



DESENVOLUPAMENT D'UN SERVIDOR D'INFORMACIÓ GEOGRÀFICA

Memòria del projecte de final de carrera corresponent als estudis d'Enginyeria Superior en Informàtica presentat per Domingo Manubens Gil i dirigit per Juan José Ramos González.

Bellaterra, Juny de 2009

El firmant, Juan José Ramos González, professor del Departament de Telecomunicacions i d'Enginyeria de Sistemes de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Domingo Manubens Gil

Bellaterra, Juny de 2009

Firmat: Juan José Ramos González

*El que fa bell el desert, és que té un pou amagat en
algun lloc, digué el petit príncep...
Antoine De Saint-Exupéry*

Agraïments

Agraïr a tothom qui m'ha fet costat directa o indirectament en l'el·laboració d'aquest projecte. Primer de tot al Juanjo, el meu tutor, per les seves indicacions, l'explicació exhaustiva de les seves idees i per la seva empenta per portar-les endavant. Al Jaume Figueras pel seu suport des de la UPC via mail i en persona.

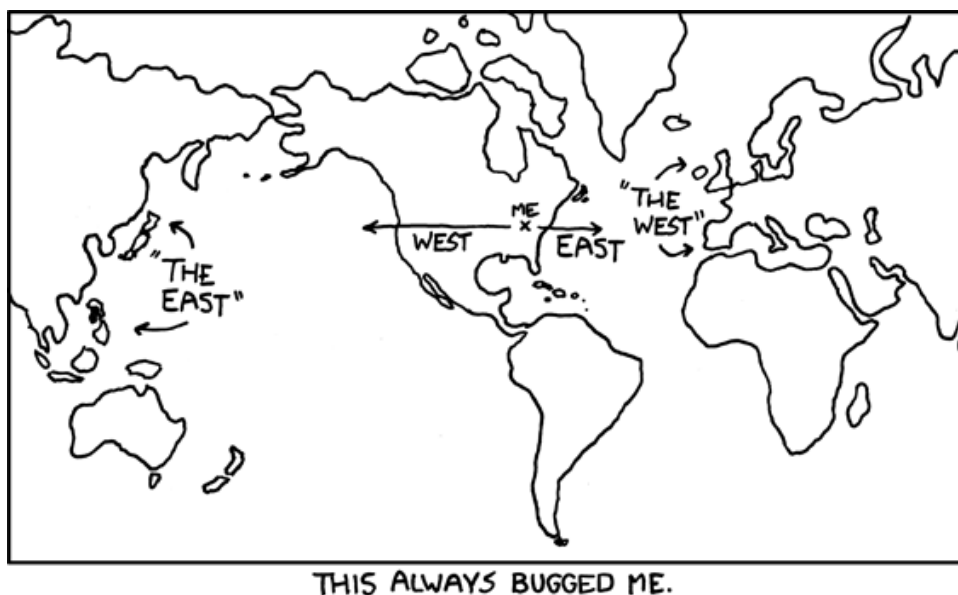
Especial menció a aquells que amb el seu treball desinteressat contribueixen a les comunitats de software lliure. Gràcies Ian Dees i Rammi per les vostres ràpides i atentes respostes.

Gràcies als companys de carrera, amb qui tantes experiències he compartit i que ara a les acaballes de la meua etapa a la ETSE no han deixat de donar-me ànims. Per haver compartit uns anys esplèndids amb vosaltres i haver après tant i tant: Uri, Ricard, Uri P., Àlex, Jordi, Xavi, Pedro i molts altres.

Gràcies Anton i Marta, que heu confiat en mi durant aquest any que a part de fer el projecte, he treballat amb vosaltres al SAF.

A la meua família, gràcies als meus pares i als meus germans. Gràcies Guillem i Pere per l'ajuda amb AutoCAD. Una salutació als amics de Santpedor, Montse, Cris, Uri, Dani, Aina, Míriam, Anna, Laura, Gisel·la i un llarg etc.

Finalment als correctors i correctores d'aquesta memòria, us agraiixo les esmenes! Acabo amb un petit acudit.



Índex

I	INTRODUCCIÓ	1
1	Motivacions	1
1.1	Aplicacions a on s'integra	3
2	Objectius del projecte	3
2.1	Principals activitats del pla de treball	5
3	Estat de l'art	5
4	Estudi de viabilitat del projecte	6
5	Estructura de la memòria	7
II	CONTEXTUALITZACIÓ	9
6	Sistemes d'informació geogràfica	9
6.1	Què és un SIG	9
6.2	Components d'un SIG	9
7	Bases de dades Cartogràfiques	10
7.1	PostGIS i l'OGC	11
8	Model tessel·les	12
9	DXF	13
10	OSM	15
10.1	Components d'OSM	17
10.2	Primitives de Dades	17
10.2.1	Way	17
10.2.2	Node	18
10.2.3	Relation	18
10.3	Objectiu a assolir	18
III	SERVIDOR	19
11	PostGIS i backend	19
11.1	Instal·lació i configuració	20
11.2	Conversor osm2pgsql	21
11.3	Backend	21

12 Mapnik	22
12.1 Instal·lació	22
12.2 Pre-renderitzat	24
12.3 Configuració	25
13 Mod Tile	25
13.1 Instal·lació	26
13.2 Configuració	26
13.3 Execució	26
IV CLIENT	28
14 Anàlisi i procés de DXF	28
14.1 GRASS	29
14.1.1 Location	29
14.1.2 v.in.dxf	29
14.1.3 Georectificació	30
14.1.4 v.out.ogr	30
14.2 Conversor shp-to-osm	30
14.2.1 Execució	32
14.3 Importació al Servidor	32
15 Open Layers	33
15.1 Classes principals	33
15.2 Instal·lació	33
15.3 Codificació d'una pàgina client	34
16 Component User Parameters	35
16.1 Joomla!	35
16.2 User Parameters	37
16.2.1 Model	37
16.2.2 View	37
16.2.3 Controller	38
16.3 toopathTracking	39
V RESULTATS	40
VI CONCLUSIONS I VIES DE CONTINUACIÓ	46
17 Conclusions	46
18 Vies de continuació	47
18.1 Portal de Mobilitat de la UAB	47
18.2 Altres vies de continuació	47

Bibliografia	48
---------------------	-----------

VII ANNEXES	50
--------------------	-----------

Índex de figures

1	Portal web	2
2	Graf	2
3	Seguiment en línia	3
4	Simulador de tràfic	4
5	Portal de mobilitat	4
6	Planificació temporal	8
7	Components d'un SIG	9
8	Xarxa de vèrtexs geodèsics a Espanya, que conformen el datum	11
9	Torre de Hemlert	11
10	Model de tesselles	13
11	Diagrama de components d'OSM	16
12	Primitives de dades d'OSM	17
13	Front-end de PostgreSQL. Taules "giscat"	22
14	Tile 0.png	23
15	Tile 12224.png	23
16	Petició http://etse-73-192.uab.cat/uab/0/0/0.png	27
17	Resposta de mod_tile a la petició de 0/0/0.png	27
18	v.in.dxf	31
19	v.out.ogr	31
20	Georectificació	31
21	Punts de control i error	31
22	Component User Parameters	38
23	LAYERS a Component Tracking	39
24	Plànol DXF - UAB	41
25	Plànol DXF capes de vials aïllades - UAB	41
26	SHP georectificat - UAB	42
27	OSM generat - UAB	42
28	Base de dades - UAB	43
29	Portal web - UAB	43
30	Portal web zoom 19 - UAB	44
31	Origen, Destí - UAB	45

32	Ruta - UAB	45
----	----------------------	----

Part I

INTRODUCCIÓ

1 Motivacions

És molt comú en entitats i empreses disposar d'arxius en format AutoCAD amb plànols detallats d'urbanisme, o de les seves instal·lacions. L'interès habitual sempre ha estat mostrar i editar aquests arxius en l'entorn CAD i poder imprimir plànols en paper. Actualment però, hi ha hagut una explosió d'aplicacions destinades a mostrar i editar cartografia en entorns web com Google Maps, Microsoft Live Maps o OpenStreetMaps entre d'altres. Aquests llocs web donen al públic accés a gran quantitat de dades geogràfiques. Però què passa si volem mostrar les nostres pròpies dades? La solució proposada en aquest projecte és la de desenvolupar un servidor d'informació geogràfica i incorporar els arxius de què disposem en el model de dades del sistema.

Un sistema d'informació geogràfica (SIG), per altra banda, té la finalitat de resoldre problemes complexos de planificació i gestió a través de la manipulació, anàlisi i desplegament de la informació capturada i emmagatzemada. Actualment els SIG estan tenint una forta implantació en els anomenats Serveis Basats en la Localització (LBS) degut a l'abaratiment i massificació de la tecnologia GPS integrada en dispositius mòbils de consum (telèfons mòbils, PDAs, Ordinadors portàtils). Els LBS permeten als dispositius mòbils amb GPS mostrar la seva ubicació respecte a punts d'interès fixes (restaurants, gasolineres, caixers, . . . més propers), mòbils (amics, treballadors, autobusos, flota de vehicles) o per a transmetre la seva posició a una ubicació central per a la seva visualització o un altre tipus de tractament.

El servidor d'informació geogràfica tindrà doncs dues funcions principals:

- Visualització dels mapes en el portal web, amb una estructura de capes personalitzada per l'usuari.
- Disposar d'una backend en forma de graf connex i dirigit, destinat a la resolució de problemes de planificació i gestió.

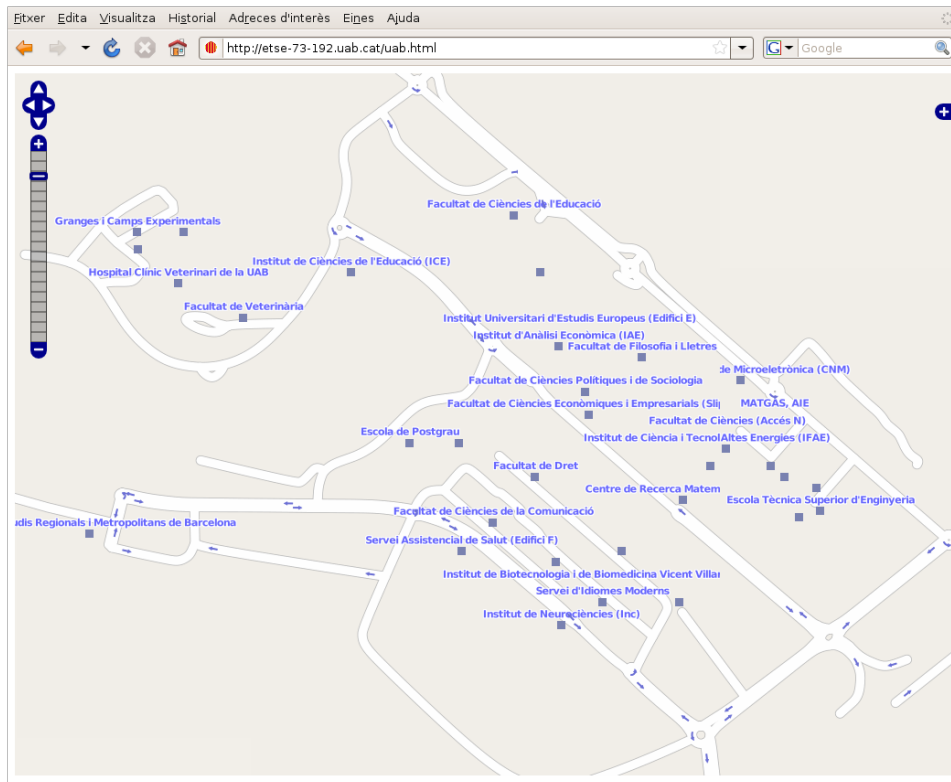


Figura 1: Portal web

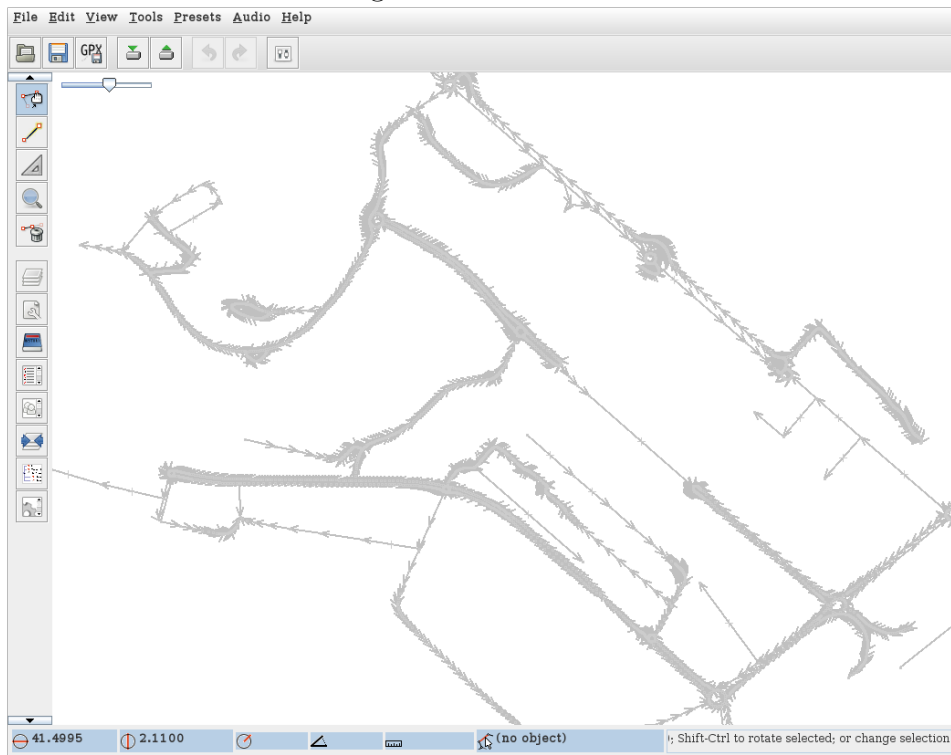


Figura 2: Graf

1.1 Aplicacions a on s'integra

Un cop en funcionament el servidor d'informació geogràfica, gràcies al portal i tot el software que el dota de funcionalitat, sobre cartografia pròpia, es podrà fer: **Seguiment de vehicles en línia:** en un portal lliure que capta la senyal GPS en temps real dels dispositius instal·lats en vehicles en moviment (Figura 3). **Simulador de tràfic:** prendre decisions en temps de simulació, per a una millor planificació i gestió d'una flota de vehicles en un determinat entorn (Figura 4). **Portal de mobilitat:** trobar rutes òptimes entre dos o més punts determinats de la nostra zona geogràfica (Figura 5).

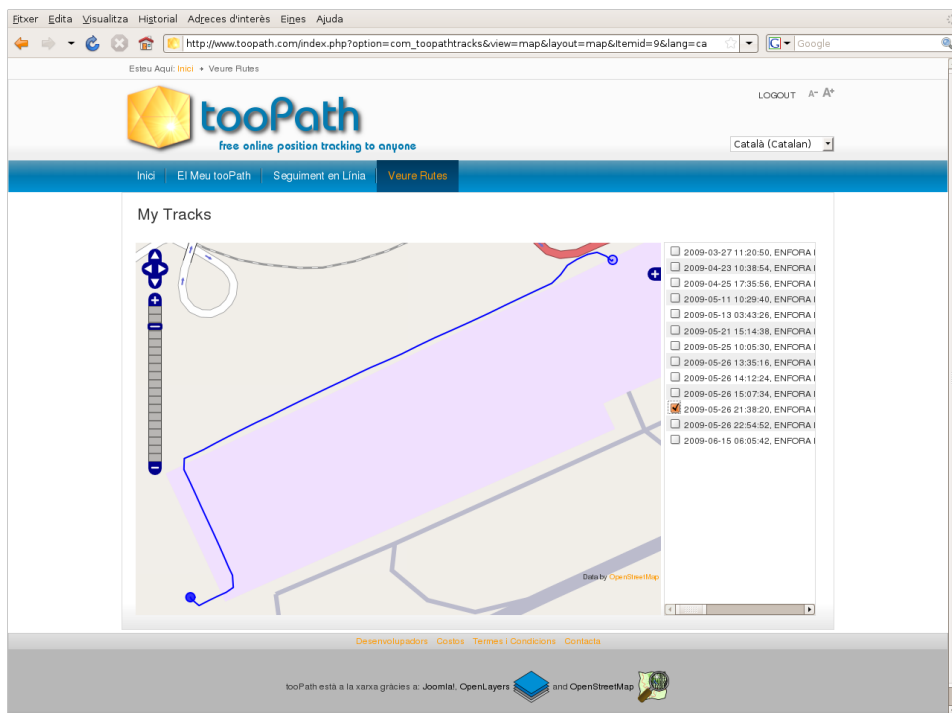


Figura 3: Seguiment en línia

2 Objectius del projecte

Establim com a principal propòsit:

El desenvolupar tècniques per automatitzar el pas de diversos formats a informació cartogràfica que respongui a dues funcionalitats:

- Servidor de mapes accessible mitjançant web (cas Toopath, Figura 3).
- Sistema d'informació geogràfica (cas UAB, Figura 5).

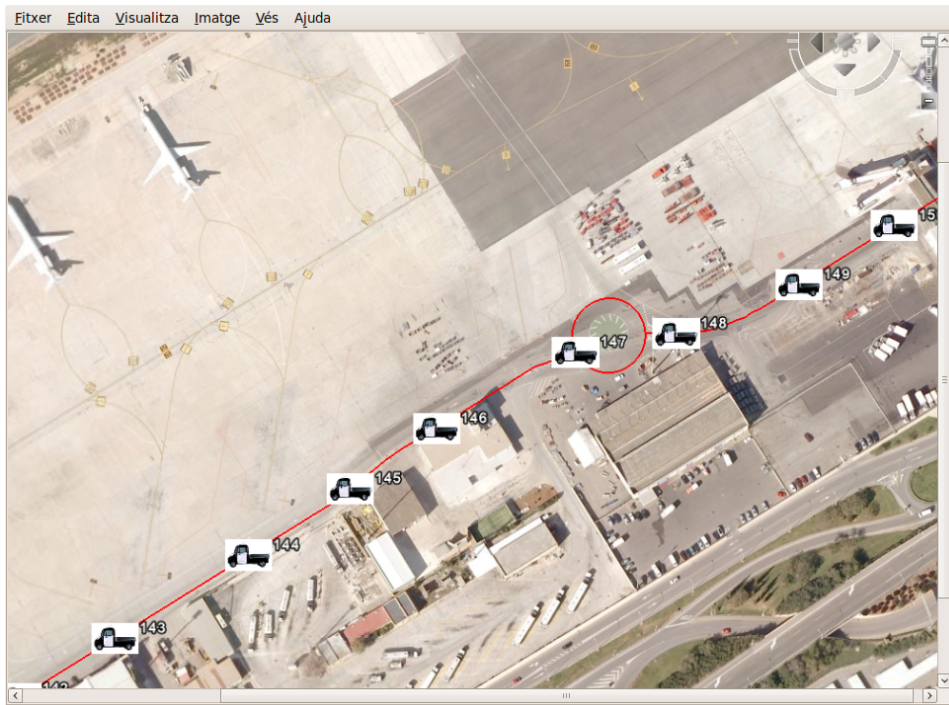


Figura 4: Simulador de tràfic

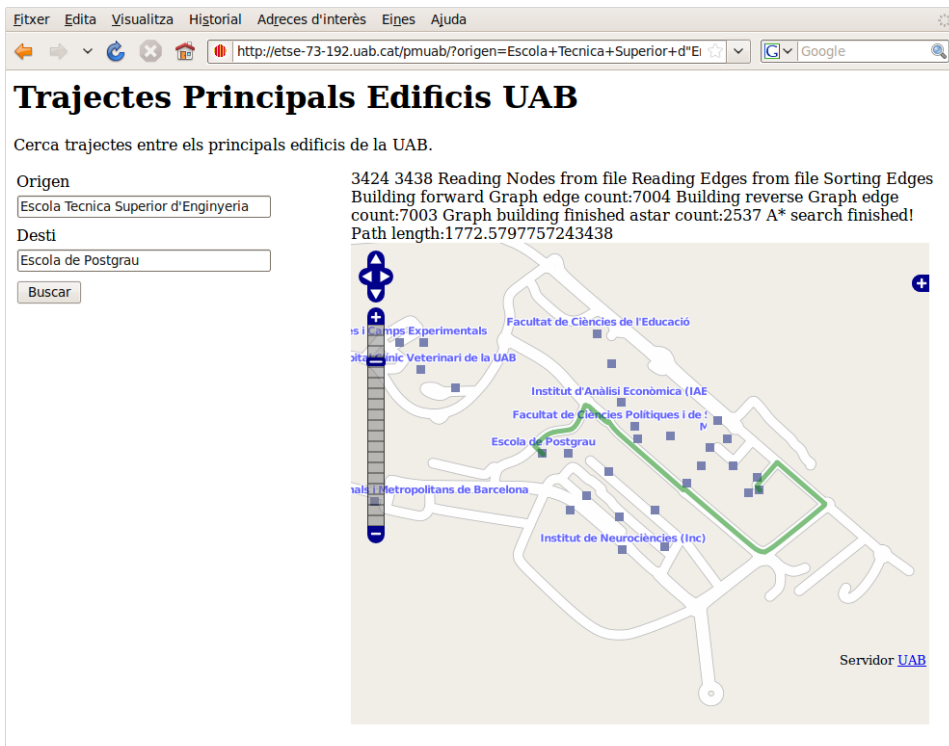


Figura 5: Portal de mobilitat

Objectius:

- Identificar i avaluar les especificacions d'un determinat format cartogràfic.
- Avaluar l'adequació de les dades de que es disposen en el format font, per incorporar i servir aquesta informació geogràfica en un servidor web.
- Desenvolupar un sistema de servidor d'informació geogràfica per visualitzar mapes dins de l'aplicació de seguiment de dispositius GPS, entre d'altres aplicacions.
- Dissenyar i desenvolupar un *parser* per fer el pas d'un format cartogràfic al model de dades del servidor d'informació geogràfica.

2.1 Principals activitats del pla de treball

Inicialment s'estudiarà el format DXF, *Data eXchange Format* d'AutoCAD. S'haurà de desenvolupar una aplicació d'anàlisi i procés (o un *parser*) o bé aprofitar eines de conversió existents, per transformar qualsevol mapa o plànol generat amb AutoCAD i exportat al format DXF, per tal d'adaptar-lo a l'esquema de la plataforma OpenStreetMap (*OSM protocol v0.6*). Es faran proves amb un plànol d'una part de Rubí i posteriorment es treballarà sobre un plànol d'una empresa interessada en l'aplicació de seguiment, que disposa de la font en format DXF.

S'importaran les dades generades a la base de dades PostGIS i es configurarà el servidor perquè serveixi les cobertures en forma d'imatge, generades per Mapnik.

Finalment caldrà visualitzar en l'aplicació web de seguiment, com una nova capa, les dades generades a partir del dibuix CAD (les rutes en la web de seguiment vindran determinades bàsicament pels carrers del plànol).

3 Estat de l'art

Google Maps permet afegir mapes fàcilment en un lloc web, però a l'utilitzar l'API de Google la possibilitat d'utilitzar dades pròpies és limitada. Si volem mostrar les nostres pròpies dades, o bé dades d'un origen diferent a Google, *OpenLayers*, una llibreria JavaScript lliure, ens dona més opcions. Permet col·locar el contingut de forma dinàmica en la web i mostrar cobertures de mapa i marcadors carregats de qualsevol font. No haurem de dependre de les dades que Google ens proporcioni ni ens haurem de veure limitats per eventuais restriccions que puguin imposar.

Toopath, és el servei de seguiment gratuït que permet a qualsevol usuari conèixer la posició dels seus dispositius mòbils i compartir la seva localització sobre Google Maps. *Toopath* és un portal no dissenyat per a usos comercials, amb finalitats lúdiques i d'experimentació. El present projecte vol dotar de flexibilitat *Toopath*, ampliant les possibilitats que té fins ara amb Google Maps. Per exemple, fer el *tracking* dels vehicles que circulen per una planta industrial, un aeroport o una zona geogràfica acotada sobre una cartografia pròpia i detallada.

El projecte *OpenStreetMap* està pensat per crear i proporcionar dades geogràfiques de manera lliure, com per exemple mapes de carrers i carreteres, a tothom qui ho desitgi. Aquest projecte va començar perquè la majoria de mapes que pensem que són lliures de fet, no ho són: tenen restriccions tècniques i/o legals que impedeixen als usuaris el seu ús de forma creativa, productiva o sorprenent en noves aplicacions.

A mesura que el projecte *OpenStreetMap* (OSM) ha anat madurant i la seva base de dades ha millorat ràpidament en qualitat i cobertura, ha anat sorgint al seu voltant tot un ecosistema d'eines informàtiques i serveis, convertint-se en una font de dades utilitzable per a projectes complexos. Així per exemple, ha sorgit abundant *software* que, o bé facilita la captura, edició, tractament i presentació de la cartografia, o per altra banda fa ús d'aquestes dades de diverses formes.

El desenvolupament d'aquest projecte, ens permetrà disposar doncs, dels nostres mapes en una font de dades al nostre abast com és OSM. L'existència d'eines com *PostGIS* i *Mapnik* ens serviran per emmagatzemar les dades en una base de dades cartogràfica i renderitzar i presentar les imatges, respectivament. *Mapnik* és un *toolkit* lliure de desenvolupament d'aplicacions de mapes. És el generador d'imatges a partir de la base de dades cartogràfica *PostGIS*. Finalment s'implementarà la interfície del mapa amb *OpenLayers*, per mostrar les dades a la web.

4 Estudi de viabilitat del projecte

1. Justificació de la necessitat del projecte

- **Objectius i abast del projecte**

Analitzar i processar sistemes d'informació geogràfica definits amb diferents formats per tal d'incorporar i servir aquesta informació geogràfica en un servidor web destinat al seguiment de dispositius GPS.

- **Àmbit d'actuació**

Toopath, web pel seguiment de dispositius GPS a on s'integrarà el projecte. Es pretén donar més flexibilitat en l'ús de mapes.

- **Destinatariis o usuaris**

Qualsevol persona amb dispositius GPS dels quals vulgui fer seguiment.

2. Descripció

- **Serveis oferts i activitats desenvolupades**

El servei ofert serà el seguiment de dispositius GPS (*tracking*) sobre el mapa desitjat. Es requereix desenvolupar un servidor de mapes amb la funcionalitat afegida d'incorporar dades SIG de diversos formats.

Es gestionarà i funcionarà en les mateixes condicions que Toopath.

El treball s'organitzarà segons la planificació temporal detallada en la Figura 6, i serà necessari com a únic recurs material una màquina, servidor (*GNU/Linux*) amb *Apache*, instal·lat al departament de Telecomunicacions i Enginyeria de Sistemes (TES) de la UAB.

- **Estimació de la demanda potencial**

S'estima que la demanda potencial usuària és la pròpia del software, serveis i llocs web de Toopath.

- **Perfil dels usuaris**

Els usuaris de l'aplicació han de poder fer-ne ús de forma senzilla i es requeriran els mínims coneixements tecnològics.

- **Manteniment**

Al ser un projecte en evolució i basar-se en projectes lliures, caldrà parar especial atenció a l'actualització conforme surtin noves versions del software utilitzat.

3. Riscos tecnològics

- **Els projectes de software lliure tenen un procés de desenvolupament dinàmic que poden comportar entre d'altres:**

Baixa dedicació i capacitat dels desenvolupadors.

Fase de disseny sobre el projecte inadequada.

Disposar d'aplicacions encara no disponibles en la seva versió final.

Desenvolupaments encara poc familiars pel potencial usuari.

5 Estructura de la memòria

En el capítol II, es presenten els conceptes bàsics de les bases de dades cartogràfiques així com els SGBD que organitzen i estructuren la informació en un SIG, així com el model en que es

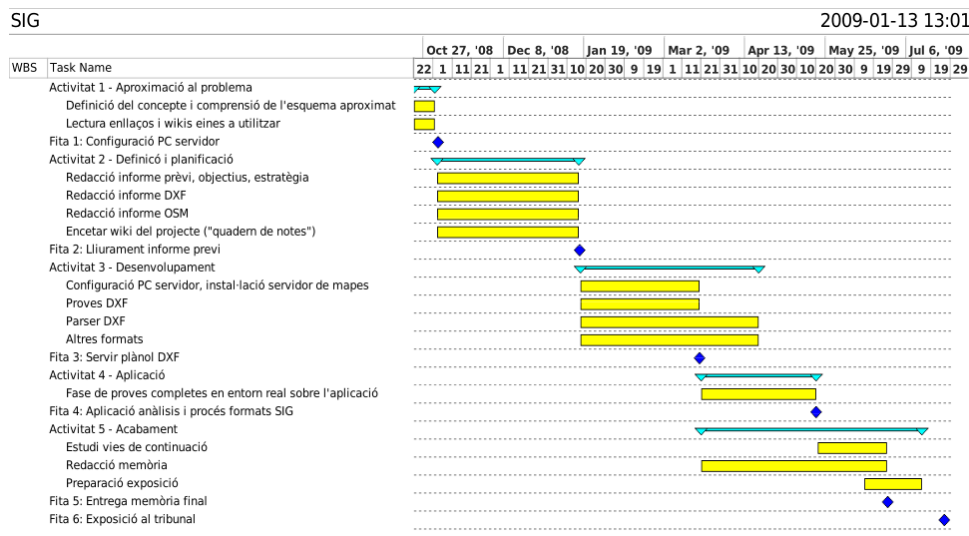


Figura 6: Planificació temporal

basa el servidor. Es detallen les especificacions dels formats de dades inicials i resultants, (DXF i OSM respectivament). Tota la terminologia introduïda s'utilitzarà en la resta de l'escrit.

Els capítols principals són el III, IV, V i VI. El capítol III descriu el desenvolupament del servidor. Segueix el capítol IV amb la implementació del client i en el capítol V amb la presentació i anàlisi dels resultats. Finalment en el capítol VI es detallen les conclusions i vies de continuació.

L'apèndix mostra uns detalls de la configuració del servidor.

Part II

CONTEXTUALITZACIÓ

En aquests moments, el software lliure s'ha convertit en sinònim d'innovació i progrés. Ús lliure, modificació i distribució de programes, més el fet de poder disposar del seu codi font garantitzen el lliure intercanvi d'idees entre usuaris i desenvolupadors, al mateix temps que permet disposar d'un sistema de llicències apropiat. Tot el software utilitzat en aquest projecte, és gratuït i de codi obert. Compta amb gran quantitat de mòduls, interfícies i subsistemes desenvolupats per comunitats de programadors actives des de fa anys i que continuament afegeixen un nombre significatiu de millores. Addicionalment existeixen llibres, manuals, cursos en línia i abundant documentació a la Xarxa.

6 Sistemes d'informació geogràfica

6.1 Què és un SIG

Un SIG és un sistema de hardware, software i procediments el·laborats per a facilitar l'obtenció, referenciació, gestió, manipulació, anàlisi i modelització de la informació geogràfica amb la finalitat de resoldre problemes de gestió i planificació [1].

6.2 Components d'un SIG

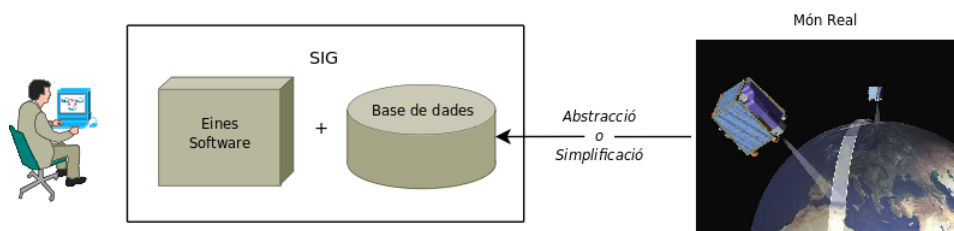


Figura 7: Components d'un SIG

- Les eines software proporcionen la capacitat d'entrada de dades, manipulació i anàlisi, així com la sortida d'informació cap a l'usuari, que té una importància cabdal en el sistema.
- La base de dades conté dades geomètriques i dades descriptives. La construcció d'una base de dades espacial implica un procés d'abstracció per passar de la complexitat del món real a una representació simplificada apta per a ser processada.

- El procés d'abstracció o simplificació consisteix en:
 - Reduir la realitat del dibuix a línies, punts o polígons. Podem definir dos tipus d'abstracció: objectes discrets (una casa) i continus (una elevació). Hi ha dues maneres d'emmagatzemar les dades: **raster** i **vectorial**.
 - Relacionar els objectes en l'espai: topologia. La topologia és el mètode matemàticològic utilitzat per a definir les relacions espaials entre objectes.

El SIG referencia dades espaials en un determinat element del mapa. La informació és emmagatzemada com atributs de la figura gràfica representada. Un SIG aprofita la informació continguda en els atributs dels elements del mapa per a generar nova informació: per exemple longituds i superfícies.

7 Bases de dades Cartogràfiques

En principi un SIG no emmagatzema mapes o imatges: conté bases de dades. Una base de dades espacial és un sistema gestor de bases de dades (SGBD), que gestiona dades existents en un espai. En aquest tipus de bases de dades és imprescindible un Sistema de Referència Espacial (SRE) per a definir la localització i la relació entre els objectes. [20]

Els **sistemes de referència espacial** poden ser de dos tipus:

- **Georeferenciats:** S'estableixen sobre la superfície terrestre. Són els que normalment s'utilitzen.
- **No-georeferenciats:** Tenen un valor x,y. Sovint es donarà el cas que partim d'informació que no té georeferència. En l'entorn CAD pot ser útil, però en aquest projecte caldrà sempre georeferenciar.

La georeferenciació és el posicionament en el que es defineix la localització dels objectes en un sistema de coordenades i un datum determinat. El **datum** geodèsic és la referència a partir de la qual es fan les mesures. En un mapa, cada projecció i cada sistema de coordenades comença per un punt proporcionat inicialment. El punt inicial i la xarxa de punts que s'extén a partir d'aquest és el datum. Cada datum té un nom i sovint una data associada, com l'European Datum del 1950 (ED50), North American Datum del 1927 (NAD27) etc. El punt de referència inicial de l'ED50 es troba a Postdam (Alemanya), a la torre de Hemlert (veure Figura 9).



Figura 8: Xarxa de vèrtexs geodèsics a Espanya, que conformen el datum



Figura 9: Torre de Hemlert

A Espanya, des del 29 d'agost del 2007, el Reial Decret 1071/2007, de 27 de juliol, regula l'adopció del sistema de referència geodèsic global **ETRS89**, substituint el sistema geodèsic de referència regional europeu ED50, oficial fins al moment. Per tant, cal tenir en compte que a dia d'avui tota la cartografia peninsular i balear esta el·laborada amb l'antic ED50 i que estem en un període de transició fins al 2015, quan podran conviure els dos sistemes.^[2]

A efectes pràctics, ETRS89 és equivalent a WGS84, però amb majors precisions a Europa. WGS84 *World Geodetic System*, és un sistema geodèsic mundial, el sistema de referència global utilitzat per OSM i Google i que per defecte vé configurat en la majoria de navegadors GPS.

La diferència de coordenades entre els datums ED50 i WGS84 pot arribar a ser de centenars de metres. Per això, per evitar errors, sempre que es designen les coordenades d'un punt és imprescindible citar, juntament amb el valor, el sistema de referència sobre el qual s'han calculat.

7.1 PostGIS i l'OGC

Una base de dades amb extensió per a base de dades espacial, és un sistema de base de dades normal al qual s'afegeix una capa per a la gestió de la geometria i per a fer el traspàs des de dades comunes a dades espacials transparents per a l'usuari. És el cas de PostgreSQL i PostGIS.

L'Open Geospatial Consortium (OGC) [21] és un consorci industrial internacional format per 334 companyies, agències governamentals i universitats que participen en un procés de consens per desenvolupar especificacions de geoprocessos. PostGIS està certificada per l'OGC cosa que garanteix la seva inter-operabilitat.

També l'OGC ha promogut la especificació Well Known Text (WKT). WKT és una codificació o sintaxi dissenyada per a descriure objectes espacials expressats en forma vectorial. Els objectes que pot descriure són punts, multi-punts, línies, multi-línies, polígons, col·leccions de geometries i punts en 3 i 4 dimensions. És la base d'altres formats més coneguts com KML. PostGIS l'utilitza en la seva versió binària (WKB) en una columna del tipus GEOMETRY.

```
POINT(30 50)
LINESTRING(1 1, 5 5, 10 10, 20 20)
LINESTRING( (1 1, 5 5, 10 10, 20 20) ,(20 30, 10 15, 40 5) )
POLYGON (0 0, 10 0, 10 10, 0 0)
POLYGON ( (0 0, 10 0, 10 10, 0 10, 0 0) ,( 20 20, 20 40, 40 40, 40 20, 20 20)
)
GEOMETRYCOLLECTION(POINT(4 6) ,LINESTRING(4 6,7 10))
POINT EMPTY
MULTIPOLYGON EMPTY
```

Listing 1: Exemples sintaxi WKT

8 Model tessell·les

La tessella és una petita peça de marbre, terra cuita, vidre, . . . amb què es fa un mosaic. A l'antic món romà era molt freqüent l'el·laboració de paviment o mosaics composts per tessell·les.

El model implementat en el servidor de mapes es basa en aquest concepte tant senzill. Opta per una implementació òptima de les dades geogràfiques basada en una pre-renderització i una fragmentació de les dades, on cadascun d'aquests fragments o tessell·les es coneix com a **tile** (en endavant les anomenarem així).

Totes aquestes aproximacions, basades en mantenir formats o estructures internes més optimitzades, acceleren la resposta del servidor.

Clients basats en tiles, com Google Maps, o WMS (*World Map Service*) [22] de l'OGC, han demostrat la utilitat dels serveis de caché de peticions URL per accelerar encara més el servei, reduint el tràfic de xarxa i eliminant peticions redundants.

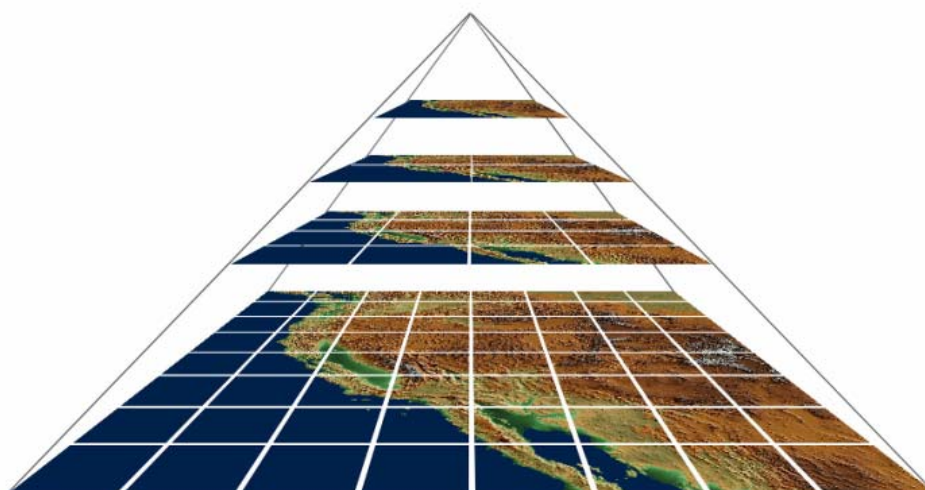


Figura 10: Model de tesselles

Amb aquest model serà possible combinar fonts potencialment grans de dades geogràfiques i obtenir com a sortida imatges regulars. L'escalabilitat del model és evident si s'incrementa la velocitat de renderitzat de *tiles*. Així doncs, coneixent la manera en que els principals proveïdors de mapes fan la projecció i divisió del món, en el capítol de desenvolupament del servidor es veurà l'aplicació de la tècnica.

9 DXF

Els fitxers DXF [23] són fitxers de text ASCII estàndard. Fàcilment es poden traduir a altres formats de sistemes CAD o poden ser processats per altres programes directament.

En general l'organització d'un fitxer DXF és la següent:

1. Secció **HEADER**. Conté informació general del dibuix, com la versió d'AutoCAD i una sèrie de variables associades al dibuix. Cada paràmetre conté el nom de variable i el valor associat.
2. Secció **CLASSES**. Inclou la informació per a classes definides en aplicació, i que són instanciades en les seccions blocs, entitats o objectes.
3. Secció **TABLES**. Conté definicions per les següents taules de símbols:
 - Linetype table (LTYPE)
 - Text style table (STYLE)
 - Layer table (LAYER)
 - View table (VIEW)

- User Coordinate System table (UCS)
 - Application Identification table (APPID)
 - Viewport configuration table (VPORT)
 - Block Record table (BLOCK_RECORD)
 - Dimension Style table (DIMSTYLE)
4. Secció **BLOCKS**. Conté definicions de blocs. Cada definició conté les entitats que formen el bloc tal com s'utilitza en el dibuix. No hi ha blocs aniuats. Per aquest propòsit hi ha les entitats *insert*.
 5. Secció **ENTITIES**. Conté els objectes gràfics (entitats) del dibuix, incloent referències a blocs (entitats *insert*).
 6. Secció **OBJECTS**. Conté els objectes no gràfics del dibuix. Tots els objectes que no són entitats o no són registres de la taula de símbols o les taules de símbols s'emmagatzemen en aquesta secció.

Un fitxer DXF està compostat de molts grups. Cada un ocupa dues línies al fitxer. La primera és el GROUP CODE, i la segona el GROUP VALUE. El format del valor depèn del tipus especificat en el GROUP CODE.

1 0	11 0	21
2 SECTION	12 SECTION	22 ENDTAB
3 2	13 2	23 0
4 ENTITIES	14 TABLES	24 TABLE
5 0	15 0	25 2
6	16 TABLE	26 LAYER
7 [...]	17 2	27
8	18 LTYPE	28 [...]
9 0	19	
10 ENDSEC	20 [...]	

Listing 2: Estructura DXF

```

1       0    % string indicant el tipus d'entitat (fixat).
2       TABLE % valor indicat: es tracta d'una taula.
3
4       0
5       POLYLINE
6       8    % Layer Name (fixat)
7       26   % valor indicat: capa 26

```

Listing 3: Exemple DXF

És interessant saber que DXF no només és un fitxer d'intercanvi per a entorns CAD, sinó que portals importants amb contingut lliure ofereixen les seves dades en aquest format. A la web de l'*Institut Cartogràfic de Catalunya* (ICC) ens podem descarregar les bases topogràfiques a escala 1:5000 en DXF [3] de tot Catalunya. Obtindrem tres fitxers:

1. **xa**. Línies, punts, centroides i textos de l'ALTIMETRIA (3D).
2. **xt**. Línies i textos de la TOPONÍMIA (2D).
3. **xp**. Línies, punts i centroides de la PLANIMETRIA (3D).

10 OSM

OpenStreetMap [24] és una font de cartografia lliure. Informació espacial que qualsevol pot utilitzar sense demanar comptes a ningú. OSM és com la Wikipedia de la cartografia, i l'objectiu principal de Wikipedia és difondre el coneixement. Resulta important digitalitzar zones sense cartografia pública, com a forma de difusió del coneixement i per estar a l'abast dels que tenen menys recursos.

Recentment (juny 2008), Google¹ ha llançat Google Map Maker amb l'objectiu de permetre als usuaris suggerir canvis en la informació geoespacial de Google Maps de cara a solucionar errors, actualitzar informació obsoleta o completar la existent. En aquest aspecte pren l'ideal de OSM. Tot i així tota la informació no anirà a parar a una base de dades lliure, sinó a ser propietat de Google.

Com a usuaris d'OSM podem afegir dades i etiquetar-les. Tenim les següents possibilitats:

1. Fer aparèixer les dades a la pàgina principal d'OpenStreetMap. La pàgina principal conté un '+' a la banda dreta, que al clicar ens dona l'opció per a la capa base, 'Mapnik', actualitzada setmanalment.
2. Generar mapes de bits gràfics renderitzats per nosaltres mateixos utilitzant una eina de render al nostre PC.
 - Mapnik: Un render molt ràpid escrit en C++ que genera mapes de bits.

En voler utilitzar el mapa per als nostres propòsits (no només construir-lo i estendre'l) podem crear-lo basant-nos en dades OSM.

¹En aquesta web podem comparar fins a quin punt ha cartografiat Google, comparat amb OSM. [5]

És possible desenvolupar la nostra pròpia aplicació usant i reutilitzant components OSM. Una visió general dels components ens guiarà per fer aplicacions avançades basades en OSM.

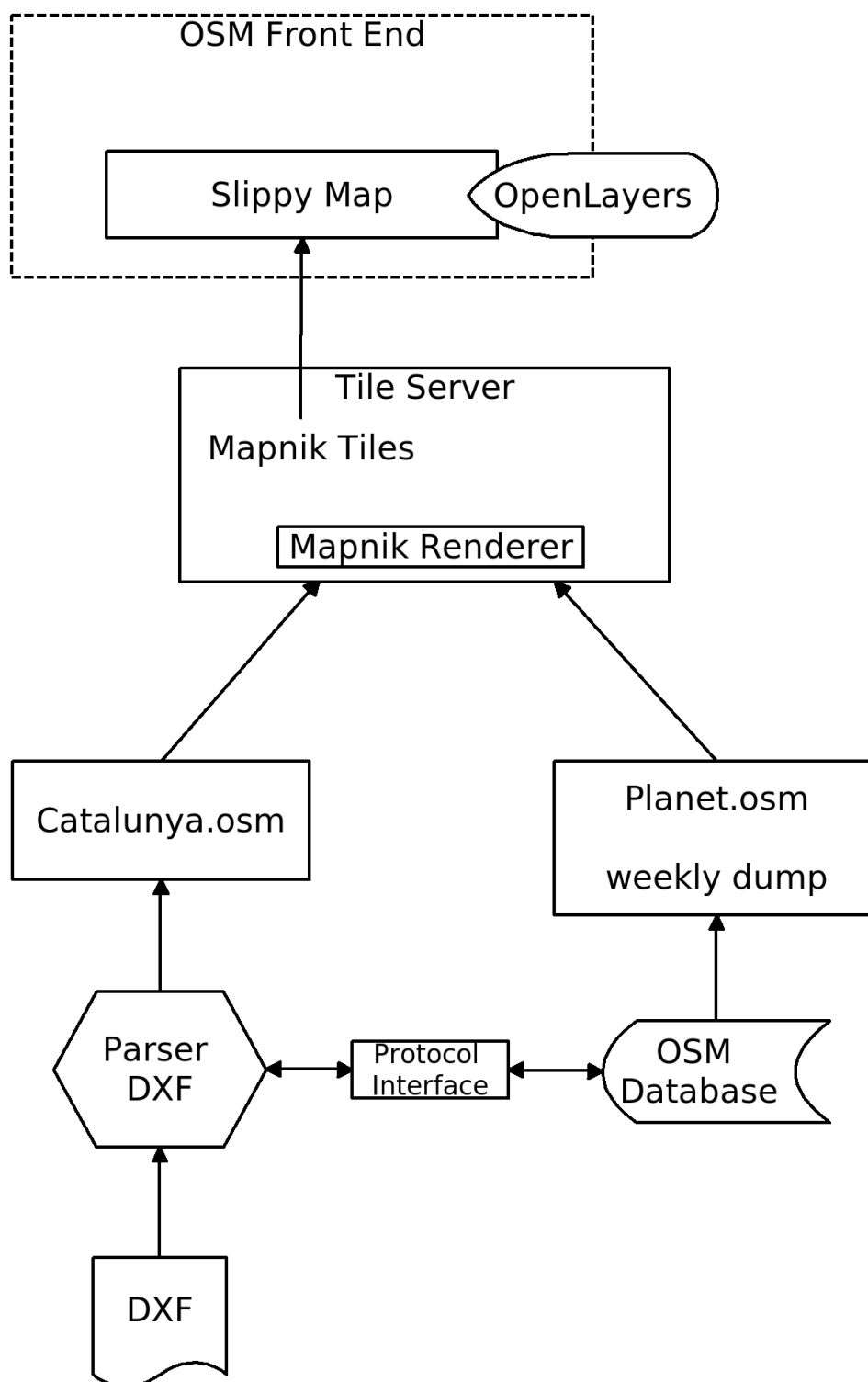


Figura 11: Diagrama de components d'OSM

10.1 Components d'OSM

Base de dades d'OSM La base de dades conté totes les dades del mapa en forma de nodes, camins i relacions. Veure l'esquema de la base de dades a [6]. Aquestes taules contenen les geodades. El que podem visualitzar normalment en un mapa són els nodes i els camins. La taula nodes conté un camp tags per representar les propietats, amb parelles clau/valor. Qualsevol interfície permet afegir aquests elements i editar-ne les claus i valors.

Protocol El Protocol OSM ("l'API") és la interfície per al *web service REST* per llegir i escriure a la base de dades. Altres components OSM poden accedir a la base de dades a través d'aquesta interfície. També s'en pot disposar des d'Internet.

OSM Front End El *front-end* que ofereix la web està escrit en *Ruby on Rails*. Presenta opcions a usuaris identificats, utilitzant connexions directes a la base de dades. La funció principal de la web es mostrar l'"Slippy Map".

Slippy Map És el que mostra el mapa. És una interfície AJAX, que permet fer zoom i desplaçar i ajustar el mapa (desplaçament "slippy").

10.2 Primitives de Dades

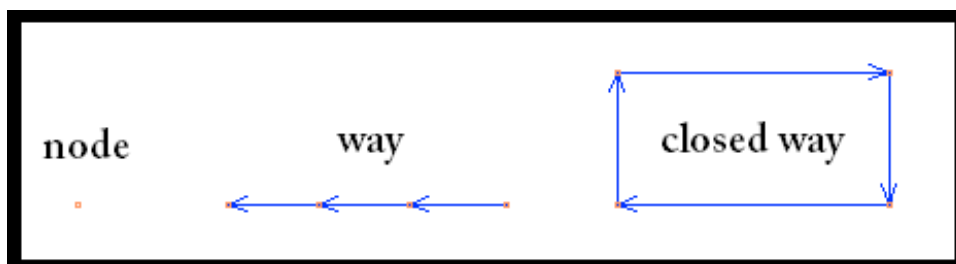


Figura 12: Primitives de dades d'OSM

10.2.1 Way

Primitiva [4] de dades que consisteix en un grup de nodes (almenys 2). Descriuen una característica lineal, com per exemple un carrer. Els nodes poden ser membres de diversos ways. Un way està caracteritzat per unes propietats uniformes, com podria ser la mateixa superfície, el mateix sentit de la marxa, etc. Cal que cada way tingui almenys una parella clau/valor vàlida [8]. Cas especial: àrees. Són ways tancats amb els tag corresponent.

```

<way>
<nd ref="" />
<nd ref="" />
<tag k="oneway" v="yes" />
<tag k="highway" v="unclassified" />
<tag k="created_by" v="Potlatch 0.6" />
<tag k="name" v="Clipstone Street" />
</way>

```

Listing 4: Exemple WAY

10.2.2 Node

Parella longitud / latitud (dades geoespacionals d'un punt). No es codifica la tercera dimensió altitud. Són necessaris per definir els camins, tot i que els podem trobar aïllats.

```

<node id="2549683" lat= lon= user= visible= timestamp= >
<tag k="created_by" v="josm" />
</node>

```

Listing 5: Exemple NODE

10.2.3 Relation

Grups d'objectes on cada un té un rol específic. Per exemple, enclavaments de territori.

```

<relation id="77" visible="true" timestamp="2006-03-14T10:07:23+00:00" user=
  "fred">
<member type="way" ref="343" role="from" />
<member type="node" ref="911" role="at" />
<member type="way" ref="227" role="to" />
<tag k="type" v="turn_restriction" />
</relation>

```

Listing 6: Exemple RELATION

10.3 Objectiu a assolir

Identificats els elements importants de OSM caldrà que l'aplicació d'anàlisi i procés inicialment contempli: afegir tags (clau i valor) als punts (nodes), carrers (ways) i àrees (ways tancats)[8] de manera que software genèric pugui interpretar i mostrar mapes amb unes característiques comunes.[7]

Part III

SERVIDOR

En els capítols d'introducció (I) i contextualització (II) s'ha desgranat quines són les necessitats del projecte, i el seu abast. Un cop fet aquest anàlisi, per assolir els objectius, la primera fita consisteix en la configuració del PC servidor.

Serà necessari com a únic recurs material una màquina, servidor (*GNU/Linux*) amb *Apache*, instal·lat al departament de Telecomunicacions i Enginyeria de Sistemes (TES) de la UAB. Tal com s'ha comentat a (II), el projecte es basa en projectes lliures i per això caldrà parar especial atenció a l'actualització conforme surtin noves versions del software utilitzat.

En el món del software lliure és molt comú referir-se al conjunt de subsistemes necessaris per a configurar llocs web o servidors dinàmics com a *LAMP*. No és res més que un acrònim amb les inicials de les següents tecnologies: Linux (el sistema operatiu), Apache (el servidor web), MySQL (el gestor de bases de dades) i Perl, PHP o Python (els llenguatges de programació). En el nostre cas no utilitzarem MySQL, però existeix la variant de l'acrònim que substitueix la M de MySQL per M de *Middleware* (fent referència als llenguatges de programació) i la P final per la P de PostgreSQL. Podem dir doncs que la combinació a configurar inicialment s'ajusta al que descriu el terme popular LAMP.

- **L:** GNU/Linux Ubuntu Server 8.10, distribuït a l'octubre del 2008
- **A:** Apache/2.2.11 (Ubuntu)
- **M:** Python 2.6.2, PHP Version 5.2.6-3ubuntu4.1
- **P:** PostgreSQL 8.3.3

11 PostGIS i backend

Tal com s'ha exposat en el capítol 7.1, necessitem un servidor de bases de dades espacial. Hem comentat que PostgreSQL², juntament amb l'extensió PostGIS [10] que proporciona les capacitats geoespaciales, complirà amb els requeriments per a la renderització d'imatges en Mapnik.

²És interessant notar que PostgreSQL és el sistema escollit recentment (Abril del 2009) per OpenStreetMap per contenir la base de dades principal del planeta en els servidors *smaug* [9]

Cal tenir espai suficient per a tota la informació cartogràfica, per això s'ha ajustat l'espai de *home* a l'usuari *postgres* a 100 GB. Per al planeta sencer (*planet*) a data de 04 abril del 2009, són necessaris 61 GB incloent taules temporals de PostgreSQL i índexs.

11.1 Instal·lació i configuració

1. Amb el servidor tenim la versió 8.3 de PostgreSQL, per tant instal·lem el postgis 8.3.

```
> sudo apt-get install postgresql-8.3-postgis
```

2. Configuració de PostgreSQL:

```
> sudo vim /etc/postgresql/8.3/main
```

```
shared_buffers = 128MB
checkpoints_segments = 20
maintenance_work_mem = 256
data_directory = '/home/postgresql/8.3/main'
```

Copiar dades de `/var/lib/postgresql/8.3/main` a `/home/postgresql`

Canviar propietari:

```
> sudo chown -R postgres:postgres /home/postgresql/8.3/main
```

Hi ha 3 enllaços que pot ser que no canviïn (fer-ho a mà)

```
> sudo /etc/init.d/postgresql-8.3 restart
```

Pot ser que es queixi de falta de memòria.

```
> sudo sysctl -w kernel.shmmax=268435456
```

```
> sudo vim /etc/sysctl.conf
```

i editar `kernel.shmmax` a mà

3. Configuració de PostGIS:

Crear usuari i base de dades

```
> sudo -u postgres -i
```

```
> createuser toopath
```

```
> createdb -E UTF8 -O toopath gis
```

```
> createlang plpgsql gis
```

Importar PostGIS

```
> psql -d gis -f /usr/share/postgresql-8.3-postgis/lwpostgis.sql
> echo "ALTER TABLE geometry_columns OWNER TO toopath; ALTER TABLE
spatial_ref_sys OWNER TO toopath;" | psql -d gis
```

Repetim el pas 3 per totes les bases de dades que volguem crear, pero canviant el nom “gis” (en aquesta s’importarà el *planet*), per “gisuab” per exemple. A “gisuab” s’importarà la cartografia de la UAB.

11.2 Conversor osm2pgsql

Descarregar al *home* i compilar el codi font:

```
> sudo apt-get install build-essential libxml2-dev libgeos-dev libpq-dev
libbz2-dev
> sudo apt-get install subversion
> cd /home/toopath/
> svn co http://svn.openstreetmap.org/applications/utils/export/osm2pgsql/
> cd osm2pgsql
> make
```

En aquest punt ja es pot importar un fitxer OSM descarregat d’Internet o generat pel client (IV).

Osm2pgsql pot importar un fitxer OSM definint unes fronteres per a una zona geogràfica acotada. Cas d’exemple: en una BDD anomenada “giscat” (Figura 13) s’importa la regió corresponent a Catalunya (definint les fronteres amb el modificador *bbox* i especificades amb 4 coordenades Nord, Sud, Est i Oest), extraient-la del fitxer de tot Espanya.

```
> cd ~/osm2pgsql
> ./osm2pgsql --bbox 0.5,40.53,3.36,42.89 -m -d giscat spain.osm.bz2
```

11.3 Backend

El mateix OSM a punt per ser introduït a la base de dades PostGIS, serà el RAW OSM, element inicial del backend sobre el qual és desenvoluparà tot el software destinat a la resolució de problemes de planificació i gestió. Aquest backend no deixa de ser una base de dades

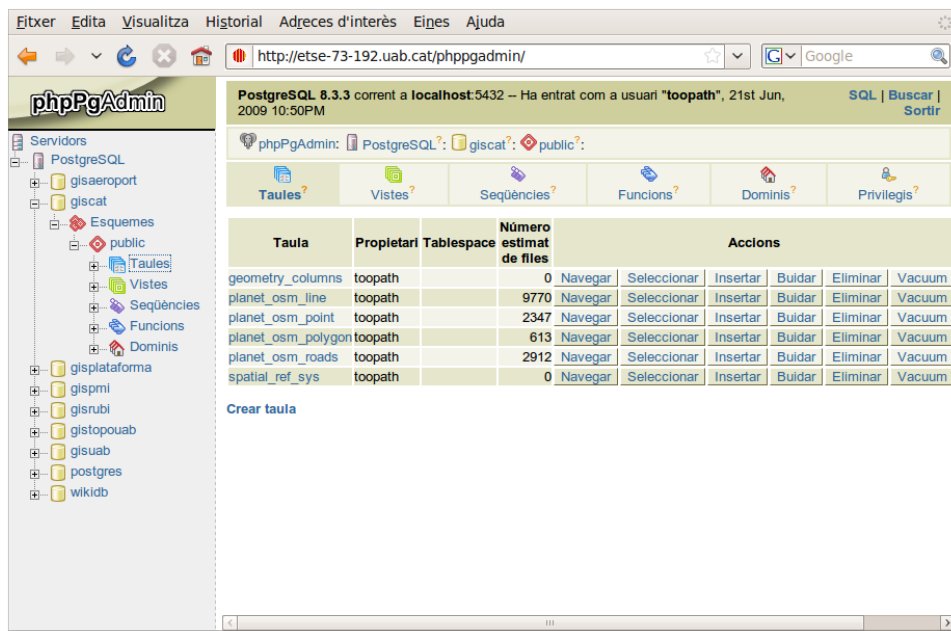


Figura 13: Front-end de PostgreSQL. Taules “giscat”

cartogràfica, on emmagatzemem aquells elements de l’OSM inicial o altra informació afegida en un procés cíclic d’anàlisi i procés, però que és totalment independent de la base de dades PostGIS del servidor de mapes que estem desenvolupant. Serà la finalitat d’un altre document explicar els detalls d’aquest projecte paral·lel.

12 Mapnik

Mapnik [11] s’utilitza per renderitzar tiles de 256 x 256 píxels, que es serveixen des del servidor de *tiles* (Figures 14 i 15). Utilitza la llibreria *Anti-Grain Geometry* per renderitzar amb *anti-aliasing* i amb *Subpixel accuracy*. Pot llegir *shapefiles*, *Rasters TIFF* i qualsevol format que soporti *GDAL* o *OGR*. El motor de fonts és *freetype*.

12.1 Instal·lació

És recomanable seguir la versió aquí descrita perquè hi pot haver porblemes amb *plugins* que porta la versió del repositori d’Ubuntu.

```
> sudo apt-get install libboost-*
> sudo apt-get install libgdal*
> sudo apt-get install imagemagik
```



Figura 14: Tile 0.png

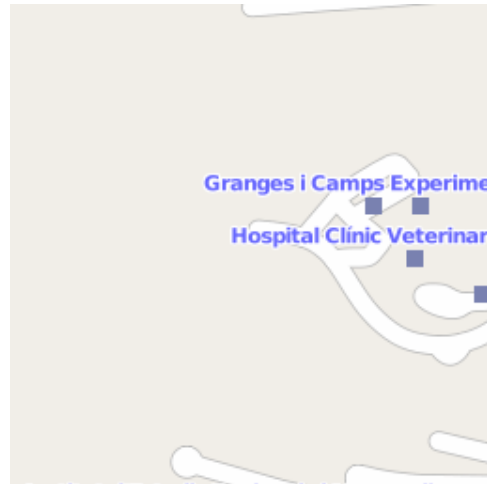


Figura 15: Tile 12224.png

```
> cd /usr/lib
> sudo ln -s libgdal1.4.0.so libgdal.so
> sudo apt-get install python-gdal
> sudo apt-get install freetype libfreetype6-dev
> cd /home/toopath/
> svn co svn://svn.mapnik.org/trunk mapnik
> python scons/scons.py PGSQL_INCLUDES=/usr/include/postgresql
PROJ_INCLUDES=/usr/include PROJ_LIBS=/usr/lib XMLPARSER=libxml2
> python scons/scons.py install PGSQL_INCLUDES=/usr/include/postgresql
PROJ_INCLUDES=/usr/include PROJ_LIBS=/usr/lib XMLPARSER=libxml2
> sudo vim /etc/ld.so.conf i inserir la linia /usr/local/lib
> sudo ldconfig
```

Tot seguit descarreguem les dades de les costes i les fronteres de països.

```
> wget http://tile.openstreetmap.org/world_boundaries-spherical.tgz (51MB)
> wget http://hypercube.telascience.org/~kleptog/processed_p.zip (~246MB)
> wget http://tile.openstreetmap.org/shoreline_300.tar.bz2 (~46MB)
> cd ~/mapnik
> tar xzf world_boundaries-spherical.tgz
> unzip processed_p.zip
> mv coastlines/* world_boundaries/
> rmdir coastlines
> tar xjf shoreline_300.tar.bz2 -C world_boundaries
```

Configurar **set-mapnik-env**. El següent *script* genera un XML que apunta a la base de dades on s'emmagatzemen les dades OSM i que podem reanomenar per a cada cas particular (osm_planet.xml, osm_uab.xml, ...).

```
> vim set-mapnik-env
# This is the name of the database where the OSM data is stored.
export MAPNIK_DBNAME='gis'

> source ./set-mapnik-env
> ./customize-mapnik-map >$MAPNIK_MAP_FILE
```

12.2 Pre-renderitzat

Crear cobertures amb Mapnik i generate_tiles.py

Aquest és el mètode utilitzat per exemple, per *OSM cycle map*. L'avantatge principal és que no ha de córrer res al servidor web, sols necessita un directori d'arxius d'imatge. Es poden transferir les cobertures generades en una altra màquina al host web amb posterioritat. Veiem l'exemple amb el *planet*.

- Baixem el fitxer planet de planet.openstreetmap.org
- Importem a una BD PostGIS utilitzant osm2pgsql
- Instal·lar mapnik i testejar amb osm.xml i generate_image.py
- Quan tot funciona, executar generat_tiles.py per crear mil·lers de cobertures en una jerarquia de carpetes especial.
- Copiar/moure cobertures al document root del servidor web.
- Canviar la instància OpenLayers per tal que apunti allà on tenim les cobertures generades.

12.3 Configuració

Amb els fulls d'estil definits amb XML, generats amb *set-mapnik-env* podem configurar totalment el renderitzat en Mapnik.

```
<Rule>
  <Filter>[amenity] = 'school' or [amenity] = 'college' or [amenity] = '
    university'</Filter>
  <MaxScaleDenominator>25000</MaxScaleDenominator>
  <TextSymbolizer name="name" face_name="DejaVu Sans Book" size="8" fill="
    #000033" halo_radius="2" wrap_width="12" />
</Rule>
```

Listing 7: Exemple de sintaxi osm_uab.xml

A Mapnik, l'ordre importa. Utilitza l'*algoritme del pintor* per dibuixar polígons i text. Això vol dir que quan es dibuixi un element en l'escena, quedarà ja situat, i el que és dibuixi per sobre ocultarà el que queda per sota. Per tant, l'ordre de les capes establert en els nostres fulls d'estil és significant. Els que s'escriuen a dalt del fitxer XML es dibuixen primer, i els de baix es dibuixen últims. Els oceans i les línies de costa han d'anar a dalt, i els punts d'interès i els noms haurien d'anar a baix.

L'ordre també importa en les dades font. Si agafem carrers d'un shapefile o d'una base de dades l'ordre en què ens els retornen és l'ordre en què seràn dibuixats.

13 Mod Tile

És una implementació en C per servir fitxers des del disc i gestionar la cua de renderitzat de forma eficient. El codi s'estructura en dues parts.

1. El **renderd**, un dimoni de renderitzat que crida a Mapnik i escriu els *tiles* al sistema de fitxers.
2. El **mod_tile**, un mòdul d'Apache que gestiona el renderitzat de *tiles* sota demanda.

Si els *tiles* ja s'han renderitzat, l'Apache els servirà eficientment des del sistema de fitxers amb la mínima sobrecàrrega. Cal deixar suficient espai a */var* per les imatges renderitzades (100 GB), on es guardaran amb la següent jerarquia:

```
/var/lib/mod_tile/Default/[Z]/nnn/nnn/nnn/nnn/nnn.png
```

13.1 Instal·lació

```
> sudo apt-get install apache2-prefork-dev
> sudo apt-get install libagg-dev
> cd /home/toopath/
> svn co http://svn.openstreetmap.org/applications/utils/mod_tile
```

13.2 Configuració

És recomanable llegir amb atenció el fitxer **readme.txt**.

1. **Tile Rendering** Modificar el fitxer de configuració `renderd.conf`, i afegir totes les bases de dades

```
[Default]
URI=/planet/
XML=/home/toopath/mapnik/osm_planet.xml
```

```
[UAB]
URI=/uab/
XML=/home/toopath/mapnik/osm_uab.xml
```

2. **Tile Serving** Modificar el fitxer de configuració `mod_tile.conf`

```
DocumentRoot /home/toopath/www/
```

13.3 Execució

Crear cobertures amb Mapnik i `mod_tile`

Aquest sistema ha estat desenvolupat en el servei principal del mapa OSM mundial, recentment (al 2008).

- Baixem el fitxer `planet` de `planet.openstreetmap.org`
- Importem a una BD PostGIS utilitzant `osm2pgsql`
- Instal·lar `mapnik` i testejar amb `osm.xml` i `generate_image.py`
- Compilar i instal·lar el `mod_tile`
- Executar el dimoni de renderitzat i assegurar-se que tingui permís d'escriptura al directori d'emmagatzematge de cobertures.

- Configurar el servidor Apache per a carregar i executar el mòdul.
- Canviar la instància OpenLayers per tal que apunti allà on tenim les cobertures generades.



Figura 16: Petició <http://etse-73-192.uab.cat/uab/0/0/0.png>

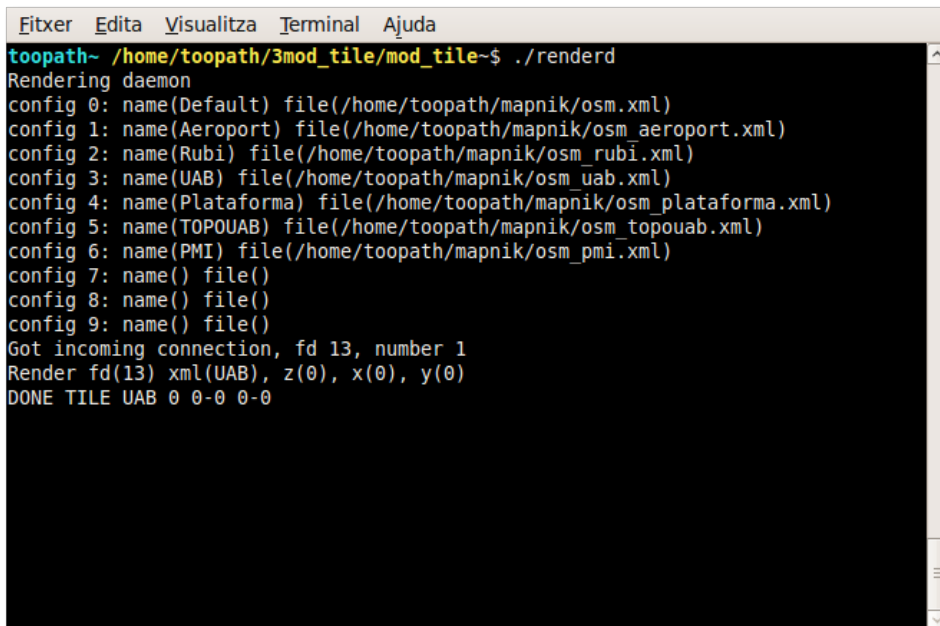


Figura 17: Resposta de mod_tile a la petició de 0/0/0.png

Part IV

CLIENT

14 Anàlisi i procés de DXF

Per tal d'obtenir un OSM que podrem processar al Servidor desenvolupat, cal fer un pre-procés al DXF. A la wiki d'OSM hi ha una apartat on podem buscar maneres de convertir d'un determinat format a un altre. Hi ha una taula amb els formats més comuns per importar d'osm i exportar a osm. Podem veure que el conversor per excel·lència és GPSBabel, però existeixen scripts i altres programes desenvolupats per ajudar a convertir dades, i que diverses persones han deixat a l'abast de tothom.

Per passar a OSM el que podem fer és servir-nos d'un conversor existent de DXF a SHP (shapefile) integrat en el sistema d'informació geogràfica GRASS, i posteriorment shp-to-osm un altre conversor programat per Ian Dees. El procediment seria el següent:

- Conversió dwg a dxf des d'autocad, si es dóna el cas que partim del format AutoCAD dwg.
- Obrir grass i aïllar capes amb v.in.dxf
- Guardar shp amb v.out.ogr
- Conversió shp a osm amb shp-to-osm

Tot seguit mitjançant un exemple s'explicarà aquest procediment. Partim d'un DXF proporcionat per la Unitat de Mobilitat de la UAB, on tenim dibuixats els centres de via dels vials principals en dues capes per separat, les de sentit únic i les de doble sentit. Serà un cas força simple pel fet que ja tenim una bona estructura de capes en AutoCAD i disposem dels centres de via o *centerlines*, que tindran una correspondència directa amb els *Way* d'OSM.

Per altra banda, el fitxer font no té georeferència, és a dir, no conté els *group codes* 40 i 41, i per tant, la importació no s'ajustarà a la regió establerta. Tot i així, es mostrarà com es fa una georectificació partint d'uns punts dels quals si que tenim coordenades.

Disposem d'eines de visualització per comprovar el que estem fent de forma visual en GRASS, obrint les capes vectorials en el *Display Map*. També podem visualitzar els DXF amb *De·Caff DXF viewer* [12], i els SHP amb *ARCexplorer* [13].

14.1 GRASS

Geographic Resources Analysis Support System o GRASS [18] és un software SIG lliure llicenciat amb la GPL desde 1999, inicialment desenvolupat per *U.S. Army Construction Engineering Research Laboratories (USA-CERL, 1982-1995)*, i que ha passat a ser un projecte de la *Open Source Geospatial Foundation*. GRASS és molt conegut i utilitzat al món universitari, a les empreses i a les agències governamentals.

Els passos per fer el procés al DXF des de GRASS són els següents:

14.1.1 Location

Crear una localització o *location* al directori de dades de grass, amb el nom corresponent, per exemple UAB, i el *mapset* vials principals. Per crear una nova localització necessitem principalment la següent informació:

1. El sistema de coordenades per a la base de dades. Normalment triarem **UTM**, però també podrem triar: x,y (per imatges o dades no georeferenciades), latitud-longitud o altres projeccions. UTM (*Universal Transversal Mercator*) és una projecció derivada de la cilíndrica, amb la particularitat que no és única per a tota la Terra, sinó que es fan 60 projeccions i s'aprofita de cadascuna d'elles la superfície en que la distorsió és nul·la o molt petita. Cada porció s'anomena *fus*, i abarca 6 graus de longitud. Així, dividint en fusos la circumferència terrestre, s'aconsegueix un sistema homogeni i igualment vàlid per a qualsevol zona de la Terra.
2. La zona o fus UTM (en el cas de Catalunya, fus 31) i el datum. En el capítol de bases de dades cartogràfiques (7) ja s'ha fet esment a la problemàtica del datum. S'escull el datum ETRS89, ja que s'ajusta a la normativa espanyola i és equivalent a efectes pràctics amb l'extés datum geocèntric WGS84.
3. Les coordenades de l'àrea que serà la regió per defecte i la resolució de la malla d'aquesta regió. En l'exemple de la UAB:
421655.260893018 S
428054.335007676 N
4597458.23749163 E
4592788.16441950 W

14.1.2 v.in.dxf

Importar i aïllar capes amb la comanda v.in.dxf (Figura 18).

```
> v.in.dxf input=/media/disk/UAB/DXF/uab_vials.dxf output=UAB_VIALS layers=
doble_sentit,sentit_unic
```

14.1.3 Georectificació

L'eina de Georectificació transforma les coordenades x/y de l'original desplaçant, rotant i esbiaixant-les el necessari per fer-les coincidir amb el sistema de referència de la localització. El procediment consisteix en identificar un conjunt de punts de control (*Ground Control Points - GCPs*-) a les dades no georectificades. Sabent les coordenades dels punts identificats, es calcula una transformació matemàtica (polinomial de primer ordre) fent regressió dels punts x/y contra les coordenades georectificades pels mateixos punts. Aquesta transformació després s'aplica al conjunt de les dades.

Les coordenades dels GCPs utilitzats per georectificar el plànol de la UAB, s'han extret de Google Earth en longitud/latitud decimal i s'han convertit a UTM mitjançant un conversor [14].

A l'iniciar l'eina *Georectify* des del menú *File*, s'ha de triar la location/mapset del mapa a rectificar, crear el grup que el contindrà i el mapa a utilitzar per marcar els GCPs. S'obre un map display on es pot fer el zoom desitjat per marcar amb un cursor els punts (Figura 20), i una finestra de gestió de GCPs (Figura 21) on es poden veure les coordenades x/y, les coordenades geogràfiques i l'error per cada GCP (la desviació de la localització esperada al aplicar l'equació de transformació). Cal marcar com a mínim 4 GCPs per una transformació de primer ordre. Aquells GCPs que tinguin un error massa elevat es poden excloure. El càlcul es fa al clicar el botó RMS tool, i ens actualitza l'error total. Experimentalment s'ha vist que un error menor que 1 genera bons resultats.

14.1.4 v.out.ogr

Exportar a SHP amb la comanda v.out.ogr (Figura 19).

```
> v.out.ogr input=UAB_VIALS_GEORECTIFIED@UAB dsn=/media/disk/UAB/SHP/UAB_VIALS
format=ESRI_Shapefile
```

14.2 Conversor shp-to-osm

Es tracta d'una utilitat java que té la possibilitat d'aprofitar de forma eficient els atributs del shapefile, i permet reprojectar si tenim shapefiles en diferents projeccions a la utilitzada

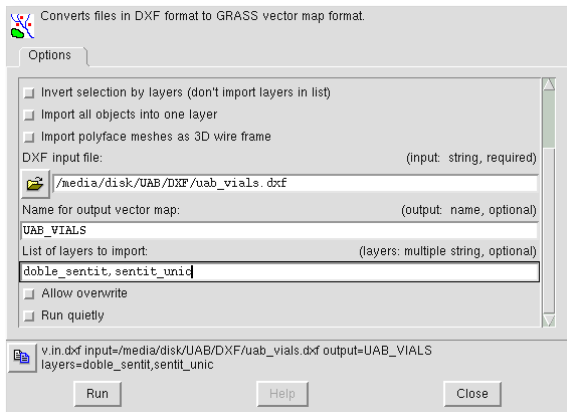


Figura 18: v.in.dxf



Figura 19: v.out.ogr

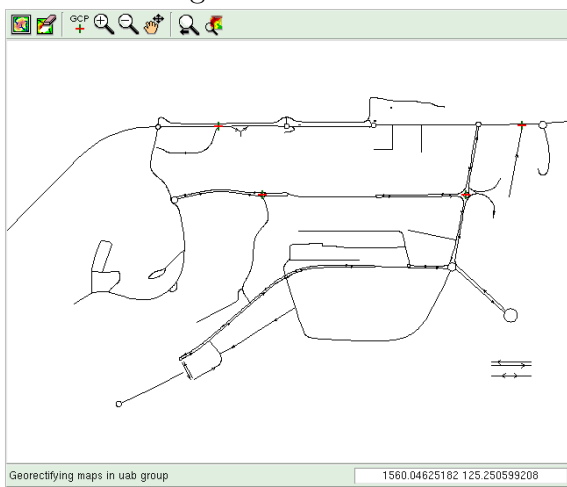


Figura 20: Georectificació

Use	xy coordinates	geographic coordinates	forward error	backward error
<input checked="" type="checkbox"/>	[1093.233458 904.146534]	425224.708929 4594996.787540	0.616018	0.614892
<input checked="" type="checkbox"/>	[1960.769729 902.693901]	425873.805706 4594420.089197	0.607685	0.604440
<input checked="" type="checkbox"/>	[2198.262748 1198.023286]	426349.849111 4594481.241040	0.406467	0.403306
<input checked="" type="checkbox"/>	[905.114824 1194.248174]	425275.717795 4595336.733903	0.414799	0.413758
<input type="checkbox"/>			0.0	0.0
<input type="checkbox"/>			0.0	0.0
<input type="checkbox"/>			0.0	0.0
<input type="checkbox"/>			0.0	0.0

Forward RMS error = 0.466054260369, backward RMS error = 0.464174815659

Figura 21: Punts de control i error

per OSM. Utilitza la llibreria lliure per a Java *GeoTools*, que proporciona mètodes per a la manipulació de dades geoespaciales.

Per fer el *matching* d'atributs del SHP amb tags a l'OSM podem configurar el fitxer *rules.txt*, que conté 5 camps:

1. Tipus en el shapefile (outer, inner, line, point)
2. El nom d'atribut del fitxer font a comparar.
3. El valor d'atribut del fitxer font a comparar. Pot ser buit per a processar tots el valors.
4. El nom del tag OSM a aplicar quan la parella clau/valor coincideix.
5. El valor del tag OSM a aplicar. Utilitzar un guió simple (“-”) per prendre el valor original.

En l'exemple de la UAB, afegim al fitxer *rules.txt* les següents línies:

```
line , cat , , highway , residential
line , layer , sentit_unic , oneway , yes
```

Listing 8: rules_uab.txt

El que indiquen és que totes les línies del SHP, sigui quina sigui la seva categoria, correspondran a vies residencials en el fitxer OSM. A més indiquem que les que provinguin de la capa *sentit_unic* del fitxer DXF (que ha quedat guardat en l'atribut *layer* del SHP), tindran el tag OSM “oneway”=*yes*.

14.2.1 Execució

```
> java -cp shp-to-osm-0.4-jar-with-dependencies.jar Main <path to input
shapefile> <path to rules file> <path to output osm file> [-t]
```

14.3 Importació al Servidor

Ja disposem doncs del fitxer OSM georeferenciat i amb els tags corresponents. L'últim pas que queda és el d'importar-lo al servidor amb l'aplicació *osm2pgsql*, i ja podrem passar al següent punt, que és configurar la pàgina que en capturarà la informació i en mostrarà el mapa en *OpenLayers*.

```
> cd ~/osm2pgsql
> ./osm2pgsql -m -d gisuab ../shp-to-osm/vials_uab.osm
```


15 Open Layers

OpenLayers [25] és un framework de mapes en la web, que permet als usuaris capturar diverses fonts de dades i mostrar-les en una interfície de mapes AJAX. L'API és JavaScript pur i ofereix suport per a nombrosos formats estàndard que són integrats com a *Layers*. Per exemple OGC WMS, GeoRSS, OpenStreetMap, etc. OpenLayers inicialment fou desenvolupada per Metacarta, i després es va començar a distribuir lliurement amb la llicència BSD, i igual que GRASS va passar a ser un projecte de l'Open Source Geospatial Consortium (OSGC). La versió 2.7 va sortir al públic al setembre del 2008.

15.1 Classes principals

OpenLayers és orientat a objectes i es divideix en múltiples classes.

- `OpenLayers.Map` És la classe principal. Compila el mapa principal en l'aplicació i proporciona mètodes per l'administració com són les capes a mostrar, els components operatius i el zoom i desplaçament. A més aquesta classe permet múltiples consultes de l'estat del mapa actiu a través de mètodes "get".
- `OpenLayers.Layer`. És el component més significatiu. `Layer.js` produeix capes individuals, estableix la transparència i resolució de cadascuna i conté els mètodes get bàsics. És la classe de la qual hereten les classes especials segons tipus: basades en tiles (com TMS *Tile Map Service*), no basades en tiles, propietàries de tercers (Google), capes imatge (per a ràsters), *canvas* per a overlays.
- `OpenLayers.Control`. Els elements operatius, relacionats amb la navegació i la informació a mostrar. També és classe base d'una sèrie de classes de controls concrets com el `PanZoom` o el `PanZoomBar`.
- `OpenLayers.Tile`. Quan el mapa segueix el model de tiles ja explicat, cadascun d'ells es defineix amb un objecte. Per generar un tile, cal tenir la capa corresponent, la posició del píxel, dimensions geogràfiques del tile i la URL i la mida de píxel del tile. La mida estàndard del tile són 256 píxels.

15.2 Instal·lació

Tant sols cal fer la còpia en el *Document Root* del Servidor on allotjarem les pàgines del fitxer *openlayers.js*, o bé fer un enllaç.[15]

A més, podem disposar d'una classe particularitzada per a l'ús en el portal toopath, en que s'hereta de la classe `OpenLayers.Layer.TMS` i es sobreescrueu el constructor i el mètode `getUrl`. Continuant amb l'exemple de la UAB, ens servirem d'aquesta classe particular que es troba en el fitxer `toopathmaposm.js`.

15.3 Codificació d'una pàgina client

Per accedir a les dades dels vials de la UAB, només cal saber la URL del servidor (la mateixa que s'ha introduït al fitxer de configuració del mod_tile `renderd.conf`) i el nom a mostrar a l'usuari. El mapa `OpenLayers` consistirà en un objecte `OpenLayers.Map` i un o més objectes `OpenLayers.Layer`. Tot seguit es mostra un *script* simple d'`OpenLayers` i l'HTML que l'acompanya per veure les imatges dels vials de la UAB en una pàgina web:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>UAB</title>
<link rel="stylesheet" href="css/style.css" type="text/css" />
<script src="js/openlayers.js"></script>
<script src="js/toopathmapsosm.js"></script>
<script type="text/javascript">
var map, layerUAB;
var bounds = new OpenLayers.Bounds(2.1337,41.5241,2.0711,41.4801);

var lat=41.5021;
var lon=2.1024;
var zoom=14;
function init(){
  OpenLayers.ImgPath = "http://etse-73-192.uab.cat/js/img/";
  map = new OpenLayers.Map('map', {maxResolution:'auto',
    numZoomLevels: 25
  });
  layerUAB = new OpenLayers.Layer.tooPath("UAB",
    "http://etse-73-192.uab.cat/uab/", {layers: 'basic' } );
  map.addLayer(layerUAB);
  map.addControl(new OpenLayers.Control.LayerSwitcher());
  var lonLat = new OpenLayers.LonLat(lon, lat).transform(new OpenLayers.
    Projection("EPSG:4326"), map.getProjectionObject());
  map.setCenter (lonLat, zoom);
}
</script>
</head>

<body onload="init()">
```

```

<h1 id="title">UAB</h1>
<div id="map" style="width: 600px; height: 500px"></div>
</body>
</html>

```

Listing 9: Exemple HTML OpenLayers vials UAB

16 Component User Parameters

16.1 Joomla!

Joomla! és un sistema gestor de continguts dinàmics (CMS) que permet crear llocs web d'alta interactivitat amb pocs coneixements tècnics [16]. L'última versió que ha publicat *Open Source Matters*, l'entitat que en fa el suport, és la 1.5.11 a data de juny del 2009, amb llicència GPL.

Joomla! està programat en llenguatge PHP i SQL. Utilitza una base de dades relacional MySQL, per guardar tota la informació i gran part de la configuració del sistema. Els *scripts* PHP són els que executen les accions de consulta i fan les modificacions en la base de dades convertint dades en simples pàgines web interpretables pel navegador. A més hi ha altres fitxers per realitzar tasques dins de Joomla! com són XML, scripts Javascript i CSS.

Els elements que conformen el CMS són:

- Un frontend: **site** el lloc web, i un backend: **admin** on es poden fer tasques de configuració, manteniment, estadístiques, etc.
- *Configuration Settings*. Decisions de configuració, opcions de control d'alta, títol de la web i altres funcions.
- Drets d'accés. En un rang entre usuaris registrats simples, autors o editors i administradors.
- Contingut. Joomla! permet qualsevol tipus de contingut: text simple, imatges, vincles, arxius multimèdia o una combinació de tots ells. Per a una bona presentació s'inclouen en estructures jerarquitzades, classificant-se per seccions i categories.
- Plantilles. Defineixen els colors, fonts, mides, imatges de fons, espais i distribucions de pàgines, és a dir, l'aparença.

- **Extensions.** Components. Tot sistema és ampliable per poder créixer en funció dels requeriments que apareguin. Les funcionalitats que pertanyen a un determinat context es cobreixen per el que coneixem com a component. Per exemple, components típics són una botiga en línia, un gestor d'usuaris, un gestor de llistes de correu, un fòrum, una galeria d'imatges, un gestor de descàrregues,...

Utilitzant el framework que proveeix es poden desenvolupar components per a Joomla!. Mentre que la idea d'un component pot semblar senzilla, el codi pot complicar-se ràpidament al anar afegint funcionalitats o adaptant la interfície.

El *Model-View-Controller* (MVC) es un patró de disseny que s'utilitza per organitzar el codi de manera que l'organització lògica i la presentació de les dades es separen. La descripció, breument, d'aquestes tres parts és:

- **Model.** Encapsula les dades de l'aplicació. Proporciona rutines per gestionar i manipular dades segons el seu tipus, així com la consulta d'aquestes. D'aquesta manera, si una aplicació ha de migrar el seu sistema d'emmagatzemament de la informació en fitxers plans, per exemple, a emmagatzemar-la en una base de dades, el model és l'únic element a modificar per aplicar els canvis, view i controller es mantenen.
- **View.** S'utilitza per a renderitzar les dades en el model de forma que sigui possible la interacció. Per a aplicacions basades en web. la vista (view) seran generalment pàgines HTML. La vista agafa les dades del model (que li passa el controlador) i estructura les dades en un *template* que s'omple i es presenta a l'usuari.
- **Controller.** És el responsable de donar resposta a les accions de l'usuari. En el cas d'una web generalment l'acció és una *request* d'una pàgina. El controlador determina quina és i respon apropiadament passant el model corresponent a la vista. El controlador només dispara mètodes del model que modifiquen les dades en les taules internes i després passa el model a la vista que és la que mostra les dades.

Toopath ha estat desenvolupat seguint aquest model i d'acord al patró de disseny MVC [17]. Els components desenvolupats per l'equip de recerca LogiSim amb el suport de DLM-Solutions, permeten el seguiment en línia de dispositius GPS i la visualització de rutes amb els components toopathTracking i toopathTracks respectivament. Altres components com toopathBase, toopathDevices, toopathDownloads permeten a l'usuari configurar el portal segons les seves necessitats. Pot afegir dispositius i configurar-los, descarregar-se rutes, mostrar-les amb la base de mapes del servidor de OpenStreetMaps, de Google i de l'Institut Cartogràfic de Catalunya.

La voluntat d'integrar el present projecte amb Toopath [19] ha fet evident la necessitat de crear un nou component que permeti a l'usuari configurar el mapa a mostrar. Introduint una sèrie de paràmetres, entre ells la URL del servidor, la instància d'OpenLayers de la secció de visualització de rutes i la de seguiment de dispositius accedirà als mapes del servidor desenvolupat. S'anomenarà User Parameters.

16.2 User Parameters

Tots els components de toopath emmagatzemen les dades en taules MySQL. Per al component User Parameters és necessària una taula nova "tp_userparams", amb una clau forana UID (user id) per fer la relació amb els usuaris.

```
CREATE TABLE IF NOT EXISTS 'tp_userparams' ('id' INT(10) AUTOINCREMENT
UNSIGNED NOT NULL, 'uid' INT(10) UNSIGNED NOT NULL, 'urlserver' VARCHAR
(200) default NULL, 'layername' VARCHAR(100) default NULL, PRIMARY KEY
('id'), FOREIGN KEY ('uid')) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Listing 10: Taula MySQL user params

16.2.1 Model

Els mètodes PHP principals afegits al model són:

- **_buildQuery**. Construeix un string amb la consulta SQL per obtenir la url (urlserver) i el layer name (layername) de la BDD.
- **_getData**. Fa un canvi de BDD de la interna de Joomla a la de toopath, organitza la informació rebuda per la consulta en un array i la retorna.

16.2.2 View

Es modifica la funció *display*, per tal que les dades que retorna *_getData*, es passin al template utilitzant el mètode *JView::assignRef*. Les variables assignades amb aquest mètode s'accedeixen des del template utilitzant:

```
$this->{propertyname}
```

Molt resumidament el template de User Parameters tindrà un formulari HTML en el cas de la introducció, i una taula HTML en el cas de mostrar dades de l'usuari.

16.2.3 Controller

Els mètodes PHP principals afegits al controlador són:

- **edit()**. Mostra el formulari d'edició.
- **save()**. Desa els canvis del formulari en el model.
- **delete()**. Borra registres del model.

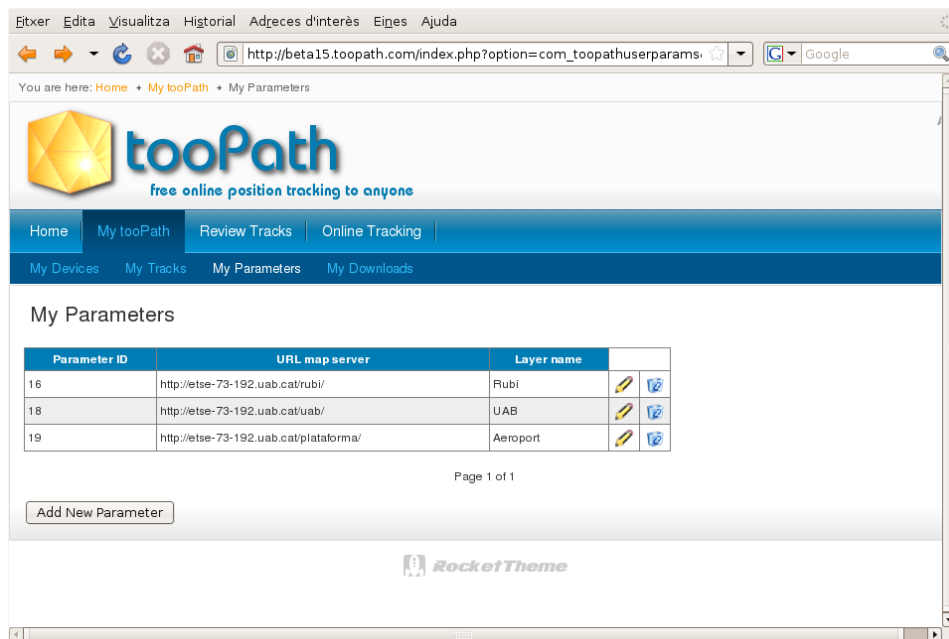


Figura 22: Component User Parameters

16.3 toopathTracking

Un cop tenim els paràmetres, cal que des de seguiment de vehicles es carregui el mapa seleccionat per l'usuari. Per això modificarem el component `toopathTracking` per tal que ens mostri un formulari de selecció de capes de mapa a mostrar. Quan l'usuari seleccioni una d'aquestes, s'afegirà el següent codi en el constructor de la classe `OpenLayers toopathtrackmap.js`.

```
addLayer(url, layername, id);
```

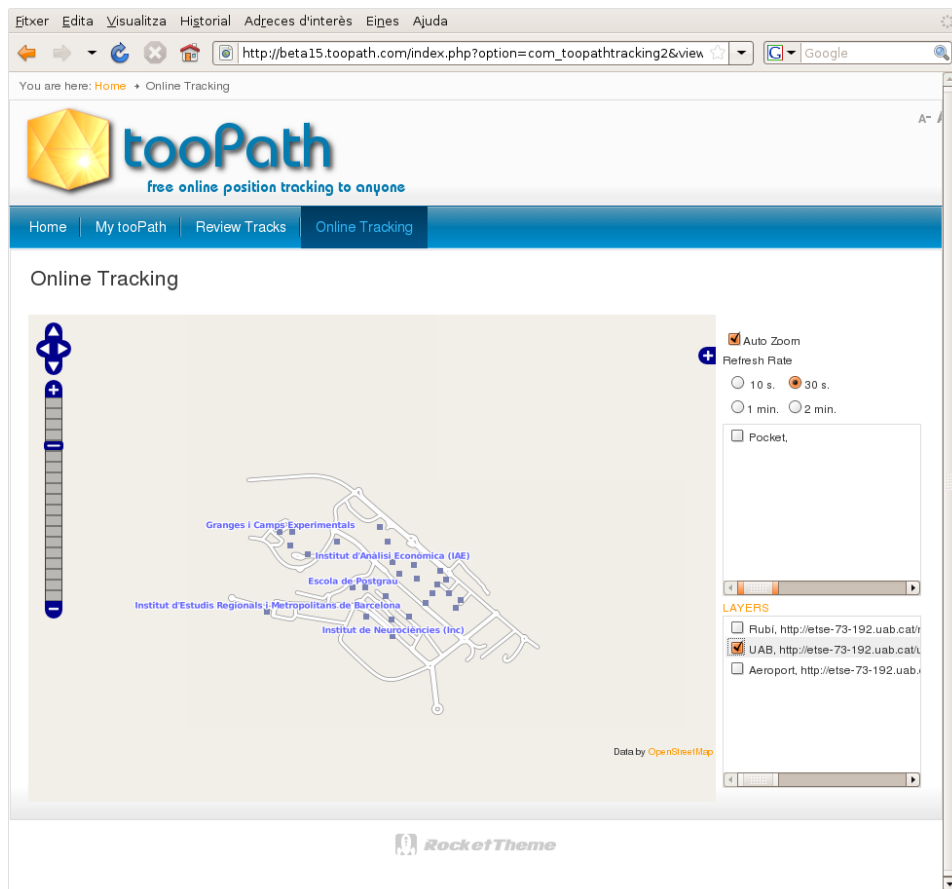


Figura 23: LAYERS a Component Tracking

Part V

RESULTATS

Els treballs i experiments realitzats que conclouen en resultats demostrables són, bàsicament, els del projecte “Mobilitat UAB”. S’ha treballat també amb plànols de l’aeroport de Mallorca en un projecte del qual no es mostraran resultats per motius de confidencialitat. També s’ha treballat sobre plànols de Rubí i l’Aeroport de Barcelona, però sols en fase d’experimentació.

Amb el plànol de Rubí es va aconseguir generar la primera base de dades en el servidor i posar en marxa el mod_tile per generar *tiles*. Amb el plànol de l’aeroport es va experimentar partint d’una font de dades poc estructurada, ja que era un fitxer PDF, que es va convertir a *.SVG* (format de dibuix vectorial) i posteriorment a DXF. Es va aconseguir generar la base de dades per a la planta sencera en una única capa. El graf subjacent per això, no servia per a propòsits de planificació i gestió, ja que representava edificis, pistes, senyalització, etc. en una sola capa. Seria necessària una iteració en l’entorn CAD per a afegir aquesta estructura i alhora una semàntica als elements.

El treball realitzat amb el plànol de la UAB, buscant resposta a les funcionalitats que requeríem en els objectius ha resultat en:

Resultat 1: Servidor de mapes En les següents figures es pot veure el DXF inicial i després d’aïllar les capes dels vials:

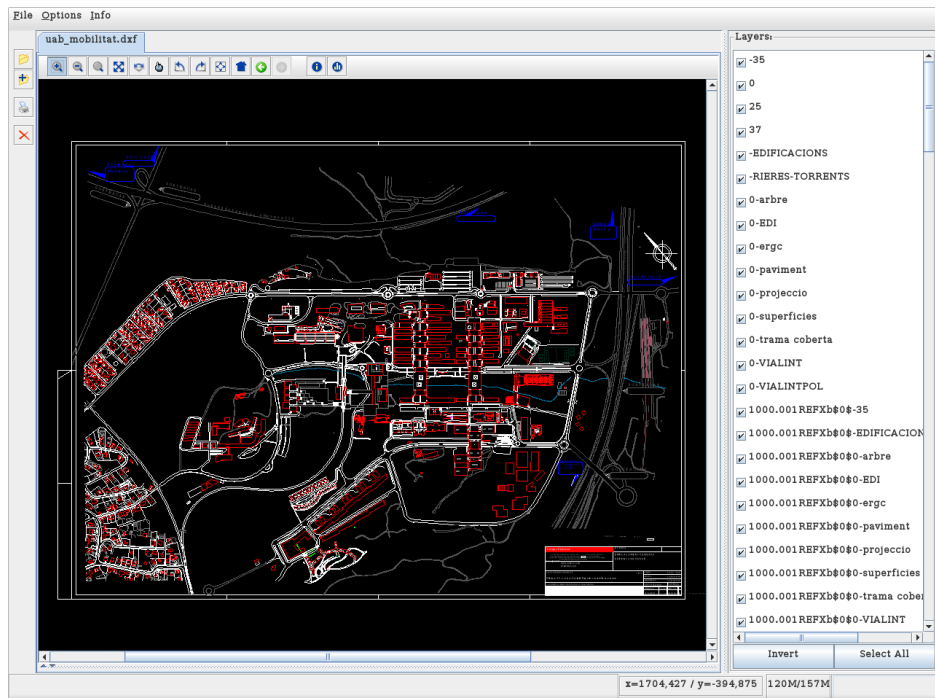


Figura 24: Plànol DXF - UAB

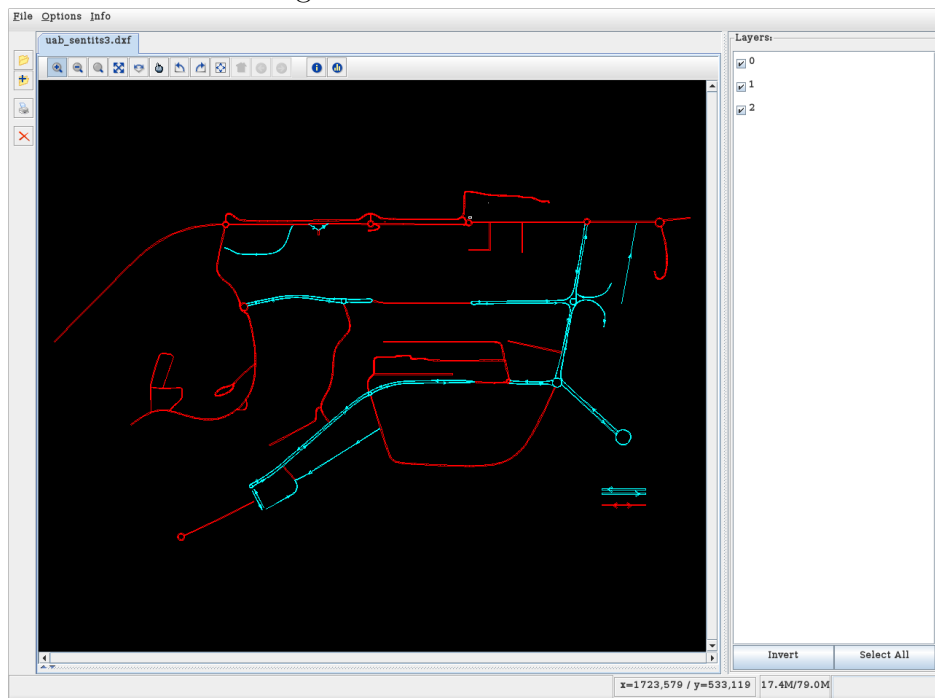


Figura 25: Plànol DXF capes de vials aïllades - UAB

A continuació, tal com s'explica en el capítol d'anàlisi i procés de DXF, mostrem el resultat de la georectificació i part de l'OSM generat:

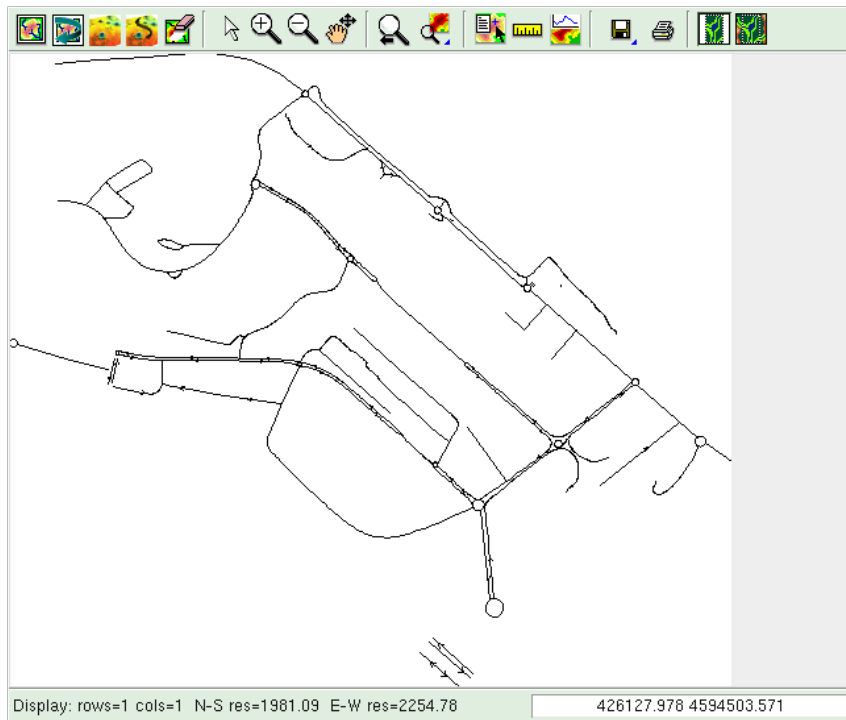


Figura 26: SHP georectificat - UAB

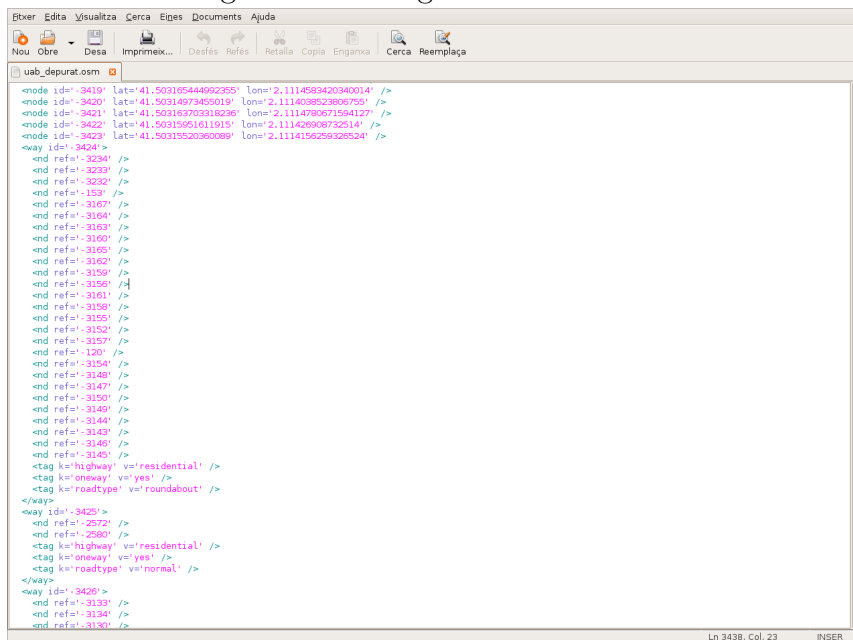


Figura 27: OSM generat - UAB

L'OSM es *parseja* i s'importa a la base de dades espacial:

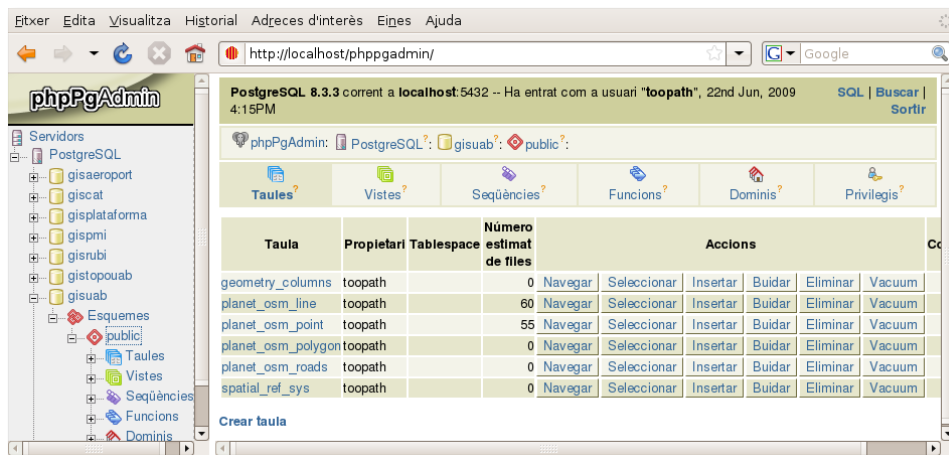


Figura 28: Base de dades - UAB

El resultat final es renderitza a la pàgina web que accedeix al servidor de tiles (amb diversos zooms).

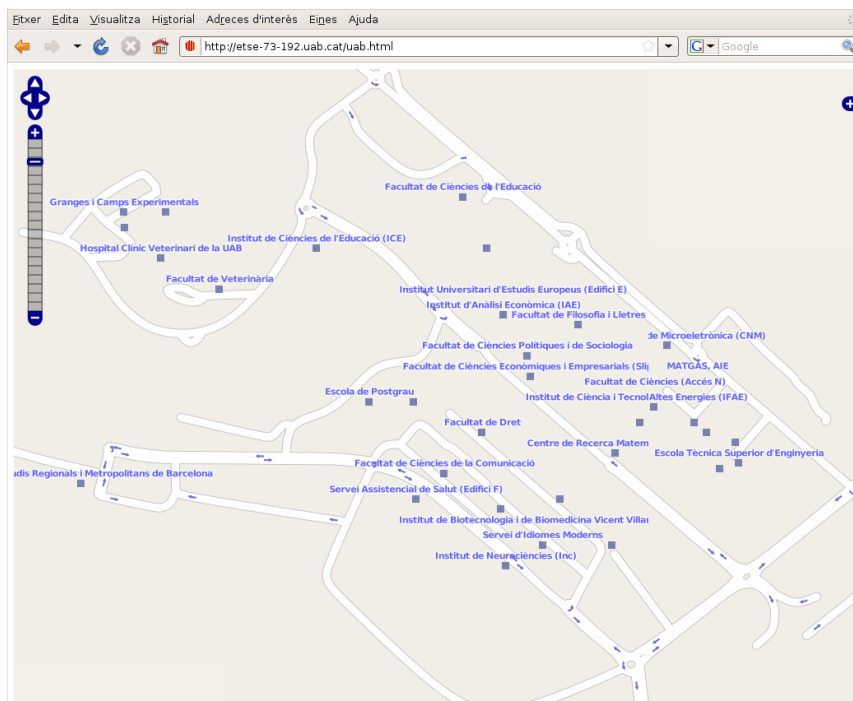


Figura 29: Portal web - UAB

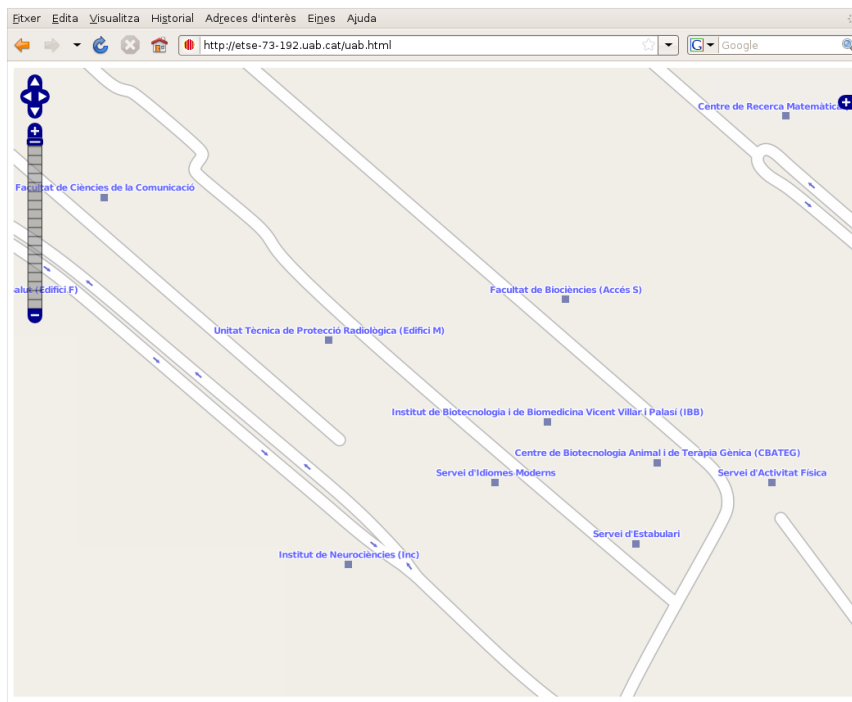


Figura 30: Portal web zoom 19 - UAB

Resultat 2: Sistema d'informació geogràfica Fent un pas més en l'anàlisi de les dades obtingudes en el Resultat 1 i pensant en una estructura de capes, podem afegir el que anomenem geocoding. Per exemple, a un node del graf se li assigna una etiqueta corresponent a una facultat, o a un pàrquing.

La geocodificació de les portes d'entrada a les diverses facultats de la UAB, permet l'execució d'una aplicació de càlcul de camins mínims amb dos paràmetres d'entrada, origen i destí. En les següents figures es pot veure un formulari codificat amb AJAX on es pot seleccionar l'origen i el destí per nom (és *case in-sensitive* i *accent in-sensitive*). La ruta òptima calculada entre aquests punts es visualitza superposada amb una capa *GPX/KML* (la línia verda).

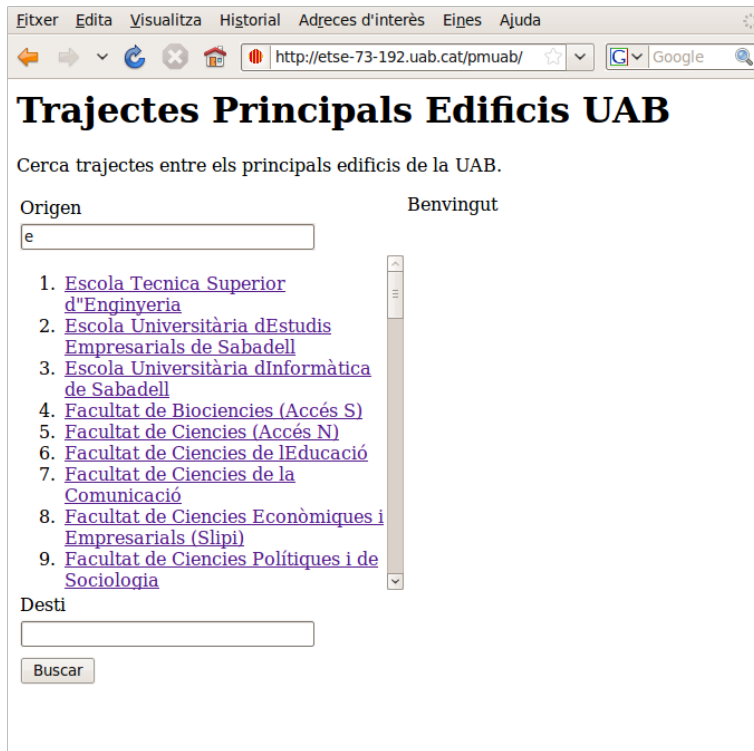


Figura 31: Origen, Destí - UAB

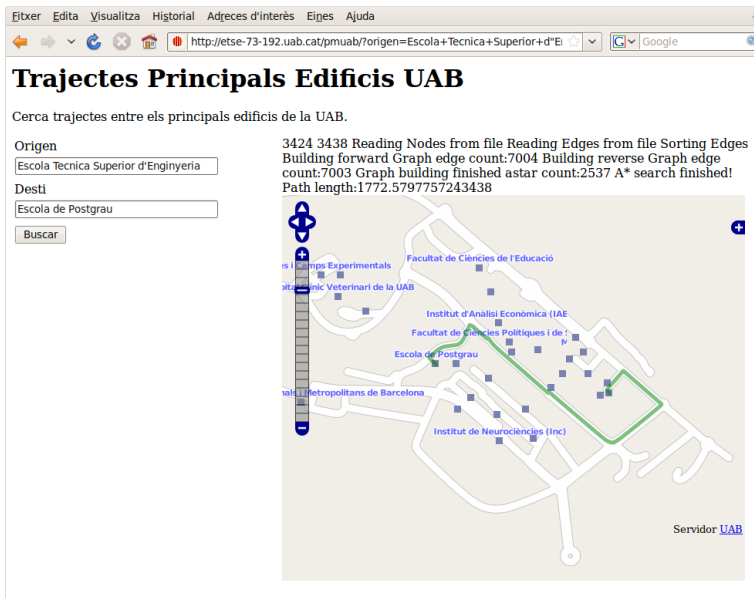


Figura 32: Ruta - UAB

Part VI

CONCLUSIONS I VIES DE CONTINUACIÓ

17 Conclusions

Els objectius d'aquest projecte s'han assolit i s'ha desenvolupat un servidor d'informació geogràfica i s'ha estudiat en detall una tècnica per automatitzar el pas de format cartogràfic a informació accessible des de la web.

El disseny del servidor amb una estructura de base de dades i un mòdul que renderitza les imatges segons el model de tessel·les és una solució àmpliament acceptada. La conclusió principal d'aquest treball és que la solució és senzilla, òptima i escalable. A partir d'informació amb un sistema de referència espacial, disposada en una base de dades cartogràfica, les eines existents per arribar al resultat, són de fàcil ús i aporten moltes possibilitats.

Veiem que partint d'informació dibuixada en entorn AutoCAD podem analitzar-la i processar-la per aconseguir un fitxer XML (OSM) base del software renderitzador d'imatges. Amb els resultats obtinguts d'aquest procés s'assoleixen els requeriments dels servidor d'acord a les seves especificacions.

S'ha aconseguit també l'objectiu de servir aquesta informació dins l'aplicació de seguiment de dispositius GPS, creant el component necessari per permetre-ho sempre ajustant-se al patró de disseny MVC. A més mitjançant OpenLayers s'han creat les pàgines client per mostrar els resultats i fer els experiments necessaris.

Bona part del temps dedicat a aquest projecte ha sigut el d'estudi de la tècnica per fer la conversió de fitxers DXF, amb i sense georeferència, a OSM. S'ha comprovat que les eines triades compleixen les expectatives i els resultats són els esperats. En obtenir fitxers OSM amb una clara codificació espacial i dotada d'unes primitives de dades ben estructurades, s'han obert noves vies d'investigació seguint una de les motivacions plantejades, la de resoldre problemes de planificació i gestió basats en informació cartogràfica.

18 Vies de continuació

18.1 Portal de Mobilitat de la UAB

Aquest nou projecte té com a objectiu el desenvolupament dels mapes de detall al campus (desplaçaments en vehicles i a peu) així com dels mòduls necessaris per integrar amb *POMO* (portal de mobilitat sostenible per a la UAB) la mobilitat dins del campus de la UAB. El present projecte hi contribueix en el desenvolupament de la BDD cartogràfica de la UAB (vies en vehicles motoritzats i a peu) amb dades de partida sigui plànols AutoCAD o captura de dades de camp via GPS.

18.2 Altres vies de continuació

- Anàlisi i procés d'informació addicional. Queda pendent l'estudi de la incorporació de noms de via a partir de bases de dades de tercers (per exemple el cadastre).
- Estudi per a l'extracció de les *centerlines* dels carrers en un preprocés raster o vectorial.
- Paral·lelització del servidor. Comprovar el rendiment del sistema i fer testos exhaustius sobre una màquina amb múltiples CPUs, o bé un DataCenter amb paral·lelització de processos.
- Provar altres eines de georectificació (ArcGIS, CompeGPS, ...)
- Unificació de codi en els components de toopath.

Bibliografia

Referències

- [1] Garrido, J. P., “Curs pràctic sobre Sistemes d’Informació Geogràfica”, *Centre Tecnològic Forestal de Catalunya*, Solsona, setembre de 2008.
- [2] García-Abad Alonso, J. J., “Antecedentes y situación actual de los sistemas geodésicos en España”, *Introducción a la Geografía Física, Licenciatura en humanidades*, Departamento de Geografía. Universidad de Alcalá, març de 2008.
- [3] “ESPECIFICACIONES PER AL FORMAT “AutoCAD Drawing Exchange File” (DXF)”, *Institut Cartogràfic de Catalunya*, desembre 2005.
- [4] http://wiki.openstreetmap.org/wiki/OSM_Protocol_Version_0.6/DTD, DTD d’OSM
- [5] <http://lamp2.fhstp.ac.at/~lbz/beispiele/ws2008/capitals/index.php>, Comparativa OSM - Google Maps.
- [6] <http://www.openstreetmap.org/Wiki/Database/Model>, Model de la base de dades OSM.
- [7] http://wiki.openstreetmap.org/wiki/Good_practice, Bones pràctiques OSM.
- [8] http://wiki.openstreetmap.org/wiki/ES:Map_Features, Elements de OSM.
- [9] <http://wiki.openstreetmap.org/wiki/Servers/smaug>, Servidor del *Planet* OpenStreetMap.
- [10] <http://www.refractions.net/>, Desenvolupadors de PostGis
- [11] <http://mapnik.org/>, Mapnik.
- [12] <http://caff.de/dxfviewer/>, DXF viewer.
- [13] <http://www.esri.com/software/arcexplorer/explorer.html>, ArcGIS explorer.
- [14] <http://www.apsalin.com/convert-geodetic-to-universal-transverse-mercator.aspx>, Conversor a UTM.
- [15] <http://www.openlayers.org/api/OpenLayers.js>, OpenLayers API.

- [16] Trevejo Alonso, J.A., “Joomla! para principiantes”, *Aprendiendo a crear y mantener sitios web*, setembre de 2006.
- [17] http://docs.joomla.org/Developing_a_Model-View-Controller_Component_-_Part_1, Com fer components amb Joomla!
- [18] Neteler, Markus, “GRASS in a Nutshell”, *University of Minnesota*, juny de 2005.
- [19] <http://www.toopath.com>.
- [20] http://es.wikipedia.org/wiki/Base_de_datos_espacial, Bases de dades espacials.
- [21] <http://www.opengeospatial.org/>, Open Geospatial Consortium.
- [22] <http://www.opengeospatial.org/standards/wms>, WMS.
- [23] http://images.autodesk.com/adsk/files/acad_dxf.pdf, DXF Specifications.
- [24] http://wiki.openstreetmap.org/wiki/Main_Page, OSM Pàgina principal.
- [25] <http://www.openlayers.org/>, OpenLayers Pàgina principal.

Part VII

ANNEXES

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf8" />
<title>Trajectes entre Principals Edificis UAB</title>
<link rel="stylesheet" href="css/style.css" type="text/css" />
<script type="text/javascript" src="js/buscar.js"></script>
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>

<script src="js/toopathmapsosm.js"></script>
<script src="http://www.openstreetmap.org/openlayers/OpenStreetMap.js"></
script>
<script type="text/javascript">
var map, layerOSM, layerUAB, layerGeoRSS, url;
var bounds = new OpenLayers.Bounds
    (2.138214,41.524936,2.060966,41.48347634763476);

var lat=41.5021;
var lon=2.1024;
var zoom=14;

layerGeoRSS = new OpenLayers.Layer.GeoRSS("uabgeorss.xml", "uabgeorss.xml", {
    calculateInRange: function() { return true; }});
function init(){
    OpenLayers.ImgPath = "http://etse-73-192.uab.cat/js/img/";
    map = new OpenLayers.Map('map', { controls:
        [ new OpenLayers.Control.Navigation(),
          new OpenLayers.Control.PanZoomBar(),
          new OpenLayers.Control.LayerSwitcher(),
          new OpenLayers.Control.Attribution() ],
        maxResolution: 'auto', numZoomLevels: 19 });
    layerUAB = new OpenLayers.Layer.tooPath("UAB",
        "http://etse-73-192.uab.cat/uab/");
    layerOSM = new OpenLayers.Layer.OSM.Mapnik("Openstreetmap");
    map.addLayer(layerUAB);
    map.addLayer(layerOSM);
    map.addControl(new OpenLayers.Control.LayerSwitcher());
    var lonLat = new OpenLayers.LonLat(lon, lat).transform(new OpenLayers.
        Projection("EPSG:4326"), map.getProjectionObject());
```

```

    map.setCenter (lonLat , zoom);
}

function addUrl(x) {
    var newl = new OpenLayers.Layer.GML( "Ruta" , x,
    {
    format: OpenLayers.Format.KML,
    style: {strokeColor: "green", strokeWidth: 5, strokeOpacity: 0.5},
    projection: new OpenLayers.Projection("EPSG:4326")
    }
    );
    map.addLayer(newl);
    map.addLayer(layerGeoRSS);
    layerGeoRSS.setVisibility( false );
}

</script>
</head>

<body>
<h1 id="title">Trajectes Principals Edificis UAB</h1>

<div id="tags"></div>

<p id="shortdesc">
Cerca trajectes entre els principals edificis de la UAB.
</p>

<div class = "options">
    <form action = "" method = "GET" autocomplete="off">
        <table>
            <tr>
                <td class = "label">Origen</td>
            </tr>
            <tr>
                <td class = "input"><input type = "text" id = "origen" name = "
                    origen" size = "30" onkeyup="request_codi_origen(this.value)"></
                    input></td>
                <input type = "hidden" id = "idorigen" name = "idorigen" value = "1"
                    />
            </tr>
            <tr align="center">
                <td colspan="6" align="left">
                    <div id="tag_update_origen"></div>

```

```

        </td>
    </tr>

    <tr>
        <td class = "label">Desti</td>
    </tr>
    <tr>
        <td class = "input"><input type = "text" id = "desti" name = "desti"
            size = "30" onkeyup="request_codi_desti(this.value)"></input></
            td>
        <input type = "hidden" id = "iddesti" name = "iddesti" value = "1"
            />
    </tr>
    <tr align="center">
        <td colspan="6" align="left">
            <div id="tag_update_desti"></div>
        </td>
    </tr>

    <tr>
        <td class = "button"><input type = "submit" value = "Buscar" /><td
            >
    </tr>
    <input type = "hidden" name = "page" value = "<?=1?>" >
</table>
</form>
</div>
<?php
    include( 'content.php' );
    $page = $_GET[ 'page' ];
    include( get_content( $page ) );
?>

</body>
</html>

```

Listing 11: Portal de mobilitat

```

# This is the Apache server configuration file for providing OSM tile support
# through mod_tile

LoadModule tile_module /usr/lib/apache2/modules/mod_tile.so

<VirtualHost *:80>
    ServerName localhost
    DocumentRoot /home/toopath/www/

# You can either manually configure each tile set
#AddTileConfig /home/toopath/www/osm_tiles2/ Default

# or load all the tile sets defined in the configuration file into this
    virtual host
    LoadTileConfigFile /etc/renderd.conf

    LogLevel debug

</VirtualHost>

```

Listing 12: Configuració apache

```

[ Default ]
URI=/osm_tiles2/
XML=/home/toopath/mapnik/osm.xml
[ Aeroport ]
URI=/aeroport/
XML=/home/toopath/mapnik/osm_aeroport.xml
[ Rubi ]
URI=/rubi/
XML=/home/toopath/mapnik/osm_rubi.xml
[ UAB ]
URI=/uab/
XML=/home/toopath/mapnik/osm_uab.xml
[ Plataforma ]
URI=/plataforma/
XML=/home/toopath/mapnik/osm_plataforma.xml
[ TOPOUAB ]
URI=/topouab/
XML=/home/toopath/mapnik/osm_topouab.xml
[ PMI ]
URI=/pmi/
XML=/home/toopath/mapnik/osm_pmi.xml

```

Listing 13: Configuració renderd

Firmat: Domingo Manubens Gil
Bellaterra, Juny de 2009

Resum

L'explosió d'aplicacions a Internet basades en oferir serveis de localització, com són portals web de mobilitat o aplicacions de seguiment de vehicles en línia han motivat aquest projecte. *Google Maps* ens permet afegir mapes fàcilment en un lloc web amb la seva API, però *OpenLayers*, una llibreria JavaScript lliure ens dona l'opció de carregar cobertures de mapa i marcadors des de qualsevol font.

OpenStreetMap proporciona dades geogràfiques de manera lliure, com per exemple mapes de carrers i carreteres. Un estudi acurat de l'estructura i agrupació de les dades en el format OSM i el desenvolupament d'un servidor basat en el model de tesselles, són els principals elements de partida per crear la nostra pròpia font de dades cartogràfiques.

En aquest projecte s'analitza i processa DXF *Drawing eXchange Format* passant al format OSM. Un fitxer OSM conté la informació geogràfica necessària per a la base de dades espacial a partir de la qual, entre d'altres aplicacions, es podran visualitzar els mapes propis en una aplicació de seguiment de vehicles o en un portal web.

Resumen

La explosión de aplicaciones en Internet basadas en ofrecer servicios de localización, como pueden ser portales web de movilidad o aplicaciones de seguimiento de vehículos en línea han motivado este proyecto. *Google Maps* nos permite añadir mapas fácilmente en un sitio web con su API, pero *OpenLayers*, una librería JavaScript libre nos da la opción de cargar coberturas de mapa y marcadores desde cualquier fuente.

OpenStreetMap proporciona datos geográficos de forma libre, por ejemplo mapas de calles y carreteras. Un estudio detallado de la estructura y agrupación de los datos en el formato OSM y el desarrollo de un servidor basado en el modelo de teselas, són los principales elementos de partida per crear nusetra propia fuente de datos cartográficos.

En éste proyecto se analiza y procesa DXF *Drawing eXchange Format* pasando al formato OSM. Un fichero OSM contiene la información geográfica necesaria para la base de datos espacial a partir de la cual, entre otras aplicaciones, se podran visualizar los propios mapas en una aplicación de seguimiento de vehículos o en un portal web.

Abstract

The growing of Internet applications based on offering localizing services, like movility portals or on-line vehicle tracking applications are the main intention of this project. *Google Maps* help us to add maps in a web site trough its API, but *OpenLayers*, a free JavaScript library can load map layers and markers from any source.

OpenStreetMap share free geographical data, like street maps and roadways. Studying the structure and data groupment of OSM format and a tile-based model server development are the main starting points to create our own cartographical data source.

In this project DXF *Drawing eXchange Format* are analized and processed to obtain OSM files. An OSM file contains the geographical information necessary in a spatial data base from whom we can display our own maps in a tracking application or in an on-line portal.