



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Urban Street Mapping  
Transfer**



Presentado por Mario Hurtado Ubierna  
en Universidad de Burgos — 17 de junio  
de 2023

Tutor: Virginia Ahedo García y Jesús Manuel  
Maudes Raedo







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



Dña. Virginia Ahedo, profesora del departamento de Ingeniería de Organización, área de Organización de Empresas y D. Jesús Manuel Maudes, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Mario Hurtado Ubierna, con DNI 71365182T, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado « > ».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 17 de junio de 2023

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

Dña. Virginia Ahedo

D. Jesús Manuel Maudes





## Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

keywords separated by commas.



---

# Índice general

---

Índice general	iii
Índice de figuras	iv
Índice de tablas	v
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Retail Store Location Problem . . . . .	5
3.2. Ciencia de Redes . . . . .	5
Técnicas y herramientas	7
4.1. Herramientas de gestión de proyectos . . . . .	7
4.2. Lenguajes de programación . . . . .	8
4.3. Base de datos . . . . .	8
4.4. Librerías . . . . .	9
4.5. APIs . . . . .	11
Aspectos relevantes del desarrollo del proyecto	13
Trabajos relacionados	15
Conclusiones y Líneas de trabajo futuras	17
Bibliografía	19

---

# Índice de figuras

---

---

# Índice de tablas

---



---

# Introducción

---

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.



---

## Objetivos del proyecto

---

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.





---

## Conceptos teóricos

---

La cuestión que se busca abordar con este proyecto es el *Retail Store Location Problem*. Este consiste en la elección de la ubicación en la que colocar una nueva tienda de forma que esta le aporte el máximo beneficio posible.

### 3.1. Retail Store Location Problem

### 3.2. Ciencia de Redes

### 3.3. Técnicas

### 3.4. Métricas de Calidad



---

# Técnicas y herramientas

---

En este apartado se tratarán las distintas técnicas y herramientas utilizadas durante el desarrollo del proyecto.

## 4.1. Herramientas de gestión de proyectos

Para la planificación del proyecto se ha buscado una herramienta que soporte los elementos característicos de un desarrollo ágil como SCRUM permitiendo la creación de distintas tareas y su gestión en el tiempo de desarrollo de la aplicación.

Inicialmente se optó por Zenhub, aunque mientras se estaba desarrollando el proyecto esta pasó a requerir una licencia de pago para su uso, por lo que se substituyó esta por Zube.

### Zenhub

Zenhub se trata de una herramienta de gestión de proyectos basada en las metodologías ágiles, por lo que cuenta con funcionalidades que facilitan la creación de tareas así como su gestión. Además de crear tareas nos permite asignar a estas un *Sprint*, puntos de poker (utilizados en SCRUM) o poder definir aquellas tareas que lo requieran como *Epics* para diferenciarlas de las otras.

Un punto importante para su elección inicial fue que esta cuenta con integración en *GitHub*, por lo que facilita la gestión de tareas al hacerla directamente desde el propio repositorio.

## Zube

Una vez Zenhub pasó a ser una herramienta de pago se optó por el uso de Zube. Esta cuenta con muchos de los elementos que Zenhub también tiene. Permite definir tareas y gestionarlas mediante un tablero *Kanban*, colocando las tareas en un area u otra en función de su estado (Backlog, Ready, In Progress,...)

## 4.2. Lenguajes de programación

Una elección bastante significativa para el proyecto será el lenguaje de programación empleado para su desarrollo.

### Python

Python se trata de un lenguaje interpretado de alto nivel bastante utilizado en ámbitos como el *Machine Learning*. Además de lo anterior cuenta con multitud de librerías de diversos propósitos desarrolladas por los propios usuarios, pudiendo facilitar las labores a realizar en el proyecto. En su elección también influye que es uno de los lenguajes que más hemos utilizado durante la carrera.

### JavaScript

Al igual que Python, JavaScript se trata de un lenguaje interpretado, pero este es utilizado por los navegadores para poder crear páginas web dinámicas. También cuenta con diversas librerías con funcionalidades que permiten manejar el DOM, hacer distintas visualizaciones, entre otras. Este lenguaje será de gran importancia en el apartado web del proyecto.

## 4.3. Base de datos

Dada la naturaleza del proyecto y los datos que se manejarán, una base de datos no relacional puede dar ciertas facilidades en el manejo de estos. También cabe destacar que la información tomará forma de grafo y se tratará usando técnicas propias de la ciencia de redes, por lo que estas necesidades tomarán gran importancia en la decisión de la base de datos.

## Neo4j

Neo4j se trata de una base de datos no relacional de la familia de los grafos escrita en Java. Esta es utilizada por grandes empresas como Intel, Adobe, AstraZeneca o incluso la NASA. También cabe destacar su uso en el terreno de la ciencia de datos gracias a *plugins* con utilidades propias de la ciencia de redes.

Cabe mencionar que cuenta con un lenguaje de consultas propio, *Cypher*, con una sintaxis que facilita la obtención de información en un contexto de grafos.

Expuestos los hechos anteriores, Neo4j se trata de la elección lógica para un proyecto de nuestras características.

### APOC (Awesome Procedures On Cypher)

Como se ha mencionado antes, Neo4j cuenta con *plugins* que facilitan algunas labores en la base de datos. Entre ellos se encuentra APOC, que tiene tanto utilidades un tanto más generales, así como algunas más específicas, como por ejemplo cargar datos directamente de peticiones a una API, que nos será bastante útil en la carga de ubicaciones durante el desarrollo del proyecto.

## 4.4. Librerías

Como se ha mencionado anteriormente, se hará uso de diversas librerías para las labores del proyecto, tanto de Python como de JavaScript.

### Driver de Neo4j

Para el manejo de la base de datos de Neo4j a través de Python existen 3 distintos drivers:

**Driver Oficial** Se trata del driver oficial de Neo4j para Python. Cuenta con cursos para su uso y una extensa documentación.

**Py2Neo** Es un driver creado por la comunidad de Neo4j con una interfaz sencilla. Lamentablemente ya no recibe soporte para las últimas versiones de Neo4j.

**Neomodel** Un driver alternativo al oficial que ofrece un OGM (Object Graph Mapper), similar a los ORMs de otras bases de datos; además de integración con el framework web *Django*.

Dado que Neo4j es una herramienta nueva, que requiere de un proceso de aprendizaje, se ha optado por el driver oficial por su simple uso y la gran cantidad de documentación existente. Se ha desestimado Py2Neo puesto que ya no recibe soporte; y en el caso de Neomodel, las funcionalidades que trae no aportan una ventaja respecto al driver oficial, puesto que en primer lugar no utilizaremos Django, sino Flask para la parte web del proyecto; y en cuanto al OGM que provee este no nos es necesario puesto que no requeriremos de las ventajas que este nos podría aportar.

## OSMPythonTools

OSMPythonTools se trata de una librería que facilita el acceso a la información de los servicios de *Open Street Map* mediante una interfaz simple.

## Flask

Flask se trata de un framework de desarrollo web para Python. Está caracterizado por ser ligero y flexible, además de contar con librerías que expanden sus funcionalidades. Permite el uso de distintas bases de datos además de contar con un motor de plantillas llamado *Jinja2* para el renderizado de páginas web.

Se ha escogido este framework frente a otros debido a su simplicidad y fácil uso, además cubre todas las necesidades del proyecto. Otro punto significativo en su elección es que ya se había trabajado con él durante la carrera en la asignatura de *Diseño y mantenimiento del Software*.

## Flask JWT Extended

Es una librería que extiende las funcionalidades de Flask para poder utilizar JWT (JSON Web Token) para las labores de autenticación de usuarios y en la seguridad de la aplicación.

## Flask WTForms

Extensión de Flask para la creación de formularios web mediante una interfaz simple, permitiendo la validación de estos desde el propio Python.

## Leaflet.js

Se trata de una librería de código abierto de JavaScript que permite la visualización de mapas interactivos en páginas web. Para sus mapas puede utilizar información extraída de *OpenStreetMap* para hacer las visualizaciones. Nos servirá para mostrar las distintas ubicaciones que almacenemos en la base de datos en la web.

## Vis.js

Es una librería de JavaScript cuyo objetivo es realizar visualizaciones interactivas de grafos en la web. Permite personalizar las visualizaciones con distintas configuraciones. Con esta podremos mostrar la información de la base de datos en forma de grafo en la forma que se haría desde el punto de vista de la ciencia de redes.

## Bootstrap 5

Se trata de un framework que se suele utilizar para la creación de interfaces web *responsive*. Cuenta con una extensa documentación y ejemplos para su uso. Facilita el diseño de páginas web ofreciendo componentes reutilizables pudiendo utilizarse directamente en las plantillas HTML.

## 4.5. APIs

Una parte importante de este proyecto será la información de las ubicaciones de las ciudades escogidas para el proyecto. Para ello se obtendrán mediante APIs de información geográfica, en este caso se utilizará OpenStreetMap para este propósito.

OpenStreetMap se trata de un proyecto abierto y colaborativo que cuenta con información geográfica recopilada por los propios usuarios para la visualización de mapas, rutas de navegación y demás.

Para este proyecto se utilizarán las siguientes APIs de este proyecto.

## Overpass

Se trata de una de las APIs que componen el proyecto de OpenStreetMap, siendo esta solo de lectura de datos. Está optimizada para las operaciones de lectura, contando con un lenguaje propio de consultas llamado *Overpass QL*,

además de introducir algunas estructuras con respecto a OpenStreetMap, como es el caso de las áreas, facilitando el acceso a la información.

## **Nominatim**

Nominatim es un motor de búsqueda que forma parte de OpenStreetMap. Sus funcionalidades están centradas en la geolocalización en base a direcciones, nombres de ubicaciones y coordenadas. Cuenta con una API propia que será de utilidad para encontrar las áreas de las distintas ciudades con las que se trabajarán en el proyecto.



---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.



---

## Trabajos relacionados

---

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.



---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.



---

## **Bibliografía**

---