código del desarrollador.
- amplia comunidad y documentación.
- ofrece una versión UMD (Definición de módulo unificado) para la migración de proyectos existentes sin necesidad de ningún proceso de compilación.
- promueve las buenas prácticas para mantener la seguridad de las aplicaciones desarrolladas con Quasar, en la documentación oficial se le dedica un capítulo a este tema [REF].

## Capítulo 2

# Descripción de las funcionalidades que se desean

El objetivo principal de este trabajo es la creación de una herramienta, sitio web, portal para informatizar y automatizar distintos procesos que se llevan a cabo en un departamento.

- informatización de los datos
- asignación de docencia
- confección de tribunales de tesis
- (mencionar todos los procesos que desea informatizar e automatizar el profe Fernando?)
- automatizar los procesos anteriores

La primera funcionalidad que se desea es la informatización de los datos que intervienen en estos procesos. Una vez se tengan los datos se desea realizar la asignación de docencia de un departamento, así como la confección de los tribunales de tesis.

### 2.1. Informatización de la datos

Primeramente se hace necesaria la informatización de la información que interviene en los procesos previamente mencionados, por tanto se desea implementar un mecanismo que permita ingresar datos en el sistema de gestión y que estos sean almacenados en bases de datos para su posterior uso.

## **2.2. Asignación de docencia**

Una vez el sistema cuente con todos los datos que intervienen en la asignación de docencia, se desea implementar un mecanismo que permita la asignación de profesores a tareas o actividades relacionadas con las asignatura (conferencias, clases prácticas, laboratorios, seminarios, entre otras). Se quiere además la generación de un documento CSV que agrupe esta información.

## **2.3. Confección de los tribunales de tesis**

Se desea implementar la funcionalidad de creación de tribunales de tesis, dada una tesis poder asignar los profesores que conformarán el tribunal en los roles de secretario, presidente y oponente, así como un lugar y fecha para la defensa de la misma. De igual forma se desea la generación de un documento CSV con esta información.

## Capítulo 3

# Descripción del diseño de la base de datos

Para el almacenamiento de los datos hizo uso de SQLite 3 como sistema de gestión de bases de datos relacionales, el cual viene integrada en Django por defecto.

### 3.1. Modelación de la asignación de docencia

Para la modelación de este proceso se crearon las siguientes entidades:

**Professor:** Agrupa los datos asociados a los profesores, tales como: nombre, apellidos, categoría docente, grado científico y departamento al que pertenece.

**Subject:** Agrupa los datos asociados a las asignaturas, tales como: nombre, departamento responsable de impartir la asignatura, plan de estudio, semestre en el que se imparte, cantidad de horas totales y carrera a la que pertenece.

**CarmenTable:** Entidad creada con el fin de representar el grupo escolar vigente en el año actual, cuenta con un 'TeachingGroup', semestre actual, período de tiempo y plan de estudio correspondiente. Por ejemplo representa que el grupo C3 (Computación tercer año) con el plan de estudio E, es el que esta vigente en el semestre actual.

**SubjectDescription:** Entidad necesaria para separar las distintas actividades que se imparten en una asignatura ( conferencia, clases prácticas, seminarios, laboratorios, otras ) con el fin de poder realizar la asignación. Cuenta con una 'subject', un tipo de actividad, y el CarmenTable correspondiente.

**TeachingAssignment:** Entidad que modela la asignación de docencia, posee un 'Professor', una 'SubjectDescription' y el porcentaje correspondiente a blah.

Los campos carrera, facultad, departamento, plan de estudio, grado científico, categoría docente, tipo de clase, semestre, período de tiempo y grupo escolar fueron modelados como nomencladores en las entidades Career, Faculty, Department, StudyPlan, ScientificDegree, TeachingCategory, ClassType, Semester, TimePeriod y TeachingGroup respectivamente.

## 3.2. Modelación de los tribunales de tesis

Para la modelación de este proceso se crearon las siguientes entidades:

**Thesis:** Agrupa los datos asociados a una tesis, tales son: título, estudiante, tutor, posibles cotutores y palabras claves.

**ThesisCommittee:** Entidad que representa un tribunal de tesis, cuenta con una 'Thesis', una fecha, un lugar y tres relaciones con la entidad 'Professor' para los roles de secretario, presidente y oponente.

Los campos lugar y palabras claves fueron modelados como nomencladores en las entidades 'Place' y 'Keyword'.

# Capítulo 4

## Descripción las herramientas implementadas

### 4.1. Salvar y llenar datos de la BD

En el servidor se implementó un módulo para el trabajo con la base de datos que permite salvar el estado de la base de datos en documentos csv, de igual forma permite llenar la base de datos a partir de documentos csv.

Se crearon dos comandos para ejecutar en la terminal con el fin de realizar las tareas mencionadas previamente:

#### 4.1.1. save database

El comando save database permite salvar tanto una tabla como la base de datos completa.

Para salvar solo una tabla se hace uso del argumento -m y luego el nombre de la tabla a salvar:

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

Para salvar la base de datos completa ejecutar solo el comando:

```
python manage.py save_database
```

#### 4.1.2. fill database

El comando fill database permite llenar desde documentos csv tanto una tabla como la base de datos completa.

Análogamente al comando anterior para llenar solo una tabla se hace uso del argumento -m y luego el nombre de la tabla a llenar:

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

Para llenar la base de datos completa ejecutar solo el comando:

```
python manage.py fill_database
```

Los posibles nombres de entidades a utilizar con el parámetro -m tanto para los comandos save database como fill database son:

ClassTypes, Faculties, ScientificDegrees, TeachingCategories, Semesters, Teaching-Groups, TimePeriods, Careers, StudyPlans, CarmenTable, Departments, Subjects, Professors, SubjectDescriptions, TeachingAssignments, Places, Keywords, Thesis, ThesisCommittee

## 4.2. Generar csv

Se implementó la funcionalidad de descargar las asignaciones de docencia y los tribunales de tesis en ficheros csv.

## 4.3. Generar asignaciones de docencia

Se implementó un modelo de optimización para la generación de asignaciones de docencia, pendiente computar el peso de las asignaciones (dado las preferencias de los profesores y sus habilidades)

# Capítulo 5

## Extensibilidad



# Conclusiones

Conclusiones

# Recomendaciones

Recomendaciones