

*(Creative Commons)*

*Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons. Podeu copiar-lo, distribuir-lo i transmetre'l públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada. La llicència completa es pot consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>*

# **BCN-Routing:** càlcul web de rutes amb pgRouting, OpenStreetMap i OpenLayers

**Treball final de carrera**

*Enginyeria Tècnica en Informàtica de Sistemes*

Juny 2010

*Alumne:* **Oriol Vallcorba Valls**

*Consultora:* **Anna Muñoz Bolas**

**Abstract**

The use of Geographic Information Systems (GIS) to solve spatial problems and get information about land-related aspects is becoming a common practice in institutions and companies. Therefore, taking advantage of already created open source geospatial tools is vital for the creation of high quality free GIS.

Routing applications are gaining relevance every day thanks to the popularization of GPS and future Galileo. A previous data preparation and a correct spatial database structure is required to successfully apply complex algorithms to it. In the past few years, the development of pgRouting as an open source library that provides routing functionality to PostgreSQL/PostGIS has opened new possibilities for shortest path search in these databases.

In this contribution, we have developed a GIS system for shortest path search between a given point and an interesting place in the city of Barcelona. The street information has been extracted from OpenStreetMap, a layer of tourist attraction points has been created, and everything has been stored in a PostGIS database. After a data structure preparation for routing capabilities, we will apply the Disjktra algorithm implementation provided in pgRouting to search for the shortest path between two nodes. Result presentation in a web-mapping application will be build using OpenLayers library.

Routing tools supplied in the pgRouting extension for PostGIS databases, as well as additional tools to work with OpenStreetMap data, simplify the process of data structure creation and facilitate the resolution of shortest path search routing problems.

# Índex

<b>Introducció.....</b>	<b>1</b>
1.1 Context del treball.....	1
1.2 Objectius.....	2
1.3 Enfocament i metodologia aplicada.....	3
1.4 Planificació del projecte.....	3
1.5 Estructura de la memòria.....	4
<b>Fonaments teòrics.....</b>	<b>5</b>
2.1 Els Sistemes d'Informació Geogràfica (SIG).....	5
2.1.2 Altres sistemes d'informació gràfica.....	6
2.2 Geodèsia i cartografia.....	6
2.2.1 Sistemes de coordenades.....	7
2.2.2 Projeccions cartogràfiques.....	7
2.2.3 Sistemes de referència.....	9
2.2.4 Tipus de dades cartogràfiques.....	10
2.3 Emmagatzematge de la informació geogràfica.....	10
2.3.1 Topologia.....	11
<b>Entorn de treball.....</b>	<b>12</b>
3.1 MapServer.....	12
3.2 PostGIS.....	13
3.3 pgRouting.....	14
3.4 OpenLayers.....	14
3.5 Eines complementaries.....	14
<b>Obtenció i tractament de dades.....</b>	<b>15</b>
4.1 Selecció de les fonts i dades cartogràfiques.....	15
4.1.2 Extracció de les dades d'OpenStreetMap.....	16
4.1.3 Referenciació dels punts turístics.....	16
4.2 Creació de la base de dades espacial PostGIS.....	18
4.3 Inserció de les dades geogràfiques a la base de dades.....	19
4.3.1 Inserció de les dades d'OpenStreetMap.....	19
4.3.2 Inserció de la capa de punts turístics.....	20
4.3.3 Definició del model de dades.....	21
4.3.4 Transformació a coordenades projectades.....	22
4.4 Implementació de consultes sobre la base de dades.....	23

<b>Càlcul de rutes amb pgRouting.....</b>	<b>28</b>
5.1 Introducció.....	28
5.2 Preparació de les dades.....	29
5.3 Càlcul de rutes mitjançant l'algoritme de Dijkstra.....	31
<b>Entorn web amb OpenLayers.....</b>	<b>35</b>
6.1 Funcionament d'OpenLayers.....	35
6.2 Connexió entre PostGIS, UMN MapServer i OpenLayers.....	36
6.3 Preparació de les dades.....	39
6.4 Entorn i càlcul de rutes via web.....	42
6.4.1 Fitxer map per accedir a les dades PostGIS.....	43
6.4.2 Codi HTML de la pàgina i de l'script d'OpenLayers.....	46
6.4.3 Codi del fitxer de càlcul de rutes (calculruta.php).....	54
6.4.4 Codi del fitxer de selecció i marca del punt turístic (mostraPunt.php).....	56
6.5 Exemple de funcionament.....	57
<b>Conclusions i línies futures.....</b>	<b>60</b>
Referències.....	63
Bibliografia general.....	64

# Capítol 1: Introducció

## 1.1 Context del treball

Actualment existeixen al mercat un gran nombre d'eines pel desenvolupament de Sistemes d'informació geogràfica<sup>1</sup> (SIG) amb diferents característiques i prestacions. Entre aquestes eines, n'hi ha de programari propietari importants com el paquet Geomedia<sup>®</sup> d'Intergraph o els productes ArcGIS<sup>®</sup> d'ESRI (*Environmental Systems Research Institute*), però també n'hi ha moltes de programari lliure<sup>2,3</sup> que permeten el desenvolupament de SIGs en tots els àmbits. El programari lliure, respecte el propietari, presenta els avantatges de tenir un cost de llicència nul, la possibilitat d'ús sense restriccions, el suport pels estàndards oberts i el fet de ser productes del desenvolupament per part d'un elevat nombre d'usuaris. Per contra, aquests programes són normalment més costosos d'instal·lar i de fer funcionar al no tenir la garantia d'una empresa al darrera, i al no disposar normalment d'una documentació complerta i exhaustiva dels mateixos.

Per el desenvolupament del SIG d'aquest treball es procurarà emprar únicament eines de programari lliure: el servidor de mapes (UNM MapServer), la base de dades geogràfica (Postgre SQL amb l'extensió PostGIS per l'emmagatzematge de geometries), el càlcul de rutes (pgRouting) i la presentació visual en un portal web (OpenLayers). Les dades geogràfiques també s'extrauran d'una font lliure, com és el cas d'OpenStreetMap.

L'àrea seleccionada pel treball és la ciutat de Barcelona i es considerarà el problema de

determinar la ruta més curta que pot prendre algú que no coneix la ciutat per tal d'arribar a una de les principals atraccions turístiques donat un punt d'origen. Es troben a la xarxa exemples similars com el càlcul del restaurant més proper a una boca de metro de Niigata<sup>4</sup> o del camí més segur en bicicleta entre dos punts de Nova York.<sup>5</sup>

Resumint, aquest projecte pretén explotar les possibilitats que ofereix un Sistema d'Informació Geogràfica (SIG) per tal de calcular la ruta de distància més curta donades dues localitzacions en un territori utilitzant eines de programari lliure o de codi obert.

## 1.2 Objectius

La realització d'aquest projecte té com a finalitat proporcionar la formació necessària per entendre el funcionament i la metodologia correcta de treball amb un SIG, i ser capaç d'aplicar-ho en un cas concret de resolució d'un problema de càlcul de rutes. Per això es poden assenyalar els següents objectius generals:

- Comprendre i familiaritzar-se amb els conceptes i la metodologia de treball de la tecnologia dels SIG.
- Conèixer l'estructura dels diferents tipus de dades amb què treballa un SIG.
- Conèixer els sistemes d'emmagatzemament estàndards de la informació ràster i vectorial, i ser capaç d'ubicar la informació en les coordenades que correspongui.
- Conèixer les operacions d'anàlisi espacial i transformacions en el SIG analitzat.
- Desenvolupar el projecte SIG utilitzant entorns i eines de programari lliure.

I els següents objectius específics referents al SIG a desenvolupar:

- Dins el marc de les aplicacions lliures, ser capaç d'aprofitar els avantatges que pot oferir un servidor de mapes d'aquest tipus com UNM MapServer per la creació dels mapes.
- Treballar amb dades extretes del projecte OpenStreetMap.
- Entendre i utilitzar degudament les funcions de càlcul de rutes disponibles a l'extensió pgRouting (Dijkstra, problemes del tipus viatjant de comerç,...).

- Presentació de les rutes obtingudes com a resultat en un entorn web mitjançant el client OpenLayers sobre el servidor de mapes UNM MapServer.

### **1.3 Enfocament i metodologia aplicada**

Per la realització del treball, primerament s'ha plantejat un problema a resoldre mitjançant l'ús d'un SIG, que caldrà desenvolupar mitjançant eines de programari lliure. Per això s'aplicarà la següent metodologia:

- Estudi teòric dels components d'un SIG, de les seves aplicacions, i dels fonaments de geodèsia i cartografia per tal de saber com es representen els resultats en forma de mapes. Es llegiran els materials de l'assignatura específics per als SIG, es seleccionarà algun llibre per aprofundir en alguns conceptes i es consultaran les pàgines web de les institucions més importants en l'àrea així com exemples d'aplicacions.
- Selecció del sistema operatiu i les eines de programari lliure a utilitzar. Caldrà instal·lar-les i familiaritzar-se amb el seu ús llegint la documentació respectiva i les experiències d'altres usuaris.
- Selecció de les fonts de dades geogràfiques necessàries pel problema. Introducció d'aquestes dades a una base de dades espacial. Serà necessari entendre el funcionament i les possibilitats d'aquest tipus de base de dades consultant la documentació.
- Aplicació dels algoritmes de càlcul de rutes sobre les dades. Aquesta part requerirà una preparació prèvia de les dades (dotar-les de tipologia).
- Presentació dels resultats en entorn web. Si s'utilitza OpenLayers caldrà familiaritzar-se amb el llenguatge Javascript, que és el que utilitza.

Paral·lelament al seguiment d'aquests punts en l'ordre proposat, s'anirà redactant la memòria del projecte a mesura que es realitzin els diferents apartats.

### **1.4 Planificació del projecte**

Per tal d'organitzar i planificar el projecte s'ha realitzat prèviament un pla de treball. En aquest



pla, s'han previst les possibles tasques o activitats necessàries pel seu desenvolupament i la seva temporització aproximada. S'han representant en un diagrama de Gantt per veure clarament les relacions entre aquestes i la temporització de forma més gràfica. Ha estat necessari distribuir les tasques en quatre blocs corresponents a les entregues parcials del projecte, aquests blocs són:

- Planificació del projecte (1r lliurament)
- Preparació de l'entorn de treball i obtenció de dades (2n lliurament).
- Càlcul de rutes i desenvolupament de la interfície d'usuari (3r lliurament).
- Lliurament final de la memòria del projecte i la presentació.

Finalment, s'ha realitzat una estimació dels possibles apartats de la memòria i la seva extensió. El pla de treball complet es troba a l'annex d'aquest treball.

## 1.5 Estructura de la memòria

La memòria del treball s'ha estructurat en capítols, que es poden agrupar en tres grups:

- **Capítols Introductoris:** Dins aquest grup s'hi inclou aquest capítol d'**introducció**, on s'exposen la planificació i objectius del treball, el capítol de **fonaments teòrics**, que pretén servir de base teòrica del treball introduint al lector els principals conceptes per entendre el funcionament d'un SIG, i finalment el capítol d'**entorn de treball**, on es descriu el programari que s'ha utilitzat pel desenvolupament del projecte.
- **Capítols de desenvolupament del SIG:** En aquests capítols s'explica el procediment pràctic seguit per la realització del treball. S'hi inclou el capítol d'**obtenció i tractament de dades** que descriu el procés de creació de la base de dades, el capítol de **càlcul de rutes amb pgRouting**, on s'aplica l'algoritme de Dijkstra sobre les dades, i el capítol de **presentació dels resultats amb OpenLayers**, amb la creació de l'entorn web per mostrar els mapes.
- **Capítol de Conclusions:** Finalment s'inclouen unes conclusions del treball, una descripció de les limitacions del SIG, les dificultats trobades i les possibilitats de futur.

## Capítol 2:

# Fonaments teòrics

En aquest capítol s'introduiran al lector els conceptes més rellevants i necessaris per entendre el funcionament dels SIG. El capítol està estructurat en subapartats on es defineix què s'entén per sistema d'informació geogràfica, com es representa la informació geogràfica i com s'emmagatzema aquest tipus d'informació.

### 2.1 Els Sistemes d'Informació Geogràfica (SIG)

La definició de Sistema d'Informació Geogràfica (SIG) és tan àmplia i avarca aspectes i funcionalitats tant diferents que hi ha llibres que dediquen capítols sencers per la seva definició.<sup>6</sup> El glossari de l'*Open Geospatial Consortium*<sup>7</sup> (organització que té com a finalitat la definició d'estàndards oberts per als SIG) defineix un SIG com

*“el sistema informàtic per capturar, emmagatzemar, comprovar, integrar, manipular, analitzar i mostrar informació relacionada amb posicions sobre la superfície terrestre”*

i defineix una aplicació SIG com

*“l'ús de les capacitats, incloent maquinari, programari i dades, que proporciona un SIG per la satisfacció d'una sèrie de requeriments de l'usuari”*

Si s'analitzen aquestes definicions:

- S'ha assenyalat que la informació amb la que treballa un SIG està relacionada amb elements

de la superfície terrestre, és a dir, es tracta d'**informació geogràfica**. Aquí sorgeix un dels aspectes més importants dels SIG, ja que la manipulació d'informació geogràfica comporta una sèrie de problemes intrínsecs pel que fa la seva representació i emmagatzematge.

- Permet la **manipulació** i l'**anàlisi** d'aquesta informació geogràfica, a més de recopilar, emmagatzemar i mostrar aquesta informació de forma eficient.
- Està destinat a la **resolució de problemes espacials** que es plantegin els usuaris.

Es pot comprovar que existeixen bàsicament dues aproximacions en la definició de SIG segons si es dóna més importància a la vessant pràctica o a la tècnica, però que ambdues són necessàries per la seva completa definició: un SIG ha de poder descriure, explicar i predir situacions o problemes a nivell geogràfic a partir de sistemes de maquinari i programari que permetin la recopilació, manipulació i anàlisi de informació geogràfica.

### **2.1.2 Altres sistemes d'informació gràfica**

La característica que diferencia els SIG d'altres sistemes que també permeten la creació i emmagatzematge de mapes o entitats gràfiques, com podrien ser la cartografia digital o els sistemes CAD (*computer-aided design*), és la **capacitat de processar dades** geogràfiques.

La cartografia digital permet el disseny, la producció i l'emmagatzematge de mapes mitjançant ordinadors mentre que els sistemes CAD s'usen per dissenyar i dibuixar de forma precisa entitats gràfiques (punts, línies, polígons), molts cops amb l'objectiu de la creació de mapes. A diferència dels SIG, aquests sistemes no permeten una **gestió, manipulació i anàlisi** de les dades de forma que es generi **nova informació** i permeti la **presa de decisions** sobre el territori, sinó que es podrien considerar com eines per generar la informació que després els SIG utilitzen per al seu anàlisi.

## **2.2 Geodèsia i cartografia**

Tal com s'ha vist a l'apartat anterior, els SIG utilitzen informació geogràfica per a resoldre els problemes espacials. Per una banda cal definir unívocament els punts de la superfície de la terra de forma precisa (georeferenciació) i per altra, cal definir sistemes de projecció de la superfície de la Terra (forma esfèrica) en superfícies planes, minimitzant les deformacions o errors en la projecció.

Aquí és on entren la geodèsia (ciència que estudia la figura, les dimensions i el camp gravitatori de la terra) i la cartografia (ciència que s'encarrega de l'estudi i la producció de representacions de la informació espacial, per exemple en forma de mapes).

### 2.2.1 Sistemes de coordenades

Per tal de definir la posició d'un objecte sobre la superfície terrestre els sistemes de coordenades més utilitzats són:

- **Sistema de coordenades geogràfiques:** Determinació d'un punt sobre la superfície a partir de dos angles mesurats des del centre de la terra, que s'anomenen **latitud** i **longitud**. La latitud ( $\alpha$ ) d'un punt és l'angle mesurat al llarg de l'equador des de qualsevol punt de la Terra i la longitud ( $\beta$ ) és l'angle mesurat des del centre de la Terra cap al nord. Les línies d'igual longitud s'anomenen **meridians** i la longitud es mesura prenent de referència el meridià de Greenwich (longitud zero). Les línies d'igual latitud s'anomenen **paral·lels** i es pren l'equador com a latitud zero.
- **Sistema de coordenades cartesianes:** Les coordenades d'un punt en un espai tridimensional es defineixen com a  $(x,y,z)$ , on l'eix Z passa pel centre de la Terra i els pols, l'eix X passa pel centre de la Terra i el meridià de Greenwich i l'eix Y forma un angle de  $90^\circ$  amb els altres dos.
- **Sistema de coordenades projectades:** Aquest sistema de coordenades es defineix sobre una superfície plana. Es defineix un origen  $(X, Y) = (0, 0)$ , que esdevé el centre d'una malla on les posicions es defineixen relatives a aquest origen.

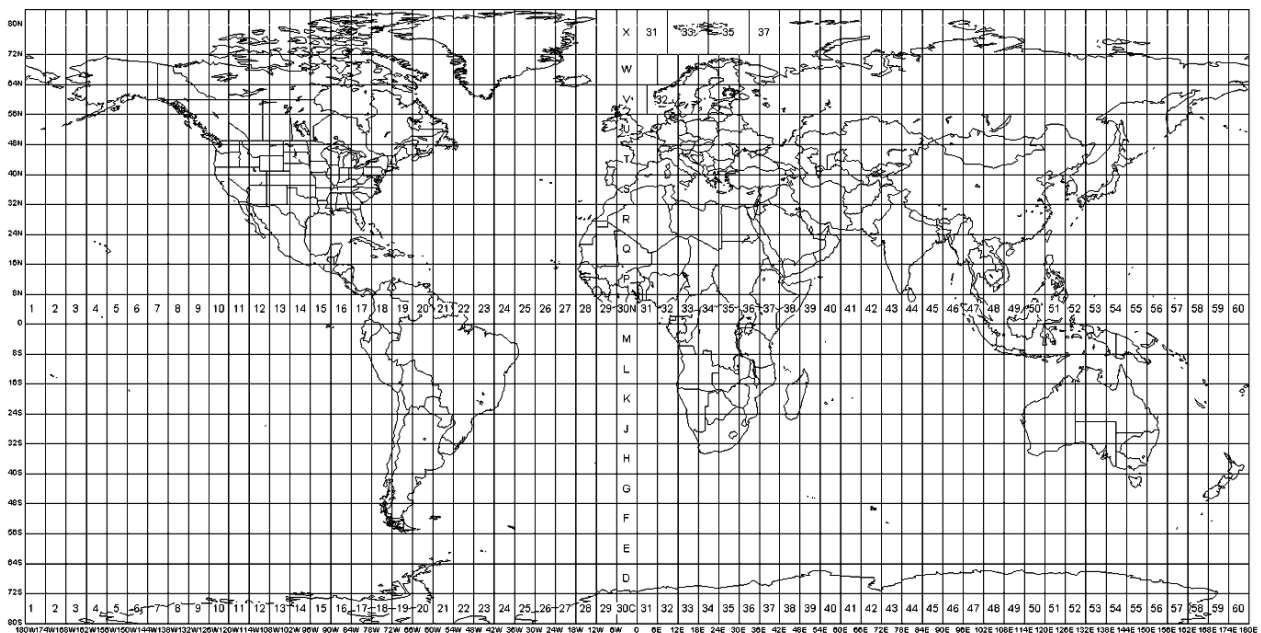
### 2.2.2 Projeccions cartogràfiques

El procés necessari per expressar un punt de la superfície esfèrica terrestre en una superfície plana es coneix com a projecció cartogràfica. Hi ha diferents tipus de projeccions que distorsionen propietats geomètriques diferents. Segons quina propietat distorsionen, hi ha projeccions conformes que mantenen la forma de la superfície, projeccions equivalents que mantenen les àrees projectades, projeccions equidistants que mantenen les distàncies, projeccions azimuthals que conserven les direccions i projeccions de compromís, que busquen un balanç en l'alteració de totes les propietats.

Les projeccions també es poden classificar segons la superfície d'on deriven (superfícies desplegable), que poden ser còniques, cilíndriques o planes.

Segons la zona que es vol cartografiar és important escollir una projecció adequada per tal d'assegurar la mínima distorsió i mantenir les propietats desitjades.

Una de les projeccions més utilitzades és la **Projecció transversal universal de Mercator (UTM)**. Aquest sistema es basa en seixanta **projeccions cilíndriques** de l'esfera terrestre amb diferents meridians de tangència, per tal de minimitzar les deformacions en tota la superfície projectada. D'aquesta forma la superfície es divideix en **fusos** de 6° que contenen al centre el meridià de tangència per cada projecció, assegurant que qualsevol punt de la terra estarà com a màxim a 3° de distància d'una línia de tangència. Addicionalment, cada fus es divideix en vint zones latitudinals de 8° (exceptuant la zona de més nòrdica de 12°), encara que a la pràctica es considera cada fus dividit únicament en dues meitats: nord i sud en referència a l'equador (figura 2-1).



**Figura 2-1.** Distribució dels fusos UTM. Es pot comprovar que a Catalunya li correspon el fus 31 Nord.

[Font: *Harvard University Graduate School of Design* ([www.gsd.harvard.edu](http://www.gsd.harvard.edu))]

Altres consideracions d'aquest sistema de projecció són:

- S'eliminen les coordenades negatives establint un valor de *fals nord* a l'equador i un valor de *fals est* al meridià central del fus. Aquests valors són de 500000 m i de 10000000 m respectivament. D'aquesta forma les coordenades de l'hemisferi sud i de l'est del meridià

central seran positives.

- Les coordenades UTM d'un punt sobre un mapa es defineixen per tres paràmetres: el fus on es troba, la coordenada E (eix horitzontal) i la coordenada N (eix vertical).
- És un sistema dissenyat per cartografiar objectes d'un únic fus, ja que projeccions d'àrees grans (de fusos diferents) comporten elevades distorsions.

### 2.2.3 Sistemes de referència

El mètode per definir de manera única la localització dels objectes sobre la Terra és el que es coneix com **georeferenciació**. Per això es defineix una superfície de referència i un sistema de referència (datum):

- **Superfície de referència:** Per simplificar la forma de la terra s'utilitzen dos models: el geoide i l'el·lipsoide. El **geoide** correspon a la superfície on la direcció de la gravetat és perpendicular en tots els llocs i es correspon amb el valor de la gravetat que s'experimenta en el nivell mitjà dels oceans. La mesura del geoide no és fàcil i el model de superfície que defineix és irregular i no el fa adequat per càlculs matemàtics. L'**el·lipsoide** és una superfície sense irregularitats, és la superfície matemàtica simple que més s'aproxima a la forma de la Terra. Per tal d'ajustar aquest model en zones específiques de la terra s'utilitzen el·lipsoides locals. Per la zona de Catalunya s'utilitzen els el·lipsoides Internacional de Hayford de 1924, el *world geodetic datum* de 1984 (WGS84), i el *geodetic reference system* de 1980 (GRS80).
- **Sistema de referència geodesic (Datum):** Els sistemes de referència són els encarregats de relacionar les localitzacions geogràfiques amb els punts de l'el·lipsoide i del geoide. Per això cal que defineixin un el·lipsoide de referència, les coordenades geogràfiques d'uns punts escollits (a partir de les xarxes geodèsiques, que consisteixen en un conjunt de punts sobre el territori localitzats amb gran precisió) i la orientació del sistema. El sistema europeu ETRS89 és el sistema de referència geodèsic oficial per la georeferenciació en territori espanyol des de 2007, que utilitza l'el·lipsoide GRS80, i substitueix el sistema WGS84 utilitzat fins aleshores.

L'existència de múltiples sistemes de referència geodèsics diferents porta que moltes vegades siguin necessàries conversions entre aquests. Aquest procediment és possible i s'aconsegueix a partir

de la conversió de les coordenades entre els el·lipsoides de referència dels dos datums.

#### **2.2.4 Tipus de dades cartogràfiques**

Per finalitzar aquest subapartat es descriuran breument els tipus bàsics de representació de dades geogràfiques amb que treballen els SIG. Principalment es distingeixen dos tipus d'estructura de dades:

- **Dades vectorials:** Defineixen l'espai segons la seva geometria en forma de punts, línies i polígons, i les seves propietats topològiques en forma de nodes, arcs i polígons. Aquests elements es representen en forma de coordenades, i aquesta estructura de dades s'utilitza normalment per representar objectes de dimensions cartogràfiques negligibles (p.e. punts interès), xarxes (p.e. carreteres) i elements de dues dimensions (p.e. edificacions o tipus de sol).
- **Dades ràster:** Defineix l'espai a partir de la seva divisió en una malla de píxels (àrea quadrada de mínima informació en una imatge digital). Aquesta és l'estructura de dades amb que s'emmagatzemen els ortofotomapes i s'utilitza principalment per la seva representació o per la representació d'altres tipus de dades en forma d'imatge.

L'elecció del tipus de dades depèn de la funcionalitat desitjada, ja que mentre que el model vectorial proporciona millor precisió gràfica i permet topologia, el model ràster representa millor els fenòmens continus i faciliten la integració de les dades.

### **2.3 Emmagatzematge de la informació geogràfica**

Per tal que els SIG permetin realitzar manipulacions, modificacions, consultes o càlculs sobre les dades geogràfiques, és imprescindible que disposin d'un mètode d'emmagatzematge d'aquest tipus de dades. En una primera instància es podria pensar en l'emmagatzematge de les geometries vectorials en bases de dades relacionals en forma normal, però aquest fet comportaria un gran nombre de taules, reduint el rendiment i dificultant el manteniment. Per això s'utilitzen **bases de dades amb capacitats espacials**, que permeten emmagatzemar les geometries en formats binaris complexos, mantenint la indexació d'aquests a la base de dades.

Segons el SGBD utilitzat la forma d'emmagatzemar la geometria i la informació geogràfica serà diferent, però en tots els cassos, un SIG que es connecti a la base de dades ha de poder determinar a quines taules hi ha informació geogràfica, en quins camps i amb quins sistemes de coordenades està referenciada.

Entre els avantatges d'emmagatzemar la informació espacial en bases de dades es poden destacar el ràpid accés a les dades que proporciona la indexació, la realització de consultes espacials, i la possibilitat de dotar de topologia a les dades:

### **2.3.1 Topologia**

En un SIG, la topologia fa referència a les propietats de les formes geomètriques que es mantenen invariables quan aquestes es sotmeten a processos d'escalat o projecció.<sup>7</sup> Les propietats topològiques més importants són la connectivitat, l'ordre, l'adjacència, la proximitat, la inclusió i la intersecció.

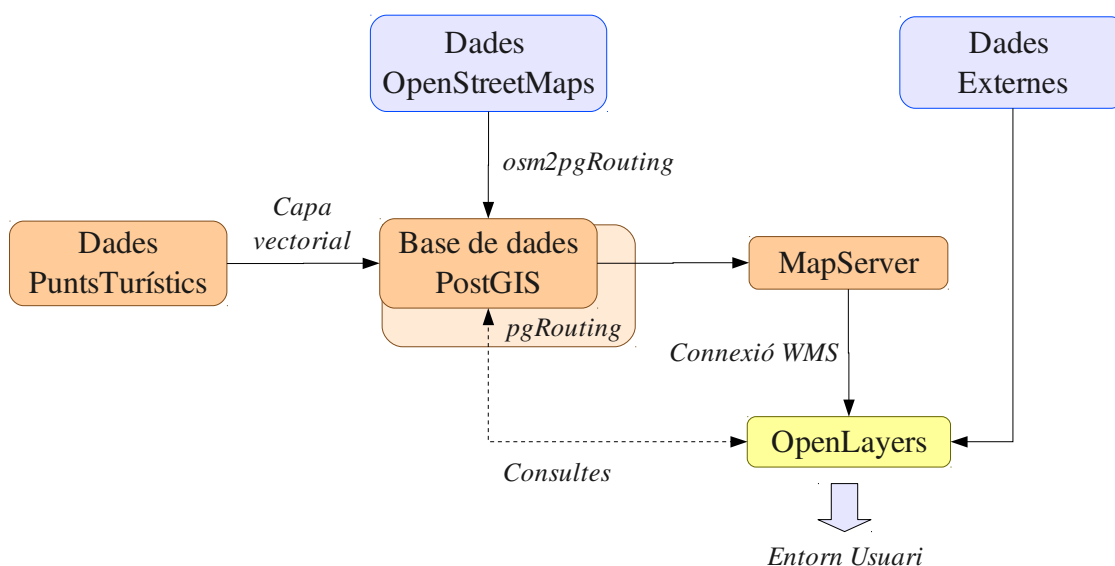
Per tal de dotar de topologia a les dades d'una base de dades espacial, s'emmagatzemen els elements bàsics de la topologia: nodes, arestes i cares, i les relacions entre aquests elements, i aleshores les entitats fan referència a aquests elements per indicar la seva posició sobre el territori.

El fet d'utilitzar topologia permet realitzar càlculs sobre la connectivitat dels elements tractant-los com a grafs dirigits, fet que redueix el cost computacional d'alguns càlculs complexos com la determinació del camí més curt entre dos punts (nodes). Un exemple clar seria la modelització d'una xarxa de carreteres a partir de nodes i arestes, i buscar el camí de menor cost (cost = distància) entre dos nodes aplicant un algoritme de càlcul sobre el graf.



## Capítol 3: Entorn de treball

En aquest apartat de la memòria es defineix l'entorn de treball i el programari utilitzat pel desenvolupament del SIG (figura 3-1). Com ja s'ha comentat en el context i als objectius del treball, s'utilitzarà únicament programari lliure corrent en un sistema operatiu Ubuntu Linux 9.10 de 64 bits.



**Figura 3-1.** Relacions entre les eines de programari i les dades a tractar.

### 3.1 MapServer

MapServer<sup>8</sup> és una plataforma de publicació de dades espacials i mapes interactius en un entorn web. Es tracta bàsicament d'un programa *CGI* instal·lat en un servidor web que rep peticions per la

generació dels mapes i publicació d'aquests com a imatges. Aquestes peticions es realitzen a través del navegador i de *Mapfiles*, que es tracta de fitxers d'extensió *map* amb la configuració referent a l'accés a la informació geogràfica i al format del mapa de sortida.

MapServer suporta els estàndards definits per l'*Open Geospatial Consortium* (OGC), com WMS (*Web Map Service*) o WFS (*Web Feature Service*), i permet la visualització i consulta de multitud de formats de dades geogràfiques, de tipus ràster, vectorial o procedents de bases de dades geogràfiques. A través de la llibreria *Proj4* té suport per la creació de diferents projeccions dels mapes de forma instantània.

En aquest treball s'utilitzarà MapServer 5.6.3 en un servidor web Apache 2.2.12.

## 3.2 PostGIS

Com a base de dades geogràfica s'utilitzarà l'extensió PostGIS<sup>9</sup> pel sistema gestor de bases de dades relacional PostgreSQL, que aporta suport per objectes geogràfics. PosGIS segueix l'especificació *Simple Features for SQL* de l'OGC.

PostGIS inclou diferents tipus de geometries vectorials (*points, linestrings, polygons, multipoints, multilinestrings, multipolygons* i *geometrycollections*), operadors espacials per determinar interaccions entre geometries, calcular mesures o realitzar operacions geomètriques (a través de les llibreries GEOS) i indexació espacial de les dades mitjançant estructures GiST (*Generalized Search Tree*).<sup>10</sup> Aquest mètode d'accés a les dades és una generalització que unifica arbres de cerca habituals (per exemple *B+-trees* o *R-trees*) i permet la seva implementació, superant, en el cas de les bases de dades PostgreSQL, algunes limitacions de la seva implementació nativa dels *R-trees*, com el suport per *features* de mides superiors a 8 kB o la possibilitat de columnes geomètriques amb valors nuls.

La informació geogràfica s'emmagatzema en columnes del tipus GEOMETRY, que queden enregistrades en la taula GEOMETRY\_COLUMNS que descriu les característiques d'aquestes columnes, com el tipus de geometria, les dimensions espacials de la columna i el sistema de referència espacial. Els diferents sistemes de referència espacials es troben a la taula SPATIAL\_REF\_SYS, que conté la seva descripció i proporciona informació per la seva projecció o per la transformació entre ells.

En aquest treball s'ha utilitzat l'extensió PostGIS 1.5.0 (sobre PostgreSQL 8.4.2), compilada manualment seguint les instruccions de la documentació oficial, juntament amb la llibreria geomètrica GEOS 3.1.1<sup>11</sup> i la llibreria de projeccions cartogràfiques Proj 4.7.0.<sup>12</sup>

### 3.3 pgRouting

Per el càlcul de rutes s'utilitzarà l'extensió pgRouting,<sup>13</sup> que es tracta d'una extensió per la base de dades postGIS que aporta la funcionalitat pel càlcul de rutes de mínim cost entre dos nodes a partir d'algoritmes eficients pel càlcul de camins mínims en grafs dirigits com el de Dijkstra. Aquests càlculs s'apliquen sobre les consultes a la base de dades PostGIS, havent dotat prèviament de topologia a les dades.

En aquest treball s'ha utilitzat pgRouting versió 1.0.3, compilat seguint les instruccions de la documentació oficial, juntament la llibreria Boost 1.38.<sup>14</sup>

### 3.4 OpenLayers

OpenLayers<sup>15</sup> es tracta d'una llibreria *JavaScript* per la visualització de mapes en entorn web. Entre els diferents mètodes d'accés a les dades geogràfiques que implementa, hi ha els protocols WMS o WFS estandarditzats per l'OGC, i permet incorporar mapes procedents de diverses fonts a una pàgina web, aportant funcionalitats de navegació i de selecció de capes.

S'utilitzarà la versió en línia d'OpenLayers de la seva pròpia pàgina web, actualment la versió 2.8.

### 3.5 Eines complementaries

Per facilitar o automatitzar algunes tasques s'utilitzaran les següents eines:

- osm2pgrouting<sup>16</sup>: Aplicació que permet importar dades d'OpenStreetMap a una base de dades postGIS amb el format apropiat per tal de poder aplicar els algoritmes de pgRouting.
- gvSIG,<sup>17</sup> uDIG,<sup>18</sup> qGIS<sup>19</sup>: SIGs d'escriptori que permeten visualitzar i treballar més fàcilment amb les dades geogràfiques de la base de dades postGIS o de servidors WMS. El seu entorn gràfic permet la creació de capes de forma visual.
- Firebug.<sup>20</sup> Extensió del Mozilla Firefox per a la depuració del llenguatge Javascript.

## Capítol 4:

# Obtenció i tractament de dades

En aquest capítol es descriu d'on s'obtenen les dades geogràfiques necessàries per la realització del projecte, com s'obtenen aquestes dades, i de quina forma s'emmagatzemen a una base de dades espacial PostGIS per tal que posteriorment es puguin resoldre els problemes de càlcul de rutes

### 4.1 Selecció de les fonts i dades cartogràfiques

L'àrea geogràfica a tractar en el problema correspon a la ciutat de Barcelona. Per tal de realitzar el càlcul de la ruta més curta cap a un punt d'interès turístic, serà necessari disposar de les dades dels carrers de la ciutat així com de les localitzacions dels punts turístics. Aquestes dades s'hauran d'obtenir de forma que es puguin introduir en una base de dades espacial per tal de poder realitzar consultes i càlculs sobre aquestes.

Les dades dels carrers de Barcelona s'extrauran d'OpenStreetMap,<sup>21</sup> que es tracta d'un projecte col·laboratiu per tal de crear i proporcionar dades geogràfiques lliures. Els punts turístics es seleccionaran i referenciaran de forma manual mitjançant la generació d'una capa amb el programari gvSIG.

En l'entorn d'usuari final s'utilitzaran altres fonts de dades cartogràfiques per tal de dotar d'un aspecte més complet al mapa resultant. Per això s'utilitzaran ortofotos de la zona obtingudes mitjançant sol·licituds al WMS de l'Institut Cartogràfic de Catalunya<sup>22</sup> (ICC).

### 4.1.2 Extracció de les dades d'OpenStreetMap

Per tal de poder utilitzar posteriorment les dades d'OpenStreetMap en una base de dades PostGIS s'exporta una selecció de la zona de Barcelona en format XML (Extensible Markup Language). D'aquesta forma s'exporta tota la informació de les vies, nodes, relacions i etiquetes en un fitxer amb extensió *osm* que posteriorment es pot convertir a format PostGIS mitjançant conversors (*osm2pgsql* o *osm2pgrouting*). L'àrea exportada en el nostre cas és el requadre comprès entre les latituds 41,3281541 i 41,4669139 i les longituds 2,0742169 i 2,2450411 (figura 4-1), coordenades geogràfiques expressades en graus decimals.

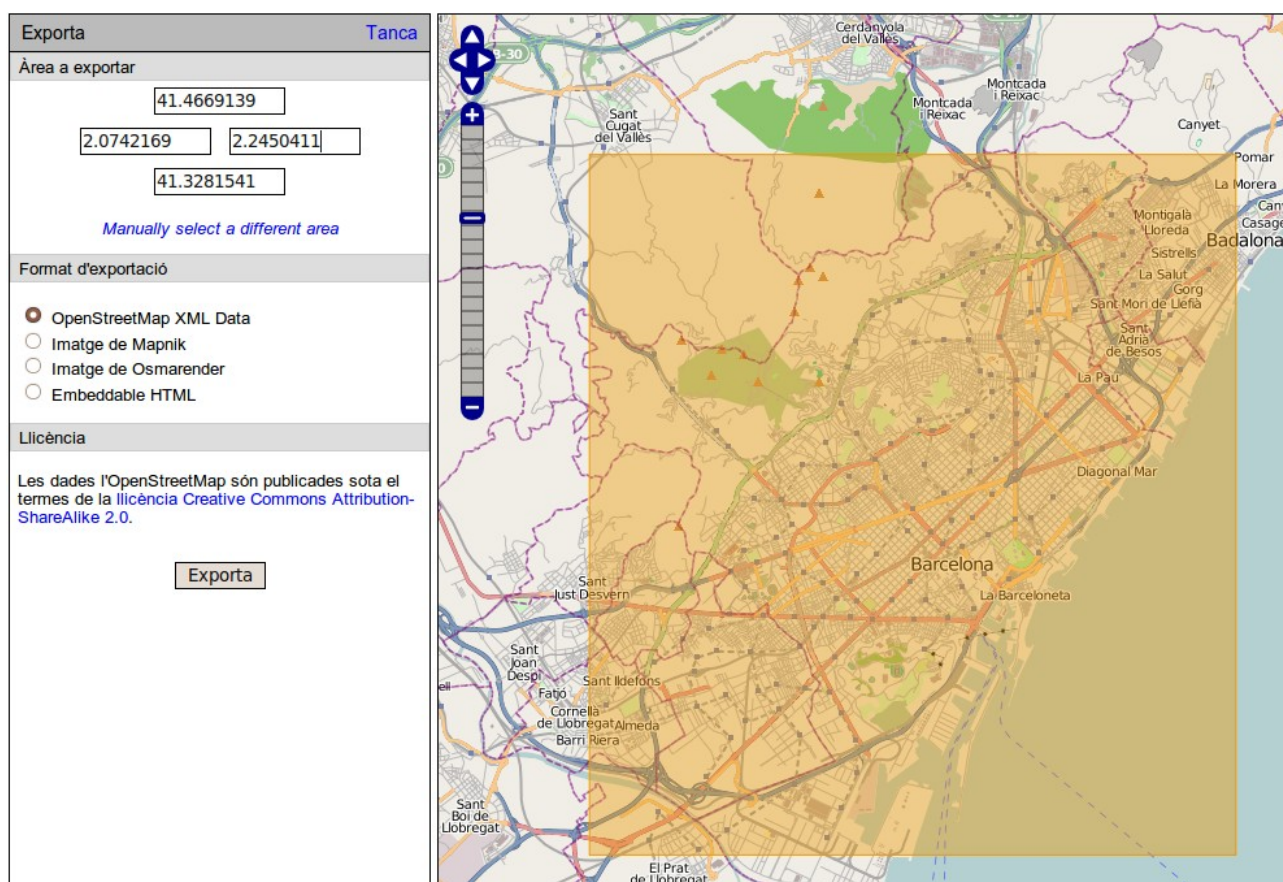


Figura 4-1. Àrea de la ciutat de Barcelona exportada d'OpenStreetMap

### 4.1.3 Referenciació dels punts turístics

S'han seleccionat inicialment 12 llocs d'interès turístic de la ciutat de Barcelona (taula 4-1). Per facilitar la seva localització s'ha utilitzat el SIG d'escriptori Quantum GIS (qGIS), des del qual s'han anat afegint els punts turístics en una capa vectorial utilitzant el tipus de geometria de punt. En una altra capa (sol·licitada al WMS de l'Institut Cartogràfic de Catalunya) s'ha visualitzat una ortofoto

de la ciutat de Barcelona per poder localitzar els llocs d'interès directament.

**Taula 4-1.** Llocs d'interès turístic seleccionats per la ciutat de Barcelona

Nom	Latitud	Longitud	Descripció
Temple de la Sagrada Família	41.403984	2.173911	Església catòlica dissenyada per l'arquitecte Antoni Gaudí, símbol del modernisme català
Les Rambles	41.385609	2.169992	Passeig emblemàtic de la ciutat de Barcelona, prop del centre històric, en el seu recorregut es troba el teatre del Liceu, el mercat de la Boqueria i el monument a Colón
La Pedrera	41.395228	2.161686	Edifici modernista dissenyat per l'arquitecte Antoni Gaudí
Parc Güell	41.413150	2.152910	Gran jardí amb elements arquitectònics dissenyat per Antoni Gaudí
Montjuïc	41.371550	2.151390	Font màgica, un dels punts per pujar a la muntanya de Montjuïc, de gairebé 200m d'altura i vistes a la ciutat
Camp Nou	41.379150	2.119570	Camp del futbol club Barcelona, millor equip del món l'any 2009
Catedral de la Santa Creu i Santa Eulàlia	41.384070	2.176000	Catedral de Barcelona, seu de l'arquebisbat i d'arquitectura gòtica
Palau de la música catalana	41.387490	2.174860	Auditori de música d'estil modernista de l'arquitecte Lluís Domènech i Muntaner
Museu d'art contemporani de Barcelona (MACBA)	41.382770	2.166630	Museu dedicat a l'exposició d'obres de la segona meitat del segle XX, situat al barri del Raval
Port Olímpic	41.388640	2.198710	Port esportiu de Barcelona
Plaça de Sant Jaume	41.382630	2.177040	Cor administratiu de la ciutat de Barcelona, on s'hi troba el seu ajuntament així com el Palau de la Generalitat de Catalunya
Gran teatre del Liceu	41.380470	2.173590	Reconegut teatre d'òpera situat a les Rambles de Barcelona

Aquesta capa de punts es pot realitzar com una capa apart (per exemple amb format ESRI *shapefile*) i convertir-la posteriorment al format de la base de dades postGIS, o bé es pot crear primer una taula a la base de dades amb una columna per la geometria i introduir-hi directament aquests punts utilitzant el mateix SIG d'escriptori. En aquest treball s'ha realitzat directament sobre la base de dades, però la metodologia per la localització és equivalent en tots dos casos. La creació de la taula per l'emmagatzematge dels punts turístics s'explica a l'apartat 4.3.2.

## 4.2 Creació de la base de dades espacial PostGIS

Per la creació de la base de dades espacial PostgreSQL / PostGIS s'ha creat una plantilla de base de dades PostgreSQL sobre la que s'ha afegit la funcionalitat PostGIS pel suport d'objectes geomètrics i la funcionalitat pgRouting per poder utilitzar els seus algorismes de càlcul de rutes. Posteriorment es crearà la base de dades de treball (bcnrouting) amb capacitats espacials (postGIS) utilitzant aquesta plantilla. S'han seguit les indicacions de la documentació oficial de postGIS<sup>9</sup> i el procediment ha estat el següent:

- 1) Creació de la base de dades PostgreSQL que servirà de plantilla postGIS i activació del llenguatge PL/pgSQL en aquesta base de dades:

```
createdb -U oriol -E UTF8 pgis_base
createlang -U oriol plpgsql pgis_base
```

- 2) Addició de la funcionalitat postGIS:

```
psql -U oriol -f /usr/share/postgresql/8.4/contrib/postgis-1.5/postgis.sql pgis_base
psql -U oriol -f /usr/share/postgresql/8.4/contrib/postgis-1.5/spatial_ref_sys.sql
pgis_base
```

- 3) Addició de la funcionalitat pgRouting (bàsica i tsp):

```
psql -U oriol -f /usr/share/postlbs/routing_core.sql pgis_base
psql -U oriol -f /usr/share/postlbs/routing_core_wrappers.sql pgis_base
psql -U oriol -f /usr/share/postlbs/routing_topology.sql pgis_base
psql -U oriol -f /usr/share/postlbs/routing_tsp.sql pgis_base
psql -U oriol -f /usr/share/postlbs/routing_tsp_wrappers.sql pgis_base
```

- 4) Comprovació de l'addició postGIS:

```
psql -U oriol pgis_base
pgis_base=# select postgis_full_version();

                postgis_full_version
-----
POSTGIS="1.5.0" GEOS="3.1.1-CAPI-1.6.0" PROJ="Rel. 4.7.1, 23 September 2009" LIBXML="2.7.5" USE_STATS
(1 row)
```

- 5) Creació de la base de dades espacial de treball bcnrouting a partir de la plantilla pgis\_base:

```
createdb -U oriol -T pgis_base bcnrouting
```

## 4.3 Inserció de les dades geogràfiques a la base de dades

Seguidament s'explicarà el procediment seguit per la inserció de les dades geogràfiques a la base de dades espacial postGIS, tant dels carrers de Barcelona com dels punts turístics. Posteriorment s'explicarà de quina forma s'emmagatzemen aquestes dades a la base de dades.

### 4.3.1 Inserció de les dades d'OpenStreetMap

Per tal d'inserir les dades geogràfiques extretes d'OpenStreetMap a una base de dades espacial PostGIS hi ha varis mètodes per realitzar-ho. S'ha considerat l'ús de les eines `osm2pgsql`<sup>23</sup> i `osm2pgrouting`,<sup>16</sup> optant per la segona ja que crea automàticament topologia a les dades, requisit indispensable si es volen aplicar funcions de càlcul de costos.

Per inserir les dades extretes d'OpenStreetMap de la ciutat de Barcelona (apartat 4.1.2) des del fitxer `bcn.osm` a la base de dades PostGIS `bcnrouting` utilitzant `osm2pgrouting` es realitza amb la següent comanda:

```
./osm2pgrouting -file ./bcn.osm -conf ./mapconfig.xml -dbname bcnrouting -user oriol -passwd
origis
```

El fitxer de configuració `mapconfig.xml` conté la informació dels tipus de vies que s'han d'inserir a la base de dades i s'ha utilitzat el que ve per defecte amb `osm2pgrouting`. El resultat després de la inserció és el següent:

```
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "nodes_pkey" for table "nodes"
Nodes table created
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "ways_pkey" for table "ways"
Ways table created
Types table created
Classes table created
create topology
NOTICE: CREATE TABLE will create implicit sequence "vertices_tmp_id_seq" for serial column
"vertices_tmp.id"
CONTEXT: SQL statement "CREATE TABLE vertices_tmp (id serial)"
PL/pgSQL function "assign_vertex_id" line 14 at EXECUTE statement
#####
size of streets: 4134
size of splitted ways : 12957
finished
```

Finalment es pot comprovar que el procés ha funcionat correctament fent un llistat de les taules generades:



```
psql -U oriol bcnrouting
bcnrouting=# \d
                List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | classes        | table | oriol
 public | geography_columns | view  | oriol
 public | geometry_columns | table | oriol
 public | nodes          | table | oriol
 public | spatial_ref_sys | table | oriol
 public | types          | table | oriol
 public | vertices_tmp    | table | oriol
 public | vertices_tmp_id_seq | sequence | oriol
 public | ways           | table | oriol
(9 rows)
```

A l'apartat 4.3.3 es farà una descripció de la informació que conté cadascuna de les taules insertades i a l'apartat 4.4 es mostraran de forma visual les dades insertades mitjançant un SIG d'escriptori.

### 4.3.2 Inserció de la capa de punts turístics

La inserció dels punts d'interès turístic de la ciutat de Barcelona que s'han seleccionat i localitzat anteriorment (apartat 4.1.3) s'ha realitzat directament sobre la base de dades postGIS de treball (bcnrouting) mitjançant qGIS.

S'ha creat prèviament una taula anomenada puntsturistics amb columnes pel nom, la latitud, la longitud, la descripció i la columna geomètrica per emmagatzemar la geometria. Les instruccions SQL per la creació de la taula és el següent:

```
CREATE TABLE puntsturistics
(
  gid serial NOT NULL,
  nom text,
  lat double precision,
  lon double precision,
  the_geom geometry,
  descripcio text,
  CONSTRAINT puntsturistics_pkey PRIMARY KEY (gid),
);
SELECT AddGeometryColumn('public', 'puntsturistics', 'the_geom', 4326, 'POINT', 2);
```

Es pot observar la funció per la creació de la columna geomètrica *AddGeometrycolumn()*, on s'especifica que es crearà la columna *the\_geom*, a la taula *puntsturistics*, de perfil *public*, amb el

sistema de coordenades espacial d'identificador 4326 (projecció longitud-latitud, datum WGS84), de dimensió espacial 2 i del tipus d'objecte espacial POINT. Amb aquesta funció queda enregistrada automàticament la columna geomètrica creada a la taula `geometry_columns`.

Seguidament s'ha visualitzat la capa corresponent a aquesta taula al qGIS (inicialment sense cap punt) i s'han addicionat els punts d'interès turístic (figura 4-2), que s'emmagatzemen com a files a la taula.



**Figura 4-2** Visualització de la capa de punts turístics sobre el mapa de carrers de Barcelona.

### 4.3.3 Definició del model de dades

La base de dades postGIS creada (`bcnrouting`) conté dues taules comunes en totes les bases de dades espacials postGIS:

- `spatial_ref_sys`: Taula on es llisten més de 3000 sistemes de referència espacial, amb els detalls de la projecció i de transformació entre ells. Cadascun té el seu identificador, una descripció i la definició segons la llibreria Proj4 per proporcionar les capacitats de transformació.
- `geometry_columns`: Taula que conté la descripció de totes les columnes geomètriques de les diferents taules de la base de dades. Per cadascuna d'aquestes columnes n'enregistra el nom,

la taula on es troba, la dimensió espacial, el sistema de referència utilitzat i el tipus de geometria dels objectes que conté.

A més d'aquestes dues taules pròpies de postGIS, tenim les taules que contenen les dades geogràfiques de treball importades d'OpenStreetMap, corresponents a la ciutat de Barcelona:

- classes i types: Taules que descriuen els tipus de vies de la ciutat i permeten classificar-les segons els seus usos.
- nodes: Taula que conté el llistat de tots els nodes del mapa, cadascun amb un identificador, la longitud, i la latitud, expressades en graus decimals.
- vertices\_tmp: Taula que conté la llista dels nodes com a element geomètric, amb un identificador únic per cadascun i una columna geomètrica amb objectes del tipus punt (POINT).
- ways: Taula que conté tots els carrers o vies del mapa. Cadascun té un identificador únic, la classe de via, el nom, la longitud, les coordenades d'inici i final (expressades com a latitud i longitud) el cost associat (pel càlcul de rutes), una columna geomètrica de tipus polilínia (MULTILINESTRING), i finalment els nodes d'origen i fi identificats per l'identificador d'aquests a la taula vertices\_tmp.

#### **4.3.4 Transformació a coordenades projectades**

De moment, a partir de les dades extretes d'OpenStreetMap i el mètode d'inserció a la base de dades emprat, el sistema de coordenades cartogràfiques utilitzat per la referenciació d'objectes geomètrics és el del Sistema Geodèsic Mundial WGS84. Les coordenades s'expressen a partir de dos angles mesurats des del centre de la terra (latitud,longitud) en graus decimals (projecció EPSG::4326). Fins aquest punt del treball s'ha utilitzat aquest sistema ja que no hi ha hagut necessitat de canviar-lo i no ha esdevingut cap problema.

A partir d'ara, si es volen realitzar càlculs de distàncies o més endavant càlculs de rutes, convé canviar a algun sistema de coordenades projectades. S'utilitzarà una projecció transversal universal de Mercator (UTM) de la zona de treball, que correspon al fus 31N. D'aquesta forma, la unitat de mesura passarà de graus a metres. Per la projecció UTM, es poden utilitzar el sistema de referència geocèntric WGS84 o el ETRS89 (*European Terrestrial Reference System*). Pel fet de ser més nou,

ser l'oficial a Europa i tenir unes precisions més elevades s'utilitzarà el ETRS89.

Per canviar el sistema de referència dels objectes geomètrics de la base de dades s'utilitzaran les funcions `updateGeometrySRID()`, `transform()` i `SetSRID()` de PostGIS. A la taula `spatial_ref_sys`, el SRID de la projecció destí (UTM utilitzant ETRS89) és el 25831, i el de la projecció actual (EPSG::4326 utilitzant WGS84) és el 4326. Les instruccions SQL utilitzades són les següents:

```
SELECT updateGeometrySRID('ways', 'the_geom', 25831);
UPDATE ways set the_geom = transform(SetSRID(the_geom, 4326), 25831);
SELECT updateGeometrySRID('vertices_tmp', 'the_geom', 25831);
UPDATE vertices_tmp set the_geom = transform(SetSRID(the_geom, 4326), 25831);
SELECT updateGeometrySRID('puntsturistics', 'the_geom', 25831);
UPDATE puntsturistics set the_geom = transform(SetSRID(the_geom, 4326), 25831);
```

D'aquesta forma s'actualitzen els SRID de la columna geomètrica i de la fila corresponent a `geometry_columns` (amb la funció `updateGeometrySRID()`) i es transformen les coordenades de cada objecte (de projecció origen 4326, definida per `SetSRID()`) a les de la nova projecció (25831, amb la funció `transform()`).

## 4.4 Implementació de consultes sobre la base de dades

Per tal de comprovar el correcte funcionament de la base de dades es realitzaran algunes consultes simples sobre les dades d'aquesta.

Un primera prova serà el càlcul de la longitud de l'Avinguda Diagonal de Barcelona. Realitzarem aquesta consulta en les diferents fases que han passat les dades per veure com varia la resposta:

- 1) Consulta prèvia al canvi de coordenades, és a dir, amb les dades en la projecció EPSG::4326 i el datum WGS84:

```
SELECT sum(ST_Length(w.the_geom))/1000 AS llargada
FROM ways w WHERE w.name = 'Av Diagonal';
```

```

      llargada
-----
4.20129582337814e-05
(1 row)
```

- 2) Consulta realitzant el canvi de coordenades en els resultats obtinguts directament des de la clàusula `SELECT`. En aquest cas es transformen a coordenades UTM utilitzant el datum

**WGS84:**

```
SELECT sum(ST_Length(ST_Transform(w.the_geom,23031)))/1000 AS llargadakm
FROM ways w WHERE w.name = 'Av Diagonal'
```

```
    llargadakm
-----
    3.5914464877732
(1 row)
```

**3) Consulta equivalent a l'anterior però transformant a UTM utilitzant el datum ETRS89:**

```
SELECT sum(ST_Length(ST_Transform(w.the_geom,25831)))/1000 AS llargadakm
FROM ways w WHERE w.name = 'Av Diagonal'
```

```
    llargadakm
-----
    3.59128867690924
(1 row)
```

**4) Finalment la mateixa consulta sobre les dades ja transformades, sense canvis de coordenades a la pròpia sentència:**

```
SELECT sum(ST_Length(w.the_geom))/1000 AS llargadakm
FROM ways w WHERE w.name = 'Av Diagonal'
```

```
    llargadakm
-----
    3.59128867690924
(1 row)
```

Si s'analitza el resultat es pot observar que al primer cas la llargada que retorna la funció `ST_Length()` és en graus decimals (unitats del sistema de referència) mentre que a l'últim, amb les dades transformades el resultat és en metres. Si es comparen els cassos 2 i 3, s'observa que en tots dos cassos el valor de retorn de la funció és en metres però la longitud del carrer difereix en 0.16 m, mostrant les diferències d'exactitud entre els datums.

Una segona prova serà determinar si dos carrers es creuen donats el seus noms utilitzant la funció `ST_Intersects()` que retorna cert (t) si les dues geometries es toquen:

```
SELECT (ST_Intersects(w1.the_geom, w2.the_geom)) AS es_Creuen
FROM ways w1, ways w2
WHERE ((ST_Intersects(w1.the_geom, w2.the_geom)) = 't') AND (w1.name = 'Carrer Tallers') AND
```

```
(w2.name = 'Carrer Jovellanos');

es_creuen
-----
t
t
(2 rows)
```

Involucrant la taula de punts turístics, es pot realitzar una consulta dels carrers a menys de 20 metres dels punts turístics seleccionats utilitzant la funció ST\_distance(). També es mostraran les coordenades dels punts turístics i l'identificador dels carrers (ja que alguns no tenen el nom enregistrat):

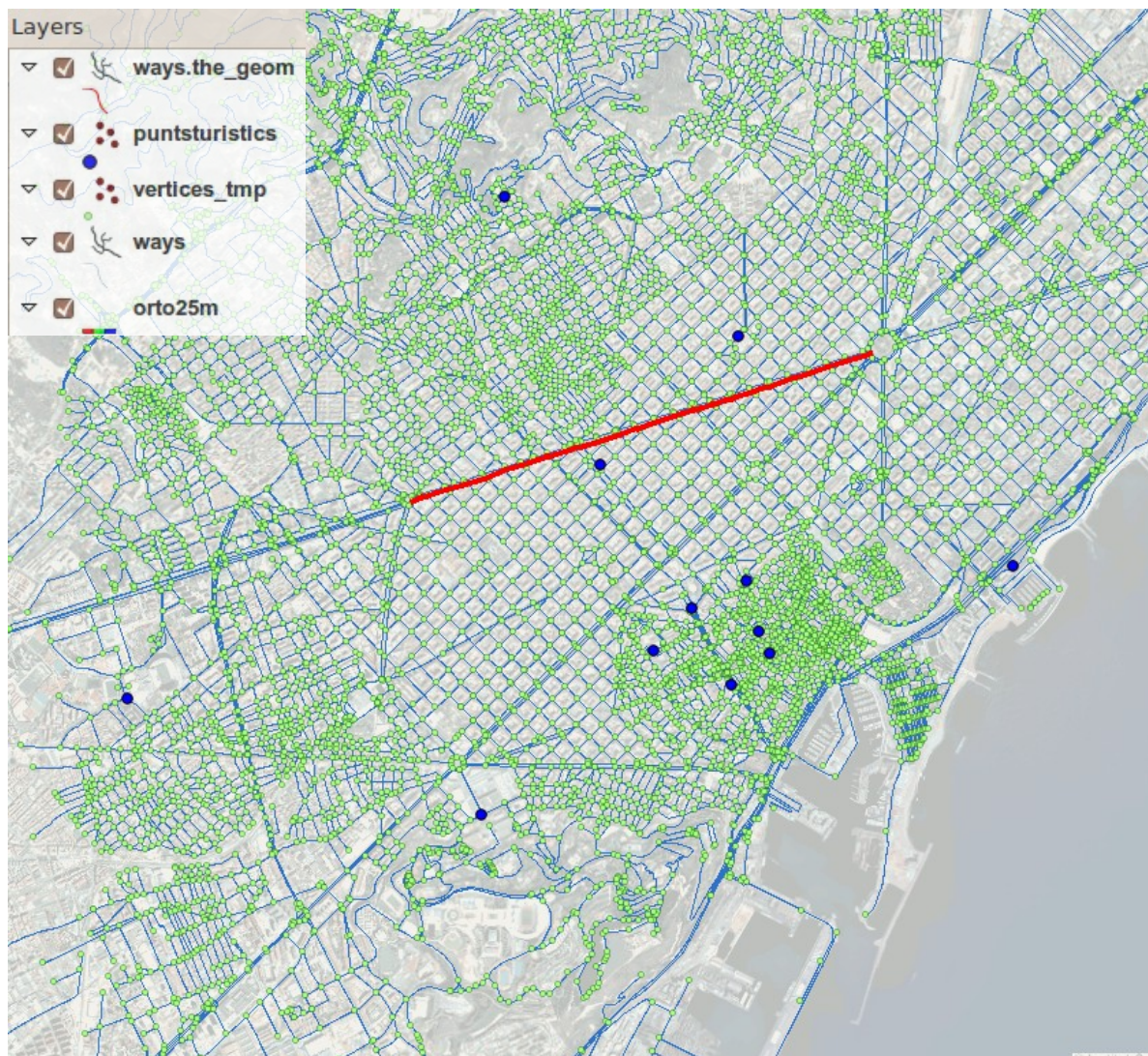
```
SELECT p.nom, (ST_X(p.the_geom)) AS coordenadaX, (ST_Y(p.the_geom)) AS coordenadaY, w.name,
       w.gid, (ST_Distance(w.the_geom, p.the_geom)) AS Distancia
FROM ways w, puntsturistics p
WHERE (ST_Distance(w.the_geom, p.the_geom)<20)
ORDER BY p.nom
```

nom	coordenadax	coordenaday	name	gid	distancia
Camp Nou	426380.104939858	4581221.87295538	Carrer d'Aristides Maillol	7883	4.46084032412384
Camp Nou	426380.104939858	4581221.87295538		4045	4.05559243756926
Camp Nou	426380.104939858	4581221.87295538	Carrer d'Aristides Maillol	7884	3.85334339708399
Catedral de la Santa Creu i Santa Eulàlia	431103.969476855	4581721.63973422	Pla de la Seu	5918	5.82127938548866
Catedral de la Santa Creu i Santa Eulàlia	431103.969476855	4581721.63973422	Pla de la Seu	5919	5.82127938548866
Catedral de la Santa Creu i Santa Eulàlia	431103.969476855	4581721.63973422		9848	1.7776092762587
Gran teatre del Liceu	430898.285390159	4581323.34656846	La Rambla	5024	10.7738189329255
Gran teatre del Liceu	430898.285390159	4581323.34656846	La Rambla	4717	3.05945794915147
La Pedrera	429919.031824374	4582971.68898467	Passeig de Gràcia	6911	17.6028209593366
Les Rambles	430603.19345539	4581897.13534466		13871	4.15584262845049
Les Rambles	430603.19345539	4581897.13534466	La Rambla	4206	18.8554525875819
Les Rambles	430603.19345539	4581897.13534466	Plaça Catalunya	4210	19.1010206021703
Les Rambles	430603.19345539	4581897.13534466	Plaça Catalunya	4696	19.1010206021703
Les Rambles	430603.19345539	4581897.13534466	Plaça Catalunya	4698	19.1010206021703
Les Rambles	430603.19345539	4581897.13534466	Plaça Catalunya	5806	19.1010206021703
Les Rambles	430603.19345539	4581897.13534466	Plaça Catalunya	5807	19.1010206021703
Les Rambles	430603.19345539	4581897.13534466		7747	9.4109761686221
Les Rambles	430603.19345539	4581897.13534466		7748	9.55811587243152
Les Rambles	430603.19345539	4581897.13534466	Carrer Pelai	12548	4.15584262845049
Les Rambles	430603.19345539	4581897.13534466	Carrer Pelai	12549	2.62470610385409
Les Rambles	430603.19345539	4581897.13534466		13870	2.80785898544473
Montjuïc	429032.269908821	4580350.84428651	Avinguda de Rius i Taulet	5032	6.84531145582965
Montjuïc	429032.269908821	4580350.84428651	Avinguda Reina Maria Cristina	3580	6.84531145582965
Montjuïc	429032.269908821	4580350.84428651	Avinguda del Marquès de Comillas	16583	4.48010341051802
Montjuïc	429032.269908821	4580350.84428651	Avinguda Reina Maria Cristina	13482	7.80641008656491
Montjuïc	429032.269908821	4580350.84428651	Avinguda de Rius i Taulet	5031	7.80641008656491
Museu d'art contemporani de Barcelona (MACBA)	430318.609048689	4581584.10409384	Plaça dels Àngels	6006	10.117858567273
Museu d'art contemporani de Barcelona (MACBA)	430318.609048689	4581584.10409384	Plaça dels Àngels	6007	10.117858567273
Museu d'art contemporani de Barcelona (MACBA)	430318.609048689	4581584.10409384	Carrer de Ferlandina	133	10.117858567273
Museu d'art contemporani de Barcelona (MACBA)	430318.609048689	4581584.10409384	Plaça dels Àngels	6004	1.29520743938369
Palau de la música catalana	431012.184738678	4582102.56874253		12074	3.63446366111791
Parc Güell	429204.656155239	4584968.61063991	Carrer d'Olot	7893	2.75970940084841
Parc Güell	429204.656155239	4584968.61063991		10334	15.7266044747061
Parc Güell	429204.656155239	4584968.61063991	Carrer de Larrard	3945	2.75970940084841
Parc Güell	429204.656155239	4584968.61063991	Carrer d'Olot	7892	1.06842988630764
Plaça de Sant Jaume	431189.560050653	4581561.09301327	Plaça Sant Jaume	5634	16.1879162433667
Port Olímpic	433007.739701851	4582211.44422019	Avenida del Litoral	15397	0.147183822500477
Temple de la Sagrada Família	430950.273612947	4583933.91868412	Provença	2975	1.85772017572257

Així doncs, mitjançant consultes a la base de dades es pot determinar quins carrers hi ha prop d'un punt turístic, o es podria fer a l'inrevés, determinar quins punts turístics hi ha prop d'un carrer o



d'un punt donat, etc,...



**Figura 4-3.** Capes dels objectes geomètrics de la base de dades PostGIS (bcnrouting).

La figura 4-3 és una captura de pantalla on es presenta de forma visual mitjançant qGIS, els objectes geomètrics de la base de dades en diferents capes. Es poden observar els nodes (vertices\_tmp) com a punts verds, els carrers (ways) com a línies blaves, els punts turístics (puntsturistics) com a punts blaus, i finalment una consulta que selecciona l'avinguda diagonal que es mostra com una línia vermella.

Per acabar, s'han realitzat dues consultes a la base de dades a partir de les coordenades d'un punt donat, ja que si més endavant s'han de realitzar càlculs de rutes a partir de punts clicats sobre un mapa mostrat amb OpenLayers, la informació que arribarà a la base de dades seran les coordenades dels punts. S'ha escollit un punt per fer proves, amb coordenades (430501,4582186) que es situa al

centre de la plaça Catalunya de Barcelona.

La primera consulta determina el vèrtex més proper a aquest punt, a partir de calcular la distància d'aquest amb tots els vèrtexs i ordenar la llista per distància creixent:

```
SELECT id, the_geom, distance(the_geom, GeometryFromText('POINT(430501 4582186)', 25831) AS
  dist
FROM vertices_tmp
ORDER BY dist;
```

id	the_geom	dist
3254	010100002031BF0D000B38C47AA6780D41C8A93105D4565341	125.70073744087
3272	010100002031BF0D000A5626E4217B0D412EAD175EFC565341	128.054770066677
1835	010100002031BF0D00038FEB74FD720D41B64C9AF4F6565341	141.389511068404
1836	010100002031BF0D0008275E4CCD0760D41605E6D0A16575341	146.40060959906
3265	010100002031BF0D0004D98E2679B720D4188D1C0B7FD565341	159.491128041806
3267	010100002031BF0D00061F45C9E8C770D41BEA6B7A019575341	159.826794463391
...	...	...

Si limitem aquesta consulta a un sol resultat (LIMIT 1) tindrem el vèrtex més proper al punt.

La segona consulta ordena els punts turístics de la base de dades segons la distància al punt donat considerant distàncies lineals (no carrers):

```
select gid, nom, descripcio, the_geom, distance(the_geom, GeometryFromText('POINT(430501
  4582186)', 25831) AS dist
FROM puntsturistics
ORDER BY dist;
```

gid	nom	descripcio	the_geom	dist
2	Les Rambles	Passeig emblemàtic de...	010100002031...	418.672394477203
8	Palau de la mú....	Auditòri de música de...	010100002031...	723.856369519325
9	Museu d'art co...	Museu dedicat a l'exp...	010100002031...	826.849934265769
7	Catedral de la...	Catedral de Barcelona...	010100002031...	1040.35947603141
11	Plaça de Sant...	Cor administratiu de...	010100002031...	1263.66611590549
12	Gran teatre de...	Reconegut teatre d'òp...	010100002031...	1279.82534447645
3	La Pedrera	Edifici modernista de...	010100002031...	1287.64938530447
1	Temple de la S...	Església catòlica de...	010100002031...	2424.55972505542
5	Montjuïc	Font màgica, un dels...	010100002031...	3111.41779433411
10	Port Olímpic	Port esportiu de Barc,..	010100002031...	3372.95154641687
4	Parc Güell	Gran jardí amb elemen...	010100002031...	4089.40348302305
6	Camp Nou	Camp del futbol club...	010100002031...	5600.26363447478

(12 rows)

En els dos casos es pot observar que s'ha convertit el punt en objecte geomètric a partir de l'expressió del punt amb notació WKT (*Well Known Text*) i la funció *GeometryFromText()*.



## Capítol 5:

# Càlcul de rutes amb pgRouting

En aquest capítol es descriurà com es poden utilitzar els algoritmes pel càlcul de rutes proporcionats per l'extensió pgRouting sobre una base de dades postGIS.

### 5.1 Introducció

PgRouting és una extensió per bases de dades espacials PostGIS/PostgreSQL que proporciona algoritmes per el càlcul de rutes sobre les dades mitjançant consultes SQL a la base de dades espacial. Els algoritmes són els següents:

- Algoritme de Dijkstra: Algoritme per la determinació del camí més curt des d'un node d'origen a la resta de nodes en un graf dirigit amb pesos a les arestes. La seva complexitat és quadràtica  $O(n^2)$  essent  $n$  el nombre de nodes.
- Algoritme A-Star: Algoritme per la determinació del camí més curt entre un node objectiu i un de destinació basat en heurística i utilitzant una funció d'avaluació. La seva complexitat depèn de la funció d'avaluació i necessita grans quantitats de memòria per emmagatzemar tots els possibles següents nodes per cada estat. Respecte l'algoritme de Dijkstra afegeix informació geogràfica a cada aresta permetent escollir arestes més properes a la destinació.
- Algoritme Shooting-Star: És un algoritme per la determinació del camí més curt, semblant a l'A-Star però que busca camins on l'origen i el destí són arestes i no nodes.

A més, inclou eines per la importació de dades d'OpenStreetMap, com `osm2pgrouting`,<sup>16</sup> de forma que les dota a les dades de la topologia necessària pel càlcul de rutes. Com s'ha descrit anteriorment (apartat 4.3.1), aquesta és l'eina que s'ha utilitzat per inserir les dades d'OpenStreetMap a la base de dades PostGIS d'aquest treball.

## 5.2 Preparació de les dades

La preparació de les dades per aplicar els algorismes de cerca és lleugerament diferent segons l'algorisme aplicat. En aquest treball s'han preparat les dades per l'ús de l'algorisme de Dijkstra per calcular la ruta de menor cost entre dos punts donats de la ciutat de Barcelona. S'ha escollit aquest algorisme perquè la finalitat de l'aplicació final és trobar la ruta de **mínima distància** entre dos punts per tal de desplaçar-s'hi a peu. En aquest sentit, no hi ha necessitat de considerar altres factors (ús de funcions d'avaluació), ja que la distància és el factor determinant. Al considerar la xarxa viària de Barcelona com un graf, on les longituds dels carrers són els pesos de les arestes, l'algorisme de Dijkstra aportarà una solució adequada i ràpida a les nostres necessitats.

Per tal de poder buscar el camí més curt entre dos punts, és necessari que totes les carrers de Barcelona (taula *ways*) tinguin assignats uns nodes d'origen (*source*) i de destinació (*target*). Amb la importació de les dades mitjançant `osm2pgrouting`, aquesta assignació es realitza de forma automàtica, però en tot cas `pgRouting` proporciona la funció `assign_vertex_id()` que assigna identificadors a cada aresta i pot col·lapsar vèrtexs propers a partir d'una tolerància passada com a paràmetre. La sintaxi d'aquesta funció és:

```
assign_vertex_id('ways', tolerància, 'the_geom', 'gid');
```

S'especifica la taula de les arestes (*ways*), la columna geomètrica (*the\_geom*), la columna que conté l'index espacial de la taula (*gid*), i el valor *tolerància*, que és un valor numèric de tipus *double* que farà que s'assigni el mateix identificador a vèrtexs separats per una distància menor que aquesta tolerància. Cal tenir en compte que les unitats de distància per aquesta tolerància seran les del sistema de referència de les dades geogràfiques de la taula.

Per tal d'accelerar l'accés a les dades, és convenient afegir índexs a aquestes columnes d'origen i destí, així com a la columna geomètrica. A la columna geomètrica s'afegeix un índex de tipus GiST (arbre de cerca generalitzada, explicat a l'apartat 3.2) que permet la indexació d'aquest tipus de

dades. Això es realitza mitjançant les sentències SQL següents:

```
CREATE INDEX source_idx ON ways(source);
CREATE INDEX target_idx ON ways(target);
CREATE INDEX geom_idx ON ways USING GIST(the_geom GIST_GEOMETRY_OPS);
```

Fins ara, la base de dades del treball que conté els carrers de Barcelona (bcnrouting) conté les següents taules:

```

                                List of relations
 Schema |          Name          | Type  | Owner
-----+-----+-----+-----
 public | classes                | table | oriol
 public | geography_columns      | view  | oriol
 public | geometry_columns       | table | oriol
 public | nodes                  | table | oriol
 public | puntsturistics         | table | oriol
 public | puntsturistics_gid_seq | sequence | oriol
 public | spatial_ref_sys        | table | oriol
 public | types                  | table | oriol
 public | vertices_tmp           | table | oriol
 public | vertices_tmp_id_seq    | sequence | oriol
 public | ways                   | table | oriol
(11 rows)
```

i la seva taula *ways* conté les següents columnes i índexs:

```

                                Table "public.ways"
 Column |          Type          | Modifiers
-----+-----+-----+-----
 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                |
```

Indexes:

```

"ways_pkey" PRIMARY KEY, btree (gid)
"geom_idx" gist (the_geom)
"source_idx" btree (source)
"target_idx" btree (target)
```

Check constraints:

```
"enforce_dims_the_geom" CHECK (st_ndims(the_geom) = 2)
"enforce_geotype_the_geom" CHECK (geometrytype(the_geom) =
    'MULTILINESTRING'::text OR the_geom IS NULL)
"enforce_srid_the_geom" CHECK (srid(the_geom) = 25831)
```

A l'haver importat les dades mitjançant l'eina `osm2pgrouting`, a la taula `ways` hi ha més columnes de les estrictament necessàries per l'enrutament, que serien l'identificador únic (*gid*), l'identificador del node d'origen (*source*), del node destí (*target*), la longitud del tram (*length*), i la geometria (*the\_geom*). Al nostre cas, tenim a més columnes que identifiquen el tipus de via (*class\_id*), les coordenades dels vèrtexs d'origen i destí (*x1*, *y1*, *x2*, *y2*) i columnes per implementar costos addicionals (*to\_cost*, *reverse\_cost*). En cas que no hi hagi columna per la longitud del tram, que estigui buida, o que no contingui valors correctes (per exemple, unitats diferents per un canvi de SRID), es pot actualitzar a partir d'aplicar la funció `length()` a la geometria:

```
UPDATE ways SET length = length(the_geom);
```

En aquest punt ja s'ha afegit topologia a la xarxa de carrers de Barcelona de la base de dades PostGIS. Amb la funció utilitzada, `assign_vertex_id()`, es crea un graf no dirigit, que no tindrà en compte si hi ha carrers d'un únic sentit. En el nostre cas, això no és un problema ja que volem calcular rutes cap a punts turístics de la ciutat de Barcelona des de la pròpia ciutat, suposant que el “turista” en qüestió es mourà a peu. Per implementar vies d'un únic sentit es poden utilitzar les columnes `to_cost` i `reverse_cost` (que se'ns han generat automàticament amb `osm2pgrouting`) assignant valors segons els sentits dels carrers. Aleshores, els algorismes de cerca de camins mínims que proporciona `pgRouting` disposen de la opció de considerar o no aquests costos inversos.

### 5.3 Càlcul de rutes mitjançant l'algoritme de Dijkstra

Amb les dades dels carrers de la ciutat de Barcelona dotades de la tipologia necessària, s'han realitzat proves de funcionament del càlcul de rutes mitjançant l'algoritme de Dijkstra. La funció principal per l'aplicació de l'algoritme de Dijkstra sobre la base de dades que proporciona `pgRouting` és la següent:

```
shortest_path(sentència_sql text, source_id integer, target_id integer, directed boolean,
    has_reverse_cost boolean)
```

A partir de la `sentència_sql` es seleccionen les files sobre les quals es realitzarà el recorregut

(arestes), els `source_id` i `target_id` indiquen els nodes d'origen i destí del recorregut i els paràmetres `directed` i `has_reverse_cost` assenyalen si s'han de tenir en compte els costos addicionals per implementar, per exemple, vies d'un únic sentit.

Per comprovar el funcionament d'aquest algoritme sobre les dades dels carrers de Barcelona, cal seleccionar a partir de la sentència SQL l'identificador únic de les files corresponents als carrers i que indi s'indiquin dos nodes per a realitzar el recorregut. Els dos nodes seleccionats per realitzar proves són el 3254 i el 512, que corresponen a nodes propers a la plaça Catalunya i a la Sagrada Família respectivament. La sentència i el resultat són els següents:

```
SELECT * FROM shortest_path('SELECT gid as id, source, target, length as cost FROM ways', 3254, 512, false, false);
```

vertex_id	edge_id	cost
3254	4208	122.066434327612
3253	4240	76.6033740849193
3272	4239	122.824532541455
3267	4233	15.380925094384
3268	4234	16.5709267563531
3269	5324	167.990171339864
1232	5325	132.341554583836
1259	5326	136.935790553466
3840	5327	130.695298716294
3841	5328	2.53865437742274
3842	5329	134.265934108649
641	5330	69.5068514853566
3636	4984	75.9540948341959
3635	6076	91.736505415882
544	631	10.9893238576156
543	630	129.308130865411
542	629	131.20780579412
541	706	8.64640445241738
612	705	122.907318813098
494	704	131.296411935767
611	703	102.667301105092
610	702	9.29761490503097
609	701	12.1175680889242
608	778	10.1886208229633
674	779	14.7370671252102
675	780	113.630500368596
676	1470	136.723264901365
736	846	130.031808296655
514	598	74.7537430434115
513	597	57.9480913544213
512	-1	0

(31 rows)

S'observa que la sortida correspon a totes les arestes que porten del node d'origen (3254) al de destí (512) i la longitud de cada tram (cost). Per tal de mostrar aquesta ruta de forma més gràfica, pgRouting proporciona funcions addicionals pel càlcul de camins mínims mitjançant l'algoritme de Dijkstra que utilitzen la mateixa funció anterior, però permeten seleccionar els elements de retorn amb la sentència SQL. D'aquesta forma es poden obtenir les arestes com a elements geomètrics i no cal que forçosament es retorni una columna pel cost. Aquestes funcions són:

```
dijkstra_sp(taula text, source_id integer, target_id integer)
dijkstra_sp_delta(taula text, source_id integer, target_id integer, bbox_buffer double)
```

Aquestes dues funcions utilitzen la funció *shortest\_path()* sobre la taula *taula* amb els nodes d'origen (*source\_id*) i destí (*target\_id*) especificats. Es diferencien entre elles pel fet que la segona limita la seva cerca en l'àrea limitada pels nodes d'origen i destí més una distància addicional especificada al *bbox\_buffer*. Pel fet que aquestes dues funcions no proporcionen càlculs addicionals i necessiten la funció *shortest\_path()* se les considera *wrapper functions*.

Si s'aplica la funció *dijkstra\_sp()* a la base de dades de treball de forma que retorni la columna geomètrica de les arestes (carrers) del camí mínim entre els dos nodes s'obté el següent:

```
SELECT gid, the_geom FROM dijkstra_sp('ways', 3254, 512);
gid | the_geom
-----|-----
4240 | 0105000020E764000001000000010200000002000000BA26C50796471A41BE...
4233 | 0105000020E7640000010000000102000000020000009C535DEE40461A413C...
5325 | 0105000020E7640000010000000102000000020000008ACD7CFE17481A412F...
5328 | 0105000020E764000001000000010200000002000000D5ED1D22964C1A41CC...
5329 | 0105000020E764000001000000010200000002000000D86F0B739E4C1A41C0...
629 | 0105000020E76400000100000001020000000200000015C2126BEE501A417B...
5326 | 0105000020E7640000010000000102000000020000003299CD9D92491A4196...
631 | 0105000020E764000001000000010200000002000000A39095190A4E1A41E6...
702 | 0105000020E7640000010000000102000000020000002AEC8A77DE4C1A4145...
846 | 0105000020E764000001000000010200000002000000186007FC194E1A4109...
4208 | 0105000020E76400000100000001020000000200000047B0A3474E481A415C...
5330 | 0105000020E764000001000000010200000002000000ACBE5DF194E1A41A7...
5324 | 0105000020E764000001000000010200000002000000CCF98A393E461A41C7...
5327 | 0105000020E7640000010000000102000000020000009CE9ACFD1A4B1A414E...
706 | 0105000020E7640000010000000102000000020000009DAB8B6D6501A41F3...
701 | 0105000020E7640000010000000102000000020000006242569FBC4C1A4189...
780 | 0105000020E764000001000000010200000002000000FA743ECA014D1A4109...
598 | 0105000020E76400000100000001020000000200000088DB30E24A4D1A4181...
4234 | 0105000020E764000001000000010200000002000000AA71FAF14A461A416E...
4984 | 0105000020E764000001000000010200000004000000D5A7686ACA4E1A4183...
778 | 0105000020E7640000010000000102000000020000006242569FBC4C1A4189...
630 | 0105000020E764000001000000010200000002000000816387D2724F1A41B0...
705 | 0105000020E76400000100000001020000000200000091458EF5854F1A4178...
```

```

6076 | 0105000020E764000001000000010200000003000000C033B966EB4D1A41D...
704 | 0105000020E764000001000000010200000002000000163B0938174E1A415C...
703 | 0105000020E76400000100000001020000000200000020460B6EF84C1A4150...
779 | 0105000020E764000001000000010200000002000000CD4ECA02DB4C1A4193...
1470 | 0105000020E76400000100000001020000000200000046D2CCF52C4E1A4156...
(30 rows)

```

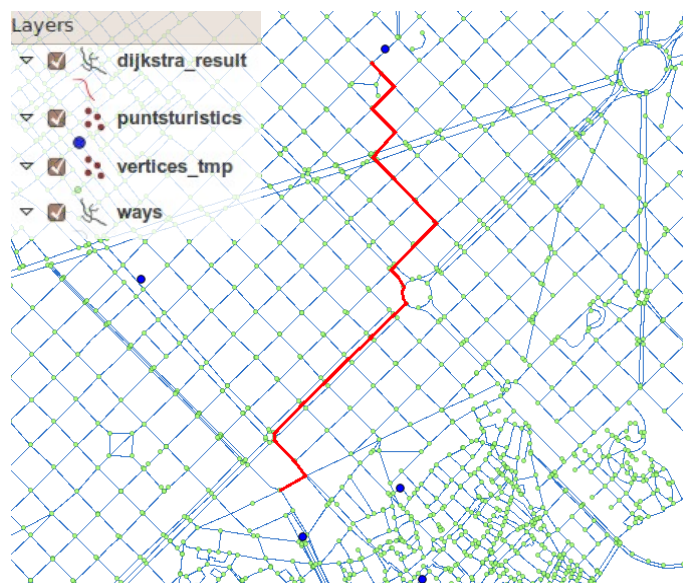
Aquestes dades es poden guardar en una nova taula resultat amb una columna geomètrica, i d'aquesta forma poder visualitzar la ruta resultat mitjançant qualsevol SIG d'escriptori o com una capa de MapServer. Per tal de dur a terme això s'utilitzen les següents sentències SQL:

```

DROP TABLE IF EXISTS dijkstra_result;
CREATE TABLE dijkstra_result(gid int4) with oids;
SELECT AddGeometryColumn('public', 'dijkstra_result', 'the_geom', '25831', 'MULTILINESTRING', 2);
INSERT INTO dijkstra_result(the_geom)
SELECT the_geom FROM dijkstra_sp('ways', 3254, 512)

```

Mitjançant aquestes sentències, primer s'eliminen possibles taules de resultats anteriors, i seguidament, es creen la taula pel resultat (*dijkstra\_result*) i la columna geomètrica per emmagatzemar les vies resultat, de tipus MULTILINESTRING i en el sistema de referència que es troben les dades (25831). Finalment, s'omple la columna geomètrica amb el resultat d'aplicar l'algoritme de Dijkstra sobre els carrers de la ciutat de la taula *ways* seleccionant com a sortida la columna geomètrica. Després d'aquest procediment s'ha generat la taula *Dijkstra\_result* amb la ruta més curta entre els nodes 3254 i 512, calculada mitjançant l'algoritme de Dijkstra i emmagatzemada com a geometria. La figura 5-1 mostra el resultat mitjançant el SIG d'escriptori qGIS.



**Figura 5-1.** Resultat del càlcul de la ruta més curta entre els nodes 3254 (plaça Catalunya) i 512 (Sagrada Família). Visualització mitjançant el SIG d'escriptori qGIS.

## Capítol 6:

# Entorn web amb OpenLayers

En aquest capítol es descriurà com donar una utilitat pràctica als aspectes més tècnics tractats als capítols 4 i 5. Fent ús de la llibreria OpenLayers es presentarà el mapa de la ciutat de Barcelona en un entorn web, de forma dinàmica, sobre el que es seleccionaran el punt d'origen i l'atracció turística per tal de calcular la ruta òptima entre aquests dos punts.

### 6.1 Funcionament d'OpenLayers

OpenLayers<sup>15</sup> és una llibreria Javascript que proporciona una interfície de programació d'aplicacions (API) per tal de mostrar mapes funcionals en els navegadors web.

El seu funcionament es basa en la creació d'un objecte anomenat **map** que guarda la informació respecte la projecció utilitzada, l'extensió del mapa, les unitats, la resolució, el nombre d'ampliacions possibles, etc..., és a dir, els paràmetres generals del mapa mostrat. Aleshores, sobre aquest objecte map, la informació es mostra a través de **layers** (capes). A cada capa s'especifica l'origen de la informació (d'on prové) i com s'ha de llegir aquesta informació (tipus d'informació).

Els tipus de capes amb què pot treballar OpenLayers és molt variat. En aquest treball s'han utilitzat capes de tipus WMS (per la connexió amb MapServer) i de tipus OSM (per la connexió amb OpenStreetMap).

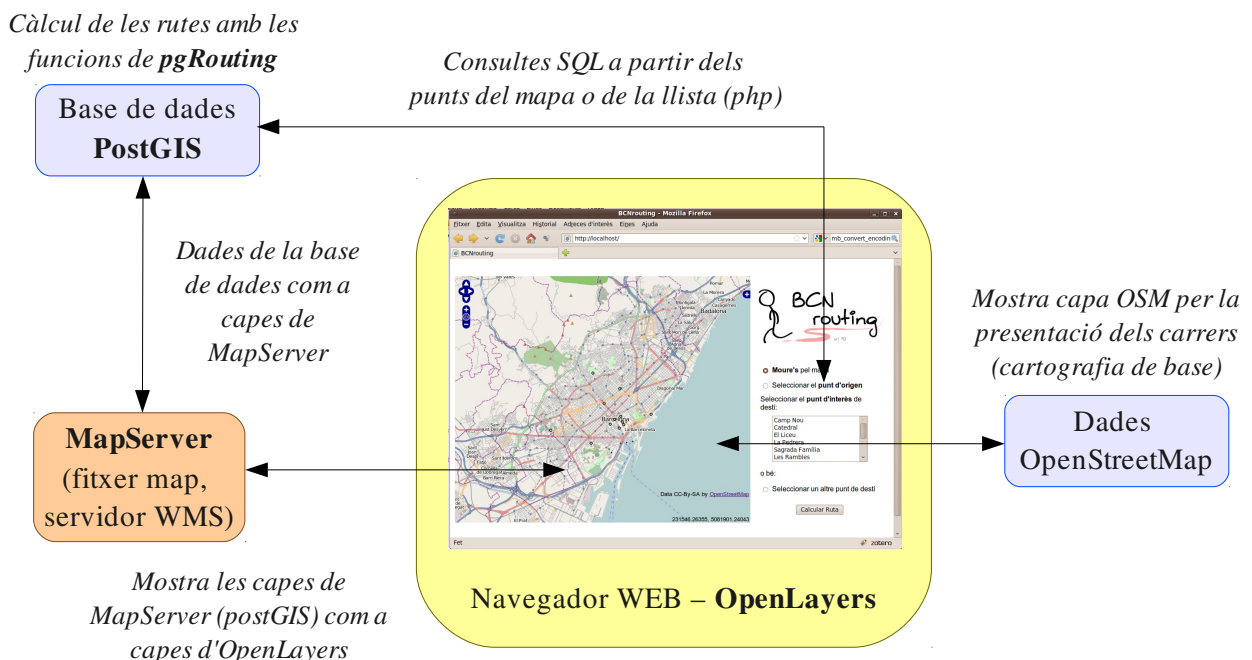
En el fitxer HTML de la pàgina es col·loca l'*script* corresponent (normalment a la capçalera), on



es crea l'objecte map, amb les opcions corresponents, i s'hi afegeixen les capes. La creació del mapa inicial s'acostuma a realitzar dins d'una funció que es crida al carregar la pàgina. A aquest *script* s'hi poden incloure funcions addicionals per tal d'implementar les accions sobre el mapa. El mapa, es dibuixa a la pàgina en una secció (<div>) amb el nom de map.

## 6.2 Connexió entre PostGIS, UMN MapServer i OpenLayers

Per poder mostrar en un entorn web el mapa de la ciutat de Barcelona amb la funcionalitat del càlcul de rutes, cal que OpenLayers accedeixi als carrers emmagatzemats a la base de dades postGIS, hi realitzi consultes, i mostri la solució al mapa. Hi ha diverses formes d'implementar aquesta comunicació i la figura 6-1 mostra com s'ha realitzat en aquest treball.



**Figura 6-1.** Comunicació entre OpenLayers i les diferents fonts de dades per mostrar el mapa i la ruta.

El procediment seguit és el següent:

- El càlcul de rutes es realitza mitjançant consultes a la base de dades (utilitzant les funcions que s'han tractat a l'apartat 5.3) i el resultat del camí més curt s'emmagatzema en un fitxer XML com una llista d'arestes expressades en format WKT (*Well Known Text*). Aquestes

arestes són interpretades per OpenLayers que mostrarà la ruta com una capa vectorial.

- MapServer actua de servidor WMS. A partir d'un fitxer *map* s'especifiquen les capes corresponents a les dades emmagatzemades a les taules de la base de dades PostGIS. Es consideren les taules de carrers (*ways*) i la dels punts turístics (*puntsturistics*). La capa dels carrers s'ha utilitzat únicament per comprovar la correcta projecció de les dades i la dels punts turístics es mostra per defecte superposada al mapa.
- OpenLayers mostra el mapa amb les capes següents:
  - Dues capes de tipus WMS servides per MapServer, que corresponen als punts turístics i als carrers de la ciutat.
  - Una capa de tipus OSM, provinent del servidor d'OpenStreetMap, que mostra els carrers de la ciutat de Barcelona i fa de cartografia de base, per tal de referenciar la posició en el territori i obtenir un aspecte visual adient per la selecció dels punts sobre el mapa.
  - Quatre capes de tipus Vector, que serveixen per mostrar els punts seleccionats al mapa i la ruta resultat. Els punts seleccionats al mapa corresponen al punt d'origen i al punt de destí (que s'hi dediquen dues capes, segons si es selecciona de la llista de punts turístics o si es selecciona un altre punt de destí qualsevol).
- Per el càlcul de rutes, es seleccionen els punts d'origen i destí (d'una llista o clicant sobre el mapa), i seguidament es clica sobre el botó de càlcul de ruta. Amb aquesta acció es realitza una connexió a la base de dades, mitjançant el llenguatge *php*, i s'hi realitzen les consultes corresponents pel càlcul de rutes utilitzant les coordenades dels punts que s'han marcat sobre el mapa i les funcions de pgRouting. La geometria de la ruta resultant és emmagatzemada en format WKT i es retorna com un fitxer XML a OpenLayers, que l'interpreta i dibuixa la ruta en una capa vectorial.
- A més de la ruta resultat, l'aplicació web rep més informació a partir de consultes a la base de dades mitjançant *php* i la tècnica AJAX (*Asynchronous Javascript and XML*), com ara les coordenades del punt turístic seleccionat de la llista, la descripció del punt turístic, o la longitud total de la ruta.

Segurament aquesta no és la única solució per realitzar consultes a la base de dades i mostrar la solució amb OpenLayers. En el desenvolupament d'aquest treball, i abans d'adoptar aquesta solució, s'han considerat i provat altres metodologies per la implementació del càlcul de rutes:

- La realització del càlcul mitjançant igualment una connexió a la base de dades amb *php* però utilitzant la taula *dijkstra\_result*, tal com s'han realitzat les proves pel càlcul de rutes a l'apartat 5.3 d'aquest treball (figura 5-1). D'aquesta forma connexió a la base de dades mitjançant *php* és d'un únic sentit, ja que la consulta crea (o modifica) la taula *dijkstra\_result* de la base de dades amb la ruta solució pels punts d'inici i final marcats a OpenLayers. Per mostrar la ruta sobre el mapa, OpenLayers activa una capa corresponent a la taula *dijkstra\_result* que obté de la base de dades mitjançant MapServer. Aquesta metodologia de càlcul i presentació de la ruta ha funcionat perfectament i integrava totes les eines utilitzades (PostGIS, MapServer, pgRouting i OpenLayers), però finalment s'ha decidit deixar de banda perquè presenta una limitació important: pot donar problemes si s'utilitza l'aplicació per més d'un usuari alhora, ja que es modifica una taula de la base de dades per calcular cada ruta.
- També s'ha considerat la possibilitat de realitzar les consultes a la base de dades mitjançant MapServer, ja que en un fitxer *map* es poden definir capes de tipus PostGIS que continguin el resultat d'una consulta SQL. Aquesta possibilitat s'ha desestimat pel fet que la connexió entre OpenLayers i MapServer és a través de WMS, sense la possibilitat de passar com a paràmetres les consultes SQL, i caldria regenerar el fitxer *map* per cada consulta.

Així doncs, finalment s'ha realitzat el càlcul i mostrat la ruta tal com s'ha explicat anteriorment: consulta SQL mitjançant *php*, retorn de la ruta resultat en format WKT i interpretació d'OpenLayers per mostrar-la com a capa vectorial. En aquesta metodologia utilitzada, MapServer queda totalment de banda en el càlcul de rutes, i només s'utilitza per la comprovació del correcte solapament dels carrers de la base de dades amb la capa base, i per mostrar els punts turístics. Aquesta solució adoptada ha estat basada en la que es troba descrita a la bibliografia en forma de *workshops*.<sup>24</sup>

### 6.3 Preparació de les dades

Un aspecte important a tenir en compte pel correcte funcionament de l'aplicació amb OpenLayers són els sistemes de referència de les dades a mostrar al mapa. OpenLayers mostra el mapa en un sistema de referència projectat, que correspon al de la capa base, i cal que sigui el mateix per totes les capes superposades.

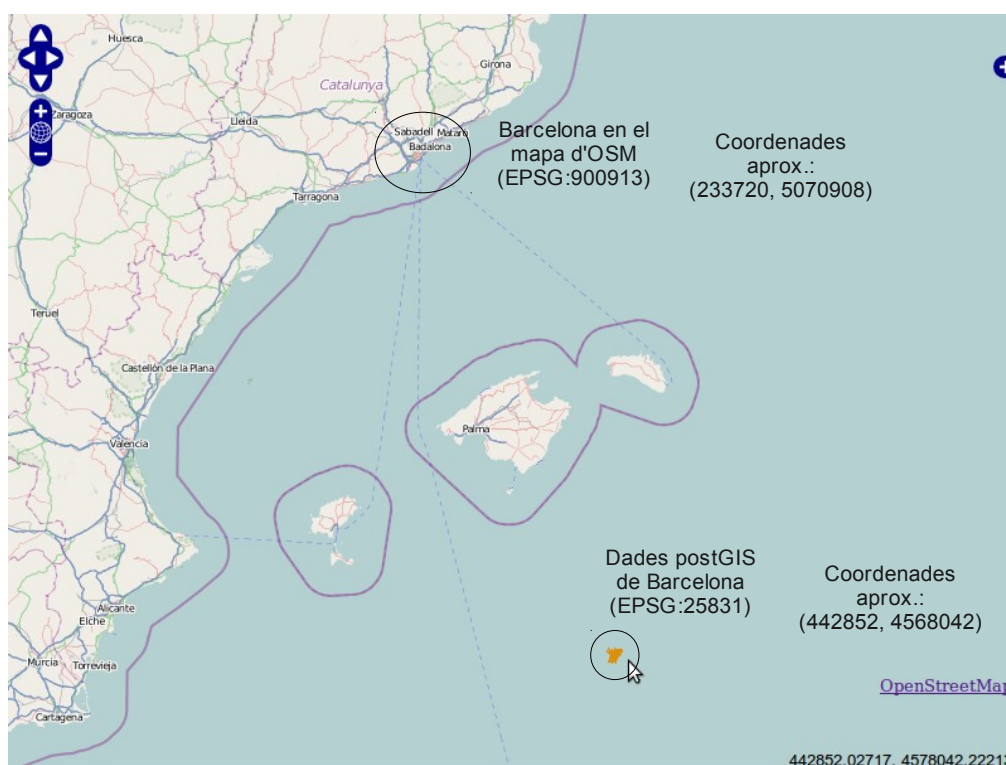


**Figura 6-2.** Mapa d'OpenLayers amb les capes de les taules *ways* i *puntsturistics* de la base de dades, proporcionades mitjançant MapServer (WMS) amb el EPSG:25831, que es pot comprovar amb les coordenades a peu de mapa de la posició senyalada pel cursor.

Les dades de la base de dades del treball (*bcnrouting*) es troben en el sistema de coordenades projectades EPSG:25831, després de la conversió realitzada a l'apartat 4.3.4 del treball. S'ha generat un fitxer *map* amb capes per es taules *ways* i *puntsturistics* de la base de dades i s'ha mostrat el mapa corresponent amb OpenLayers amb la projecció de les dades EPSG:25831 sense problemes (figura 6-2).

El problema ha esdevingut a l'utilitzar la capa d'OpenStreetMap com a capa principal (*base layer*), ja que la projecció que utilitza OpenLayers per aquest tipus de capa és la EPSG:900913, una projecció de Mercator (*SphericalMercator*), coneguda com a *Google projection*. Els intents

d'utilitzar la projecció EPSG:25831 com a projecció principal al mapa d'OpenLayers per mostrar les capes de la base de dades juntament amb la d'OSM no han resultat satisfactoris, ja que a la majoria de cassos s'ha obtingut un missatge d'error indicant que les dades de les diferents capes no es trobaven en la mateixa projecció. En els cassos que s'ha aconseguit ajuntar els dos tipus de capes ha aparegut un desplaçament entre les capes (figura 6-3), i tot i especificar a les opcions del mapa la projecció EPSG:25831, les dades d'OSM es projecten amb l'EPSG:900913 segons es pot veure en les coordenades de la figura.



**Figura 6-3.** Mapa d'OpenLayers on es mostra el desplaçament entre les capes d'OSM i la WMS (MapServer/PostGIS) a l'utilitzar l'EPSG:25831 com a projecció del mapa. Les coordenades d'OSM corresponen a les de la ciutat de Barcelona amb l'EPSG:900913.

Després de diverses proves, on s'ha provat d'utilitzar també una projecció EPSG:4326 amb els mateixos problemes, s'ha optat per canviar el sistema de referència de les dades de la base de dades PostGIS, seguint el mateix procediment que a l'apartat 4.3.4, especificant el SRID d'origen 25831 i el de destí 900913. Per realitzar aquesta transformació entre sistemes de referència i per poder especificar-lo en un fitxer *map* posteriorment, és necessari afegir la descripció d'aquesta projecció a la llibreria *proj4*, que és la que utilitzen tant postGIS com MapServer per realitzar transformacions i reprojeccions. El procés per afegir aquesta projecció a la llibreria està àmpliament documentat, fins i tot a la mateixa documentació d'OpenLayers,<sup>25</sup> o als *workshops*<sup>24</sup> que s'havien comentat

anteriorment. Cal afegir al fitxer `/usr/share/epsg/proj` la definició següent:

```
<900913> +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +no_defs
```

I a la base de dades PostGIS cal afegir la definició del sistema a la taula `spatial_ref_sys`:

```
INSERT INTO spatial_ref_sys (srid, auth_name, auth_srid, srtext, proj4text) VALUES (900913, 'EPSG', 900913,
'PROJCS["unnamed",GEOGCS["unnamed ellipse",DATUM["unknown",SPHEROID["unnamed",6378137,0]],
PRIMEM["Greenwich",0],UNIT["degree",0.0174532925199433]],
PROJECTION["Mercator_2SP"],PARAMETER["standard_parallel_1",0],PARAMETER["central_meridian",0],PARAMETER["false_easting",0],
PARAMETER["false_northing",0],UNIT["Meter",1],EXTENSION["PROJ4","+proj=merc +a=6378137 +b=6378137
+lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +wktext +no_defs"]]',
'+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +wktext +no_defs');
```

Posteriorment, s'ha generat el fitxer `map` per accedir a les dades transformades de la base de dades mitjançant MapServer i s'ha realitzat la connexió amb OpenLayers, observant ara sí el correcte solapament de les dades de les diferents capes (figura 6-4). Per això ha estat necessari especificar en el fitxer `map` la projecció general que ha d'utilitzar per servir el mapa de la forma:

```
PROJECTION
  'init=epsg:900913'
END
```

I per cada capa realitzar la connexió a la base de dades (en aquest exemple `bcnrouting3` que conté les dades amb el SRID 900913) indicant també el SRID de les dades:

```
LAYER
  NAME 'ways'
  ...
  CONNECTIONTYPE postgis
  CONNECTION "dbname='bcnrouting3' host=127.0.0.1 port=5432 user='oriol' password='origis' sslmode=disable"
  DATA 'the_geom FROM "ways" USING UNIQUE gid USING srid=900913'
  ...

LAYER
  NAME 'puntsturistics'
  ...
  CONNECTIONTYPE postgis
  CONNECTION "dbname='bcnrouting3' host=127.0.0.1 port=5432 user='oriol' password='origis' sslmode=disable"
  DATA 'the_geom FROM "puntsturistics" USING UNIQUE gid USING srid=900913'
  ...
```

Finalment, en el fitxer HTML on es troba l'`script` per OpenLayers, s'especifica com a projecció del mapa la EPSG:900913 i s'addicionen les capes de forma normal:

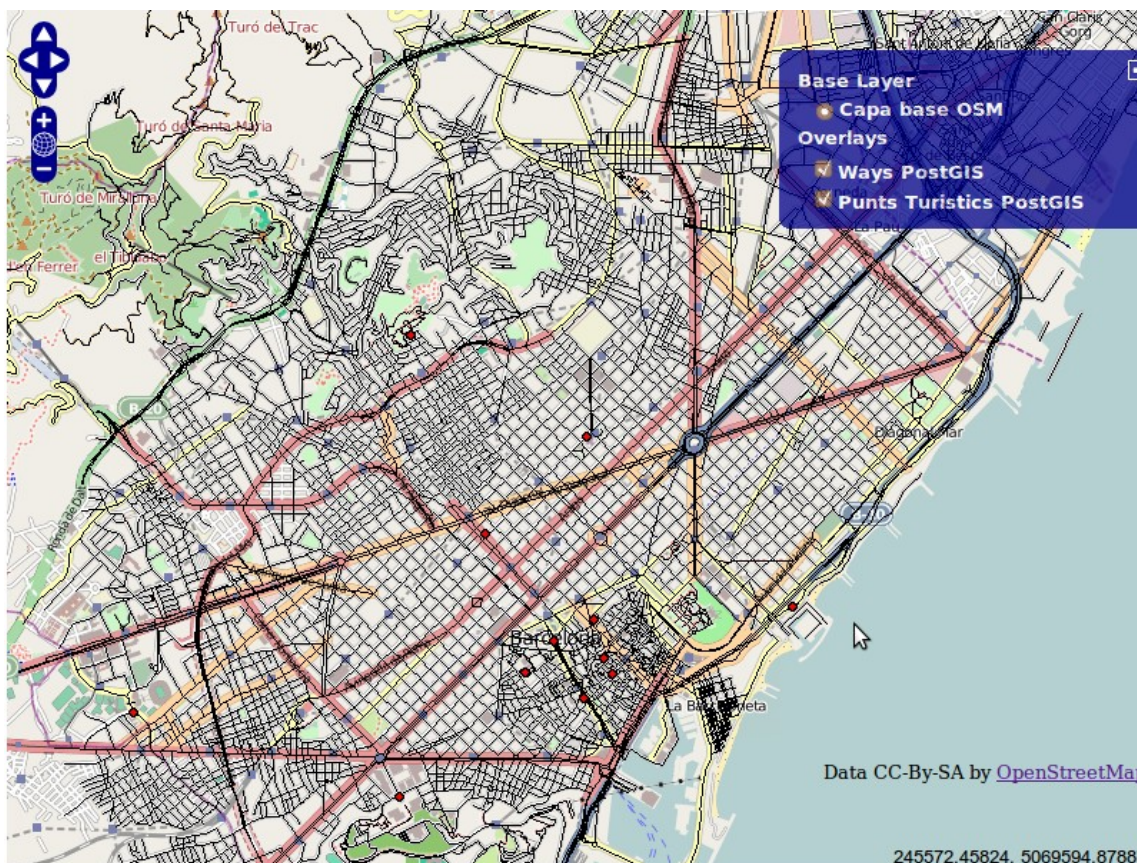
```
// S'afegeix el mapa
var options = { projection: new OpenLayers.Projection("EPSG:900913")}
map = new OpenLayers.Map('map', options);
// Capa OSM
var mapnik = new OpenLayers.Layer.OSM("Capa base OSM");
// Capes WMS de la base de dades via MapServer
var wms = new OpenLayers.Layer.WMS("Ways PostGIS", "http://localhost/cgi-bin/mapserv?
```



```

map=/var/www/BCNrouting/BCNrouting.map", {layers: 'ways', transparent:'true'} );
wms.setVisibility(false);
var puntswms = new OpenLayers.Layer.WMS( "Punts Turistics PostGIS", "http://localhost/cgi-bin/mapserv?
map=/var/www/BCNrouting/BCNrouting.map", {layers: 'puntsturistics', transparent:'true'} );
// S'afegeixen les capes
map.addLayers([mapnik, wms, puntswms]);

```



**Figura 6-4.** Mapa d'OpenLayers on es mostra el correcte solapament entre la capa base d'OSM i les capes MapServer/PostGIS a l'utilitzar l'EPSG:900913 com a projecció del mapa.

A l'annex del treball s'hi poden trobar els fitxers *map* i HTML complets per aquests exemples.

## 6.4 Entorn i càlcul de rutes *via web*

Pel càlcul de rutes s'ha partit del mapa d'OpenLayers anterior (figura 6-4) i s'ha dissenyat una pàgina amb tres botons d'opció (*radio buttons*), una llista dels punts turístics (*listbox*) i un botó. Els tres botons d'opció activen les opcions de moure's a través del mapa, de seleccionar el punt d'origen i de seleccionar un punt de destí (permet seleccionar un punt qualsevol al mapa en el cas que no es vulgui utilitzar la llista de punts turístics). La llista conté els punts turístics de la base de dades i permet seleccionar-ne un com a punt de destí. Al seleccionar-lo es marca sobre el mapa i apareix la

descripció del punt d'interès a sota el mapa. Tant les coordenades per marcar el punt al mapa com la descripció s'obtenen a partir de consultes a la base de dades. El botó realitza el càlcul de la ruta entre els punts d'inici i destí marcats al mapa (figura 6-5).

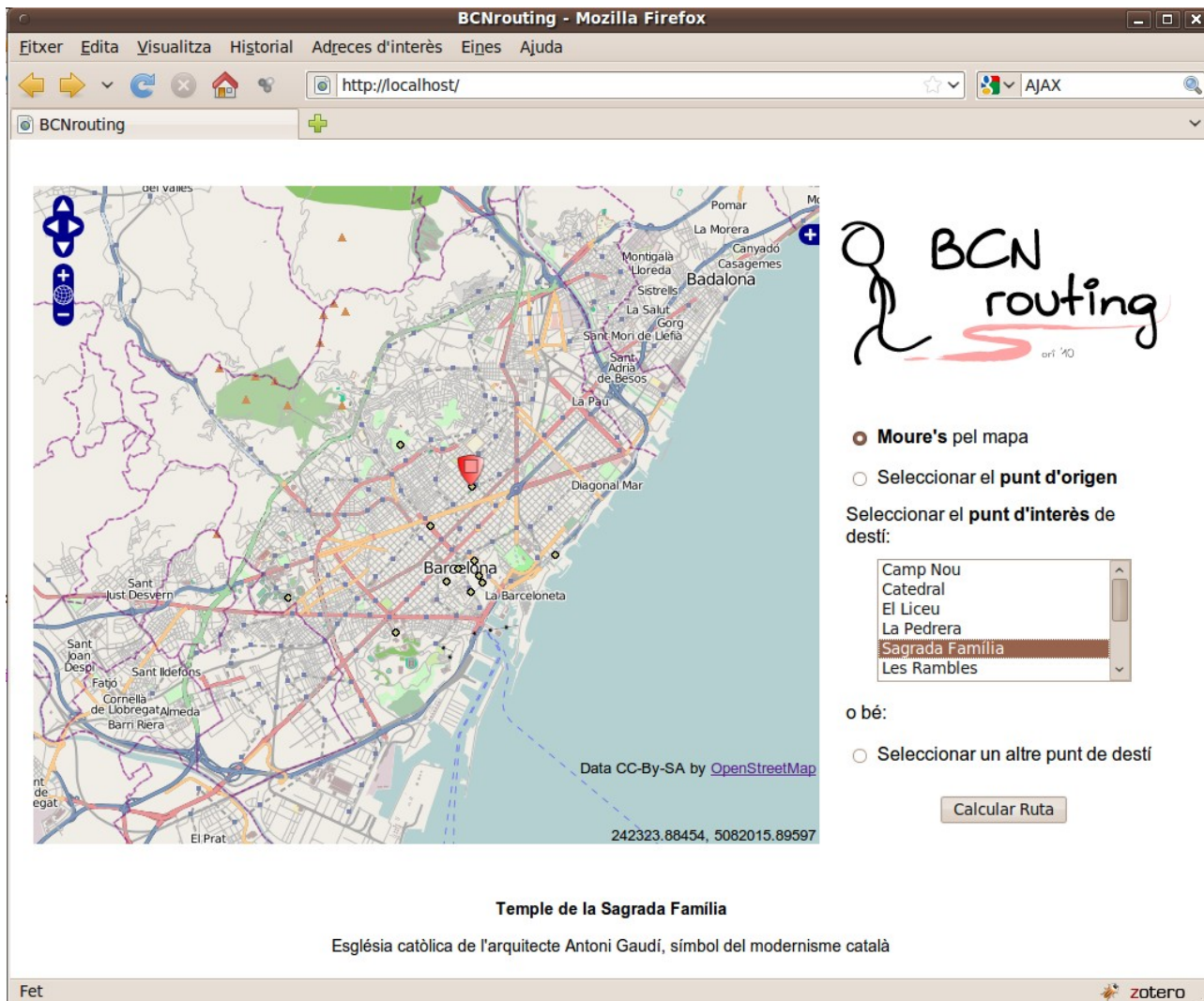


Figura 6-5. Aspecte de l'entorn web de l'aplicació BCNrouting mitjançant OpenLayers

A continuació es descriuran cadascun dels fitxers que constitueixen i aporten la funcionalitat a aquest entorn web.

### 6.4.1 Fitxer map per accedir a les dades PostGIS

El fitxer *map* utilitzat perquè MapServer actuï de servidor WMS i serveixi les dades de la base de dades PostGIS com a capes per mostrar-se en l'entorn web mitjançant OpenLayers és el següent:



```
MAP
NAME "BCNrouting"
# Mida de la imatge
SIZE 600 600
UNITS meters

EXTENT 226246.726704 5060350.187500 254732.132671 5081972.312500
FONTSET './fonts.txt'
SYMBOLSET './symbols.sym'
PROJECTION
  'init=epsg:900913'
END

# Color de fons del contenidor del mapa
IMAGECOLOR 255 255 255
IMAGEQUALITY 95
IMAGETYPE jpeg

OUTPUTFORMAT
NAME jpeg
DRIVER 'GD/JPEG'
MIMETYPE 'image/jpeg'
IMAGEMODE RGBA
EXTENSION 'jpeg'
END

# Llegendra
LEGEND
  IMAGECOLOR 255 255 255
  STATUS ON
  KEYSIZE 18 12
  LABEL
    TYPE BITMAP
    SIZE MEDIUM
    COLOR 0 0 89
  END
END

# Definició de l'interfície web
WEB
# Directori on MapServer escriu i llegeix les imatges generades
IMAGEPATH '/BCNrouting/tmp/'
IMAGEURL '/BCNrouting/tmp/'
# Característiques del servidor WMS
METADATA
  'ows_title'          'BCNrouting'
  'ows_onlineresource' 'http://localhost/cgi-bin/mapserv?map=/var/www/BCNrouting/BCNrouting.map'
  'ows_srs'            'EPSG:4326 EPSG:900913'
  'wms_srs'            'EPSG:4326 EPSG:900913'
END
TEMPLATE 'fooOnlyForWMSGetFeatureInfo'
END
```

```
LAYER
  NAME 'ways'
  TYPE LINE
  DUMP true
  TEMPLATE fooOnlyForWMSGetFeatureInfo
  EXTENT 226246.726704 5060350.187500 254732.132671 5081972.312500
  CONNECTIONTYPE postgis
  CONNECTION "dbname='bcnrouting3' host=127.0.0.1 port=5432 user='oriol' password='origis' sslmode=disable"
  DATA 'the_geom FROM "ways" USING UNIQUE gid USING srid=900913'
  METADATA
    'ows_title' 'ways'
  END
  STATUS OFF
  TRANSPARENCY 100
  CLASS
    NAME 'ways'
    STYLE
      SYMBOL 0
      SIZE 7.0
      OUTLINECOLOR 208 222 138
      COLOR 0 0 0
    END
  END
END

LAYER
  NAME 'puntsturistics'
  TYPE POINT
  DUMP true
  TEMPLATE fooOnlyForWMSGetFeatureInfo
  EXTENT 226246.726704 5060350.187500 254732.132671 5081972.312500
  CONNECTIONTYPE postgis
  CONNECTION "dbname='bcnrouting3' host=127.0.0.1 port=5432 user='oriol' password='origis' sslmode=disable"
  DATA 'the_geom FROM "puntsturistics" USING UNIQUE gid USING srid=900913'
  METADATA
    'ows_title' 'puntsturistics'
  END
  STATUS OFF
  TRANSPARENCY 100
  CLASS
    NAME 'puntsturistics'
    STYLE
      SYMBOL 'circle'
      SIZE 7.0
      OUTLINECOLOR 0 0 0
      COLOR 201 201 127
    END
  END
END
END
```

Els aspectes més importants a destacar d'aquest fitxer *map* són la especificació de la projecció EPSG:900913 (com ja s'ha comentat a l'apartat 6.3), l'extensió del mapa, i la definició de les capes. Es defineixen dues capes, una per la taula de carrers (*ways*) de tipus *line*, i una pels punts turístics (*puntsturistics*) de tipus *point*, que es mostraran mitjançant el símbol *circle* (els símbols es troben descrits a un fitxer de text apart, *simbols.sym*). Per cadascuna de les dues capes s'estableix una connexió amb la base de dades (*bcnrouting3* en aquest cas, que és la que conté les dades amb el SRID 900913) i es seleccionen les dades corresponents mitjançant una consulta que obté la columna geomètrica (*the\_geom*) de la taula.

### 6.4.2 Codi HTML de la pàgina i de l'script d'OpenLayers

La pàgina HTML és la que conté l'script per mostrar el mapa d'OpenLayers i tota la funcionalitat relacionada amb aquest. És on hi ha els controls per marcar els punts sobre el mapa, la llista dels punts turístics i les seccions on es mostra la informació que s'obté de la base de dades mitjançant crides a fitxers *php* externs. El codi bàsic del fitxer HTML de la pàgina, comentat per parts, és el següent:

A l'inici de la capçalera s'especifica la localització de la llibreria OpenLayers de forma remota. La llibreria també es pot emmagatzemar al servidor web i utilitzar-la de forma local.

```
<html>
<head>
  <title>BCNrouting</title>
  <script src="http://openlayers.org/api/OpenLayers.js"></script>
```

Seguidament, comença l'script d'OpenLayers. S'especifiquen els estils de les etiquetes pels punts d'origen i destí, i un tipus de variable (classe d'OpenLayers) per emmagatzemar els punts que es cliquin sobre el mapa. Les etiquetes dels punts d'origen i destí utilitzades són arxius d'imatge *png* externs que s'han dibuixat per aquest treball (s'inclouen com a annex).

```
<script type="text/javascript">
  var SinglePoint = OpenLayers.Class.create();
  SinglePoint.prototype = OpenLayers.Class.inherit(OpenLayers.Handler.Point, {
    createFeature: function(evt) {
      this.control.layer.removeFeatures(this.control.layer.features);
      OpenLayers.Handler.Point.prototype.createFeature.apply(this, arguments);
    }
  });
```

```

var start_style = OpenLayers.Util.applyDefaults({
  externalGraphic: "inici.png",
  graphicWidth: 24,
  graphicHeight: 27,
  graphicYOffset: -27,
  graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

var stop_style = OpenLayers.Util.applyDefaults({
  externalGraphic: "fi.png",
  graphicWidth: 24,
  graphicHeight: 27,
  graphicYOffset: -27,
  graphicOpacity: 1
}, OpenLayers.Feature.Vector.style['default']);

var result_style = OpenLayers.Util.applyDefaults({
  strokeWidth: 3,
  strokeColor: "#ff0000",
  fillOpacity: 0
}, OpenLayers.Feature.Vector.style['default']);

```

Es declaren a continuació unes variables globals que s'utilitzaran en més d'una funció de l'*script*. Aquestes variables són el propi mapa, els punts d'origen i destí (*start* i *stop*), la variable *controls* (per guardar l'acció que s'està duent a terme sobre el mapa), i les capes *rutaResultat* i *puntTuristic*. La primera per emmagatzemar la geometria de la ruta calculada i la segona per emmagatzemar el punt seleccionat de la llista de punts d'interès turístic.

```
var map, start, stop, controls, rutaResultat, puntTuristic;
```

Comença la funció *init()* que s'executarà al carregar la pàgina. En la variable *options* s'especifiquen les característiques del mapa a mostrar: la projecció, les unitats, els nivells de zoom, la resolució i l'extensió màxima del mapa. Seguidament es crea l'objecte mapa (*map*) amb aquestes opcions i s'afegeixen dos controls addicionals al mapa, un per poder canviar les capes mostrades i l'altre per mostrar les coordenades sota el cursor al senyalar el mapa.

```

function init(){
  var options = {
    projection: new OpenLayers.Projection("EPSG:900913"),
    units: "m",
    numZoomLevels: 18,
    maxResolution: 156543.0339,
    maxExtent: new OpenLayers.Bounds( 226247, 5060351, 254733, 5081973)

```

```

}

map = new OpenLayers.Map('map', options);
map.addControl(new OpenLayers.Control.LayerSwitcher());
map.addControl(new OpenLayers.Control.MousePosition());

```

Seguidament, es declaren i descriuen les capes que s'afegiran al mapa. Tenim una primera capa (*mapnik*) de tipus OSM, que s'encarregarà d'aportar la capa base d'OpenStreetMap al nostre mapa. Seguidament, trobem dues capes de tipus WMS (*wms* i *puntswms*), que aportaran la informació de la base de dades PostGIS al mapa, a través de peticions WMS a MapServer, i a partir de la selecció de les capes corresponents definides al fitxer *map*. Amb l'opció `transparent: 'true'` s'indica que es tracta de capes superposades a la capa base d'OSM. Finalment es declaren quatre capes de tipus Vector que contindran les geometries corresponents als punts d'inici i final de la ruta, i a la ruta resultat calculada, i es dibuixaran al mapa segons els estils especificats. Hi ha dues geometries que poden marcar el punt final de la ruta perquè a l'aplicació es dona l'opció de seleccionar aquest punt d'una llista de punts d'interès o bé de seleccionar un punt lliure al mapa (i s'han utilitzat capes diferents per aquests dos tipus diferents de punt de destí)

```

var mapnik = new OpenLayers.Layer.OSM("OSM base layer");

var wms = new OpenLayers.Layer.WMS( "Ways PostGIS", "http://localhost/cgi-bin/mapserv?
map=/var/www/BCNrouting/BCNrouting.map", {layers: 'ways', transparent:'true'} );
wms.setVisibility(false);

var puntswms = new OpenLayers.Layer.WMS( "Punts Turistics", "http://localhost/cgi-
bin/mapserv?map=/var/www/BCNrouting/BCNrouting.map", {layers: 'puntsturistics',
transparent:'true'} );

start = new OpenLayers.Layer.Vector("Punt d'origen", {style: start_style});
stop = new OpenLayers.Layer.Vector("Punt de desti", {style: stop_style});
puntTuristic = new OpenLayers.Layer.Vector("Punt Seleccionat", {style: stop_style});
rutaResultat = new OpenLayers.Layer.Vector("Routing results", {style: result_style});

```

S'afegeixen les capes al mapa i s'enquadra la vista a la ciutat de Barcelona a partir d'unes coordenades EPSG:4326 que es transformen a la projecció utilitzada al mapa.

```

map.addLayers([mapnik, puntswms, wms, rutaResultat, start, stop, puntTuristic]);
var proj4326 = new OpenLayers.Projection("EPSG:4326");
var limits = new OpenLayers.Bounds(2.064378, 41.324683, 2.256328, 41.470384);
limits.transform(proj4326, map.getProjectionObject());
map.zoomToExtent(limits);

```

La última part de la funció *init()* i la funció *toggleControl()* s'encarreguen de determinar què farà un clic sobre el mapa, i d'emmagatzemar els punts d'origen i destí que es cliquin sobre el mapa a les variables *start* i *stop*. Al tenir dues possibilitats per la selecció del punt de destí (llista i clic al mapa) és necessari que a la funció *toggleControl()*, quan es decideix seleccionar el punt per clic al mapa, s'esborri la possible selecció de punt que s'hagi efectuat anteriorment a la llista. Això es realitza accedint al *listbox* per l'identificador i establint la propietat de *selectedIndex* a -1. També s'elimina el punt de la capa *puntTuristic* perquè es deixi de mostrar al mapa.

```

controls = {
  start: new OpenLayers.Control.DrawFeature(start, SinglePoint),
  stop: new OpenLayers.Control.DrawFeature(stop, SinglePoint),
}
for (var key in controls) {
  map.addControl(controls[key]);
}
} // fi init

function toggleControl(element) {
  for (key in controls) {
    if (element.value == key && element.checked) {
      controls[key].activate();
    } else {
      controls[key].deactivate();
    }
  }
  if (element.value == "stop" && element.checked) {
    document.getElementById("llistaPuntsT").selectedIndex = -1;
    puntTuristic.removeFeatures(puntTuristic.features);
  }
}

```

A continuació es descriuen les dues funcions de l'*script* encarregades del càlcul de ruta. La funció *compute()* (que s'executa a l'apretar el botó de la pàgina) comprova que s'hagin marcat els punts d'inici i final de ruta, i en cas que així sigui, carrega la pàgina *calculruta.php* en un segon pla i li passa com a paràmetres les coordenades (x,y) dels punts d'inici i final. Per al punt final, es comprova anteriorment si n'hi ha algun marcat a la llista o bé si cal agafar el clicat sobre el mapa. Si la càrrega de la pàgina *calculruta.php* acaba correctament (sense errors), s'executa la funció *mostraRuta()*. *Calculruta.php* (del qual es detallarà la seva funcionalitat més endavant) retorna un fitxer XML que conté totes les arestes corresponents a la ruta resultat en format WKT. La funció

*mostraRuta()* és l'encarregada d'interpretar aquesta informació (rep el fitxer XML retornat com a paràmetre) i d'afegir aquestes arestes a la capa vectorial *rutaResultat* per tal de mostrar-la al mapa d'OpenLayers. Per això s'utilitza la tècnica AJAX i s'accedeix als elements de l'XML resposta de la funció *compute()* a partir del les seves etiquetes (*tag names*).

```
function compute() {
    var startPoint = start.features[0];
    var llista = document.getElementById("llistaPuntsT");
    if (llista.selectedIndex != -1){
        var stopPoint = puntTuristic.features[0];
    } else {
        var stopPoint = stop.features[0];
    }
    if (startPoint && stopPoint) {
        OpenLayers.loadURL("calculruta.php?startpoint="+startPoint.geometry.x+"
"+startPoint.geometry.y+"&finalpoint="+stopPoint.geometry.x+" "+stopPoint.geometry.y, null, null,
mostraRuta, null);
    }
}

function mostraRuta(ruta) {
    if (ruta && ruta.responseXML) {
        rutaResultat.removeFeatures(rutaResultat.features);
        function parseWKT(wkt) {
            parser = new OpenLayers.Format.WKT();
            var geometry = parser.read(wkt)
            var features = parser.read(wkt);
            var bounds;
            if(features) {
                if(features.constructor != Array) {
                    features = [features];
                }
                for(var i=0; i<features.length; ++i) {
                    if (!bounds) {
                        bounds = features[i].geometry.getBounds();
                    } else {
                        bounds.extend(features[i].geometry.getBounds());
                    }
                }
                rutaResultat.addFeatures(features);
            } else {
                alert ("wrong");
                element.value = 'Bad WKT';
            }
        } // final function parseWKT
        var arestes = ruta.responseXML.getElementsByTagName('aresta');
        var features = [];
        for (var i = 0; i < arestes.length; i++) {
            parser = new OpenLayers.Format.WKT();
```

```

    var tram = parseWKT(arestes[i].getElementsByTagName('wkt')[0].textContent);
    features.push(new OpenLayers.Feature.Vector(tram));
  }
  rutaResultat.addFeatures(features);
} // final if

```

A la mateixa funció *mostraRuta()* s'aprofita per calcular la distància total del recorregut (ja que al fitxer XML s'ha incorporat una etiqueta amb la informació de la longitud acumulada per tram) i del temps estimat per fer-lo a peu (considerant una velocitat de 80 m al minut<sup>26</sup>). Ambdues dades es mostren en seccions (<div>) de la pàgina (*distRecorregut* i *tempsRecorregut*), ocultes fins al moment. Finalment, per comoditat, es passa el control del mapa a moure.

```

var mostraDist = document.getElementById('distRecorregut');
var dist = ruta.responseXML.getElementsByTagName('longitudm')[i-1].textContent;
mostraDist.innerHTML = (dist/1000).toFixed(2);
document.getElementById('distancia').style.visibility = 'visible';
mostraDist = document.getElementById('tempsRecorregut');
mostraDist.innerHTML = ((dist/1000)/0.08).toFixed();

document.getElementById("moure").click();
} // fi mostraRuta

```

Les dues últimes funcions de l'*script* aporten la funcionalitat a la llista dels punts turístics: al clicar un element de la llista, es mostra la seva situació al mapa (juntament amb una descripció al peu) i es prepara per actuar de punt de destí. La funció *llista()* s'executa al seleccionar un element del *listbox* i rep com a paràmetre el nom del punt turístic seleccionat. Seguidament, la funció borra del mapa el punt que hi hagués seleccionat anteriorment com a destí i realitza la càrrega en segon pla de *mostraPunt.php*, que retorna un XML amb la informació de la base de dades postGIS del punt seleccionat corresponent al nom, les coordenades (x,y) i la descripció del punt. Si aquesta sol·licitud no dóna errors, s'executa la funció *mostraPunt()*, que rep com a paràmetre el XML retornat pel fitxer *php*.

La funció *mostraPunt()* crea una nova geometria d'OpenLayers de tipus punt, amb les coordenades del punt d'interès, i l'afegeix a la capa *puntTuristic* per tal d'assenyalar-lo al mapa. Seguidament es mostra el nom del punt i la descripció obtinguda de la base de dades en seccions (<div>) de la pàgina (*nomPuntTuristic* i *descripcio*), obtenint aquesta informació a partir de l'accés a les diferents etiquetes del fitxer XML.



```

function llista(nomPunt){
    puntTuristic.removeFeatures(puntTuristic.features);
    stop.removeFeatures(stop.features);
    document.getElementById("moure").click(); // Canvi del control del mapa a moure
    OpenLayers.loadURL("mostraPunt.php?nom="+nomPunt, null, null, mostraPunt, null);
}

function mostraPunt(coordenades){
    var x = coordenades.responseXML.getElementsByTagName('x')[0].firstChild.nodeValue;
    var y = coordenades.responseXML.getElementsByTagName('y')[0].firstChild.nodeValue;
    var punt = new OpenLayers.Geometry.Point(x, y);
    var feature = new OpenLayers.Feature.Vector(punt);
    puntTuristic.addFeatures(feature);

    var mostraDescripcio = document.getElementById('nomPuntTuristic');
    mostraDescripcio.innerHTML = coordenades.responseXML.getElementsByTagName('nom')[0].firstChild.nodeValue;
    mostraDescripcio = document.getElementById('descripcio');
    mostraDescripcio.innerHTML = coordenades.responseXML.getElementsByTagName('desc')[0].firstChild.nodeValue;
}
</script>
</head>

```

Finalment es mostra el codi HTML de la pàgina en sí, on es pot veure la secció pel mapa (`<div id="map">`), els botons de radio exclusius (amb mateix nom) que criden la funció `toggleControl()`, la llista de punts turístics (`<select>`) amb les seves opcions i valors, el botó de càlcul de ruta que crida la funció `compute()`, i totes les seccions (`<div>`) per mostrar la informació.

```

<body onLoad="init()">
    <br><br>
    <table cellpadding="0" cellspacing="0" border="0" align="center" valign="middle" width="1000"
height="600">
        <tr align="center" valign="middle">
            <td width="680" align="center"><div style="width:100%; height:100%" id="map"></div></td>
            <td width="320" align="center">
                <p></p>
                <br>
                <table width="280" border="0">
                    <tr>
                        <td width="23" height="32"><input type="radio" name="control" value="moure"
id="moure" onClick="toggleControl(this);" checked/></td>
                        <td width="247"><strong>Moure's</strong> pel mapa</td>
                    </tr>
                    <tr>
                        <td height="34"><input type="radio" name="control" value="start" id="selStart"
onClick="toggleControl(this);" /></td>
                        <td><label for="selStart">Seleccionar el <strong>punt
d'origen</strong></label></td>

```

```

</tr>
<tr>
  <td height="46" colspan="2">Seleccionar el <strong>punt d'interès</strong>
de destí;: </td>
</tr>
<tr>
  <td height="114" colspan="2" align="center"><p>
    <select name="llistaPuntsT" size="6" id="llistaPuntsT"
onChange="llista(this.value)" style="width:220px;">
      <option value="Camp Nou">Camp Nou</option>
      <option value="Catedral de la Santa Creu">Catedral</option>
      <option value="Gran teatre del Liceu">El Liceu</option>
      <option value="La Pedrera">La Pedrera</option>
      <option value="Temple de la Sagrada">Sagrada Família</option>
      <option value="Les Rambles">Les Rambles</option>
      <option value="Montju">Montjuïc</option>
      <option value="Museu d'art contemporani">MACBA</option>
      <option value="Sant Jaume">Plaça de Sant Jaume</option>
      <option value="Port Ol">Port Olímpic</option>
      <option value="Parc G">Parc Güell</option>
    </select>
  </td>
</tr>
<tr>
  <td height="40" colspan="2">o b</td>
</tr>
<tr>
  <td height="29"><input type="radio" name="control" value="stop"
id="selPuntTuristic" onClick="toggleControl(this);" /></td>
  <td>Seleccionar un altre punt de destí;</td>
</tr>
</table>
<br>
<button onClick="compute()">Calcular Ruta</button>
</td>
</tr>
<tr align="center" valign="middle">
  <td width="680" height="32" align="center"><div id="distancia" style="visibility:
hidden">Hauràs de caminar <strong><span id="distRecorregut"></span></strong>
quil·metres! (aproximadament <strong><span id="tempsRecorregut"></span></strong> minuts a
peu)</div></td>
</tr>
</table>
<br>
<strong><div id="nomPuntTuristic" align="center"></div></strong><br>
<div id="descripcio" align="center"></div>
</body>
</html>

```

### 6.4.3 Codi del fitxer de càlcul de rutes (*calculruta.php*)

El fitxer *calculruta.php*, cridat al pressionar el botó per calcular la ruta, és l'encarregat de comunicar-se amb la base de dades PostGIS i realitzar les consultes.

Inicialment, es defineixen els paràmetres per la connexió amb la base de dades. Es defineix el nom de la base de dades, el servidor i el port de connexió, les dades de l'usuari, i les taules amb les que es treballarà, que seran la que conté la informació dels carrers (*ways*) i la que conté els nodes (*vertices\_tmp*).

```
<?php
define("PG_DB" , "bcnrouting3");
define("PG_HOST", "127.0.0.1");
define("PG_USER", "oriol");
define("PG_PASS", "origis");
define("PG_PORT", "5432");
define("T_CARRERS", "ways");
define("T_VERTEXS", "vertices_tmp");
```

A continuació es guarden les coordenades dels punts d'origen i destí que s'han passat com a paràmetres en les variables *startPoint* i *endPoint* com a llistes (x, y).

```
// Guardem els punt d'origen i destí passats com a paràmetres
$start = split(' ', $_REQUEST['startpoint']);
$startPoint = array($start[0], $start[1]);

$end = split(' ', $_REQUEST['finalpoint']);
$endPoint = array($end[0], $end[1]);
```

Seguidament, es busquen els nodes més propers als punts d'origen i destí, a partir de les seves coordenades. Una consulta SQL per realitzar això ja s'ha vist a l'apartat 4.4 d'aquest treball, i s'aplica aquesta mateixa consulta en aquesta connexió amb la base de dades mitjançant *php*. Es guarda el node resultant (l'identificador i la geometria, encara que la geometria no és necessària pel càlcul de la ruta) i es retorna amb la funció. En aquest procés els nodes d'origen i destí queden emmagatzemats a les variables *startEdge* i *endEdge*.

```
$startEdge = buscaNodeMesProper($startPoint);
$endEdge = buscaNodeMesProper($endPoint);

function buscaNodeMesProper($punt) {
// Connexió a la base de dades
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS." user=".PG_USER);
// Sentència SQL i cerca:
```

```

$sql = "SELECT id, the_geom, distance(the_geom, GeometryFromText( 'POINT(".$punt[0]." ".
$punt[1].")', 900913)) AS dist FROM ".T_VERTEXS." ORDER BY dist LIMIT 1;";
$query = pg_query($con,$sql);
// Guardem els resultats
$node['id']      = pg_fetch_result($query, 0, 0);
$node['the_geom'] = pg_fetch_result($query, 0, 1);
// Tanquem la connexió amb la base de dades
pg_close($con);
// Retornem el node
return $node;
}

```

A continuació, a partir dels nodes d'origen i destí, es realitza la consulta pel càlcul de la ruta. S'utilitza la funció *dijkstra\_sp()* de pgRouting sobre la taula dels carrers de Barcelona. El resultat d'aplicar *dijkstra\_sp()* és una taula (que hem anomenat *ruta*) amb les arestes de la ruta i una sola columna, la geomètrica (*the\_geom*). La consulta utilitza aquesta taula resultant per obtenir la informació desitjada. Per això s'utilitzen les funcions *AsText()* i *length()* de postGIS sobre la columna geomètrica, per tal de convertir la geometria al format WKT i de calcular la longitud de cada tram respectivament. Així doncs, la taula final resultat de la consulta contindrà 3 columnes: l'identificador de l'aresta, la geometria en format WKT i la longitud.

```

// Consulta SQL:
$sql = "SELECT ruta.gid, AsText(ruta.the_geom) as wkt, length(ruta.the_geom) as long
      FROM dijkstra_sp('".T_CARRERS."', ".$startEdge['id'].", ".$endEdge['id'].") as ruta";
// Connexió a la base de dades i aplicació de la consulta
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS." user=".PG_USER);
$query = pg_query($con,$sql);

```

Amb el resultat de la consulta es prepara el fitxer XML de retorn, en el que s'afegeixen diverses etiquetes per cada tram (aresta): una numeració seqüencial que identifica el tram, la geometria en WKT i la longitud acumulada. Finalment, es tanca la connexió i es retorna el fitxer XML.

```

// S'afegeixen els carrers de la ruta (arestes) al fitxer xml (tenim la geometria en WKT)
$xml .= "<ruta>\n";
// per cada aresta (fila de la consulta) guardem la informació
while($aresta=pg_fetch_assoc($query)) {
    $longitudAcumulada += $aresta['long'];
    $xml .= "\t<aresta id='".$n."'>\n"; //numeració dels trams
    $xml .= "\t\t<wkt>".$aresta['wkt']."</wkt>\n";
    $xml .= "\t\t<longitudm>".$longitudAcumulada."</longitudm>\n";
    $xml .= "\t</aresta>\n";
}
$xml .= "</ruta>\n";

```

```
// Tanquem la connexió amb la base de dades
pg_close($con);

// Retornem el resultat com a xml
header('Content-type: text/xml');
echo $xml;
?>
```

#### 6.4.4 Codi del fitxer de selecció i marca del punt turístic (mostraPunt.php)

El fitxer `MostraPunt.php`, cridat al seleccionar un punt d'interès de la llista, és l'encarregat de d'obtenir la informació sobre el punt (coordenades i descripció) de la base de dades.

Igual que al càlcul de la ruta, primer es defineixen els paràmetres per la connexió amb la base de dades i es guarda el nom del punt turístic (provinent del *listbox*) que s'ha passat com a paràmetre.

```
<?php
define("PG_DB" , "bcnrouting3");
define("PG_HOST", "127.0.0.1");
define("PG_USER", "oriol");
define("PG_PASS", "origis");
define("PG_PORT", "5432");
define("TABLE" , "puntsturistics");

$nom = $_REQUEST['nom']; // Guardem el nom del punt passat com a parametre
```

A continuació es realitza la consulta. Per això es busca el punt turístic a la taula *puntsturistics* a partir del nom que s'havia entrat com a paràmetre. Del punt turístic s'obtenen el nom, les coordenades mitjançant les funcions `ST_X()` i `ST_Y()` de postGIS sobre la columna geomètrica, i la descripció.

```
// connexió amb la base de dades:
$con = pg_connect("dbname=".PG_DB." host=".PG_HOST." password=".PG_PASS." user=".PG_USER);
// sentència sql de cerca i cerca:
$sql = "select nom, ST_X(the_geom) as X, ST_Y(the_geom) as Y, descripcio from ".TABLE." where
(nom like '%".$nom."%')";
$query = pg_query($con,$sql);
```

Finalment es guarda el resultat de la consulta, es converteix la codificació dels caràcters de UTF-8 (de la base de dades) a ISO-8859-1 per tal de que els accents i caràcters especials es mostrin correctament a la pàgina, i es retorna el resultat en un fitxer XML, estructurat en etiquetes pel nom, les coordenades i la descripció del punt turístic en qüestió.

```

// guardem el resultat:
$coordenades = pg_fetch_row($query);
// Convertim la codificació perquè es mostrin accents correctament :
$coordenades[0] = mb_convert_encoding($coordenades[0], "ISO-8859-1", "UTF-8");
$coordenades[3] = mb_convert_encoding($coordenades[3], "ISO-8859-1", "UTF-8");

// Preparem el XML de retorn:
$xml = '<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>'. "\n";
$xml .= "<puntTuristic>\n";
$xml .= "<nom>".$coordenades[0]."</nom>\n";
$xml .= "<x>".$coordenades[1]."</x>\n";
$xml .= "<y>".$coordenades[2]."</y>\n";
$xml .= "<desc>".$coordenades[3]."</desc>\n";
$xml .= "</puntTuristic>\n";

// Tanquem connexió amb la base de dades
pg_close($con);
// Retornem el XML resultat
header('Content-type: text/xml');
echo $xml;
?>

```

## 6.5 Exemple de funcionament

En aquest apartat es mostrarà de forma visual el funcionament de l'aplicació web mitjançant captures de pantalla. La figura 6-6 mostra una ruta calculada a partir de la selecció d'un punt d'origen proper a Poblenou i d'un punt de destí seleccionat de la llista: el teatre del Liceu.

El càlcul de la ruta és ràpid (es percep de forma instantània), és possible moure o ampliar el mapa (es selecciona automàticament l'opció *moure* un cop calculada la ruta), i es poden canviar els punts d'origen i destí i recalculer rutes sobre el mateix mapa. Es pot observar també la informació que apareix sota el mapa, que correspon a la descripció del punt turístic seleccionat a la llista i la distància total del recorregut (i l'estimació temporal), que apareix al calcular la ruta.

La figura 6-7 mostra una ampliació de la zona de destinació per la ruta calculada, i es pot comprovar com la ruta no acaba exactament al punt de destí indicat. Això és degut a que el càlcul del camí més curt es realitza entre dos nodes (i no entre punts enmig de les arestes), i com a conseqüència la ruta acaba al node més proper al punt de destí introduït, que és com s'ha programat la aplicació. La figura 6-8 és un exemple de la cerca del node més proper a un punt clicat sobre el mapa. Es pot observar que segons la zona de la plaça Catalunya on es cliqui, es pren un node o un altre segons la proximitat, per tal d'ajustar la ruta el millor possible.



**Gran teatre del Liceu**

Reconegut teatre d'òpera situat a les Rambles de Barcelona

Figura 6-6. Càlcul d'una ruta entre un punt de Poble Nou i el teatre del Liceu utilitzant l'aplicació web basada en OpenLayers sobre els carrers de Barcelona en una base de dades PostGIS.

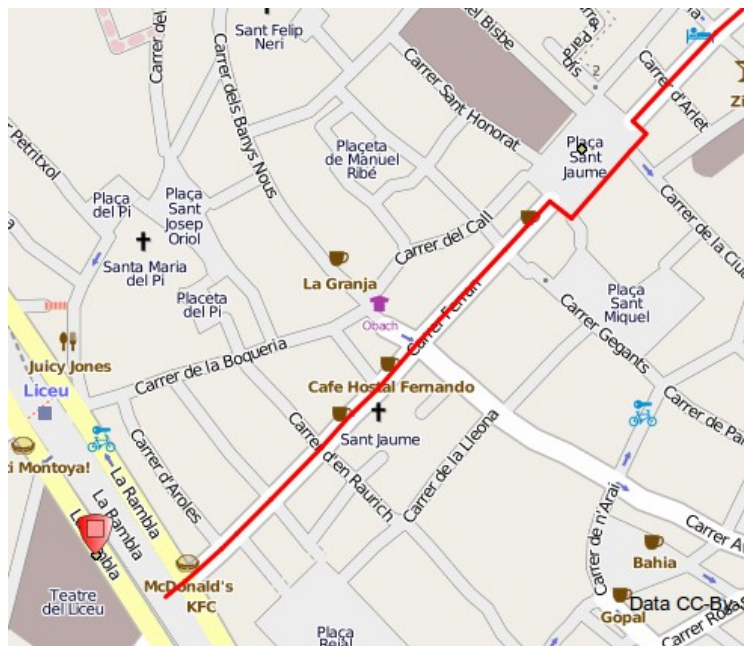
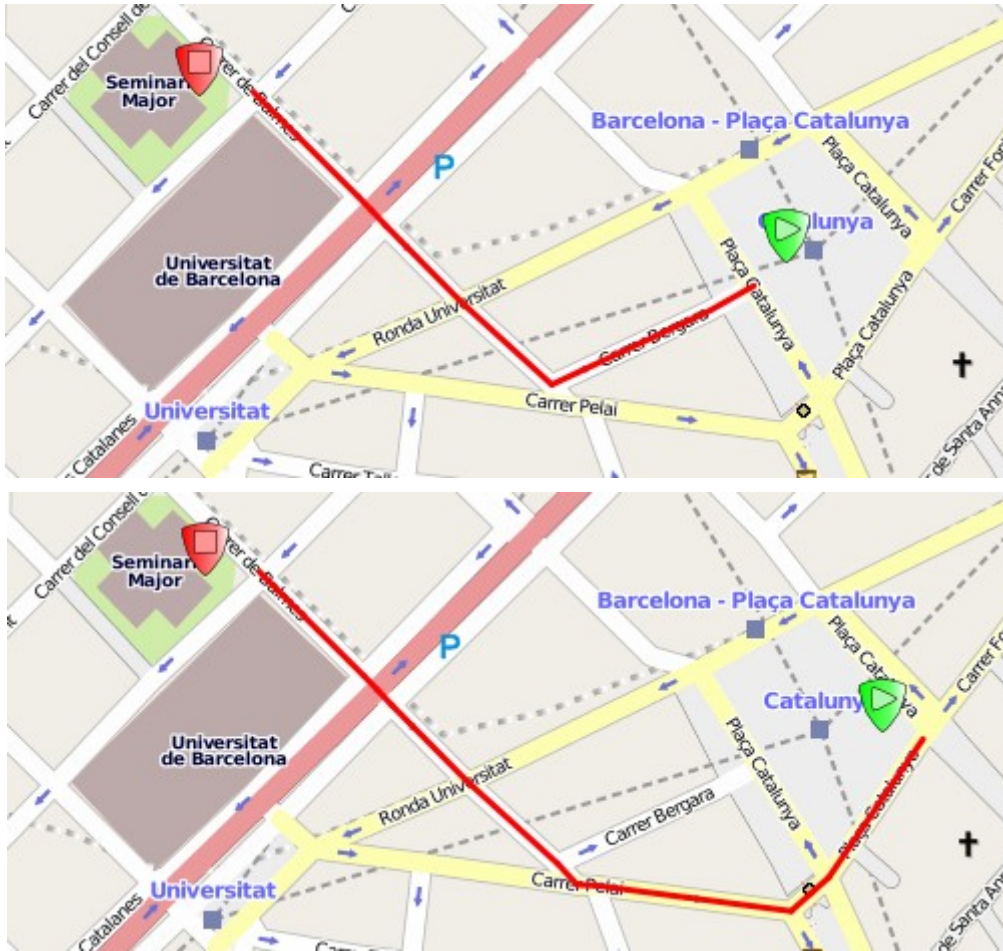


Figura 6-7. Ampliació de la zona de destí a la ruta calculada a la figura 6-6.





**Figura 6-8.** Exemple de selecció del node més proper al punt clicat sobre el mapa per al càlcul de la ruta.



## Capítol 7:

# Conclusions i línies futures

En aquest apartat s'exposen les conclusions del treball i les possibles ampliacions i línies de treball futures que en poden derivar.

Les conclusions que s'han tret en la realització d'aquest treball són:

- Fent ús únicament d'eines de programari lliure és possible desenvolupar un SIG per al càlcul de rutes de potencial equiparable a eines propietàries. La majoria de les eines utilitzades (postgreSQL/postGIS, MapServer i OpenLayers) es troben perfectament documentades, fet que en facilita el seu ús. PgRouting, en canvi, no té una documentació tan extensa ni estructurada com les anteriors, i el seu ús és més confós i difícil. Malgrat això, pgRouting ha resultat una eina molt potent per al càlcul de rutes, i aquest treball pretén ser una petita guia per la seva utilització.
- El projecte OpenStreetMap és una molt bona font d'informació geogràfica, que a més, és lliure. La facilitat per extreure'n les dades geogràfiques i l'existència d'eines com osm2pgrouting per importar aquestes dades a una base de dades postGIS simplifiquen el procés d'emmagatzematge de dades geogràfiques amb topologia per a realitzar-hi càlculs.
- Les possibilitats proporcionades per OpenLayers i el llenguatge Javascript (amb la tècnica AJAX) són molt àmplies, i permeten la presentació de la informació extreta del SIG a un usuari final, d'una forma interactiva i de qualitat. En aquest sentit, l'extensió Firebug ha estat de molta utilitat per a l'anàlisi i depuració del codi directament en temps d'execució.

Al finalitzar el treball s'han complert els objectius que s'havien marcat en un inici, encara que són moltes les millores que es poden realitzar a l'aplicació que s'ha desenvolupat. Algunes d'aquestes millores podrien ser:

- Considerar aspectes d'eficiència, de disseny, i de seguretat de l'aplicació, que no han estat objectiu d'aquest treball. Aquestes consideracions podrien incloure, per exemple, l'ús d'un usuari amb permisos mínims per accedir a la base de dades, la optimització de les consultes (temps de resposta) i les funcions utilitzades, la implementació de mecanismes de control d'errors, i l'adaptació per als diferents navegadors web del mercat (al treball només s'ha adaptat per al Mozilla Firefox).
- Implementar una ordenació a la llista dels punts turístics, de forma que, al clicar un punt d'origen, la llista quedi ordenada segons la proximitat de cada punt turístic a aquest punt d'origen. Per això cal realitzar una consulta a la base de dades cada vegada que es marca un punt d'origen i reordenar la llista segons la resposta de la base de dades (utilitzant AJAX).
- Mostrar el llistat de carrers pels que passa la ruta calculada. La implementació d'aquesta funcionalitat passa per mostrar el nom de les arestes de la ruta (de forma que no es repetissin, ja que mateix carrer pot estar format per moltes arestes) a la pàgina web (utilitzant també la tècnica AJAX, com ja s'ha fet en el treball per la distància total). Per això també cal que totes les arestes de la base de dades tinguin el nom del carrer al que pertanyen.
- Les rutes es calculen prenent d'origen i destí els vèrtexs més propers del punts clicats al mapa, i no entre aquests punts exactes o els mes propers a aquests sobre una via. Encara que l'error que es comet normalment no és molt gran (per els càlculs del centre de la ciutat), seria convenient arreglar-ho. Aquest és un problema que es troba solucionat en exemples de la xarxa,<sup>24</sup> encara que la ruta resultant la dibuixen sobre OpenLayers com una capa vectorial, recuperant la informació geomètrica de la consulta realitzada amb *php*.
- Finalment, es podrien millorar aspectes d'interactivitat, com per exemple poder seleccionar el punt turístic sobre el mateix mapa, o bé donar la possibilitat de seleccionar altres medis de transport.

Aquest treball pot servir de punt de partida pel desenvolupament d'altres aplicacions més complexes. Com a idea, es podria considerar una aplicació que al seleccionar un punt d'origen et

preguntés quina distància estàs disposat a caminar, i segons aquesta distància, proposar uns punts d'interès turístic a visitar. Respecte els punts d'interès turístic, si es volgués realitzar una pàgina web pública, es podria considerar un sistema d'introducció i valoració dels punts d'interès per part dels usuaris, de forma que hi hagués un llistat de punts d'interès del sistema i un llistat dels punts d'interès proposats pels propis usuaris, tots ells amb les valoracions dels usuaris. També es podrien organitzar aquests punts d'interès per categories.

Per acabar, únicament assenyalar que aquest ha estat el meu primer contacte amb el món dels SIG i, que al llarg del treball, ha estat necessari consultar molta documentació i superar alguns entrebancs que han anat sorgint (i que ja s'han comentat al llarg de la memòria). En aquest sentit, ha estat necessari familiaritzar-se amb diferents conceptes i tècniques pel desenvolupament del SIG. Per una banda, hi ha tots els conceptes, terminologia i eines específiques dels SIG, però per altra banda, també ha estat necessari un coneixement fonamental de bases de dades, de desenvolupament de pàgines web i de Javascript. Aquesta interdisciplinarietat fa que el desenvolupament d'un SIG no sigui una tasca trivial.

## Referències

- 1 **Wikipedia, List of geographic information systems software**, [consulta 9 d'abril de 2010], <[http://en.wikipedia.org/wiki/List\\_of\\_geographic\\_information\\_systems\\_software](http://en.wikipedia.org/wiki/List_of_geographic_information_systems_software)>.
- 2 (a) **Steiniger, S.; Bocher, E.** (2009) "An overview on current free and open source desktop GIS developments." *International Journal of Geographical Information Science*, (vol. 23, num. 10, p 1345-1370). (b) **Montesinos, M.; Sanz, J.G.** (2007). "Panorama actual del ecosistema de software libre para SIG". *I Jornadas de SIG Libre*. Girona: Servei de sistemes d'informació geogràfica i teledetecció, UdG.
- 3 (a) **The FreeGIS project**, [Consulta 9 d'abril de 2010], <<http://www.freegis.org/>>. (b) **Open Source GIS**, [Consulta 9 d'abril de 2010], <<http://opensourcegis.org/>>. (c) **OSGeo, Your Open Source Compass**, [Consulta 9 d'abril de 2010], <<http://www.osgeo.org/>>.
- 4 **Sushi-kiwami**, [Consulta: 9 de març de 2010], <<http://www.sushi-kiwami.com>>.
- 5 **Ride the City**, [Consulta: 9 de març de 2010], <<http://www.ridethecity.com>>.
- 6 (a) **Galati, S.** (2006) *Geographic information systems demystified*, Boston, Artech House. (b) **Longley, P.** (2005) *Geographical Information Systems and Science*, England, Wiley.
- 7 **OGC Glossary**, [Consulta: 8 de maig de 2010], <<http://www.opengeospatial.org/ogc/glossary>>
- 8 **MapServer, Open Source Web Mapping**, [Consulta 9 d'abril de 2010], <<http://mapserver.org>>
- 9 **PostGIS spatial database extension for PostgreSQL**, [Consulta 9 d'abril de 2010], <<http://postgis.refractions.net>>
- 10 **The GiST indexing project**. [Consulta 15 d'abril de 2010], <<http://gist.cs.berkeley.edu/>>
- 11 **GEOS - Geometry Engine, Open Source**. [Consulta 9 d'abril de 2010], <<http://trac.osgeo.org/geos/>>
- 12 **PROJ.4 - Cartographic Projections Library**, [Consulta 9 d'abril de 2010], <<http://trac.osgeo.org/proj/>>
- 13 **pgRouting project**, [Consulta 9 d'abril de 2010], <<http://pgrouting.postlbs.org/>>
- 14 **Boost c++ libraries**, [Consulta 9 d'abril de 2010], <<http://www.boost.org/>>
- 15 **OpenLayers: Free Maps for the Web**, [Consulta 9 d'abril de 2010], <<http://www.openlayers.org/>>
- 16 **Osm2pgrouting tool**, [Consulta 9 d'abril de 2010], <<http://pgrouting.postlbs.org/wiki/tools/osm2pgrouting>>
- 17 **gvSIG, Conselleria d'infrastructures i transport**, [Consulta 9 d'abril de 2010], <<http://www.gvsig.gva.es/>>
- 18 **uDIG, User-friendly Desktop Internet GIS**, [Consulta 9 d'abril de 2010], <<http://udig.refractions.net/>>
- 19 **Quantum GIS project**, [Consulta 9 d'abril de 2010], <<http://www.qgis.org/>>
- 20 **Firebug**, [Consulta 30 de maig de 2010], <<http://getfirebug.com/>>
- 21 **OpenStreetMap - The Free Wiki World Map**, [Consulta 9 d'abril de 2010], <<http://www.openstreetmap.org>>
- 22 **Institut Cartogràfic de Catalunya**, [Consulta 9 d'abril de 2010], <<http://www.icc.es/>>
- 23 **Osm2pgsql**, [Consulta 9 d'abril de 2010], <<http://wiki.openstreetmap.org/wiki/Osm2pgsql>>
- 24 (a) **Philipona, C.; Kastl, D.; Junod, F.; Patrushev, A.** "Web-Based Routing: An Introduction to pgRouting with OpenLayers?". *FOSS4G 2007*. [Consulta 11 de maig de 2010], <<http://pgrouting.postlbs.org/wiki/WorkshopFOSS4G2007>>. (b) **Behncke, K.; Westerholt, R.** "Workshop pgRouting in OpenLayers (2.8) with own

- OSM-data". [Consulta 11 de maig de 2010], <<http://pgrouting.postlbs.org/wiki/WorkshopOL2.7andOSM>>
- 25 **SphericalMercator**, [Consulta 11 de maig de 2010], <<http://trac.openlayers.org/wiki/SphericalMercator>>
- 26 **Wikipedia, walking**, [consulta 29 de maig de 2010], <[http://en.wikipedia.org/wiki/Walking#Walking\\_speed](http://en.wikipedia.org/wiki/Walking#Walking_speed)>

## **Bibliografia general**

- Pérez, A.; Botella, A.; Muñoz, A.; Olivella, R.; Olmedillas, J.C.; Rodríguez, J.** (2009). *Sistemes d'informació geogràfica i geotelemàtica* (1a ed.). Barcelona: FUOC.
- Pérez, A.; Bataller, A.; Beneito, R.; Sáenz, N.; Vidal, R.** (2008). *Treball final de carrera* (1a ed.). Barcelona: FUOC.
- Galati, S.** (2006) *Geographic information systems demystified*, Boston: Artech House.
- Longley, P.** (2005) *Geographical Information Systems and Science* Paul A. Longley, England: Wiley.