



*ugr* | Universidad  
de **Granada**

# TRABAJO DE FIN DE GRADO INGENIERÍA EN INFORMÁTICA

GRANADA

(BORRADOR A FECHA DE) 25 DE AGOSTO DE 2019

---

## Encaminamiento con GeoPackage

Sistemas de Información Geográfica

### Autor

ALEJANDRO NÚÑEZ PÉREZ

### Tutor

JOSÉ SAMOS JIMÉNEZ



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS  
INFORMÁTICA Y DE TELECOMUNICACIÓN



## *Agradecimientos*

*Quiero agradecer este trabajo, culmen de mi estancia en la Universidad de Granada, tanto a mis familiares más cercanos, mi madre y mi hermano, por estar siempre ahí y apoyarme cuando lo necesitaba, como a todos los amigos que he hecho durante estos años en la Universidad por hacer de estos una experiencia inolvidable y a todos mis profesores, indistintamente de mi opinión personal, por compartir conmigo sus conocimientos, resolver mis dudas y hacerme crecer como estudiante, con mención de honor a mi tutor por su tiempo y ayuda con su supervisión. También, una sincera muestra de gratitud a cada uno de los lectores del documento, pues hubiese sido una gran pena que este esfuerzo cayese en el olvido.*



## Resumen

El objetivo de este proyecto consiste en llevar a cabo un análisis de viabilidad y desarrollo de una prueba de concepto para dotar de características de encaminamiento al estándar *GeoPackage*, un contenedor de datos geográficos.

Para lograr dicha tarea, se ha utilizado herramientas de software libre, así como el desarrollo de una pequeña biblioteca que, dado una estructura similar a un grafo de nodos y enlaces, gestiona la carga, tratamiento y manipulación de datos geográficos para obtener un camino entre dos puntos.

**Palabras Clave** encaminamiento, geopackage, información geográfica, qgis, pgRouting, gis, networkx, python, openstreetmap, geopandas

## Abstract

The goal of this project is to analyze the viability and to develop a proof-of-concept system of adding routing capabilities to the *GeoPackage* standard, a geospatial data container.

To achieve said goal, a variety of free software tools were used, as well as the development of a small library that, given a graph-like structure of nodes and links, is able to load, process and manipulate geospatial data to obtain a route between two points.

**Keywords** routing, geopackage, geospatial information, qgis, pgrouting, gis, networkx, python, openstreetmap, geopandas



# Índice general

Índice general	1
Índice de figuras	3
Índice de listados	4
Índice de cuadros	5
<b>1. Introducción</b>	<b>6</b>
1.1. Objetivos . . . . .	7
1.2. Motivación . . . . .	7
1.3. Estructura del documento . . . . .	8
<b>2. Sistemas de Información Geográfica</b>	<b>10</b>
2.1. Información Geográfica . . . . .	10
2.1.1. Modelo Ráster . . . . .	11
2.1.2. Modelo Vectorial . . . . .	12
2.2. Contenedores de Datos . . . . .	13
2.2.1. <i>ESRI Shapefile</i> . . . . .	13
2.2.2. <i>PostgreSQL</i> y <i>PostGIS</i> . . . . .	14
2.2.3. <i>GeoPackage</i> . . . . .	14
<b>3. Análisis de Redes</b>	<b>16</b>

## ÍNDICE GENERAL

3.1.	Teoría de Grafos . . . . .	16
3.2.	Topología . . . . .	17
3.3.	Encaminamiento . . . . .	19
3.3.1.	<i>pgRouting</i> . . . . .	20
3.3.2.	<i>NetworkX</i> . . . . .	20
<b>4.</b>	<b>Encaminamiento con GeoPackage</b>	<b>21</b>
4.1.	Obtención de los Datos . . . . .	22
4.2.	Transformaciones y Filtrado . . . . .	23
4.3.	Estructura y Empaquetado . . . . .	25
4.4.	Preparación de la Topología . . . . .	26
4.5.	Aplicación del Encaminamiento . . . . .	29
4.6.	Conclusiones . . . . .	31
<b>5.</b>	<b>Conclusión y Trabajo Futuro</b>	<b>33</b>
5.1.	Objetivos . . . . .	33
5.2.	Trabajo Futuro . . . . .	34
<b>A.</b>	<b>Planificación del Proyecto</b>	<b>36</b>
<b>B.</b>	<b>Integración con <i>QGIS</i></b>	<b>39</b>
<b>C.</b>	<b>Bibliografía y Referencias</b>	<b>41</b>
	Lectura Adicional . . . . .	41
	Recursos Electrónicos . . . . .	41

# Índice de figuras

2.1. Representación de datos geográficos de tipo ráster . . . . .	11
2.2. Tipos básicos de los elementos vectoriales . . . . .	12
3.1. Navegabilidad en los grafos dirigidos . . . . .	17
3.2. Topología geográfica y modificación . . . . .	18
3.3. Encaminamiento en un grafo dirigido . . . . .	19
4.1. Extensión de los datos originales . . . . .	23
4.2. Operación de recorte sobre datos geográficos . . . . .	24
4.3. Filtrado por atributos específicos . . . . .	24
4.4. Definición mínima de las entidades de una red . . . . .	26
4.5. Descomposición topológica de una red . . . . .	28
4.6. Topología resultante, visualizada con el software <i>QGIS</i> . . . .	29
4.7. Camino más corto (distancia) entre dos puntos . . . . .	30

# Índice de listados

4.1.	Algoritmo para generar una topología . . . . .	27
4.2.	Carga de los datos de <i>OpenStreetMap</i> con <code>gpkgRouting</code> . . . . .	28
4.3.	Definición de los datos finales obtenidos por <code>gpkgRouting</code> . . .	29
4.4.	Encaminamiento con <i>GeoPackage</i> y <i>NetworkX</i> . . . . .	31
B.1.	Integración con QGIS . . . . .	40

# Índice de cuadros

A.1. Planificación general del proyecto . . . . .	36
A.2. Planificación por actividad . . . . .	37

# Capítulo 1

## Introducción

Desde siempre me han fascinado los mapas y todo aquello representable en estos, lo cuál me llevó a cursar la asignatura de SISTEMAS DE INFORMACIÓN GEOGRÁFICA, dónde realmente comprendí el inmenso potencial que la información geográfica tiene como representación visual de la realidad.

Podemos decir que la información geográfica es aquella compuesta por datos geográficos y metadatos de estos así como datos no geográficos con los que guarda relación. Los datos geográficos suelen ser modelados como elementos *vectoriales* (puntos, líneas o polígonos), o de tipo *ráster*, en forma matricial (colores codificados en matrices *RGBA* o *CMYK*). Este trabajo se centra en el primer tipo, los datos vectoriales.

Por otra parte, la primera vez que oí del concepto teórico de grafo fue en la asignatura de LÓGICA Y MÉTODOS DISCRETOS, momento en el que pensé en su obvio uso en aplicaciones como *Google Maps* para buscar una ruta entre dos o más puntos. Esto también me pareció algo muy interesante y con mucho potencial.

En su forma más primitiva, un grafo consiste en un conjunto de *nodos* y *enlaces* (o conexiones) entre nodos, en nomenclatura del *Análisis de Redes*. Cada nodo o enlace, a su vez, puede contener información adicional, como por ejemplo un *coste* asociado. En el caso de que los enlaces tengan restricciones en la navegabilidad entre nodos, el grafo se denomina dirigido o digrafo.

Este trabajo consiste en la unión del *Análisis de Redes* con los *Sistemas de Información Geográfica*. Concretamente, en la creación de una topología de red en el formato *GeoPackage*, donde podremos aplicar algoritmos de encaminamiento, búsqueda de camino.

## 1.1. Objetivos

Como objetivo principal se pretende llevar a cabo una prueba de concepto, un análisis del proceso a seguir para obtener una topología de red con datos geográficos almacenados en un archivo *GeoPackage* sobre los que poder realizar encaminamiento.

Además de este objetivo principal, se plantea una serie de objetivos secundarios a conseguir en la elaboración del trabajo, listados a continuación sin ningún orden en particular:

- Aprender cómo se estructura la base de datos de *OpenStreetMap* y cómo usar este como origen de datos para posibles análisis de información de carácter geográfico.
- Conocer en mayor profundidad qué tipos de contenedores de datos geográficos existen, aquellos que son usados en la industria y qué ventajas e inconvenientes pueden tener.
- Descubrir nuevas herramientas para la creación, manipulación y visualización de datos geográficos en un entorno general.
- Indagar en la manipulación de datos geográficos desde un entorno PYTHON, así como aprender qué herramientas existen y son de uso frecuente en el ecosistema.

## 1.2. Motivación

Al cursar la asignatura de SISTEMAS DE INFORMACIÓN GEOGRÁFICA, realicé un estudio y una presentación sobre *GeoPackage* como tecnología. Al ver las características de este, pensé que era un estándar muy prometedor aunque joven, con poco más de cinco años desde su primera aparición. Al ser gestionado por el *Open Geospatial Consortium*, goza de ciertas particularidades que la hace preferible a otras tecnologías, como puedan ser aquellas propias de *ESRI*, una empresa muy conocida en el mundo de los sistemas de información geográfica.

Después de, en la misma asignatura, al estudiar *pgRouting* para realizar trazado de rutas entre varios puntos, pensé que había similaridad entre ambas tecnologías, solo que esta dependía de una base de datos relacional específica,

### **1.3. ESTRUCTURA DEL DOCUMENTO**

---

*PostgreSQL* con la extensión *PostGIS*, y que *GeoPackage* no tenía ningún soporte aparente.

Como ventajas frente a *pgRouting*, realizar encaminamiento sobre *GeoPackage* permitiría la búsqueda de caminos en dispositivos con capacidades reducidas o sin conexiones a alguna red. Las principales desventajas radican en la ausencia de una biblioteca especializada en esta tarea, aunque existe la posibilidad de usar un conjunto de herramientas para cada subtarea e integrarlas bajo un único software.

## **1.3. Estructura del documento**

El documento sigue una estructura en cinco capítulos, cada uno con un tema concreto a tratar. Los capítulos dos y tres contienen contexto relevante para comprender el desarrollo principal, el cuál se encuentra en el capítulo cuatro.

Para comenzar, el Capítulo 1, este que está leyendo, sirve de *Introducción* y está dedicado a dar un desglose general de qué se puede encontrar en el documento, así como contextualizar el desarrollo de este con unos objetivos y qué ha motivado su desarrollo.

A continuación, el Capítulo 2 pretende dar a conocer de forma superficial qué son los *Sistemas de Información Geográfica*, así como qué es la información geográfica, qué particularidades tienes y cómo se representa, y además los contenedores de datos donde podemos almacenarla.

Después, el Capítulo 3 trata sobre el *Análisis de Redes* desde un enfoque geográfico y topológico. Así pues, se explica superficialmente la teoría de grafos y la topología como tal, para finalmente explicar qué es el encaminamiento y qué relación guarda con la información geográfica.

Entonces, el Capítulo 4 contiene el grueso del documento, objetivo principal del trabajo, y explica todo el proceso a realizar, el *Encaminamiento con GeoPackage*, llámeselo generar una topología de red, almacenarla en un *GeoPackage* y aplicar algoritmos de búsqueda de camino para obtener rutas entre dos o más puntos. Este capítulo es un híbrido teórico-práctico donde se explica, por apartados, primero el caso general y segundo un caso específico, como prueba de concepto.

Por último, el Capítulo 5, de nombre *Conclusión y Trabajo Futuro*, trata

### 1.3. ESTRUCTURA DEL DOCUMENTO

de recopilar todas las conclusiones obtenidas en el desarrollo del documento, así como un desglose de cómo se han completado los objetivos y aquellos posibles trabajos futuros para llevar a cabo sobre este trabajo.

Además, están presente dos apéndices que, aunque no estén contenidos en la temática principal, pueden considerarse relevantes.

El Apéndice A, titulado *Planificación del Proyecto* pretende contextualizar el desarrollo del trabajo, cómo se ha subdividido las tareas a realizar, así como las proporciones de tiempo dedicado. Este apéndice no podría haberse realizado de no haber cursado asignaturas como DESARROLLO Y GESTIÓN DE PROYECTOS o METODOLOGÍAS DE DESARROLLO ÁGILES.

El Apéndice B, de carácter más técnico, explica como realizar una *Integración con QGIS*, un software *GIS*<sup>1</sup>, del desarrollo principal para poder visualizar los caminos encontrados. Este apéndice contiene un *script* de código que muestra la sencilla integración con PYTHON, un lenguaje de programación muy utilizado en el desarrollo.

---

<sup>1</sup>GIS es la abreviación preferida al hablar de *Sistemas de la Información Geográfica*, del inglés *Geographical Information Systems*

# Capítulo 2

## Sistemas de Información Geográfica

---

**Resumen** Partimos con la explicación de qué es y de qué se compone la información geográfica, así como explicando los dos principales modelos de representación. Después, detallamos cómo se almacena esta y aquellos contenedores de datos más comunes.

---

### Introducción

En este capítulo, se pretende dar a conocer sobre los *GIS, Sistemas de la Información Geográfica* (del inglés *Geographical Information Systems*), formado por aquellas herramientas que trabajan con información geográfica, sea manipulación, almacenamiento o visualización.

#### 2.1. Información Geográfica

La información geográfica es aquella que está georreferenciada, es decir, que tiene alguna referencia geográfica asociada. Es inmediato la posible división de esta en dos, las componentes espacial y temática [Ola14, pág. 69].

## 2.1. INFORMACIÓN GEOGRÁFICA

La componente temática corresponde a todo dato no espacial que contextualice, aporte más información sobre qué representa esa marca geográfica, mientras que la componente espacial tiene una estructura a modelos en tres niveles[Ola14, pág. 80]:

**Modelo Geográfico** Modelo conceptual de la realidad geográfica y el comportamiento de esta.

**Modelo de Representación** Conjunto finito de elementos que representa las características del modelo anterior.

**Modelo de Almacenamiento** Esquema de cómo almacenar los distintos elementos usados para la representación.

### **2.1.1. Modelo Ráster**

Es un modelo en el que los datos están ordenados en cuadrillas o celdas uniformemente espaciadas. En cada celda se encuentra un valor, que representa algo, generalmente una clasificación o un valor dentro de un espectro.

132.90	133.90	135.60	139.20	140.60	144.40
126.80	128.50	131.40	133.90	135.90	139.90
122.30	124.10	125.80	128.50	131.80	135.60
118.70	119.90	121.90	124.30	128.20	131.50

Figura 2.1: La información tipo ráster suele ser representada como una matriz numérica o alfanumérica. Fuente: [Ola14, pág. 86]

Esta propiedad de separación uniforme es muy interesante pues, aún siendo rígida como composición, se puede usar para afinar la granularidad de los datos, el ámbito de estos.

## 2.1. INFORMACIÓN GEOGRÁFICA

---

Posibles usos para estos puede incluir clasificación o Modelos Digitales del Terreno que definan elevación, sentido de bajada de flujos como la lluvia, entre otros.

### 2.1.2. Modelo Vectorial

El modelo vectorial es aquel en el que la información se representa con formas geométricas. Esto, en contraposición al modelo anterior, nos permite mayor flexibilidad a la hora de dar representación. Esto, por ejemplo, nos permite asignar clasificaciones a zonas de diferente tamaño y forma.

Existen tres primitivas de tipo vectorial [Ola14, págs. 92-95]: puntos, líneas y polígonos.

**Puntos** Elementos compuestos por coordenadas espaciales.

**Líneas** Sucesión ordenada de puntos. Pueden ser cerradas o abiertas, dependiendo de si coincide o no el primer y último punto.

**Polígonos** Línea cerrada que delimita un área. Puede tener o no líneas interiores que delimiten un hueco vacío.

Estos elementos pueden almacenarse formas textuales o binarias, respectivamente, en *WKT* (WELL-KNOWN TEXT) o *WKB* (WELL-KNOWN BINARY).

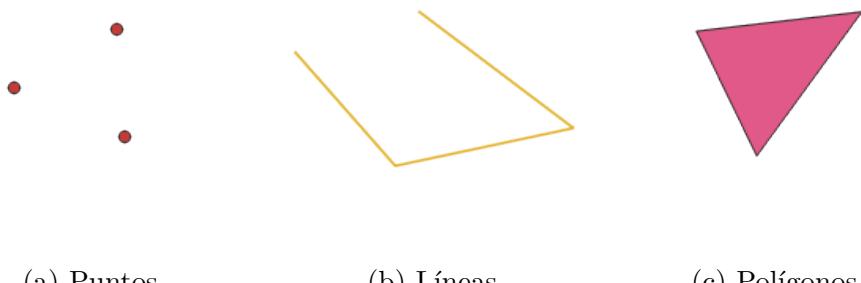


Figura 2.2: Elementos vectoriales primitivos, los bloques de construcción básicos que definen la información geográfica vectorial.

## 2.2. Contenedores de Datos

Se acostumbra a almacenar la información geográfica en bases de datos [Ola14, págs. 203-227] bien definidas, minimizando los recursos necesarios para su almacenamiento y facilitando su uso y transmisión, así como liberando de posibles dependencias a aplicaciones concretas.

Dentro del sector, existen varios estándares, entre ellas bases de datos planas como *Shapefile* de *ESRI*, *GeoJSON*<sup>1</sup> e incluso *CSV*<sup>2</sup>. Por otra parte, existen bases de datos relacionales, como *Microsoft SQL Server* o *PostgreSQL* con su extensión *PostGIS*. En un punto intermedio, podemos encontrar el llamativo y novedoso *GeoPackage*, sujeto de estudio en este trabajo.

### 2.2.1. *ESRI Shapefile*

*Shapefile*, desarrollado por *ESRI*<sup>3</sup>, es un formato vectorial en el que los datos se dividen en varios archivos.

Aunque su uso es muy extenso incluso llegándose a considerarse *lingua franca* del mundo *GIS*, este formato es, en realidad, una reliquia del pasado. Desarrollado para un uso interno del software *ArcView* de *ESRI*, el formato era muy útil y novedoso, y gran cantidad de información fue codificada en este al liberarse la descripción técnica<sup>4</sup> para que otros *GIS* pudiesen usarlos.

Dado su origen, en propósito y época, el formato tiene muchas limitaciones como:

- Existe un máximo de almacenamiento, 2GB de espacio, así que no es utilizable en casos de grandes colecciones de información.
- El nombre de los atributos, de las columnas está restringido a 10 caracteres de máximo.
- Es un formato *ASCII* y no acepta *Unicode* para codificar idiomas que no comparten el alfabeto inglés.

---

<sup>1</sup>GeoJSON: Extensión geográfica del formato de serialización JSON. <https://geojson.org/>

<sup>2</sup>CSV: Valores separados por coma, formato muy sencillo y de temática agnóstica.

<sup>3</sup>ESRI: Empresa dominante en la industria, con grandes aportes e influencia, así como desarrolladores de *ArcGIS*. <https://www.esri.com/>

<sup>4</sup>Descripción técnica de Shapefile. <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

## 2.2. CONTENEDORES DE DATOS

---

- Existe un máximo de 255 atributos.
- Los números decimales son almacenados como texto, y por tanto pueden contener errores de redondeo.

En esencia, *Shapefile* es una base de datos plana muy capaz, pero con severas limitaciones que causan molestia o imposibilitan ciertas tareas.

### 2.2.2. *PostgreSQL* y *PostGIS*

*PostgreSQL* es una base de datos relacional muy extensible y con gran cantidad de características, aunque no goza de soporte *GIS* de base. Para eso, está la extensión *PostGIS*, que dota a una base de datos *PostgreSQL* con soporte para interpretar datos vectoriales, textuales o binarios, así de múltiples funciones *SQL* para trabajar con datos geográficos en consultas u operaciones [OS15, págs. 6-10].

En definitiva, *PostGIS* (como extensión de *PostgreSQL*) es un software muy capaz, proporcionando una base de datos relacional con características *GIS* pero que no tiene cabida en ciertos entornos más limitados en recursos, como terminales móviles. Otra base de datos reseñable que compite con *PostGIS* es *SQL Server* de *Microsoft*, aunque también comparte las mismas pegas.

### 2.2.3. *GeoPackage*

*GeoPackage*, como estándar de la *OGC*, se halla en un punto intermedio entre ambos, digamos una adaptación moderna de *Shapefile* puesto que es una base de datos contenida en un único fichero con similitud a *PostgreSQL* ya que dicho fichero es en realidad una base de datos relacional *SQLite*<sup>5</sup>.

Al contenerse en un solo archivo, su almacenamiento y copia es sumamente sencillo, simplificando el envío masivo de información por, por ejemplo, correo electrónico. En contraposición, al ser un archivo binario, es complicado su serialización y transmisión por la red, ya que no hay puntos bien delimitados donde empiece y acabe la estructura.

---

<sup>5</sup>*SQLite*: Base de datos relacional de dominio público cuya principal virtud es su existencia en un único archivo. <https://www.sqlite.org>

## 2.2. CONTENEDORES DE DATOS

Al tratarse de una base de datos relacional, esto le permite relacionar diferentes entidades relacionadas entre sí, como por ejemplo datos de las calles de un municipio y la descomposición topológica de dichas calles. Por desgracia, como es un simple contenedor, a diferencia de herramientas como *Spatialite*<sup>6</sup> o *PostGIS*, no dispone de tantas funciones para operar con primitivas, y tiene una fuerte dependencia en las herramientas.

En esencia, la mayor limitación que tiene *GeoPackage* es simplemente su corta edad. Esto es así porque, frente a otros competidores, no existen tantas herramientas que trabajen con *GeoPackage* como formato. La solución para esto es simple y orgánica, ya que con el tiempo se desarrollan nuevas herramientas y maduran las ya existentes.

---

<sup>6</sup>*Spatialite*: Herramienta para dotar de capacidades *GIS* a una base de datos *SQLite*.  
<https://www.gaia-gis.it/fossil/libspatialite/index>

# Capítulo 3

## Análisis de Redes

---

**Resumen** Empezamos explicando qué es la Teoría de Grafos y aquellas propiedades relevantes al estudio. Continuamos haciendo lo mismo con la Topología. Entonces, se explica el encaminamiento, además de comentar algunas herramientas *GIS* aplicarlo.

---

### Introducción

Se plantea el Análisis de Redes, basado en la Teoría de Grafos y la Topología, desde el punto de vista de un *GIS*. Este nos permite crear modelos de redes basados en la realidad, consultar propiedades como conectividad o adyacencia y realizar estudios basados en estas y otras propiedades.

#### 3.1. Teoría de Grafos

Un grafo es una estructura combinatoria [Val02, págs. 3-4] que consiste en un conjunto finito de vértices y aristas, relaciones entre estos<sup>1</sup>. Aunque

---

<sup>1</sup>Incluso si esta definición es correcta, usa una terminología matemática, la cuál no es habitual cuando hablamos de Análisis de Redes, prefiriéndose los términos *nodo* para los vértices y *enlace* para las aristas.

estas estructuras tienen multitud de propiedades, solo se explicarán aquellas relevantes al área de estudio.

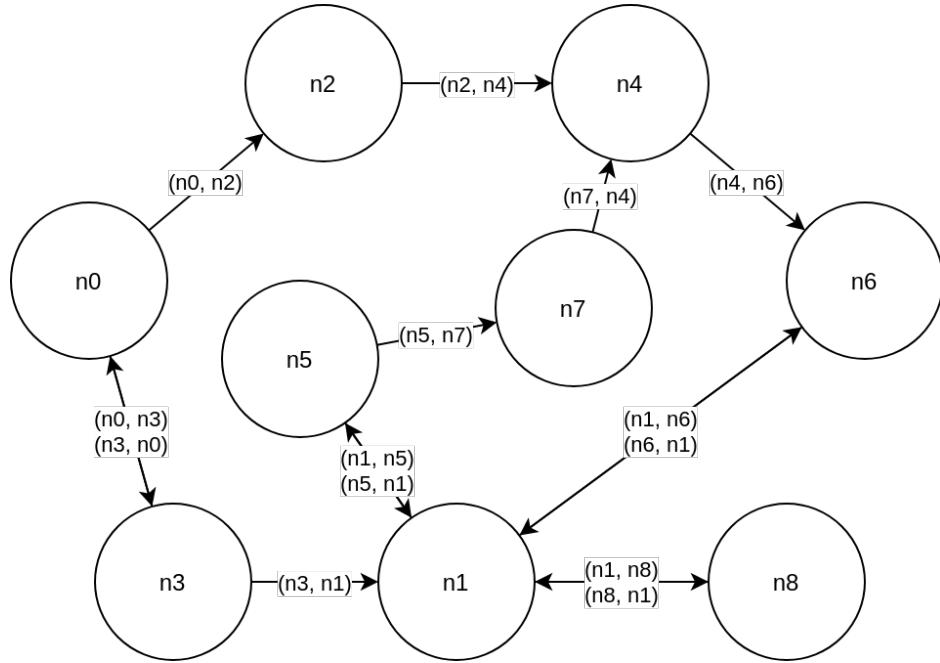


Figura 3.1: Un grafo dirigido es muy útil a la hora de representar problemas de navegabilidad, como por ejemplo en la búsqueda de caminos

Los grafos pueden tener restricciones, como por ejemplo de navegabilidad, pues decimos que un grafo es dirigido en el caso de que algún enlace solo pueda recorrerse en un único sentido.

Se dice que dos nodos son adyacentes si existe un enlace que une dichos nodos, si son vecinos. Esto nos permite definir los caminos, sucesiones ordenadas de nodos vecinos entre sí cuyo el punto de partida, el primer nodo, se denomina origen y su final, el último nodo, denota el objetivo.

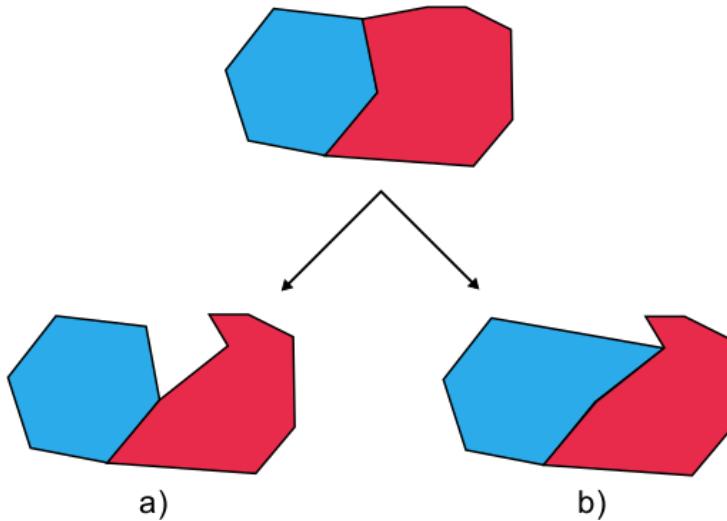
## 3.2. Topología

La Topología [Ola14, pág. 96] es una disciplina matemática que estudia las características de objetos geométricos, como grafos, que no varían al aplicar una transformación, como mover un nodo de lugar. En el contexto GIS, la topología es generalmente definida [GforE] como la relación espacial entre elementos vectoriales adyacentes o conectados entre sí. Es decir, en términos

### 3.2. TOPOLOGÍA

informales, una topología GIS es una estructura que nos detalla como están conectados los elementos.

Estas relaciones, dicha estructura, permanecen constantes cuando se aplican ciertas deformaciones [OS15, pág. 310], como un giro o una traslación, y nos permiten verificar la integridad de los datos.



[Ola14, pág. 97]

Figura 3.2: Aplicación de una transformación a un punto sin (a) y con (b) información topológica

Un ejemplo real de esto es una red de autobuses, si en algún momento se moviese la ubicación de alguna parada, el conjunto de conexiones, o enlaces, seguiría siendo el mismo, exceptuando que dicha parada, o nodo, ha sido movida de lugar.

Los datos vectoriales no necesariamente deben contener información topológica pero, generalmente, se puede obtener a través de un proceso de descomposición de primitivas, partiendo de todos los enlaces, y descomponiendo estos en subdivisiones entre nodos con un grado mayor que dos, o entre nodos origen y objetivo. Es decir, enlaces que contengan aquellos nodos que sean usados por algún otro enlace.

### 3.3. Encaminamiento

Una de las principales áreas de aplicación del Análisis de Redes es el encaminamiento, la búsqueda de caminos, generalmente basados en distancia (camino más corto) pero también en tiempo (camino más rápido). Indistintamente de cuál es la heurística principal, suelen darse un conjunto de condiciones adicionales, como la navegabilidad o pasaje por algún nodo intermedio.

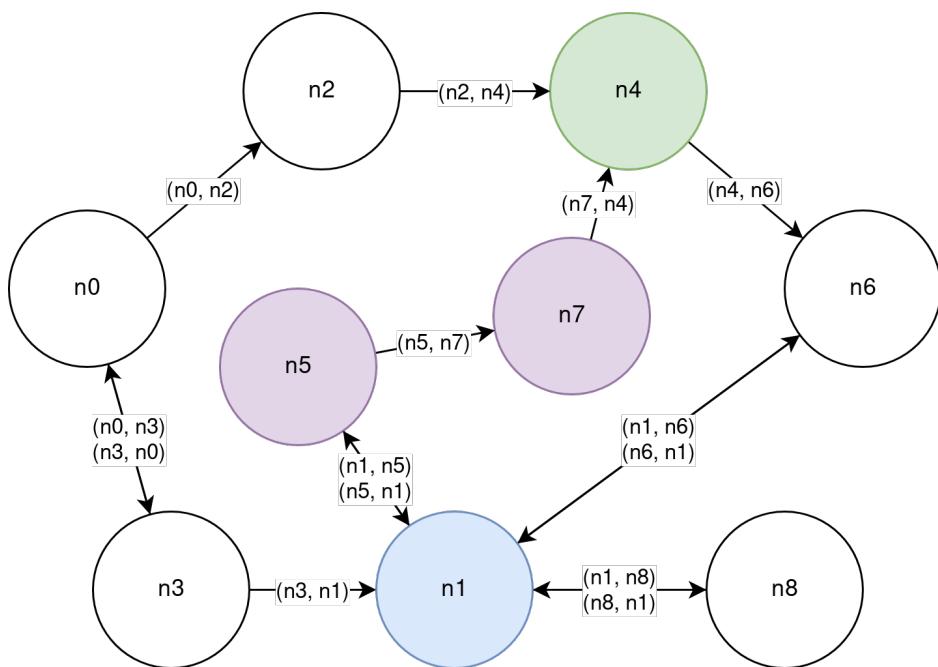


Figura 3.3: Partiendo del origen  $n_1$  (en azul) y buscando llegar al destino  $n_4$  (en verde), se encuentra un camino (en lila) óptimo y, casualmente, único denotado como la secuencia de nodos  $\{n_1, n_5, n_7, n_4\}$  o la secuencia de enlaces  $\{(n_1, n_5), (n_5, n_7), (n_7, n_4)\}$  a través de un algoritmo de búsqueda de caminos.

Esto tiene una sinergia tremenda con la información geográfica pues, como ya se ha comentado, podemos apreciar ciertos paralelismos entre grafos y datos geográficos. Aún así, este área no es algo sencillo, pero hay algoritmos muy conocidos, además de existir muchas herramientas que nos permiten trabajar con redes sobre las que aplicar técnicas de encaminamiento, comentadas a continuación.

### 3.3. ENCAMINAMIENTO

---

#### 3.3.1. *pgRouting*

*pgRouting* es una extensión de *PostgreSQL* para el Análisis de Redes, concretamente en el ámbito de búsqueda de caminos [OS15, págs. 410-416].

Como tal, al habilitar esta extensión en una base de datos, se nos permite el uso de las siguientes funciones, entre otras.

**pgr\_CreateTopology** Encargada de crear la topología de una entidad que contenga líneas vectoriales, descomponiendo cuando sea necesario.

**pgr\_Dijkstra** Usada para buscar el camino más corto usando el algoritmo de *Dijkstra*

Esto, con relativa sencillez, nos proporciona una herramienta muy eficaz para realizar búsqueda de caminos, pero está fuertemente ligada a la disposición de una base de datos *PostgreSQL*, con *PostGIS* y *pgRouting*, limitando su despliegue en dispositivos con menos recursos.

#### 3.3.2. *NetworkX*

*NetworkX* es un *framework*<sup>2</sup> de *Python* para trabajar con grafos, y una excelente herramienta que nos permite aplicar encaminamiento en una red, al presentarse de manera agnóstica a ningún tipo de origen de datos, pero con un mayor coste en tiempo de preparación. Incluso con esto dicho, también facilita mucho la integración con diferentes fuentes de datos, como por ejemplo con *Shapefile*, *CSV*, entre otros.

**networkx.from\_pandas\_edgelist(df, \*)** Función para cargar un grafo a partir de un *DataFrame* de *Pandas* o *GeoPandas*.

**networkx.shortest\_path(G, b, e, \*)** Una función que toma un grafo y dos identificadores de nodos, realizando una búsqueda de camino de forma rápida.

---

<sup>2</sup>Conjunto de herramientas, marco de trabajo

# Capítulo 4

## Encaminamiento con GeoPackage

---

**Resumen** De partida, se presenta una explicación de qué datos son necesarios y cómo obtenerlos. Luego se explican algunas técnicas de manipulación para filtrar los datos que consideremos interesantes en el estudio. Después de esto, se presenta la estructura que esta topología tomará en el contenedor de datos. A continuación, se detalla el procedimiento a seguir para obtener dicha topología a partir de los datos filtrados. Finalmente, se demuestra las posibilidades de encaminamiento con dicho contenedor de datos.

---

### Introducción

Este capítulo es el principal del desarrollo y está dedicado a analizar la viabilidad de generar y estructurar una topología de alguna red geográfica, como puedan ser carreteras o ferroviaria, y su almacenamiento en un contenedor GeoPackage. Esto se enlaza con la presencia de un caso práctico en el que se genera la topología de red del término municipal de Granada, partiendo de unos datos generales sin tratar, hasta conseguir un contenedor *GeoPackage* donde se encuentre la información relevante, así como la misma

#### *4.1. OBTENCIÓN DE LOS DATOS*

---

topología de red.

Se pretende llevar un desarrollo mixto, ligando un análisis general con el caso práctico, explicando qué hacer en ámbitos generales y después aplicar lo explica al ámbito concreto de la ciudad de Granada. Concretamente, para el caso práctico, además se nombran bastantes herramientas usadas y cómo se integran con aquellas desarrolladas.

### **4.1. Obtención de los Datos**

El pilar básico de cualquier análisis es la información, como conjunto de datos estructurado. En el caso del objeto de este estudio, necesitamos datos que nos proporcione información sobre una red de carreteras sobre las que queramos aplicar técnicas de encaminamiento, de búsquedas de caminos. A grandes rasgos, existen dos posibilidades.

Por una parte, podemos generar la información geográfica a partir de la digitalización [Bol12, págs. 140-152]. Esto es, por ejemplo, mediante el uso de ortofotografías (fotografías corregidas a un ángulo ortogonal a la tierra) podemos generar un mapa vectorial, sea manual o pseudoautomatizado, pero este es un proceso costoso.

Por suerte, también existe la posibilidad de usar datos recopilados por otras fuentes, ya sean puestos a libre disposición o como producto a vender. Estas pueden ser distribuidas por algún proyecto gubernamental, empresarial o comunitario, e incluso por particulares.

En definitiva, los datos de partida, irrelevantes sea su origen, deben contener, al menos, suficiente información sobre una estructura en red sobre la que podamos extraer una topología válida, por ejemplo, una red de carreteras de un municipio o la red ferroviaria de un país.

### **Caso Práctico**

Se parte de los datos geográficos de toda España[SpaDa], recopilados por el proyecto *OpenStreetMap* y empaquetados de forma gratuita por la empresa *GeoFabrik*, con un formato XML comprimido en *ProtocolBuf*<sup>1</sup>.

---

<sup>1</sup>Protocol Buffers: Protocolo de serialización diseñado por Google. <https://developers.google.com/protocol-buffers/>

## 4.2. TRANSFORMACIONES Y FILTRADO



Figura 4.1: Los datos comprenden la extensión vista en la figura, siendo gran parte de la península ibérica y las islas de Majorca, y no solo contienen información de carreteras, si no también términos municipales, reservas naturales, edificios, entre muchas otras cosas.

Estos datos contienen, en términos de *OpenStreetMap*, elementos NODE, WAY y RELATION. Para un estudio básico, basta con las primitivas WAY, que forman las líneas que describen la composición geográfica de los camino, así como atributos que nos permitan seleccionar qué datos nos interesan. Concretamente, usaremos el atributo `highway` para identificar qué tipo de carretera representa y eliminar aquellos que no tengan relevancia.

## 4.2. Transformaciones y Filtrado

Muy comúnmente, es de interés profundizar en un área concreta para realizar análisis con un ámbito más específico del que nos proporcionan los datos originales. Cuando esto ocurre, es conveniente utilizar operaciones espaciales, como recorte (o *clipping* [Bol12, págs. 354-355]), delimitando el área a conservar con un polígono.

Otra posibilidad es, en vez de utilizar el ámbito geográfico de los datos, usar los datos en sí mismos. Es decir, si consideramos que nuestros datos como una serie de atributos  $(x_1, \dots, x_n)$ , podemos usar un atributo  $x_m$  para filtrar todos aquellos conjuntos que no cumplan las condiciones necesarias para el estudio a desarrollar.

El realizar estas operaciones nos permite eliminar aquellos datos que no sean relevantes a lo que queremos analizar, minimizando el riesgo a error y la carga de procesamiento a realizar.

## 4.2. TRANSFORMACIONES Y FILTRADO

---

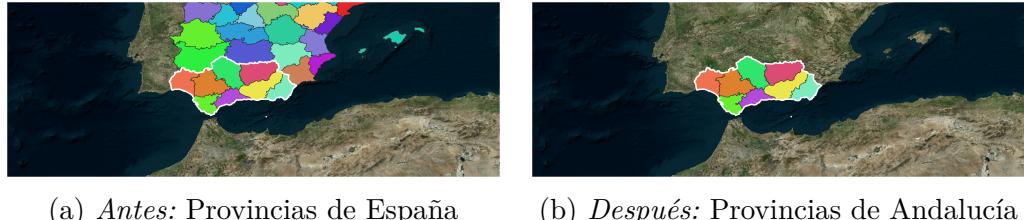


Figura 4.2: Recorte de provincias españolas, delimitado por el polígono de la comunidad andaluza, mostrado como traza en blanco.

	fid	geom	osm_id	name	highway	waterway	aerialway	...
								tags
1	1	LINE...	407780	Oviedo-...	secondary	NULL	NULL	NULL
2	2	LINE...	2497138	pit lane	raceway	NULL	NULL	NULL
3	3	LINE...	2497139	Circuito ...	raceway	NULL	NULL	NULL
4	4	LINE...	2497152	Curva Ayrton...	raceway	NULL	NULL	NULL
5	5	LINE...						
6	6	LINE...	1	3996189	M-40	motorway	9	"ref"=>"M-...
7	7	LINE...	2	3996190	Avenida de ...	primary	17	"junction"=...
8	8	LINE...	3	3996192	Plaza de ...	primary	7	"lanes">>"..."
9	9	LINE...	4	3996196	Avenida de ...	primary	7	"oneway"=...
10	10	LINE...	5	3996197	Avenida de ...	primary	7	"lanes">>"1..."
11	11	LINE...	6	4049477	Avenida de ...	primary	7	"lanes"=>"..."
12	12	LINE...	7	4049478	Avenida de ...	primary	7	"lanes"=>"..."
13	13	LINE...	8	4049479	Avenida de ...	primary	7	"destination..."
14	14	LINE...	9	4049533	Paseo de la ...	primary	7	"cycleway:la..."
15	15	LINE...	10	4049535	Avenida de l...	motorway	9	"ref"=>"M-..."

Figura 4.3: En este caso de filtrado por atributos, con datos en formato OpenStreetMap, se ha restringido aquellos posibles valores del campo **highway**, además de eliminado columnas no relevantes, obteniendo un conjunto de datos reducido pero más enfocado al análisis a desarrollar.

#### 4.3. ESTRUCTURA Y EMPAQUETADO

### Caso Práctico

Partiendo de los datos originales a nivel estatal, es decir, de toda España, se ha seleccionado solo aquellos datos que correspondían con carreteras aptas para vehículos motorizados (como coches) en el término municipal de Granada. Esto se ha realizado a través de las utilidades de *Osmium*<sup>2</sup>.

**Osmium Extract** Se ha usado un polígono, descrito en **GeoJSON**, que corresponde al área municipal de Granada [GrPoly] y se ha invocado de la siguiente forma:

```
|>> osmium extract spain-latest.osm.pbf -p granada.json -o granada-latest.osm.pbf
```

**Osmium Tags Filter** Se ha filtrado por todas las primitivas WAY cuyo valor de **highway** comprenda las calles autorizadas para vehículos motorizados, de la siguiente manera:

```
|>> osmium tags-filter granada-latest.osm.pbf 'w/highway=*_link'  
|> 'w/highway=motorway,trunk,primary,secondary,tertiary,residential,living_street' -o  
|> granada-latest-highways.osm.pbf
```

### 4.3. Estructura y Empaquetado

Como ya se ha explicado antes, *GeoPackage* es en esencia un contenedor *SQLite*, una base de datos relacional muy portable. Como tal, resulta un formato muy adecuado para datos compuestos por atributos, aunque no necesariamente estén relacionados entre sí.

Un *GeoPackage* puede estar compuesto por entidades cualesquiera, pero en este caso concreto, necesitaremos dos entidades adicionales, una para nodos y otra para enlaces, tal y como se define en Figura 4.4.

A ambas entidades, como información interesante, se puede añadir una estimación de coste, bien en distancia o tiempo, restricciones de navegabilidad en una u otra dirección, entre otras. Esto se podría hacer directamente en la misma entidad, o en una entidad externa que contenga metadatos y referencie

---

<sup>2</sup>Osmium: Proyecto con numerosas herramientas para trabajar con datos de OpenStreetMap. <https://osmcode.org/>

#### 4.4. PREPARACIÓN DE LA TOPOLOGÍA

---

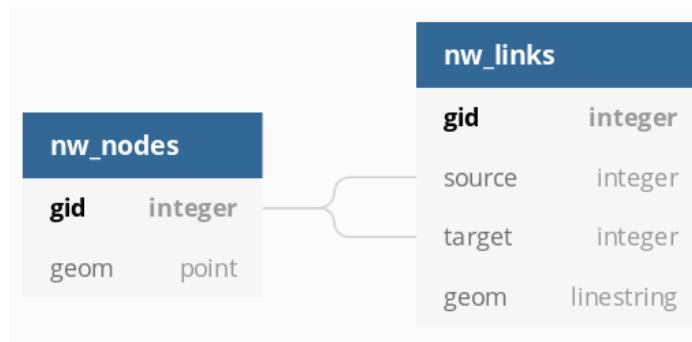


Figura 4.4: Como mínimo, debe almacenarse la geometría, y un elemento identificativo, usado para referenciar los nodos en cada enlace, como nodos de origen y de objetivo.

por el identificador

### Caso Práctico

En el caso práctico, se ha almacenado los datos de Granada, ya procesados y reducidos, sobre un fichero *GeoPackage*. Esto se ha hecho gracias a una herramienta de *GDAL*<sup>3</sup> llamada *ogr2ogr*<sup>4</sup>.

```
|>> osmium extract spain-latest.osm.pbf -p granada.json -o granada-latest.osm.pbf
```

Por razones prácticas, no creamos las entidades aún. Esto lo hacemos en el siguiente apartado, directamente con la herramienta que se ha desarrollado para generar la topología.

## 4.4. Preparación de la Topología

Para generar una topología, tan solo necesitamos todas los enlaces que componen la red a analizar. Para crear una topología de dicha red, cada

<sup>3</sup>*GDAL/OGR*: Colección de herramientas de software libre para trabajar con datos geográficos. <https://gdal.org/>

<sup>4</sup>*ogr2ogr*: Herramienta para convertir entre formatos de datos geográficos.

## 4.4. PREPARACIÓN DE LA TOPOLOGÍA

enlace ha de descomponerse en cada intersección entre dos o más enlaces. Para ello, aplicaremos un sencillo algoritmo definido en Listado 4.1.

```
def create_topology(linestrings, valid_nodes):
    sublinks = []
    for ls in linestrings: # linestrings: List of LineStrings
        source = ls[0] # LineStrings are assumed array-like
        for target in ls[1:]:
            if target not in valid_nodes: # valid_nodes: Nodes visited more than once
                continue
            sublink = split(ls, source, target) # split: Creates LineString from two nodes
            sublinks.append(sublink) # sublinks: List of decomposed LineStrings
            source = target
    return sublinks
```

Listado 4.1: Dado un conjunto de líneas, para cada línea, dividimos entre dos puntos intermedios, siempre y cuando estos sean válidos, hayan sido usado más de una vez en el total de las líneas. Como invariante necesaria, se considera que tanto el primer y último punto de una línea son válidos.

El objetivo de este proceso es la descomposición en unidades únicas, una por cada intersección encontrada a lo largo de una línea. Para llevar esto a cabo, debemos mantener un registro de cuántas veces se utiliza cada punto contenido en cada línea y subdividir cada línea entre dos nodos. Una vez tenemos la topología de red procesada, debemos almacenarla en el *GeoPackage* conforme a la estructura descrita en Figura 4.4.

## Caso Práctico

Para el caso práctico, se ha desarrollado un software [GpkgR]] en *Python* que utiliza *PyOsmium*<sup>5</sup> para leer todos las primitivas WAY de *OpenStreetMap*.

Una vez hemos cargados todos los datos, se aplica el algoritmo descrito en Listado 4.1 para descomponer los caminos en enlaces de una topología y posteriormente almacena la red en un *DataFrame* de *GeoPandas*<sup>6</sup>, desde la cual exportaremos a GeoPackage. Una vez realizado esto, se almacena el

<sup>5</sup> *PyOsmium: Bindings* en Python para la biblioteca *Osmium*

<sup>6</sup> *GeoPandas*: Extensión *GIS* de *Pandas*, framework de *Python* con estructuras *DataFrames*, marcos de datos. <http://www.geopandas.org/>

#### 4.4. PREPARACIÓN DE LA TOPOLOGÍA

---

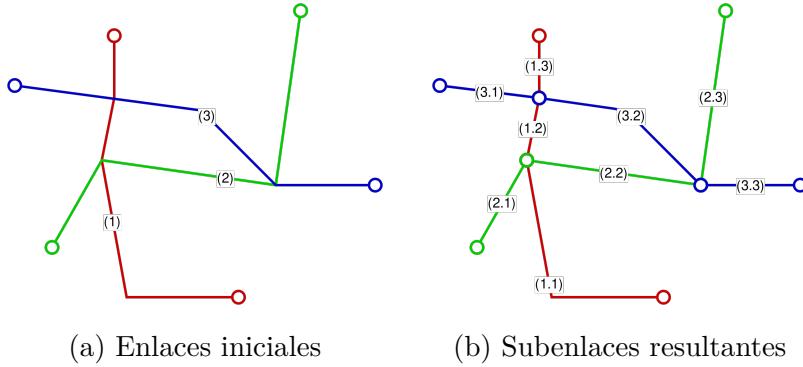


Figura 4.5: Descomposición topológica de una red, nótese como cada intersección está delimitada con un redondel. Esto es porque cada enlace se ha dividido en varios enlaces, conservando los lados con varios puntos sin intersección en el proceso.

```

def way(self, w: osmium.osm.Way):
    # Store all node WKB location, tracked by node id
    self._nodes.update({n.ref : wktfab.create_point(n) for n in w.nodes})

    # Store all node ids on a way, tracked by way id
    self._ways.update({w.id : [n.ref for n in w.nodes]})

    # Keep a count of used nodes for pruning. Extremities count twice
    self._counter.update(([n.ref for n in w.nodes], w.nodes[0].ref, w.nodes[-1].ref))

```

Listado 4.2: El software carga todos las primitivas WAY de un fichero OpenStreetMap y además, para cada elemento, realiza un conteo del número de usos de cada nodo.

#### 4.5. APLICACIÓN DEL ENCAMINAMIENTO

contenido del *DataFrame* en el fichero *GeoPackage* donde tenemos los datos de partida.

```
self.nodes = gpd.GeoDataFrame(  
    data=nodes_data,  
    columns=['node_id', 'geom'],  
    geometry='geom',  
    crs={'init': 'epsg:4326'},  
)  
self.links = gpd.GeoDataFrame(  
    data=links_data,  
    columns=['way_id', 'source', 'target', 'geom'],  
    geometry='geom',  
    crs={'init': 'epsg:4326'},  
)
```

Listado 4.3: Los DataFrame de GeoPandas son una estupenda representación intermedia, ya que son contenedores muy eficientes y además gozan de mucha funcionalidad, como pueda ser directamente exportar a una base de datos *GeoPackage*.



Figura 4.6: Resultado gráfico de la topología resultante al aplicar el software *gpkgRouting* y visualizado en *QGIS*

### **4.5. Aplicación del Encaminamiento**

En este momento, tenemos un *GeoPackage* enrutable, el cual podemos inmediatamente utilizar o compartir con terceros como recurso. En el ca-

#### 4.5. APLICACIÓN DEL ENCAMINAMIENTO

---

so de aplicar inmediatamente alguna técnica de encaminamiento, debemos construir un grafo con la información almacenada en nuestro *GeoPackage*.



Figura 4.7: Visualizamos el camino obtenido de Listado B.1 en la suite *QGIS*. En este caso, podemos ver que partimos de ETSIIT y acabamos cerca de la Cámara de Comercio de Granada.

Con el grafo ya construido, podremos aplicar algún algoritmo de búsqueda de caminos [Val02, págs. 255-287] para el encaminamiento, existiendo mejores elecciones en función de los datos que tenemos. En el peor caso, siempre podremos aplicar un algoritmo en profundidad usando como coste la distancia de cada enlace.

Heurísticas utilizables podrían ser también el tiempo medio que tarda un vehículo en recorrer una calle, si es que dispusiésemos de tal información. En el caso de haber subdividido un enlace, podríamos considerar la distancia de la división respecto a la distancia total y obtener de ahí un valor proporcional.

### Caso Práctico

A partir de *NetworkX*<sup>7</sup>, se genera un grafo dirigido en el que todos los enlaces se representan como tuplas (*origen, destino*) y tienen sus atributos concretos. Con el mismo, se aplican algoritmos como A\* para calcular secuencias de nodos por los que pasar para encontrar el camino más óptimo.

---

<sup>7</sup> *NetworkX*: Framework de Python para tratamiento y operaciones con grafos. <https://networkx.github.io/>

```

import geopandas as gpd
import networkx as nx
df = gpd.from_file('granada-latest-highways.gpkg', driver='GPKG', layer='nw_links')
G = nx.from_pandas_edgelist(df, create_using=nx.DiGraph)

try:
    path = nx.shortest_path(G, SOURCE, TARGET)
    # Do something with the retrieved path
    ...
except nx.NetworkXNoPath:
    # No path found
    ...

```

Listado 4.4: *Python* presenta un ecosistema muy maduro con multitud de herramientas, entre ellas *GeoPandas* y *NetworkX*, que nos permiten trabajar con datos *GIS* y grafos de una forma indolora. Este último, además, tiene funcionalidades de búsqueda de caminos.

## 4.6. Conclusiones

Dado el análisis y desarrollo llevado a cabo, cabe concluir que es posible almacenar topología en contenedores GeoPackage, y que estos pueden ser usados de forma reiterada para buscar caminos entre nodos de una red.

Es más, al ser un formato relativamente sencillo y casi universalmente aceptado, al ser una base de datos *SQLite*, es trivial profundizar en ámbitos más específicos, que cubran casos de uso concretos. Lo inmediato, y muchas veces mencionado, sería su uso en terminales móviles que, sea por capacidades técnicas o software, no son capaces de aceptar otras soluciones para almacenamiento en local.

Esto, de igual forma, es complejo, ya que no existe un reglamento que enuncie cómo deben estructurarse los datos para que estos sean interpretados de una forma única y correcta, pero por suerte la especificación GeoPackage es abierta y extensible, permitiendo desarrollar extensiones para añadir funcionalidad<sup>8</sup>.

Aún así, sea complejo o no, es una posibilidad más para empaquetar y compartir datos geográficos con la oportunidad de realizar consultas de encaminamiento, como herramientas como *pgRouting*. Al fin y al cabo, un ingeniero debe conocer multitud de herramientas o tecnologías para decidir

---

<sup>8</sup>Véase un ejemplo esta extensión titulada *Tiled Gridded Coverage Data Extension*. <http://docs.opengeospatial.org/is/17-066r1/17-066r1.html>

#### *4.6. CONCLUSIONES*

---

cual es la más conveniente para un problema concreto.

# Capítulo 5

## Conclusión y Trabajo Futuro

Como se ha podido comprobar en el capítulo anterior, es muy viable la creación de un *GeoPackage* con información topológica para realizar análisis de redes de carreteras u otro tipo.

Esto no quiere decir que sea la única ni mejor solución, si no una más, para ámbitos en los que otras soluciones más maduras no puedan implementarse. De igual manera, sigue siendo una posibilidad más que válida, con un nicho de mercado muy interesante como pueda ser el encaminamiento en local o el empaquetado y envío de datos geográficos enrutables.

Así pues, a continuación se presentan dos secciones, una para desglosar qué objetivos se han cumplido y cómo se ha llevado esto a cabo, y otra para definir qué trabajo futuro podría realizarse sobre las conclusiones resultantes de este análisis.

### 5.1. Objetivos

Tanto el objetivo principal como los objetivos secundarios han sido completados a lo largo del desarrollo del TRABAJO DE FIN DE GRADO. Estos objetivos son listados en la Sección 1.1.

El objetivo principal se ha cumplido en toda la realización del Capítulo 4, llevándose a cabo el análisis que se esperaba, obteniendo como conclusiones que sí es viable y las pautas a seguir para obtener un *GeoPackage* enrutable, así como la estructura interna que podrían seguir los datos y algoritmos o extractos de código fuente para completar el proceso.

## **5.2. TRABAJO FUTURO**

---

Los objetivos secundarios se han completado de igual manera, en el proceso de documentación y realización previa a la redacción del documento, y esto se puede observar en el texto y herramientas desarrolladas, a disposición en este mismo documento o en los recursos adicionales presentados al final del mismo.

- Se ha conocido la estructura de la base de datos de *OpenStreetMap* a partir de la *OsmWiki*<sup>1</sup> y se ha aprendido cómo usar este a partir de la extracción de datos a partir de las herramientas del proyecto *Osmium*.
- Se han identificado más tipos de contenedores de datos geográficos a través de recursos como sean [Ola14] o [Bol12] y qué ventajas o desventajas presentan.
- Se ha encontrado herramientas como *Osmium*, *GeoPandas*, *NetworkX*, entre otras, para la creación, manipulación y visualización de datos geográficos en un entorno general.
- Referente al punto anterior, muchas herramientas anteriormente citadas funcionan bajo un entorno PYTHON, o tienen algún tipo de *binding*, así pues se exhibe que dicho entorno es muy capaz a la hora de trabajar con información geográfica y análisis estadísticos o de otro tipo.

## **5.2. Trabajo Futuro**

El trabajo, una vez finalizado y con sus conclusiones presentadas, podría llegar a extenderse por varios caminos.

Por un lado, se podría ver la posibilidad de la estandarización del modelo de representación de componentes topológicas, la propia red de conexiones entre nodos y los nodos en sí, como una extensión del estándar *GeoPackage*. De esto llevarse a cabo, habría que producir un boceto técnico y presentarse al *Open Geospatial Consortium* como una extensión de la comunidad<sup>2</sup>. Una vez esto, debería ser publicitado, implementado y revisado reiteradamente hasta que finalmente acepte la madurez necesaria para que se convierta en una extensión oficial.

Por otro lado, cabe la posibilidad de extender la prueba de concepto desarrollada, añadiendo mayor cantidad de funcionalidades aunque dependa

---

<sup>1</sup>Véase <https://wiki.openstreetmap.org/wiki/>

<sup>2</sup>Véase <https://www.geopackage.org/extensions.html>

## 5.2. TRABAJO FUTURO

internamente de los mismos proyectos ya comentados, llámese *NetworkX*, *GeoPandas*, *Osmium*, además de otros. El objetivo principal de este proyecto debería ser la accesibilidad a desarrolladores, así pues siendo un nexo de enlace entre las muchas herramientas ya comentadas.

Otra posibilidad relacionada con la anterior, sería utilizar las herramientas de más bajo nivel, fuera del ecosistema PYTHON, y buscar herramientas más portables a otras plataformas, como pueda ser basadas en tecnologías como C++ o JAVA, idealmente con el mismo planteamiento que la propuesta anterior, la accesibilidad a otros desarrolladores.

Este trabajo también puede, propiamente dicho, ayudarme a encontrar trabajo en el sector *GIS*, donde podría plasmar todo el conocimiento adquirido y refinado en la elaboración del análisis y de la misma implementación.

# Apéndice A

## Planificación del Proyecto

Aunque no se ha llevado ninguna metodología concreta como podría ser, por ejemplo, *Scrum*, se aplicado una metodología ágil e incremental al realizarse prototipos o bocetos sobre los que se ha continuado el desarrollo, también con gran cantidad de realimentación al solicitar opinión a mi tutor, José Samos Jiménez, tanto por correo como en horario de tutorías.

Dicho esto y considerando que, en España, un crédito *ECTS* equivale a 25 horas de estudio, y que la asignatura de TRABAJO DE FIN DE GRADO se compone de 12 créditos *ECTS*, el tiempo a dedicar a este trabajo debería haber sido 240 horas, el cual se ha desarrollado por fases descritas en Tabla A.1. Este tiempo ha sido alcanzado e incluso superado, por dedicarle horas en las que no conseguía un rendimiento muy alto.

Este trabajo se ha desglosado por actividades en las proporciones que se muestran en Tabla A.2.

Período	Enfoque	Tiempo	Dedicación
Primer Semestre	Estudio e Investigación	36 horas	15 %
Segundo Semestre	Experimentar y Codificar	72 horas	30 %
Período de Verano	Desarrollo del Documento	132 horas	55 %
	Trabajo de Fin de Grado	240 horas	100 %

Cuadro A.1: El esfuerzo se ha repartido a lo largo del periodo académico comprendido entre los años dos mil dieciocho y diecinueve (2018-2019), en una proporción irregular, por cuestiones de tiempo disponible.

---

Período	Actividad	Tiempo	Dedicación
I	Estudio de Herramientas	18 horas	7.5 %
I	Análisis de Viabilidad	18 horas	7.5 %
II	Diseño del Software	36 horas	15 %
II	Codificación del Software	36 horas	15 %
III	Desarrollo del Documento	120 horas	50 %
III	Incorporación en otras Soluciones	6 horas	2.5 %
III	Creación de Recursos Gráficos	6 horas	2.5 %
	Trabajo de Fin de Grado	240 horas	100 %

Cuadro A.2: El desarrollo de cada actividad ha estado realmente intercalado con otras actividades, por ejemplo la creación de recursos gráficos se ha llevado a cabo durante la realización del documento. Los tiempos y porcentajes dados son estimaciones de la realidad.

## Explicación

Elegí desarrollar y exponer este trabajo en un año, académicamente, muy pesado al estar cursando once asignaturas además de este trabajo, completando un total de 78 créditos *ECTS*.

Como tal, el proyecto partió de una planificación rígida a algo más flexible. Quise partir de repartir mi tiempo a partes iguales en investigar, desarrollar y escribir. Esto, como cabía esperar, no se cumplió y acabó siendo diferente, véase Tabla A.1.

Como tal, empecé en el primer semestre del año académico buscando ideas sobre qué desarrollar como objeto de mi TRABAJO DE FIN DE GRADO. En la susodicha asignatura de SISTEMAS DE INFORMACIÓN GEOGRÁFICOS elegí el tema a tratar, por mi cariño a la asignatura y mi interés en una tecnología libre como *GeoPackage*.

Ese semestre no logré mucho más que aprender sobre *GIS* y diseñar ciertos prototipos, como la inyección de funciones *SQL* para imitar a *pgRouting*, pero por limitaciones de *SQLite*, tuve que desestimararlo.

Al siguiente, segundo semestre, logré desarrollar más el texto, particularmente la estructura y los elementos a introducir, pero la ausencia de tiempo me hizo dejar la entrega final a Septiembre.

En los meses de verano, ya con todo o gran parte del código desarrollado y

---

la idea bien clara, he dedicado gran parte de mi tiempo diario a el desarrollo del documento, sea texto o recursos como figuras o códigos.

Consolidando toda esta información explicada arriba, podríamos decir que este proyecto se ha desarrollado en tres fases, primer y segundo semestre académico y los meses de verano, cada uno con un enfoque y una dedicación concreta.

## Apéndice B

### Integración con *QGIS*

*QGIS*, una herramienta *GIS* de software libre muy capaz y multiplataforma, cuenta tanto con un sistema de *plugins* como con un entorno de consola, ambas con soporte para *Python* y su ecosistema. Por tanto, integrar la prueba de concepto con *QGIS* es casi inmediato, ya que todo lo desarrollado usa *Python* como forma de implementación.

En el siguiente ejemplo Listado B.1, se carga un fichero *GeoPackage* ya enrutable, obtenido a través del proceso descrito en el capítulo ENCaminamiento con *GEOPACKAGE*, se genera un grafo y se realizan consultas de encaminamiento sobre este. Concretamente, el ejemplo mostrado es aquel que genera la imagen mostrada en Figura 4.7.

Por ser una mera prueba de concepto, se trata de un simple *script* y no es un *plugin* como tal, ya que existen otras herramientas que te ofrecen una herramienta similar dentro de *QGIS*, como las extensiones de *GRASS*.

```

1  # Example configuration
2  FILENAME = '/home/alejnp/Projects/UGR/TFG/gpkgrouting' \
3      '/gpkgrouting/output/TopologyHandler' \
4      '/granada-latest-highways.gpkg'
5  SOURCE_NODE, TARGET_NODE = 33400024, 330186722
6
7  # Imports
8  import geopandas as gpd
9  import networkx as nx
10 import qgis.core as qgs
11 import shapely.ops as ops
12
13 # From GeoPackage, load as GeoPandas dataframe
14 nodes = gpd.read_file(FILENAME, layer='network_nodes')
15 links = gpd.read_file(FILENAME, layer='network_links')
16
17 # From Dataframes, create NetworkX graph
18 graph = nx.from_pandas_edgelist(links,
19     edge_attr=True, create_using=nx.DiGraph)
20
21 # Perform pathfinding within the network
22 path_nodes = nx.dijkstra_path(graph,
23     source=SOURCE_NODE, target=TARGET_NODE,
24     weight=lambda u, v, e: e['geometry'].length)
25
26 # Create a LineString with the path found
27 linestring = ops.linemerge(graph.edges[u,v]['geometry']
28     for u, v in zip(path_nodes, path_nodes[1:]))
29 )
30
31 # Create a new QGIS LineString layer
32 layer = qgs.QgsVectorLayer('LineString?crs=epsg:4326',
33     f'Shortest Path from {SOURCE_NODE} to {TARGET_NODE}',
34     'memory')
35
36 # Create QGIS geometry from path LineString
37 geometry = qgs.QgsLineString()
38 geometry.fromWkt(linestring.to_wkt())
39
40 # Create new QGIS feature to add onto layer,
41 # from previous geometry
42 feature = qgs.QgsFeature()
43 feature.setGeometry(geometry)
44
45 # Retrieve data provider of the layer and add feature
46 prov = layer.dataProvider()
47 prov.addFeature(feature)
48
49 # Add layer to current project instance, displaying it
50 qgs.QgsProject.instance().addMapLayer(layer)

```

Listado B.1: qgis\_gpkgrouting.py

# Bibliografía y Referencias

- [Bol12] Paul Bolstad. *GIS Fundamentals: A first text on geographic information systems*. 4.<sup>a</sup> ed. AtlasBooks, 2012. ISBN: 9780971764736.
- [Ola14] Víctor Olaya. *Sistemas de Información Geográfica*. 2014. URL: <https://volaya.github.io/libro-sig/>.
- [OS15] Regina O. Obe y Leo S. Hsu. *PostGIS in Action*. 2.<sup>a</sup> ed. Manning Publications, 2015. ISBN: 9781617291395.
- [Val02] Gabriel Valiente. *Algorithms on Trees and Graphs*. Springer, 2002. ISBN: 3540435506.

## Lectura Adicional

- [Ros12] Charles Roswell. *Modeling of Geographic Information*. Ed. por Wolfgang Kresse y David M. Danko. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 978-3-540-72680-7. DOI: 10.1007/978-3-540-72680-7. URL: <https://doi.org/10.1007/978-3-540-72680-7>.
- [Wes16] Erik Westra. *Python Geospatial Development (3rd Edition)*. Packt, 2016. ISBN: 9781785288937.

## Recursos Electrónicos

- [GforE] GIS for Educators, ed. URL: [http://linfiniti.com/dla/worksheets/6\\_Topo.pdf](http://linfiniti.com/dla/worksheets/6_Topo.pdf).
- [GpkgR] Alejandro Núñez Pérez. *gpkgRouting*. URL: <https://www.github.com/Alxe/gpkgrouting>.

## *RECURSOS ELECTRÓNICOS*

---

- [GrPoly] *GeoJSON equivalente al polígono del área municipal de la ciudad de Granada.* URL: <https://gist.github.com/Alxe/eca7f2b8c878959d9558e3972069ceba>.
- [NetwAn] The GIS Encyclopedia, ed. *Network Analysis.* URL: [http://wiki.gis.com/wiki/index.php/Network\\_Analysis](http://wiki.gis.com/wiki/index.php/Network_Analysis).
- [PgRout] *pgRouting.* URL: <https://pgrouting.org/>.
- [SpaDa] OpenStreetMap y GeoFabrik, eds. *Data of Spain.* URL: <http://download.geofabrik.de/europe/spain.html>.