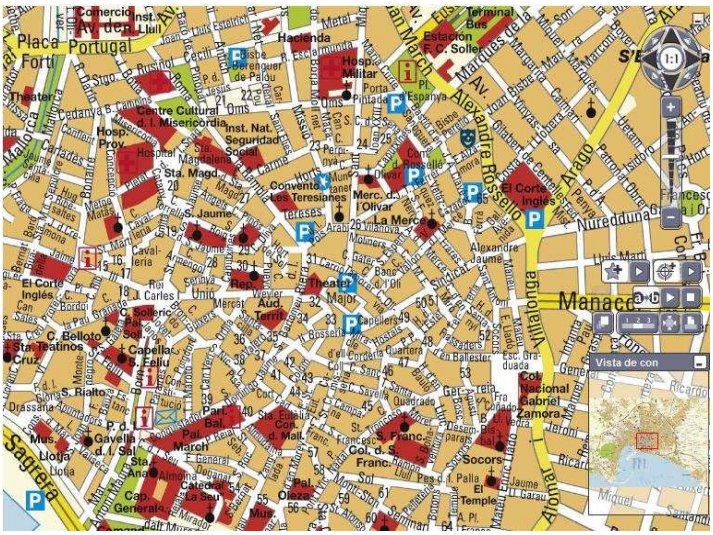
Palma Routing



Cálculo web de rutas con pgRouting, OpenStreetMap y OpenLayers

Proyecto de Fin de Carrera

Roshni Budhrani

b_roshni@uoc.edu

Proyecto de fin de carrera dirigido por Anna Muñoz Bollas

Semestre 2009/2010 Q2

Dedico este proyecto a todas las personas que me han acompañado estos años de carrera. A mis compañeros de universidad, que se han convertido en mis mejores amigos y de los que guardo muy buenos recuerdos; a mi familia, por haber aguantado mis largas y agónicas noches de estudio, y en especial a mi futuro marido, por haberme apoyado en todo momento y darme los ánimos que siempre me han hecho falta para seguir adelante. De no ser por todos vosotros, no hubiera llegado hasta aquí. Gracias.

Resumen

El objetivo de este proyecto es abordar y conocer el concepto de Sistema de Información Geográfica (SIG), así como las tecnologías utilizadas, desarrollos actuales, así como sus ventajas y utilidades. Para ello, el primer paso es tener una base teórica sobre conceptos relativos a la cartografía, la topología, la geodesia, así como los mismos sistemas SIG y los componentes que los forman. Pero el proyecto no sólo tiene un componente teórico sino que también práctico, por lo que para aplicar los conocimientos teóricos adquiridos en la primera parte, se implementa una aplicación que calcula y muestra el camino a seguir entre dos puntos de una geografía. Para ello, es necesario primero disponer de un conjunto de datos válido y óptimo de manera que sea posible aplicar unos algoritmos de cálculos de rutas que devolverán como resultado el camino a seguir entre dos puntos. Tanto el resultado obtenido como los puntos de inicio y fin de la ruta deben mostrarse sobre un mapa en un navegador. Como particularidad del PFC, el mapa a mostrar corresponde a la ciudad de Palma de Mallorca, y se establece como puntos de destino para las rutas los atractivos turísticos o puntos de interés de la ciudad. Finalmente, se definen las líneas futuras de trabajo.

Índice

1 Descripción del proyecto

El primer paso a realizar en un proyecto es precisamente determinar cómo es el proyecto a realizar. En este apartado se define exactamente cuál es el objetivo del proyecto y lo que se pretende conseguir.

1.1 Descripción

Este proyecto consiste en diseñar un servicio de enrutamiento entre dos puntos situados en la ciudad de Palma de Mallorca. Se trata de calcular el camino más corto entre dos puntos de una misma ciudad. El concepto de camino más corto es lo que se conoce como camino o ruta de menor coste.

Estrictamente hablando, existen miles de caminos que unen dos puntos. Por ejemplo, es posible ir de Barcelona a Madrid en línea recta, pero también sería posible hacerlo pasando por Sevilla. Al fin y al cabo, si se empieza en Barcelona y se termina en Madrid, todos los caminos que unen dichos puntos son considerados como posibles rutas entre dos puntos.

Otro ejemplo son las compañías de vuelo, donde para un trayecto de Palma de Mallorca a Sevilla es posible comprar un billete en un vuelo directo, aunque también se nos ofrece la posibilidad de viajar vía Barcelona. Es decir, es posible realizar la ruta Mallorca – Barcelona – Sevilla. En ambos casos el origen es Mallorca y el destino es Sevilla, pero está claro que un camino es más largo y lento que el otro. Normalmente, siempre se intenta coger el camino directo (en este caso Palma de Mallorca – Sevilla), para ahorrar tiempo. Es lo que se conoce como la ruta más corta o de menor coste.

Como servicio adicional, el punto de destino puede ser una atracción turística de la ciudad, de manera que alguien que se dirija a algún lugar emblemático de la ciudad tan sólo tendría que conocer su nombre y la localización actual en la que se encuentra el usuario. Finalmente se ha optado por la ciudad de Palma de Mallorca ya que se trata de la ciudad donde resido.

A grandes rasgos, la finalidad de este proyecto es calcular la ruta mínima entre dos puntos seleccionados por el usuario. El punto de origen es la posición del usuario, y el punto de destino es el punto seleccionado por el usuario. Es decir, el usuario desea ir desde su posición actual hasta su destino, y pretende obtener el camino más corto para ir a su meta.

Tecnológicamente hablando, existen varios proyectos libres en marcha que contienen información geográfica y que utilizaremos en el proyecto. Un ejemplo es OpenStreetMap, proyecto libre y colaborativo donde los usuarios aportan información de localizaciones. Los datos que se utilizarán como muestra en el proyecto se obtendrán de OpenStreetMap. A nivel visual, se utilizará OpenLayers para crear la interficie web que interactuará con la aplicación SIG.

1.2 Objetivos

Mediante este proyecto de fin de carrera se espera conseguir los siguientes objetivos.

1.2.1 Objetivos Generales

- ✔ Comprender los conceptos de la tecnología SIG y su metodología.
- ✔ Conocer la estructura de los diferentes tipos de datos con los que trabaja un SIG.
- ✔ Conocer los sistemas de almacenamiento estándar, tanto de información ráster como vectorial, y ser capaces de ubicar la información en las coordenadas que corresponda.
- ✔ Encontrar, generar y manipular datos geográficos.
- ✔ Saber plantear un proyecto SIG.
- ✔ Conocer las operaciones de análisis espacial y transformaciones en el SIG analizado.

1.2.2 Objetivos Específicos

- ✔ Conocer el estado actual del ecosistema de aplicaciones libres, y en particular las que corresponden a los servidores de mapas.
- ✔ Trabajar con datos de OpenStreetMap.

- ✔ Analizar las diferentes funciones de cálculo de rutas de los que dispone pgRouting para establecer el escenario óptimo de uso de cada una de ellas.
- ✔ Implementar un visor de la ruta obtenida con OpenLayers.

1.3 Alcance

Se considera dentro del alcance del proyecto:

- ✔ Estudio de la tecnología SIG y su metodología, así como de conceptos generales relacionados con la cartografía, como la topología, la orientación, etc. Este estudio incluye también la estructuras de los diferentes tipos de datos.
- ✔ Búsqueda, recopilación y tratamiento de los datos que se tomarán de referencia en el proyecto (situación geográfica de algunas atracciones turísticas de Palma).
- ✔ Creación de la base de datos (PostgreSQL con la extensión PostGIS) que se utilizará en el proyecto.
- ✔ Estudio de la herramienta OpenStreetMap con ámbito Palma de Mallorca para obtener los datos de las localizaciones a insertar en la base de datos.
- ✔ Estudio de los algoritmos de cálculo de ruta que dispone pgRouting y de su correcto funcionamiento junto con la geodatabase PostGIS.
- ✔ Implementación del portal web que servirá para consultar los servicios de enrutamientos. El portal web se implementará mediante UMN MapServer junto con OpenLayers para la visualización de las rutas.
- ✔ Redacción de la memoria del proyecto.
- ✔ Realización de la presentación virtual.
- ✔ Participación en el debate virtual.

2 Planificación del proyecto

Una vez se define el proyecto a realizar, es el momento de planificar el tiempo que se va a invertir en llevarlo a cabo. Para ello se identifican las tareas que lo componen de manera que se pueda realizar una estimación temporal de cada una de ellas. De esta manera, tenemos una duración establecida para la totalidad del proyecto, así como una planificación de tareas que puede llegar a ser muy útil de cara a la metodología a llevar a cabo a lo largo de todo el desarrollo del proyecto.

2.1 Descripción de las tareas

El enunciado del proyecto establece una serie de tareas divididas en bloques, que corresponderán a diferentes aspectos del proyecto.

Bloque 1

- ✔ Instalación del software

Bloque 2

- ✔ Recopilación de datos
- ✔ Tratamiento e importación de los datos

Bloque 3

- ✔ Cálculo de rutas mediante pgRouting
- ✔ Comparativas de resultados mediante diferentes algoritmos

Bloque 4

- ✔ Diseño de la interfaz de usuario

No existe una relación unívoca de los diferentes bloques del proyecto con cada una de las entregas de las pacs. Los bloques agrupan actividades similares en cuanto a su contenido. Por otro lado, las pacs contienen diferentes tareas del proyecto según la planificación que se presenta en este documento. A grandes rasgos, los bloques 1 y 2

corresponden a la entrega de la pac2, mientras que los bloques 3 y 4 están planificados para la entrega de la pac3.

A lo largo de todo el proyecto se procederá al estudio de los conceptos de cartografía y geografía que irán surgiendo en cada fase. De esta manera, se profundizará en los conceptos que forman los sistemas de información geográfica, pues son la base de las tareas a realizar en el proyecto.

2.1.1 Instalación de software

Se procederá a la instalación del software necesario para llevar a cabo el proyecto. Básicamente se trata de lo siguiente.

- **UMN MapServer**

Se trata de un entorno de desarrollo de código abierto para construir aplicaciones de Internet espaciales. Se utiliza para la creación de aplicaciones SIG con el fin de visualizar, consultar y analizar información geográfica.

- **Base de datos PostgreSQL con las extensiones PostGIS y pgRouting**

Software de base de datos con la se va a trabajar.

PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos, convirtiéndola en una base de datos espacial para su utilización en un SIG.

PgRouting es una extensión para bases de datos geográficas PostGIS/PostgreSQL que proporciona herramientas para servicios basados en la localización.

- **OpenLayers**

OpenLayers es una biblioteca javascript de código abierto para mostrar mapas interactivos en navegadores web. Ofrece una API para crear aplicaciones parecidas a las que se crean a partir de *Google Maps*.

- **Firebug**

Se trata de una extensión de Firefox que permite analizar y depurar el código fuente de una página web de manera instantánea.

2.1.2 Recopilación de datos

Se trata de una tarea importante, ya que a partir de esta información basaremos todo el proyecto. Por ello es determinante la búsqueda de fuentes de datos cartográficos. En lo que concierne este proyecto, la información necesaria a recopilar es la que nos permitirá caracterizar correctamente la cartografía de los puntos de interés seleccionados para la muestra del proyecto.

La búsqueda de datos se realizará en OpenStreetMap, que básicamente consiste en un proyecto colaborativo para crear mapas libres utilizando información geográfica introducida por los usuarios registrados en el proyecto. Normalmente, los usuarios suelen subir sus coordenadas desde el GPS para crear localizaciones nuevas o para corregir datos vectoriales ya existentes. También es posible introducir o corregir dicha información manualmente, digitalizando los mapas existentes en el proyecto.

Para nuestro caso, buscaremos datos de ejemplos que constituirán parte del proyecto cartográfico de Palma de Mallorca.

2.1.3 Tratamiento e importación de los datos

Los datos recopilados en la tarea anterior debe ser incorporada al sistema. Para ello se procede al estudio de la estructura de los datos y se construirá un modelo de base de datos adecuado para dicha estructura. Básicamente se procederá al estudio de la información recopilada y su relación, así como las características de los atributos y valores únicos.

Se creará un modelo de entidad relación para el sistema, y consecuentemente se procederá a la creación de los objetos de base de datos para tal.

2.1.4 Cálculo de rutas mediante pgRouting

PgRouting se utiliza mediante consultas a la base de datos en la que se almacenan los datos. Incluye varios algoritmos para el cálculo de rutas, herramientas de importación para datos de OpenStreetMap.

Para la tarea se crearán una serie de consultas que se utilizarán en el cálculo de rutas. A partir de allí, se utilizarán procedimientos para que a partir de una localización, se devuelva la ruta de mínimo coste para un destino concreto. Dado que pgRouting incluye varios algoritmos, se estudiarán algunos de ellos en función al rendimiento y a las necesidades del proyecto.

2.1.5 Comparativas de resultados mediante diferentes algoritmos

En esta tarea se estudiará para un mismo ejemplo (origen A y destino B), los diferentes resultados que se proporcionan a partir de los diferentes algoritmos presentes en pgRouting. Se propondrán diferentes ejemplos, a partir de los cuales será posibles realizar gráficas comparativas entre los algoritmos.

2.1.6 Diseño de la interfaz de usuario

Mediante OpenLayers es posible crear una interfaz web que permitirá obtener las rutas deseadas gráficamente. Se ofrecerá al usuario la opción de seleccionar el algoritmo de búsqueda, y se ofrecerá como resultado la ruta obtenida mediante pgRouting.

2.2 Relación de actividades

A continuación se detallan las actividades que forman todo el proyecto, y que corresponden a grosso modo a los cuatro bloques expuestos en el enunciado. Para empezar, las actividades se han agrupado de la siguiente manera.

Entrega	Tarea
PAC 1	Elaboración del plan de trabajo
PAC 2	Instalación del software. Recopilación y tratamiento de datos
PAC 3	Cálculo de rutas y diseño de la interficie web
PAC 4	Revisión de la memoria del proyecto y creación de la presentación virtual

2.2.1 PAC 1 – Elaboración del plan de trabajo

Redacción Plan de Trabajo		Horas 30
Inicio 02/03/2010	Fin 10/03/2010	Páginas 19
Planificación del proyecto y redacción del documento de trabajo		

Entrega borrador Plan de Trabajo		Horas 0
Inicio 10/03/2010	Fin 10/03/2010	Páginas 0
Entrega del borrador del Plan de Trabajo para su corrección		

Revisión informe Plan de Trabajo		Horas 9
Inicio 11/03/2010	Fin 13/03/2010	Páginas 25
Revisión del documento entregado como borrador y mejora de los aspectos corregidos o modificados a partir de la corrección realizada		

Entrega final Plan de Trabajo		Horas 0
Inicio 13/03/2010	Fin 13/03/2010	Páginas 0
Entrega del documento del Plan de Trabajo del proyecto		

2.2.2 PAC 2 – Instalación del software. Recopilación y tratamiento de datos

Instalación del software

Instalación de todo el software necesario para llevar a cabo el proyecto.

UMN Mapserver		Horas 2
Inicio 14/03/2010	Fin 14/03/2010	Páginas 0
Instalación de UMN Mapserver como entorno de desarrollo de la aplicación SIG		

Geodatabase PostgreSQL		Horas 2
Inicio 14/03/2010	Fin 14/03/2010	Páginas 0
Instalación de PostgreSQL como SGBD para gestionar la base de datos de puntos de		

interés y localizaciones a utilizar como muestra en el proyecto

Extensiones PostGIS y pgRouting		Horas 1
Inicio 14/03/2010	Fin 14/03/2010	Páginas 0
Instalación de las extensiones PostGIS y pgRouting como complemento para la base de datos y añadir funcionalidad específica para trabajar con SIG		

OpenLayers y Firebug		Horas 1
Inicio 14/03/2010	Fin 14/03/2010	Páginas 0
Instalación de OpenLayers como aplicación para crear la interficie web de la aplicación SIG		

Recopilación de datos

Recopilación de los datos que se utilizarán como muestra en el proyecto. Localizaciones de los distintos puntos de interés turísticos de Palma de Mallorca.

Estudio previo de conceptos cartográficos		Horas 10
Inicio 15/03/2010	Fin 18/03/2010	Páginas 4
Búsqueda de información acerca de conceptos cartográficos útiles para el sistema SIG a crear. Estudio de los diferentes sistemas de coordenadas así como de cualquier tipo de información que pueda ser útil para el proyecto		

Búsqueda en OpenStreetMap		Horas 16
Inicio 18/03/2010	Fin 22/03/2010	Páginas 0
Búsqueda en OpenStreetMap de los datos de muestra a utilizar en el proyecto como información de referencia. En este caso se trata de las localizaciones de los puntos turísticos con los que la aplicación interactuará y que se guardarán en la base de datos		

Tratamiento e importación de los datos

- Diseño del modelo de datos

Diseño del modelo de datos del sistema.

Modelo conceptual		Horas 4
Inicio 22/03/2010	Fin 23/03/2010	Páginas 2
Creación del modelo conceptual de la información a almacenar		

Diseño lógico de datos		Horas 4
Inicio 24/03/2010	Fin 25/03/2010	Páginas 2
Transformación de las entidades persistentes del modelo conceptual en un modelo de datos relacional		

Diseño físico de datos		Horas 4
Inicio 25/03/2010	Fin 27/03/2010	Páginas 2
Creación del diseño físico de la base de datos		

Implementación del modelo de datos		Horas 4
Inicio 27/03/2010	Fin 27/03/2010	Páginas 2
Creación de la base de datos a utilizar por la aplicación		

- Importación de datos

Digitalización de los datos recopilados e importación de los mismos en la base de datos.

Digitalización de datos		Horas 12
Inicio 28/03/2010	Fin 30/03/2010	Páginas 0
Digitalización de los datos obtenidos de manera que los datos de muestra sean fácilmente convertibles en el formato de datos requerido para su posterior inserción en la base de datos		

Inserción de datos en la BD		Horas 22
Inicio 31/03/2010	Fin 06/04/2010	Páginas 0
Inserción de la información digitalizada en la base de datos		

Documentación

Redacción del informe relativo a esta entrega

Redacción informe PAC 2		Horas 20
Inicio 06/04/2010	Fin 11/04/2010	Páginas 40
Redacción el documento del informe de la entrega de la pac 2		

Entrega borrador PAC2		Horas 0
Inicio 11/04/2010	Fin 11/04/2010	Páginas 0
Entrega del borrador de la pac 2 para su corrección		

Revisión informe PAC 2		Horas 9
Inicio 12/04/2010	Fin 14/04/2010	Páginas 40
Revisión del documento entregado como borrador y mejora de los aspectos corregidos o modificados a partir de la corrección realizada		

Entrega final informe PAC 2		Horas 0
Inicio 14/04/2010	Fin 14/04/2010	Páginas 0
Entrega del documento que contiene el informe de la PAC2		

2.2.3 PAC 3 – Cálculo de rutas y diseño de la interficie web

Cálculo de rutas mediante pgRouting

Proceso de cálculo de rutas mediante pgRouting. Estudio de los diferentes algoritmos existentes y comparativa de los resultados obtenidos.

Estudio previo de pgRouting		Horas 6
Inicio 15/04/2010	Fin 18/04/2010	Páginas 0
Estudio previo de los conceptos relacionados con pgRouting, su funcionamiento.		

Pruebas varias de las herramientas que proporciona a fin de tener una primera toma de contacto

Implementación de las consultas y procedimientos		Horas 12
Inicio 18/04/2010	Fin 21/04/2010	Páginas 6
Implementación de las consultas y procedimientos a utilizar para realizar los cálculos de rutas. Diseño, prototipo e implementación para cada uno de los procedimientos		

Pruebas de cálculo de rutas		Horas 14
Inicio 22/04/2010	Fin 25/04/2010	Páginas 10
Esquematización de las diferentes pruebas a realizar para cada uno de los algoritmos. Juego de pruebas y resultados		

Comparativa de algoritmos		Horas 16
Inicio 25/04/2010	Fin 01/05/2010	Páginas 8
Pruebas de cálculo de rutas utilizando diferentes algoritmos y comparativas de los resultados obtenidos por cada uno de ellos		

Diseño de la interfaz de usuario

Creación de la interfaz de usuario e integración con la aplicación SIG

Estudio previo de OpenLayers		Horas 4
Inicio 01/05/2010	Fin 02/05/2010	Páginas 0
Estudio previo de las herramientas proporcionadas por OpenLayers y su funcionalidad. Primera toma de contacto		

Creación de la interfaz de usuario		Horas 16
Inicio 02/05/2010	Fin 06/05/2010	Páginas 0
Creación de una interfaz de usuario mediante OpenLayers para integrar junto con la aplicación SIG		

Pruebas de usuario de la interfaz		Horas 8
Inicio 06/05/2010	Fin 09/05/2010	Páginas 4
Realización de las pruebas pertinentes para verificar el correcto funcionamiento de la interfaz creada junto con la aplicación SIG		

Documentación

Redacción del informe relativo a esta entrega

Redacción informe PAC 3		Horas 20
Inicio 09/05/2010	Fin 16/05/2010	Páginas 35
Redacción el documento del informe de la entrega de la pac 3		

Entrega borrador PAC3		Horas 0
Inicio 16/05/2010	Fin 16/05/2010	Páginas 0
Entrega del borrador de la pac 3 para su corrección		

Revisión informe PAC 3		Horas 12
Inicio 16/05/2010	Fin 19/05/2010	Páginas 35
Revisión del documento entregado como borrador y mejora de los aspectos corregidos o modificados a partir de la corrección realizada		

Entrega final informe PAC 3		Horas 0
Inicio 19/05/2010	Fin 19/05/2010	Páginas 0
Entrega del documento que contiene el informe de la PAC3		

2.2.4 PAC 4 - Memoria del proyecto y presentación virtual

Proceso final del proyecto. Preparación de la memoria completa del proyecto y creación de la presentación virtual a presentar ante el Tribunal de Evaluación.

Creación presentación virtual		Horas 19
Inicio 25/05/2010	Fin 30/05/2010	Páginas 0
Realización de una presentación virtual donde se esquematiza y resume el proyecto completo para permitir al Tribual de Evaluación tener una perspectiva inicial del		

proyecto y de sus resultados

Redacción memoria del proyecto		Horas 14
Inicio 30/05/2010	Fin 03/06/2010	Páginas 80
Revisión del contenido de la memoria del proyecto que se ha ido elaborando en cada una de las entregas.		

Entrega borrador presentación virtual y memoria del proyecto		Horas 0
Inicio 03/06/2010	Fin 03/06/2010	Páginas 0
Entrega de la presentación virtual y de la memoria como borrador para su posterior corrección		

Revisión presentación virtual y memoria del proyecto		Horas 15
Inicio 05/06/2010	Fin 07/06/2010	Páginas 85
Revisión de la memoria final del proyecto y de la presentación. Mejora de los aspectos corregidos o modificados a partir de la corrección realizada		

Entrega final del proyecto		Horas 0
Inicio 07/06/2010	Fin 07/06/2010	Páginas 0
Entrega final del proyecto, que contendrá la presentación virtual y la memoria completa del proyecto		

2.2.5 Debate virtual

Debate virtual		Horas -
Inicio 21/06/2010	Fin 23/06/2010	-
Periodo en el que el Tribunal de Evaluación podrá formular preguntar sobre el proyecto. El plazo para responder es de 24 horas.		

3 Calendario

Se detalla a continuación el calendario del proyecto para cada una de sus fases.

3.1 Calendario general del proyecto

La distribución de horas disponibles para el proyecto se ha realizado de la siguiente manera.

- Lunes a Jueves: 3 horas
- Fines de semana: 6 horas

Se especifican como vacaciones los siguientes periodos.

- 17 de abril.
- 15 de mayo.
- De 20 a 24 de mayo, ambos inclusive.

A continuación se muestra el diagrama de Gantt del calendario general del proyecto.

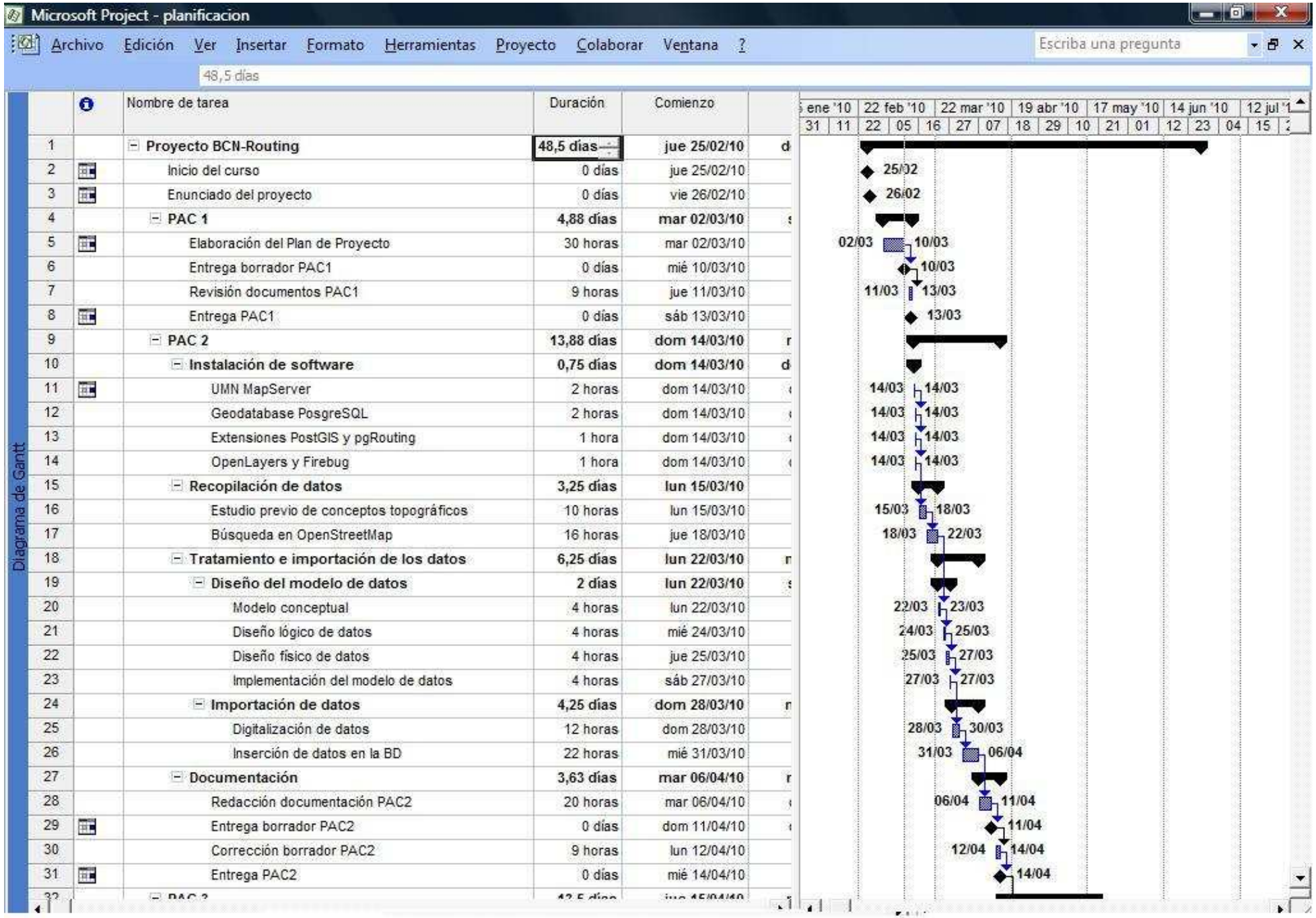


Figura 1. Planificación del proyecto (1)

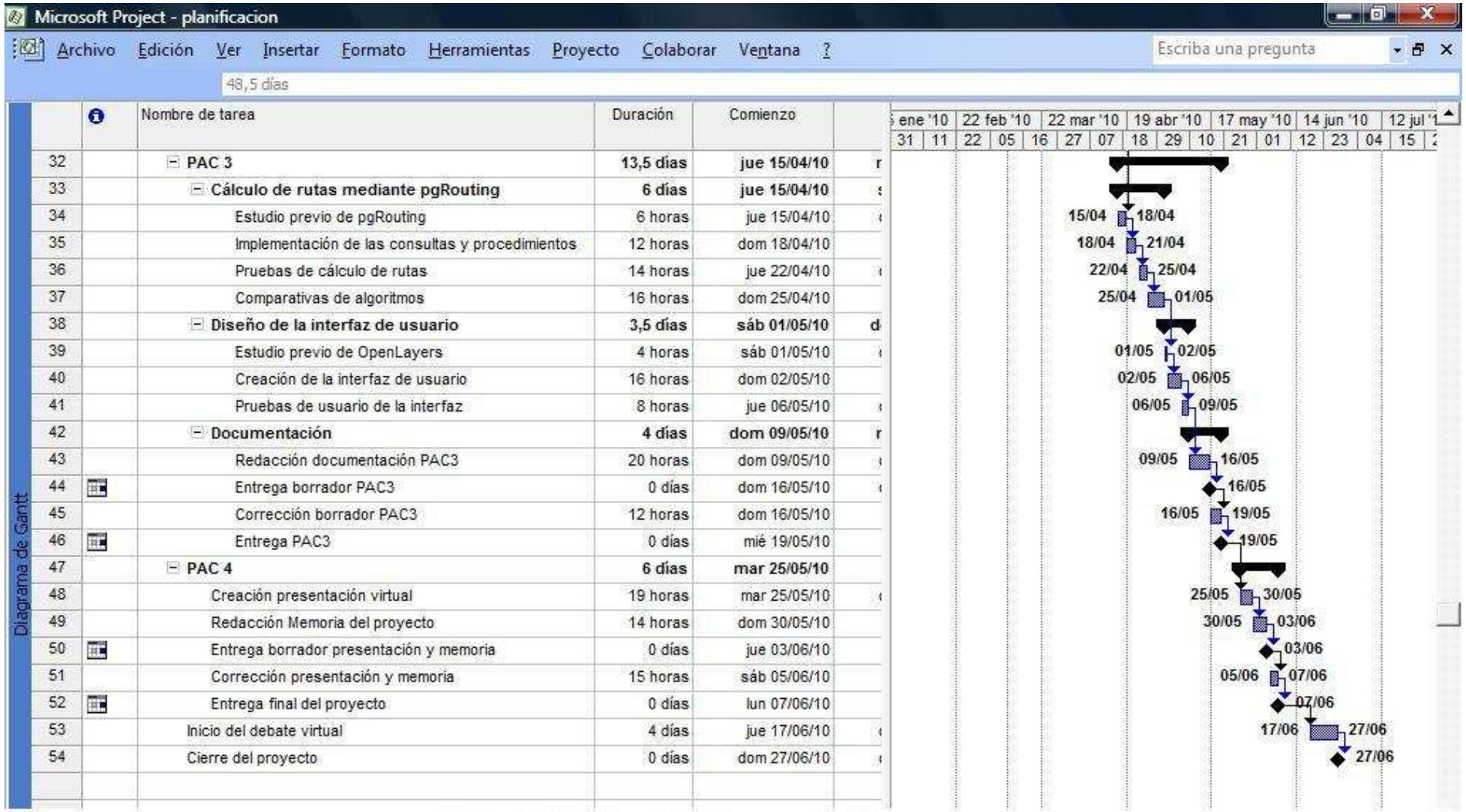


Figura 2. Planificación del proyecto (2)

3.2 Hitos principales del proyecto

A continuación se muestra un listado de los principales hitos del proyecto. Se observa que las fechas de entrega de la pac 2 y 3 se han retrasado tan sólo un día. No se ha querido retrasar más dichas fechas para poder tener ese margen en caso de imprevisto.

Fecha	Descripción
25/02/2010	Inicio del curso
26/02/2010	Enunciado proyecto
10/03/2010	Entrega borrador PAC 1
13/03/2010	Entrega PAC 1
11/04/2010	Entrega borrador PAC 2
14/04/2010	Entrega PAC 2
16/05/2010	Entrega borrador PAC 3
19/05/2010	Entrega PAC 3
03/06/2010	Entrega borrador PAC 4
07/06/2010	Entrega PAC 4
21/06/2010	Inicio del debate virtual
23/06/2010	Cierre del proyecto

4 Evaluación del material

La plataforma de desarrollo a utilizar en el proyecto estará compuesta por el software y la maquinaria listados a continuación.

4.1 Requerimientos de maquinaria

La estación de trabajo es un ordenador portátil marca Hewlett-Packard con las siguientes características. Se ha utilizado esta máquina para todas las prácticas y trabajos realizados durante mis estudios en la UOC.

- Intel Core 2 Duo T7500 2.20 Ghz
- 2 GB de memoria RAM
- Disco duro Samsung HM250JI, 250GB SATA 5400 rpm
- Tarjeta gráfica NVIDIA GeForce 8400M GS 128 MB
- Pantalla panorámica de alta definición 14,1" 1024 x 768
- Conexión a Internet mediante WiFi ADSL 1Mbps

4.2 Requerimientos de software

El software utilizado para crear la memoria y la presentación virtual será el siguiente. No se listarán aquí el software propio a instalar para la realización del proyecto, puesto que forma parte del proyecto en sí.

- Windows Vista Home Premium Service Pack 2 (sistema operativo)
- VMWare con máquina virtual Ubuntu Xfce 8.04 (sistema operativo)
- Microsoft Office Project 2003 (diagramas de Gantt)
- OpenOffice Writer (redacción de la memoria)
- OpenOffice Calc (cálculos y tablas adicionales)
- MagicDraw UML (diagramas y esquemas)
- Microsoft PowerPoint 2007 (diapositivas presentación virtual)
- Camtasia Studio (vídeo presentación del proyecto)

5 Análisis de Riesgos

Se entiende como riesgo en un proyecto cualquier problema o eventualidad surgida durante el desarrollo del mismo.

Riesgo	Avería de la estación de trabajo
Descripción	Imposibilidad temporal de seguir con el desarrollo del proyecto debido a una avería en la maquinaria (tanto hardware como software) utilizada para llevar a cabo el proyecto
Impacto	Cumplimiento de los hitos del proyecto
Probabilidad	Baja
Acción	Instalación del software en otro equipo y seguir trabajando intentando apurar el tiempo perdido para volver a estar acorde a la planificación. *Backup* diario del proyecto: datos, informes, presentación, etc., de manera que la reinstalación en otro equipo o en el mismo sea lo más rápido posible.

Riesgo	Problemas de disponibilidad
Descripción	Durante el proyecto pueden surgir ocupaciones o imprevistos que impidan llevar a rajatabla la planificación del proyecto
Impacto	Cumplimiento de los hitos del proyecto
Probabilidad	Media
Acción	Recuperar el tiempo intentando apurar de manera que el impacto del desvío de planificación sea mínimo

Riesgo	Errores en la planificación debido a la falta de experiencia en el sector
Descripción	La planificación puede haberse realizado de manera incorrecta si las estimaciones no son las que realmente deberían ser según la magnitud de la tarea.
Impacto	Cumplimiento de los hitos del proyecto

Probabilidad	Alta
Acción	Dedicar más horas al proyecto desde el principio de manera que se evite cualquier desvío posible. A medida que el proyecto va avanzado, se reajusta la planificación para corregir las mejoras que se puedan hacer. En caso de retraso en la planificación, dedicar más horas al día a las tareas actuales para minimizar en la medida de lo posible el desvío.

6 ¿Qué es un SIG?

Este apartado corresponde al estudio de los conceptos asociados a la tecnología de los SIG. Estos conceptos ayudan a la comprensión de la temática del proyecto y forman el pilar del mismo. En primer lugar, se define el concepto de SIG y sus utilidades, así como sus aplicaciones y una perspectiva global actual de la tecnología así como su evolución. Tanto este apartado como el siguiente, referente a Cartografía, constituyen la base teórica sobre la cual se fundamenta todo este trabajo.

6.1 Descripción

Un Sistema de Información Geográfica[1] o SIG es una integración organizada de hardware, software y datos geográficos diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada.

Los sistemas de información geográfica utilizan la información geográfica (base de datos georeferenciada) almacenada en un sistema de información para demostrar su efectividad, especialmente la resolución de problemas espaciales, servir de soporte para la toma de decisiones y para ayudar en la planificación.

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Las principales cuestiones que puede resolver un SIG son los siguientes.

- ✔ **Localización**, preguntar por las características de un lugar concreto.
- ✔ **Condición**, el cumplimiento o no de unas condiciones impuestas al sistema.
- ✔ **Tendencia**, comparación entre situaciones temporales o espaciales distintas de alguna característica.

1 Definición obtenida de la Wikipedia: http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica

- ✔ **Rutas,** cálculo de rutas óptimas entre dos o más puntos.
- ✔ **Pautas,** detección de pautas espaciales.
- ✔ **Modelos**, generación de modelos.

Los datos SIG representan los objetos del mundo real (carreteras, ubicaciones, etc.). Existen dos formas de almacenar los datos SIG: en formato raster y vectorial. De forma resumida, diremos que un tipo de datos raster es básicamente cualquier tipo de imagen digital representada en mallas. Este formato divide el espacio en celdas regulares donde cada una de ellas representa un único valor. Por otro lado, el formato de datos vectorial guarda las características geográficas como vectores, tal y como indica su nombre. Una base de datos SIG contiene la información bien en un formato, bien en otro, bien en una combinación de los dos. A partir de esta base de datos, se crean mapas según las necesidades del usuario. Además de producir y almacenar mapas, un SIG puede procesarlos, ya que los datos del mundo real están almacenados en formato digital, a diferencia de los mapas de papel convencional.

Una de las propiedades fundamentales de los SIG y que ha supuesto el verdadero éxito de esta tecnología es que va más allá de la cartografía y de una base de datos con información de objetos. Si bien con la cartografía obtenemos respuestas del tipo: *donde está este objeto? A cuánta distancia se encuentran estos dos puntos?*, etc, ahora con los SIG obtenemos respuestas a preguntas de tipo:

- ✔ *Cuál es el camino más corto entre A y B? Y cuál es el camino más corto entre A y B si evitamos pasar por X y Z?*
- ✔ *Cuántos objetos del tipo A hay a menos de 500 metros de B?*

Un SIG no solamente es un software instalado en un ordenador, sino que es un conjunto de componentes, listados a continuación.

- ✔ Tecnología (programas y equipos, software y hardware)
- ✔ Ideas
- ✔ Personal
- ✔ Red
- ✔ Datos
- ✔ Métodos

Cualquier uso de un SIG debe incluir obligatoriamente cada uno de estos componentes. No son componentes de un campo de aprendizaje, sino del proceso de diseño, creación y manejo de un SIG.

La información geográfica puede ser consultada, transferida, transformada, superpuesta, procesada y mostrada utilizando numerosas aplicaciones de software. El manejo de este tipo de sistemas suele ser llevado a cabo generalmente por profesionales de diversos campos del conocimiento con experiencia en Sistemas de Información Geográfica (cartografía, geografía, topografía, etc.), ya que el uso de estas herramientas requiere un aprendizaje previo. Aunque existen herramientas gratuitas para ver información geográfica, el acceso del público en general a los geodatos está dominado por los recursos en línea, como Google Earth. Originalmente hasta finales de los 90, el software era un producto independiente. Sin embargo, con el mayor acceso a Internet y a la demanda de datos geográficos distribuidos, el software SIG ha cambiado gradualmente su perspectiva hacia la distribución de datos a través de redes. Los SIG que en la actualidad se comercializan son combinaciones de varias aplicaciones interoperables.

Hoy por hoy, dentro del software SIG se distingue a menudo cuatro grandes tipos de programas informáticos.

- ✔ **SIG de escritorio**. Son la categoría de software más ampliamente utilizada. El origen de este tipo de software es el ordenador personal. Los SIG de escritorio se ejecutan en el mismo PC y ofrecen un gran número de herramientas para una gran variedad de usuarios en diferentes campos.

  Los software SIG de escritorio ofrecen un amplio rango de aplicaciones, desde simples visualizadores, como ArcReader, hasta software de creación de mapas y análisis, como GeoMedia o MapInfo, así como tecnología avanzada en sistemas de edición y análisis profesional, como ArcInfo.
- ✔ **SIG web**. Son productos localizados en un servidor al cual acceden los usuarios a través de una red, lo que implica que dichas aplicaciones tengan una arquitectura cliente-servidor. Los SIG web tienen una interficie de usuario que ofrecen funcionalidades de consulta, edición y análisis espacial.

  A día de hoy, existen varias opciones en Internet para ejecutar operaciones con información geográfica, como generar mapas, calcular rutas óptimas, visualizar

datos específicos, e incluso hacer análisis basados en criterios seleccionados por el usuario.

- ✔ **Componentes de desarrollo SIG**. Son paquetes de herramientas de funciones SIG. Es necesario tener conocimientos de programación para poder implementar los componentes SIG, los cuales se utilizan para la creación de aplicaciones SIG; es decir, programas con unas funcionalidades específicas.
  Se trata de productos de componentes SIG, como por ejemplo ArcGis Engine de ESRI, o MapX de MapInfo. La mayoría de las aplicaciones SIG que utilizan estos componentes están implementan bajo los estándares de Microsoft Net.
- ✔ **SIG móviles**. Sistemas ligeros diseñados para ser utilizados en dispositivos móviles. Con la adopción generalizada por parte de estos dispositivos de localización integrados, el software SIG permite utilizarlos para la captura y manejo de datos en campo. En el pasado la recogida de datos en campo se realizaba mediante la señalización de la información geográfica en un mapa de papel, y a continuación se volcaba esa información a formato digital una vez de vuelta frente al ordenador. Hoy en día a través de la utilización de dispositivos móviles, los datos geográficos pueden ser capturados directamente mediante levantamientos de información en trabajo de campo.
  Actualmente, los SIG móviles ofrecen un gran número de funcionalidades similares a los SIG de escritorio de hace unos años. Un ejemplo claro es ArcPad de ESRI. Se trata de una aplicación SIG destinada a dispositivos móviles y orientada a proyectos de captura de datos y recogida de información geográfica mediante posicionamiento GPS en campo. Otro ejemplo son los *smartphones*, capaces de trabajar con grandes cantidades de datos a pesar de sus diminutas dimensiones. Estos teléfonos funcionan de manera que se conectan a la red cuando el proceso lo requiere, y por tanto, utilizan los datos y las aplicaciones de los servidores.

6.2 Web Mapping

El mundo de los SIG ha asistido en los últimos años a una explosión de aplicaciones destinadas a mostrar y editar cartografía en entornos web Google Maps u OpenStreetMap[2]. Estos sitios web dan al público acceso a enormes cantidades de datos geográficos. Algunos de ellos utilizan software que, a través de una API,

2 Utilizamos OpenStreetMap para exportar los datos a utilizar en el proyecto. Ver apartado 8

permiten a los usuarios crear aplicaciones personalizadas. Estos servicios ofrecen por lo general callejeros, imágenes aéreas y de satélite o funcionalidades de enrutamiento. Todo esto es posible gracias al desarrollo de Internet y las redes de comunicación, pero también existe otro factor clave; los estándares OGC que facilitan la interoperabilidad de los datos espaciales[3].

6.3 El futuro de los SIG

Muchas disciplinas se han beneficiado de la tecnología subyacente en los SIG. Este activo mercado se ha traducido en una reducción de costes y mejoras continuas en los componentes de hardware y software de sistemas. Esto ha provocado que el uso de esta tecnología haya sido asimilada por multitud de universidades e instituciones de diversa índole.

Un claro ejemplo es el proyecto IDEE (Infraestructura de Datos Espaciales de España) impulsado por el Ministerio de Fomento de España. Su objetivo es integrar a través de Internet los datos, metadatos, servicios e información geográfica que se producen en España, facilitando a todos los usuarios la localización, identificación, selección y acceso a dichos recursos a través del portal www.idee.es.

A nivel europeo, existe el proyecto *INSPIRE (*Infrastructure for Spatial Information in Europe*)*, iniciativa de la Comisión Europea cuyo objetivo es la creación de una *Infraestructura de Datos Espaciales en* Europa. *Inspire* ha sido desarrollada con el propósito de hacer disponible información geográfica, relevante y de calidad de la Comunidad Europea. *INSPIRE* es el primer paso de una amplia iniciativa multilateral que inicialmente dirigirá su interés sobre la información espacial necesaria para políticas medioambientales y que estará disponible para satisfacer las necesidades prácticas de otras áreas, tales como la agricultura y el transporte.

Multitud de organizaciones se han creado respecto a esta tecnología. Aquí en España existen los Centros Cartográficos de España, establecidos tanto a nivel nacional como a nivel autonómico. A nivel nacional citamos por ejemplo el *Instituto Geológico y Minero de España* dirigido por el Ministerio de Educación y Ciencia, o el *Centro Nacional de Información Geográfica* del Ministerio de Fomento.

3 Ver apartado 6.3.1 sobre estándares OGC

A nivel autonómico, podemos citar el *Servicio de Información Territorial de Las Islas Baleares S.A. (sitibsa)* dirigido por la C.A. de las Islas Baleares, o el *Institut Cartogràfic de Catalunya*, de la C.A. de Cataluña.

A nivel internacional se ha creado la OGC, descrita a continuación.

6.3.1 OGC (Open Geospatial Consortium)

La organización Open Geospatial Consortium (http://www.opengeospatial.org/) fue creado en 1994 y agrupa a 372 organizaciones públicas y privadas. Su fin es la definición de estándares abiertos e interoperables dentro de los SIG y de la World Wide Web (www) y persigue acuerdos entre las diferentes empresas del sector que posibiliten la interoperación de sus sistemas de geoprocesamiento y facilitar el intercambio de la información geográfica en beneficio de los usuarios.

Las especificaciones más importantes y representativas surgidas del OGC son las enumeradas a continuación.

- ✔ GML. Lenguaje de Marcado Geográfico
- ✔ KML. Keyhole Markup Language, lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones.
- ✔ WFS. Web Feature Service, servicio de entidades vectoriales que proporciona la información relativa a la entidad almacenada en una capa vectorial que reúnen las características formuladas en la consulta.
- ✔ WMS. Web Map Service, servicio de mapas en la web que produce mapas en formato imagen a la demanda para ser visualizados por un navegador web.
- ✔ WCS. Web Coverage Service.
- ✔ CSW. Web Catalogue Service.

El servicio Web Map Service (WMS) produce mapas de datos referenciados espacialmente de forma dinámica a partir de información geográfica. Este estándar internacional define un mapa como una representación de la información geográfica en forma de un archivo digital. Los mapas producidos se generan normalmente en un formato de imagen como PNG, GIF o JPEG, y opcionalmente como gráficos vectoriales en formato SVG o WebCGM.

Por otro lado, WFS es un servicio estándar que ofrece un interfaz de comunicación que permite interactuar con los mapas servidos por WMS, como por ejemplo editar o analizar la imagen ofrecida por WMS. Para realizar estas operaciones se utiliza el lenguaje GML, estándar a través del cual se transmiten las órdenes WFS.

7 ¿Qué es la cartografía?

Este apartado trata algunos aspectos básicos relacionados con la cartografía. Se define el término de cartografía, así como conceptos básicos como los sistemas de coordenadas y las proyecciones. Este apartado constituye junto al anterior, la base teórica sobre la que se apoya el proyecto.

7.1 Descripción

Se considera Cartografía[4] como la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos y territoriales, entre otros. A grandes rasgos, se trata de implementar una representación de la Tierra sobre una superficie plana llamada mapa. De manera simple, un mapa es la representación del mundo real reducido a puntos, líneas y polígonos mediante el uso de símbolos gráficos.

Al tener la Tierra una forma esférica se debe utilizar un sistema de proyecciones para poder representar una superficie esférica a una superficie plana[5].

7.2 Escala de un mapa

Un factor esencial a la hora de elaborar mapas es la escala a utilizar, que consiste en la proporción entre los elementos representados en un mapa y la realidad. El tamaño final del mapa, así como la precisión en la representación de los elementos dependen de la selección de una escala apropiada.

En cartografía, la escala de un mapa es la relación constante entre una distancia medida sobre un mapa o plano y la distancia correspondiente medida sobre el terreno representado. Normalmente, se expresa la escala como una fracción o una proporción, por ejemplo 1/10000 o 1:10000. Esta representación en forma de fracción de la escala significa que una unidad de medida en el mapa, por ejemplo, 1 cm, representa 10000 cm (100 m) en el terreno. Como el rango de escala es un cociente, es válida en cualquiera de las unidades en las que se expresan los elementos de la fracción, siempre y cuando ambos elementos utilicen la misma unidad.

4 Definición obtenida de la Wikipedia: http://es.wikipedia.org/wiki/Cartografía

5 Más información sobre sistemas de proyecciones en el apartado 7.4

Cuanto más grande es una escala, más pequeño es el denominador de la fracción. Los mapas de escalas grandes, en general suelen mostrar más detalle que los mapas de escalas pequeñas puesto que en una escala grande hay más espacio en el mapa para mostrar entidades. Los mapas de escalas pequeñas muestran menos detalle que los mapas de escala grande pero recubren porciones de terreno más grande. Éstos no suelen ser suficientes para mostrar todos los detalles disponibles. Por ejemplo, en un mapa de una escala pequeña, una ciudad deberá representarse como un punto, mientras que en un mapa de escala grande puede llegar a distinguirse hasta el nivel de las calles que lo componen.

7.2.1 Selección de la escala

Una misma área puede cartografiarse a diferentes escalas. La selección de la escala depende de la finalidad del mapa y determina el grado de detalle de los elementos representados, así como su resolución y precisión.

Grado de detalle del mapa

El nivel de detalle de un mapa hace referencia a la cantidad de información geográfica que se representa en él. El proceso de simplificación de las entidades en los mapas de escala pequeña se conoce como generalización, de manera que entidades como ríos y carreteras se representan para líneas simples. Cuando se generalizan entidades, se reduce el nivel de detalle para evitar aglomeraciones en el mapa, aunque se mantienen la forma genérica y la posición.

Resolución del mapa

Se define como la medida de la entidad más pequeña que se puede representar en la superficie. Puede considerarse como la capacidad para distinguir entre objetos separados próximos entre sí.

Precisión del mapa

La precisión depende los datos geográficos utilizados para generar el mapa, con qué precisión se han transferido los datos originales al mapa.

Cuanto más pequeña es la escala de un mapa, una unidad de distancia representa una distancia más grande sobre el terreno. De esta manera, si una de las entidades

mostradas en un mapa de escala muy pequeña se aleja tan sólo un milímetro de su posición real, la imprecisión de la medida sobre el terreno es enorme.

7.3 Sistemas de coordenadas

Un sistema de coordenadas es una creación artificial para permitir la definición analítica de la posición de un objeto o fenómeno. Se trata de un conjunto de valores y puntos que permiten definir unívocamente la posición de cualquier punto de un espacio n-dimensional. Desde un punto de vista puramente matemático, todos los sistemas de coordenadas son admisibles y el único motivo para seleccionar uno u otro es la conveniencia que permite cada uno de ellos.

Los sistemas de coordenadas más utilizados para representar la superficie de la Tierra son el sistema de coordenadas geográficas, el sistema de coordenadas cartesiano y el sistema de coordenadas proyectadas. Para el proyecto se extraen los datos de OpenStreetMap, que utiliza el sistema de coordenadas geográfico. Es el que se expone a continuación.

7.3.1 Coordenadas geográficas

El sistema de coordenadas geográfico utiliza una superficie esférica tridimensional para definir las localizaciones sobre la superficie terrestre. En otras palabras, se usa para definir puntos sobre una superficie esférica. En este sistema, cualquier punto sobre la superficie terrestre se determina con dos ángulos medidos desde el centro de la Tierra, conocidos como latitud y longitud.

La latitud mide el ángulo entre cualquier punto y el ecuador. Las líneas de latitud se llaman paralelos al ecuador en la superficie de la Tierra. El concepto de latitud tiene las siguientes características.

- ✔ La latitud se expresa en grados sexagesimales.
- ✔ Todos los puntos ubicados sobre el mismo paralelo tienen la misma latitud.
- ✔ Aquellos que se encuentran al norte del Ecuador reciben la denominación Norte (N) y los demás denominación Sur (S).
- ✔ Se mide de 0º a 90º.

- ✔ Al Ecuador le corresponde la latitud de 0º.
- ✔ Los polos Norte y Sur tienen latitud 90ºN y 90ºS respectivamente.

La longitud mide el ángulo a lo largo del ecuador desde cualquier punto de la Tierra. Se considera el suburbio de Greenwich como longitud 0. Las líneas de longitud son círculos máximos que pasan por los polos y se llaman meridianos. El concepto de longitud tiene las siguientes características.

- ✔ Todos los puntos ubicados sobre el mismo meridiano tienen la misma latitud.
- ✔ Aquellos que se encuentran al este del meridiano principal será positivo y al oeste será negativo.
- ✔ Se mide de -180º a +180º.
- ✔ Al meridiano de Greenwich le corresponde la longitud 0º.

Combinando estos dos ángulos, se puede expresar la posición de cualquier punto de la Tierra. El ecuador es un elemento importante de este sistema de coordenadas, pues representa el cero de los ángulos de latitud y el punto medio entre los polos. Es el plano fundamental del sistema de coordenadas geográficas.

7.3.2 Sistema de coordenadas UTM

El Sistema de Coordenadas Universal Transversal de Mercator (UTM) es un sistema de coordenadas basado en la proyección cartográfica transversa de Mercator, que se construye como la proyección de Mercator normal, pero en vez de hacerlo por la tangente al Ecuador, se la hace tangente a un meridiano. A diferencia del sistema de coordenadas geográficas, las magnitudes en el sistema UTM se expresan en metros únicamente al nivel del mar que es la base de la proyección del elipsoide de referencia.

La proyección de Mercator es una proyección cilíndrica y por tanto, resulta de la proyección de una esfera (la Tierra) en un cilindro tangente a un meridiano central, en particular al ecuador de la Tierra. Cualquier proyección transversal de Mercator generada a partir de un meridiano como línea de tangencia es útil tan sólo cerca del meridiano seleccionado. El sistema UTM permite cartografiar no sólo las zonas

cercanas al ecuador o las zonas cercanas a un único meridiano de tangencia, sino cualquier zona de la Tierra. Para poder llegar a esto, se definen sesenta proyecciones estándar diferentes, de manera que cada una es una proyección transversal Mercator con un meridiano diferente como línea de tangencia. Cada huso tiene seis grados de amplitud y están numerador de oeste a este. Cada huso tiene su propio meridiano central y se divide por el ecuador en dos mitades: norte y sud. Se definen como origen de coordenadas de cada huso la intersección entre su meridiano central y el ecuador.

Para cartografiar cualquier punto sobre la Tierra en el sistema UTM, se selecciona la línea central del huso UTM más cercano al punto y se utiliza la proyección cilíndrica de dicho huso. Para eliminar coordenadas negativas, el sistema de coordenadas modifica el valor de las coordenadas en el origen con los valores de *falso este* y *falso norte*[6]. El valor asignado al meridiano central es el falso este y el valor asignado al ecuador es el falso norte.

En coordenadas UTM una posición se define por tres elementos: el huso en el que se encuentra, las coordenadas E (eje horizontal) y N (eje vertical), y el hemisferio en el que se encuentra. Para indicar nuestra posición, es necesario conocer el huso en el que estamos.

7.4 Proyecciones

Para representar la totalidad de la superficie terrestre sin ningún tipo de distorsión, un mapa debe tener una superficie esférica como la de un globo terráqueo. Un mapa plano no puede representar con exactitud la superficie redondeada de la Tierra, excepto en áreas muy pequeñas en las que la curvatura es despreciable. Para mostrar grandes porciones de la superficie, la superficie esférica de la Tierra debe transformarse en una superficie plana. El sistema de transformación es lo que se conoce como proyección cartográfica. Cuando una superficie esférica se transfiere a un plano modifica su geografía y la distorsiona, pero existen muchas transformaciones que mantienen una o varias de las propiedades geométricas del globo.

6 Falso este y falso norte son dos de los parámetros utilizados para definir una proyección. El conjunto de los parámetros se define en el apartado de proyecciones

Dependiendo de la extensión y ubicación de la zona a representar, el cartógrafo elegirá un tipo de proyección teniendo en cuenta las características geométricas que cada uno de ellos conserva y las que no, así como los efectos que su uso tendrá en la representación de los ángulos, áreas, distancias y direcciones de la superficie a cartografiar.

No hay ninguna proyección que no tenga algún error de deformación, llamado distorsión. Como es imposible conservar todas las propiedades a la vez, hay que decidir qué tipo de mapa se pretende realizar, ya que cada proyección puede conserva alguna de sus propiedades geométricas. A pesar de todos los problemas relativos a la distorsión, todas las proyecciones mantienen una característica importante: la exactitud del posicionamiento.

La clasificación de las proyecciones es compleja, pues hay diferentes criterios. A continuación se expone un listado.

Clasificación según las propiedades geométricas

- ✔ Proyecciones conformes
- ✔ Proyecciones equivalentes o de igual área
- ✔ Proyecciones equidistantes
- ✔ Proyecciones acimutales, cenitales o de dirección verdadera
- ✔ Proyecciones de compromiso

Clasificación según la superficie de la que derivan

- ✔ Proyecciones cónicas
- ✔ Proyecciones cilíndricas
- ✔ Proyecciones planas o acimutales

La gran diversificación de proyecciones se debe al hecho que se definen de manera que minimicen la distorsión dentro de un área determinada. Elegir una proyección centrada en el lugar que se pretende cartografiar nos asegura la mínima distorsión para aquella zona. Por otro lado, cada proyección tiene un conjunto de parámetros que la definen. Se expone un listado de los parámetros más comunes.

- ✔ *Falso este y falso norte*, valor lineal arbitrario que se aplica al origen de las coordenadas X,Y respectivamente, para asegurar que todos los valores en X,Y sea positivos.
- ✔ *Meridiano central o longitud origen*, define el origen de coordenadas X.
- ✔ *Paralelo central o latitud origen*, define el origen de coordenadas Y.
- ✔ *Paralelo estándar 1 y paralelo estándar 2*, definen las líneas de latitud donde la escala tiene valor 1,0.
- ✔ *Factor de escala*, valor sin unidades asociadas que se aplica al punto o línea central de una proyección. El factor de escala recude la distorsión total de la proyección en el área de interés.

La selección de una proyección u otra depende del propósito para el cual se crea un mapa. Para mapas de carreteras nos interesa que se mantengan las distancias (equidistantes) y en los mapas temáticos es importante que se conserve la medida (equivalentes) y la forma (conformes) de las regiones cartografiadas.

Otros aspectos a tener en cuenta para seleccionar la mejor proyección son la extensión y la localización del área. En referencia a la extensión, cuanto más grande sea el área, más importancia tiene la curvatura de la Tierra y, por tanto, más grande es la distorsión de ciertas propiedades. En relación a la localización, para latitudes bajas (regiones ecuatoriales y tropicales) se utilizan proyecciones cilíndricas; para latitudes medianas se usan proyecciones cónicas, mientras que para regiones polares se usan proyecciones planas. Implícito en esta regla está el hecho de que estas zonas se proyectan en las áreas en que cada proyección tiene la menor distorsión.

- ✔ Las proyecciones cilíndricas no tienen distorsión en el ecuador. Ésta se incrementa a medida que nos acercamos a los polos.
- ✔ Las proyecciones cónicas no tienen distorsión a lo largo de algún paralelo entre el ecuador y el polo. La distorsión aumenta a medida que nos alejamos de este paralelo estándar.
- ✔ Las proyecciones acimutales tan sólo no tienen distorsión en su punto central. Las mayores distorsiones se dan en los extremos del mapa.

8 OpenStreetMap

El punto inicial del proyecto consiste en obtener datos cartográficos de Palma de Mallorca en un formato tal que facilite su posterior tratamiento. Para que esto sea posible, utilizamos OpenStreetMap, que contiene herramientas suficientes para poder obtener dicha información. En este apartado se explica en qué consiste, así como el formato de datos que implementa y el proceso mediante el cual se obtienen estos datos.

8.1 Descripción

OpenStreetMap es un proyecto dirigido expresamente a crear y ofrecer datos geográficos libres, como por ejemplo planos de calles. Básicamente, se trata de la creación de mapas libres y editables. Normalmente, los usuarios suelen subir sus coordenadas desde el GPS para crear localizaciones nuevas o para corregir datos vectoriales ya existentes.

El proyecto comenzó debido a que muchos mapas tienen en realidad restricciones legales o técnicas para su uso, lo cual evita que cualquier persona los pueda utilizar a su libre elección de forma creativa o productiva. Las licencias de uso no permiten corregir errores, añadir nuevos datos o emplear esos mapas de determinados modos sin pagar por ellos.

OpenStreetMap se fundó en el año 2006 y se inscribe en el registro de Inglaterra y Gales de esta manera.

La fundación OpenStreetMap es una organización internacional no lucrativa dedicada a fomentar el crecimiento, desarrollo y distribución de datos geoespaciales libres y a proveer datos geoespaciales a cualquiera para usar y compartir.

Se trata de un proyecto con bastante impacto a nivel mundial. Cada día se añaden 25.000 km nuevos de carreteras y caminos, con un total de casi 34.000.000 km viales. A todo esto hay que añadirle otro tipo de datos, como puntos de interés,

instituciones, etc. El tamaño de la base de datos *planet.osm* ha superado ya los 160 GB, con un incremento diario de aproximadamente 10 MB.

8.2 Formato de datos

OpenStreetMap utiliza una estructura de datos topológica. Los datos se almacenan en el datum WGS84 *lat/lon* de proyección de Mercator. Los elementos básicos de la cartografía OSM son los listados a continuación.

- ✔ Nodos, son puntos que recogen una posición geográfica dada.
- ✔ Vías (*ways*), son una lista de nodos que representa una polilínea o polígono.
- ✔ Relaciones, son grupos de nodos, caminos y otras relaciones a las que se pueden asignar determinadas propiedades.
- ✔ Etiquetas *(tags),* se pueden asignar a nodos, caminos o relaciones y constan de una clave (*key*) y de un valor (ej. *highway=trunk*).

La forma más común de capturar los datos espaciales es mediante el uso de dispositivos GPS. Estos datos se pueden cargar en el servidor de OpenStreetMap. Para convertir los datos en un formato adecuado para el proyecto se utilizan programas de conversión. Todos los datos son recopilados en formato OSM bajo el sistema de coordenadas cartográficas mundial WGS84 lat/lon (proyección de Mercator, sistema de coordenadas UTM)[7].

El formato OSM es el formato de archivo XML propio de OpenStreetMap y se distribuye bajo la licencia Creative Commons 2.0.

Un dato importante es que el formato de datos de OpenStreetMap no cumple los esquemas y software del OGC[8], pues la manera de trabajar de ambas instituciones son diferentes. OpenStreetMap argumenta que OGC centra la atención en los datos asociados a callejeros y el mantenimiento de una wiki enfocada a la edición donde todos los cambios que se registran puedan ser revertidos. Según OpenStreetMap, estas herramientas no son compatibles con su forma de trabajar. Sin embargo, no se trata de una puerta cerrada, sino que están abiertos a cualquier ayuda de

7 Ver apartado 7.3.2 Sistema de coordenadas UTM

8 Ver apartado 6.3.1 Open Geospatial Consortium (OGC)

colaboradores para escoger las herramientas y normas OGC necesarias para conseguir una integración eficaz con otros sistemas ya existentes. Sin embargo, parece ser que se trata de un proyecto un tanto a largo plazo.

8.3 Descarga de datos

Para obtener datos directamente en formatos SIG, desde OpenStreetMap es posible descargar los datos en formato OpenStreetMap XML Data (.osm). Desde la pestaña Export, se selecciona la zona a exportar así como el formato de salida (ver figura 1).

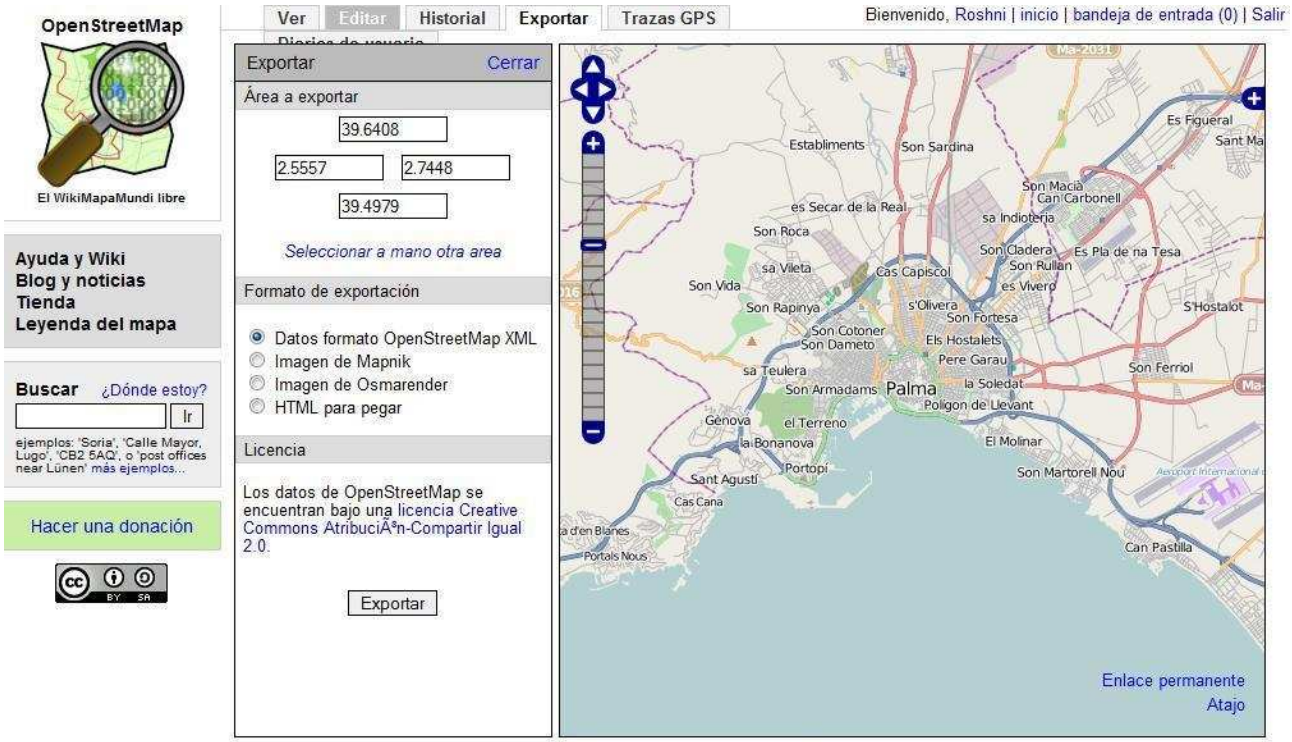


Figura 3. Exportación de datos en OpenStreetMap

Para el PFC, se toman como referencia la ciudad de Palma de Mallorca, y para ellos las coordenadas del área a exportar son las siguientes.

Minimum Latitude: 39.4979
Minimum Longitude: 2.5557
Maximum Latitude: 39.6408
Maximum Longitude: 2.7448

Se obtiene un archivo con extensión .osm en formato OpenStreetMap XML Data con la información descargada.

9 Base de Datos Geográfica

Los datos obtenidos en OpenStreetMap deben almacenarse y procesarse en una base de datos con soporte geográfico. Para ello se utiliza PostgreSQL junto con PostGIS para crear la base de datos espacial necesaria para importar los datos.

Para empezar, este apartado introduce el concepto de base de datos geográfica y expone su creación en PostGIS. Seguidamente, se importan a la base de datos los datos obtenidos de OpenStreetMap en el apartado anterior. Por último, se explica cuál es la estructura de base de datos creada para poder almacenar la información adicional de los puntos de interés que se van a tomar como referencia en el proyecto.

9.1 Descripción

Los datos geográficos son la representación digital de entidades, objetos o fenómenos que suceden sobre la superficie de la Tierra. Un elemento geográfico es el conjunto de la geometría que lo representa sobre el territorio y los atributos que describen sus propiedades.

Todos los datos geográficos tienen en común una localización sobre la Tierra que puede ser capturada y almacenada y unos atributos y propiedades que caracterizan estos datos. Los datos geográficos se pueden guardar en multitud de formatos y modelos. Escoger un formato u otro es tarea de los diseñadores y analistas y muchas veces suele ser la clave del éxito de un proyecto SIG.

Un modelo de datos geográfico es una abstracción del mundo real que emplea un conjunto de objetos dato, para soportar el despliegue de mapas, consultas, edición y análisis. Los datos geográficos presentan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, redes, superficies, ubicaciones e imágenes, a los cuales se les asignan sus respectivos atributos que los definen y describen.

Existen dos maneras de modelar los datos: datos vectoriales y datos raster. El almacenaje de datos vectoriales contiene los siguientes elementos.

- ✔ **Objetos puntuales**. Objetos que se encuentran en un punto concreto del territorio.
- ✔ **Objetos lineales**. Objetos que se distribuyen linealmente sobre el territorio.
- ✔ **Objetos área**. Objetos que ocupan un área de territorio.

9.2 PostgreSQL y PostGIS

Para este PFC se utiliza el sistema de gestión de base de datos PostgreSQL[9]. Se trata de un SGBD libre y de código abierto.

Se ha instalado la extensión PostGIS que convierte a la base de datos PostgreSQL en una base de datos espacial. Se trata de un módulo que añade soporte de objetos geográficos. PostGIS ha sido certificado por el Open Geospatial Consortium[10], lo que garantiza la interoperabilidad con otros sistemas. PostGIS almacena la información geográfica en una columna del tipo *GEOMETRY*. Con PostGIS es posible usar todos los objetos que aparecen en la especificación GIS como puntos, líneas, polígonos, multilineas, multipuntos y colecciones geométricas. La especificación OpenGIS requiere que los objetos incluyan el identificador del sistema de referencia espacial (SRID). Se requiere el SRID al insertar un objeto espacial en la base de datos.

9.2.1 Usar el estándar OpenGIS

Para crear la base de datos, se utiliza el siguiente comando.

```
CREATE DATABASE gis
WITH ENCODING='UTF8'
OWNER=postgres
TEMPLATE=template_postgis
CONNECTION LIMIT=-1
TABLESPACE=pg_default;
```

9 http://www.postgresql.org/

10 Ver apartado 6.3.1 Open Geospatial Consortium (OGC)

Para definir la base de datos de forma espacial, lo más sencillo es usar una plantilla. La nueva base de datos se crea a partir de una plantilla que coge como base para su definición. Dicha plantilla consiste en otra base de datos, en este caso llamada template_postgis y que se instala automáticamente con postgis, que contiene las funcionalidades y tablas necesarias para poder trabajar con geometrías, siempre de acuerdo con la especificación OpenGIS. De esta manera, obtenemos una base de datos espacial.

La especificación OpenGIS define dos tablas de metadatos.

Spatial_ref_sys contiene un identificador numérico y una descripción textual del sistema de coordenadas espacial de la base de datos. La tabla se define de la siguiente manera.

SPATIAL_REF_SYS			
#	Nombre	Tipo	Comentario
1	srid	Int4	Valor entero que identifica el sistema de referencia espacial.
2	auth_name	varchar(256)	El nombre del estándar para el sistema de referencia.
3	auth_srid	Int4	El identificador según AUTH_NAME.
4	srtext	varchar(2048)	Representación en texto de la geometría
5	proj4text	varchar(2048)	Cadena con definición de las coordenadas proj4 (librería de PostGIS para transformar coordenadas)

Geometry_columns define los tipos de geometría contenida en la base de datos. La tabla se define de la siguiente manera.

GEOMETRY_COLUMNS			
#	Nombre	Tipo	Comentario
1	f_table_catalog	varchar(256)	
2	f_table_schema	varchar(256)	Esquema propietario de la tabla
3	f_table_name	varchar(256)	Nombre de la tabla que contiene la geometría

4	f_geometry_column	varchar(256)	Nombre de la columna geométrica
5	coord_dimension	int4	Dimensión espacial de la columna (2D o 3D)
6	srid	int4	FK referencia Spatial_Ref_Sys
7	type	varchar(30)	Tipo de objeto espacial. POINT, LINESTRING, POLYGON, MULTIPOINT, GEOMETRYCOLLECTION, GEOMETRY

En este momento la base de datos contiene tan sólo estas dos tablas.

9.2.2 Importación de OpenStreetMap XML Data

El siguiente paso ahora es importar los datos obtenidos en OpenStreetMap dentro de la base de datos PostGIS creada. Para ello se utiliza **osm2pgrouting**, herramienta de conversión que transforma los datos OSM del fichero OpenStreetMap en un formato compatible para realizar la carga en la base de datos PostgreSQL. Para ello, el comando utilizado es el siguiente.

sudo osm2pgrouting/osm2pgrouting -file mapa.osm -conf osm2pgrouting/mapconfig.xml -dbname gis -user postgres

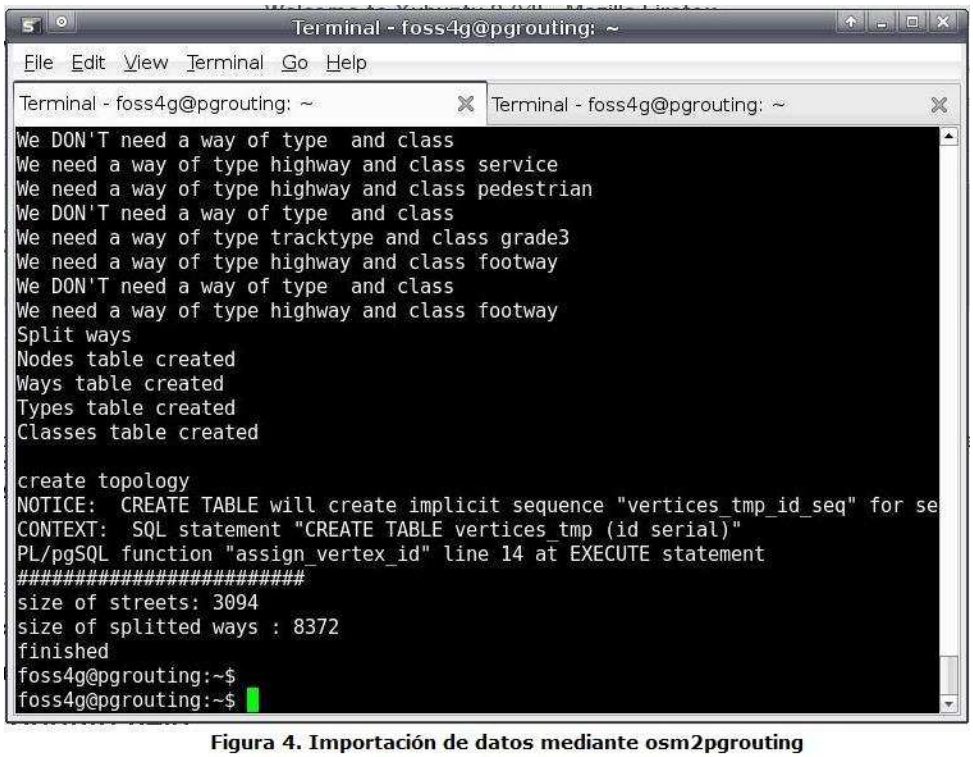


Figura 4. Importación de datos mediante osm2pgrouting

Por el resultado del log que aparece de la figura 2, se observa cuál ha sido todo el proceso. El resultado final es que se crean las tablas en la base de datos donde se va a almacenar toda la información y se rellenan con los datos pertinentes proporcionados por el fichero OSM. Las tablas creadas son las siguientes.

- ✔ classes
- ✔ nodes
- ✔ types
- ✔ vertices_tmp
- ✔ ways

Seguidamente, osm2pgrouting realiza las conversiones necesarias e importa los datos en las tablas creadas. El resultado es que toda la información seleccionada de OpenStreetMap se encuentra volcada en una base de datos geográfica.

9.3 Modelo conceptual

El modelo conceptual es la representación de los conceptos u objetos más significativos del dominio del problema y de sus relaciones en forma de clases de objetos, asociaciones y atributos. Para realizar la representación del problema, el modelo hace uso del lenguaje UML como lenguaje estándar de modelado.

Dada la naturaleza del PFC, el modelo no se muestra muy complejo. Al fin y al cabo, la aplicación a crear consiste básicamente en calcular la ruta más corta entre dos puntos. La aplicación no necesita de ninguna estructura adicional para guardar ningún tipo de información. En general, podemos decir que el grueso del PFC se encuentra en los algoritmos de cálculos de ruta más que en el modelo, y por tanto nuestra base de datos tan sólo será de consulta.

Para empezar, se parte de la tabla espacial creada por la herramienta osm2pgrouting durante el proceso de importación *(ways).* Dicha tabla incluye la información geográfica necesaria para situarse en una posición de la ciudad. En cuanto a los datos, gran parte de los que se usarán en la aplicación son gestionados por PostGIS, por lo que no se analizará el modelo de datos para estas entidades. El análisis se centrará

por tanto en el modelo de datos derivado de la nueva información con la que se necesita: los puntos de interés. Básicamente, se trata de almacenar los puntos de interés que se utilizarán en la muestra del proyecto. Para ello crearemos una tabla llamada **POI**, con la información mínima necesaria para poder localizar un punto geográfico sobre el mapa.

Además de estas tablas, osm2pgrouting crea otras para el manejo y la tipificación de la información espacial contenida en la tabla *ways*. Concretamente, las tablas *classes* y *types* contienen la configuración de tipo de vías de los que puede disponer el mapa. A modo ilustrativo, se expone el contenido establecido por osm2pgrouting para la tabla *types*.

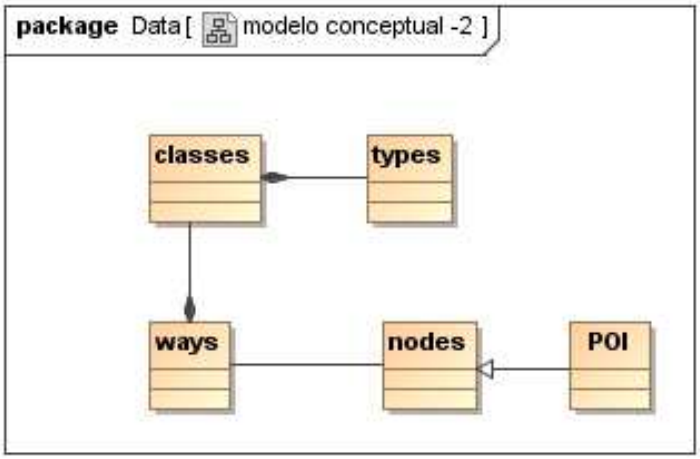
Types	
Id	name
1	highway
2	cycleway
3	tracktype
4	junction

La información contenida en *classes* contiene una subtipificación de *types*, y consiste en la información que se ve reflejada en *ways*. Es decir, para cada calle definida en *way*, se establece la clase de vía a la que pertenece.

A continuación se describe el modelo conceptual de nuestro PFC.

9.4 Modelo de datos

El modelo de datos describe la estructura que utiliza el sistema de software para almacenar los datos persistentes. Este modelo se diseña a partir del modelo conceptual del apartado anterior. Como resultado se obtienen los modelos lógicos y físicos de la base de datos.



9.4.1 Diseño lógico de datos

El diseño lógico se obtiene del modelo conceptual de datos y consiste en la descripción de la estructura de la base de datos en la que consiste la aplicación. La única clase de nuestro modelo es POI, cuya estructura es la siguiente.

POI			
#	Nombre	Tipo	Comentario
1	Name	Char(50)	Nombre del POI
2	Description	Char(255)	Texto descriptivo para el POI
3	X	Double precision	Coordenada X del POI
4	Y	Double precision	Coordenada Y del POI

X,Y almacenan las coordenadas de longitud y latitud de cada punto de interés a mostrar en la aplicación.

Se ha decidido crear esta nueva tabla para los puntos de interés para poder manipular las descripciones a mostrar en la aplicación, sin tener que modificar la información original de OpenStreetMap. De todas maneras, el objetivo es que se trate de contenido estático, es decir, PalmaRouting no va a poder modificar el contenido de la tabla, sino que será el administrador del proyecto quien se encargue de dicha tarea. La información de la tabla es la siguiente.

POI (Data)				
Id	Name	Description	X	Y
1	Catedral	La Seu (catedral)	2.6479107	39.5682409
2	Castillo de Bellver	Único castillo circular de toda Europa	2.6251714	39.5670644
3	Paseo de Portitxol	Paseo de Portitxol	2.6632209	39.5624125
4	Placa de la Reina	Placa de la Reina	2.6463902	39.5691991
5	Iglesia de Sant Francesc	Basílica y colegio e instituto de Sant Francesc	2.6527746	39.5684605
6	Pueblo Espanol	Pueblo Español	2.6259475	39.5731245
7	Parc de la Mar	Parc de la mar	2.6400166	39.5693707
8	Marivent	Residencia de vacaciones de los Reyes	2.6128421	39.5519959
9	Na Burguesa	mirador de Na Burguesa	2.6012414	39.5654738
10	Plaza Mayor	Placa Major	2.6518487	39.5715617

En la aplicación a crear mediante OpenLayers, se visualizarán los nombres estandarizados en la tabla POI, así como una descripción adicional del lugar.

9.4.2 Diseño físico de datos

El diseño físico se obtiene del diseño lógico y consiste en la descripción de la implementación de la base de datos, describiendo las estructuras de almacenamiento y los métodos de acceso a esos datos.

Para PalmaRouting, la estructura de tablas se ha creado automáticamente mediante osm2pgrouting, y si bien no describiremos el modelo de datos, sí se hará hincapié en la tabla ways, que al final y al cabo contiene la información de geometría necesaria para renderizar el mapa. La estructura de la tabla es la siguiente.

Ways			
#	Nombre	Tipo	Comentario
1	Gid	Integer	Id de cada registro
2	Class_id	Integer	Tipo de vía – FK a classes
3	Length	Double precision	Longitud del tramo

4	Name	Char(200)	Nombre de la vía, si aplica
5	X1	Double precision	Coordenada X del punto de inicio
6	Y1	Double precision	Coordenada Y del punto de inicio
7	X2	Double precision	Coordenada X del punto de fin
8	Y2	Double precision	Coordenada Y del punto de fin
9	the_geom	Geometry	Geometría de la vía
10	Source	Integer	Id del nodo de inicio de la vía
11	Target	Integer	Id del nodo de fin de la vía

Además de la información contenida en el campo de geometría the_geom, los campos source y target, así como las posiciones x1, y1, x2, y2 nos permiten realizar cálculos de grafos para obtener rutas a seguir entre dos puntos del mapa. Los campos source y target definen un id para cada nodo (definidos en la tabla vertices_tmp), y de esta manera es posible unir dos vías que empiezan y terminan en el mismo nodo.

```
CREATE TABLE ways
(  gid integer NOT NULL,
  class_id integer,
  length double precision,
  "name" character(200),
  x1 double precision,
  y1 double precision,
  x2 double precision,
  y2 double precision,
  reverse_cost double precision,
  "rule" text,
  to_cost double precision,
  the_geom geometry,
  source integer,
  target integer,
  CONSTRAINT ways_pkey PRIMARY KEY (gid),
  CONSTRAINT enforce_dims_the_geom CHECK (ndims(the_geom) = 2),
  CONSTRAINT enforce_geotype_the_geom CHECK (geometrytype(the_geom) =
'MULTILINESTRING'::text OR the_geom IS NULL),
```

```
  CONSTRAINT enforce_srid_the_geom CHECK (srid(the_geom) = 4326)
)
```

Por otro lado, ya se ha establecido en el punto anterior la finalidad de la tabla POI, que contiene la información relevante para cada punto de interés a referenciar en el proyecto de PalmaRouting.

```
create table POI
(id int4 not null, name varchar(50), description varchar(255),
x double precision, y double precision);
```

10 Cálculo de rutas

Hasta el apartado anterior se ha conseguido almacenar y procesar los datos de localización en una base de datos geográfica. El siguiente paso es tratar dicha información para obtener información de enrutamiento.

10.1 Descripción

Se define como cálculo de rutas o enrutamiento[11] a la función de buscar un camino entre todos los posibles en una red cuya topología posee una gran conectividad. Normalmente, la finalidad de estos algoritmos es hallar la ruta más corta entre dos nodos, por lo que debe darse énfasis en este término, a su vez simple pero también complejo.

Entendemos por mejor ruta aquella que cumple las siguientes condiciones.

- ✔ Presenta el menor retardo medio de tránsito.
- ✔ Consigue mantener acotado el retardo entre pares de nodos de la red.
- ✔ Consigue ofrecer altas cadencias efectivas independientemente del retardo medio del tránsito.
- ✔ Permite ofrecer el menor costo.

El criterio más sencillo es elegir el camino más corto, es decir, la ruta que pasa por el menos número de nodos. Una generalización de este criterio es el de "coste mínimo".

Un SIG destinado al cálculo de rutas óptimas es capaz de determinar el camino más corto entre dos puntos teniendo en cuenta impedancias como direcciones de circulación, giros prohibidos o evitando determinadas áreas impracticables. Un SIG para la gerencia de una red de abastecimiento de aguas sería capaz de determinar, por ejemplo, a cuántos abonados afectaría el corte del servicio en un determinado punto de la red.

11 Definición obtenida de la Wikipedia: http://es.wikipedia.org/wiki/Encaminamiento

10.2 PgRouting

Una herramienta utilizada en este PFC es pgRouting[12], cuyo objetivo principal es proveer a PostgreSQL y PostGIS con funcionalidades de enrutamiento. Es decir, permite realizar cálculos de rutas entre dos puntos utilizando la información almacenada en la base de datos.

PgRouting consiste en una herramienta que utiliza librerías de cálculo de grafos. Es una extensión que se renueva con cierta asiduidad, y lo más interesante es que propone formas de publicación de cartografía con el servidor de mapas MapServer, OpenLayers y PHP. Estas tres son las herramientas que componen la parte práctica o técnica del PFC.

La finalidad de pgRouting es calcular el camino a seguir desde un punto del mapa hasta otro definido como destino. Más concretamente y aplicado al caso del PFC, diremos que calcularemos el camino a seguir desde nuestra posición actual hasta un punto de interés a visitar en la ciudad de Palma de Mallorca.

El primer paso es disponer de un conjunto de datos adecuado sobre los cuales trabajar. Es decir, pgRouting necesita datos geográficos para crear una red de enrutamiento. A su vez, para crear el enrutamiento, los datos necesitan una topología, es decir, una cadena de comunicación usada por los nodos de la red para comunicarse[13]. Esto se consigue mediante la información contenida en la columna de base de datos *ways.the_geom*[14]. Este campo contiene la información geométrica necesaria para crear la topología de enrutamiento deseada. Por otro lado, los algoritmos de enrutamiento requieren información de origen y destino *(source* y *target* en la tabla *ways*) para cada línea del mapa para poder realizar una búsqueda del camino más corto. Generar esta información sobre las líneas implica generar una topología sobre la red.

12 http://pgrouting.postlbs.org/

13 Definición obtenida de la Wikipedia: http://es.wikipedia.org/wiki/Topología_de_red

14 Ver apartado 9.4.2 Diseño físico de datos

La herramienta de importación de datos utilizada en el PFC, osm2pgrouting[15], realiza los cálculos, conversiones y transformaciones necesarias, de manera que la base de datos creada mediante dicha herramienta contiene una topología válida que se considera buena para realizar cálculos de obtención de rutas. Sin embargo, no está de más comprobar la calidad de los datos para asegurar que los datos son correctos para realizar un buen enrutamiento.

Un buen ensayo es comprobar los segmentos correctos e identificar los segmentos '*dead end*', es decir, los que al fin y al cabo equivalen a un callejón sin salida, ya sea porque realmente lo son, o bien porque los segmentos se encuentran mal unidos. En una topología de enrutamiento, lo ideal es que los segmentos *dead end* sean mínimos. Para comprobarlo con nuestro conjunto de datos, este análisis se ha implementado como capa adicional del mapa.

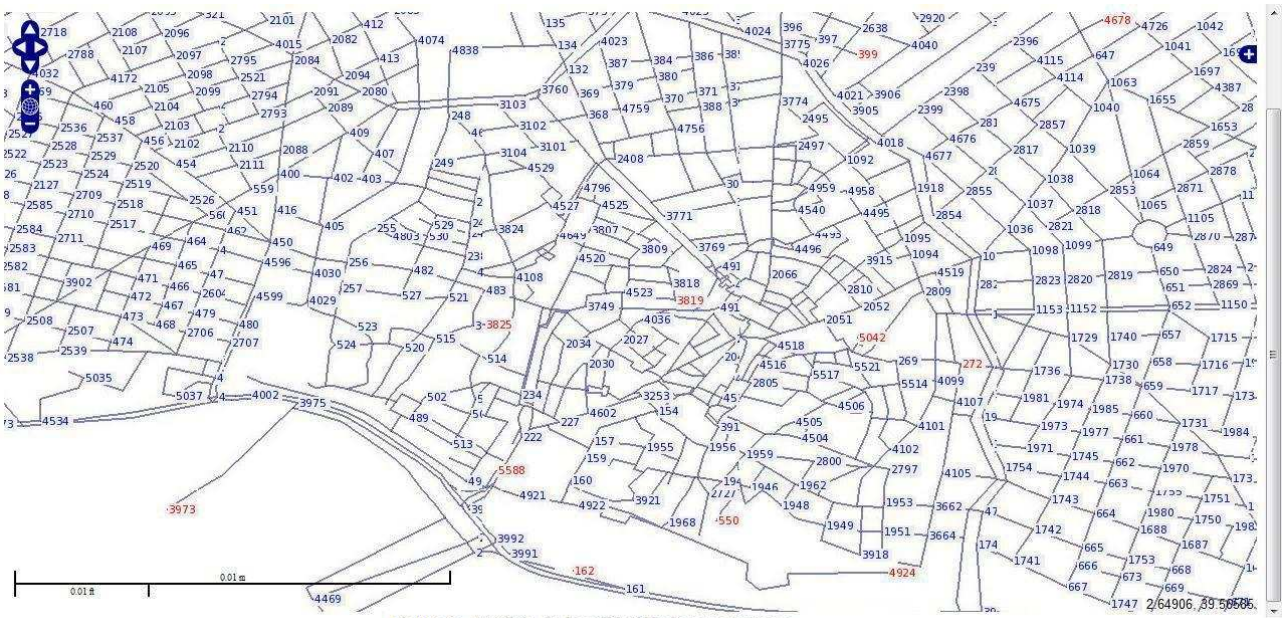


Figura 6 . Análisis de la calidad de los segmentos

En la figura 3, podemos observa el resultado del análisis de cada uno de los nodos del mapa. Los nodos marcados en rojo son aquellos unidos a un sólo segmento, es decir, la única manera de llegar a este nodo es a través de este único segmento. Este nodo equivale a un callejón sin salida, y el segmento que llega hasta él podría considerarse como un segmento muerto. En la figura 3 observamos que el número de nodos *dead-end* es muy bajo, por lo que daremos el conjunto de datos por válido.

15 Ver apartado 9.2.2 Importación de OpenStreetMap XML Data

10.3 Algoritmos

Existen varios algoritmos para obtener las posibles rutas entre dos puntos de una topología. A continuación se describen los algoritmos proporcionados por pgRouting y las características de cada uno. De esta manera será posible obtener comparativas entre los resultados de los diferentes algoritmos.

10.3.1 Algoritmo de Dijkstra

El algoritmo *Shortest Path* de Dijkstra[16] hace honor a su creador, Edsger Dijkstra, quien diseñó el algoritmo en 1959. Este algoritmo, también conocido como algoritmo de caminos mínimos, tiene como finalidad la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo dirigido y con pesos en cada arista.

La lógica del algoritmo consiste en explorar todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices. Cuando se obtiene el camino más corto desde el vértice origen al resto de vértices que componen el grafo, el algoritmo se detiene.

A continuación se presentan las diferentes funcionalidades que ofrece pgRouting implementando el algoritmo de Dijkstra.

10.3.1.1 Shortest_path

PgRouting proporciona el algoritmo de Dijkstra en la forma de la función *shortest_path*, cuya especificación es la siguiente.

CREATE OR REPLACE FUNCTION shortest_path(
sql text,
source_id integer,
target_id integer,
directed boolean,
has_reverse_cost boolean

16 Definición obtenida de la Wikipedia: http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra

) RETURNS SETOF path_result;

Este algoritmo es aplicable de la siguiente manera[17].

*Select * from shortest_path(*
'select the_geom,gid as id,source,target,x1,y1,x2,y2,length as cost,reverse_cost
from ways',
405, 225, false, false);

10.3.1.2 Dijkstra_sp

Computacionalmente hablando, una función contenedora[18] es una función cuya finalidad principal es llamar a otra función, sin añadir prácticamente funcionalidad adicional alguna. Las funciones contenedoras se utilizan normalmente para adaptar la clase y obtener una interfaz diferente, o bien para capturar mensajes de error y sobreescribirlos.

En las funciones pgRouting, la razón por las que se han creado funciones contenedoras para el cálculo de rutas es reducir el tiempo de respuesta de pgRouting. Dicho tiempo de respuesta se divide en dos partes: la primera de carga de datos, la segunda de cálculo de resultados. El tiempo de la primera parte depende del tamaño o la cantidad de datos existentes. Para reducir este tiempo, el algoritmo debe reducirse la cantidad de datos a cargar. Para conseguirlo, se reducen los datos de la carga a la información contenida en un *bounding box*, o cuadro delimitador, tomando como referencia las posiciones de nodo inicio y fin cuyo camino se pretende calcular. De esta manera, en la primera fase del algoritmo se cargan menos datos, reduciendo así el tiempo total de ejecución del algoritmo. Es por eso que en pgRouting se concluye que las funciones contenedoras suelen ser siempre más rápidas.

La función *dijkstra_sp* es una función contenedora de *shortest_path*, tomando como cuadro delimitador los datos contenidos desde *source* hasta *target*. Esta función es, por su propia definición, más óptima que la función original *shortest_path*, aunque los resultados devueltos serán siempre los mismos.

17 El ejemplo a escenificar para todos los algoritmos de aquí en adelante es el cálculo de rutas desde el nodo 405 hasta el 225.

18 Definición obtenida de la Wikipedia: http://en.wikipedia.org/wiki/Wrapper_function

CREATE OR REPLACE FUNCTION dijkstra_sp(
geom_table character varying,
source integer,
target integer
) RETURNS SETOF geoms;

Ahora esta función debe utilizarse para retornar unos resultados que posteriormente deben tratarse para obtener la información completa. El ejemplo anterior se aplica de la siguiente manera.

```
select rt.gid, AsText(rt.the_geom) AS wkt,length(rt.the_geom) AS length,
    ways.gid, ways.source, ways.target from ways,
   (SELECT gid, the_geom FROM dijkstra_sp('ways',405,225)) as rt
 where ways.gid=rt.gid;
```

10.3.1.3 Dijkstra_sp_delta

La función dijkstra_sp_delta consiste en una versión delta de la función *dijkstra_sp*. Las funciones delta son análogas a sus predecesoras, con la peculiaridad de que aceptan un parámetro más, llamado precisamente delta, que indica el margen a añadir al *bounding box* utilizado por la función. En resumidas cuentas consiste en un margen de error sobre los datos, para no descartar en el proceso alguna vía que de otro modo formaría parte del camino más corto entre los dos puntos. Si no se estableciera un margen correcto, los dos algoritmos podrían retornar resultados diferentes.

La especificación de esta función es la siguiente.

CREATE OR REPLACE FUNCTION dijkstra_sp_delta(
geom_table character varying,
source integer,
target integer,
delta double precision
) RETURNS SETOF geoms;

Un ejemplo para este caso sería el siguiente.

```
select rt.id, rt.gid, AsText(rt.the_geom) AS wkt, length(rt.the_geom) AS length,
    ways.gid, ways.source, ways.target from ways,
   (SELECT id, gid, the_geom FROM dijkstra_sp_delta('ways',405,225,3000)) as rt
 where ways.gid=rt.gid;
```

10.3.2 Algoritmo A*

El algoritmo *Shortest Path A**[19] consiste básicamente en una extensión del algoritmo de Dijkstra usando heurística, con la ventaja de que consigue mejorar el rendimiento en cuanto a la velocidad de cálculo. El término de heurística en un sentido computacional consiste en una función matemática *h(n)* que sirve como estimación del coste del camino más económico de un nodo dado hasta el nodo objetivo. La búsqueda escogerá el nodo que tiene el valor más bajo en la función heurística. La función *h(n)* es admisible cuando no supera nunca los costes de encontrar el objetivo.

A continuación se presentan las diferentes funcionalidades que ofrece pgRouting para el algoritmo A*.

10.3.2.1 Shortest_path_astar

PgRouting proporciona el algoritmo A* en la forma de la función *shortest_path_astar*, cuya especificación es la siguiente.

```
CREATE OR REPLACE FUNCTION shortest_path_astar(
sql text,
source_id integer,
target_id integer,
directed boolean,
has_reverse_cost boolean
) RETURNS SETOF path_result;
```

Este algoritmo es aplicable de la siguiente manera.

```
select * from shortest_path_astar(
'select the_geom,gid as id,source,target,x1,y1,x2,y2,length as cost,reverse_cost
```

19 Definición obtenida de la Wikipedia: http://en.wikipedia.org/wiki/A*_search_algorithm

from ways',
405,225,false,false);

10.3.2.2 Astar_sp_delta

La función astar_sp_delta es la equivalente a la función *shortest_path_astar*, en su versión contenedora y con parámetro delta.

Su especificación es la siguiente.

```
CREATE OR REPLACE FUNCTION astar_sp_delta(
geom_table character varying,
source integer,
target integer,
delta double precision
) RETURNS SETOF geoms;
```

Un ejemplo para este algoritmo es el siguiente.

```
select rt.id, rt.gid, AsText(rt.the_geom) as wkt,length(rt.the_geom) as length,
    ways.gid, ways.source, ways.target from ways,
   (select id, gid, the_geom from astar_sp_delta('ways',405,225,3000)) as rt
 where ways.gid=rt.gid;
```

10.3.3 Algoritmo Shooting Star

El algoritmo *Shooting Star* calcula una ruta óptima entre dos aristas, a diferencia de los dos algoritmos anteriores que realizan los cálculos de vértice a vértice. La ruta calculada mediante este algoritmo devuelve como punto de partida el nodo de inicio de la línea *(source).*

La particularidad de este algoritmo es que permite añadir restricciones a nivel de vía. Hasta ahora se calculaba el coste de una vía *(cost)* en función de su longitud *(length),* sin que por ello consten otras dificultades en el camino, como por ejemplo, semáforos, restricciones de giros prohibidos, etc. *Shooting Star* pretende tener en cuenta todos

estos detalles para calcular las rutas teniendo en cuenta estos impedimentos presentes.

En la base de datos de PalmaRouting, no existe información alguna en los campos de ways.to_cost y ways.rule[20], por lo que el cálculo de ruta se está realizando sin impedimento alguno.

A continuación se presentan las diferentes funcionalidades que ofrece pgRouting para el algoritmo *Shooting Star*.

10.3.3.1 Shortest_path_shooting_star

PgRouting proporciona el algoritmo *Shooting Star* en la forma de la función *shortest_path_shooting_star*, cuya especificación es la siguiente.

```
CREATE OR REPLACE FUNCTION shortest_path_shooting_star(
sql text,
source_id integer,
target_id integer,
directed boolean,
has_reverse_cost boolean
) RETURNS SETOF path_result;
```

La función devuelve un conjunto de filas, cada una de las cuales corresponde a una línea cruzada, además de una fila adicional que contiene el vértice terminal. La información devuelta por cada fila es la siguiente.

- vertex_id: identificador del vértice origen de cada línea. Al final existe una adicional, que contiene el vértice identificador del vértice destino.
- edge_id: identificador de la arista cruzada
- cost: el coste asociado a la línea actual. El coste total de la ruta es la suma de todos los costes parciales.

Este algoritmo es aplicable de la siguiente manera.

20 Ver apartado 9.4.2 Diseño físico de datos

```
select * from shortest_path_shooting_star(
'select the_geom, gid as id,source,target,x1,y1,x2,y2,length as cost,rule,to_cost
from ways',
405,225,false,false);
```

10.3.3.2 Shootingstar_sp

La función *shootingstar_sp* es la equivalente a la función *shortest_path_shooting_star*, en su versión contenedora y con parámetro delta.

Su especificación es la siguiente.

```
CREATE OR REPLACE FUNCTION shootingstar_sp(
geom_table character varying,
source integer,
target integer,
delta double precision,
cost_field character varying,
directed boolean,
has_reverse_cost boolean
)  RETURNS SETOF geoms;
```

Un ejemplo para este algoritmo es el siguiente.

```
select rt.id, rt.gid, AsText(rt.the_geom) AS wkt, length(rt.the_geom) AS length,
      ways.gid, ways.source, ways.target from ways,
    (select id, gid, the_geom from shootingstar_sp('ways',405,225,3000, 'length',
false, false)) as rt
 where ways.gid=rt.gid;
```

10.4 Comparativas entre algoritmos

Ahora que ya se ha estudiado cada algoritmo y sus funciones por separado, pasamos a comparar su eficiencia para poder seleccionar uno de los tres.

El primer algoritmo que apareció es el de Dijkstra, cuyos resultados son buenos y devuelve valores bastante fiables. Más tarde apareció el algoritmo A*, que si bien devuelve siempre (siempre que se utilice un cuadro delimitador con parámetro delta suficiente) los mismos resultados que su antecesor, mejora considerablemente su eficiencia al utilizar una heurística.

Por otro lado, tenemos el algoritmo de *Shooting Star*, que cambia totalmente el concepto de búsqueda, ya que realiza el cálculo sobre las caras o aristas de la topología, y no sobre los vértices. Además, tiene en cuenta las restricciones e impedancias presentes en el sistema. En nuestro caso y como ya se ha comentado anteriormente, no disponemos de información de restricciones en la base de datos de muestra.

Si nos viéramos en la situación de tener que elegir un sólo algoritmo entre los existentes, el elegido es el A*. Ya hemos comentado anteriormente que *Shooting Star* queda descartado en nuestro caso particular debido a la falta de información. Ahora queda por elegir entre entre Dijkstra y A*. Recordamos que A* no sólo está basado en el algoritmo de Dijkstra, sino que constituye además una mejora. Por tanto, como conclusión final es que el algoritmo más apropiado para este proyecto es A*, utilizado además por muchos buscadores como *Google Maps*.

11 Publicación del mapa

Todo el trabajo realizado hasta ahora debe reflejarse en algún lugar. La finalidad de este proyecto es poder mostrar en un entorno web el mapa creado, enviar peticiones de cálculo de rutas, recibir los resultados y mostrarlos en el navegador. Para ello el primer paso es publicar el mapa de manera que el servidor local (localhost) tenga acceso a él y posteriormente lo pueda mostrar.

11.1 Descripción

La información almacenada en la base de datos debe tener una salida gráfica e interactiva. Los datos guardados representan en formato geométrico la información geográfica de Palma de Mallorca. Para dar un formato visual a la información, es necesario disponer de herramientas que lean contenido geográfico, lo interpreten correctamente y reflejen visualmente. Existen diversas herramientas para ello, como OpenJump, gvSIG o MapServer. GvSIG contiene además, una herramienta de publicación de mapa que crea un *mapfile*[21] o configuración de mapa, que contiene la información relativa al mapa propiamente dicho así como sus propiedades. Este fichero de configuración debe luego analizarse mediante OpenLayers para mostrar el mapa desde el navegador web.

11.2 MapServer

MapServer[22] consiste en un entorno de desarrollo libre y de código abierto para la creación de aplicaciones SIG en Internet con el fin de visualizar, consultar y analizar información geográfica a través de la red. Es decir, se pretende publicar información espacial y mapas interactivos al navegador.

MapServer fue inicialmente desarrollado por la Universidad de Minnesota en colaboración con la NASA. En la actualidad forma parte del proyecto OSGeo y está respaldado por numerosas organizaciones que fomentan su desarrollo y mantenimiento, así como sus mejoras.

21 Ver apartado 11.2.2 Mapfile

22 Definición obtenida de la Wikipedia: http://es.wikipedia.org/wiki/MapServer

Son varias las características que lo distinguen y que lo hacen destacar como una de las mejores herramientas para la publicación de mapas.

- ✔ Salida cartográfica avanzada
- ✔ Soporte a entornos de desarrollo y scripting
  - PHP, Python, Perl, Ruby, Java, .NET
- ✔ Soporte multi-plataforma
- ✔ Soporte a varios estándares OGC[23]
  - WMS, WFS, WMC, WCS, Filter Encoding, SLD, GML, SOS, OM
- ✔ Soporte a multitud de formatos raster y vectorial
  - ESRI, PostGIS, Oracle Spatial, MySQL
- ✔ Soporte a proyección de mapas

En su estado más básico y puro, MapServer es un programa CGI que permanece inactivo en el servidor web. Cuando recibe una petición, utiliza la información aportada en la URL y un Mapfile[24] para crear un imagen del mapa solicitado.

11.2.1 Arquitectura MapServer

A continuación se describe la arquitectura básica de una aplicación MapServer. Un MapServer sencillo está compuesto por los siguientes elementos.

- ✔ *Map File*

Fichero de configuración para la aplicación MapServer. Contiene información tal como el área del mapa, dónde encontrar el ejecutable del MapServer, así como las capas del mapa. Además, para cada una de las capas definidas, se especifica la fuente de datos, así como las proyecciones y su simbología. La extensión del fichero debe ser .map.

- ✔ Datos geográficos

MapFile puede utilizar varios tipos de fuentes de datos geográficos. Para el PFC utilizaremos la base de datos PostGIS.

23 Ver apartado 6.3.1 Open Geospatial Consortium (OGC)

24 Fichero de configuración de un mapa para publicación en servidor. Ver apartado 11.2.2 Mapfile.

- ✔ Páginas HTML

Es la interfaz entre el usuario y MapServer. Una aplicación MapServerCGI debe contener por lo menos dos páginas html: index.html, y template.html.

- ✔ CGI MapServer

Archivo ejecutable que retorna imágenes, datos, etc., tras recibir peticiones. Comúnmente suele conocerse como *mapserv.exe*.

- ✔ *HTTP Server*

Servidor HTTP en local para crear el entorno. Se utilizará Apache.

11.2.2 Mapfile

El fichero Mapfile es el núcleo de MapServer. Define las relaciones entre objetos, indica a MapServer dónde se encuentran los datos, así como la manera en la que deben dibujarse.

Todo *mapfile* debe seguir una estructura específica. Si bien muchos parámetros son opcionales, otros muchos no sólo son recomendables, sino obligatorios. La especificación de cada uno de estos campos se define en este documento oficial.
http://ms.gis.umn.edu:8081/ms_plone/docs/reference/mapfile

En el Anexo I:Mapfile[25] se incluye el *mapfile* creado para PalmaRouting.

Dos aspectos fundamentales a tener en cuenta en un *mapfile* son la proyección o proyecciones utilizadas y las capas de información creadas.

- Proyecciones

Es importante tener en cuenta la proyección en la que han sido creados los datos geográficos. Para ello es posible consultar la información contenida en las tablas de metadatos[26]. Mediante la tabla *geometry_columns* podemos conocer cuál es la

25 Ver apartado 13 Anexo I:Mapfile

26 Ver información de metadatos en apartado 9.2.1 Usar estándar OpenGIS

proyección actual de los datos. En este caso se trata del srid 4326, cuya proyección tiene la siguiente definición.

"+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs "

Los datos geográficos deben mostrarse en un formato tal que sean visible por una capa web. En este caso hemos tenido suerte, ya que la proyección 4326 se ajusta perfectamente a una salida en un navegador, por lo que no deberemos realizar ninguna transformación adicional. Ésta será, por tanto, la proyección resultante del mapfile.

```
PROJECTION
    "init=epsg:4326"
END
```

- Capas

En el *mapfile* deben definirse las diferentes capas de datos a mostrar en el navegador. Separar la información es capas puede ser muy útil, ya que mediante OpenLayers es posible habilitar o deshabilitar a gusto del usuario la información que presentan en cada una de ellas.

Para PalmaRouting, se ha creado una serie de capas que listamos a continuación.

- ✔ *My Ways*

Capa principal que renderiza la información geométrica contenida en la base de datos. Es en este punto donde se especifica la fuente de datos que contiene toda la información relevante. El elemento LAYER → DATA contiene la consulta a base de datos que especifica el campo geometría que contiene toda la información de la siguiente manera.

the_geom from ways using unique gid using srid = 4326

Donde *using unique gid* especifica la clave primaria de la tabla, y *using srid = 4326* especifica la proyección en la que se encuentran los datos en la tabla.

- ✔ *Street Names*

Capa superpuesta que muestra los nombres de las calles y carreteras. Se ha decidido disponer de esta información como una capa superpuesta (en vez de aprovechar y mostrar las etiquetas en la propia capa base) por un simple tema de legibilidad.

- ✔ *Dead Ends*

Capa superpuesta que muestra la calidad de los segmentos que intersectan con cada nodo. Es una capa de verificación de datos. [27]

El mapfile generado para PalmaRouting implementa el estándar WMS para posteriormente mostrarlo mediante OpenLayers.

http://localhost/cgi-bin/mapserv.exe?
map=/ms4w/apps/palmarouting/palma_routing.map

11.3 OpenLayers

Hasta el punto anterior se ha conseguido publicar un mapa en el servidor local. El siguiente paso es ahora poder trabajar con este mapa, solicitando información o bien tratando los resultados de dichas peticiones. Este es el punto final a la práctica, donde por fin se consigue interactuar con el mapa en un navegador. Podremos observar las localizaciones creadas, cambiar nuestra posición actual, y decidir a qué punto de interés nos dirigiremos.

OpenLayers consiste en una librería de Javascript de código abierto para mostrar mapas interactivos en un navegador web. ofrece una API muy amplia y variada para crear multitud de aplicaciones. Podemos estudiarla con más detalle en este enlace.
http://dev.openlayers.org/docs/files/OpenLayers-js.html

OpenLayers soporta varios estándares OGC[28], como WMS y WFS, y además está basado en orientación a objectos de Javascript. Como concepto básico, la intención de esta librería es poder separar las herramientas de mapas de las herramientas de datos, de manera que cualquier herramienta pueda trabajar sobre cualquier fuente de datos.

27 Ver validación de segmentos en apartado 10.2 PgRouting

28 Ver apartado 6.3.1 Open Geospatial Consortium (OGC)

11.3.1 Implementación de PalmaRouting

Para utilizar OpenLayers partimos de la base de un fichero PHP con código javascript. El primer paso es incluir las librerías de OpenLayers en el código de la siguiente manera.

<script src="http://www.openlayers.org/api/OpenLayers.js"></script>

De esta manera, tendremos siempre la última versión disponible del API para PalmaRouting.

Para obtener y mostrar la configuración definida en el *mapfile*, se deben crear tantas capas OpenLayers como capas del mapfile hay disponibles y deseemos mostrar. En OpenLayers es posible definir dos tipos de capas: capas de base y capas superpuestas. Además, cada capa se puede definir de un tipo diferente (WMS, Markers y Vector en nuestro caso).

A continuación se detalla cada una de las capas disponibles en la aplicación web y su equivalencia en el mapfile, si existe.

- ✔ **Capas base**
  - OpenLayers.Layer.WMS::'OpenLayers WMS' == MapFile::'MyWays'

- ✔ **Capas superpuestas**
  - OpenLayers.Layer.WMS::'Street Names' == MapFile::'StreetNames'
  - OpenLayers.Layer.WMS::'Dead Ends' == MapFile::'DeadEnds'
  - OpenLayers.Layer.Markers::'My Position'
  - OpenLayers.Layer.Vector::'Routing Results'

La capa superpuesta *My Position* muestra con dos sencillos marcadores la posición de inicio actual del usuario y la posición final seleccionada por el usuario (el punto de interés a visitar). No está definida en el *mapfile.*

La capa superpuesta *Routing Results* muestra la ruta calculada por *pgRouting* entre los dos puntos seleccionados. Tampoco está definida en el *mapfile.*

Para calcular una ruta, la información que se necesita es la siguiente.

- ✔ Punto de partida
- ✔ Punto de llegada
- ✔ Algoritmo a utilizar

Una vez definidas estas variables, se procede a calcular la ruta. Lo que muestra un resultado final de la siguiente manera, marcando el inicio y el fin de la ruta a seguir entre los dos puntos.

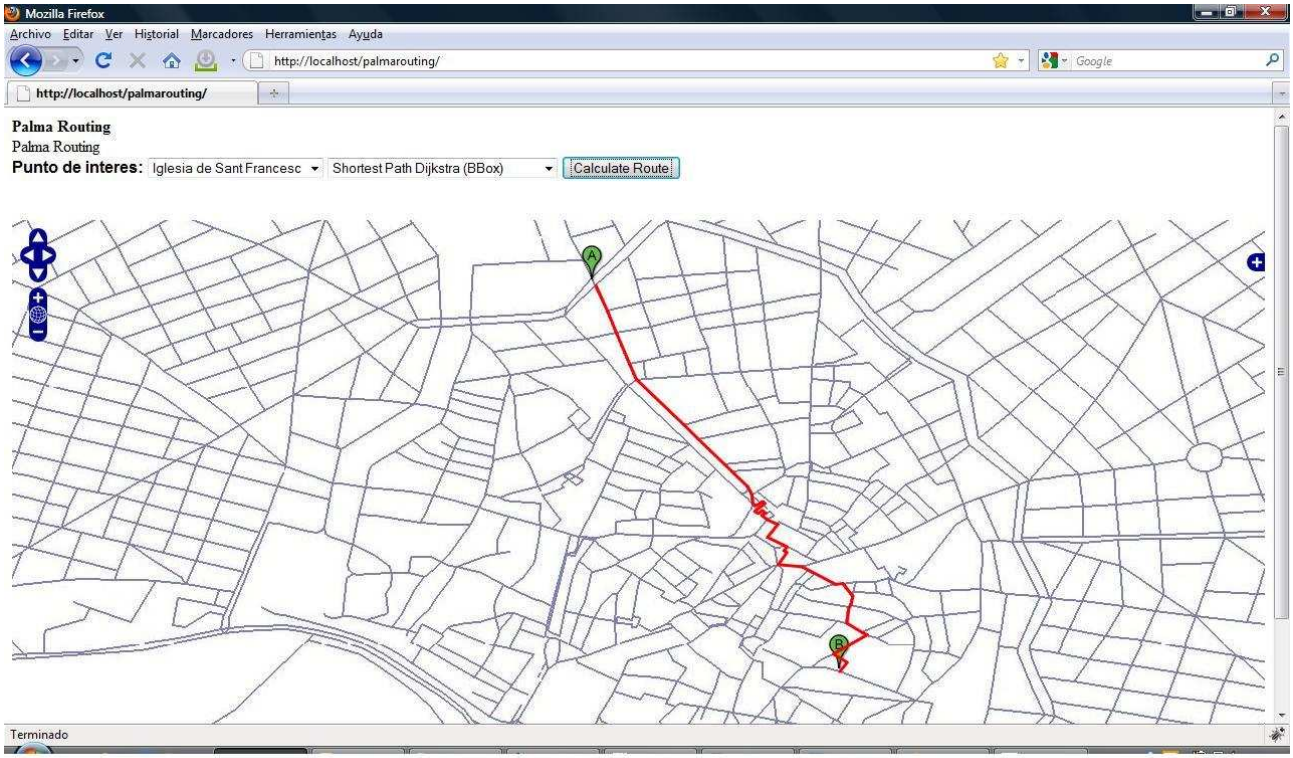


Figura 7. Ejemplo de cálculo de ruta mediante el algoritmo de Dijkstra

En la figura 7 podemos ver claramente el resultado de aplicar el algoritmo de Dijkstra a dos puntos inicio y fin. El punto fin es un punto de interés seleccionado por el usuario en la lista desplegable que aparece en pantalla, mientras que el punto de inicio lo marca el usuario al teclear con el ratón en algún punto del mapa. El resultado es lo que aparece en pantalla. En cuanto al algoritmo A*, el cálculo de la ruta ha

devuelto el mismo resultado, tal y como anticipábamos y esperábamos. No ocurre lo mismo con *Shooting Star*, cuyo resultado se muestra en la figura 8.

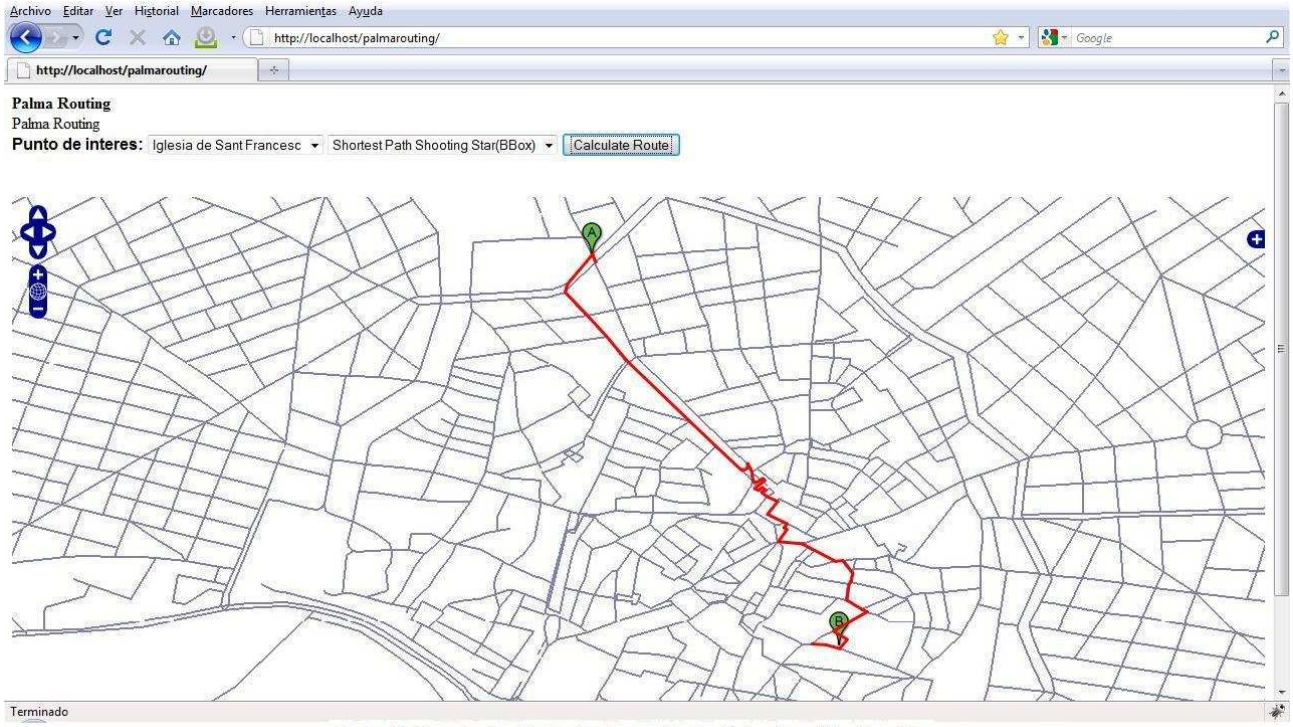


Figura 8. Ejemplo de cálculo de ruta mediante el algoritmo Shooting Star

Podemos observar ahora los controles que ofrece OpenLayers para el manejo de capas y probamos su uso.

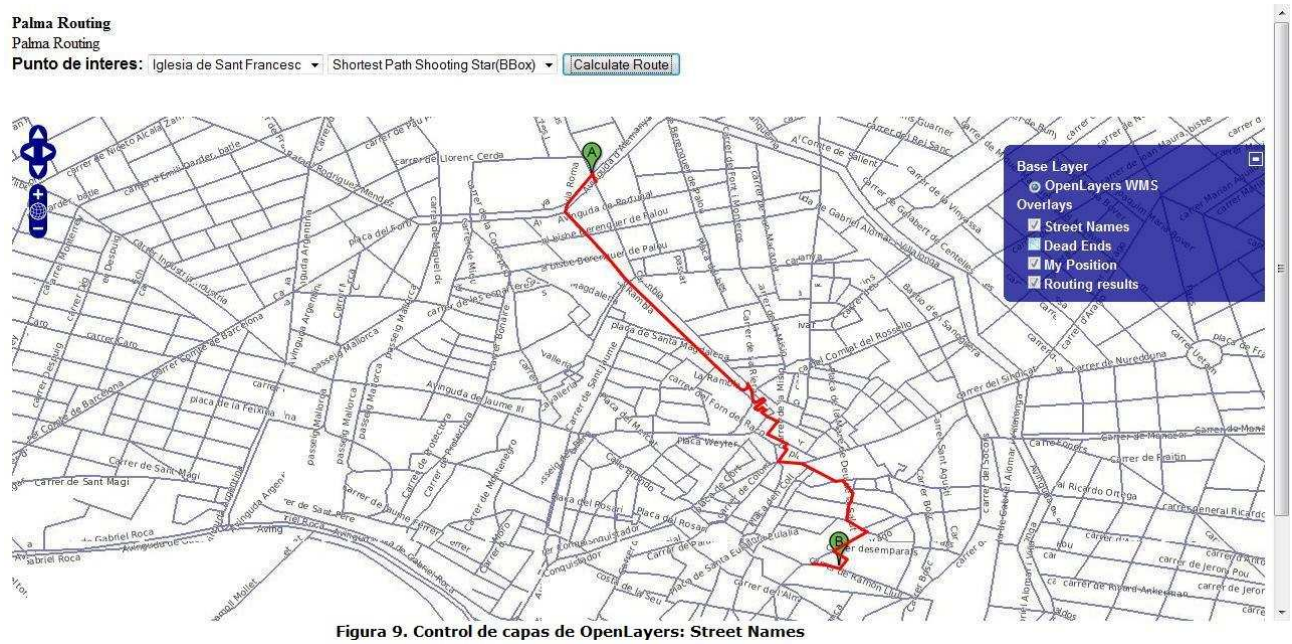


Figura 9. Control de capas de OpenLayers: Street Names

En la imagen de la figura 9, vemos claramente el listado de capas diferenciando la capa base de las demás capas superpuestas. No es posible prescindir de la capa base, pero sí de las superpuestas, y podemos probar a jugar con ellas. En esta imagen hemos habilitado la capa *Street Names*, que muestra los nombres de calles y carreteras.

En la siguiente figura (figura 10), se ha habilitado la capa Dead Ends para mostrar la diferencia (siguiendo la línea de las anteriores figuras), y además se ha aprovechado para modificar los puntos de origen y fin del algoritmo a calcular.

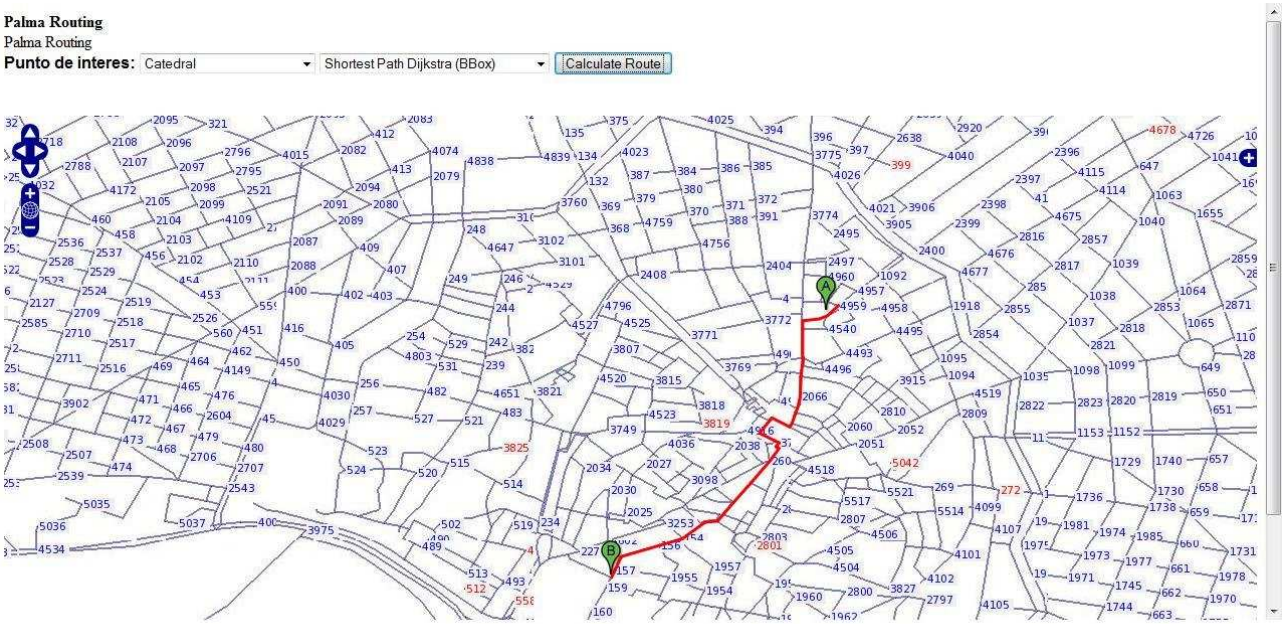


Figura 10. Control de capas de OpenLayers: Dead Ends

El resultado devuelto por esta prueba tiene una particularidad, que podemos ver más detalladamente en la siguiente figura (figura 11).

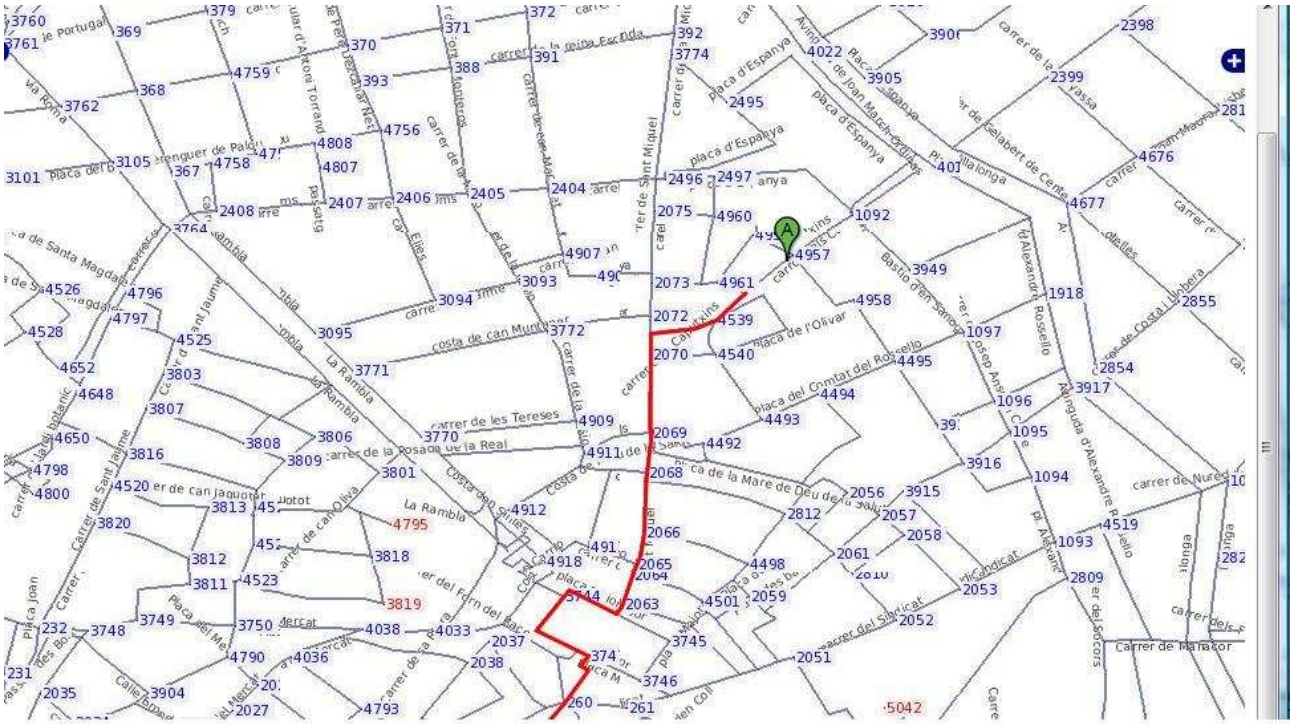


Figura 11. Inicio de ruta calculada vs punto de inicio del usuario

Podemos ver que el punto de inicio de la ruta no se corresponde exactamente con la posición inicial seleccionada por el usuario. Esto se debe a que la posición del usuario no se corresponde exactamente con ningún nodo de la red. La solución implementada busca el nodo más cercano a la posición del usuario y calcula el algoritmo tomando este nuevo punto como referencia. El resultado es que visualmente no se corresponden los dos puntos y da la sensación de que el algoritmo es incorrecto, pero en realidad no es así.

Una posible solución a este detalle podría ser mover el marcador y situarlo exactamente sobre el inicio de la ruta calculada.

12 Conclusiones

El proyecto ha llegado a su fin y se realiza un resumen y un listado de conclusiones de cómo ha ido el proyecto a lo largo de su desarrollo.

12.1 Objetivos

El proyecto ha cumplido sus objetivos. La idea básica era calcular la ruta a seguir entre dos puntos de una geografía y mostrarla en un navegador. Además, la solución se ha implementado mediante las herramientas especificadas en el enunciado del proyecto (OpenStreetMap, PostgreSQL + PostGIS, pgRouting, OpenLayers).

12.2 Líneas futuras de trabajo

Esta aplicación tiene varios frentes abiertos en los que se puede seguir trabajando. El primero, y desde mi punto de vista, el más urgente, es solucionar el problema de la representación exacta del nodo de inicio del usuario con respecto al inicio de la ruta calculada[29]. En realidad se trata de un problema meramente visual, ya que todo el tratamiento interno y la programación son correctos. Pero también es verdad que a día de hoy, los problemas visuales dan sensaciones de errores más graves de lo que realmente son.

Otra mejora a considerar es la eliminación de los segmentos *dead-end*[30], ya que se trata de segmentos que conducen a ninguna parte y son totalmente irrelevantes en el proceso de cálculo de rutas. Además, no estaría de más mejorar el mapa de manera que tengo capas de color para que la aplicación, aunque funcione bien y sin problemas, sea más vistosa de cara al usuario, así como una interfaz más amigable.

Finalmente, sería interesante realizar comparativas con otros algoritmos de rutas, como TSP (*Travelling Sales Person*), y ampliar las posibilidades de la aplicación.

29 Ver figuras 10 y 11 del apartado 11.3.1 Implementación de PalmaRouting

30 Ver validación de segmentos en apartado 10.2 PgRouting

12.3 Mi experiencia

En primer lugar, debo admitir que me ha costado mucho trabajo y esfuerzo realizar este PFC. La carga de trabajo ha sido mucho mayor de la esperada, lo que ha supuesto continuos retrasos en la planificación inicial. Se trata de una tecnología muy nueva para mí, y adquirir los conocimientos, así como aprender a manejar las herramientas necesarias para llevarlo a cabo.

Realizar una planificación inicial (PAC1) ha sido muy útil, pues si bien marca las pautas de tiempo, también sirve para darse cuenta uno mismo si está cumpliendo la planificación o si se está retrasando en el proyecto. En cuanto a fechas y duración de tareas, la previsión inicial no se ha ajustado a lo que realmente era necesario para llevarlo a cabo. Esto es debido a que es muy difícil realizar una estimación en duración de una tecnología desconocida. En mi caso, y dado mi gran desconocimiento en el terreno SIG, las horas dedicadas a cada una de las tareas prácticamente siempre ha sido superior a lo estimado en la planificación inicial.

En cuanto a las herramientas utilizadas, he tenido problemas con la herramienta de importación de formato OSM a la base de datos PostgreSQL. Inicialmente realicé la importación con la **osm2pgsql,** sin darme cuenta de que esta herramienta es muy buena para renderizar el mapa, pero no es apta para aplicar sobre ella cálculo de rutas. Inicialmente perdí mucho tiempo investigando cuál era la causa del error, sin resultado alguno. Cuando me di cuenta de que el fallo estaba en los datos y no en los algoritmos, tuve que volver atrás para realizar el proceso de importación mediante **osm2pgrouting,** la herramienta correcta. También en este punto me retrasé, ya que esta herramienta no está disponible para Windows, y para utilizarla he tenido que utilizar una máquina virtual con Ubuntu para realizar el proceso de importación correctamente, y posteriormente pasarlo a mi máquina Windows, donde ya tenía todas las herramientas configuradas.

Pero no todo ha sido malo este semestre. La temática de este PFC me ha adentrado en una tecnología hasta ahora desconocida por mí. Ahora que ya he terminado todo el desarrollo del proyecto, resulta gratificante ver el resultado obtenido. Debo admitir que no es fácil conciliar la vida laboral, la vida familiar y el PFC, pero el esfuerzo ha merecido la pena.

13 Anexo I: Mapfile

```
#======================================================================
# Map Definition
#======================================================================
MAP
 NAME PalmaRouting
 EXTENT 2.5326453 39.4782307 2.9157413 39.6900794
 SIZE 600 600
 SYMBOLSET "palma_routing.sym"
 FONTSET "fonts.txt"
 DEBUG on
 CONFIG "MS_ERRORFILE" "C:\ms4w\apps\palmarouting\ms_error.txt"
 LEGEND
 IMAGECOLOR -1 -1 -1
 LABEL
 FONT "vera"
 ANGLE FOLLOW
 COLOR 0 0 0
 ENCODING "UTF-8"
 TYPE truetype
 SIZE 8
 END
 STATUS ON
 TRANSPARENT ON
 END
#======================================================================
# Web Interface Definition
#======================================================================
 WEB
 TEMPLATE "template.html"
 IMAGEPATH "C:/ms4w/apps/palmarouting/tmp/"
 IMAGEURL  "C:/ms4w/apps/palmarouting/tmp/"
 METADATA
 "wms_encoding" "UTF-8"
 "wms_title" "Mapserver WMS"
 "wms_abstract" "Esta es la capa que muestra el routing de Palma de Mallorca."
 "wms_srs" " EPSG:4326"
 "wms_onlineresource" "http://localhost/cgi-bin/mapserv.exe?map=\ms4w\apps\palmarouting\palma_routing.map"
 END
 END
#======================================================================
# Projection Definition
#======================================================================
 PROJECTION
                    "init=epsg:4326"
```

```
        END
#======================================================================
# Layer: PlanetOsmLine
#======================================================================
        LAYER
                NAME "MyWays"
                STATUS ON
                TYPE LINE
                OFFSITE 255 255 255
                TOLERANCE 200
                CONNECTIONTYPE POSTGIS
                CONNECTION "user=postgres password=postgres host=localhost port=5432 dbname=gis"
                DATA "the_geom from ways using unique gid using srid=4326"
                MAXSCALE -1.0
                MINSCALE -1.0
                SIZEUNITS pixels
                DEBUG on
                PROJECTION
                        "init=epsg:4326"
                END
                CLASS
                        STYLE
                                COLOR 128 128 192
                                WIDTH 2
                        END
                        NAME "default"
                END
                METADATA
                        "wms_title" "public.ways"
                        "wms_abstract" "generated by gvSIG"
                        "wms_name" "name"
                        "wms_extent" "2.5326453 39.4782307 2.9157413 39.6900794"
                        "gml_include_items" "all"
                END
        END # Layer MyWays

#======================================================================
# Layer: DeadEnds
#======================================================================
        LAYER
                NAME "DeadEnds"
                STATUS ON
                TYPE POINT
                OFFSITE 255 255 255
                TOLERANCE 20
                CONNECTIONTYPE POSTGIS
```

```
        CONNECTION "user=postgres password=postgres host=localhost port=5432 dbname=gis"
        DATA "the_geom from vertices_tmp as foo using unique id using SRID=4326"
        PROJECTION
                "init=epsg:4326"
        END
        CLASSITEM 'cnt'
        CLASS
                Text ([id])
                EXPRESSION /1/
                STYLE
                        SIZE 11
                        COLOR 255 0 0
                END
                LABEL
                        TYPE TRUETYPE
                        ANTIALIAS TRUE
                        FONT 'vera'
                        COLOR 255 0 0
                        BACKGROUNDCOLOR 240 240 240
                        POSITION cr
                        MINSIZE 8
                        MAXSIZE 12
                        BUFFER 2
                END
        END
        CLASS
                TEXT ([id])
                EXPRESSION /./
                STYLE
                        SIZE 11
                        COLOR 0 0 255
                END
                LABEL
                        TYPE TRUETYPE
                        ANTIALIAS TRUE
                        FONT 'vera'
                        COLOR 0 0 255
                        BACKGROUNDCOLOR 240 240 240
                        POSITION cr
                        MINSIZE 8
                        MAXSIZE 12
                        BUFFER 2
                END
        END
    END # Layer DeadEnds
#======================================================================
```

```
# Layer: StreetNames
#==========================================================================
        LAYER
                NAME "StreetNames"
                STATUS ON
                TYPE ANNOTATION
                OFFSITE 255 255 255
                TOLERANCE 200
                CONNECTIONTYPE POSTGIS
                CONNECTION "user=postgres password=postgres host=localhost port=5432 dbname=gis"
                DATA "the_geom from ways using unique gid using srid=4326"
                MAXSCALE -1.0
                MINSCALE -1.0
                SIZEUNITS pixels
                PROJECTION
                        "init=epsg:4326"
                END
                CLASSITEM 'name'
                CLASS
                   Text([name])
                        EXPRESSION /./
                        STYLE
                                COLOR   128 128 192
                                WIDTH   2
                        END
                        LABEL
                                TYPE TRUETYPE
                                ANGLE AUTO
                                FONT 'vera'
                                COLOR 55 55 55
                                POSITION auto
                                MINSIZE 7
                                MAXSIZE 9
                                BUFFER 2
                        END
                END
                CLASS
                        STYLE
                                COLOR 128 128 192
                                WIDTH 2
                        END
                        NAME "default"
                END
        END # Layer DeadEnds
END # Map File
#==========================================================================
```

14 Bibliografía

Sistemas de Información Geográfica (SIG)

- http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica
- http://es.wikipedia.org/wiki/OGC
- http://es.wikipedia.org/wiki/Web_Map_Service
- http://es.wikipedia.org/wiki/WFS
- http://www.opengeospatial.org/
- http://es.wikipedia.org/wiki/Open_Geospatial_Consortium
- http://www.idee.es/show.do?to=pideep_pidee.ES
- http://www.cnig.es/
- http://www.iie.org.mx/boletin022007/tend.pdf

Cartografía

- http://es.wikipedia.org/wiki/Cartografía
- http://cartografia.supaw.com/

Sistemas de coordenadas

- http://es.wikipedia.org/wiki/Sistema_de_coordenadas
- http://es.wikipedia.org/wiki/Coordenadas_geográficas
- http://es.wikipedia.org/wiki/Sistema_de_Coordenadas_Universal_Transversal_de_Mercator

OpenStreetMap

- http://es.wikipedia.org/wiki/OpenStreetMap
- http://wiki.openstreetmap.org/wiki/ES:FAQ
- http://www.openstreetmap.org/export/

Base de datos

- http://es.wikipedia.org/wiki/PostGIS
- http://es.wikipedia.org/wiki/Normalización_de_bases_de_datos

Routing

- http://es.wikipedia.org/wiki/Encaminamiento
- http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra
- http://es.wikipedia.org/wiki/Problema_del_viajante
- http://es.wikipedia.org/wiki/Topología_de_red
- http://en.wikipedia.org/wiki/A*_search_algorithm
- http://www.osgeo.jp/wordpress/wp-content/uploads/2009/11/workshop_manual.pdf
- http://en.wikipedia.org/wiki/Wrapper_function

MapServer

- http://es.wikipedia.org/wiki/MapServer
- http://mapserver.org/
- http://mapserver.org/mapfile/index.html
- http://ms.gis.umn.edu:8081/ms_plone/docs/reference/mapfile

OpenLayers

- http://es.wikipedia.org/wiki/OpenLayers
- http://www.openlayers.org/
- http://dev.openlayers.org/docs/files/OpenLayers-js.html