(Llicència GPL)
Pot copiar i distribuir el Programa (o un treball basat en ell, segons s'especifica en l'apartat 2, com a codi objecte o en format executable segons els termes dels apartats 1 i 2, suposat que a més compleixi una de les següents condicions:
1. Acompanyar-lo amb el codi font complet corresponent, en format electrònic, que ha de ser distribuït segons s'especifica en els apartats 1 i 2 d'aquesta Llicència en un medi habitualment utilitzat per a l'intercanvi de programes, o
2. Acompanyar-lo amb una oferta per escrit, vàlida durant almenys tres anys, de proporcionar a qualsevol tercera part una còpia completa en format electrònic del codi font corresponent, a un cost no major que el de realitzar físicament la distribució del font, que serà distribuït sota les condicions descrites en els apartats 1 i 2 anteriors, en un medi habitualment utilitzat per a l'intercanvi de programes, o
3. Acompanyar-lo amb la informació que vas rebre oferint distribuir el codi font corresponent. (Aquesta opció es permet només per a distribució no comercial i només si vostè va rebre el programa com a codi objecte o en format executable amb tal oferta, d'acord amb l'apartat 2 anterior).

BCN-Routing:

Càlcul web de rutes amb pgRouting, OpenStreetMaps i OpenLayers.

Memòria

Alumne: Antonio Gómez Solivellas
Directora: Ana Muñoz Bollas

Treball de fi de carrera
Enginyeria Tècnica Informàtica de Sistemes
Curs 2009-10, 2on semestre

Resum

Es presenta un sistema acurat de càlcul web d'itineraris mínims (per a vianants i per a vehicles) entre dos punts de la ciutat de Barcelona, un dels quals es triat per l'usuari directament a sobre un mapa i l'altre, alternativament, a sobre el mateix mapa o bé a sobre una llista de selecció de les principals atraccions turístiques de Barcelona.

El sistema es troba implementat per medi de *MapServer (1)* com a servidor, *OpenLayers* (2) per a la interfície d'usuari, una base de dades *PostrgreSQL* (3)/*PostGIS* (4) que recull dades d'*OpenStreetMaps* (5) per a la navegació i dades introduïdes manualment, per a la llista de selecció d'atraccions turístiques. Per al càlcul d'itineraris es fa servir, pgRouting (6) alhora que s'accedeix a la cartografia de *CartoCiudad* (7) per a mostrar un mapa de base i opcionalment els noms dels carrers i punts d'interès a partir de les capes FondoUrbano, Vial i Topónimo del servidor WMS de CartoCiudad.

Tot el sistema corre a sobre *Windows 7 Home Premium (8)*.

Així mateix es presenten quatre noves funcions i un tipus definit per l'usuari de *PostgreSQL* per al càlcul acurat d'itineraris mínims i l'estudi teòric que justifica la seva bondat.

Taula de continguts

1 Figures 6
2 Taules 7
3 Introducció 8
4 Objectius del projecte 10
4.1 Objectius Generals 11
4.2 Objectius específics 12
5 Metodologia de treball 13
5.1 Requeriments materials 14
5.1.1 Requeriments de maquinari *14*
5.1.2 Requeriments de programari *14*
5.1.2.1 Software propietari 14
5.1.2.2 Software lliure 15
5.2 Instal·lació del programari necessari 16
5.2.1 Instal·lació de UMN MapServer *17*
5.2.2 Instal·lació de la geodatabase PostgreSQL *18*
5.2.2.1 Instal·lació de PostgreSQL a sobre Ubuntu 9.1 18
5.2.2.2 Instal·lació de PostgreSQL a sobre Windows 7 19
5.2.3 Instal·lació de les extensions PostGIS i pgRouting *19*
5.2.3.1 Instal·lació PostGIS/pgRouting a sobre Ubuntu 9.10 19
5.2.3.2 Instal·lació PostGIS/pgRouting a sobre Windows 7 21
5.2.4 Instal·lació d'OpenLayers i Firebug *22*
5.2.5 Instal·lació d'altre software necessari pel projecte *22*
5.3 Recopilació i tractament de les dades 23
5.3.1 Recerca de les fonts de dades WMS i WFS cartogràfiques necessàries *23*
5.3.1.1 Servidor WMS de CartoCiudad 23
5.3.1.2 Servidor WFS de CartoCiudad 25
5.3.2 Recerca de dades a OpenStreetMap *27*
5.3.3 Importació de les dades OSM a la geodatabase PostGIS *28*
5.4 Estudi de programari anterior 32
5.5 Plantejament de les necessitats per al càlcul d'itineraris 36
5.6 Estratègia utilitzada 37
5.6.1 Nomenclatura *37*

5.6.2 Dificultat del càlcul 38
5.6.3 Selecció de l'algoritme de base 39
5.6.4 Estratègia 39
5.7 Estudi teòric 40
6 Descripció de la funcionalitat implementada 49
7 Interfície d'usuari 52
8 Resultats 56
9 Conclusions 58
10 Millores i línies de futur 59
11 Bibliografia 60

1 Figures

Figura 1. Pantalla de presentació de MapServer 2.3.1 18
Figura 2. Consulta Servidor WMS CartoCiudad 24
Figura 3. Consulta Servidor WMS-Cache CartoCiudad 25
Figura 4. Consulta servidor WFS-vial CartoCiudad: CATALUNYA / Barcelona 26
Figura 5. Consulta servidor WFS-portal CartoCiudad: CATALUNYA, 1 / Barcelona 27
Figura 6. Captura pantalla importació OpenStreetMap 28
Figura 7. Vies superposades 30
Figura 8. Culs de sac 30
Figura 9. Errades geomètriques 31
Figura 10 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben després dels punts origen i destinació 34
Figura 11 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben abans dels punts origen i destinació 34
Figura 12. Nomenclatura utilitzada a l'estudi teòric 37
Figura 13 Dificultat del càlcul 38
Figura 14 Possibles recorreguts entre dos punts situats a sobre dues arestes per shortest_path i addicions 41
Figura 15 Itinerari TS→SE, únic possible entre dues arestes unidireccionals 42
Figura 16. Possibles itineraris SS→SE per al cas d'aresta inicial bidireccional i final unidireccional 44
Figura 17 Possibles itineraris TS→SE entre una aresta inicial bidireccional i una final unidireccional 45
Figura 18. Diagrama de interacció de les rutines 51
Figura 19 Pantalla inicial de la interfície d'usuari 52
Figura 20 Pantalla introducció punt origen 53
Figura 21 Pantalla tria mètode selecció punt destinació (Punt o Atracció) 53
Figura 22 Pantalla selecció punt final a sobre el mapa 54
Figura 23 Pantalla selecció punt final a atracció 54
Figura 24 Pantalla amb les capes Fondo Urbano, Vial i Topónimo activades 55
Figura 25 Itinerari per a vianants amb el programari desenvolupat 56
Figura 26 Itinerari per a vehicles amb el programari desenvolupat 57

2 Taules

Taula 1. Software instal·lat a sobre Windows 7 16
Taula 2. Software instal·lat a sobre Ubuntu 9.10 17
Taula 3. Origen de les dades trobades. 23
Taula 4. Trajectes possibles en funció de la bidireccionalitat de les arestes inicial i final 46
Taula 5 Recorregut net total a sobre les arestes inicial i final en funció de la bidireccionalitat d'aquestes 47
Taula 6. Recorreguts nets a sobre les arestes inicial i final en funció del trajecte 47
Taula 7. Correccions geometries algoritme shortest_path 48

3 Introducció

Una necessitat intrínseca de la humanitat ha estat disposar de medis per a la descripció del seu entorn. Seria raonable pensar que aquesta necessitat pot haver evolucionat des de la simple descripció dels itineraris per a desplaçar-se d'un lloc a un altre cap a un coneixement més depurat de la distribució a sobre el terreny de les distintes matèries primeres o punts d'interès.

Des d'antic s'ha disposat d'una eina molt valuosa per a aconseguir aquests objectius, parlem de la cartografia, que ens permetria tant la determinació dels itineraris com la descripció del què i a on es troba cada cosa a sobre el territori.

Aquesta informació merament descriptiva no tindria gaire valor sinó fora per les conclusions que es poden extreure a partir del seu estudi que, inicialment restringit a la cartografia convencional, s'ha vist enormement facilitat per l'aparició d'un programari que permet d'alguna manera automatitzar el tractament d'aquesta mena d'informació. Aquesta tipus de programari s'ha vingut a anomenar com a Sistemes d'informació geogràfica (SIG) i en resum es tracten de sistemes amb programari preparat per a treballar amb dades georeferenciades.

Els SIG avui dia permeten moltes possibilitats entre les quals destaquem per l'interès que suposa per al present treball la determinació de les rutes òptimes per al desplaçament entre dos punts.

L'aparició del *Web* 2.0 ha permès portar als exploradors d'Internet aplicacions que abans eren típicament d'escriptori posant a l'abast d'una gran part de la població aquesta mena d'aplicatius on-line. Si bé podem trobar programari propietari resulta d'especial interès, per als objectius del present treball, l'existència de software lliure.

L'estudi funcional de la implementació d'un SIG, amb programari lliure, que permeti la determinació de rutes òptimes a través del web es podria dividir en quatre blocs principals:

- Magatzem de dades georeferenciades.
- Servidor de mapes.
- Client per a l'interacció amb l'usuari.
- Software per al càlcul de la ruta òptima.

Així, el servidor de mapes ens permetria servir aquests a partir de les dades subministrades per l'usuari al client el qual s'encarregaria de mostrar les rutes calculades per el software de càlcul a sobre les dades incloses al magatzem.

Quan parlem de magatzem de dades georeferenciades ens podem estar referint a simples arxius de dades o en el que en principi resulta més eficient, a vertaders sistemes de bases geogràfiques. Dintre d'aquesta definició podem trobar el gestor de bases de dades relacional de codi lliure *PostgreSQL* al que la seva extensió *PostGIS* dota de capacitats com a base de dades geogràfica.

Pel que fa al servidor de mapes trobem *UMN MapServer*, molt potent i de codi lliure.

També trobem *OpenLayers*, es tracta d'un client *webGIS* lleuger desenvolupat amb classes *javascript*, sense dependència de servidors de mapes concrets. Ofereix una interfície d'usuari que permet la connexió amb serveis *WMS (Web Map Service)* i *WFS (Web Feature Service)* de forma transparent per l'usuari i pel desenvolupador.

Ja per a acabar i pel que fa al software de càlcul de rutes, podem parlar de *pgRouting*, extensió per bases de dades geogràfiques *PostGIS/PostgreSQL* que proporciona eines per a serveis basats en la localització. Inclou diversos algorismes per al càlcul de rutes, eines per a la comparació de dades de carreteres en format *OpenStreetMap* (OSM) i la seva explotació es realitza mitjançant consultes SQL a la mateixa BD que conté les dades. D'aquesta manera es pot realitzar un anàlisi de rutes, calculant el camí més curt entre dos punts donats, mitjançant l'ús de *PostGIS* i visualitzar els resultats usant el servidor de mapes *UMN MapServer* amb un client SIG lleuger com per exemple *OpenLayers*.

4 Objectius del projecte

L'objectiu del present Treball de fi de carrera (TFC) es crear un portal web que presentarà un servei d'enrutament circumscrit a la ciutat de Barcelona. Aquest portal pretén oferir als nou vinguts un sistema senzill d'obtenció de la ruta de menor cost, per a vianants i vehicles, des d'un punt d'origen donat sobre el mapa a una de les principals destinacions turístiques de la ciutat.

Més concretament, es pretén crear un portal web amb *UMN MapServer* que presentarà la ruta més curta obtinguda a partir de les dades subministrades a través de la interfície d'usuari a la mateixa plana web. L'obtenció de les rutes es farà per medi de consultes amb *pgRouting* a una base de dades *PostgreSQL* (amb l'extensió *PostGIS*) fent servir el client lleuger *OpenLayers* per a la visualització de les mateixes. Les dades necessàries per l'assoliment dels objectius indicats hauran de ser obtingudes per medi de recerca (principalment a *OpenStreetMap*) i inserides, amb *osm2pgrouting*, a la base de dades abans indicada alhora que es crea la topologia a sobre les mateixes.

Podem considerar com a objectius parcials d'aquest projecte:

1. La instal·lació i configuració, si s'escau, del programari necessari per el funcionament del sistema (*MapServer*, *PostgreSQL, PostGIS, pgRouting i OpenLayers*).
2. Recopilació i tractament de les dades per a la seva inserció a la base de dades.
3. Disseny de consultes amb *pgRouting* per al càlcul de rutes.
4. Disseny de la interfície d'usuari.

Ja per a acabar amb aquest apartat, podem considerar objectius col·laterals d'aquest projecte la presa de contacte amb el programari lliure per a la creació de portals web d'enrutament i així mateix la comprovació, tal com es comenta més endavant, del funcionament d'aquest software a sobre *Windows 7*, sistema propietari de nova fornada.

Als apartats 4.1 i 4.2 presentem els objectius generals i específics establerts a l'enunciat del TFC.

4.1 Objectius Generals

En acabar aquest treball, s'espera que l'estudiant sigui capaç de manipular i analitzar explícitament la informació espacial en un entorn web, concretament amb dades d'*OpenStreetMap* i amb un visor *OpenLayers* emprant les funcions de càlcul de rutes de què disposa *pgRouting*, i que assoleixi els següents objectius:

1. Comprendre els conceptes de la tecnologia SIG i la seva metodologia.
2. Conèixer l'estructura dels diferents tipus de dades amb què treballa un SIG.
3. Conèixer els sistemes d'emmagatzemament estàndards, tant d'informació *ràster* com vectorial, i ser capaç d'ubicar la informació en les coordenades que correspongui.
4. Trobar, generar i manipular dades geogràfiques.
5. Saber plantejar un SIG. Conèixer les operacions d'anàlisi espacial i transformacions en el SIG analitzat.

4.2 Objectius específics

El SIG que cal desenvolupar té els següents objectius:

1. Conèixer l'estat actual de l'ecosistema d'aplicacions lliures, i en particular les que corresponen als servidors de mapes.
2. Treballar amb dades d'*OpenStreetMap*.
3. Analitzar les diferents funcions de càlcul de rutes de què disposa *pgRouting*: *Dijkstra*, *TSP problems*, etc . Per a establir l'escenari òptim d'ús de cadascuna d'elles.
4. Implementar un visor de la ruta obtinguda amb *OpenLayers*.

5 Metodologia de treball

La metodologia seguida ha constat d'una sèrie d'etapes que s'han seguit en un ordre bàsicament rigorós, tret de la instal·lació de Kosmos GIS que en un principi no havia estat prevista.

Les etapes, que es recopilen a continuació, han estat les següents:

1. Establiment dels requeriments materials per al desenvolupament del treball proposat (Apartat 5.1).
2. Instal·lació del programari específicament relacionat amb el sistema de càlcul d'itineraris (Apartat 5.2).
3. Recopilació i tractament de les dades per a la navegació i presentació de mapes (Apartat 5.3).
4. Estudi de programari anterior (Apartat 5.4).
5. Plantejament de les necessitats per al càlcul d'itineraris (Apartat 5.5).
6. Disseny de l'estratègia per al sistema de càlcul (Apartat 5.6).
7. Estudi teòric del problema plantejat (Apartat 5.7).

5.1 Requeriments materials

Podem considerar dos tipus de requeriments materials per al desenvolupament del projecte:

- Requeriments de maquinari.
- Requeriments de programari.

5.1.1 Requeriments de maquinari

Tot i que els requeriments de maquinari es corresponen amb els del punt de treball estàndard de la UOC, s'ha fet servir el següent maquinari:

- *Intel Core2 Quad CPU* Q9550 2.83 GHz
- 6 GB de memòria *RAM*
- RAID 1 de 2 Discs durs *Western Digital*® WD10EARS 1TB SATA 2, 7200 rpm 64MB cache
- Targeta gràfica ATI® RADEON HD 4850
- Monitor HP® W22 22" 1680 x 1024
- Connexió a Internet amb cable 3 Mbps
- HP Media Vault per copies de seguretat

5.1.2 Requeriments de programari

Podem dividir el programari que s'ha fet servir en el desenvolupament del projecte, la confecció de la memòria i el disseny de la presentació virtual en dos grans blocs, per una banda el software propietari i per una altra el software lliure, a continuació es relacionen els softwares que s'ha fer servir agrupats segons la divisió anteriorment indicada.

5.1.2.1 Software propietari

El software propietari utilitzat ha estat el següent:

- Microsoft® Windows® 7 Home Premium 64 bits. Sistema operatiu
- Microsoft® Office Word® 2007. Redacció dels diferents documents del projecte: informes de les PACs, memòria, etc.
- Microsoft® Office Project® 2003. Diagrames de Gantt del pla de projecte.

- Microsoft® Office PowerPoint® 2007. Preparació de la presentació virtual.
- Microsoft® Office Excel® 2007. Càlculs i taules addicionals.
- Camtasia Studio 6. Presentació virtual.

5.1.2.2 Software lliure

Pel que fa al software lliure s'ha fet servri:

- *PostgreSQL* amb les seves extensions *PostGIS i pgRouting*, en les seves versions per a *Ubuntu* i *Windows 7*.
- *UMN MapServer* en la seva versió per a *Windows 7*.
- *OpenLayers* en la seva versió per a *Windows 7*.
- *Firebug* en la seva versió per a *Windows 7*.
- *NotePad++*
- *Kosmo GIS* 2.0 RC1

5.2 Instal·lació del programari necessari

Com ja s'ha comentat anteriorment, amb aquest treball s'ha pretès, a més a més dels objectius marcats a l'enunciat, explorar les possibilitats que ofereix el software lliure per a la implementació d'un sistema de *web routing* a sobre *Windows 7*, sistema operatiu propietari de darrera fornada.

Cal fer constar que, com ja es preveia inicialment, una de les principals dificultats amb la què hem topat al llarg de la realització d'aquest treball ha estat la problemàtica de la importació de les dades, que es discuteix més detingudament a l'apartat: **5.3.3.Importació de les dades *OSM* a la *geodatabase PostGIS***.

Com a resultat de la indicada dificultat ha estat necessari procedir a la importació de les dades fent una primera càrrega de les mateixes a sobre una base de dades *PostgreSQL/PostGIS*, amb *pgRouting*, instal·lada a sobre un sistema operatiu lliure (*Ubuntu 9.1*) per a la posterior exportació a una base de dades de les mateixes característiques a sobre *Windows 7*. És per aquest motiu que encara que no es va preveure inicialment al pla de treball ha estat necessari incloure posteriorment, a aquest apartat, un subapartat de la instal·lació de *PostgreSQL, PostGIS, pgRouting* i *osm2pgrouting* a sobre *Ubuntu 9.10*.

A continuació afegim taules resum del software instal·lat a sobre cadascun dels dos sistemes operatius:

Taula 1. Software instal·lat a sobre *Windows 7*

Software	Origen
MapServer 2.3.1	http://maptools.org/dl/ms4w/ms4w_2.3.1.zip
PostgreSQL 8.4.3-1	http://www.enterprisedb.com/products/pgdownload.do#windows
Pgadmin3 1.10.1	Instal·lat amb *PostgreSQL*
PostGIS 1.5.1	*Stack Builder*
pgRouting 1.03	http://pgrouting.postlbs.org/wiki/pgRoutingDownload
OpenLayers 2.8	http://www.maptools.org/ms4w/index.phtml?page=downloads.html
Firefox 3.6.2	http://www.mozilla-europe.org/es/firefox/
Firebug 1.5.2	http://getfirebug.com/
Kosmo GIS 2.0 RC1	http://www.opengis.es/index.php?option=com_docman&task=cat_view&gid=17&It

	emid=42

Taula 2. Software instal·lat a sobre *Ubuntu 9.10*

Software	Origen
PostgreSQL 8.4	*Synaptic*
Pgadmin3 1.10.0-1	*Synaptic*
PostGIS 1.4.2	http://postgis.refractions.net/download/
pgRouting 1.03	http://files.postlbs.org/pgrouting/source/pgRouting-1.03.tgz
Osm2pgrouting r.356	http://pgrouting.postlbs.org/svn/pgrouting/tools/osm2pgrouting/trunk osm2pgrouting

5.2.1 Instal·lació de UMN MapServer

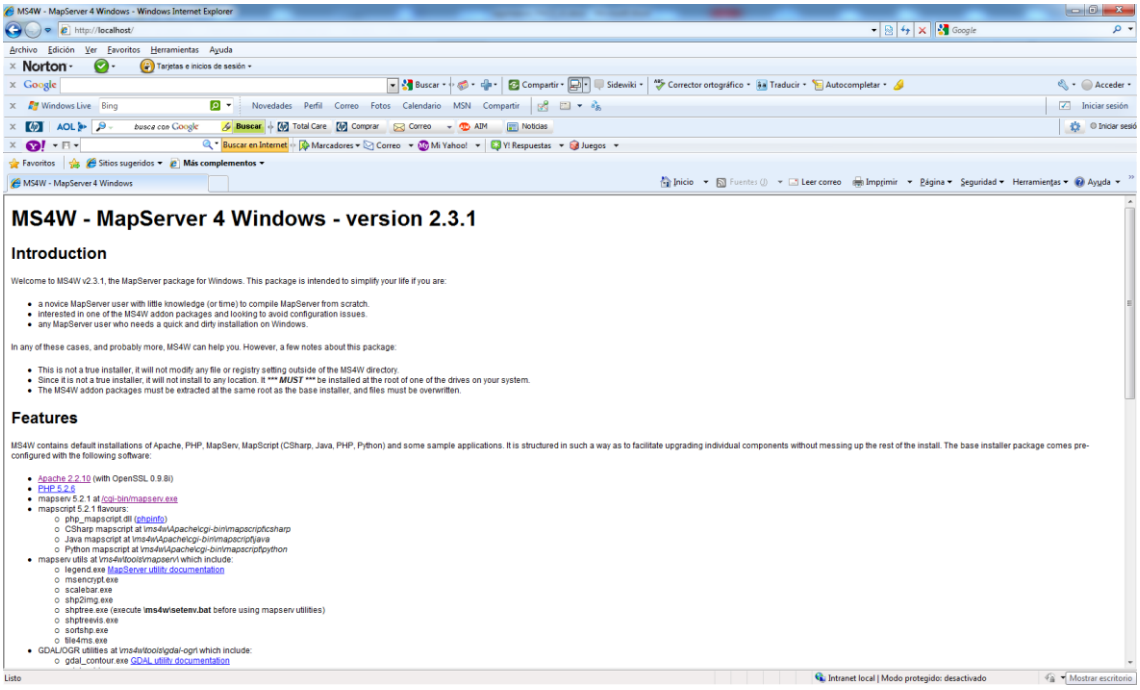
Comentar a aquest apartat que una vegada feta la instal·lació ens hem trobat amb un conflicte que feia que les connexions al *localhost* es readrecessin al Servei de Publicació *Word Wide Web* de *Windows 7*, de manera que les proves van demostrar la impossibilitat de connectar amb *MapServer*. La solució ha consistit en deshabilitar el dit servei amb la qual cosa el *MapServer* ha quedat disponible.

Pel que fa a la instal·lació pròpiament dita, aquesta s'ha dut a terme a sobre *Windows 7.* Per això s'ha obtingut l'arxiu d'instal·lació corresponent el qual no és, ni conté, un vertader instal·lable, en el sentint estricte de la paraula, donat que no cal modificar cap arxiu ni registre propi del sistema operatiu amfitrió per al seu funcionament. Aquesta característica simplificarà l'instal·lació *d'OpenLayers* com s'indica a l'apartat **5.2.4.Instal·lació d'OpenLayers i Firebug**.

L'arxiu descarregat: ms4w_2.3.1.zip ha estat descomprimit i el directori msw4 creat durant al descompressió ha estat copiat a l'arrel del disc dur primari del sistema operatiu. Per a acabar amb la instal·lació no ha calgut més que executar l'arxiu bat: apache-install.bat que instal·la i inicia el servidor *Apache*.

Per a la comprovació del funcionament de la "instal·lació", una vegada deshabilitat el IIS, hem intentat la connexió amb el navegador *Internet Explorer* a l'adreça *localhost* que ens ha permès visualitzar la pantalla de benvinguda de *MapServer*, tal qual es pot observar a la següent captura de pantalla:

Figura 1. Pantalla de presentació de *MapServer* 2.3.1



5.2.2 Instal·lació de la geodatabase *PostgreSQL*

Pels motius comentats a l'apartat **5.2.Instal·lació del programari necessari**, hem dividit la discussió d'aquesta instal·lació en dues parts: la instal·lació a sobre *Ubuntu 9.1* i la instal·lació a sobre *Windows 7*.

5.2.2.1 Instal·lació de PostgreSQL a sobre Ubuntu 9.1

Per a la instal·lació de *PostgreSQL* s'ha fet servir una instal·lació nova d'*Ubuntu 9.1* a sobre una màquina virtual generada per la versió V 3.1.4r57640 de *VirtualBox* (*Sun Microsystems*). Aquest sistema operatiu ha estat actualitzat amb totes les actualitzacions disponibles al moment de realitzar aquest treball.

La versió de *PostgreSQL* que s'ha fet servir ha estat la 8.4 i s'ha instal·lat a partir de *Synaptic*, per una altra banda i fent servir la mateixa eina, s'ha instal·lat pgadmin3 amb la seva versió 1.10.0-1.

Cal comentar que, com era d'esperar, la instal·lació ha transcorregut sense incidències ressenyables.

5.2.2.2 Instal·lació de PostgreSQL a sobre Windows 7

Hem obtingut i instal·lat la darrera versió de *PostgreSQL* per a *Windows* (8.4.3-1).

L'arxiu descarregat és un executable que instal·la l'aplicatiu i en acabar la instal·lació ofereix la possibilitat d'iniciar *l'Stack Builder*, aplicatiu per a afegir extensions, cosa que hem aprofitat per a instal·lar també la darrera versió, al moment de redactar aquest treball, de *PostGIS* (1.5.1).

5.2.3 Instal·lació de les extensions PostGIS i pgRouting

Com en el cas de l'apartat d'instal·lació de *PostgreSQL*, han calgut fer dues instal·lacions, una a sobre *Ubuntu 9.10* i l'altre a sobre *Windows 7*. Essent la primera, sens dubte, la més complicada.

5.2.3.1 Instal·lació PostGIS/pgRouting a sobre Ubuntu 9.10

Per una banda ha calgut instal·lar les eines de desenvolupament i compilació *build-essentials* i per l'altre *PostGIS* és una llibreria que te dependències de les següents llibreries:

- libgeos-dev: : Afegeix a *PostGIS* la possibilitat de geoprocessament.
- proj: Permetrà fer conversions entre diferents projeccions cartogràfiques.

Per tant un pas previ per a la nostra instal·lació ha estat instal·lar les esmentades llibreries de les quals hem instal·lat les darreres versions:

- build-essentials 11.4 (a partir de *Synaptic*)
- libgeos-dev 3.1.0-1 (a partir de *Synaptic*).
- proj 4.6.1-5 (a partir de *Synaptic*).

Si bé la intenció, en principi, era instal·lar la revisió actual de *PostGIS* (1.5.1) hem optat per la versió 1.4.2 donat que la darrera requereix la versió 3.1.0 de la llibreria libgeos mentre que la versió 1.5.1 de *PostGIS* requereix la versió 3.1.1 de la mateixa llibreria, la qual encara no es troba disponible per a *Ubuntu 9.10* i donat que tot i que aquest treball té un caire principalment experimental, no hem considerat adient afegir a la tasca la importació experimental d'una llibreria a sobre *Ubuntu 9.10* .

Amb l'objectiu de simplificar el treball amb les bases de dades geogràfiques hem creat una plantilla de base de dades que inclou les extensions *PostGIS* i a la qual hem denominat: template_postgis.

Fins a aquí el treball ha transcorregut sense incidències importants.

El pas següent (i el més complicat des del punt de vista d'incidències) ha estat la instal·lació de *pgRouting*.

Cal comentar que a la *Web* de *pgRouting*, la versió superior per a instal·lar és la ***Ubuntu 9.04*** **amb** ***PostgreSQL 8.3*** i **nosaltres ho hem fet amb** ***Ubuntu 9.10*** **i** ***PostgreSQL 8.4***.

Novament per a la compilació de *pgRouting* hem hagut d'instal·lar la llibreria cmake a banda de que *pgRouting* té dependències de les següents llibreries:

- libboost-graph-dev: per als algoritmes *Shortest-Path*.
- gaul-devel:per al TSP
- libcgal3 i ligcgal-dev: per al *Driving Distance*

Les versions que han estat instal·lades han sigut:

- cmake 2.6.4-1ubuntu2 (a partir de *Synaptic*).
- libboost-graph-dev 1.38 (també a partir de *Synaptic*)
- gaul-devel 0.1850-0 obtés a partir de: wget http://sourceforge.net/projects/gaul/files/gaul-devel/0.1850-0/gaul-devel-0.1850-0.tar.gz/download
- libcgal3 i ligcgal-dev 3.4-4ubuntu1 (a partir de *Synaptic*).

Pel que fa a *pgRouting* hem instal·lat la versió 1.03.

Cal destacar que hem generat els arxius make afegint TSP i DD, i que s'han produït una sèrie d'errades de compilació, aquest comportament es troba descrit a la plana *Web* de *pgRouting* al següent *link*:

http://pgrouting.postlbs.org/ticket/160

Es resol afegint #include "catalog/pg_type.h" als arxius:

./core/src/dijkstra.c
./core/src/astar.c
./core/src/shooting_star.c

Al nostre cas també ens ha calgut modificar els arxius següents amb el mateix include :

./extra/driving_distance/src/alpha.c

./extra/driving_distance/src/drivedist.c

El motiu és la inclusió del DD a la generació dels nostres arxius make.

Després de compilar i instal·lar *pgRouting*, hem creat una plantilla de base de dades amb la funcionalitat de *pgRouting* a la qual hem denominat template_routing creada a partir de template_postgis i executant els *scripts* sql:

./usr/share/postlbs/routing_core.sql
./usr/share/postlbs/routing_core_wrappers.sql
./usr/share/postlbs/routing_topology.sql
./usr/share/postlbs/routing_tsp.sql
./usr/share/postlbs/routing_tsp_wrappers.sql

A continuació hem instal·lat *osm2pgrouting* en la seva revisió 356.

La compilació ha produït una sèrie d'errors descrits, també, al següent *link* de la *Web* de *pgRouting*:

http://pgrouting.postlbs.org/discussion/message/1426

La solució consisteix en afegir #include "stdio.h" a ./src/XMLPArser.cp

5.2.3.2 Instal·lació PostGIS/pgRouting a sobre Windows 7

Com ja hem comentat a l'apartat d'instal·lació de *PostgreSQL*, en acabar aquesta ens demana si volem iniciar *l'Stack Builder* i amb aquest hem instal·lat la versió 1.5.1 de *PostGIS*, durant la instal·lació s'ha generat també una plantilla de base de dades que inclou les extensions *PostGIS* (template_postgis).

Pel que fa al *pgRouting*, hem instal·lat la versió 1.03.

La instal·lació s'ha limitat a descomprimir l'arxiu *zip* obtingut i a afegir les carpetes lib i share creades al directori d'instal·lació de *PostgreSQL*.

A continuació s'ha creat una plantilla de base de dades template_routing a partir de la plantilla template_postgis.

Ja per a acabar hem afegit les funcionalitats de *pgRouting* executant les consultes del sots directori d'instal·lació de *PostgreSQL*:

.\share\contrib

Routing_core.sql

Routing_core_wrappers.sql

Routing_topology.sql

5.2.4 Instal·lació d'OpenLayers i Firebug

Donat que com ja s'ha comentat repetidament, el treball es pretén realitzar a sobre *Windows 7*, no ha calgut instal·lar *OpenLayers* ni *Firebug* a sobre el sistema operatiu *Ubuntu 9.10*.

Pel que fa a *Windows 7* i *OpenLayers*, hem instal·lat la versió 2.8 a partir de l'arxiu: openlayers-2.8_ms4w.zip

Només ha calgut descomprimir I copiar la carpeta així obtinguda a sobre la de *MapServer*: ms4w.

La instal·lació de *Firebug* a sobre *Windows 7* té com a requisit previ la instal·lació de *Firefox*, del qual hem instal·lat la versió 3.6.2.

Pel que fa a *Firebug* hem instal·lat la versió 1.5.2.

5.2.5 Instal·lació d'altre software necessari pel projecte

S'ha cregut adient la instal·lació de *Kosmo GIS*, es tracta d'un SIG Lliure, distribuït sota llicència GNU/GPL amb unes funcionalitats de consulta, edició i anàlisis molt potents.

L'arxiu *zip* obtingut es correspon amb la versió 2.0 RC1 i per a la seva instal·lació ha estat suficient descomprimir i executar un arxiu bat denominat Kosmo.bat.

5.3 Recopilació i tractament de les dades

A continuació incloem un resum dels tipus de dades obtingudes i els seus orígens que més endavant passem a discutir detalladament:

Taula 3. Origen de les dades trobades.

Dades	Servidor	Adreça
WMS	WMS	http://www.cartociudad.és/wms/CARTOCIUDAD/CARTOCIUDAD?
	WMS Tessel·lat	http://www.cartociudad.és/wms-c/CARTOCIUDAD/CARTOCIUDAD?
WFS	WFS vial	http://www.cartociudad.és/wfs-vial/services
	WFS portal	http://www.cartociudad.és/wfs-portal/services
OSM	OpenStreetMap	http://www.openstreetmap.org/

5.3.1 Recerca de les fonts de dades WMS i WFS cartogràfiques necessàries

Les adreces dels servidors de dades *WMS* i *WFS* els hem obtingut a la plana *Web* de *CartoCiudad*.

http://www.cartociudad.es/portal/1024/index.htm

D'acord amb la definició que podem trobar al mateix portal: «...*CartoCiudad* és resultat de la integració i harmonització de dades aportades per diferents organismes públics (principalment la Direcció general del Cadastre, l'Institut Nacional d'Estadística, la Societat Estatal de Correus i Telègrafs i l'Institut Geogràfic Nacional) que ha donat lloc a un sistema d'informació geogràfica de xarxa viària contínua, i informació parcel·lària, censal i postal, l'àmbit de la qual és tot el territori nacional...».

A la mateixa plana web de *CartoCiudad* podem trobar el següent document que conté les especificacions dels distints serveis disponibles:

http://www.cartociudad.es/portal/pdf/CARTOCIUDAD_ServiciosWeb.pdf

5.3.1.1 Servidor WMS de CartoCiudad

Els serveis de mapes (*Web Map Service WMS*) produeixen mapes de forma dinàmica a partir d'informació geogràfica vectorial o *raster*. S'invoquen a través d'un navegador web o client, enviant una petició en forma d'*URL* (*Uniform Resource Locator*) i retornant a continuació una imatge digital al client.

A CartoCiudad podem trobar dos tipus de servidors WMS, per una banda un servidor convencional que compleix amb l'especificació del OGC per a WMS versió 1.1.1 a sobre el qual s'han definit les següents capes d'interès per al nostre projecte:

FondoUrbano: Conté les següents entitats:

- *Manzana* (geo:*Manzana*)

- *Parcela* (geo:*PARCELA*)

- *Construcción* (geo:CONSTRUCCIÓN)

- *Líneas Auxiliares* (geo:LINEA_AUXILIAR)

Vial: conté l'entitat *viales*, tant urbans com interurbans (geo:TRAMO_VIAL_SIN_BBOX)

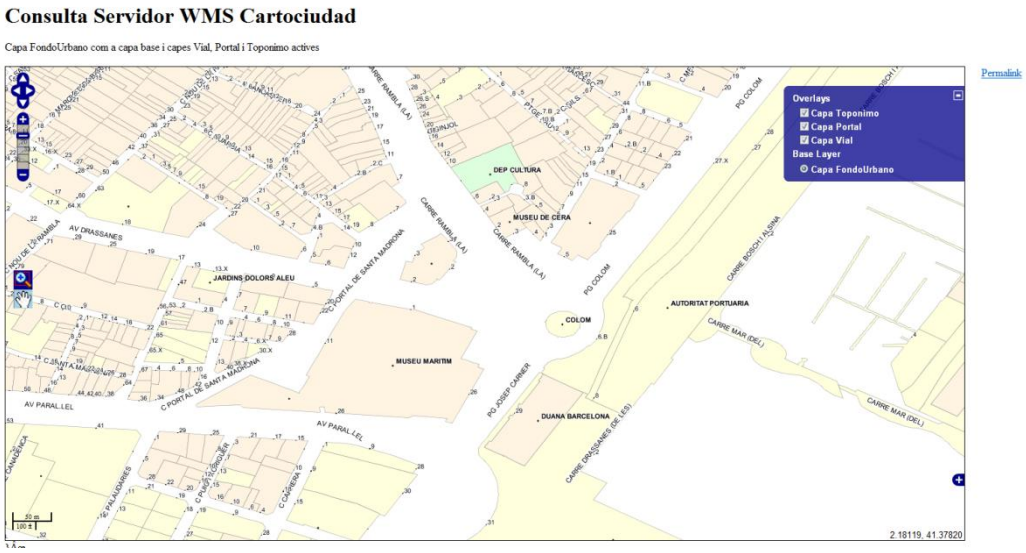
Portal: comprèn l'entitat portal, que inclou tant els portals com els punts kilomètrics (geo:PORTAL_PK).

Toponimo: conté la capa de topònims (geo:TOPONIMO)

El sistema de referència de les dades per a aquest servidor és el ETRS89 en coordenades geogràfiques, amb codi EPSG: 4258. Així mateix, admet entre d'altres el sistema de referència EPSG:4326 (WGS84).

A continuació afegim una captura de pantalla del FondoUrbano com a capa base i les capes Vial, Portal i Toponimo actives:

Figura 2. Consulta Servidor WMS CartoCiudad



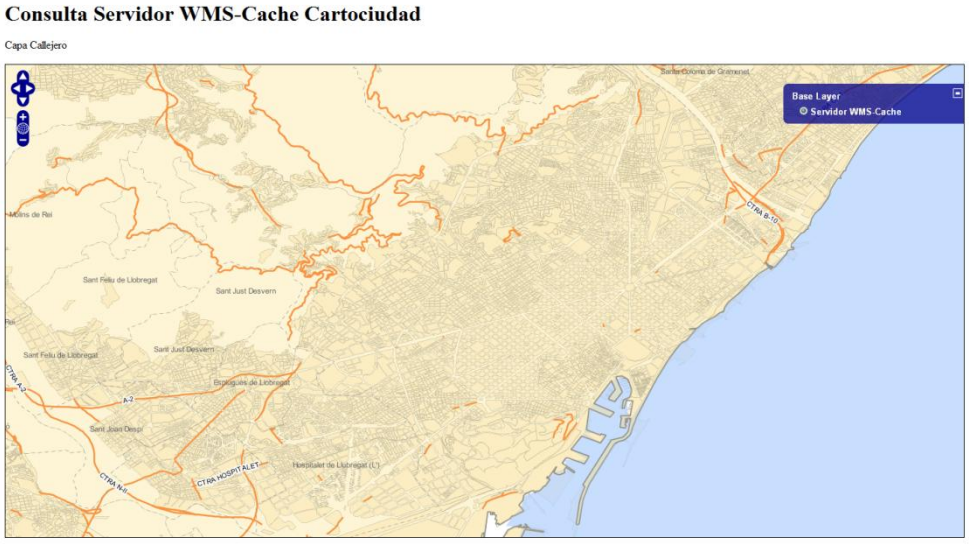
L'altre servidor WMS de que disposa CartoCiudad és un servidor WMS Tessel·lat, denominat WMS-Cache) que permet reduir el temps de resposta del client gràcies a la utilització de la tecnologia TileCache 2.0 de MetaCarta, en particular aquesta tecnologia treballa amb clients com OpenLayers que permeten el tessel·lat, la qual cosa és d'interès per al nostre projecte.

El servidor WMS-Cache es correspon amb la versió 1.0.0 i en concret, entre d'altres, té definida la següent capa d'interès per al projecte:

- Callejero: comprèn les capes del WMS de *DivisionTerritorial, FondoUrbano, Vial, Toponimo i Portal.*

A continuació incloem una captura de pantalla de la consulta al servidor *WMS-Cache* de *CartoCiudad* amb la capa *Callejero* activa.

Figura 3. Consulta Servidor *WMS-Cache CartoCiudad*



5.3.1.2 Servidor WFS de CartoCiudad

El servei de Nomenclàtor de *CartoCiudad* compleix l'especificació *WFS* (*Web Featured Service*) definida per l'OGC. Un servei WFS permet, en general, recuperar i modificar (consultar, inserir, actualitzar i eliminar) dades espacials en format vectorial codificats en *Geography Markup Language* GML. La versió WFS empleada en *CartoCiudad* és la 1.1.0.

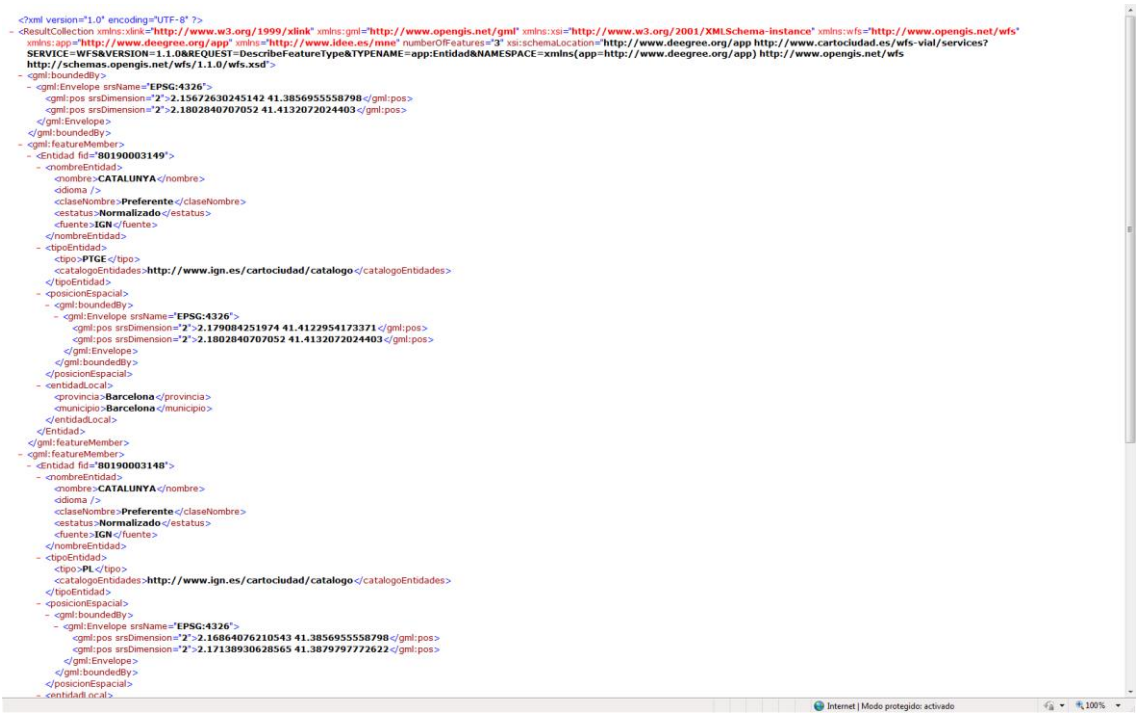
En el cas concret de CartoCiudad els servidors WFS es troben orientats a la localització geogràfica, tenen especial interès per a nosaltres la localització geogràfica a partir del nombre del portal i el nom del carrer (Nomenclàtor de Portal) així com la localització a partir del nom del carrer (Nomenclàtor de vial).

El servidor Nomenclàtor de vial retorna la localització geogràfica (o localitzacions geogràfiques, si més d'un vial compleix els criteris), juntament amb un identificador únic, d'un vial a partir del seu nom, tipus, localitat i província. El paràmetre mínim de consulta és el nom de vial.

A continuació mostrem una consulta i la corresponent resposta per al vial CATALUNYA de Barcelona (com podem observar retorna més d'una entitat: Plaça Catalunya, L'Estatut de Catalunya...):

http://www.cartociudad.es/wfs-vial/services?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&NAMESPACE=xmlns(app=http://www.deegree.org/app)&TYPENAME=app:Entidad&FILTER=<Filter><And><PropertyIsEqualTo><PropertyName>nombreEntidad/nombre</PropertyName><Literal>CATALUNYA</Literal></PropertyIsEqualTo><PropertyIsEqualTo><PropertyName>entidadLocal/municipio</PropertyName><Literal>Barcelona</Literal></PropertyIsEqualTo></And></Filter>

Figura 4. Consulta servidor WFS-vial CartoCiudad: CATALUNYA / Barcelona

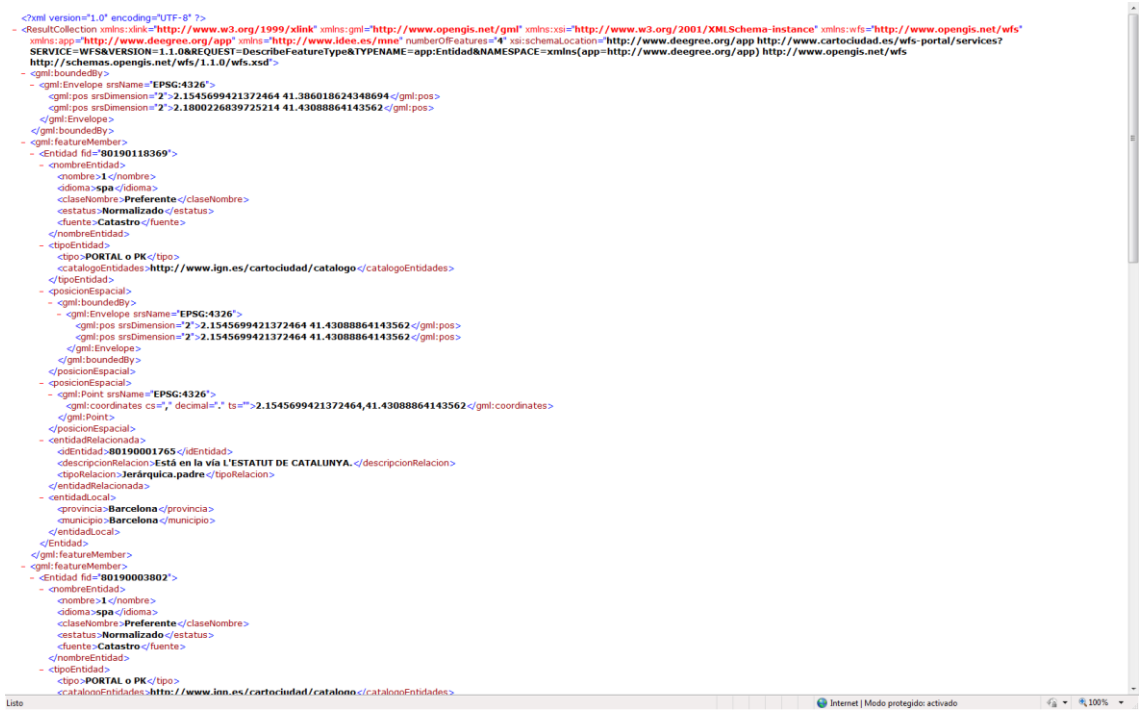


Pel que fa al Nomenclàtor de Portal, podem obtindre la localització geogràfica d'un portal a partir del seu nombre, nom del carrer, municipi i província. En tot cas el resultat pot ser múltiple donat que el Nomenclàtor de Portal no distingeix entre els tipus de vial, per a evitar-ho podem recórrer a obtindre l'identificador únic del vial i a fer servir aquest a la consulta en lloc del seu nom (per això es disposa de la propietat fid de la consulta *WFS-vial*).

A continuació incloem una consulta per al nombre 1 del vial CATALUNYA de Barcelona (també en aquest cas retornarà més d'una entitat):

http://www.cartociudad.es/wfs-portal/services?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&NAMESPACE=xmlns(app=http://www.deegree.org/app)&TYPENAME=app:Entidad&FILTER=<Filter><And><PropertyIsEqualTo><PropertyName>nombreEntidad/nombre</PropertyName><Literal>1</Literal></PropertyIsEqualTo><PropertyIsLike%20wildCard="*"%20singleChar="_"%20escapeChar="!"><PropertyName>entidadRelacionada/descripcionRelacion</PropertyName><Literal>*CATALUNYA*</Literal></PropertyIsLike><PropertyIsEqualTo><PropertyName>entidadLocal/municipio</PropertyName><Literal>Barcelona</Literal></PropertyIsEqualTo></And></Filter>

Figura 5. Consulta servidor WFS-portal CartoCiudad: CATALUNYA, 1 / Barcelona



5.3.2 Recerca de dades a OpenStreetMap

Les dades d'*OpenStretMap* en format *OSM* han estat obtingudes a partir del següent *link*:

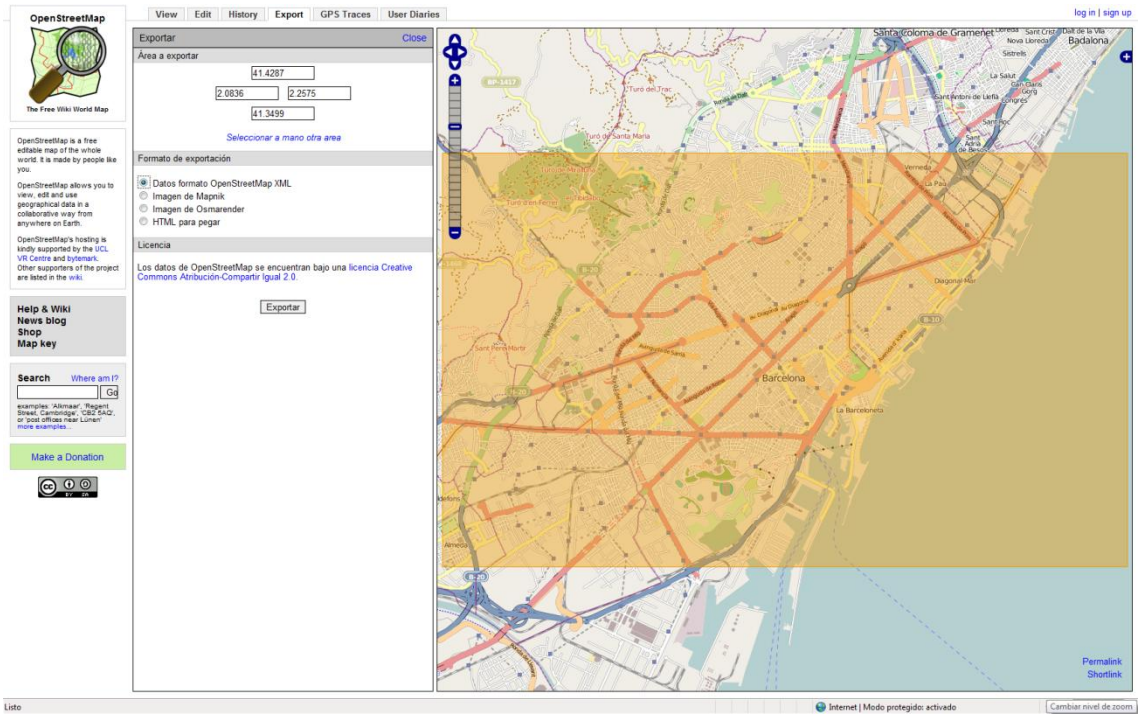
http://www.openstreetmap.org/

Les coordinades dels límits de la importació han estat les següents:

minlat="41.3499" minlon="2.0836" maxlat="41.4287" maxlon="2.2575"

A continuació mostrem una captura de pantalla del marc importat:

Figura 6. Captura pantalla importació OpenStreetMap



5.3.3 Importació de les dades OSM a la geodatabase PostGIS

El principal escull trobat al llarg d'aquest projecte ha estat la importació de les dades. El fet de que es demani explícitament la importació de les dades en format osm planteja una sèrie de qüestions que a continuació passem a exposar.

S'han estat valorant, i provant, tota una sèrie de mecanismes per a la importació de les dades en format osm a la base de dades *PostgreSQL/PostGIS*. Després de moltes de proves hem arribat a la conclusió de que el millor sistema és *osm2pgrouting* donat que és el que millor reprodueix una topologia per a la seva explotació amb *pgRouting*. Només ressenyar entre les diferents proves *osm2pgsql* i *osm2pgr*, en qualsevol dels casos *osm2pgrouting* s'ha postulat com la millor opció.

El millor sistema, doncs, és la importació de les dades amb *osm2pgrouting* per la seva eficàcia en crear la topologia de manera adequada al seu us per *pgRouting*. Això, tanmateix, presenta una problemàtica especial. Es tracta de la no disponibilitat del programari *osm2pgrouting* compilat per a sistemes operatius *Windows*. Com a solució de compromís, s'ha optat per fer la importació de les dades a sobre una base de dades *PostgreSQL/PostGIS/pgRouting* a sobre *Ubuntu 9.10* per a després fer

l'exportació de la base de dades així construïda en format de text planer i posterior importació a sobre *PostgreSQL/PostGIS/pgRouting* a sobre *Windows 7*.

Aquest és el motiu pel qual hem hagut de fer dues instal·lacions de *PostgreSQL/PostGIS/pgRouting*, una primera a sobre *Ubuntu 9.10* suportat per una màquina virtual *VirtualBox* i una segona a sobre *Windows 7*. La primera instal·lació ens ha servit per a la importació de les dades i posterior exportació de la base de dades així obtinguda que a posteriorment ha estat importada des de la base de dades de *Windows 7*.

El resultat de la importació son bàsicament dues taules de interès per a la topologia: per una banda tenim la taula vertices_tmp que conté un identificador únic (enter positiu) per a cada vèrtex involucrat a la topologia així com un camp de tipus geomètric.

La segona taula involucrada és la taula ways que conté arestes, una per a cada segment de carrer delimitat per dues interseccions.

La taula ways conté els següents camps d'interès per a cada una de les possibles funcions de pgRouting:

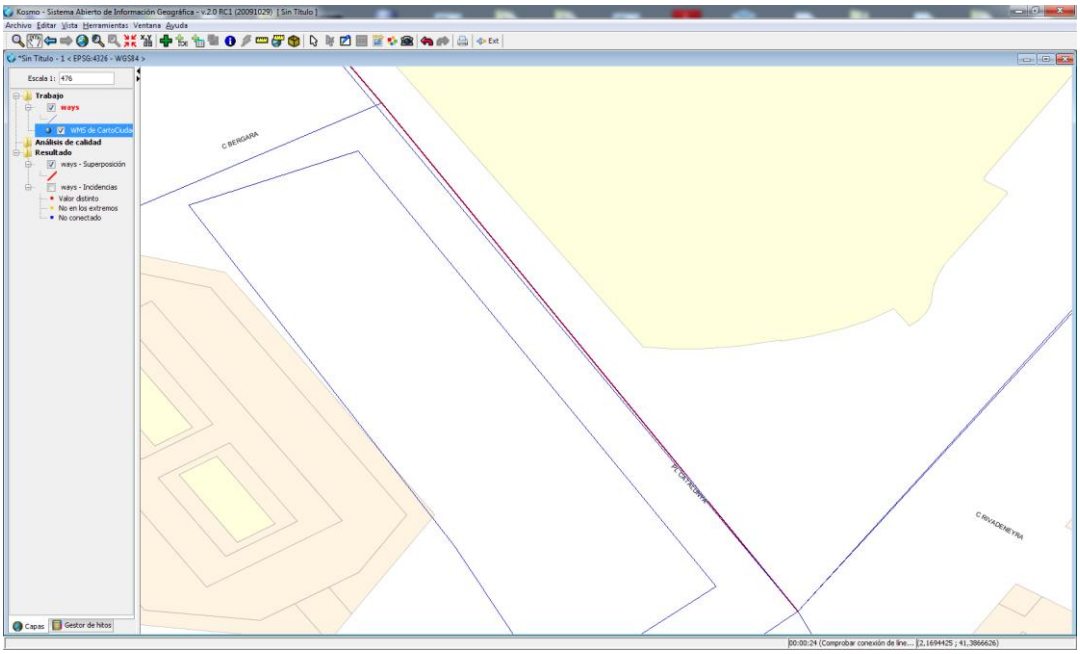
- Per a l'algoritme de Dijikstra trobem:
  - gid: un identificador únic de tipus enter per a l'aresta.
  - source: un identificador de tipus enter per al vèrtex origen.
  - target: un identificador de tipus enter per al vèrtex destinació.
  - cost: un valor de tipus doble precisió per al cost de travessar l'aresta.
  - reverse_cost (opcional): és el cost per a travessar l'aresta en sentit contrari.
- Per a l'algoritme A-star a més a més hem d'afegir:
  - x1: coordinada x del vèrtex d'inici de l'aresta.
  - y1: coordinada y del vèrtex d'inici de l'aresta.
  - x2: coordinada x del vèrtex destinació de l'aresta
  - y2: coordinada y del vèrtex destinació de l'aresta.
- Per a Shotting Star a més a més:
  - rule: una cadena amb una llista de vèrtexs separada per comes que estableix una regla de tipus: "si procedim dels vèrtexs indicats podem passar per l'aresta amb un cost igual a l'indicat en la columna to_cost".
  - to_cost: el cost de passar d'acord amb la regla establerta.

Una vegada importades les dades a sobre *Windows 7*, hem pogut comprovar que la qualitat de les mateixes no resulta especialment bona, la qual cosa no té res d'estrany per a unes dades obtingudes de manera col·laborativa a on el control de qualitat resulta especialment complicat.

Per a corregir la topologia hem fet servir les possibilitats de *Kosmo*, en primer lloc hem fet servir l'opció de comprovació d'errors de línies que ens ha detectat dos tipus d'errors:

Vies superposades, tal qual podem comprovar a la següent captura de pantalla:

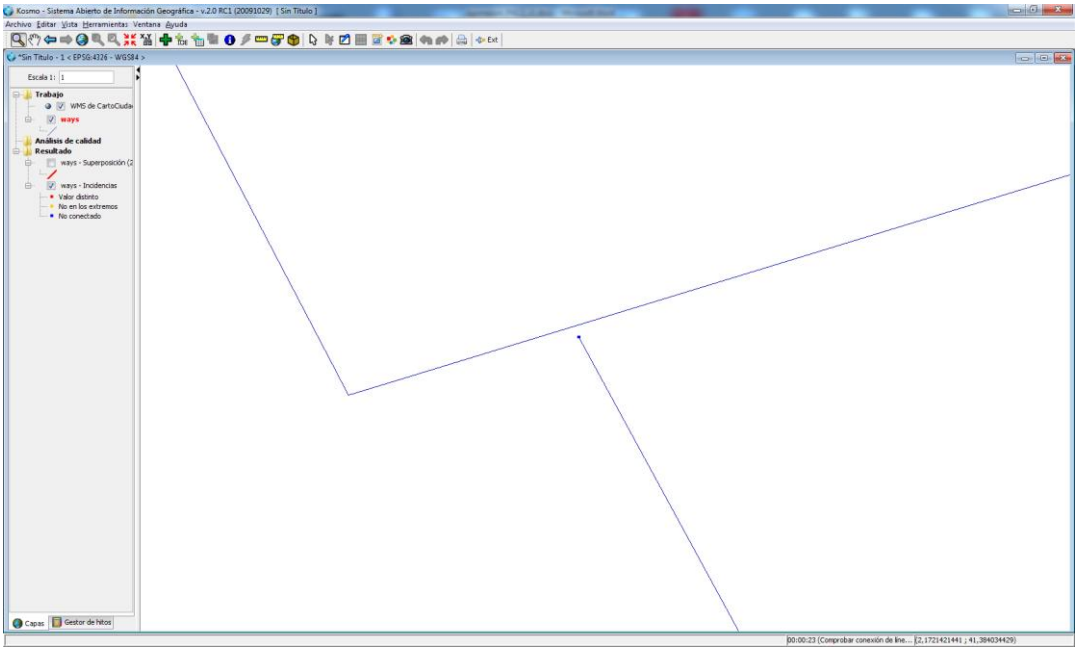
Figura 7. Vies superposades



Que caldrà corregir.

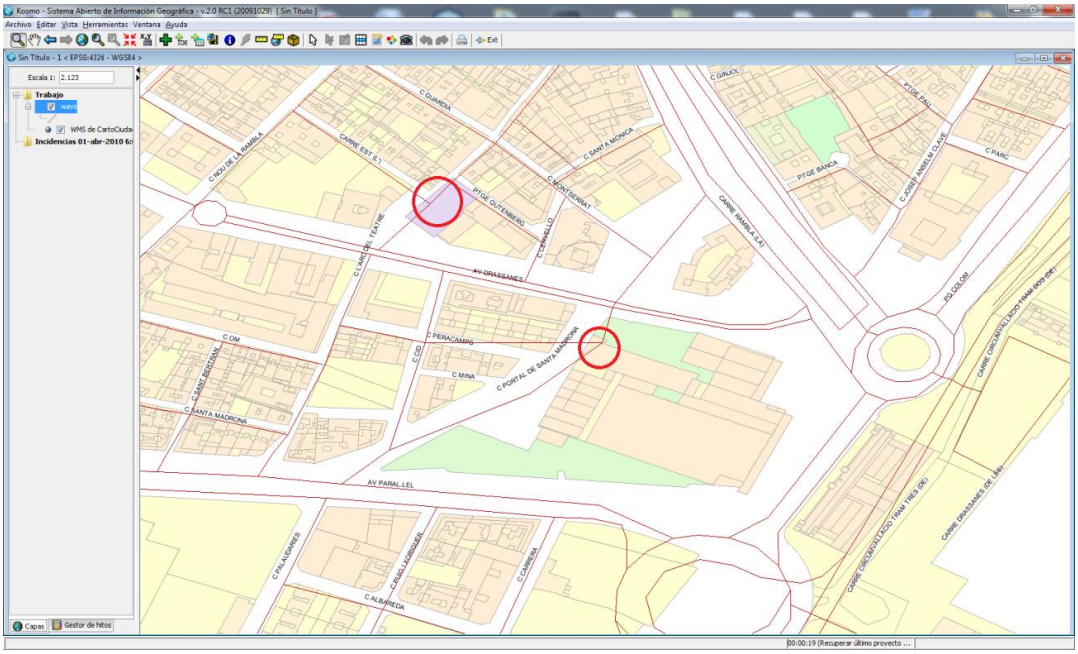
I per una altra banda ens indicarà aquells punts als quals podem trobar línies que no connecten, que hem anomenat «culs de sac». En particular en el nostre cas *Kosmo* ens indicarà els extrems sense enllaçar en color blau:

Figura 8. Culs de sac



Per una altra banda podem trobar rutes que es desvien de les traces naturals pels carrers i passen pel damunt d'edificacions. Això ho podem comprovar fent servir *Kosmo GIS* al qual establirem una connexió WMS a les capes vial i *fondo urbano*, a la següent captura de pantalla podem observar algunes de les errades d'aquest tipus:

Figura 9. Errades geomètriques



Caldrà, doncs, corregir aquestes anomalies a sobre el mateix *Kosmo GIS* que en establir una connexió directa a sobre la base de dades, reflectirà a sobre la base de dades les modificacions que a nivell gràfic realitzem a sobre la capa en qüestió.

5.4 Estudi de programari anterior

Diuen que no cal tornar a inventar la roda a cada pas. Per aquest motiu i abans de començar el nostre disseny hem decidit fer l'estudi d'una proposta de sistema de càlcul d'itineraris que apareix com a exemple a la mateixa plana web de pgRouting i que va ser presentat al FOSS4G 2007 celebrat a Victoria (Illa de Vancouver, Canadà) amb el títol *Web-based Routing: An Introduction to pgRouting with OpenLayers*.

El programari proposat es troba disponible al següent link:

http://pgrouting.postlbs.org/raw-attachment/wiki/WorkshopFOSS4G2007/20071004_tutorial.zip

i l'hem descarregat i adaptat a les nostres necessitats. Bàsicament aquest treball proposa dos arxius, per una banda un arxiu HTML per a la interfície d'usuari que inclou codi javascript per a la lògica de la interfície, i per una altra banda un arxiu de codi php per al càlcul de rutes.

La interfície de l'usuari permet triar entre tres mètodes de càlcul d'itineraris:

- Shortest Path Dijkstra: basat en la funció dijkstra_sp_delta (no direccional).
- Shortest Path A Star: basat en la funció astar_sp_delta (no direccional)
- Shortest Path Shooting Star: basat en la funció shootingstar_sp (que tot i que és direccional es fa servir com a no direccional en aquest cas).

Cal comentar que els dos primers algoritmes (dijkstra_sp_delta i astar_sp_delta) son algoritmes de càlcul d'itineraris basats en vèrtexs, mentre que el darrer es basat en arestes.

Pel que fa a la funcionalitat del càlcul d'itineraris la podem dividir en dues parts, una primera basada en una consulta SQL que ens permet trobar l'aresta més propera a un punt (que és aplicada per a determinar les arestes d'origen i destinació) i una altra que es correspon amb els diferents algoritmes de càlcul d'itineraris anteriorment comentats i que es basa també en una consulta SQL (distinta per a cada mètode de càlcul d'itineraris). Aquesta divisió de la funcionalitat sembla aprofitable per als nostres interessos, encara que com veurem més endavant serà sotmesa a importants modificacions.

Per al càlcul dels itineraris el sistema determina en primer lloc les arestes més properes al punt origen i al punt final. A partir d'aquí, en el cas dels dos primers mètodes, aplica els respectius algoritmes subjacents fent servir com a **vèrtex inicial** el corresponent **vèrtex final de l'aresta inicial** i com a **vèrtex final** el

corresponent **vèrtex inicial de l'aresta final**. Pel cas del darrer mètode simplement es fan servir les **arestes inicial i final trobades**.

En tot cas cap d'ells ens serviria, per a les nostres necessitats, pels motius que a continuació veurem.

Per la nostra banda i a tall d'estudi hem adaptat el codi a les nostres dades, sense modificar de cap manera el sistema de càlcul d'itineraris, de manera que puguem fer un estudi detallat dels seus avantatges i mancances.

El resultat de tot això és que com a primera mancança important, detectem el fet de que tal qual estan plantejats, cap dels mètodes proposats és direccional (no tenen en compte la possibilitat de que el cost de recórrer una aresta variï en funció del sentit en que aquesta es recorre). És a dir, els dos primers mètodes estan basats en algoritmes no direccionals mentre que el segon, tot i que te a sota seu un algoritme direccional, s'aplica en aquest cas en la seva forma no direccional.

Una altra mancança a considerar és la que podem observar a la **Figura 10 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben després dels punts origen i destinació** que ens mostra una captura de pantalla d'un càlcul d'itineraris entre dos punts fent servir el mètode Shortest Path Dijkstra a on podem veure un efecte curiós, els itineraris comencen abans del punt inicial i acaben després del punt final.

També podem trobar l'altre extrem, és a dir l'itinerari comença abans del punt origen i acaba abans del punt destinació, tal qual es pot observar a la **Figura 11 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben abans dels punts origen i destinació**.

Aquest comportament que en principi crida l'atenció s'explica amb facilitat si revisem el codi php, al qual podem comprovar que per al cas dels dos primers algoritmes és fan servir per al càlcul de les rutes el punt final de l'aresta d'origen (target de l'aresta origen) i el punt inicial de l'aresta de destinació (source de l'aresta destinació). En tractar-se de càlcul d'itineraris no dirigits, resultarà que si el camí més curt no inclou l'aresta de partida i/o l'aresta d'arribada, els corresponents punts d'origen/destinació quedaran aïllats en relació als corresponents itineraris. Per una altra banda, si el camí més curt inclou les arestes corresponents, els punts de sortida i/o arribada quedaran inclosos a mig camí de l'aresta corresponent.

Pel que fa al tercer algoritme, les arestes de partida i arribada sempre seran incloses al trajecte i per tant, en tots els casos els punts origen i destinació quedaran inclosos a dintre del trajecte i no a l'origen i la destinació del mateix, tal qual seria d'esperar.

Figura 10 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben després dels punts origen i destinació

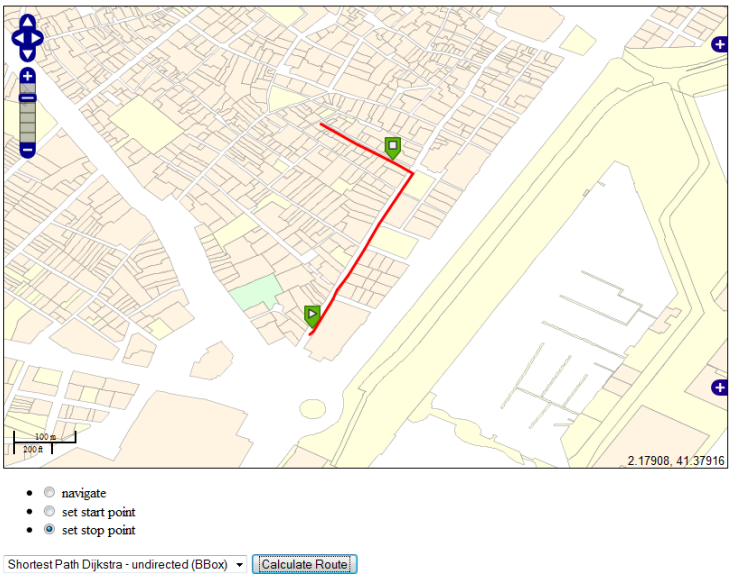
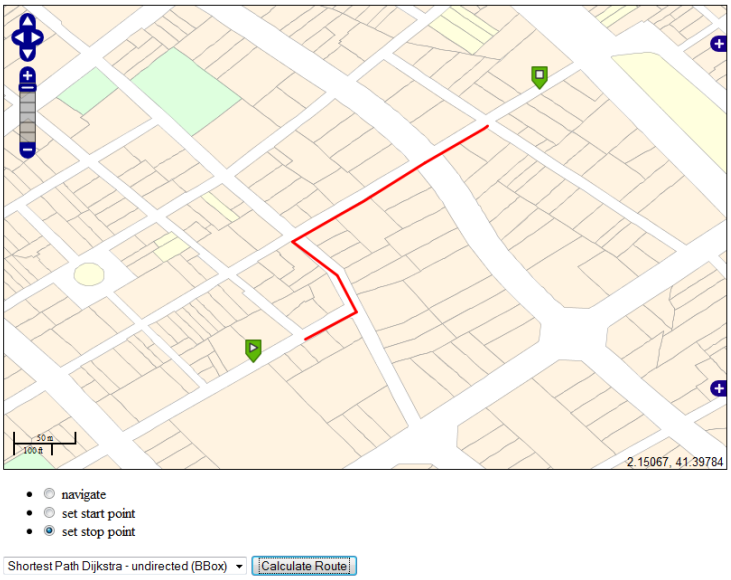


Figura 11 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben abans dels punts origen i destinació



Tret dels casos anteriorment esmentats, també és poden trobar situacions mixtes i inclòs situacions excepcionals a les quals els punts origen i destinació coincideixin, de manera casual, amb el punt final i el punt inicial de les arestes respectives, amb la qual cosa els itineraris semblaran mostrar-se d'una manera acurada.

Resumint aquest apartat podem dir:

- El programari estudiat no s'adapta completament a les nostres necessitats donat que és no direccional, cosa que en el cas de càlcul d'itineraris de vehicles és un punt crític.

- Els càlculs dels camins més curts no son acurats donat que no es basen en els punts d'origen i destinació sinó en les arestes a les quals es troben inclosos (punt final de l'aresta origen / punt inicial de l'aresta destinació o aresta origen / aresta destinació), la qual cosa en cas de rutes per a vianants i vies llargues pot arribar a ser crític. Per una altra banda la decisió de cap a quin dels dos extrems d'un carrer s'ha de partir o per quin dels dos extrems del carrer de destinació s'ha d'arribar en el cas de carrers bidireccionals pot així mateix arribar a ser crítica.

5.5 Plantejament de les necessitats per al càlcul d'itineraris

En el nostre cas hem decidit facilitar als usuaris la possibilitat de triar entre el càlcul d'itineraris per a vianants o per a vehicles.

La diferència entre els dos mètodes de càlcul d'itineraris radicarà en que per una banda els vianants no poden circular per vies ràpides (vies de cintura, autopistes) i per l'altra, que els vehicles es veuen sotmesos a una sèrie de limitacions pel que fa als sentits de recorregut de determinades vies (aquelles d'un sol sentit) mentre que els vianants no tenen aquesta limitació.

Cal comentar que aquesta diferenciació resulta una mica grossera en funció de les dades d'OpenStreetMap disponibles donat que, com es veurà més endavant, les dades es troben capturades pensant en la navegació en vehicle, no tenint en compte semàfors per a vianants, la qual cosa fa que per a un canvi de sentit sigui necessari en alguns casos recórrer primer el carrer en sentit oposat fins a arribar a una cruïlla que permeti aquest canvi.

No obstant, considerem això admissible donat que estem parlant d'un treball més aviat de caire teòric encara que, de tota manera, no resultaria complicat adaptar el programari que sortirà d'aquí per a fer servir dades reals, ja sigui fent servir taules diferents per a vianants i vehicles o bé incloent ambdues geometries a una mateixa taula i assegurant-se de que les geometries queden adequadament classificades.

Interessa que els camins siguin realment mínims i que la seva representació a sobre el mapa sigui el més acurada possible, és a dir, l'itinerari comenci al punt de partida(i no en un extrem del carrer corresponent) i acabi al punt d'arribada (i no a un extrem del carrer d'arribada).

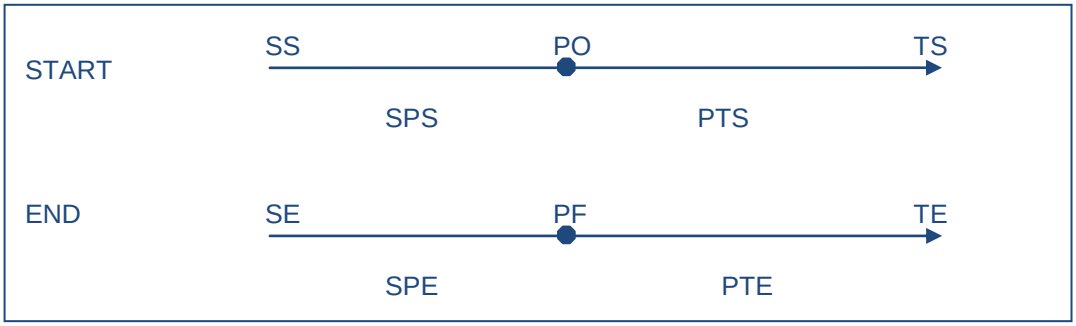
5.6 Estratègia utilitzada

Dintre d'aquest apartat passarem a definir la nomenclatura que farem servir al llarg del mateix, així com a l'estudi teòric, per a continuació centrar-nos en la dificultat del càlcul d'itineraris acurats que ens conduirà a haver de definir l'algoritme de base a utilitzar i a definir l'estratègia a utilitzar per a resoldre el nostre problema.

5.6.1 Nomenclatura

Ens caldrà definir una nomenclatura per a la dissertació que ve a continuació, per això, anem a considerar una aresta dirigida A amb un punt d'origen S i un punt de destinació T, per una altra banda considerarem a sobre aquesta aresta un punt P situat a una distància SP de l'origen i PT de la destinació. Si apliquem això a les arestes i punts d'origen i destinació tindrem el que es resumeix al gràfic següent:

Figura 12. Nomenclatura utilitzada a l'estudi teòric



A on:

- START: aresta origen
- SS : punt d'origen de l'aresta d'origen
- TS : punt de destinació de l'aresta d'origen
- PO :punt d'origen del recorregut
- SPS : distància origen aresta origen al punt d'origen
- PTS : distància punt origen al punt destinació de l'aresta d'origen
- END: aresta destinació
- SE : punt d'origen de l'aresta final
- TE : punt de destinació de l'aresta final
- PF : punt final del recorregut
- SPE : distància origen aresta final al punt final
- PTE : distància punt final al punt destinació de l'aresta final

5.6.2 Dificultat del càlcul

Cal comentar que pgRouting disposa de dos tipus d'algoritmes base:

- Els basats en vèrtexs als qual es calcula el camí mínim entre dos vèrtexs donats.
- Els basats en arestes als quals es calcula el camí mínim entre dues arestes donades que son incloses sempre a l'itinerari.

Entre els dos primers tenim el shortest_path i el shortest_path_astar i del segon tipus tenim el shortets_path_shooting_star

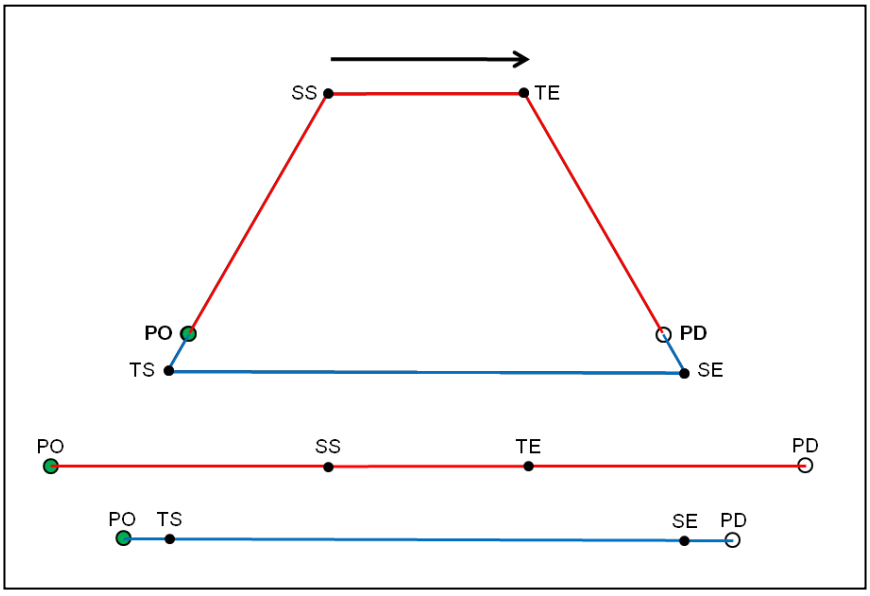
Anem ara a considerar la xarxa que es mostra a la **Figura 13 Dificultat del càlcul** i que consta de quatre arestes. A sobre l'aresta esquerra tenim el nostre punt origen i a sobre la dreta el punt destinació.

Si apliquéssim un algoritme base basat en arestes per al càlcul del recorregut mínim, resulta obvi que l'itinerari inclouria l'aresta inicial, l'aresta final i l'aresta marcada per la fletxa, com podem comprovar tindríem un excés de recorregut que hauríem de corregir.

Pel que fa als recorreguts basats en vèrtexs, tindríem que ens cal fixar un vèrtex inicial i un vèrtex final. Si per exemple adoptem el criteri del codi estudiat anteriorment, es a dir SS→ TE, tindríem que el camí mínim inclouria també l'aresta marcada per la fletxa.

Per tant fent les correccions oportunes el camí seleccionat seria, en tots dos casos, el que apareix en color vermell, no obstant, si desenvolupem els dos itineraris alternatius, podem observar que el camí seleccionat no es el més curt.

Figura 13 Dificultat del càlcul



5.6.3 Selecció de l'algoritme de base

Per a les nostres necessitats hem estat valorant quin tipus d'algoritme bàsic podia resultar més adient per als nostres interessos, hem descartat l'algoritme basat en arestes (shortest_path_shoting_star) donat que una vegada que l'algoritme tria el recorregut mínim entre les arestes inicial i final, no hi ha manera de comprovar un itinerari alternatiu de manera que, es pot donar el cas de que el recorregut mínim entre les arestes inicial i final sigui un donat, però per una altra banda, pot ser que els punts d'origen i destinació es trobin situats més a prop dels altres vèrtexs, amb la qual cosa l'itinerari mínim acurat real podria no ser el que calcula l'algoritme.

Pel que fa als algoritmes basats en vèrtexs, podem dir que son adequats perquè ens permeten calcular itineraris alternatius simplement triant vèrtexs alternatius. Per exemple si considerem els vèrtexs TS i SS a la figura anterior, el resultat de càlcul ens donaria l'aresta alternativa i per tant ens permetria determinar l'itinerari alternatiu que seria correcte.

Tindrem doncs quatre possibles itineraris mínims segons considerem els extrems de sortida de l'aresta inicial i d'arribada a la final. Si poguéssim calcular aquets quatre recorreguts, estaríem en condicions de determinar quin d'ells és el mínim. Això ho podrem fer amb un algoritme basat en vèrtexs com veurem a continuació. En concret la nostra tria ens ha conduït a l'algoritme shortest_path que és un algoritme ben provat i per al qual no existeix cap ambigüitat en relació al seu funcionament: es detalla un vèrtex d'origen, un de destinació i és calcula el camí més curt considerant un graf dirigit, i un cost invers per a cada aresta (cas de seleccionar les corresponents opcions).

5.6.4 Estratègia

Com ja s'ha comentat, per a aquest estudi considerarem que el cas del càlcul d'itineraris per a vianants és un cas particular del càlcul d'itineraris per a vehicles al qual totes les arestes son bidireccionals, de manera que ens centrarem en aquest darrer.

L'estratègia proposada passa per les següents fases:

a) Considerar totes les **possibles** combinacions de vèrtexs (per al cas de dues arestes bidireccionals: SS→SE, SS→TE, TS→SE i TS→TE).
b) Determinar els recorreguts per shortest_path per a aquells itineraris possibles.
c) Fer les oportunes correccions per als recorreguts a sobre les arestes.
d) Determinar quin dels quatre es l'itinerari mínim.

5.7 Estudi teòric

Considerem dues arestes **bidireccionals**, d'una xarxa viària, a sobre les quals es troben uns respectius punts origen i destinació d'un itinerari per al qual es vol determinar el camí mínim a partir de l'algoritme shortest_path. Si considerem el punt de partida des de l'aresta origen i el d'arribada a l'aresta destinació, tindrem 4 possibles combinacions.

Ara bé, l'algoritme de shortest_path pot presentar quatre situacions diferents pel que fa a la inclusió o no al recorregut calculat de les arestes extremes, aquestes son les següents:

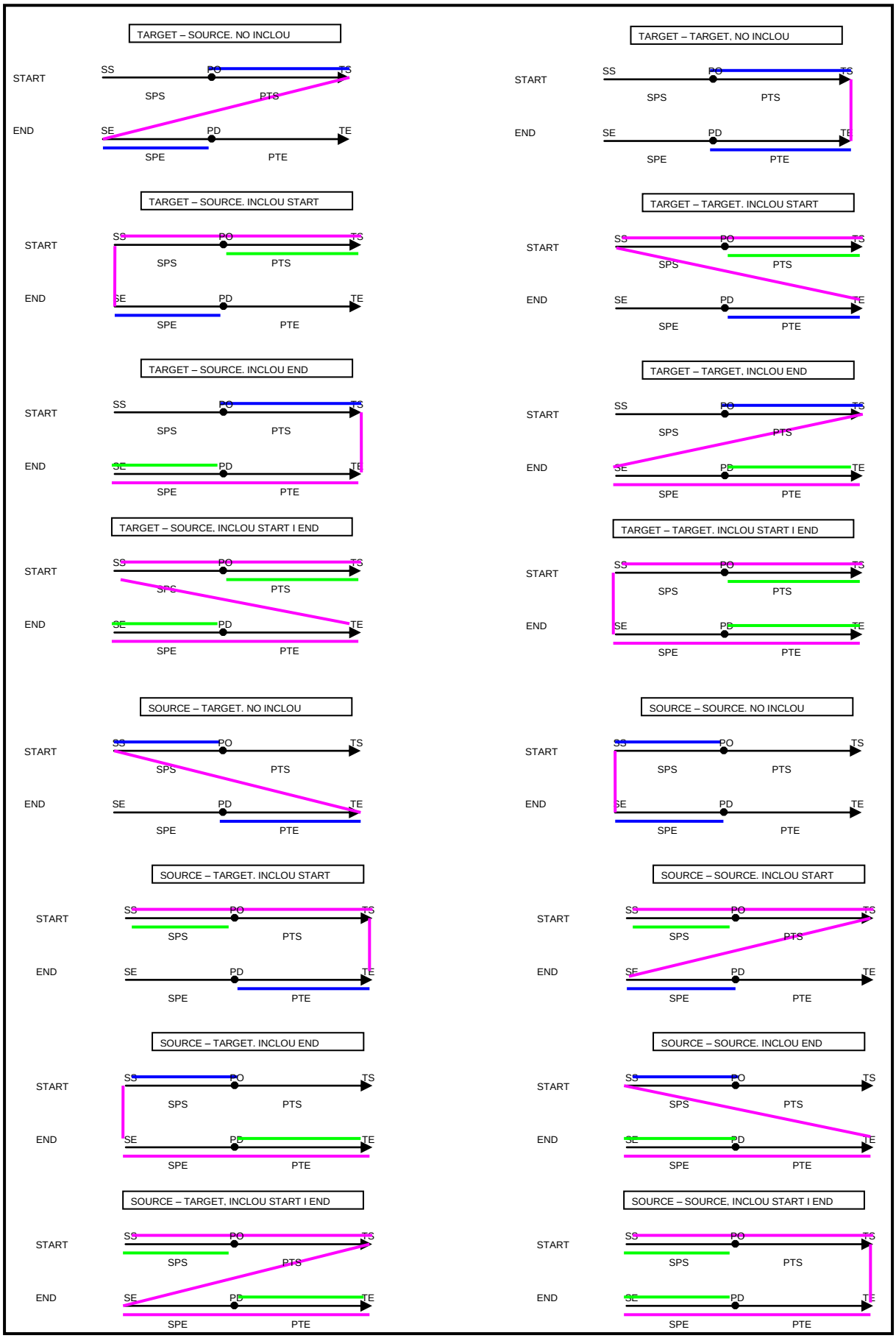
- No inclou S ni E.
- Inclou S però no E.
- No inclou S però inclou E.
- No inclou ni S ni E.

De manera que en realitat tindríem 4*4= 16 possibles itineraris que han quedat recollits a **la Figura 14 Possibles recorreguts entre dos punts situats a sobre dues arestes per shortest_path i addicions** juntament amb les correccions necessàries per a cada cas per a obtindre la mida de l'itinerari entre els punts PO i PD.

A la figura esmentada, hem ressenyat:

- El recorregut del shortest_path en color magenta.
- Els valors a afegir per als trams no comptabilitzats a sobre les arestes inicial i final en color blau.
- L'excés de recorregut en color verd.

Figura 14 Possibles recorreguts entre dos punts situats a sobre dues arestes per shortest_path i addicions



Ara bé, si considerem les arestes extremes d'un itinerari a sobre una xarxa viària podem considerar quatre possibles combinacions de bidireccionalitat:

1. L'aresta origen és unidireccional i l'aresta destinació unidireccional.
2. L'aresta origen és bidireccional i la de destinació unidireccional.
3. L'aresta origen és unidireccional i la de destinació bidireccional.
4. L'aresta origen és bidireccional i la de destinació és bidireccional.

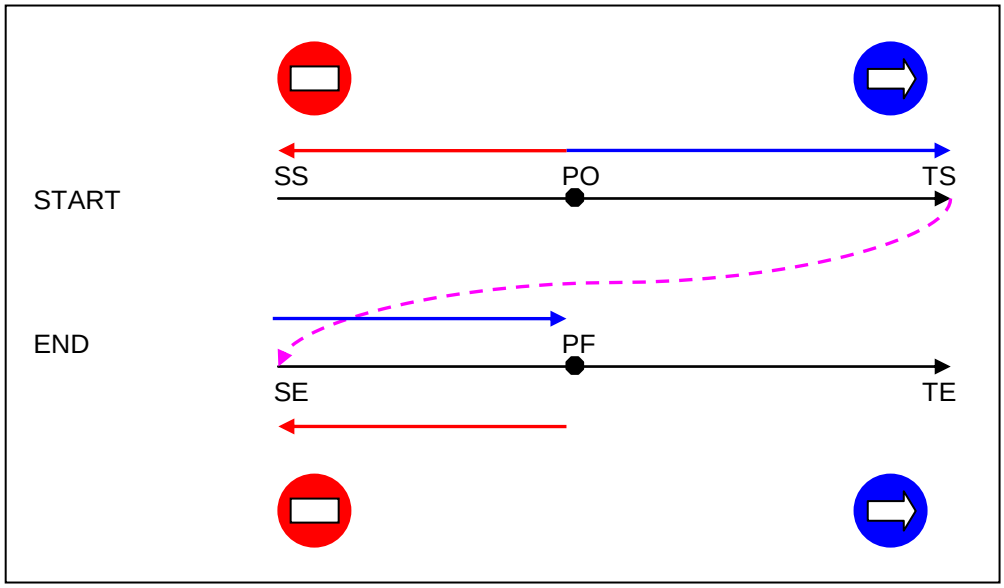
Resulta obvi que el cas del vianant es correspon amb el darrer d'aquests quatre casos, donat que tal com ho s'ha definit a l'apartat **5.6.4 Estratègia** aquest és no direccional i al cas esmentat totes les arestes serien bidireccionals i en particular les d'origen i destinació.

Anem a continuació a estudiar quines de les quatre combinacions de vèrtexs de les arestes inicial i final te sentit considerar per al càlcul del camí mínim en funció de la bidireccionalitat.

Pel que fa al cas dels vehicles i, a tall d'exemple, passarem a estudiar detalladament les dues primeres possibilitats, és a dir: a) L'aresta origen és unidireccional i l'aresta destinació és unidireccional i b) L'aresta origen és bidireccional i la de destinació unidireccional.

a) L'aresta origen es unidireccional i la de destinació unidireccional:

Figura 15 Itinerari TS→SE, únic possible entre dues arestes unidireccionals



A la **Figura 15 Itinerari TS→SE, únic possible entre dues arestes unidireccionals** es representen:

- El recorregut del shortest_path en color magenta.
- Els valors a afegir per als trams no comptabilitzats a sobre les arestes inicial i final en color blau.

D'acord amb la figura anterior, per a dues arestes unidireccionals, només tenim un itinerari mínim possible TS→SE. Això es així donat que només podem recórrer les arestes en els sentits permesos, de manera que no seria possible recórrer per exemple l'aresta start en el sentit PO→SS.

Es podria objectar que resulta possible el recorregut SS→TS→SS, no obstant en fer les correccions pertinents ens quedaria el mateix que per a TS→SE, de manera que només caldrà considerar-ne aquest darrer.

b) L'aresta origen és bidireccional i la de destinació unidireccional:

Anem ara a considerar el cas en que l'aresta origen sigui bidireccional i la de destinació unidireccional. D'acord amb l'estratègia anteriorment esmentada, tindríem dos recorreguts mínims:

a. El recorregut SS→SE
b. El recorregut TS→SE.

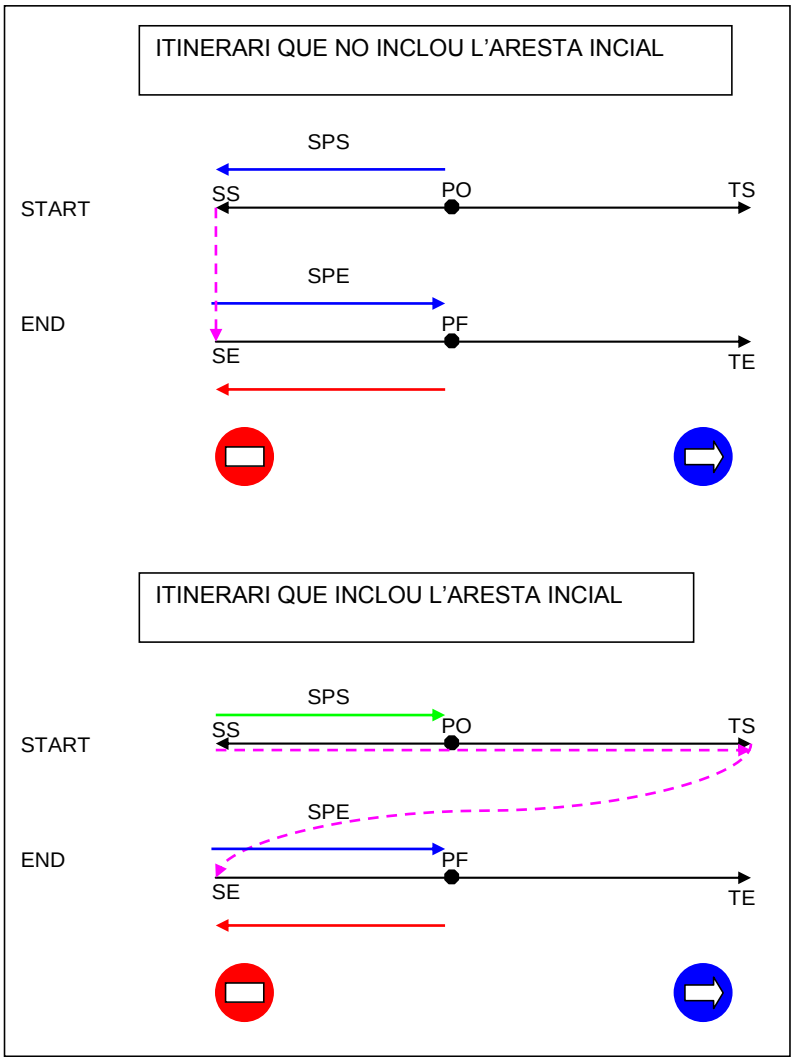
Passem a analitzar detingudament aquests casos:

a. Recorregut SS→SE

A la **Figura 16. Possibles itineraris SS→SE per al cas d'aresta inicial bidireccional i final unidireccional** podem observar representats:

- El recorregut del shortest_path en color magenta.
- Els valors a afegir per als trams no comptabilitzats a sobre les arestes inicial i final en color blau.
- L'excés de recorregut en color verd.

Figura 16. Possibles itineraris SS→SE per al cas d'aresta inicial bidireccional i final unidireccional



Si observem la figura anterior, en principi només caldria afegir a la mida del recorregut entre SS i SE la distància entre el punt origen i el punt origen de l'aresta origen i la distància entre el punt destinació i el punt origen de l'aresta destinació (es a dir SPS+SPE). No obstant ara hem de considerar un cas que no calia considerar per al cas de dues arestes unidireccionals, es tracta de la possibilitat de que l'itinerari calculat per l'algoritme shortest_path inclogui l'aresta inicial. Si tenim en compte això, tindríem per una banda que ens caldria afegir la distància entre el punt inicial de l'aresta final i el punt final SPE i que per una altra banda s'estaria incloent un excés de recorregut que seria la distància entre el punt inicial de l'aresta inicial i el punt d'origen, és a dir SPS que hauríem de restar al recorregut anterior, per tant el recorregut net total a sobre les arestes, entenent aquest

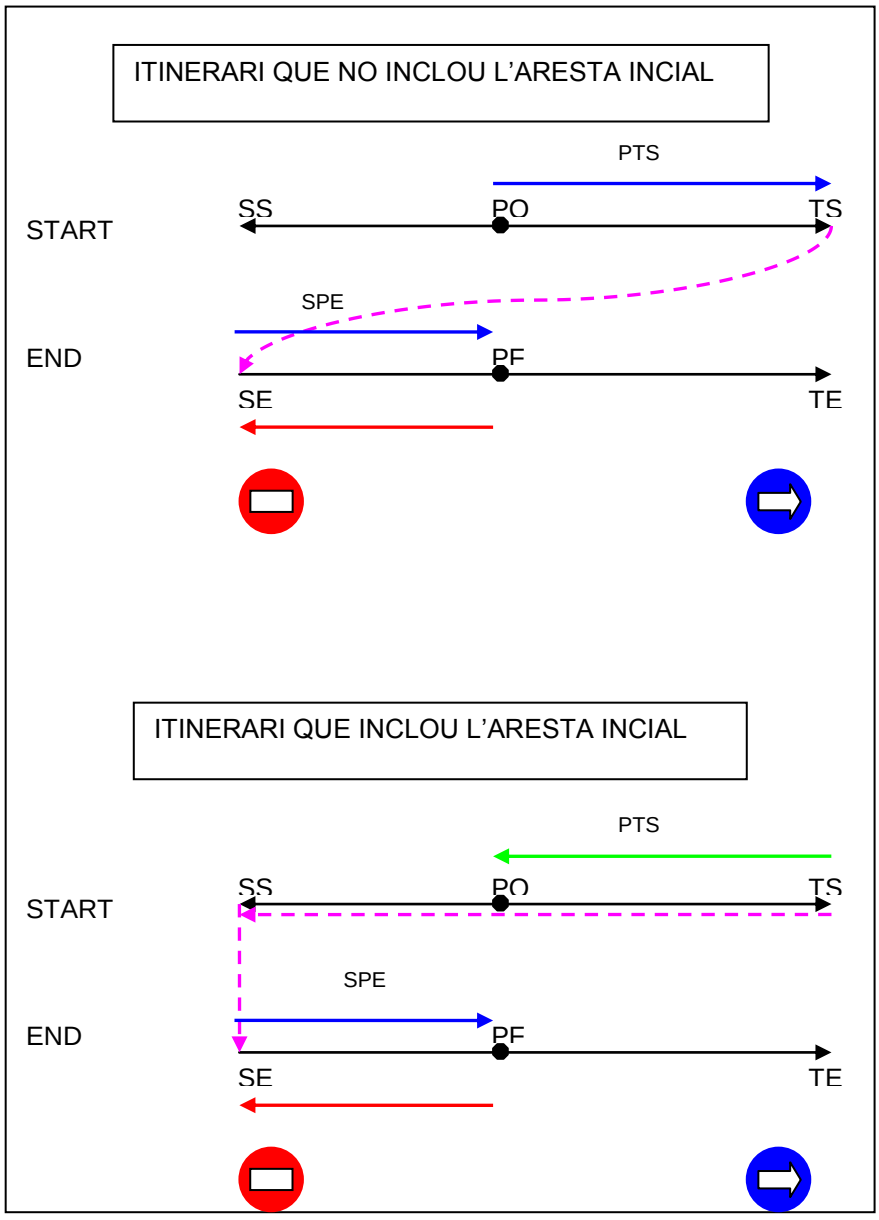
com la distància que s'hauria d'afegir al càlcul del shortest_path per a obtenir el recorregut entre el punt inicial i el punt final, seria SPE-SPS.

b. Recorregut TS→SE

A la **Figura 17 Possibles itineraris TS→SE entre una aresta inicial bidireccional i una final unidireccional** podem observar representats:

- El recorregut del shortest_path en color magenta.
- Els valors a afegir per als trams no comptabilitzats a sobre les arestes inicial i final en color blau.
- L'excés de recorregut en color verd.

Figura 17 Possibles itineraris TS→SE entre una aresta inicial bidireccional i una final



A la figura anterior i per al recorregut TS→SE, podem observar una situació anàloga a la del recorregut SS→SE, amb recorreguts nets PTS+SPE (en el cas de que l'aresta inicial no es trobi inclosa) i SPE-PTS (per al cas de que l'aresta inicial es trobi inclosa).

A la **Taula 4. Trajectes possibles en funció de la bidireccionalitat de les arestes inicial i final** s'han resumit els recorreguts a considerar en funció de la bidireccionalitat de les arestes.

Aquesta taula representa un algoritme que permet determinar els recorreguts que cal considerar entre els vèrtexs d'una aresta inicial i una altra final en funció de la bidireccionalitat de cada una de les dues arestes. Aquest algoritme ha estat implementat, com es veurà més endavant a la funció barcelona_routing

Taula 4. Trajectes possibles en funció de la bidireccionalitat de les arestes inicial i final

SENTITS		
START	END	Trajectes possibles
Simple	Simple	TS→ SE
Doble	Simple	SS→ SE
		TS→ SE
Simple	Doble	TS→ SE
		TS→ TE
Doble	Doble	SS→ SE
		SS→ TE
		TS→ SE
		TS→ TE

Si ara combinem **Figura 14 Possibles recorreguts entre dos punts situats a sobre dues arestes per shortest_path i addicions** amb l'algoritme anterior, arribarem a la **Taula 5 Recorregut net total a sobre les arestes inicial i final en funció de la bidireccionalitat d'aquestes**.

Ara bé, el que resulta de destacar d'aquesta taula és el fet de si ens hi fixem els recorreguts nets totals a sobre les arestes inicial i final son una funció del trajecte i de la inclusió o no de les arestes origen i destinació, actuant els sentits de les arestes com un factor limitador que permet la seva inclusió o no en funció de la seva bidireccionalitat , la qual cosa, com sabem, ve controlada per l'algoritme shortest_path i per tant no ens hem de preocupar.

Taula 5 Recorregut net total a sobre les arestes inicial i final en funció de la bidireccionalitat d'aquestes

SENTITS			Recorregut net total a sobre les arestes inicial i final			
START	END	Trajecte	No Inclou	Inclou S	Inclou E	Inclou S i E
Simple	Simple	TS→ SE	PTS + SPE	NO	NO	NO
Doble	Simple	SS→ SE	SPS + SPE	SPE - SPS	NO	NO
		TS→ SE	PTS + SPE	SPE - PTS	NO	NO
Simple	Doble	TS→ SE	PTS + SPE	NO	PTS - SPE	NO
		TS→ TE	PTS + PTE	NO	PTS - PTE	NO
Doble	Doble	SS→ SE	SPS + SPE	SPE - SPS	-SPE + SPS	-SPE - SPS
		SS→ TE	SPS + PTE	PTE - SPS	SPS - PTE	-SPS - PTE
		TS→ SE	PTS + SPE	SPE - PTS	PTS - SPE	-PTS - SPE
		TS→ TE	PTS + PTE	PTE - PTS	PTS - PTE	-PTS - PTE

De manera que podem extreure la **Taula 6. Recorreguts nets a sobre les arestes inicial i final en funció del trajecte** que no representa més que un algoritme que ens permet determinar les correccions a efectuar a sobre la mida del recorregut calculat pel shortest_path en funció de la inclusió o no de les arestes i dels vèrtexs presos com a inici i destinació del shortest_path.

Taula 6. Recorreguts nets a sobre les arestes inicial i final en funció del trajecte

	Recorregut nets total a sobre les arestes inicial i final			
Trajecte	No Inclou	Inclou S	Inclou E	Inclou S i E
SS→ SE	SPS + SPE	SPE - SPS	-SPE + SPS	-SPE - SPS
SS→ TE	SPS + PTE	PTE - SPS	SPS - PTE	-SPS - PTE
TS→ SE	PTS + SPE	SPE - PTS	PTS - SPE	-PTS - SPE
TS→ TE	PTS + PTE	PTE - PTS	PTS - PTE	-PTS - PTE

Pel que fa a les geometries del recorregut entre el punt origen i el punt final, a sobre les arestes origen i final, ens podem remetre a la **Figura 14 Possibles recorreguts entre dos punts situats a sobre dues arestes per shortest_path i addicions** i analitzar a tall d'exemple el cas del trajecte SS→SE.

Per una banda tenim el cas en que el shortest_path no inclou cap de les arestes inicial i final, resulta obvi que caldria afegir les geometries corresponents a unes hipotètiques arestes SPS i SPE.

Pel cas Inclou S, podem observar que caldria afegir l'aresta SPE i per una altra banda substituir l'aresta START per l'aresta complementaria de SPS, és a dir PTS que resumint ens quedaria –START + SPE +PTS.

Pel cas Inclou E, ens caldria afegir SPS i substituir END per PTE, és a dir –END +SPS + PTE.

Pel cas Inclou S i E ens caldria substituir START i END per PTS i PTE respectivament, és a dir: -START – END + PTS + PTE.

A la **Taula 7. Correccions geometries algoritme shortest_path**, a la qual hem substituït START i END per S i E a efectes d'espai, és resumeix el resultat considerar tots els trajectes possibles.

Taula 7. Correccions geometries algoritme shortest_path

	Correccions geometries Shortest_path			
Trajecte	No Inclou	Inclou S	Inclou E	Inclou S i E
SS→ SE	SPS + SPE	SPE – S + PTS	-E +PTE + SPS	-E + PTE - S +PTS
SS→ TE	SPS + PTE	PTE – S + PTS	SPS – E +SPE	-S +PTS – E + SPE
TS→ SE	PTS + SPE	SPE – S + SPS	PTS – E + PTE	-S + SPS – E + PTE
TS→ TE	PTS + PTE	PTE – S + SPS	PTS – E + SPE	-S + SPS – E + SPE

Novament, aquesta taula representa un algoritme que ens permetrà determinar les correccions geomètriques necessàries al recorregut calculat pel shortest_path en funció de la bidireccionalitat de les arestes extremes. Aquest algoritme ha estat implementat a la funció dijkstra_anele, tal com veurem posteriorment.

Com a conclusió d'aquest estudi teòric, podem afirmar que en funció de les bidireccionalitats de l'aresta inicial i la final, estem en condicions de determinar els recorreguts possibles d'acord amb la **Taula 4. Trajectes possibles en funció de la bidireccionalitat de les arestes inicial i final.**

Per una altra banda i donat un recorregut possible, podem determinar les modificacions de les geometries calculades pel shortest_path d'acord amb la **Taula 7. Correccions geometries algoritme shortest_path**.

6 Descripció de la funcionalitat implementada

En un principi es va valorar implementar la funcionalitat de càlcul d'itineraris per medi de consultes sql a partir d'un script php, no obstant i donat l'elevat nombre de consultes necessàries és va optar per implementar la funcionalitat de càlcul d'itineraris directament a sobre el SGBD per medi de funcions, de manera que l'script php només hauria de fer una única consulta per a obtenir el resultat cercat. Es va optar per aquesta alternativa perquè reduïa el nombre de connexions i consultes externes a la BD amb la qual cosa, en principi, el client obtindria els resultats d'una manera més àgil i ràpida sense sobrecarregar l'accés a la BD.

Per a això ha calgut definir un tipus personalitzat, a sobre la base de dades, denominat edge que representa el conjunt d'una aresta i la projecció d'un punt a sobre la mateixa, i que conté la següent informació:

- gid: identificador de l'aresta de tipus integer
- the_geom geometria de l'aresta de tipus geometry
- Length: llargària de l'aresta de tipus float
- Source: identificador únic del vèrtex origen de tipus integer
- Target: identificador únic del vèrtex destinació de tipus integer
- dobleSentit: indicador de doble sentit de tipus boolean
- distanciaInici: distància a sobre l'aresta des del punt inicial de l'aresta a un punt donat a sobre l'aresta de tipus float
- distanciaFinal: distància a sobre l'aresta des del punt donat a sobre l'aresta al punt inicial de l'aresta de tipus float
- segmentInicial: geometria d'un hipotètic segment definit entre el punt inicial de l'aresta i el punt donat de tipus geometry
- segmentFinal: geometria d'un hipotètic segment definit entre el punt donat i el punt final de l'aresta de tipus geometry

Així mateix ens ha calgut definir quatre funcions de les quals a continuació presentem la seva signatura:

```
FUNCTION find_nearest_edge_information(IN taula text,
                                       IN posicio1 double precision,
                                       IN posicio2 double precision,
                                       OUT nearestedge edge
                                       ) RETURNS edge AS

FUNCTION barcelona_routing(IN taula text,
                           IN startpointx double precision,
                           IN startpointy double precision,
                           IN endpointx double precision,
                           IN endpointy double precision,
                           IN vehicle boolean,
```

```
                                    OUT id integer,
                                    OUT gid integer,
                                    OUT the_geom geometry,
                                    OUT lengthrecorregut double precision
                                    ) RETURNS SETOF record AS

FUNCTION recorregut(IN taula text,
                                    IN startedge edge,
                                    IN endedge edge,
                                    IN startend text,
                                    IN vehicle boolean,
                                    OUT arrayofgeoms geoms[],
                                    OUT lengthrecorregut double precision,
                                    OUT elements integer
                                    ) RETURNS record AS

FUNCTION dijikstra_anele(taula text,
                                    startedge edge,
                                    endedge edge,
                                    startpoint text,
                                    endpoint text,
                                    vehicle boolean
                                    ) RETURNS SETOF geoms AS
```

Pel que fa a la funció find_nearest_edge_information, aquesta funció accepta com a paràmetres d'entrada el nom de la taula a on es troben les geometries de les arestes de la xarxa dirigida i les coordinades d'un punt i retorna un edge calculat tenint en compte l'aresta més propera al punt en qüestió.

La funció barcelona_routing implementa l'algoritme de recorreguts possibles en funció de la bidireccionalitat (**Taula 4. Trajectes possibles en funció de la bidireccionalitat de les arestes inicial i final**).

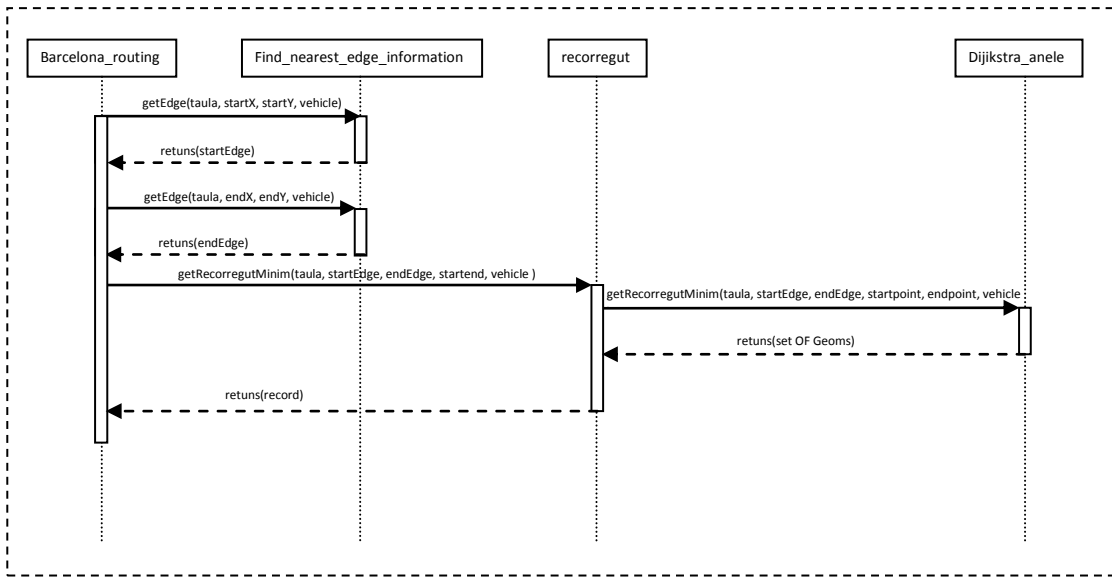
La funció dijkstra_anele implementa l'algoritme que permet determinar la correcció de la geometria en funció dels extrems considerats i de la inclusió o no d'aquests al recorregut del shortest_path(**Taula 7. Correccions geometries algoritme shortest_path**).

En síntesi barcelona_routing recupera els edges més propers corresponents al les coordinades de dos punts, inicial i final, per medi de crides a la funció find_nearest_edge per a continuació i fent servir l'algoritme de recorreguts possibles en funció de la bidireccionalitat fer consultes repetides a la funció recorregut que es limita a passar-les a Dijikstra_anele la qual fent servir l'algoritme de correcció de geometries i la funció de pgRouting shortest_path, retorna el recorregut mínim.

Recorregut una vegada que obté la geometria pertinent, calcula la seva llargària i retorna un registre que conté la geometria i la seva llargària.

Per la seva banda barcelona_routing comprova quin dels registres obtinguts té una llargària inferior i retorna la seva geometria. Tot això queda resumit al diagrama d'interacció de la **Figura 18. Diagrama de interacció de les rutines.**

Figura 18. Diagrama de interacció de les rutines



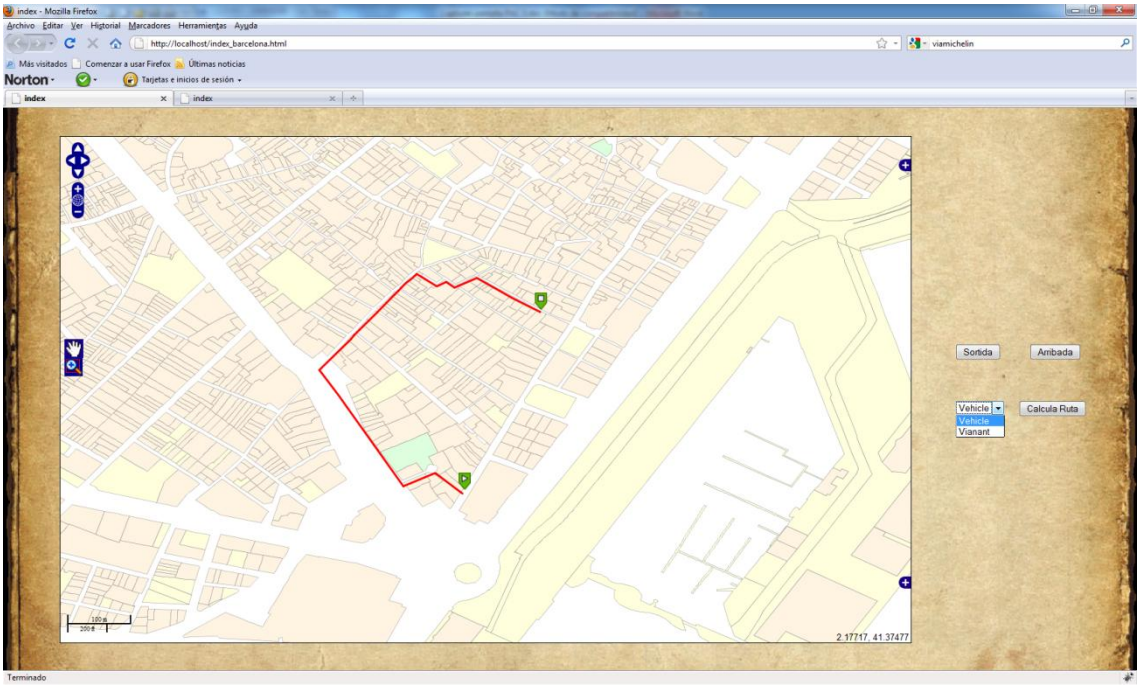
Pel que fa a la interfície d'usuari, l'hem dividit en dues parts principals, la primera en forma de document html denominat init.html amb algun codi php incrustat i una crida a un arxiu php denominat atraccio.php (que serveix per a recuperar la posició d'una atracció donada) i un arxiu denominat barcelona_routing.php que és cridat per a fer el càlcul de les rutes, basant-se aquest en la crida a la funció barcelona_routing de la base de dades.

7 Interfície d'usuari

Pel que fa a la interfície d'usuari, hem optat per una interfície minimalista composada de tres botons i una llista de selecció a banda de la interfície clàssica d'OpenLayers, aquesta darrera en l'esperança de que l'usuari l'hagi pogut trobar amb anterioritat a algun lloc.

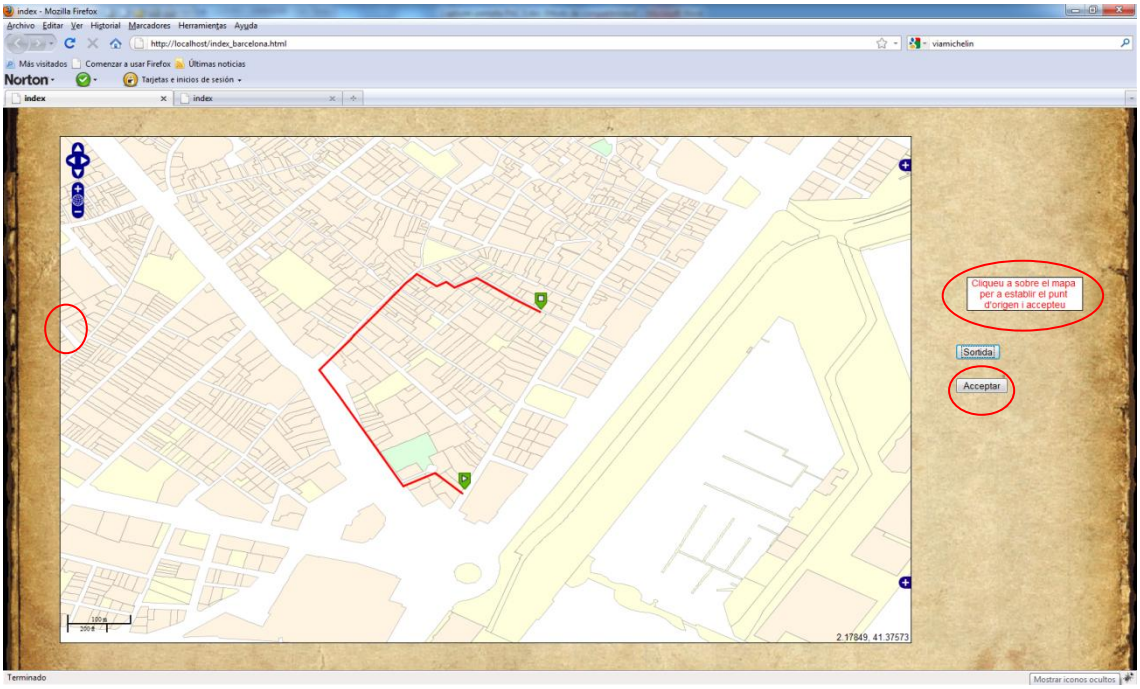
Els botons Sortida i Arribada, tal qual els seus noms suggereixen, serveixen per a establir els punts origen i destinació del recorregut. La llista de selecció permet triar entre un recorregut per a vianants o per a vehicles i finalment el boto Calcula Ruta realitza el càlcul de la ruta en base als punts fixats i el tipus d'itinerari seleccionat. A la **Figura 19 Pantalla inicial de la interfície d'usuari** tenim una captura de pantalla a la qual es pot observar la interfície.

Figura 19 Pantalla inicial de la interfície d'usuari



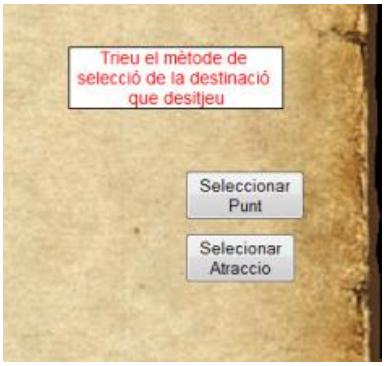
Pel que fa a la tria del punt origen ens apareix un missatge explicatiu per a que cliquem a sobre el mapa per a establir el punt de sortida, així com un botó acceptar per a acceptar el punt establert i retornar a la pantalla inicial. Cal comentar que la NavToolBar es troba deshabilitada mentre fem aquesta operació. A la captura de pantalla de la **Figura 20 Pantalla introducció punt origen** podem observar el mencionat.

Figura 20 Pantalla introducció punt origen



Si seleccionem el botó Arribada per a establir el punt destinació de l'itinerari, ens apareixen dos botons (Selecciona Punt i Selecciona Atracció) i un missatge explicatiu amb l'objectiu de permetre seleccionar un punt de destinació de manera lliure o be de seleccionar una atracció d'una llista de manera que el sistema se n'encarregui de mostrar el punt a sobre el mapa, a la **Figura 21 Pantalla tria mètode selecció punt destinació (Punt o Atracció)** podem observar un zoom de la dita pantalla:

Figura 21 Pantalla tria mètode selecció punt destinació (Punt o Atracció)

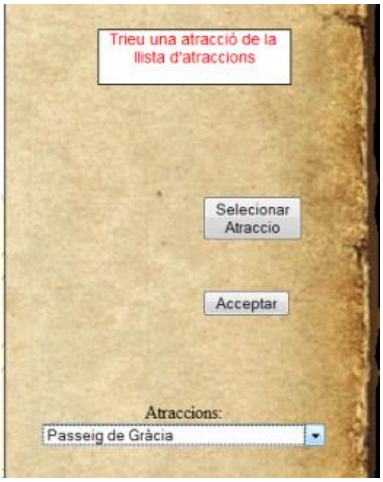


Si triem Seleccionar Punt ens apareix un menú semblant al de selecció del punt d'origen, que podem observar a la captura de pantalla de la **Figura 22 Pantalla selecció punt final a sobre el mapa**

Figura 22 Pantalla selecció punt final a sobre el mapa

Per contra, si triem Selecciona Atracció, ens apareix un missatge explicatiu, un botó Accepta i una llista de selecció a la qual podem seleccionar l'atracció i acceptar, retornant al la pantalla inicial, ho podem observar a la captura de pantalla de la **Figura 23 Pantalla selecció punt final a atracció.**

Figura 23 Pantalla selecció punt final a atracció

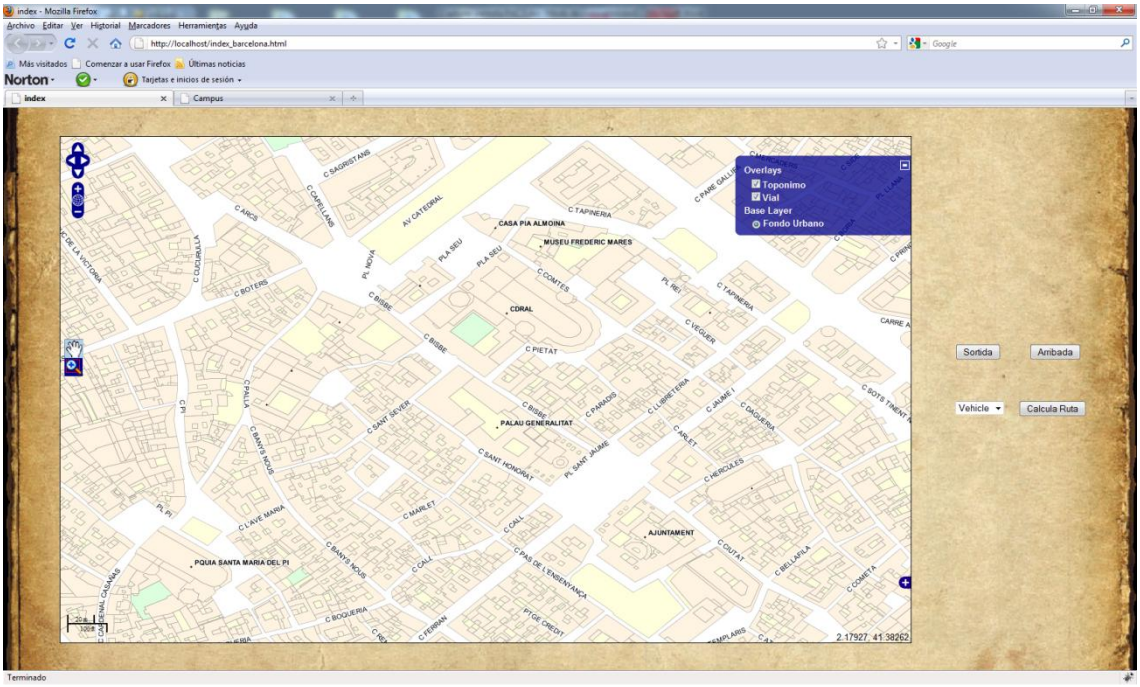


Per una altra banda tenim tres capes, una capa base corresponent al Fondo Urbano de CartoCiudad, i dues més que per defecte es troben deshabilitades amb l'objectiu d'agilitzar la càrrega del mapa. Es tracta de la capa Toponimo de CartoCiudad que hem inclòs per a poder determinar possibles punts d'interès no contemplats a la nostra BD i la capa Vial que, com el seu propi nom indica, es correspon amb la capa Vial de CartoCiudad i inclou els noms dels carrers. A continuació a la **Figura 24 Pantalla amb les capes Fondo Urbano, Vial i Topónimo activades** en tenim una captura de pantalla de les dites capes.

De manera que podem dir que s'han assolit tots els objectius proposats:

- Itinerari per a vianants no direccional.
- Itinerari per a vehicles direccional.
- Itineraris acurats.

Figura 24 Pantalla amb les capes Fondo Urbano, Vial i Topónimo activades



8 Resultats

Amb l'objectiu de comprovar si s'han assolit els objectius proposats, passem a fer a continuació una comparació dels resultats de càlcul d'itineraris mínims que s'obtenen amb el sistema estudiat i amb el sistema que es proposa a aquest treball.

Per una banda, podem considerar els mateixos punts que hem considerat a la **Figura 10 Mètode Shortest Path Dijkstra programari estudiat, els itineraris comencen i acaben després dels punts origen i destinació** per al cas del mètode Shortest Path Dijkstra que com sabem és no direccional i per tant el compararem amb els resultats que hem obtingut amb el nostre sistema per al cas de l'itinerari per a vianants. A la **Figura 25 Itinerari per a vianants amb el programari desenvolupat** podem comprovar que al contrari del que succeïa al cas anterior l'itinerari comença de manera precisa al punt origen i acaba al punt destinació.

Per una altra banda, si considerem els mateixos punts i un itinerari per a vehicles, podem comprovar a la **Figura 26 Itinerari per a vehicles amb el programari desenvolupat** que el recorregut es veu modificat degut als sentits de circulació dels carrers, tot i que com podem comprovar també l'itinerari comença al punt origen de manera precisa i acaba de la mateixa manera al punt destinació.

Figura 25 Itinerari per a vianants amb el programari desenvolupat

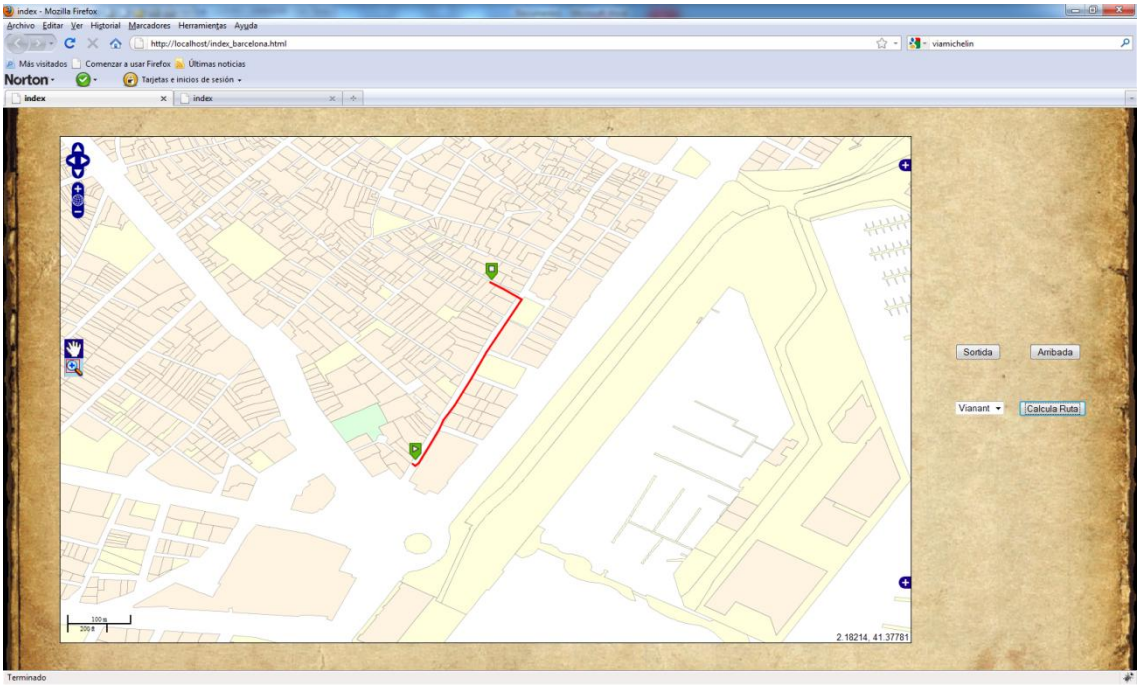
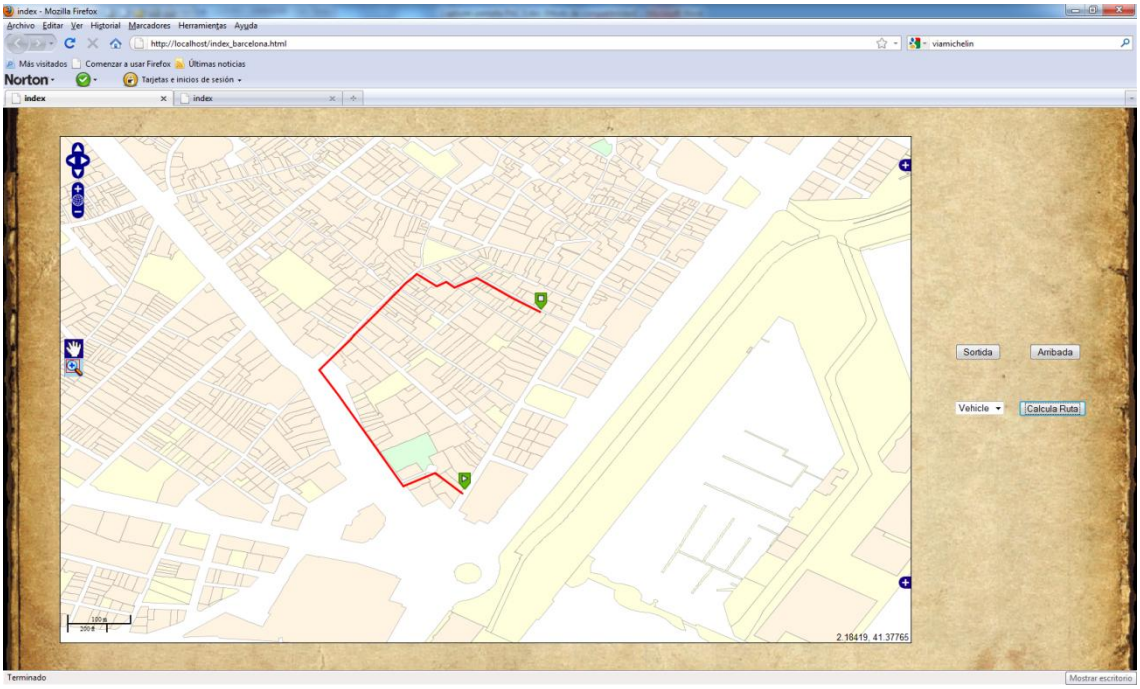


Figura 26 Itinerari per a vehicles amb el programari desenvolupat



9 Conclusions

S'ha millorat el sistema de routing analitzat afegint les funcionalitats i millores proposades.

S'ha comprovat que, amb certes limitacions pel que fa a la càrrega de les dades d'enrutament, resulta possible obtenir un sistema d'enrutament plenament funcional a sobre Windows 7 amb programari lliure.

S'han instal·lat versions de programari de les quals no es tenia constància de la seva compatibilitat amb Windows 7, resultant en instal·lacions positives i funcionals.

Ha quedat demostrat que, almenys pel que fa a la ciutat de Barcelona, les dades OpenStreetMap no son una font òptima per a un sistema d'enrutament real, donat que en tractar-se de dades obtingudes d'una manera col·laborativa tenen defectes de qualitat importants. Tot i que cal plantejar-se una correcció manual de les mateixes, caldria verificar que els sentits de circulació son reals, que les vies tenen la classificació adequada (es podria donar el cas de que alguna via estigues mal classificada)...

Pel que fa a la funcionalitat d'enrutament, caldria comprovar el seu funcionament en unes condicions de càrrega i de treball reals amb l'objectiu de comprovar si podria resultar interessant l'addició d'algun índex i la velocitat real de l'aplicació.

Hem implementat un sistema d'enrutament que minimitza, dintre del possible, el trànsit de consultes externes al SGBD.

Hem implementat un conjunt de funcions (find_nearest_edge_information, recorregut i barcelona_routing) i un nou tipus (edge) que donades les coordinades de dos punts i el tipus de recorregut considerat (vehicle o vianant) permeten calcular el recorregut mínim acurat entre els dos punts, retornant un conjunt de geometries que descriuen aquest itinerari.

Cal comentar que sembla que hi ha algun tipus d'incompatibilitat entre *OpenLayers* 2.8 i *I. Explorer* (7 i 8) que fa que, mentre el programari desenvolupat funcioni perfectament amb *Chrome* (*Google*) i *Firefoxs*, es produeixin errades d'*OpenLayers* amb l'explorador de *Microsoft*. Malauradament, amb el temps disponible no hem pogut arribar a trobar-hi una sol·lució.

Ja per a acabar, comentar que s'han assolit en excés tots els objectius definits al pla de treball.

10 Millores i línies de futur

Pel que fa a possibles millores del sistema d'enrutament, tenim per una part el que fa a la millora de la interfície d'usuari, donat que les principals atraccions les hem incloses a una base de dades, seria possible afegir-hi informació addicional, tal com descripcions literals de les atraccions, amb informació dels seus horaris i calendaris d'obertura, i d'imatges de les mateixes que es podrien mostrar en seleccionar a la llista corresponent per a permetre a l'usuari valorar si resulta interessant accedir-hi o no.

Per una altra banda caldria considerar la possibilitat d'implementar un sistema de consultes als nomenclator de Vial i nomenclator de Portal del sevidor WFS de Cartociudad per a facilitar a l'usuari l'establiment del punt d'origen del recorregut a partir del nom del carrer i el portal.

Si bé aquesta possibilitat resulta factible, també s'ha estat valorant una altra possibilitat que a partir de dos noms de carrer que es troben a una cruïlla, permetria determinar el punt de la mateixa i per tant el punt que podria ser d'origen del recorregut, a aquests efectes s'aprofitaria la funció ST_ClosestPoint de PostGIS que a partir de dues geometries en format WKF retorna la geometria corresponent al punt més proper:

```
SELECT ST_AsText(ST_ClosestPoint(AA.A ,BB.b)),
                ST_Distance(AA.a, BB.b) AS distance
FROM (select the_geom AS a
                FROM  ways
                WHERE name='nomDeCarrer1') as AA,
        (select the_geom as b
                FROM  ways
                WHERE name='nomDeCarrer2') AS BB
ORDER BY distance LIMIT 1;
```

Aquesta consulta juntament amb una altra que obtindria els noms dels carrers de la base de dades que nodririen dues llistes desplegables a la interfície d'usuari per a que aquest tries dos carrers que servirien de llavor per a la consulta anterior.

Seria interessant esbrinar quin es el problema amb l'*Internet Explorer* que hem mencionat a l'apartat anterior i cercar-hi solucions.

Ja per a acabar queden dos punts relacionats amb les dades OSM que son susceptibles de millora. Per una banda el desenvolupament d'un sistema de càrrega directa de dades per a Windows a la manera de osm2pgrouting i per l'altre la recollida de dades específiques per a vianants que incloguin semàfors i passos per a vianants, així com escales (amb un cost diferenciat), passatges, etc.

11 Bibliografia

1. **MapServer**. http://mapserver.org/.

2. **OpenLayers** http://openlayers.org/.

3. **PostgreSQL** http://www.postgresql.org/.

4. **PostGIS**. http://postgis.refractions.net/.

5. **OpenStreetMap**. http://www.openstreetmap.org/.

6. **pgRouting Project**. http://pgrouting.postlbs.org/.

7. **Gonzàlez A. , Rubio J.M., Velasco A.** WWW.Cartociudad.es v 2.0, 05 de 02 de 2010. http://www.cartociudad.es/portal/pdf/CARTOCIUDAD_ServiciosWeb.pdf.

8. **Windows 7**. http://windows.microsoft.com/es-ES/windows7/products/home?os=win7.

9. **Philipona, C. I Kastl, D.** "Web-based Routing: An Introduction to pgRouting with OpenLayers", FOSS4G 2007 – Victoria, Illa de Vancouver, Canadà

10. **Bataller Díaz, A.** "Gestió i desenvolupament de projectes. Conceptes i suggeriments", Universitat Oberta de Catalunya

11. **Bataller Díaz, A.** "Gestió i desenvolupament de projectes. Conceptes i suggeriments", Universitat Oberta de Catalunya

12. **Nita Sàenz, H. i Rut Vidal, O.** "Redacció de textos cientificotècnics", Universitat Oberta de Catalunya

13. **Beneito Mountagut, R.** "Presentació de documents i elaboració de presentacions" , Universitat Oberta de Catalunya

14. **Rodríguez Lloret, J. i Olivella, R.** "Introducció als sistemes d'informació geogràfica. Conceptes i operacions fonamentals" , Universitat Oberta de Catalunya

15. **Botella Plana, A.** "Bases de dades geogràfiques. Magatzems de dades geogràfiques" , Universitat Oberta de Catalunya

16. **Botella Plana, A.** "Noves tendències en SIG. Present i futur dels sistemes d'informació geogràfica", Universitat Oberta de Catalunya