

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)

Projecte final del Màster en Tecnologies de la Informació Geogràfica
11ena Edició

11 mtig 2009

Professionals per a la societat de la informació



Organitzador :



Amb el suport de :



Realització :
David Ramos González

Tutors :
Miquel Àngel Vargas Garcia
Jaume Figueras Jové
Jordi Valls Alsedà

Febrer 2010

"Los cortesanos, sin salir de sus aposentos ni de los umbrales de la corte se pasean por todo el mundo mirando un mapa, sin costarles blanca ni padecer calor ni frío, hambre ni sed y los caballeros audaces que, expuestos al sol, al frío, al aire, a las inclemencias del cielo, de noche y de día, de a pie y a caballo, medimos toda la tierra con nuestros mismos pies, y no solamente conocemos los enemigos pintados, sino en su mismo ser, y en todo trance y en toda ocasión los acometemos". Don Quijote

Resum

La cartografia, en aquests moments ha evolucionat al mateix nivell que han anat evolucionant les noves tecnologies. Amb el sorgiment de la tecnologia SIG, molts tipus de formats i programari han sorgit per fer cada dia una cartografia més rigorosa, completa i amb la major precisió possible.

Dintre de l'àmbit econòmic i global les diferents empreses i països han realitzat la seva pròpia cartografia amb més o menys encert, representant els diferents àmbits que podem trobar a la geografia del món. En aquests últims anys ha sorgit una nova paraula a l'hora de definir la forma de realitzar la cartografia, la **neogeografia**, el nou us que li estan donat a la cartografia un seguit de persones a internet.

Aquesta nova geografia intenta donar llibertat d'accés als usuaris per **georeferenciar** els llocs, geoetiquetació i crear un entorn on es puguin editar i crear els mapes a través de la integració de recursos en entorns webs mitjançant l'ús de API's i la utilització de aparells de posicionament GPS.

Una d'aquestes tecnologies **OpenSource** la trobem en **OpenStreetMap**, una iniciativa mundial que permet a tots els usuaris de tecnologia SIG l'oportunitat de crear mapes lliures, entenent per lliure el poder aportar els diferents punts de vista i expressió creativa els diferents llocs de la geografia mundial.

L'àrea de difusió i coordinació cartogràfica dintre del servei d'informació i estudis territorials, de la Mancomunitat de Municipis de l'Àrea Metropolitana de Barcelona estableix un conveni de col·laboració amb Logisim on volen aprofitar la tecnologia OSM i varen presentar tres objectius principals.

- El primer es l'estudi de la viabilitat i anàlisi de la migració de les dades cartogràfiques de la guia de carrers de l'AMB al format OSM.
- El segon es la implementació d'un servidor local OSM complet com a plataforma de proves per a la esmentada migració.
- El tercer i últim objectiu, comporta realitzar l'anàlisi dels requeriments, funcionalitat i implementació dels algorismes necessaris per a realitzar una migració de la cartografia de la guia de carrers de l'AMB al format OSM i una coexistència en sincronia d'ambdós formats.

Per realitzar aquest projecte on es migrarà la guia de carrers de l'AMB al format de OSM, s'han desglossat els diferents processos que s'han realitzat per transformar la cartografia al format de OSM, la integració d'un servidor amb el programari que conforma el projecte OSM, la creació de les bases de dades geoespacionals, el protocol d'accés a aquestes, els editors de mapes i dades, els sistemes de renderització i la possibilitat d'editar i actualitzar la cartografia per diferents entitats, usuaris experts i usuaris externs.

S'ha aconseguit realitzar la migració de la guia de carrers amb la seva totalitat al format OSM i la seva posterior migració al servidor de dades OSM instal·lat a l'AMB, on es poden realitzar edicions amb les eines que proporciona OSM i la posterior actualització de la base de dades creades.

OSM resulta una eina potent per a realitzar actualització de cartografia, però s'ha de tenir molt clar quins seran els usuaris finals d'aquesta eina, en el cas de que es lliuri la cartografia a la major part d'usuaris de OSM o si son els ajuntaments dels municipis els usuaris finals de l'aplicació web.

Paraules claus

OpenStreetMap, Base de dades geoespacial, Base de dades OSM, PostgreSQL, PostGIS, OSM, XML, XSLT, OSM, OpenSource, Potlatch, JOSM, API, Renderitzador, Mapnik, Rails Port, Ruby on Rails, TileMap, Osmosis, shp2pgsql, osm2pgsql, GIS, portal web, frontend, mod_tiles, Apache2.

Resumen

La cartografía, en estos momentos ha evolucionado al mismo nivel que han ido evolucionando las nuevas tecnologías, con el surgimiento de la tecnología SIG, muchos tipos de formatos y software han surgido para hacer cada día una cartografía más rigurosa, completa y con la mayor precisión posible.

Dentro del ámbito económico y global las distintas empresas y países han realizado su propia cartografía con más o menos acierto, representando los diferentes ámbitos que podemos encontrar en la geografía del mundo. En estos últimos años ha surgido una nueva palabra a la hora de definir la forma de realizar la cartografía, la **neogeografía**, el nuevo uso que le están dando a la cartografía una serie de personas a internet.

Esta nueva geografía intenta dar libertad de acceso a los usuarios para **georeferenciar** los lugares, geotiquetació y crear un entorno donde se puedan editar y crear los mapas a través de la integración de recursos en entornos web mediante el uso de API's y la utilización de aparatos de posicionamiento GPS.

Una de estas tecnologías **OpenSource** la encontramos en **OpenStreetMap**, una iniciativa mundial que permite a todos los usuarios de tecnología SIG la oportunidad de crear mapas libres, entendiéndose por libre el poder aportar con diferentes puntos de vista y expresión creativa los diferentes lugares de la geografía mundial.

El área de difusión y coordinación cartográfica dentro del servicio de información y estudios territoriales, de la Mancomunidad de Municipios del Área Metropolitana de Barcelona establece un convenio de colaboración con Logisim donde quieren aprovechar la tecnología OSM y presentaron tres objetivos principales.

- El primero es el estudio de la viabilidad y análisis de la migración de los datos cartográficos de la guía de calles de la AMB al formato OSM.
- El segundo es la implementación de un servidor local OSM completo como plataforma de pruebas para la mencionada migración.
- El tercer y último objetivo, conlleva realizar el análisis de los requerimientos, funcionalidad e implementación de los algoritmos necesarios para realizar una migración de la cartografía de la guía de calles de la AMB al formato OSM y una coexistencia en sincronía de ambos formatos.

Para realizar este proyecto donde se migrará el callejero de AMB al formato de OSM, se han desglosado los diferentes procesos que se han realizado para transformar la cartografía en formato de OSM, la integración de un servidor con el software que conforma el proyecto OSM, la creación de las bases de datos geospaciales, el protocolo de acceso a éstas, los editores de mapas y datos, los sistemas de renderización y la posibilidad de editar y actualizar la cartografía por diferentes entidades, usuarios expertos y usuarios externos.

Se ha conseguido realizar la migración de la guía de calles en su totalidad al formato OSM y su posterior migración al servidor de datos OSM instalados en AMB, donde se pueden realizar ediciones con las herramientas que proporciona OSM y la posterior actualización de la base de datos creada.

OSM resulta una herramienta potente para realizar actualización de cartografía, pero hay que tener muy claro quien serán los usuarios finales de esta herramienta, en el caso de que se entregue la cartografía en la mayor parte de usuarios de OSM o si son los ayuntamientos de los municipios los usuarios finales de la aplicación web.

Palabras claves

OpenStreetMap, Base de datos geoespacial, Base de datos OSM, PostgreSQL, PostGIS, OSM, XML, XSLT, OSM, OpenSource, Potlatch, JOSM, API, Renderizador, Mapnik, Rails Port, Ruby on Rails, TileMap, Osmosis, shp2pgsql, osm2pgsql, GIS, portal web, frontend, mod_tiles, Apache2.

Abstract

The cartography, in these moments has evolved to the same level that the new technologies have been evolving, with the emergence of the GIS technology, many types of formats and software have arisen to do every day a more rigorous, complete cartography and with the major possible precision.

Inside the economic and global area the different companies and countries have realized his own cartography with more or less success, representing the different areas that we can find in the geography of the world. In the latter years a new definition has arisen at the moment of defining the way of realizing the cartography, the **newgeography**, the new use that a series of persons are given him to the cartography to Internet.

This new geography tries to give freedom of access to the users for **georefencing** the places, geoetagging, and to create an environment where they could edit and create the maps across the integration of resources in environments web by means of API's use and the utilization of devices of positioning GPS.

One of these technologies we find **OpenSource** in **OpenStreetMap**, a world initiative that allows to all the users of technology SIG the opportunity to create free maps, dealing for free the power to express with different points of view and creative expression the different positions of the world geography.

The Area of diffusion and cartographic coordination inside the service of information and territorial studies, the Union of Municipalities of the Metropolitan Area of Barcelona * establishes an agreement of collaboration with Logisim where they want to take advantage of the technology OSM and they presented three principal goals.

- The first one is the study of the viability and analysis of the migration of the cartographic information of the guide of streets of AMB to the format OSM.
- The second one is the implementation of a local servant complete OSM as platform of tests for the mentioned migration.
- The third and last aim, he carries to realize the analysis of the requirements, functionality and implementation of the necessary algorithms to realize a migration of the cartography of the guide of streets of AMB to the format OSM and a coexistence in synchrony of both formats.

To realize this project where one will migrate the street directory of AMB to OSM's format, there have been removed the different processes that have been realized to transform the cartography into OSM's format, the integration of a servant with the software that shapes the project OSM, the creation of the geospatial databases, the protocol of access to these, the editors of maps and information, the systems of rendering and the possibility of editing and updating the cartography for different entities, expert users and external users.

One has managed to realize the migration of the guide of streets in its entirety to the format OSM and his later migration to the servant of information OSM installed in AMB, where editions can be realized by the tools that there contains OSM and the later update of the created database.

OSM turns out to be a powerful tool to realize update of cartography, but it is necessary to have very clearly who they will be the final users of this tool, in case the cartography is delivered in most of users of OSM or if they are the town halls of the municipalities the final users of the application web.

Keywords

OpenStreetMap, Base de datos geoespacial, Base de datos OSM, PostgreSQL, PostGIS, OSM, XML, XSLT, OSM, OpenSource, Potlatch, JOSM, API, Renderitzador, Mapnik, Rails Port, Ruby on Rails, TileMap, Osmosis, shp2pgsql, osm2pgsql, GIS, portal web, frontend, mod_tiles, Apache2.

Agraïments

Agraeixo i dedico aquest projecte final de Màster, a totes les persones que han estat involucrades d'una manera directa o indirecta i que sempre han tingut consciència de la etapa que he passat per la definició del mateix.

Primer de tot a Montserrat Monteagudo, per donar-me la oportunitat de formar part de l'equip que ha realitzat el projecte i també en definitiva per ajudar-me i motivar-me a l'Àrea Metropolitana de Barcelona en aquests primers anys de la meua vida professional com a Enginyer, conscient de la sort que tinc al tenir una persona que vetlla per la formació continua dels professionals que té al seu càrrec, un altre vegada moltes gràcies.

Per la mateixa banda, una menció especial per a Jordi Valls, que ha patit els meus ànims i desànims durant tot el transcurs del projecte i que en definitiva els hem patit a la vegada. Així com les persones que han estat donant ànims quan els explicava que portava entre mans, Pilar, Ricardo, Carmelo i Josep.

A tots els components de la 11a edició del Màster en Tecnologies de la informació Geogràfica, companys/es, amics/gues i professors/es que ens han fet augmentar els coneixements en el món dels SIG i les seves tecnologies.

No podia oblidar-me d'un dels pilars on m'he recolzat més, el meu tutor del projecte Miquel Angel Vargas, sempre disposat i que ha compartit totes les fases del projecte i reunions, fins i tot les que no sabien que estaven programades.

Als companys de projecte de Logisim Jaume Figueras, responsable del projecte que ha fet honor al seu nom, que ha estat dedicant-li tantes hores i en canvi no tenia inconvenient en explicar-me sempre com funcionava tot i per a que servia cada eina del sistema.

A Alejandro Perea que ha tret temps per estar amb mi per donar-me coneixements en el complex món de la programació en Java.

Finalment agrair també a la meua família i els meus amics, prometo tornar-vos el temps que no he estat amb vosaltres.

En definitiva, per la meua part i de tot cor...

MOLTES GRÀCIES !!!

1. Introducció	12
1.1. Origen del projecte	12
1.2. Marc institucional	13
1.3. Objecte d'estudi i estructura del treball	14
2. Objectius	16
2.1. Objectius generals	16
2.2. Objectius específics	16
2.3. Funcionalitats	17
3. Anàlisi de requeriments	18
3.1. Programari	18
3.1.1. Quantum GIS	19
3.1.2. PostgreSQL 8.3 + PostGIS	20
3.1.3. NetBeans IDE 6.7.1	21
3.1.4. Altres programes utilitzats	21
3.1.4.1. ArcGIS	21
3.1.4.2. Notepad ++	22
3.1.4.3. shp2pgsql	23
4. Fases del projecte	24
4.1. Aprenentatge d'eines OSM	24
4.1.1. Introducció a OSM	25
4.1.2. Potlatch	26
4.1.3. JOSM.	26
4.1.4. Wiki d'OSM	27
4.1.5. Map features	27
4.2. Catalogació	28
4.2.1. Obtenció, verificació, tractament i validació de dades	28
4.2.2. Creació de diccionari AMB-OSM	31
4.2.3. Estructura del format OSM	32
4.2.4. Importació de dades a POSTGRESQL + PostGIS	36
4.2.4.1. Creació de sentències .sql amb shp2pgsql	37
4.2.4.2. Importació amb fitxer .bat	38
4.3. Conversió de dades OSM	40
4.3.1. Extracció i tractament de dades geospacials de la BD en PostGIS	41
4.3.1.1. Definició de la connexió	41
4.3.1.2. Construcció de la sentència principal (taules a convertir)	42
4.3.1.3. Generació dels XML's i conversió a XML-OSM	42

4.3.2. Generació del XML	44
4.3.2.1. Justificació de la llibreria SAX en lloc de DOM	44
4.3.2.2. Imports i constants	45
4.3.2.3. Constructora	45
4.3.2.4. Creació del XML	47
4.3.3. Transformació XSL	51
4.3.3.1. Imports i atributs	51
4.3.3.2. Constructora	51
4.3.3.3. Funció de conversió	51
4.3.3.4. XSL's	52
4.3.3.4.1. Estructura bàsica dels XML's per geometries MultilineString i MultiPolygon	52
4.3.3.4.2. Estructura bàsica dels XML's per geometries tipus point	55
4.3.3.4.3. Estructura bàsica dels XML's per geometries tipus Tram	56
4.3.3.4.4. Estructura bàsica dels XML's per geometries tipus Portals	58
4.3.4. Utilitats de l'aplicació	58
4.3.4.1. Accedir a recursos dintre d'un fitxer JAR	58
4.3.4.2. Paràmetres i exemples d'execució	59
4.3.5. Renderització de mapes OSM	60
4.3.6. Verificació i validació	60
4.4. Regeneració de dades AMB	62
4.4.1. Edició de Mapes OSM sobre dades AMB	63
4.4.2. Exportació de dades OSM a POSTGIS	64
4.4.3. Sincronització	65
5. Instal·lació i configuració del PC dedicat a tasques OSM	66
5.1. Servidor OSM	67
5.2. Renderitzador OSM - Mapnik	68
5.3. Instal·lació del servidor a AMB	70
6. Presentació de resultats. Interfície Servidor web	72
6.1. Funcionalitat de les edicions	72
6.2. Renderització de Mapes AMB al servidor OSM	76
7. Conclusions	79
7.1. Valoració del treball	79
7.2. Valoració personal	80
8. Referències bibliogràfiques/web	81
9. Annexos	82
Annex I Diccionari AMB-OSM (Map Features específic)	83
Annex II Manual bàsic de Potlatch	91
Annex III Manual bàsic de JOSM	104

• Índex de figures

• <i>Figura 1.1. Esquema general de les fases del projecte</i>	15
• <i>Figura 3.1. Esquema del programari utilitzat</i>	18
• <i>Figura 3.2. Interfície gràfica de QGIS</i>	19
• <i>Figura 3.3. Interfície gràfica del SGBD PostgreSQL + PostGIS</i>	20
• <i>Figura 3.4. Interfície gràfica de NETBEANS IDE 6.7.1</i>	21
• <i>Figura 3.5. Interfície gràfica de Notepad ++</i>	22
• <i>Figura 3.6. Comandes de shp2pgsql</i>	23
• <i>Figura 4.1. Diagrama de components d'OSM</i>	24
• <i>Figura 4.2. Interfície gràfica de JOSM</i>	26
• <i>Figura 4.3. Pàgina web de map features d'OSM</i>	27
• <i>Figura 4.4. Guia de carrers MMAMB</i>	29
• <i>Figura 4.5. Mostra del diccionari AMB-OSM</i>	31
• <i>Figura 4.6. Fitxer .xml de la shape d'aeroports</i>	32
• <i>Figura 4.7. Fitxer .osm de la shape d'aeroports</i>	32
• <i>Figura 4.8. Dades primitives OSM</i>	33
• <i>Figura 4.9. Fitxer .osm de nodes de les gasolineres</i>	33
• <i>Figura 4.10. Fitxer .osm de vies de un tram de riu</i>	34
• <i>Figura 4.11. Fitxer .osm de vies tancades (edificació aïllada)</i>	35
• <i>Figura 4.12. Taules de geometria</i>	36
• <i>Figura 4.13. Opcions de la eina shp2pgsql</i>	37
• <i>Figura 4.14. Exemple de sentència de importació (Trams.shp)</i>	38
• <i>Figura 4.15. Extracte de l'executable .bat dels fitxers .shp</i>	38
• <i>Figura 4.16. Base de dades PostgreSQL amb</i>	39
• <i>Figura 4.17. Estructura de l'aplicació ambosm</i>	40
• <i>Figura 4.18. Model de Tiles o Tesel·les</i>	60
• <i>Figura 4.19. Trams de Badalona al servidor de proves</i>	61
• <i>Figura 4.20. Comanda de bulk_upload per pujar arxius .osm</i>	62
• <i>Figura 4.21. Script d'automatització per pujar els fitxers .osm</i>	62
• <i>Figura 4.22. Estructura de la sincronització</i>	63
• <i>Figura 4.23. Exemple de fitxer osmChange. Modificar</i>	64
• <i>Figura 4.24. Comanda de importació de la base de dades OSM a PostGIS</i>	64
• <i>Figura 4.25. Comanda de osmosis de verificació d'interval de temps</i>	65
• <i>Figura 4.26. Comanda de creació del fitxer .osm de canvis</i>	65
• <i>Figura 5.1. Estructura del servidor OSM a AMB</i>	66
• <i>Figura 5.2. Finestra de interfície de connexió remota (WinVNC)</i>	70
• <i>Figura 5.3. Interfície de Ubuntu 8.04TLS Server</i>	71
• <i>Figura 6.1. Estils i simbologia en el servidor AMB-OSM</i>	72
• <i>Figura 6.2. Edició maquetada en paper de un mapa OSM</i>	73
• <i>Figura 6.3. Historial d'edicions en el servidor AMB-OSM</i>	74
• <i>Figura 6.4. Historial d'edicions específic en el servidor AMB-OSM</i>	74
• <i>Figura 6.5. Interpolació de Números policia</i>	75
• <i>Figura 6.6. Interfície web del servidor AMB-OSM</i>	76
• <i>Figura 6.7. Vista general dels trams complets de l'AMB</i>	76
• <i>Figura 6.8. Trams de carrers de Barcelona amb simbologia puntual</i>	77
• <i>Figura 6.9. Representació de camins i corriols</i>	77
• <i>Figura 6.10. Jerarquització de les xarxes de comunicació</i>	78

1. Introducció

1.1. Origen del projecte

Al concloure la part teòrica del Màster en Tecnologies de la Informació Geogràfica, 11a edició, organitzat pel departament de geografia de la Universitat Autònoma de Barcelona, teníem la oportunitat de escollir diferents projectes proposats per diferents empreses que englobessin els diferents mòduls que realitzem durant el postgrau, però personalment em van donar la opció de realitzar el projecte al meu lloc de feina actual, a la Mancomunitat de Municipis de l'Àrea Metropolitana de Barcelona(MMAMB) a l'àrea de difusió i coordinació cartogràfica dintre del servei d'informació i estudis territorials.

Des del departament d'Ordenació Urbanística, es va contactar amb Logisim, equip d'investigació de la universitat Politècnica de Catalunya que col·labora amb la Facultat d'informàtica de Barcelona per a crear un conveni on es realitzaria la migració de la Guia de carrers de l'Àrea metropolitana de Barcelona al sistema de informació geogràfica lliure d'OpenStreetMap.

Una vegada acordats els objectius i las funcionalitats que es realitzaran, al Juliol de 2009 es signa un conveni per iniciar el projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el mapa de Carrers d'OpenstreetMap. Al mateix temps es forma l'equip de quatre tècnics que desenvoluparan el projecte en la seva totalitat.

- Coordinador del Projecte : Jaume Figueras - Enginyer de recerca
- Coordinador del Projecte : Antoni Guasch - Doctor Enginyer Industrial
- Suport de Cartografia i estudiant de Postgrau : David Ramos - Enginyer tècnic en topografia
- Suport de Programació : Alejandro Perea - Estudiant i Becari del LCFIB

Des de l'Àrea metropolitana de Barcelona, es designen dos tutors de seguiment :

- Jordi Valls : Enginyer Industrial
- Montserrat Monteagudo : Enginyera tècnica en topografia
Cap de l'Àrea de Difusió i Coordinació Cartogràfica
Servei d'Informació i Estudis Territorials

Des de el departament de Geografia de la Universitat autònoma s'estableix un tutor per realitzar el seguiment del desenvolupament del projecte i el seu recolzament tècnic, englobat en la onzena edició del Màster en Tecnologies de la Informació Geogràfica.

- Miquel Angel Vargas : Llicenciat en Geografia
Laboratori d'Informació Geogràfica i Teledetecció(LIGIT)

1.2. Marc institucional

Cal esmentar abans de continuar, les dues principals organitzacions que han realitzat el projecte i quina es la seva situació actual en el marc empresarial i les seves tasques principals.

LOGISIM

Logisim neix de la voluntat dels equips d'investigació en Simulació del Laboratori de Càlcul de la FIB (UPC), del departament de Telecomunicacions i Enginyeria de Sistemes (UAB) i del Institut de Robòtica i Informàtica Industrial (UPC/CSIC) per treballar junts i fer una oferta conjunta al mercat. Aquest grup va rebre un fort recolzament per part del CIDEM en valorar que Logisim es podia integrar dins de la Xarxa de Centres d'Innovació Tecnològica (Xarxa IT). En aquests darrers anys Logisim ha consolidat la seva imatge de marca i ha esdevingut un centre d'excel·lència per la qualitat dels seus resultats.

Els seus principals objectius son :

- Aplicar noves tècniques de modelatge, simulació i optimització per millorar el rendiment de sistemes logístics, de producció, de transport i de servei tant del món industrial com a les administracions.
- Minimitzar el risc en la presa de decisions mitjançant l'anàlisi i avaluació experimental de les diferents alternatives abans de la seva implantació física sobre el sistema real.
- Disseny d'entorns d'ajuda a la presa de decisions basats en eines integrades de modelatge, simulació i optimització.
- Anàlisi i millora de plans d'evacuació.

Area Metropolitana de Barcelona (Servei d'Informació i Estudis Territorials)

Al servei d'Estudis Territorials utilitzen instruments per al coneixement i anàlisi del territori metropolità que faciliten la presa de decisions i la formulació de propostes d'actuació en l'àmbit supramunicipal o local i se sustenten en les bases alfanumèriques i cartogràfiques, suport essencial per a aquests treballs.

Els estudis de nivell metropolità se centren en els principals fenòmens que incideixen en la configuració i en les transformacions de l'espai metropolità i abasten aspectes com la identificació dels teixits residencials i l'evolució del mercat de l'habitatge, el seguiment de l'oferta industrial, el creixement demogràfic, l'impacte de la immigració estrangera, l'evolució del mercat de treball i les característiques socioeconòmiques de la població.

El territori es pot analitzar des de diferents enfocaments que són desenvolupats per les diverses institucions que hi treballen.

Les principals tasques que es realitzen son :

- Realització de la Cartografia Digital - 1:1000
 - Fotogrametria aèria
- Planejament Urbanístic municipal
- Realització de la Guia de Carrers de l'AMB
- Creació dels indicadors metropolitans

1.3. Objecte d'estudi i estructura del treball

El projecte amb el que treballem estaria orientat cap als projectes de programació SIG on es desenvoluparà una aplicació web que faran servir usuaris no experts i experts en SIG i cartografia.

El projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el mapa de Carrers d'OpenstreetMap(OSM), per una banda revisa la cartografia de la que disposem i la volem transformar o utilitzar per a realitzar-ne una de nova o millorar la que hi ha actualment, també desitgem conèixer i aprendre la nova tecnologia lliure OpenSource que es OSM i aprofitar els seus recursos per obtenir una informació de més qualitat i estructurada.

El que es vol aconseguir des de AMB es la total sincronització entre els dos tipus de cartografia mitjançant unes aplicacions que desenvolupem per transformar-les al format OSM i que els usuaris finals tot i que no siguin experts en tecnologia SIG, es puguin aprofitar de les eines que tenim i creem per a actualitzar i realitzar nova cartografia.

Al acabar el projecte , es donaria un ús municipal, on cadascun dels 36 municipis pugues accedir al servidor de dades OSM instal·lat per editar les dades en el moment que mes desitgessin, així com des de l'AMB controlar les edicions que es realitzen.

L'estructura del projecte s'ha dividit en quatre parts, abans de començar en cada fase del treball, hem dedicat un temps a aprendre com s'estructura els components d'OSM i les eines que utilitza per a obtenir una idea general sobre el que ens trobarem a les diferents fases del projecte.

La primera part comença amb un procés **d'anàlisi i catalogació** de les dades que disposem per a transformar al format OSM i adequar les dades que tenim per a realitzar la catalogació de una manera ordenada i eficient. Trobem desglossat aquesta part en el **punt 4.2** de l'índex del treball

El segon pas ha estat la **conversió de les dades** amb la realització d'una aplicació que es pugui utilitzar per transformar les capes que tenim de la guia de carrers i les posteriors proves de renderització en un servidor de proves creat expressament a la UPC, per realitzar els testos. Per mes informació podem consultar el **punt 4.3** de l'índex del treball.

A la tercera part realitzem la **regeneració de dades** carregades al servidor OSM comprovant la **sincronització** que existeix entre les dues bases de dades mitjançant osmosis i osm2pgsql i la posterior actualització al servidor geospacial PostGIS. Per a més explicacions busquem els **punts 4.4** de l'índex.

La quarta i última part compren l'**instal·lació** final del servidor de dades OSM a l'AMB, amb tots els components i càrrega de la base de dades Postgis i OSM. En el **punt 5** de l'índex trobarem tot el referent a la quarta part del projecte.

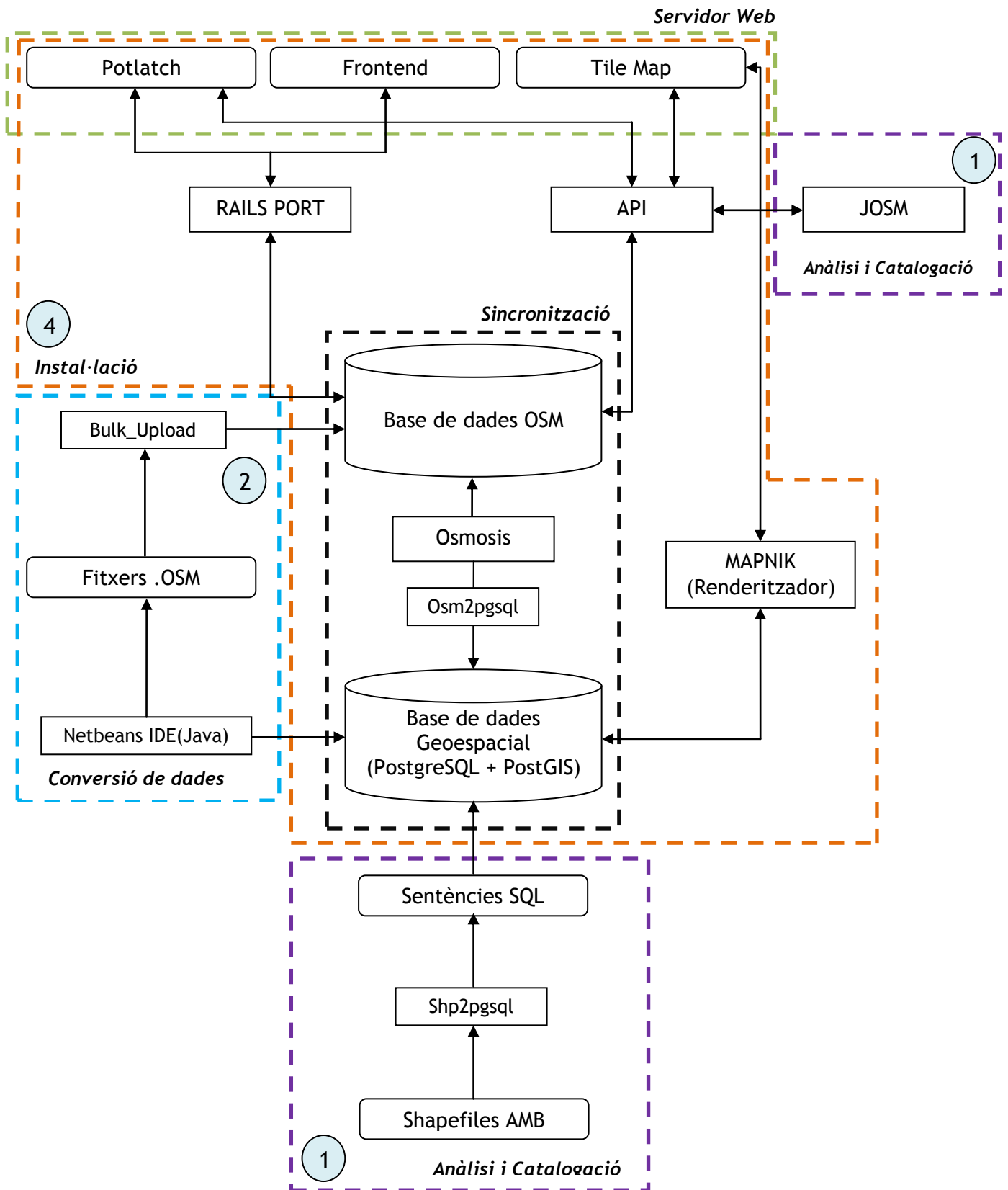


Fig 1.1. Esquema general de les fases del projecte.

2. Objectius

2.1. Objectius generals

L'objectiu del projecte consisteix en :

- 1) Estudi de viabilitat i anàlisi de la migració de dades cartogràfiques de la guia de carrers de l'AMB al format OpenStreetMap(OSM).
- 2) Implementació d'un servidor local OSM complet com a plataforma de proves per a la migració.
- 3) Anàlisi dels requeriments, funcionalitat i implementació dels algoritmes necessaris per a realitzar una migració de la cartografia de la guia de carrers de l'AMB al format OSM i una coexistència en sincronia d'ambdós formats.

2.2. Objectius específics

Dintre dels objectius generals, ens marquem uns de específics per concretar la feina a fer durant tot el projecte.

- 1) Estudi de viabilitat i anàlisi de la migració de dades cartogràfiques de la guia de carrers de l'AMB al format OpenStreetMap(OSM).
 - Especificació i disseny d'un procediment de sincronització entre la base de dades de la guia de carrers de l'AMB i la base de dades OSM.
 - Realització d'un catàleg d'equivalències entre ambdós formats.
 - Estudi de les diferents capes d'informació i selecció de les que seran objecte de la migració.
- 2) Implementació d'un servidor local OSM complet com a plataforma de proves per a la migració.
 - L'avaluació i prova de la integració de la cartografia de la Guia de Carrers de l'AMB amb la cartografia OSM.
 - La derivació de la cartografia de l'AMB amb independència de la cartografia publicada en el servidor de OSM principal.
 - Avaluar els mecanismes d'actualització de la base de dades local.
- 3) Anàlisi dels requeriments, funcionalitat i implementació dels algoritmes necessaris per a realitzar una migració de la cartografia de la guia de carrers de l'AMB al format OSM i una coexistència en sincronia d'ambdós formats.
 - Implementació dels procediments especificats i dissenyats en el punt 1.
 - Avaluació del editor en línia Potlatch i fora de línia JOSM.
 - Test i verificació del funcionament del sistema.

2.3. Funcionalitats

Una vegada acomplert els objectius, el servidor de dades OSM-AMB, ens permetrà :

- Aplicar una estandardització de la base cartogràfica per a tots els municipis
- Permetrà una possible edició distribuïda de la guia de carrers per parts dels Ajuntaments
- Adequació i canalització de la informació que poden aportar els usuaris
- Detecció de errors i incoherències per millorar la guia de carrers
- Aplicació de diferents projectes OSM

3. Anàlisi de requeriments

3.1. Programari

El principal programari que utilitzem per l'anàlisi de dades i realització de les aplicacions son bàsicament cinc : *Quantum GIS*, *PostgreSQL + Postgis*, *Potlatch*, *JOSM* i *Netbeans*.

En l'apartat numero 4.1.1 s'expliquen el programari OSM més específicament(Potlatch i JOSM).

S'han utilitzat altres, però d'una manera més secundària i per recolzar a l'hora de compilar la informació i les dades obtingudes.

A continuació comentem el programari utilitzat, explicant la seva funció en el marc del nostre projecte.



Fig 3.1. Esquema del programari utilitzat.

3.1.1. Quantum GIS

Sistema de informació geogràfic de codi lliure per a plataformes Linux, Unix, Mac OS i Windows que permet utilitzar formats ràsters i vectorials, així com bases i geobases de dades.

Comprèn un suport per a la extensió espacial PostgreSQL + Postgis, que permet connectar a la base de dades creades i visualitzar les taules creades.

Arxius que permet utilitzar :

- Vectorial : Shapefile, cobertures d'ArcInfo, Mapinfo, GRASS, etc..
- Ràster : TIFF, GeoTIFF, JPG,etc...
- Capes WMS I WFS

S'ha emprat principalment per visualitzar les taules creades a PostGIS i la creació del diccionari AMB-OSM. Conté una eina que transforma i importa fitxers shape a PostgreSQL.

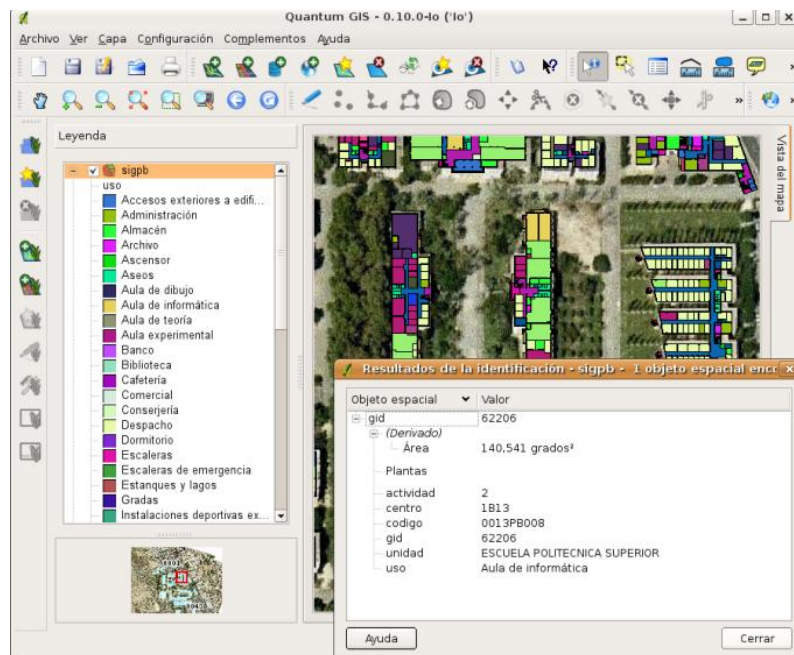


Fig 3.2. Interficie gràfica de QGIS.

3.1.2. PostgreSQL 8.3 + PostGIS

PostgreSQL és un programari lliure que implementa un sistema de gestió de bases de dades objecte relacional, distribuït sota llicència BSD.

Com molts altres projectes OpenSource, el desenvolupament de PostgreSQL no està manejat per una sola companyia sinó que està dirigit per una comunitat de desenvolupadors i organitzacions comercials les quals treballen en el seu desenvolupament. Aquesta comunitat és denominada el PGDG (PostgreSQL Global Development Group).

Amb la combinació del PostGIS afegeix suport d'objectes geogràfics a la base de dades objecte-relacional PostgreSQL, convertint-la en una base de dades espacial per la seva utilització en Sistemes d'Informació Geogràfica. Es va crear com un projecte de recerca de tecnologies de bases de dades espacials sota llicència Pública General de GNU.

Postgis permet l'ús d'objectes GIS que inclou un suport per índexs GiST basats en R-Tree i funcions bàsiques per a l'anàlisi d'objectes GIS.

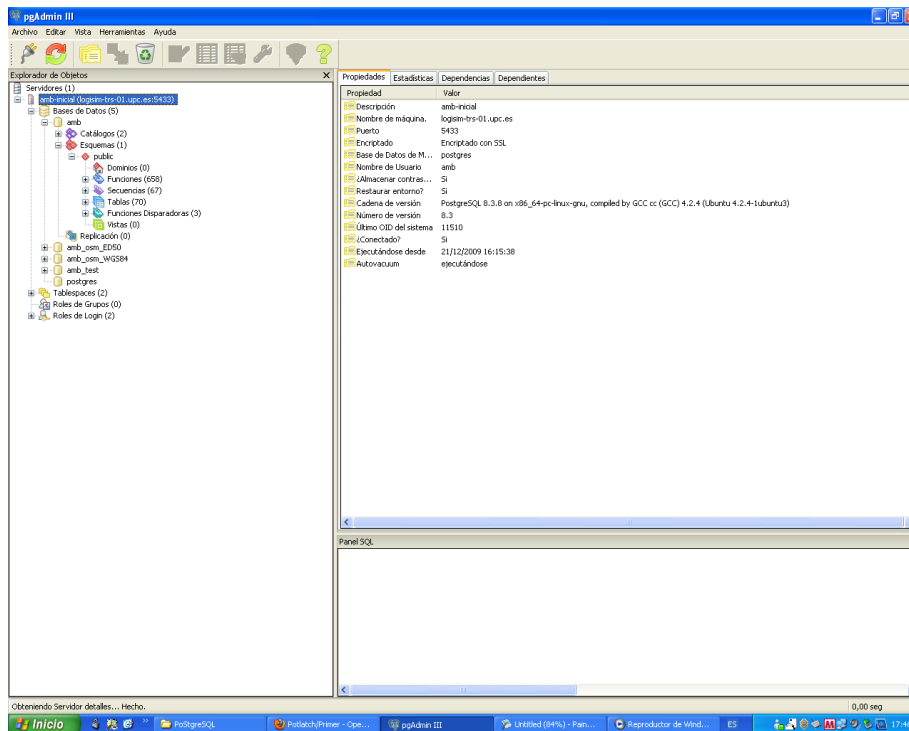


Fig 3.3. Interficie gràfica del SGBD PostgreSQL + PostGIS.

S'ha emprat per crear la base de dades geospacial que inclourem a la estructura de components OSM i ens servirà per actualitzar-la i sincronitzar amb la base de dades OSM.

3.1.3. NetBeans IDE 6.7.1

L'aplicació s'ha creat amb una eina de programari lliure java anomenada Netbeans que agafa diferents mòduls per aconseguir crear els objectes, classes i funcions que ens permetrà transformar les taules carregades a la BD geoespacial de PostGIS al format XML i a la vegada traduir-ho amb les plantilles XSLT, amb el que aconseguirem transformar en fitxers .osm.

Netbeans IDE (Integrated Development Environment), es una eina per a programadors pensada per escriure, compilar, depurar i executar programes, està escrit en Java però serveix per qualsevol altre llenguatge de programació.

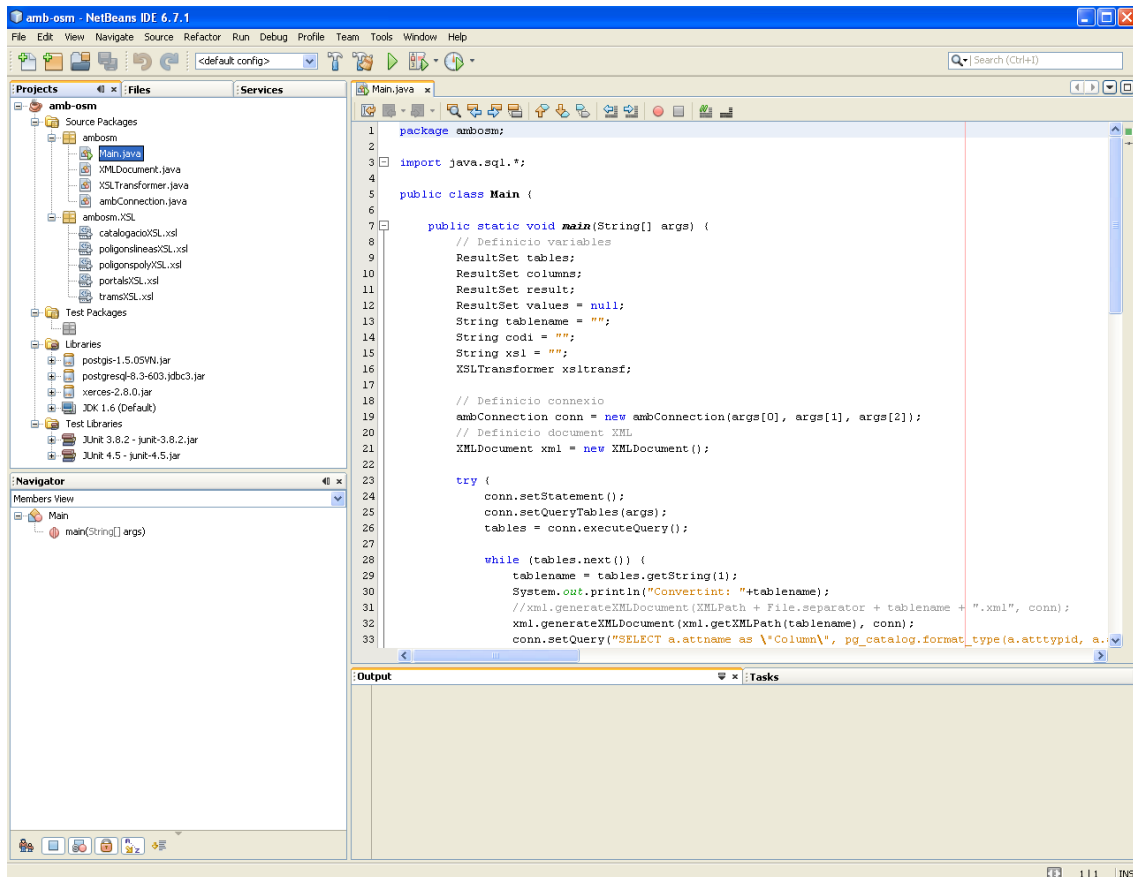


Fig 3.4. Interfície gràfica de NETBEANS IDE 6.7.1.

Es poden afegir llibreries per crear classes i objectes nous, que ens permeten llegir i escriure al SGBD que utilitzem, PostgreSQL + POSTGIS.

3.1.4. Altres programes utilitzats

3.1.4.1. ArcGIS

ArcGIS és el nom d'un conjunt de productes de programari en el camp dels Sistemes d'Informació Geogràfica o SIG. Produït i comercialitzat per ESRI, sota el nom genèric ArcGIS s'agrupen diverses aplicacions per a la captura, edició, anàlisi, tractament, disseny, publicació i impressió d'informació geogràfica.

ArcGIS Desktop, la família d'aplicacions SIG d'escriptori, és una de les més àmpliament utilitzades, incloent en les seves últimes edicions les eines ArcReader, ArcMap, ArcCatalog, ArcToolbox, ArcScene i ArcGlobe, a més de diverses extensions. ArcGIS Desktop es distribueix comercialment sota tres nivells de llicències que són, en ordre creixent de funcionalitats (i cost): ArcView, ArcEditor i ArcInfo.

Hem utilitzat les eines ArcCatalog i Arcmap per a l'anàlisi de les dades i per realitzar unions espacials dels diferents fitxers shapes que conformen la guia de carrers de l'AMB, tanmateix ens ha ajudat a comprovar les correctes reprojeccions dels datums ED50 a WGS84.

3.1.4.2. Notepad ++

Editor de text i de codi font lliure, amb suport per a variis llenguatges de programació en diferents plataformes o sistemes operatius. És un editor molt ràpid i permet una navegació mitjançant pestanyes (veure Fig 3.5) que li dona molta més agilitat de lectura de fitxers que altres editors de text.

L'hem utilitzat per :

- Creació dels fitxers .bat per a la importació de dades a PostgreSQL.
- Edició d'etiquetes en fitxers OSM, XML,XSL i java.

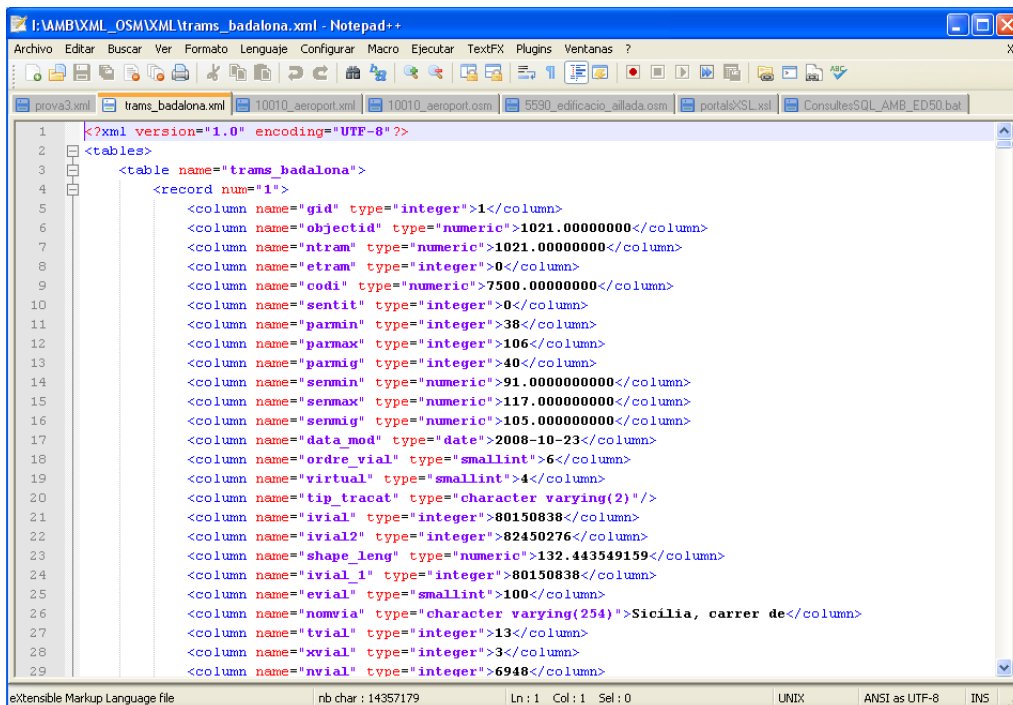


Fig 3.5. Interfície gràfica de Notepad ++.

3.1.4.3. shp2pgsql

Eina que permet importar fitxers .shp a la base de dades geoespacial PostGIS, creant una sentència SQL amb la geometria per poder tractar dades geoespacionals, s'executa des de la consola de comandes del sistema operatiu (veure Fig 3.6) i conté diferents opcions per importar les fitxers .shp.

L'utilitzem per crear les sentències .sql de els fitxers .shp.

```

C:\WINDOWS\system32\CMD.exe
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin>shp2pgsql.exe
RCSID: $id: shp2pgsql.c 4204 2009-06-24 11:18:29Z mcayland $ RELEASE: 1.4 USE_GE
OS=1 USE_PROJ=1 USE_STATS=1
USAGE: shp2pgsql.exe [<options>] <shapefile> [<schema>.]<table>
OPTIONS:
  -s <sruid> Set the SRID field. If not specified it defaults to -1.
  (-dialect) These are mutually exclusive options:
    -d Drops the table, then recreates it and populates
        it with current shape file data.
    -a Appends shape file into current table, must be
        exactly the same table schema.
    -c Creates a new table and populates it, this is the
        default if you do not specify any options.
    -p Prepare mode, only creates the table.
  -g <geometry_column> Specify the name of the geometry column
    (mostly useful in append mode).
  -D Use postgresql dump format (defaults to sql insert statements.
  -k Keep postgresql identifiers case.
  -i Use int4 type for all integer dbf fields.
  -I Create a GiST index on the geometry column.
  -S Generate simple geometries instead of MULTI geometries.
  -w Use wkt format (For postgis-0.x support - drops M - drifts coordinates).
  -W <encoding> Specify the character encoding of Shape's
    attribute column. (default : "ASCII")
  -N <policy> Specify NULL geometries handling policy (insert,skip,abort)
  -n Only import DBF file.
  -? Display this help screen

D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin>shp2pgsql E:\A
MB\Guia_Carrers_LOGISIM_AMB\trans.shp trans >trans.sql
Shapefile type: Arc
Postgis type: MULTILINESTRING[2]

D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin>
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin>

```

Fig 3.6. Comandes de shp2pgsql.

4. Fases del projecte

4.1. Aprenentatge d'eines OSM

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 penseu que són lliures no ho són, de fet, tenen restriccions tècniques i/o legals que impedeixen als usuaris el seu ús de forma creativa, productiva o sorprenent en noves aplicacions.

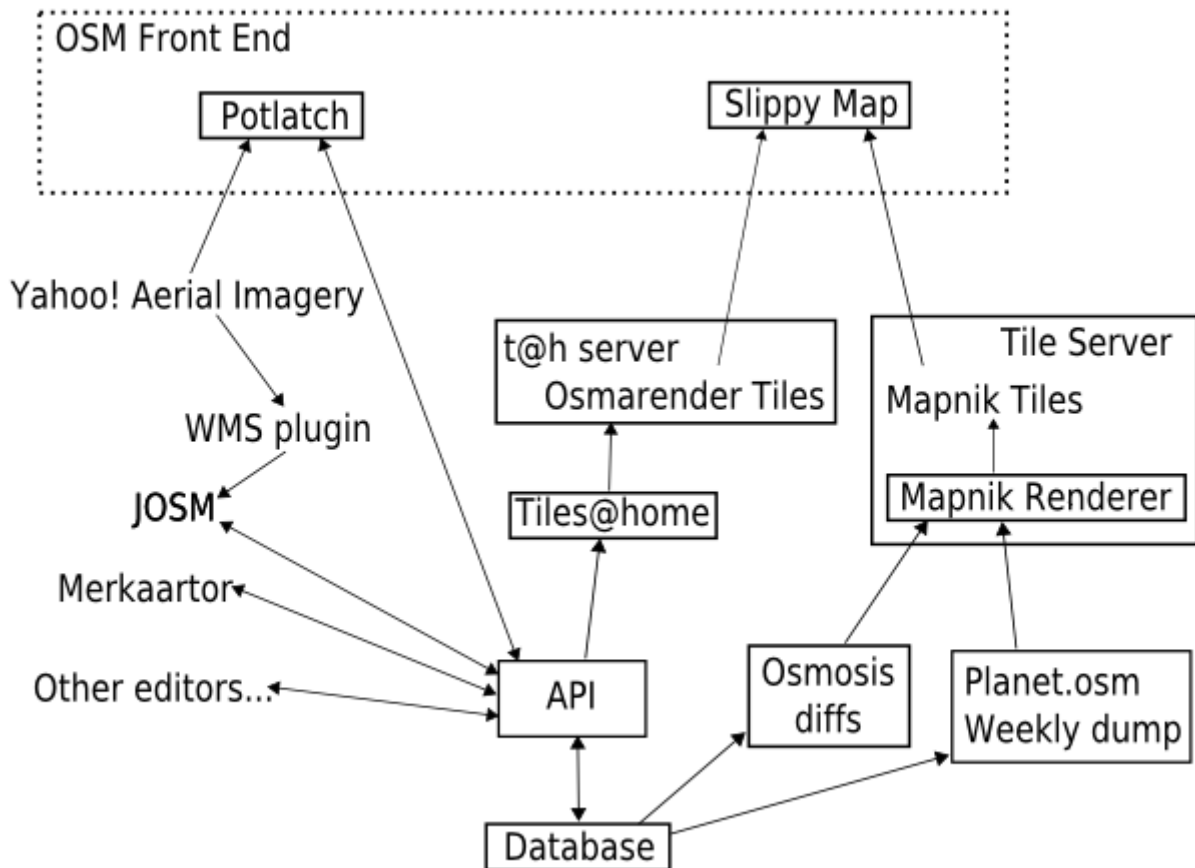


Fig 4.1. Diagrama de components d'OSM.

Els principals components de OSM (veure Fig 4.1) són :

- **Database** : La base de dades conté tota la informació del mapa en forma de nodes, camins i relacions en format .osm, 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.
- **API** : El protocol OSM, són el conjunt de crides web mitjançant un protocol SOA-REST, que ens permet accedir i escriure a la base de dades de OSM. Altres components poden accedir a través d'aquesta interfície.

- **Osmosis** : Programa que permet fer tractament de dades a partir de la base de dades OSM.
- **Planet.osm** : Repositori de les línies, punts i relacions de tot el món en format .osm, es poden descarregar porcions del lloc que desitgis per a poder editar, s'actualitzen cada setmana i actualment ocupa uns 170 Gb d'espai.
- **Editors externs** : JOSM, Merkaartor, editors offline que permeten modificar els fitxers .osm de la base de dades, poden tenir accés directe a la base de dades geoespacial per poder utilitzar qualsevol GIS capaç de treballar amb SGBD'S.
- **Renderitzadors** : El principal és Mapnik Tiles i Osmarender Tiles , amb la informació de la base de dades generen mapes de bits renderitzats, creant un conjunt d'imatges o tesel·les que conformen el dibuix del mapa. Guardem les tesel·les creades a un servidor específic.
- **Slippy map** : Conjunt d'imatges renderitzades que mostra la interfície AJAX i permet realitzar les diferents eines de visualització, desplaçar-se i ampliar la vista del mapa. Es mostra una finestra en el lateral dret amb un "+" i permet seleccionar els diferents tipus de renderització que disposem (Mapnik, Osmarender, cycle map,etc...)
- **Potlatch** : Editor web online de la base de dades.

4.1.1. Introducció a OSM

Una de les característiques d'OpenStreetMap és que no utilitza les primitives geomètriques habituals(punts, línies i polígons), si no un model de dades orientades a topologia.

La primitiva més senzilla que podem trobar és el *node*, es defineix únicament amb un parell de coordenades i un identificador únic, fent servir la representació XML comú a quasi tot el programari de OSM.

Varis *nodes*, formen una *way* (o camí), donats els identificadors únics dels nodes. Una *way* es un conjunt ordenat de nodes, una primitiva similar a una *linestring*.

En el model de dades OSM no hi ha una diferència explícita entre línies tancades i polígons, la diferència ha de ser semàntica. Per exemple, si tenim una *way* tancada, amb una etiqueta de *landuse:farmland*, vol dir que el polígon definit per aquesta *way* es una granja. Si, en canvi, tenim una *way* tancada amb una etiqueta de *barrier:fence*, vol dir que hi ha una superfície dedicada a granja i que el seu perímetre es un filat. Fixem-nos que en aquest últim exemple no fa falta definir per separat el polígon de la superfície de la granja i la línia pel filat del perímetre.

Al poder dotar el sentit semàntic, tant als nodes com a les *ways*, es poden relacionar fàcilment les unes amb les altres.

Per últim existeixen les relacions, que permeten enllaçar entre si *nodes*, *ways* i altres relacions. S'utilitzen per definir *multipolígons*(relacionant uns *ways* tancats entre si) i rutes (línies d'autobús, metro, rutes a peu, etc...).

4.1.2. Potlatch

És un dels editors de la informació espacial que utilitza OSM, és el més accessible i està basat en tecnologia Flash, que s'executa directament al navegador web. Només tenim que ampliar la zona a editar, i prémer a sobre de la pestanya edit en la part superior de la pàgina.

Potlatch està dissenyat per a ésser fàcil d'utilitzar per a que els seus usuaris no es tinguin que preocupar en excés de conèixer com funciona la topologia o el model de dades OSM. Per ajudar a l'etiquetatge, mostra una sèrie d'opcions que generen el conjunt d'etiquetes més comunament utilitzats.

4.1.3. JOSM

Potlatch és suficient per a usuaris no experts en temàtica SIG, per a operacions més grans es queda curt. El segon editor mes emprat es JOSM, acrònim de "Java OpenStreetMap editor", té moltes més funcions que potlatch i està destinat a usuaris amb més coneixement d'eines SIG.

Permet un control més fi sobre les primitives geomètriques i pot treballar amb una gran quantitat de dades al mateix moment, així com Potlatch té l'opció d'anar actualitzant les correccions que realitzem, a JOSM (veure Fig 4.2) és l'usuari el que té que indicar quan envia els canvis realitzats.

Suporta diferents projeccions, permet visualitzar dades WMS per facilitar la feina d'actualització. També controla col·lisions entre versions de dades i valida diversos errors topològics i semàntics.

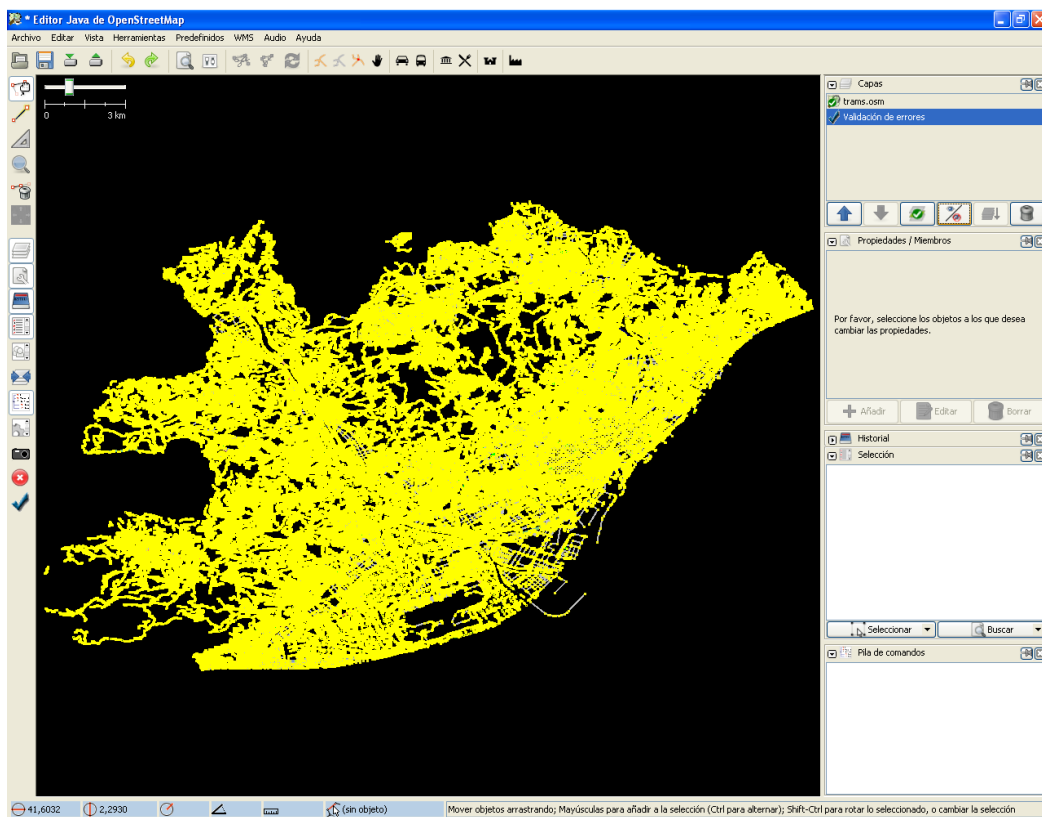


Fig 4.2. Interfície gràfica de JOSM.

4.1.4. Wiki d'OSM

OSM disposa d'una pagina web que et mostra tot el necessari per a crear els nostres propis mapes. Una vegada registrats a la wiki, podem accedir a les explicacions de les eines que necessitem per a crear els nostres mapes.

4.1.5. Map features

Map Features (veure Fig 4.3) és un diccionari de claus i valors que li donem als elements de OSM, totes aquestes claus i valors estan definits per a que es puguin crear els mapes amb les mateixes característiques.

Així doncs podem trobar els diferents elements físics que representem en un mapa amb la seva clau (*Key*) i el seu valor (*Values*) corresponent i amb una petita representació de com es simbolitzarà una vegada renderitzat amb Mapnik per a fer-nos una idea de com es visualitzarà.

No és una llista completa de tot el que es pot posar, hi ha un apartat que s'anomena *proposed features*, on es poden proposar nous valors o modificar els actuals.

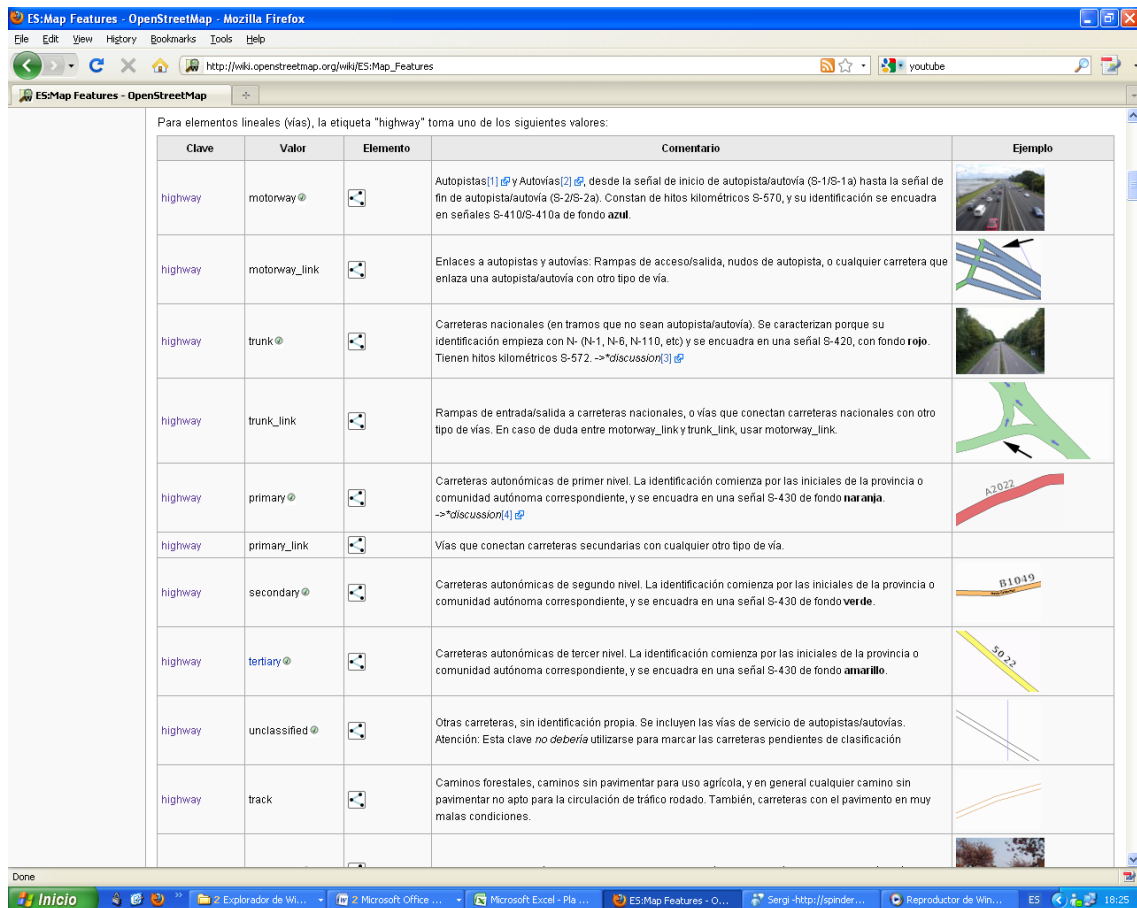


Fig 4.3. Pàgina web de map features d'OSM.

4.2. Catalogació

L'objectiu de la fase de catalogació, és obtenir una base de dades geoespacial de la cartografia que utilitza la Mancomunitat de Municipis de l'Àrea Metropolitana de Barcelona per realitzar la Guia de carrers dels 36 municipis que componen l'AMB.

Amb la finalitat d'adequar-la al format que utilitza OSM, tindrem que tenir en compte el tipus de fitxer geogràfic que utilitza i la seva estructura de les dades per poder obtenir la geobase de dades més adequada i coherent als propòsits del projecte.

4.2.1. Obtenció, verificació, tractament i validació de dades.

El format ESRI Shapefile (SHP) és un format d'arxiu informàtic propietari obert de dades espacials desenvolupat per la companyia ESRI, que crea i comercialitza software per a Sistemes d'Informació Geogràfica com Arc / Info o ArcGIS.

Originalment es va crear per la utilització amb el seu producte ArcView GIS, però actualment s'ha convertit en format estàndard de facto per a l'intercanvi d'informació geogràfica entre Sistemes d'Informació Geogràfica per la importància que els productes ESRI tenen en el mercat SIG i per estar molt bé documentat.

Un Shapefile és un format vectorial d'emmagatzematge digital on es guarda la localització dels elements geogràfics i els atributs associats a ells. El format no té capacitat per emmagatzemar informació topològica.

Es compon per diversos arxius. El nombre mínim requerit és de tres i tenen les extensions següents:

- **.shp** : és l'arxiu que emmagatzema les entitats geomètriques dels objectes.
- **.shx** : és l'arxiu que emmagatzema l'índex de les entitats geomètriques.
- **.dbf** : el dBASE, o base de dades, és l'arxiu que emmagatzema la informació dels atributs dels objectes.

A més d'aquests tres fitxers requerits, opcionalment es poden utilitzar altres per millorar el funcionament en les operacions de consulta a la base de dades, informació sobre la projecció cartogràfica, o emmagatzematge de metadades. Aquests arxius són:

- **.prj** : és el fitxer que guarda la informació referida a la projecció i el sistema de coordenades
- **.shp.xml** : emmagatzema les metadades del shapefile.
- **.sbx** : índex temàtics (Alfanumèrics).

AMB ens va lliurar la cartografia (veure Fig 4.4) dels 36 municipis que utilitza per realitzar la guia de carrers en paper, a format 1:10000 en format .shp.

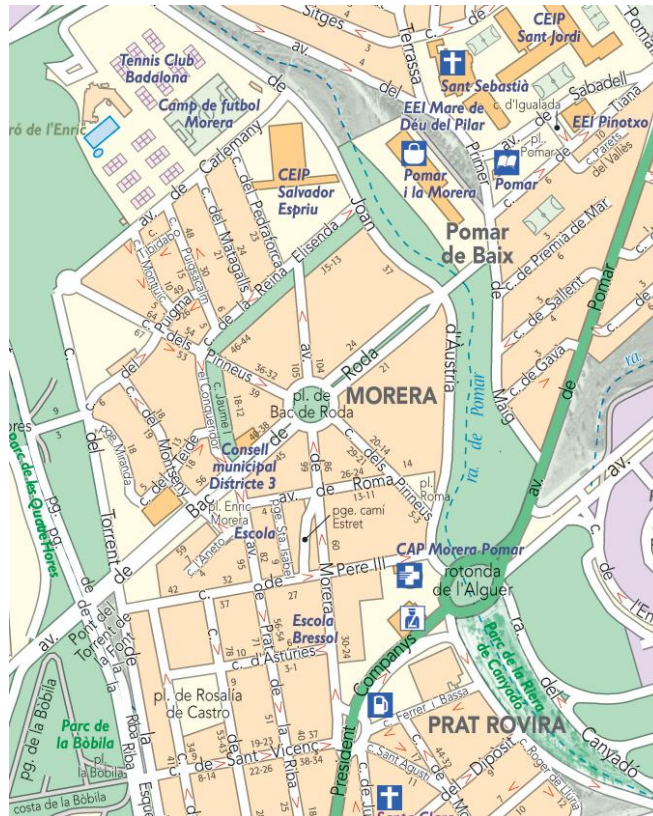


Fig 4.4. Guia de carrers MMAMB.

Cadascun dels .shp estan georeferenciats amb el datum europeu de 1950 (ED50) en coordenades UTM (x,y).

A efectes de visualització tindrem que reprojectar els arxius .shp al datum WGS84 (World Geodetic System 1984), sistema geodèsic mundial i de referència global utilitzat per OSM i Google i que per defecte ve configurat a la majoria de navegadors GPS.

La diferència de coordenades entre els datums ED50 i WGS84 pot arribar a ser de centenars de metres. 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, a l'hora d'inserir les capes a la base de dades espacials de PostgreSQL.

El mètode que utilitzarem per reprojectar els fitxers .shp serà crear un procés a partir de Geotrans que inclourem a la base de dades PostGIS, que ens reprojectarà els fitxers .shp una vegada executem les sentències sql a la base de dades PostgreSQL.

Obtenim la cartografia de la guia de carrers dels 36 municipis, separat per categories i municipis, on a cadascun d'ells trobem tres carpetes diferenciades :

- **01_Text_Transport** : Mitjans de comunicació principals
 - Aeroport, estació de FGC, estació de metro, estació de RENFE, parada del funicular, terminals d'autobusos, parades d'autobús, parades de tramvia, parades de tramvia blau, port comercial, port esportiu.
- **02_simbologia** : Directori de serveis
 - Ajuntaments, punts d'informació, bombers, centres ITV, cossos de seguretat, edificis religiosos, gasolineres, mercats municipals, oficines de correus, pàrquing municipal de pagament, pàrquing municipal gratuït, punt de reciclatge, hospitals, centres d'atenció primària, biblioteques, centres comercials, cinema, museus, teatres, parades de taxi.
- **03_Poligons_lineas** : Zones urbanes, equipaments, activitats econòmiques, parcs i jardins
 - Costa, rius, platges, canals, piscines, basses, molls en construcció, cementiri, aeroport, ferrocarril RENFE, ferrocarril FGC, ferrocarril TGV, telefèric, tramvia, bosc, límit de Jardins, Zona de conreu, límit parc AMB, límit parc natural, polígon parc natural, alineació edificada EC, alineació edificada EC urbanització, alineació no edificada EC, edificació interior EC, alineació edificada EI, alineació no edificada EI, edificació pública singular, edificació aïllada, zona esportiva, límit municipal.

Una vegada seleccionats els .shp dels elements que volem representar, realitzem una unió espacial de cada element de cada municipi, obtenint una .shp de tots els municipis de l'Àrea metropolitana de Barcelona per a cadascuna de les categories esmentades.

La unió espacial es va realitzar amb ArcGis 9.2 i l'eina d'unió espacial de arctoolbox.

- **Trams.shp**

Fitxer .shp d'elements lineals del carrer de la guia de AMB.

Hem de tenir en compte que aquest arxiu .shp, es van incloure els atributs de la base de dades GAMB.MDB, on podem trobar tota la informació referida als vials dels municipis amb els camps dels atributs més importants :

- NTRAM - Identificadors dels segments formats per cada vial
- CODI - Tipus de trams
- Nodel - Node inicial
- NodeF - Node final
- IVIAL - Identificador del vial
- NOMVIA - Nom del vial concatenat a partir de GAMB.MDB
- TVIAL - Tipus de vial
- XVIAL - Nexa o article del vial
- NVIAL - Nom del vial
- CINE - Codi INE Municipal
- DESC - Descripció del tipus de tram

- **Portals_amb.shp**

Informació de la numeració dels portals referides a cadascun dels trams del fitxer trams.shp

- **GAMB.MDB**

Base de dades relacional on podem trobar relacionades les diferents taules que conformen els atributs dels vials de la guia.

4.2.2. Creació de diccionari AMB-OSM

Com hem anomenant abans el Map Features de OSM és un diccionari de claus (Key) i valors (Value) que li donem als elements la cartografia.

Una vegada organitzats tots els fitxers .shp i decidir quins inclourem, tenim que relacionar els atributs dels .shp amb el llenguatge que utilitza OSM, per aquest motiu hem creat un diccionari on traduirem els atributs amb el valor corresponent a les claus i el valors que donen per a cada node, línia i polígon.

Consultant la wiki de OSM, en l'apartat de Map Features, anem buscant quina clau i quin valor li correspon a cada .shp i quin tipus d'element representem.

S'ha creat un full de càlcul amb excel (veure Fig 4.5) amb tots els elements per tal de facilitar la transformació al format .osm, podem trobar el diccionari sencer de AMB-OSM en el ANNEX I.









		Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el mapa de Carrers d'OpenStreetMap(OSM)				
		Diccionari AMB-OSM Trams				
	CODI	Descripció		KEY	VALUE	Element
	0	Vial Indefinit		highway layer	residential 0	
	7500	Vial carrers		highway layer	residential 0	
	7501	Vial carrers tunel		highway tunnel layer	residential yes -1	
	7502	Vial carrers construccio		highway layer	construction 0	
	7503	Vial carrers peatonals		highway layer	pedestrian 0	
	7504	Vial pas elevat		highway bridge layer	residential yes 1	
	7506	Vial laterals		highway layer	residential 0	

Fig 4.5. Mostra del diccionari AMB-OSM.

4.2.3. Estructura del format OSM

El format OSM són fitxers XML (Extensible Markup Language), un metallenguatge extensible d'etiquetes, on presenta tota la informació estructurada de manera que es compona de parts ben definides (veure Fig 4.6 i Fig 4.7) i que a la vegada aquestes parts formen les d'un altre, la seva estructura es semblant a la d'un arbre.

```
<tables>
  <table name="10010_aeroport">
    <key name="aeroway" value="aerodrom"/>
    <key name="layer" value="2"/>
    <record num="1">
      <column name="gid" type="integer">1</column>
      <column name="text_angle" type="double precision">0</column>
      <column name="text_size" type="double precision">2.5</column>
      <column name="textstring" type="character varying(254)">Aeroport del Prat -
Barcelona</column>
      <point name="lat" type="double precision">41.3028412792064</point>
      <point name="lon" type="double precision">2.06166842837146</point>
    </record>
  </table>
</tables>
...
```

Fig 4.6. Fitxer .xml de la shape d'aeroports.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6">
<node id="-1" visible="true" lat="41.3028412792064" lon="2.06166842837146">
  <tag k="name" v="Aeroport del Prat - Barcelona"/>
  <tag k="amb:gid" v="1"/>
  <tag k="amb:text_angle" v="0"/>
  <tag k="amb:text_size" v="2.5"/>
  <tag k="amb:textstring" v="Aeroport del Prat - Barcelona"/>
  <tag k="aeroway" v="aerodrom"/>
  <tag k="layer" v="2"/>
</node>
...
```

Fig 4.7. Fitxer .osm de la shape d'aeroports.

Els fitxers osm es defineixen a partir de les dades primitives (veure Fig 4.8), així tindrem diferents etiquetes per a cadascuna d'elles :

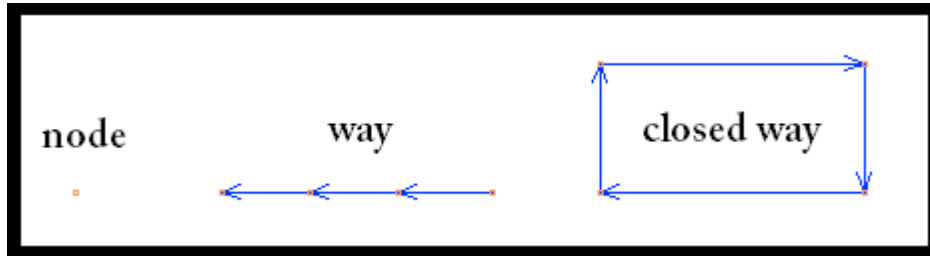


Fig 4.8. Dades primitives OSM.

- **Node**

Els nodes són els elements bàsics de l'esquema dels fitxers .osm, els nodes són punts georeferenciats amb una latitud i longitud determinada (dades geoespacionals d'un punt), en general no es registra l'altitud (Z). Els nodes són necessaris per definir a les altres dades primitives, per exemple les vies (ways) es formen amb varis nodes units i pot formar part de varies vies.

La seva etiqueta és `<node id="" lat="" lon=""></node>`, on definim el seu identificador, les dades geoespacionals (lat/lon) i les etiquetes de claus i valors `<tag k="" v=""/>`, que ens defineixen els atributs del node.

Els nodes, les formes, i les relacions de les dades primitives són representats, juntament amb les etiquetes corresponents (veure Fig 4.9). Un fitxer de OSM pot incloure també etiquetes que comporten canvis.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6">
<node id="-1" visible="true" lat="41.4718361740174" lon="2.25399691166599">
  <tag k="name" v=""/>
  <tag k="amb:gid" v="1"/>
  <tag k="amb:fid_gasoli" v="0"/>
  <tag k="amb:21070_g_id" v="0"/>
  <tag k="amb:area" v="0.000"/>
  <tag k="amb:fid_21070_" v="-1"/>
  <tag k="amb:id" v="0"/>
  <tag k="amb:orig_fid" v="0"/>
  <tag k="amenity" v="fuel"/>
  <tag k="layer" v="2"/>
</node>
...
```

Fig 4.9. Fitxer .osm de Nodes de les gasolineres

- **Way (Línies o vies)**

Una way o via és una interconnexió d' almenys dos nodes que representa un element lineal com un carrer o similar, i poden formar part d'una altre via. Per a poder ser considerada com a tal té que tindre almenys una clau amb un valor vàlid.

Primer s'identifica els nodes que formen la via `<node id="" lat="" lon="" />`. Amb l'etiqueta `<way id="" />` identifiquem el valor de la via i dintre d'aquesta referenciem `<nd ref="" />` els nodes que conformen la way anteriorment identificats fora de la tag way (veure Fig 4.10).

Finalment com a totes les dades primitives assignem les seves claus i valors que defineixen el tipus de via que és.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6">
<node id="-81" action="modify" visible="true" lat="41.4877806371541"
lon="2.23911305418168"/>
<node id="-82" action="modify" visible="true" lat="41.4875932331914"
lon="2.23928211283366"/>
<way id="-83">
  <nd ref="-81"/>
  <nd ref="-82"/>
  <tag k="amb:gid" v="2"/>
  <tag k="amb:4010_ri_id" v="0"/>
  <tag k="amb:24160_s_id" v="0"/>
  <tag k="waterway" v="river"/>
  <tag k="layer" v="-3"/>
</way>
...
```

Fig 4.10. Fitxer .osm de vies de un tram de riu

- **Closed way (vies tancades)**

Les àrees tancades no són elements. Aquestes formes es creen i es tanquen d'acord amb les etiquetes que se'ls ha assignat i tancat amb el node inicial i final de la via, per exemple li pot assignar una etiqueta amb valor landuse = forest i serà considerada un àrea de la mateixa manera que si tingués la línia tancada. Com les vies, les àrees han de tenir com a mínim una clau amb un valor específic d'àrea.

Per a mapes en els quals es necessita representar "forats" en una àrea, per exemple un clar en un bosc o un edifici amb un gran pati interior, s'ha d'utilitzar relacions de tipus multipolígon.

L'estructura es la mateixa que les ways amb la diferència que sempre acaba la via en el node que ha començat a dibuixar-la, així definim l'àrea tancada esmentada abans. En aquest cas (veure Fig 4.11) veiem que comença a dibuixar en el node -1 i acaba la way en aquest mateix punt.

Aquest cas (veure Fig 4.11) donem la clau de edificacions (building) i li confirmem amb el valor (yes), una de les altres formes de definir una àrea tancada.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6">
<node id="-1" action="modify" visible="true" lat="41.5125312761306" lon="2.14028848313872"/>
<node id="-2" action="modify" visible="true" lat="41.5125344990992" lon="2.14023405974035"/>
<node id="-3" action="modify" visible="true" lat="41.5125636720681" lon="2.14023989301793"/>
<node id="-4" action="modify" visible="true" lat="41.5125577838554" lon="2.1402920269302"/>
<node id="-5" action="modify" visible="true" lat="41.5125312761306" lon="2.14028848313872"/>
<way id="-6">
  <nd ref="-1"/>
  <nd ref="-2"/>
  <nd ref="-3"/>
  <nd ref="-4"/>
  <nd ref="-5"/>
  <nd ref="-1"/>
  <tag k="amb:gid" v="1"/>
  <tag k="amb:5590_ed_id" v="0"/>
  <tag k="amb:area" v="0"/>
  <tag k="building" v="yes"/>
  <tag k="layer" v="-2"/>
</way>
...

```

Fig 4.11. Fitxer .osm de vies tancades (edificació aïllada)

• **Altres dades primitives**

Per últim existeixen les **relacions**, que permeten enllaçar entre si *nodes*, *ways* i altres relacions, que no utilitzem en el projecte actual però serveixen per ampliar la informació dels elements.

Alguns exemples per als que s'utilitza :

- Definir *multipolígons*(relacionant uns *ways* tancats entre si).
- Rutes (línies d'autobús, metro, rutes a peu, bici, etc...).
- Relacions de restriccions de gir als nodes de les vies.
-

Per a més informació, podem veure les que utilitza en el següent enllaça:

<http://wiki.openstreetmap.org/wiki/Relations>

4.2.4 Importació de dades a POSTGRESQL + PostGIS

Podem utilitzar tots els objectes bàsics que apareixen en l'especificació OpenGIS com a punts, línies, polígons, multilínies, multipunts i col·leccions geomètriques. Els objectes GIS suportats per PostGIS són de característiques simples definides per OpenGIS.

Exemples de la representació de la geometria en mode text :

```
POINT(0 0 0)
LINESTRING(0 0,1 1,1 2)
POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))
MULTIPOINT(0 0 0,1 2 1)
MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))
MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))
GEOMETRYCOLLECTION(POINT(2 3 9),LINESTRING((2 3 4,3 4 5))
```

Per a crear les taules en PostGIS, afegim en la sentència de creació una columna per a la geometria que es diferencia per a cadascun dels elements a representar.

Es creen dues taules principals (veure Fig 4.12) a la base de dades :

- Taula “**SPATIAL_REF_SYS**” : Conté els identificadors numèrics i descripcions dels sistemes de referència.
- Taula “**GEOMETRY_COLUMNS**” : Emmagatzema un índex de taula que conté algun tipus de geometria.

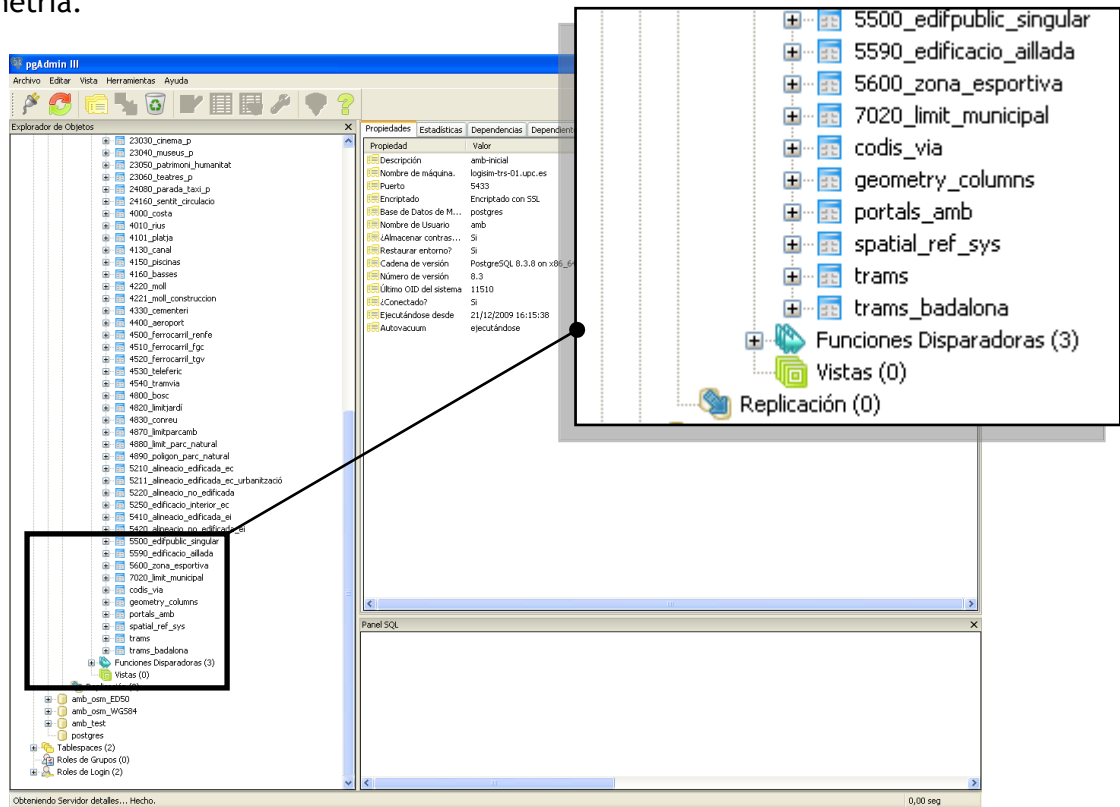


Fig 4.12. Taules de geometria.

Veiem un exemple de sentència .sql :

```
SET STANDARD_CONFORMING_STRINGS TO ON;
BEGIN;
CREATE TABLE "10120_port_esportiu" (gid serial PRIMARY KEY,
"text_angle" float8,
"text_size" float8,
"textstring" varchar(254));
SELECT AddGeometryColumn('','10120_port_esportiu','the_geom','23031','POINT',2);
INSERT INTO "10120_port_esportiu" ("text_angle","text_size","textstring",the_geom)
VALUES ('0.00.....
```

Veiem que afegim diverses columnes amb *AddGeometryColumn*, on posem el nom de la taula, el valor de la geometria, l'identificador i el tipus d'element que inserim, en aquest cas un punt i per últim la dimensió en que estem treballant, només amb coordenades X i Y.

A continuació realitzem l' inserció amb els valors de les taules que tenim a les .shp.

Per al nostre projecte creem una base de dades local per anar carregant les dades dels .shp que hem convertit en sentències .sql

4.2.4.1 Creació de sentències .sql amb shp2pgsql

Per omplir la base de dades PostgreSQL, es necessari crear les sentències per a implementar les taules de les shape de la guia de carrers.

Hi ha diferents mètodes per importar fitxers .shp a PostgreSQL, en aquest cas hem utilitzat una eina anomenada *shp2pgsql* que ens crearà la sentència .sql de cadascun dels fitxers .shp per a poder importar-los a PostGIS.

L'executable es realitza des de la consola del sistema operatiu on li tindrem que mostrar les diferents opcions (veure Fig 4.13) per a realitzar la importació amb èxit, com ara les rutes dels fitxers orígens i finals, la projecció, i els noms dels arxius .shp (veure Fig 4.14).

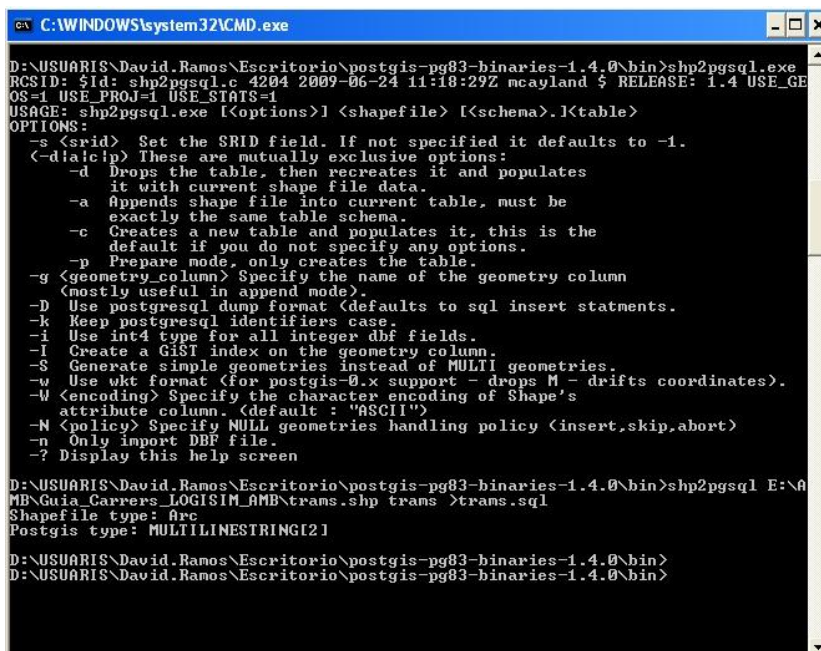


Fig 4.13. Opcions de la eina shp2pgsql.

```
Shp2pgsql -s<projecció> <Ruta del fitxer shp> <Ruta de la taula> <Nom del fitxer .sql>
shp2pgsql -s 4326 E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_WGS84\Trams.shp Trams
>E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_WGS84\SQL\Trams.sql
```

Fig 4.14. Exemple de sentència de importació (Trams.shp).

4.2.4.2 Importació amb fitxer .bat

El problema que va comportar anar important taula per taula, ens va obligar a crear un fitxer executable .bat on concatenem (veure Fig 4.15) totes les ordres que permet automatitzar tot el procés d'importació de .shp a PostGIS.

```
@echo *****
@echo SHP2PGSQL CARTOGRAFIA AMB_ED50
@echo LOGISIM_AMB
@echo *****
@echo off
Pause
@echo *****Trams, Portals i Trams de Badalona*****
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin\shp2pgsql -s 23031
E:\AMB\Guia_Carrers_LOGISIM_AMB\trams.shp Trams
>E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\SQL\trams.sql
...
Pause
@echo *****03_Poligonos_lineas*****
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin\shp2pgsql -s 23031
E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\03_Poligonos_lineas\4000_costa.shp
4000_costa >E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\03_Poligonos_lineas\SQL\4000.sql
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin\shp2pgsql -s 23031
...
@echo *****01_text_transport*****
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin\shp2pgsql -s 23031
E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\01_text_transport\10010_aeroport.shp
10010_aeroport
>E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\01_text_transport\SQL\10010.sql
...
Pause
@echo *****02_simbologia*****
D:\USUARIS\David.Ramos\Escritorio\postgis-pg83-binaries-1.4.0\bin\shp2pgsql -s 23031
E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\02_simbologia\21010_Ajuntaments_P.shp
21010_Ajuntaments_P
>E:\AMB\Guia_Carrers_LOGISIM_AMB\SHP_AMB_ED50\02_simbologia\SQL\21010.sql
...
Pause
```

Fig 4.15. Extracte de l'executable .bat dels fitxes .shp.

Finalment quan ja tenim totes les sentències, l'únic que ens resta és executar-les a PostgreSQL (veure Fig 4.16) i omplir la base de dades que utilitzarem per realitzar la transformació al format de OSM.

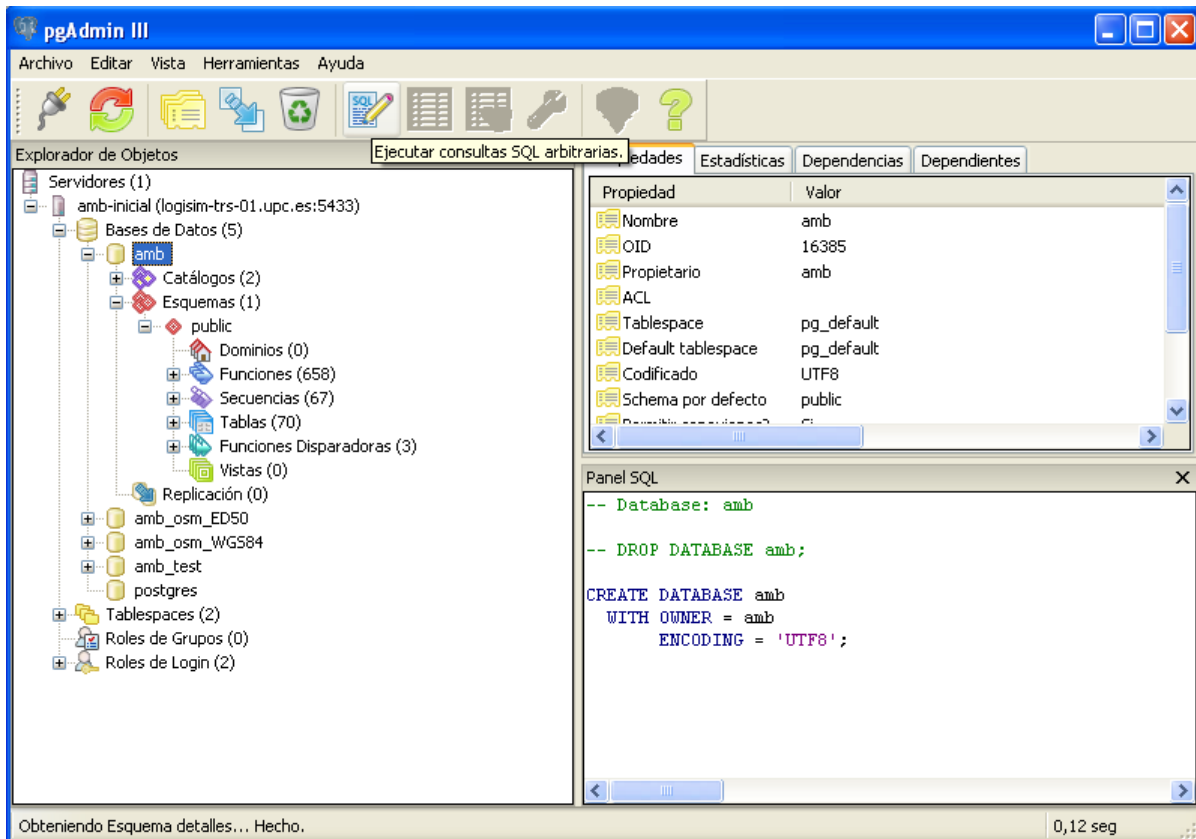


Fig 4.16. Base de dades PostgreSQL amb.

4.3. Conversió de dades OSM

L'objectiu en aquest apartat és explicar pas per pas com convertim les dades de la base geoespacial en PostGIS, en un fitxer en format XML-OSM, que pot ser llegit per qualsevol aplicació d'edició d'OpenStreetMap.

L'aplicació s'ha creat amb una eina de programari lliure java anomenada Netbeans que agafa diferents mòduls per aconseguir crear els objectes, classes i funcions que ens permetrà transformar les taules carregades a la BD geoespacial de Postgis al format XML i, a la vegada, traduir-ho amb les plantilles XSLT, amb el que aconseguirem completar la transformació a fitxers .osm.

Les **XSLT** o **transformacions XSL** és un estàndard de l'organització W3C que presenta una forma de transformar documents XML en altres i fins i tot a formats que no són XML. Els fulls d'estil XSLT, encara que el terme de fulles d'estil no s'aplica sobre la funció directa de l'XSLT, realitzen la transformació del document utilitzant una o diverses regles de plantilla. Aquestes regles de plantilla unides al document font a transformar alimenten un processador de XSLT, el qual realitza les transformacions desitjades posant el resultat en un fitxer de sortida.

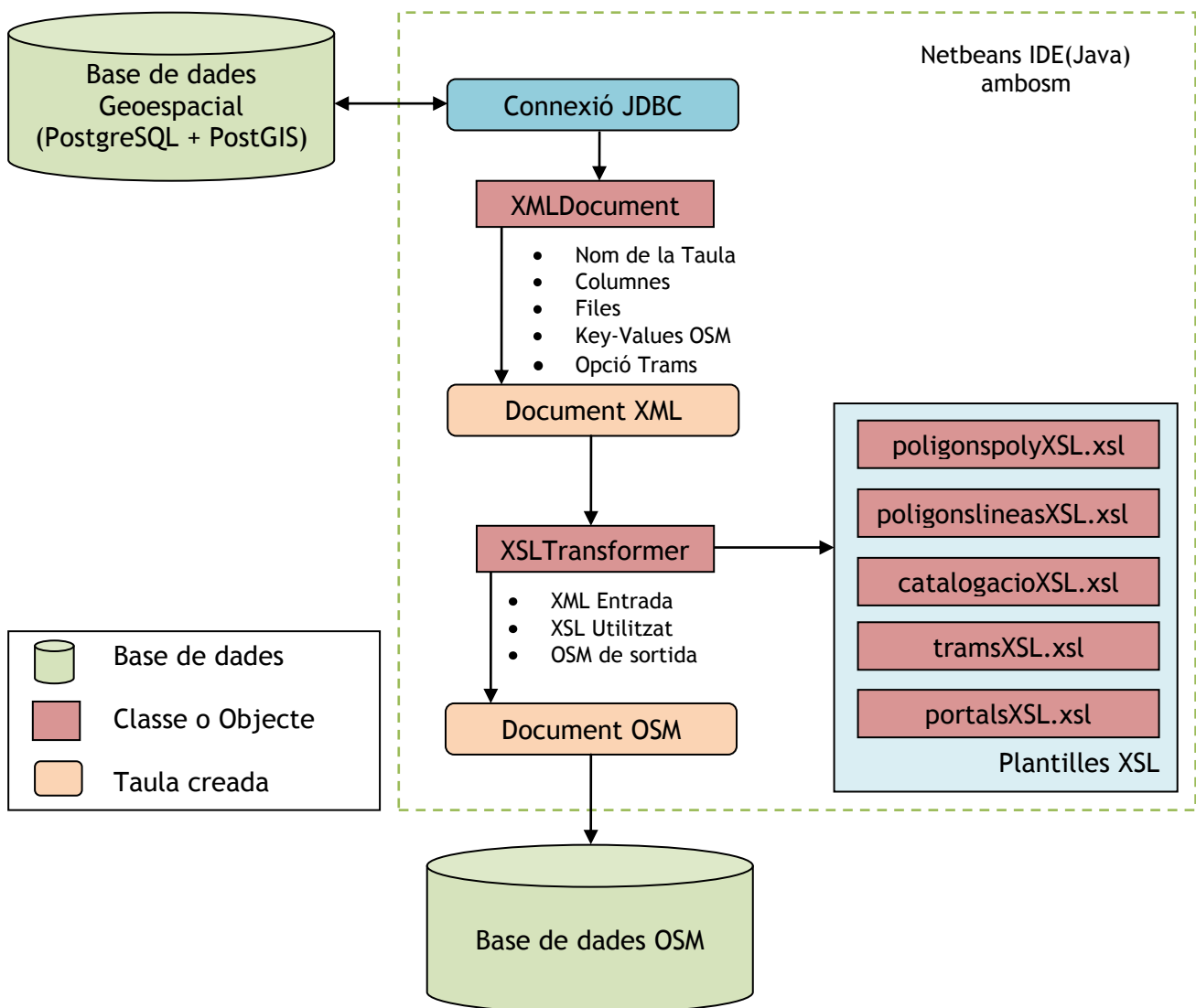


Fig 4.17. Estructura de l'aplicació ambosm.

4.3.1. Extracció i tractament de dades geospacials de la BD en PostGIS

4.3.1.1. Definició de la connexió

El primer pas que hem de realitzar és la definició dels paràmetres de connexió a la BD. Des del main de l'aplicació crearem un objecte de tipus ambConnection, que contindrà la connexió a la BD:

```

/* Main.java */
{...}
ambConnection conn = new ambConnection(args[0], args[1], args[2]);
{...}

/* ambConnection.java */

public class ambConnection {

    private java.sql.Connection connection;
    private Statement statement;
    private String query;
    // Parametres de connexió

    static final private String driverUrl = "jdbc:postgresql://";

    // Constructora de la connexió
    public ambConnection(String url, String user, String pass) {
        try {
            // Indiquem el driver per a la connexió
            Class.forName("org.postgresql.Driver");
            // Establim la connexió
            connection = DriverManager.getConnection(driverUrl+url, user,
pass);
            ((org.postgresql.PGConnection)
connection).addDataType("geometry", Class.forName("org.postgis.PGgeometry"));
            ((org.postgresql.PGConnection)
connection).addDataType("geometry", Class.forName("org.postgis.PGbox3d"));

        } catch (java.lang.ClassNotFoundException e) {
            System.out.println("No se encontró el driver para la base de
datos");
        } catch (java.sql.SQLException e) {
            System.out.println("Error al conectar a la Base de Datos");
        }
    }
}

```

Com podem veure, els paràmetres de connexió són arguments de l'execució de l'aplicació, més concretament, els tres primers, definits de la següent forma: {URL BD}, {user}, {pass} Amb aquesta informació realitza la connexió utilitzant la llibreria JDBC de Java (veure annex) i indica a la connexió que pot rebre dos tipus d'objectes: "org.postgis.PGgeometry" i "org.postgis.PGbox3d".

4.3.1.2. Construcció de la sentència principal (taules a convertir)

El temps de conversió de totes les taules de la BD és elevat, degut a l'extensió d'algunes d'aquestes. Per tant, s'ha inclòs la possibilitat de convertir només les taules indicades, o bé totes.

```

/* Main.java */
{...}
conn.createStatement();
conn.setQueryTables(args);
tables = conn.executeQuery();
{...}

/* ambConnection.java */

public void setQueryTables(String[] args) {
    // Construim la consulta inicialment sense el nom de les taules
    if (args[3].equals("all")) {
        String tablequery = "SELECT tablename FROM pg_tables WHERE tableowner =
'amb' AND tablename != 'codis_via' AND tablename != 'portals_amb'";
        this.setQuery(tablequery);
    } else {
        String tablequery = "SELECT tablename FROM pg_tables WHERE tableowner =
'amb' AND (";
        for (int i = 3; i < args.length - 1; i++) {
            // Concatenem el nom de les taules excepte l'ultima
            tablequery += "tablename = '" + args[i] + "' OR ";
        }
        // Afegim l'ultima taula sense cap OR i tancant el parèntesi
        tablequery += "tablename = '" + args[args.length - 1] + "'";
        this.setQuery(tablequery);
    }
}
}

```

Els arguments relatius a les taules a convertir comencen a partir de la quarta posició. Si aquest paràmetre és “all”, la consulta retornarà el nom de totes les taules, exceptuant la de “codis_via” i “portals_amb”, que no s'han de convertir. En canvi, si hi ha un llistat d'una o més taules, el bucle construirà la sentència per a que la consulta retorni el nom de les taules indicades.

4.3.1.3. Generació dels XML's i conversió a XML-OSM

En aquest punt descriurem el procediment principal de l'aplicació, sense entrar en detall de com es fan les conversions (punts 3 i 4). Els passos son:

1. Creació de l'objecte XMLDocument() fora del bucle.

```
XMLDocument xml = new XMLDocument();
```

2. Generar en disc el fitxer amb extensió xml i obtenció del nom de la taula.

```

while (tables.next()) {

tablename = tables.getString(1);
xml.generateXMLDocument(xml.getXMLPath(tablename), conn);
}

```

3. Consulta de columnes de la taula

```
conn.setQuery("SELECT a.attname as \"Column\", pg_catalog.format_type(a.atttypid,
a.atttypmod) as \"Datatype\" FROM pg_catalog.pg_attribute a WHERE a.attnum > 0
AND NOT a.attisdropped AND a.attrelid = (SELECT c.oid FROM pg_catalog.pg_class c
LEFT JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamespace WHERE c.relname ~
'^(\" + tablename + "\")$' AND pg_catalog.pg_table_is_visible(c.oid))");
conn.setStatement();
columns = conn.executeQuery();
```

4. Consulta de les dades de la taula

```
conn.setQuery("SELECT *, ST_AsText(the_geom) as geom FROM \"\" + tablename +
\"\"");
conn.setStatement();
result = conn.executeQuery();
```

5. Ús del codi de la taula per a obtenir els parells de key-values propis d'OSM i escriptura del XML.

```
codi = tablename.split("_")[0];

if (codi.equals("trams")) {
    xml.saveXMLDocument(tablename, columns, result, values, true);
} else {
    conn.setQuery("SELECT * FROM codis_via WHERE codi=" + codi);
    conn.setStatement();
    values = conn.executeQuery();
    values.next();
    xsl = values.getString("xsl");
    xml.saveXMLDocument(tablename, columns, result, values, false);
}
```

El tractament és el mateix per a totes les taules excepte per la de trams. Aquesta taula conté a cada fila un codi que representa un tipus de vial, per tant, no podem obtenir els parells de key-values per a tota la taula, sinó que per a cada entrada haurem de fer la consulta pertinent. Tota la informació es troba a la taula codis_via.

Finalment es farà la crida a la funció saveXMLDocument, que s'encarregarà de fer tot el tractament de les dades. Com a paràmetres es passen:

- Nom de la taula (tablename).
- ResultSet de Columnes (columns).
- ResultSet amb les files de la taula (result).
- ResultSet amb els parells key-values (values).
- Booleà que indica si la taula es de tipus trams o no.

6. Tancament del XML

```
xml.closeXMLDocument();
```

7. Transformació a XML-OSM

```
if (codi.equals("trams")) {
    xsl = "tramsXSL.xsl";
    xsltransf = new XSLTransformer(xml.getXMLPath(tablename),
xml.getXSLPath(xsl), xml.getOSMPath(tablename));
} else {
    xsltransf = new XSLTransformer(xml.getXMLPath(tablename),
xml.getXSLPath(xsl), xml.getOSMPath(tablename));
}

xsltransf.convert();
```

Amb el xsl agafat a l'apartat 5 (o tramsXSL.xsl si es una taula de tipus tram), creem l'objecte de tipus XSLTransformer, passant com a paràmetres:

- Path del XML origen.
- Path del XSL a utilitzar per a la conversió.
- Path del OSM on s'escriurà el resultat.

4.3.2. Generació del XML

4.3.2.1. Justificació de la llibreria SAX en lloc de DOM

En un primer estadi de la implementació de l'aplicació, es va començar a utilitzar DOM per a la generació del XML. Per al tractament de taules petites no suposava cap problema, però per al tractament de taules grans donava problemes de memòria (Not enough Java Heap Space) que no es van poder solucionar.

El motiu d'aquesta falta de memòria era la generació de XML's de mida molt gran i la forma d'escriure a disc de la llibreria DOM. El seu funcionament es basa en emmagatzemar en un objecte de la classe StreamResult tota la informació a escriure en el XML. Posteriorment, aquest StreamResult s'ha de passar a String, pel que ha de fer una còpia de tot el contingut a memòria, ja que utilitzarà aquestes dades per a escriure mitjançant un OutputBuffer.

Problema amb la còpia a memòria: Degut a la gran mida de l'StreamResult, al generar l'String amb la crida objecte.toString(), necessita una gran quantitat de memòria (+GB). Per tant, la problemàtica està en que la forma de treballar amb DOM exigeix tenir totes les dades a memòria abans de començar l'escriptura.

Per una altra banda, SAX és molt menys intensiu en l'ús de memòria, ja que a mesura que es van definint els tags i la informació que contindrà el XML, va escrivint a disc, permetent l'alliberació d'aquella informació que ja ha estat escrita i, per tant, no necessària.

4.3.2.2. Imports i constants

La classe XMLDocument requereix una sèrie de llibreries per al seu funcionament. Tanmateix, per motius d'eficiència, s'han descrit un seguit de constants, que defineixen les etiquetes (tags) i els seus atributs o valors.

```
/* ### Imports ###
 * Input / Output
 * XML Serializer
 * Clases Postgis per al tractament de geometries
 * SAX per al tractament de l'XML
 */
import java.io.*;
import java.sql.SQLException;
import java.sql.ResultSet;
import org.apache.xml.serialize.*;
import org.postgis.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
```

```
// Constants tipus String per a tags i atributs

private static final String TAG_COLUMN = "column";
private static final String TAG_GEOMETRY = "geometry";
private static final String TAG_KEY = "key";
private static final String TAG_LAT = "lat";
private static final String TAG_LON = "lon";
private static final String TAG_NODE = "node";
private static final String TAG_RECORD = "record";
private static final String TAG_ROOT = "tables";
private static final String TAG_TABLE = "table";
private static final String TAG_WAY = "way";
private static final String TAG_POINT = "point";
private static final String ATTR_ID = "id";
private static final String ATTR_NAME = "name";
private static final String ATTR_NUM = "num";
private static final String ATTR_TYPE = "type";
private static final String ATTR_VALUE = "value";
private static final String XML_ENCODING = "UTF-8";
```

4.3.2.3. Constructora

La constructora bàsicament prepara els objectes que posteriorment s'utilitzaran per a la construcció del XML. Tenim:

- **ambConnection conn**: Connexió a la BD, per realitzar les consultes pertinents.
- **AttributesImpl atts**: Objecte contenidor dels atributs d'un tag.
- **FileOutputStream fos**: Stream d'escriptura de fitxer. Té com a paràmetre la ruta on es crearà el fitxer xml i un booleà que indica si volem escriure al final del fitxer (append) o reescriure'l. En el nostre cas, voldrem reescriptura.
- **OutputFormat of**: Objecte que indica quin format de sortida volem. En el nostre cas, utilitzarem format XML amb indentació 1 activada.

- **XMLSerializer *serializer***: Construïm un objecte del tipus XMLSerializer, passant com a paràmetre el FileOutputStream i el tipus de codificació.
- **ContentHandler *hd***: Es el contenidor on es posarà la informació. Per a la creació d'aquest contenidor cal un objecte del tipus XMLSerializer.

Per últim, iniciarem el document (funció startDocument()) i afegirem també el node arrel del document (funció startElement()).

```
// Objectes
ContentHandler hd;
AttributesImpl atts;
FileOutputStream fos;
OutputFormat of;
ambConnection conn;

public void generateXMLDocument(String file, ambConnection conn) {
    try {
        // Assignem connexió
        this.conn = conn;
        // Obrim el fitxer corresponent
        fos = new FileOutputStream(file, false);
        // Indiquem la codificació, identació
        of = new OutputFormat("XML", XML_ENCODING, true);
        of.setIndent(1);
        of.setIndenting(true);
        // Definim el XML serialitzable
        XMLSerializer serializer = new XMLSerializer(fos, of);

        // Generem el contentHandler
        hd = serializer.asContentHandler();
        // Generem el document
        hd.startDocument();
        atts = new AttributesImpl();

        // Afegim al document el tag inicial
        hd.startElement("", "", TAG_ROOT, atts);
    } catch (java.io.FileNotFoundException e) {
        e.printStackTrace();
    } catch (java.io.IOException e) {
        e.printStackTrace();
    } catch (org.xml.sax.SAXException e) {
        e.printStackTrace();
    }
}
```

4.3.2.4. Creació del XML

Els passos per a la generació del XML són els següents:

1. Crida a la funció saveXMLDocument:

```
public void saveXMLDocument(String tablename, ResultSet columnname, ResultSet result, ResultSet values, boolean trams) {
```

Els paràmetres son:

- String tablename: cadena de caràcters que conté el nom de la taula a tractar.
- ResultSet columnname: Conté el nom de les columnes d'aquella taula.
- ResultSet result: Conté totes les files de la taula (de la BD) a tractar.
- ResultSet values: Conté la fila de la taula "codis_via" per a la taula a tractar (parells key - value: tunnel, bridge, tracktype...).
- Boolean trams: Booleà que indica si la taula a tractar es de tipus tram o no.

2. Creació del tag taula amb el seu nom:

```
// Creem el tag table amb un atribut name que conte el nom de la taula
atts.clear();
atts.addAttribute("", "", ATTR_NAME, "CDATA", tablename);
hd.startElement("", "", TAG_TABLE, atts);
```

Per a afegir un tag, el procediment es sempre el mateix:

- Buidat dels atributs del tag (atts.clear()).
- Creació de la llista d'atributs que contindrà el tag, per tant, tantes crides a la funció addAttribute com siguin necessàries. Els paràmetres importants: el tercer, que serà el nom de l'atribut, i el cinquè que serà el valor de l'atribut.

```
atts.addAttribute("", "", ATTR_NAME, "CDATA", tablename);
```

- Creació del tag, assignant la seva llista d'atributs (atts). El nom del tag és el tercer paràmetre de la funció startElement.

3. Si la taula no és de tipus tram, crida a la funció `addXMLKeyValues(values)`;

Afegir parells key - values, extrets de la taula `codis_via`. Per tant, són els parells propis d'OSM. Hi hauran diferents condicions, per a comprovar si aquest parell key-value s'ha de posar al XML o no.

```
atts.clear();
atts.addAttribute("", "", ATTR_NAME, "CDATA", values.getString(3));
atts.addAttribute("", "", ATTR_VALUE, "CDATA", values.getString(4));
hd.startElement("", "", TAG_KEY, atts);
hd.endElement("", "", TAG_KEY);

if (Integer.valueOf(values.getString(5)) != -10) {
    atts.clear();
    atts.addAttribute("", "", ATTR_NAME, "CDATA", "layer");
    atts.addAttribute("", "", ATTR_VALUE, "CDATA", values.getString(5));
    hd.startElement("", "", TAG_KEY, atts);
    hd.endElement("", "", TAG_KEY);
}
if (Integer.valueOf(values.getString(6)) != 0) {
    atts.clear();
    atts.addAttribute("", "", ATTR_NAME, "CDATA", "bridge");
    atts.addAttribute("", "", ATTR_VALUE, "CDATA", "yes");
    hd.startElement("", "", TAG_KEY, atts);
    hd.endElement("", "", TAG_KEY);
}
if (Integer.valueOf(values.getString(7)) != 0) {
    atts.clear();
    atts.addAttribute("", "", ATTR_NAME, "CDATA", "tunnel");
    atts.addAttribute("", "", ATTR_VALUE, "CDATA", "yes");
    hd.startElement("", "", TAG_KEY, atts);
    hd.endElement("", "", TAG_KEY);
}
if (values.getString(8) != null) {
    atts.clear();
    atts.addAttribute("", "", ATTR_NAME, "CDATA", "tracktype");
    atts.addAttribute("", "", ATTR_VALUE, "CDATA", values.getString(8));
    hd.startElement("", "", TAG_KEY, atts);
    hd.endElement("", "", TAG_KEY);
}
```

4. Comencem el bucle que tractarà tots els registres (files de la taula). Afegim el tag record, amb un atribut "num" que contindrà el valor d'un comptador que s'incrementarà a cada iteració:

```
while (result.next()) {
    contcolumn = 1;
    atts.clear();
    atts.addAttribute("", "", ATTR_NUM, "CDATA", String.valueOf(cont));
    hd.startElement("", "", TAG_RECORD, atts);
```

5. Per a cada record, posarem totes les seves columnes, mitjançant un bucle:

Per a la columna "the_geom", a més d'afegir la columna corresponent, cridarem a la funció `getLatLonPoints`, que ens retornarà una matriu amb tots els punts de la geometria, de mida `[numPoints][2]`, és a dir, una entrada per a cada punt, on guardarem la latitud - longitud.

```
while (columnname.next()) {
    atts.clear();
    atts.addAttribute("", "", ATTR_NAME, "CDATA", columnname.getString(1));
    atts.addAttribute("", "", ATTR_TYPE, "CDATA", columnname.getString(2));
    hd.startElement("", "", TAG_COLUMN, atts);

    if (columnname.getString(1).equals("the_geom")) {
        geom = new PGgeometry(result.getString("geom"));
        points = getLatLonPoints(geom);
        hd.characters(result.getString("geom").toCharArray(), 0,
            result.getString("geom").length());
    }else {
        if (result.getString(contcolumn) != null) {
            hd.characters(result.getString(contcolumn).toCharArray(), 0,
                result.getString(contcolumn).length());
        }
    }
    hd.endElement("", "", TAG_COLUMN);
    contcolumn++;
}
```

6. Afegim l'estructura de la geometria al XML.

```
// Afegim l'estructura de nodes per a poder llegir la geometria
nodenum = addXMLGeometry(geom.getGeoType(), points, nodenum);
```

Bàsicament, aquesta funció el que fa és, a partir del tipus de geometria (`geom.getGeoType()`), la matriu de punts, i el numero de node, definir l'estructura d'aquesta geometria al XML, mitjançant tags, tal i com s'ha explicat fins ara.

7. Retrocedir el punter del ResultSet columnes per al pròxim record.

```
// Punter al resultSet cap a l'inici
columnname.beforeFirst();
```

8. Si la taula és de tipus trams afegim els parells key - value, passant per paràmetre el codi del record actual de la taula trams.

```
addXMLKeyValuesTrams(Double.valueOf(result.getString("codi")).intValue());
```

El procés es anàleg al del punt 3. L'únic que canvia és la consulta que hi ha a l'inici de la funció. Aquesta consulta s'ha de fer perquè les taules de tipus trams no tenen un mateix codi per a tota la taula, sinó que cada record en té un.

```
private void addXMLKeyValuesTrams (int codi) {
// Query que retorna els valors de la taula de codis_via indicat
result = getCodiResult(codi);
{...}
}

private ResultSet getCodiResult(int codi) {
// Query que retorna els valors de la taula de codis_via indicat
conn.createStatement();
conn.setQuery("SELECT * FROM codis_via WHERE codi=" + codi);
return conn.executeQuery();
}
```

9. Tanquem el tag del record actual.

```
hd.endElement("", "", TAG_RECORD);
```

10. Incrementem el comptador i tanquem el tag table.

```
cont++;
}
hd.endElement("", "", TAG_TABLE);
```

4.3.3. Transformació XSL

4.3.3.1. Imports i atributs

Bàsicament necessitem la llibreria per a escriure un fitxer a disc, i la relacionada amb les transformacions. I com a atributs de la classe:

L'objecte de la classe *TransformerFactory*, encarregat de fer la transformació.

- El XML d'origen.
- El XSL per a la transformació.
- El path del XML destí.

```
import java.io.FileOutputStream;
import javax.xml.transform.*;

public class XSLTransformer {

    TransformerFactory tFactory;
    private String xmlOrigen;
    private java.io.InputStream xslOrigen;
    private String xmlDestino;
```

4.3.3.2. Constructora

En la constructora, l'únic que ens cal fer es instanciar el *TransformerFactory* i assignar les direccions del *xmlOrigen*, el path del *XMLDestí* i el *xsl* a utilitzar.

```
public XSLTransformer(String xmlOrigen, java.io.InputStream xslOrigen, String
xmlDestino) {
    this.tFactory = TransformerFactory.newInstance();
    this.xmlOrigen = xmlOrigen;
    this.xslOrigen = xslOrigen;
    this.xmlDestino = xmlDestino;
}
```

4.3.3.3. Funció de conversió

```
// Assignem el XSL que utilitzarà per a la transformació
Transformer transformer = tFactory.newTransformer(new
javax.xml.transform.stream.StreamSource(this.xslOrigen));

// Indiquem el XML origen i el XML destí
transformer.transform(new
javax.xml.transform.stream.StreamSource(this.xmlOrigen), new
javax.xml.transform.stream.StreamResult(new
FileOutputStream(this.xmlDestino)));
```

La funció de conversió assigna el XSL al *TransformerFactory*. Posteriorment, crida a la funció *transform*, passant com a paràmetre:

- El xml original, en forma de *StreamSource*.
- El path del XML destí, en forma de *FileOutputStream*, que acabarà convertint en un *StreamResult* abans d'escriure'l a disc.

4.3.3.4. XSL's

4.3.3.4.1. Estructura bàsica dels XML's per geometries Multilinestring i MultiPolygon

```
<tables>
  <table>
    <key name="z" value="abc"/>
    ...
    <record num="1">
      <column name="x">...</column>
      <column name="y">...</column>
      ...
      <geometry>
        <node id="-1">
          <lat>40.2</lat>
          <lon>20.5</lon>
        </node>
        <node id="-2">
          <lat>40.5</lat>
          <lon>20.2</lon>
        </node>
      </geometry>
    </record>
    <record num ="2">
      ...
    </record>
    ...
  </table>
</tables>
```

Per a cada taula, hi ha un primer llistat de parells key-value, propis d'OSM i de la taula en concret que es vol convertir. Després, tenim un llistat de registres (files de la taula de la BD), i per a cada registre:

- Llista de columnes de la Base de Dades.
- Llistat de punts que conformen la geometria, dividida en nodes amb parells latitud-longitud.

- XSL per a la transformació de polígons (poligonspolyXSL.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match='tables'>
    <osm version='0.6'>
      <xsl:for-each select='table'>
        <xsl:for-each select='record'>
          <xsl:for-each select='geometry/node'>
            <xsl:variable name="lat" select="lat"/>
            <xsl:variable name="lon" select="lon"/>
            <xsl:variable name="nodenum" select="@id"/>
            <node id="{nodenum}" action='modify' visible="true"
lat="{lat}" lon="{lon}"/>
          </xsl:for-each>
          <xsl:variable name="wayid" select="geometry/way/@id"/>
          <way id="{wayid}">
            <xsl:for-each select="geometry/node">
              <nd ref="{@id}"/>
            </xsl:for-each>
            <!-- Agafem el primer node, per tancar l'area -->
            <nd ref="{geometry/node[1]/@id}"/>
            <xsl:for-each select="column">
              <xsl:variable name="value" select="."/>
              <tag k="{concat('amb:',./@name)}" v='{value}'/>
            </xsl:for-each>
            <xsl:for-each select="../key">
              <tag k='{@name}' v='{@value}'/>
            </xsl:for-each>
          </way>
        </xsl:for-each>
      </xsl:for-each>
    </osm>
  </xsl:template>
</xsl:stylesheet>
```

El XSL funciona mitjançant un procediment principal, que tracta cada registre que realitza les següents tasques:

- Un bucle intern que escriu la geometria, agafant per a cada node del XML els parells latitud longitud. L'identificador a escriure, l'agafarem de l'atribut del tag node (select="@id").

- **Exemple de sortida parcial (XML-OSM):**

```
<node id="-1" action="modify" visible="true" lat="41.2704398920841"
lon="2.04653494703801"/>
<node id="-2" action="modify" visible="true" lat="41.2705237606025"
lon="2.04694812929189"/>
<node id="-3" action="modify" visible="true" lat="41.2705351668479"
lon="2.04697673637076"/>
<node id="-4" action="modify" visible="true" lat="41.2705533638983"
lon="2.04706398460344"/>
<node id="-5" action="modify" visible="true" lat="41.2705644121661"
lon="2.0471472775906"/>
```

- Formació del way

Una way es forma fent el llistat d'identificadors de nodes dintre del tag way, que tindrà el seu propi identificador. En el cas d'un polígon, existeix la peculiaritat que l'últim element de la llista ha de ser el mateix que el primer, per tancar el polígon. Per últim, caldrà posar el llistat de nodes, tant els propis d'OSM com els de AMB.

- **Exemple de sortida parcial (XML-OSM):**

```
<way id="-283">
  <nd ref="-275"/>
  <nd ref="-276"/>
  <nd ref="-277"/>
  <nd ref="-278"/>
  <nd ref="-279"/>
  <nd ref="-280"/>
  <nd ref="-281"/>
  <nd ref="-282"/>
  <nd ref="-275"/>
  <tag k="amb:gid" v="7"/>
  <tag k="amb:4101_pl_id" v="0"/>
  <tag k="amb:area" v="0"/>
  <tag k="natural" v="beach"/>
  <tag k="layer" v="-3"/>
</way>
```

- **XSL per a la transformació de Línies (poligonslineasXSL.xsl)**

El funcionament es anàleg al de polígons, l'únic que no cal tancar. És a dir, quan es conforma la way no cal que l'últim element de la llista sigui el mateix que el primer.

4.3.3.4.2. Estructura bàsica dels XML's per geometries tipus point

```
<tables>
  <table>
    <key name="z" value="abc"/>
    ...
    <record num="-1">
      <column name="x">...</column>
      <column name="y">...</column>
      <point name="lat" type="double
precision">41.373628591334</point>
      <point name="lon" type="double
precision">2.17913662146749</point>
    </record>
    <record num ="-2">
      ...
    </record>
    ...
  </table>
</tables>
```

Per a cada taula, hi ha un primer llistat de parells key-value, propis d'OSM i de la taula en concret que es vol convertir. Després, tenim un llistat de registres (files de la taula de la BD), i per a cada registre:

- Llista de columnes de la Base de Dades.
- Parell latitud - longitud del punt.
- XSL per a la transformació de punts (catalogacioXSL.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match='tables'>
    <osm version='0.6'>
      <xsl:for-each select='table'>
        <xsl:for-each select='record'>
          <xsl:param name='nodenum'>
            <xsl:number/>
          </xsl:param>
          <node id="{concat('-', $nodenum)}" visible="true"
lat="{point[@name='lat']}" lon="{point[@name='lon']}">
            <tag k='name' v="{column[@name='textstring']}" />
            <xsl:for-each select="column">
              <xsl:variable name="value" select="."/>
              <tag k="{concat('amb:',./@name)}" v='{$value}' />
            </xsl:for-each>
            <xsl:for-each select="../key">
              <tag k='{@name}' v='{@value}' />
            </xsl:for-each>
          </node>
        </xsl:for-each>
      </xsl:for-each>
    </osm>
  </xsl:template>
</xsl:stylesheet>
```


El XSL funciona mitjançant un procediment principal, que tracta cada registre i realitza les següents tasques:

- El node amb el parell latitud - longitud.
- Llistat de parells key-values tant els propis d'OSM com els d'AMB.
- **Exemple de sortida parcial (XML-OSM):**

```
<record num="1">
  <column name="gid" type="integer">1</column>
  <column name="fid_ajunta" type="integer">0</column>
  <column name="21010_a_id" type="integer">0</column>
  <column name="area" type="numeric">0.000</column>
  <column name="fid_21010_" type="integer">-1</column>
  <column name="id" type="integer">0</column>
  <column name="orig_fid" type="integer">0</column>
  <point name="lat" type="double precision">41.4501444593703</point>
  <point name="lon" type="double precision">2.24774700009543</point>
</record>
```

4.3.3.4.3. Estructura bàsica dels XML's per geometries tipus Tram

```
<tables>
  <table>
    <record num="1">
      <column name="x">...</column>
      <column name="y">...</column>
      ...
      <geometry>
        <node id="-1">
          <lat>40.2</lat>
          <lon>20.5</lon>
        </node>
        <node id="-2">
          <lat>40.5</lat>
          <lon>20.2</lon>
        </node>
      </geometry>
      <key name="z" value="abc"/>
      ...
    </record>
    <record num="2">
      ...
    </record>
    ...
  </table>
</tables>
```

Tenim un llistat de registres (files de la taula de la BD), i per a cadascun d'ells :

- Llista de columnes de la Base de Dades.
- Un bucle intern que escriu la geometria, agafant per a cada node del XML els parells latitud longitud. L'identificador a escriure, l'agafarem de l'atribut del tag node (select="@id").

La particularitat de les taules tipus trams, es que cada registre té un codi que conté informació sobre parells key-values (bridge, tunnel...), per tant, cal consultar aquests valors a la taula "codis_via".

- XSL per a la transformació de taules tipus trams (tramsXSL.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match='tables'>
    <osm version='0.6'>
      <xsl:for-each select='table'>
        <xsl:for-each select='record'>
          <xsl:for-each select='geometry/node'>
            <xsl:variable name="lat" select="lat"/>
            <xsl:variable name="lon" select="lon"/>
            <xsl:variable name="nodenum" select="@id"/>
            <node id="{\$nodenum}" action='modify' visible="true"
lat="{\$lat}" lon="{\$lon}"/>
          </xsl:for-each>
          <xsl:variable name="wayid" select="geometry/way/@id"/>
          <way id="{\$wayid}">
            <!-- Geometria -->
            <xsl:for-each select="geometry/node">
              <nd ref="{@id}"/>
            </xsl:for-each>
            <tag k='name' v="{column[@name='nomvia']}" />
            <!-- Parells key-value AMB -->
            <xsl:for-each select="column">
              <xsl:if test="./@name != 'nomvia'">
                <xsl:variable name="value" select="."/>
                <tag k="{concat('amb:',./@name)}"
v="{\$value}"/>
              </xsl:if>
            </xsl:for-each>
            <!-- Parells key-value OSM -->
            <xsl:for-each select="key">
              <tag k="{@name}" v="{@value}"/>
            </xsl:for-each>
          </way>
        </xsl:for-each>
      </xsl:for-each>
    </osm>
  </xsl:template>
</xsl:stylesheet>
```

4.3.3.4.4. Estructura bàsica dels XML's per geometries tipus Portals

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match='tables'>
    <osm version='0.6'>
      <xsl:for-each select='table'>
        <xsl:for-each select='record'>
          <xsl:param name='nodenum'>
            <xsl:number/>
          </xsl:param>
          <node id="{concat('-', $nodenum)}" visible="true"
            lat="{point[@name='lat']}" lon="{point[@name='lon']}" />
          <xsl:for-each select="column" >
            <xsl:variable name="value" select="."/>
            <xsl:choose>
              <xsl:when test="./@name='text'">
                <tag k="addr:housenumber"
                  v="{ $value }" />
              </xsl:when>
              <xsl:otherwise>
                <tag k="{concat('amb:', ./@name)}"
                  v="{ $value }" />
              </xsl:otherwise>
            </xsl:choose>
          </xsl:for-each>
        </node>
      </xsl:for-each>
    </osm>
  </xsl:template>
</xsl:stylesheet>
```

Bàsicament l'únic que hem de controlar al procediment és que si la columna és la de text (numero de portal), aquesta cal posar-la amb la key d'OSM addr:housenumber i com a value, el valor de la columna.

4.3.4. Utilitats de l'aplicació

4.3.4.1. Accedir a recursos dintre d'un fitxer JAR

L'aplicació es vol presentar en un fitxer Java Archiver (JAR) per a que es pugui executar directament. Dintre del propi JAR es troben recursos que no poden ser accedit utilitzant els paths relatius al sistema de fitxers, sinó que s'ha d'indicar on es troba dintre del JAR.

```
this.getClass().getClassLoader().getResourceAsStream("ambosm/XSL/" + xsl);
```

S'ha de carregar el ClassLoader, agafar el recurs com un Stream, passant com a paràmetre la ruta amb el format (package + ruta). En el nostre cas, el package es diu ambosm, i el xsl està a la carpeta XSL/nom.

4.3.4.2. Paràmetres i exemples d'execució

Per proporcionar una certa flexibilitat a l'aplicació, es permet passar diferents paràmetres, per condicionar l'execució del programa. Aquests son:

1. Direcció de la base de dades. Ex: exemple.upc.es:5433/database
2. Usuari.
3. Password
4. Tenim dues possibilitats:
 - Paràmetre 'all'. Iniciarà la conversió de totes les taules de la BD.
 - Llistat de noms de taules separades per espai.

Degut l'ús extensiu de memòria que necessita durant la transformació XSL, passarem dos paràmetres de configuració addicionals, per a proporcionar una major quantitat de memòria a la màquina virtual de Java. Ho farem executant el jar.

Exemple amb paràmetre "all":

```
java -Xms32m -Xmx1524m -jar amb-osm.jar exemple.upc.es:5433/database  
user pass all
```

Exemple amb llistat de taules:

```
java -Xms32m -Xmx1524m -jar amb-osm.jar exemple.upc.es:5433/database  
user pass 2100_taula1 2200_taula2 ....
```

4.3.5. Renderització de mapes OSM

En aquest apartat implementem la base dades osm amb els fitxers creats per obtenir el pre-renderitzat gràfic amb Mapnik i obtenir les tiles o tesel·les destinat al nostre servidor gràfic. Utilitza la llibreria Anti-Grain Geometry per renderitzar amb anti-aliasing i amb Subpixel accuracy. Pot llegir shape, Rasters, TIFF i qualsevol format que soporti GDAL o OGR.

Mapnik , utilitza per renderitzar tesel·les (veure Fig 4.18) de 256 x 256 píxels en format .png, que es serveixen des del servidor de tiles, on crea el conjunt d'imatges per a cada nivell de visualització.

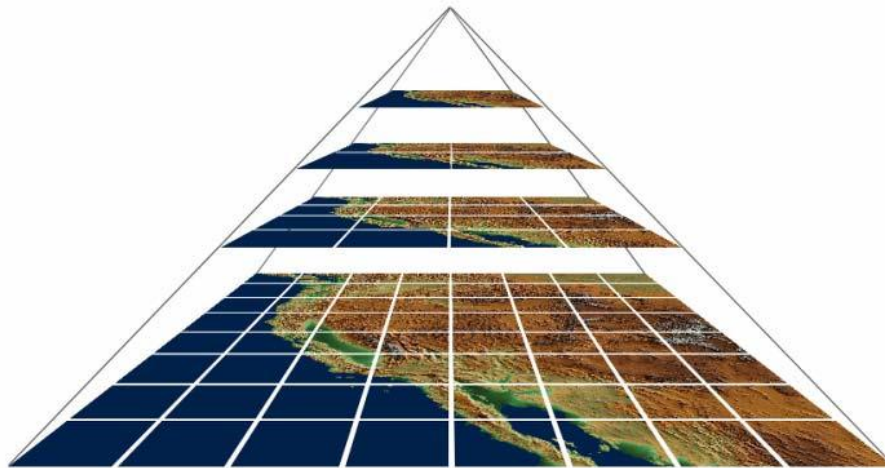


Fig 4.18. Model de Tiles o Tesel·les.

Al mateix temps, Mapnik ens permet amb osm2pgsql realitzar el pas invers per actualitzar les dades a la base de dades PostGIS, que s'editen amb els editors en línia i fora de línia.

4.3.6. Verificació i validació

Per comprovar la correcta transformació dels fitxers .osm, es va crear un servidor OSM amb tots els components que necessita OSM per fer proves de la correcta visualització de les dades i validar que hem escollit les etiquetes correctes al diccionari de AMB-OSM.

Es va fer una primera prova amb els arxius de la simbologia i comprovem que surten les icones que OSM ens permet dibuixar, més endavant es poden modificar per establir una sèrie de icones que s'adeqüin a la guia de carrers de AMB

El següent pas es comprovar la correcta visualització del fitxer de Trams, els carrers i els noms dels mateixos així com l'adequada visualització dels nivells dels carrers, especificades amb la tag de layers (k="Layers v="-5 a 5").

Per aquesta prova es va seleccionar el municipi de Badalona (veure Fig 4.19) per importar-ho a la base de dades i fer el renderitzat.

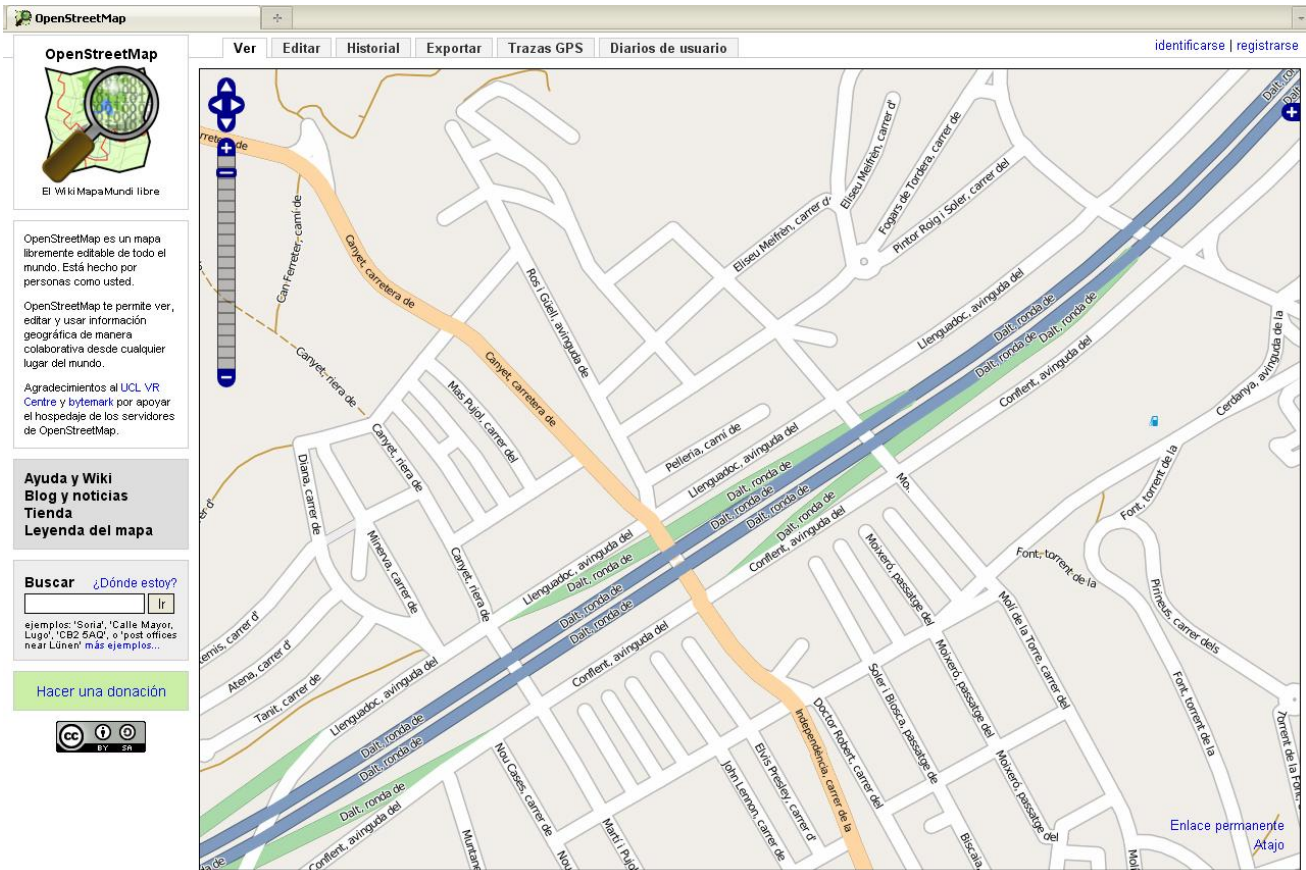


Fig 4.19. Trams de Badalona al servidor de proves.

Establím que les etiquetes per diferenciar els tipus de via ha estat correcte, en aquesta imatge es pot veure la jerarquització entre carreteres principals, secundàries i carrers, es poden veure a la vegada els camins i corriols amb el seu nom representat.

Es poden veure els carrers que pugen a un altre nivell com els accessos o els passos elevats, per això validem la transformació que s'ha realitzat del fitxer de trams.

4.4. Regeneració de dades AMB

Una vegada obtenim tots els fitxers en format OSM, és el moment de pujar les dades al servidor OSM. A l'inici es va plantejar de pujar totes les dades des de l'editor offline JOSM, però el temps de càrrega de les dades era molt elevat, degut al registre d'errors i incoherències que realitza durant el procés i a la quantitat d'informació que contenen algunes de les taules.

La solució s'ha trobat amb un script programat en Python, anomenat Bulk_upload (veure Fig 4.20) que ens importa dades osm massives en un temps relativament més curt, obviant els errors que surten genèricament. Els formats que llegeix son els formats de JOSM i *osmChange*.

En la figura 4.20 podem veure un exemple de la comanda bulk_upload per carregar un arxiu .osm, on marquem els diferents paràmetres per posar-les al servidor :

- Path del script
- Path de la base de dades amb el nom del fitxer (\$file)
- Mail de l'usuari
- Contrasenya de l'usuari

```
python -u /home/ambosm/osm/source/osmosis/osmosis-0.32/bin/bulk_upload.sh
-i /home/ambosm/osm/data/osm-v2/$file -u jaume.figueras@upc.edu
-p jaumefigueras -c $file
```

Fig 4.20. Comanda de bulk_upload per pujar arxius .osm.

Per automatitzar el procés de càrrega es va realitzar un procés per a que pugés cadascun dels fitxers .osm (veure Fig 4.21).

```
#!/bin/bash

for file in `dir -d *.osm` ; do
echo $file
python -u
/home/ambosm/osm/source/osmosis/osmosis-0.32/bin/bulk_upload.py -i
/home/ambosm/osm/data/osm-v2/$file -u jaume.figueras@upc.edu -p
jaumefigueras -c $file
done
```

Fig 4.21. Script d'automatització per pujar els fitxers .osm.

4.4.1. Edició de Mapes OSM sobre dades AMB

Amb la base de dades OSM carregades amb `bulk_upload`, entrem en la part de sincronització de les dues bases de dades i el procés d'actualització dels fitxers OSM quan s'editi al servidor web amb Potlatch o JOSM.

En aquesta fase sincronitzarem les dues bases de dades per mantenir actualitzada la informació en el moment que es faci qualsevol edició amb els dos editor principals, per aquesta tasca s'utilitza `osmosis` i `osm2pgsql`, cadascuna amb unes funcions determinades.

L'estructura estreta de la fase de treball en la sincronització es la següent :

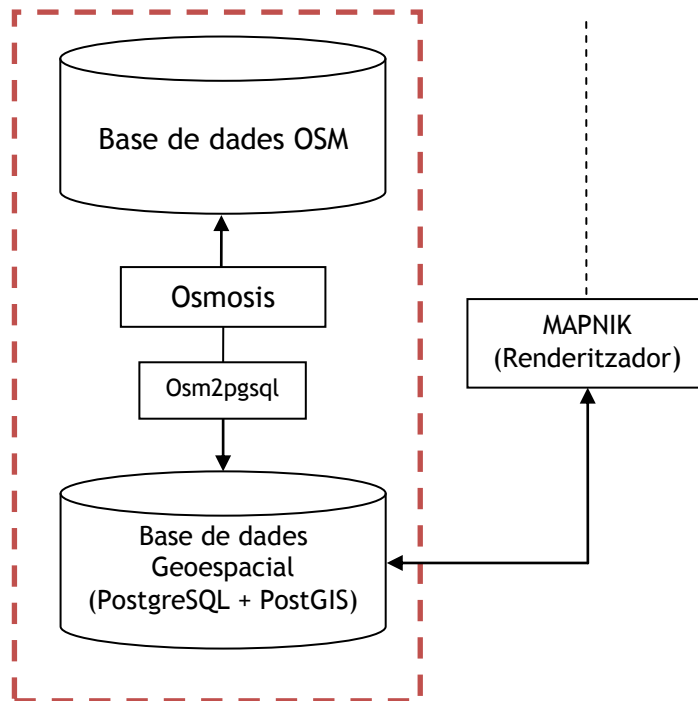


Fig 4.22. Estructura de la sincronització.

Osmosis és una comanda de línies d'aplicació en Java per al processament de dades OSM, per a realitzar lectura i escriptura a la base de dades i els seus arxius, creant un fitxer que ens mostri els canvis que s'han realitzat als fitxers OSM.

El procés de Osmosis es el següent :

- Crea un arxiu conjunt `.osm` de totes les dades inserides a la base de dades
- Compara els dos arxius `.osm` de diferents dates i crea un de nou amb els canvis realitzats(**osmChange**)
- Aplica els canvis que s'hagin realitzat entre dos arxius `.osm`

osmChange es el format d'arxiu OSM que utilitza `osmosis` per crear la diferència entre els dos arxius `.osm` de la base de dades.

Identifica les etiquetes que s'han modificat i els etiqueta dependent de quina es l'acció que s'ha realitzat a l'editor (veure Fig 4.23), crear, modificar o eliminar.

```
<osmChange version="0.3" generator="Osmosis">
  <modify version="0.3" generator="Osmosis">
    <node id="12050350" timestamp="2007-01-02T00:00:00.0+11:00" lat="-
33.9133118622908" lon="151.117335519304">
      <tag k="created_by" v="JOSM"/>
    </node>
  </modify>
</osmChange>
```

Fig 4.23. Exemple de fitxer osmChange. Modificar.

4.4.2. Exportació de dades OSM a PostGIS

Quan creem les modificacions a la base de dades OSM amb osmosis, cal retornar aquest canvis a la base de dades PostgreSQL amb PostGIS, aquesta funció es realitzarà a partir de la eina *osm2pgsql*.

osm2pgsql és una aplicació que permet la importació de dades en format osm a una base de dades PostgreSQL amb PostGIS amb els formats necessaris per a que puguin ser dibuixats per el renderitzador Mapnik.

Com hem comentat en el anterior apartat osmosis crea un fitxer únic on hi són tots els elements de la base de dades osm, amb aquest fitxer conjunt i *osm2pgsql* bolquem totes les dades a la base de dades PostgreSQL amb PostGIS (veure Fig 4.24).

```
./osm/source/osm2pgsql/osm2pgsql -m -Uambosm -W -H localhost -P5434
-dmapnikdata -S ./osm/source/osm2pgsql/default.style
/home/ambosm/osm/data/dump.osm
```

Fig 4.24. Comanda de importació de la base de dades OSM a PostGIS.

4.4.3. Sincronització

Per tal de mantenir la base de dades en PostGIS actualitzada, cal fer el pas invers al que havíem realitzat prèviament.

És a dir, cal llegir la BD d'OSM i generar un fitxer osm, utilitzant l'eina osmosis i posteriorment passar-ho a postgresQL amb l'eina osm2pgsql.

L'objectiu, una vegada realitzat el bulk_upload inicial, és fer canvis incrementals, tant de nodes nous com modificats, per tant, a l'eina osmosis li haurem de passar un interval de temps (veure Fig 4.25), per a que ens retorni només els canvis produïts en aquesta franja.

```
./osmosis --read-apidb-change validateSchemaVersion=no intervalBegin="2010-01-01_00:00:00" intervalEnd="2010-01-02_00:00:00" host="localhost:5433" database="osm" user="ambosm" password="ambosm" --write-xml-change file="canvis.osm"
```

Fig 4.25. Comanda de osmosis de verificació d'interval de temps.

El que fa la comanda és llegir la Base de Dades d'OSM i extreure un fitxer de canvis en l'interval indicat. L'intervalEnd és completament opcional, en el cas de no posar-ho, s'agafarà per defecte la data actual.

Una vegada obtingut el fitxer .osm, cal utilitzar l'eina osm2pgsql per convertir aquest fitxer a sentències SQL(veure Fig 4.26) per a PostgreSQL.

```
./osm2pgsql -m -Uambosm -W -H localhost -P5434 -dmapnikdata -S ./default.style canvis.osm
```

Fig 4.26. Comanda de creació del fitxer .osm de canvis.

Quan es produeix l'actualització de les dades sobre la BD PostgreSQL amb Postgis, es realitza el mateix procés de renderització amb Mapnik i fabrica les tesel·les que mostrarà el servidor web. El procés es repeteix cada vegada que un usuari retorna uns canvis a la Base de dades OSM.

5. Instal·lació i configuració del PC dedicat a tasques OSM

En aquest apartat expliquem a grans trets quins són els components que instal·lem al servidor de AMB i els requeriments que es necessiten per a realitzar amb èxit la seva instal·lació.

Per fer-nos una idea dels components podem observar l'arquitectura del servidor amb els components que utilitzarem (veure Fig 5.1).

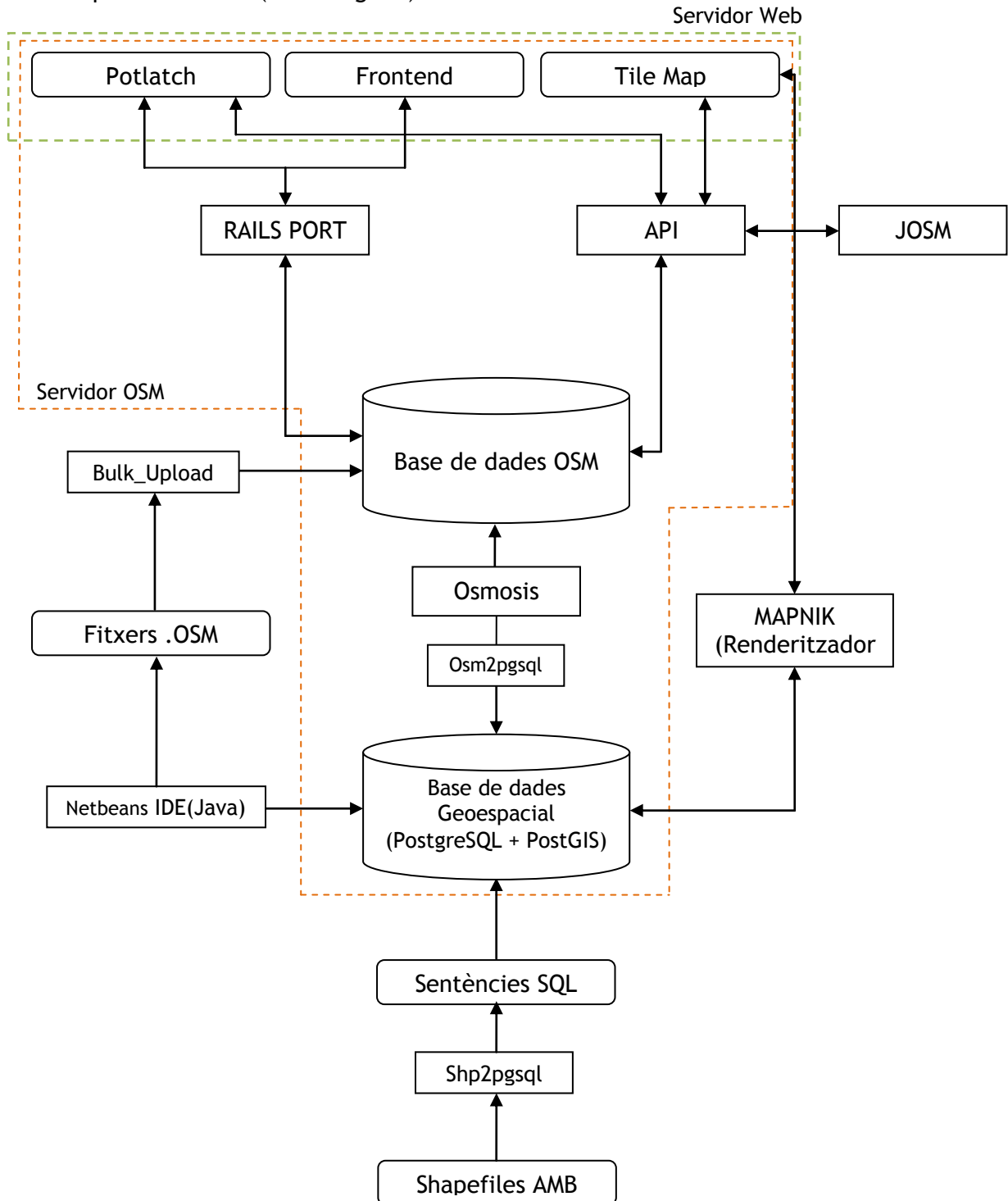


Fig 5.1. Estructura del servidor OSM a AMB.

5.1. Servidor OSM

- Requeriments

La instal·lació i posta a punt d'un servidor tipus OpenStreetMap necessita de:

1. Un ordinador tipus PC (tant pot ser real com virtualitzat) o Mac.
2. Sistema Operatiu tipus Unix, que pot ser Linux o OS-X
3. Sistema gestor de bases de dades PostgreSQL o MySQL
4. Sistema de fitxers no FAT ni FAT32

Les proves s'han realitzat, i per tant recomanem, amb les següents especificacions :

- Sistema Operatiu Ubuntu 8.04LTS Server
- Sistema gestor de bases de dades PostgreSQL 8.3
- POSTGIS compatible amb la versió 8.3 de PGSQL

- Configuració dels RAILS PORT

Rails Port és el conjunt de programari que crea la interfície web del sistema, i la versió actual del codi dels servidor OSM-API. Està programat en *Ruby on Rails*, un framework de desenvolupament web, escrit en *Ruby*.

El seu objectiu és fer del desenvolupament de la base de dades impulsada per aplicacions basades en web més ràpid i més productiu.

Es parteix de la base que s'ha instal·lat correctament un servidor linux (Ubuntu en aquest cas) i un sistema gestor de base de dades (PostgreSQL en aquest cas) i que funcionen correctament.

Gems es un paquet de llibreries per treballar amb Rails Port i està escrit en Ruby.

Suposem que l'usuari amb que es realitza la instal·lació no té privilegis, però pertany al grup d'administradors amb permissos i per tant pot executar comandes com a root.

- Dependències

Abans d'instal·lar Rails Port, hem de configurar Ruby, per això realitzem unes passes :

1. Afegir les dependències de Ruby.
2. Es suposa que en la instal·lació del servidor ja hi ha els paquets d'Apache i PostgreSQL, en cas contrari cal instal·lar-los.
3. Instal·lar les gems de Ruby, i actualitzar-les.
4. Configuració del sistema : Modificació del fitxer de configuració de les llibreries de gems.
5. Instal·lar les dependències de gems i finalment rails.

- **Instal·lació de RAILS PORT**

Fet això ja es pot instal·lar el rails port de OpenStreetMap.

Hem utilitzat el directori `/home/ambosm/osm` com a directori on hi ha tot el referent a OpenStreetMap i dins d'aquest `/source` pel codi font, `/database` per les bases de dades i `/logs` per als logs

Ara ja tenim tot el codi necessari per a crear el nostre servidor OSM.

El següent pas és :

- Creació de les bases de dades necessàries per la gestió del servidor.
- Crearem un clúster de PostgreSQL per no interferir amb res. Si la instal·lació és neta, la creació del clúster configurarà automàticament el port 5433 com a punt d'entrada per les bases de dades d'OSM. En cas que es vulgui o que ja hi hagin d'altres clústers aquest port es pot modificar.
- Creació d'usuari i assignació de password (el password pot ser modificat).

Per poder treballar amb Rails Port cal configurar en aquest les bases de dades que hem creat, amb el fitxer de configuració (es pot agafar com a base l'exemple que ja porta el rails port).

En el nostre cas el fitxer es troba a `/home/ambosm/osm/source/rails_port/config` i l'anomenem `database.yml`

Finalment :

- Instal·lem els requeriments del Rails Port.
- Migrem la informació necessària per la base de dades.

I amb això ja tenim instal·lat tot el sistema d'OpenStreetMap al nostre servidor.

5.2. Renderitzador OSM - Mapnik

- **Instal·lació de osm2pgsql**

osm2pgsql és una aplicació que permet la importació de dades en format osm a una base de dades PostgreSQL amb PostGIS amb els formats necessaris per a que puguin ser dibuixats per el renderitzador Mapnik

Es parteix de la mateixa base instal·lada esmentada anteriorment :

- Servidor linux (Ubuntu)
- SGBD (PostgreSQL en aquest cas)

Fet això, instal·lem les llibreries de osm2pgsql amb les seves dependències.

Utilitzem el mateix directori */home/ambosm/osm* com a directori on hi ha tot el referent a OpenStreetMap i dins d'aquest */source* pel codi font, */database* per les bases de dades i */logs* per als logs i creant una carpeta específica per al renderitzador.

- **Instal·lació de Mapnik**

Per a omplir el maptile necessitem un renderitzador de mapes gràfic, en aquest servidor instal·lem Mapnik.

Instal·lem seves llibreries i dependències.

- **Eines OSM per a Mapnik**

Per que el renderitzador funcioni correctament tenim que instal·lar les principals eines i les dades que conformen el renderitzador :

- Estils de dibuix
- Simbologia
- Catalogació
- Dades inicials (world boundaries)

Les eines principals s'instal·len al directori */home/ambosm/osm/source/mapnik*

- **Configuració de PostgreSQL**

En l'apartat anterior es va configurar un cluster en el port 5433 per a la base de dades geospacial PostgreSQL, el que fem a continuació es crear un nou cluster al port 5434 per gestionar les tiles creades per Mapnik i no interferir amb altres dades de PostGIS.

Com a nova base de dades també creem un usuari i una contrasenya que es podrà modificar en el futur.

- **Instal·lació i Configuració de Apache2 + mod_tiles**

Instal·lem el servidor web apache2, es posa conjuntament amb el servidor el mòdul mod_tiles, que ens servirà especialment per gestionar les tesel·les creades amb el renderitzador Mapnik.

5.3. Instal·lació del servidor a AMB

La instal·lació del servidor a AMB es va realitzar seguint tots els processos esmenats anteriorment, adjuntant tota la documentació del projecte per a futures consultes i renovacions de les bases de dades.

Una vegada posades totes les eines necessàries per crear el servidor OSM, s'estableix el PC dedicat a aquesta tasca en un sala on romandrà encès i amb disponibilitat de connexió remota per connectar-se des de qualsevol ordinador del departament de Cartografia.

Es va establir una connexió remota amb el servidor a partir d'una direcció ip amb una aplicació de connexió remota (WinVNC), on podem accedir al sistema en una finestra des de Windows (veure Fig 5.2).

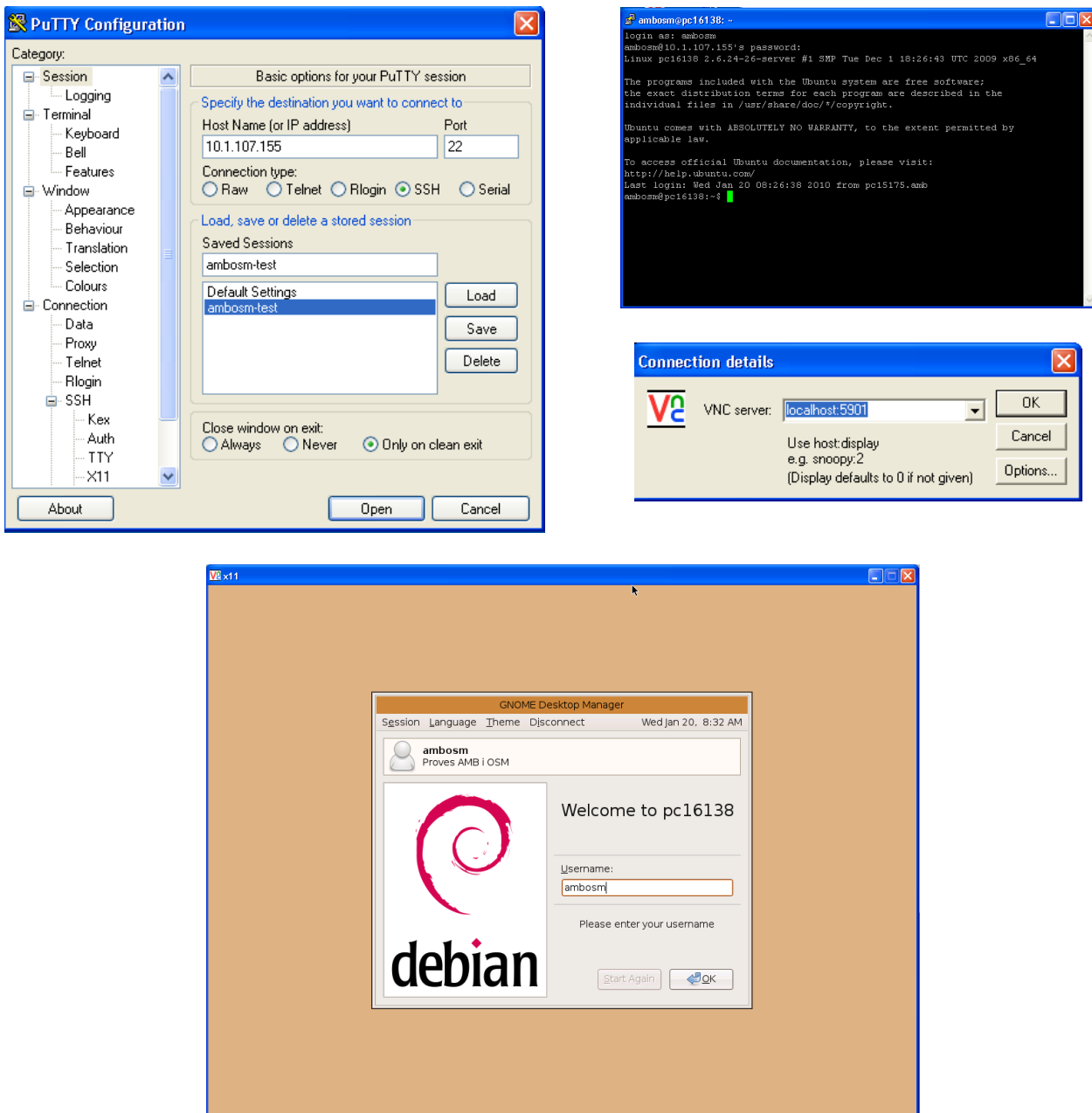


Fig 5.2. Finestra de interfície de connexió remota (WinVNC).

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)
Projecte final del Màster en Tecnologies de la Informació Geogràfica (11^a Edició)

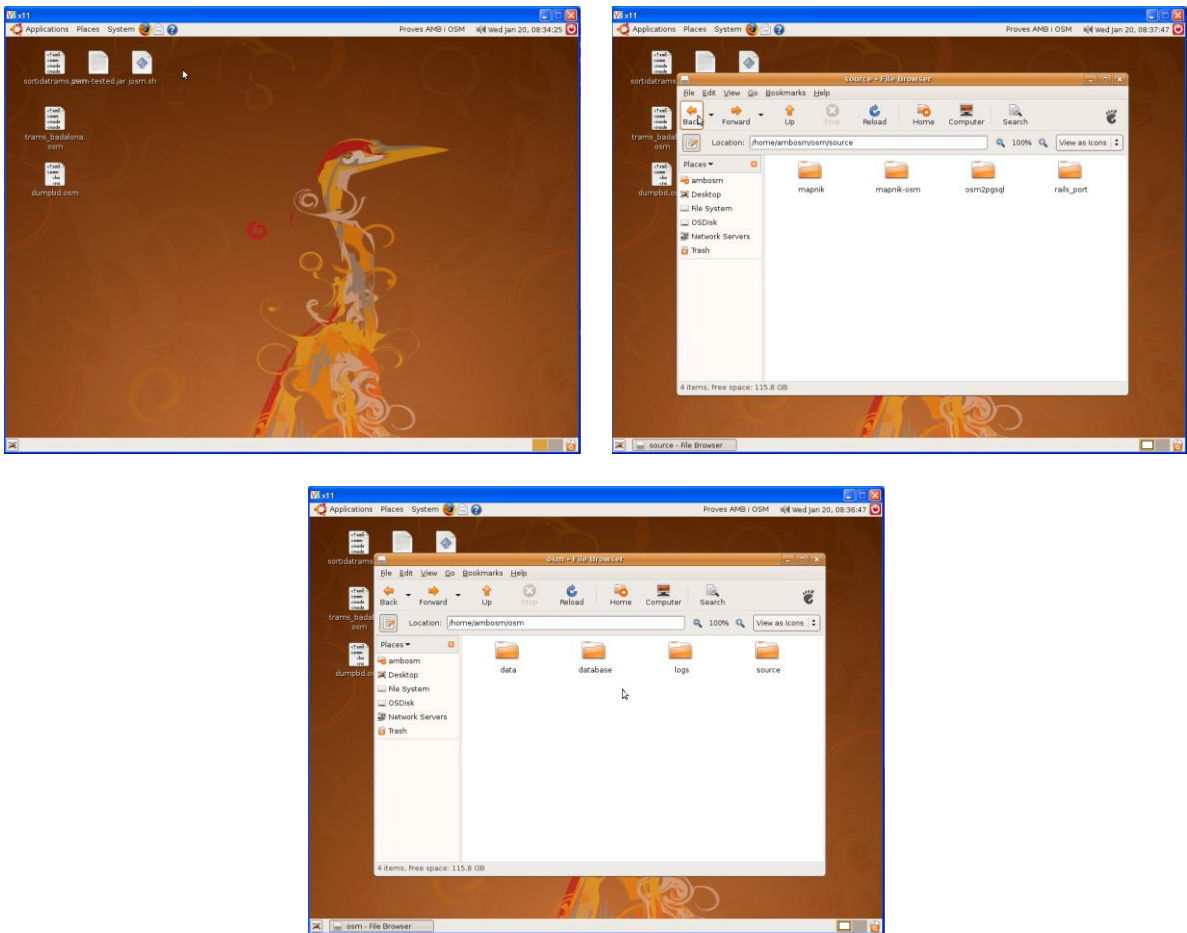


Fig 5.3. Interfície de Ubuntu 8.04TLS Server.

6. Presentació de resultats. Interfície Servidor web

6.1. Funcionalitat de les edicions

Una vegada acomplert els objectius, el servidor de dades cartogràfiques AMB-OSM, ens permetrà treballar amb les mateixes eines que conté el projecte OpenStreetMap en la seva versió mundial però aplicat al marc municipal de l'Àrea Metropolitana.

Es contempen diferents funcions que és volen realitzar a partir de la migració realitzada :

- **Aplicar una estandardització de la base cartogràfica per a tots els municipis**

Visualitzant els estils i la simbologia de la cartografia que ofereix OSM (veure Fig 6.1) i la possibilitat de modificar-les amb diferents eines (Editors online, openlayers,..) els permetrà adequar un estàndard d'estils propis que crearà una cartografia única i personal de l'Àrea Metropolitana de Barcelona basat en aquest entorn.

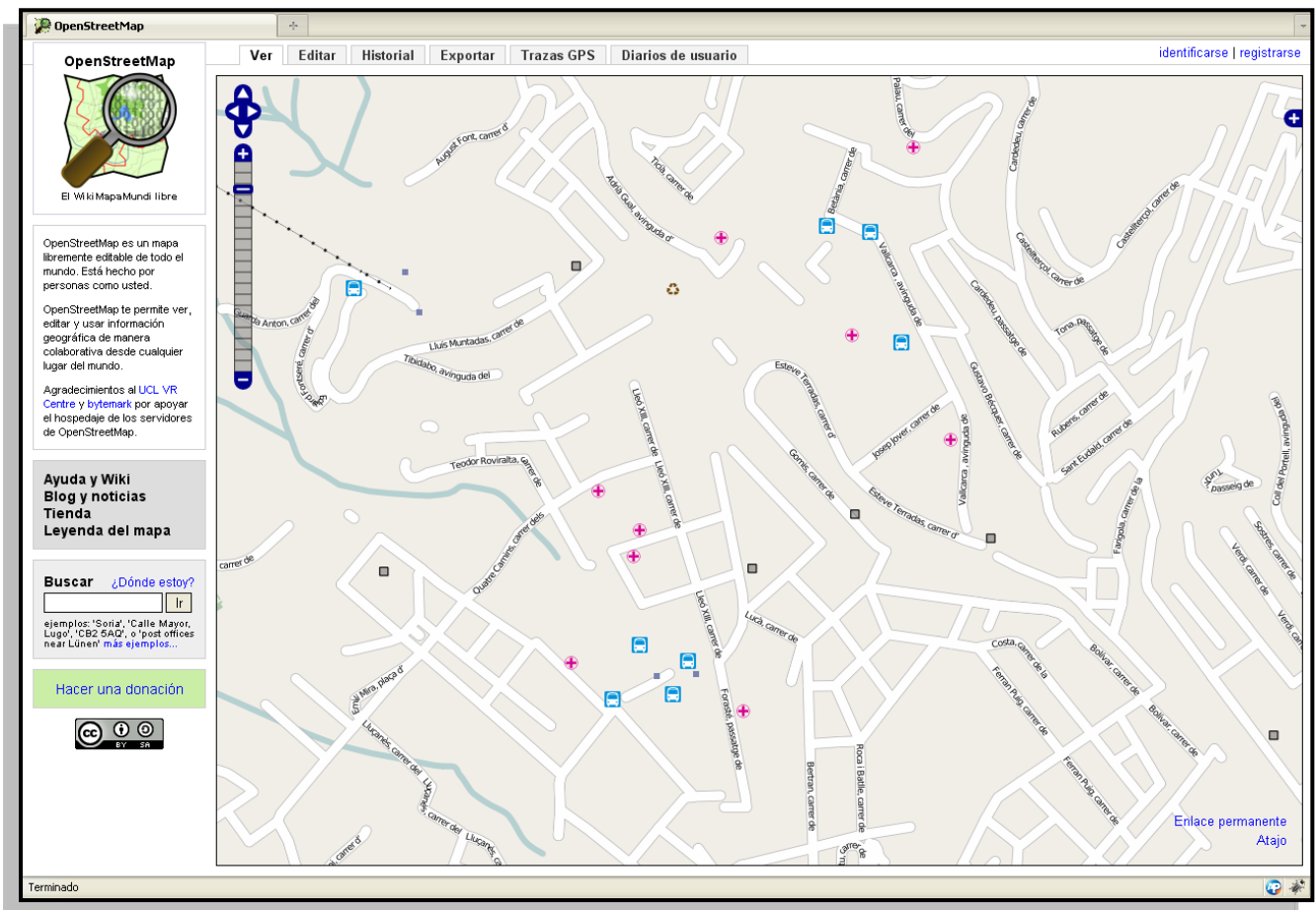


Fig 6.1. Estils i simbologia en el servidor AMB-OSM.

- Permetrà una possible edició distribuïda de la guia de carrers per parts dels Ajuntaments**

OSM permet crear diferents usuaris que poden modificar les dades fàcilment amb Potlatch, i amb més precisió amb JOSM, així pot comportar la creació d'usuaris en cada ajuntament de cada municipi que pot aportar actualitzacions de la cartografia de forma ràpida i senzilla.

Aquest mètode ofereix una base cartogràfica única i col·laborativa on la realització de la mateixa no es feina de una o vàries persones, si no de moltes que aporten nova informació.

El control de qualitat de la toponímia es converteix també en un valor afegit, ja que obtenim de primera mà la informació dels topònims i ofereixen una veracitat altament fiable, sense entrar en la possibilitat de que en algun moment es pugui obrir a gent que no pertany a la administració pública i pugui aportar també la seva informació.

Una de les eines que es podent aprofitar es la possibilitat de importar i maquetar la cartografia de OSM per a edicions en paper(veure Fig 6.2), amb el propi JOSM es pot importar a formats gràfics per realitzar tractaments de les imatges amb qualsevol editor gràfic com freehand o photosop (png, svg,etc...).

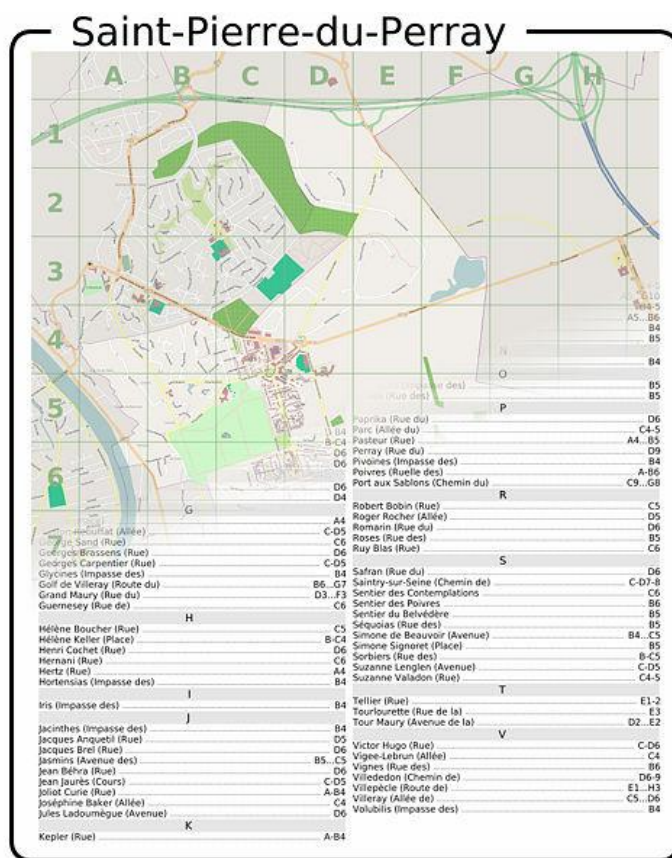


Fig 6.2. Edició maquetada en paper de un mapa OSM.

També es disposa de aplicacions (GNUPlot,OSMBook,..) que et maqueten la zona que desitgis del mapa, creant una edició pròpia i personal de les teves dades.

• **Adequació i canalització de la informació que poden aportar els usuaris**

El mètode que utilitza OSM per revisar la cartografia que aporten els usuaris, es registrarà cadascuna de les edicions que han realitzat (veure Fig 6.3), mostrant en la pestanya historial :

- Identificador de l'edició
- Nom de l'usuari
- Data d'edició
- Comentari
- Àrea editada

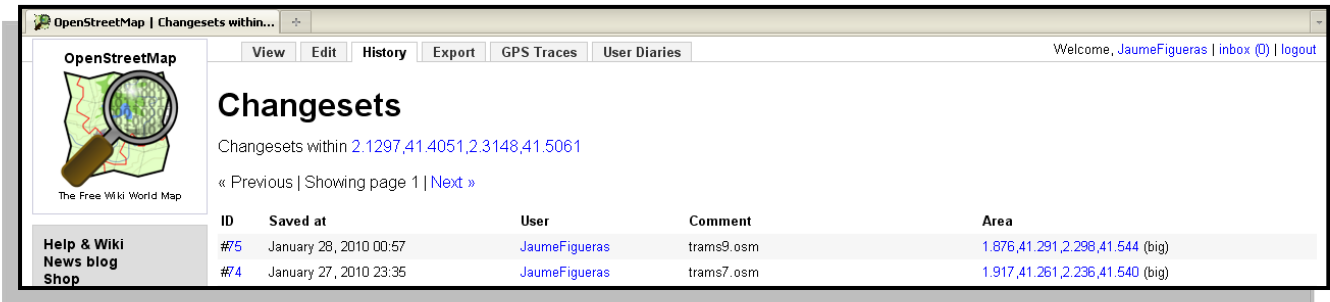


Fig 6.3. Historial d'edicions en el servidor AMB-OSM.

Prement l'identificador ens mostra amb més detall (veure Fig 6.4) el que s'ha modificat, amb un petit mapa on es veu la zona editada on mostra :

- Mètode d'edició (Potlatch, JOSM, ...)
- Elements modificats (node o way)
- Zona de modificació (bounding box)

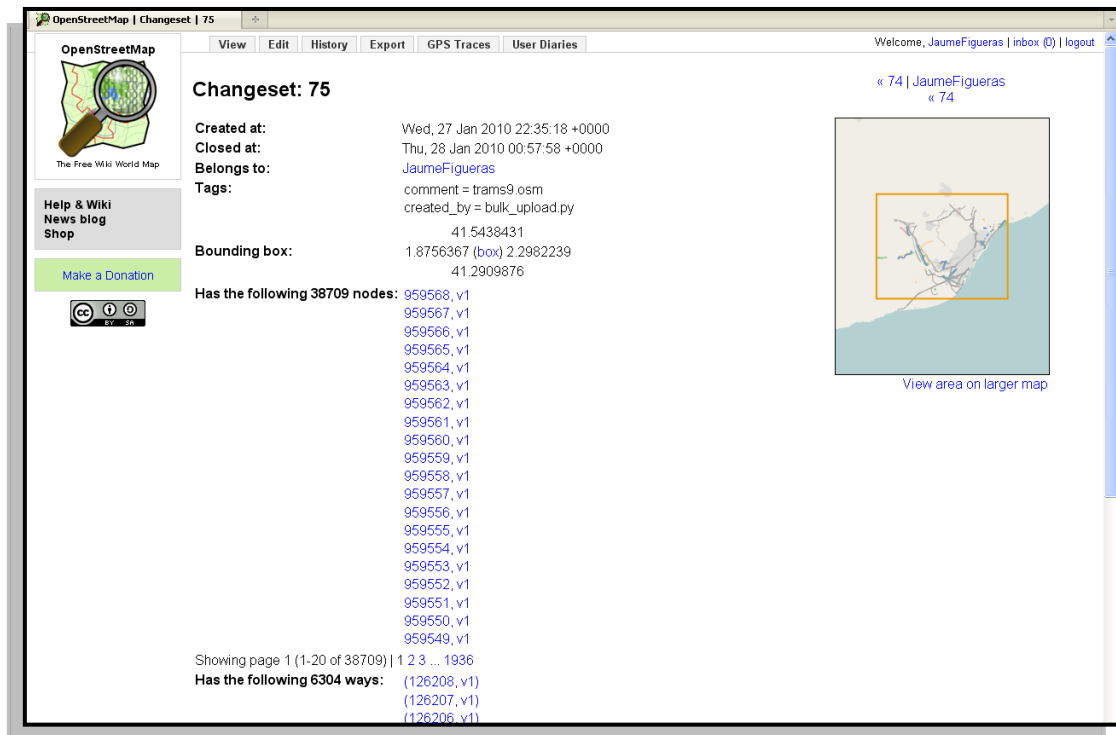


Fig 6.4. Historial d'edicions específic en el servidor AMB-OSM.

Amb aquest mètode de control col·lectiu no hi ha un sol moderador que controli les edicions que es realitzen, hi ha tants com usuaris puguin estar registrats i permet aportar idees i conceptes de molts usuaris.

És possible baixar les dades d'aquests canvis en format OSM , per treballar en altres editors fora de línia i després pujar els canvis al servidor.

- **Detecció de errors, incoherències i millora de la guia de carrers**

Com hem vist en la documentació es poden afegir moltes classes d'etiquetes als elements de la cartografia i això permet trobar els errors que no es troben normalment.

Amb les *relacions*, podem incorporar als elements moltes formes de representar la informació que no es pot representar normalment en els mapes, per exemple les relacions de restriccions de gir en els carrers així com les direccions dels carrers, unificar les línies de transport amb una simbologia pròpia, delimitar temàticament zones administratives o d'esbarjo.

Una de les possibilitats de millora es la *geocodificació* dels números policia (veure Fig 6.5), on els edificis estan georeferenciats amb els vials dels carrers, com a tal poden estar relacionats amb el seu codi postal però posseeixen la seva pròpia geolocalització.

Ofereix moltes possibilitats per a referenciar aquests elements als vials, podent *interpolar* els números dels carrers i assignar un seguit d'ells a un mateix edifici.

http://wiki.openstreetmap.org/wiki/ES:N%C3%B0Ameros_de_polic%C3%ADa

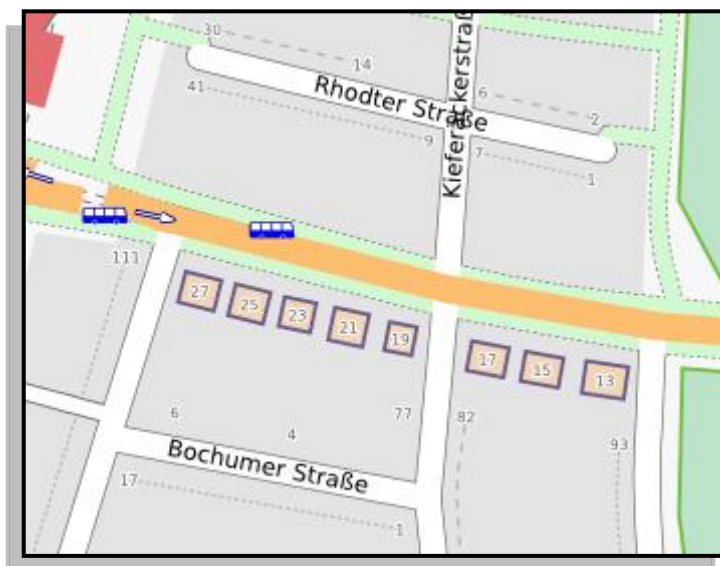


Fig 6.5. Interpolació de Números Policia.

Totes aquestes eines permeten la creació de nova cartografia on hi ha molta més informació unificada i homogènia.

6.2. Renderització de mapes AMB a al servidor OSM

En el següent apartat, mostrem diferents captures dels resultats gràfics que s'han obtingut al servidor de AMB-OSM.

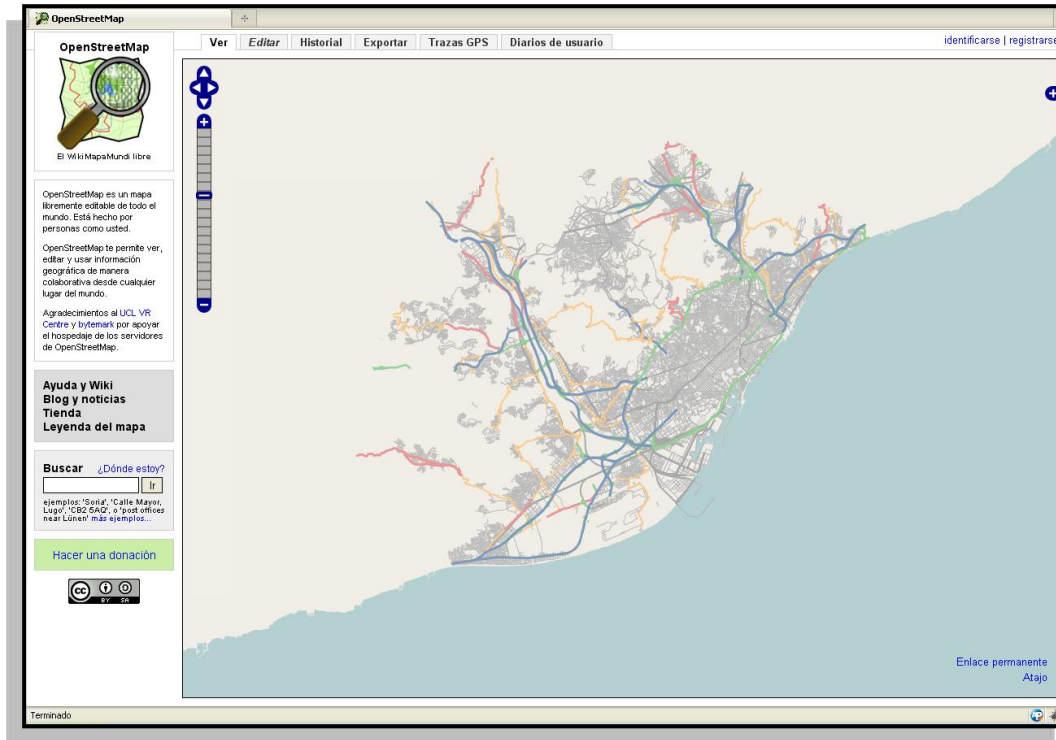


Fig 6.6. Interficie web del servidor AMB-OSM.

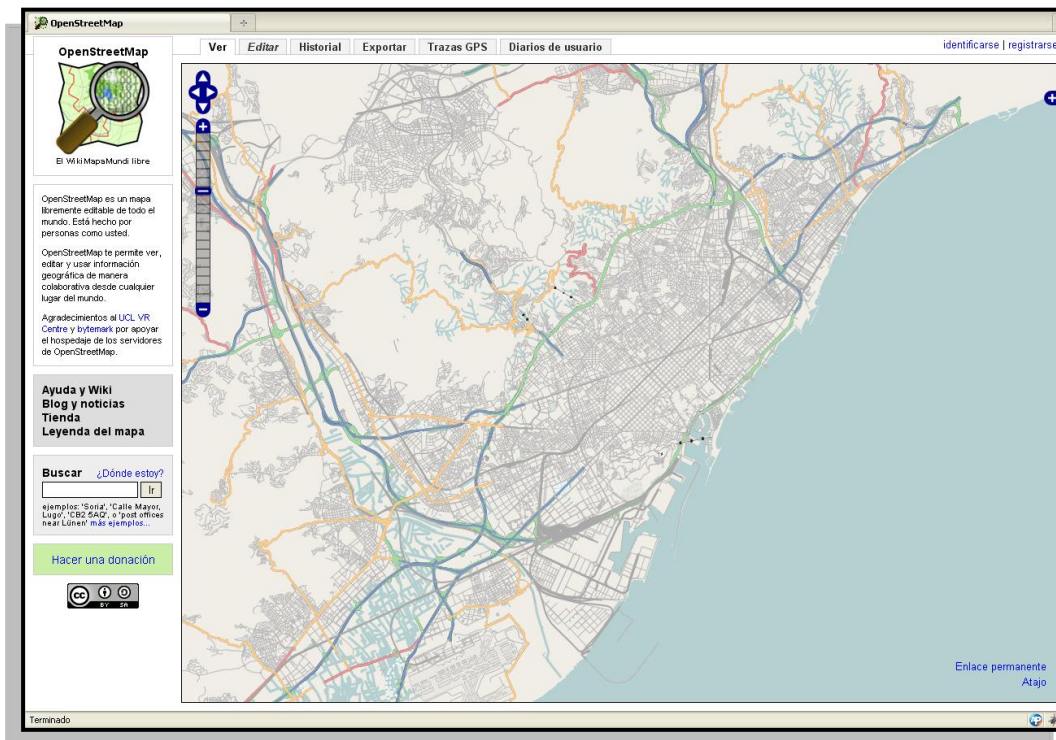


Fig 6.7. Vista general dels trams complerts de l'AMB.

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)
Projecte final del Màster en Tecnologies de la Informació Geogràfica (11ª Edició)

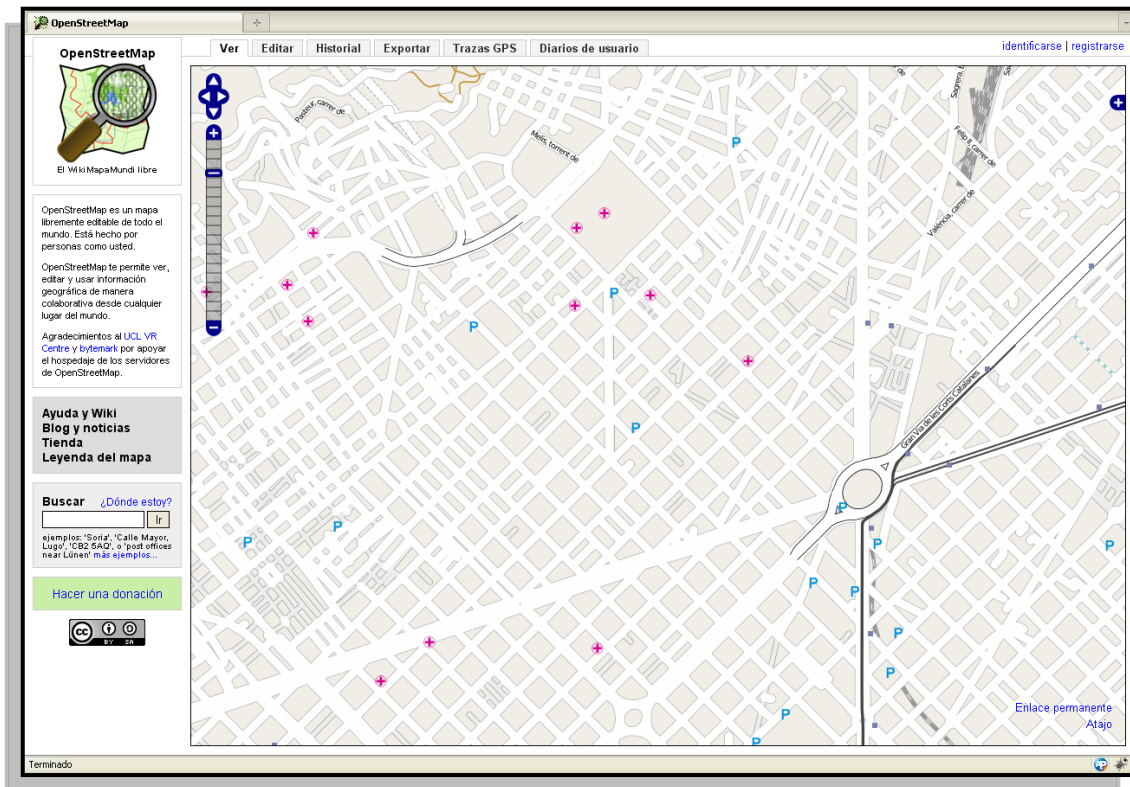


Fig 6.8. Trams de carrers de Barcelona amb simbologia puntual.

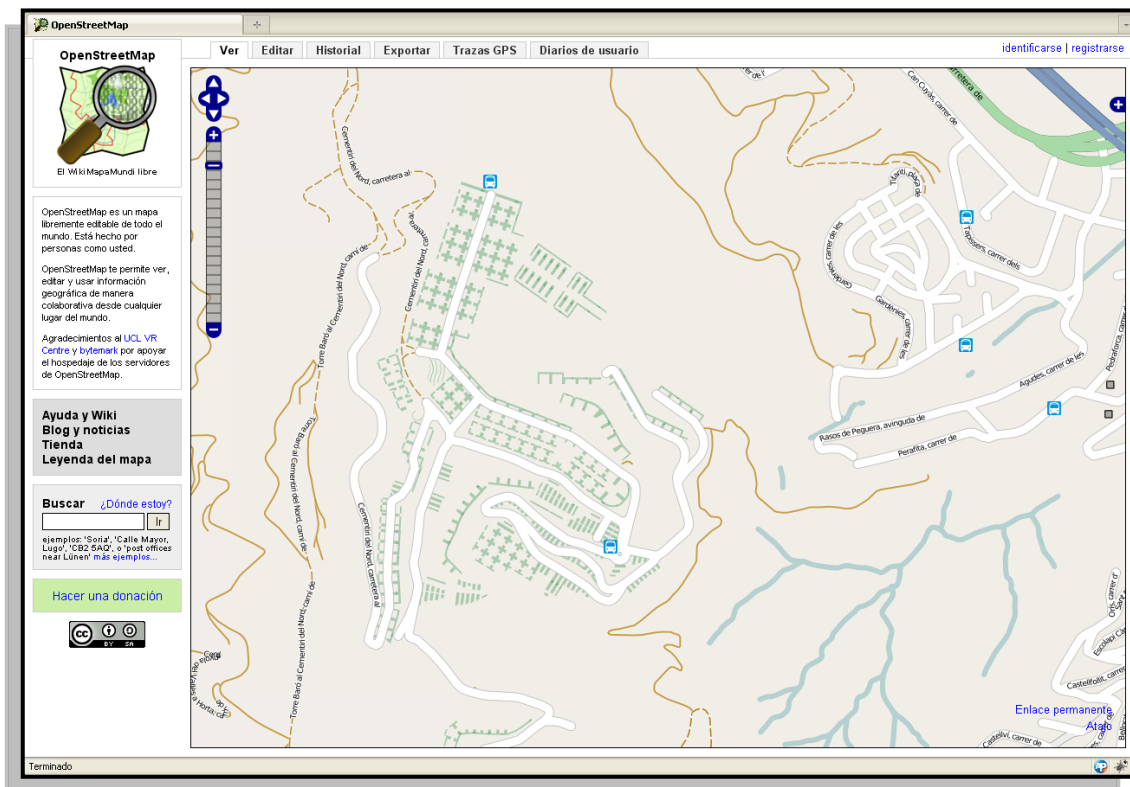


Fig 6.9. Representació de camins i corriols.

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)
 Projecte final del Màster en Tecnologies de la Informació Geogràfica (11ª Edició)

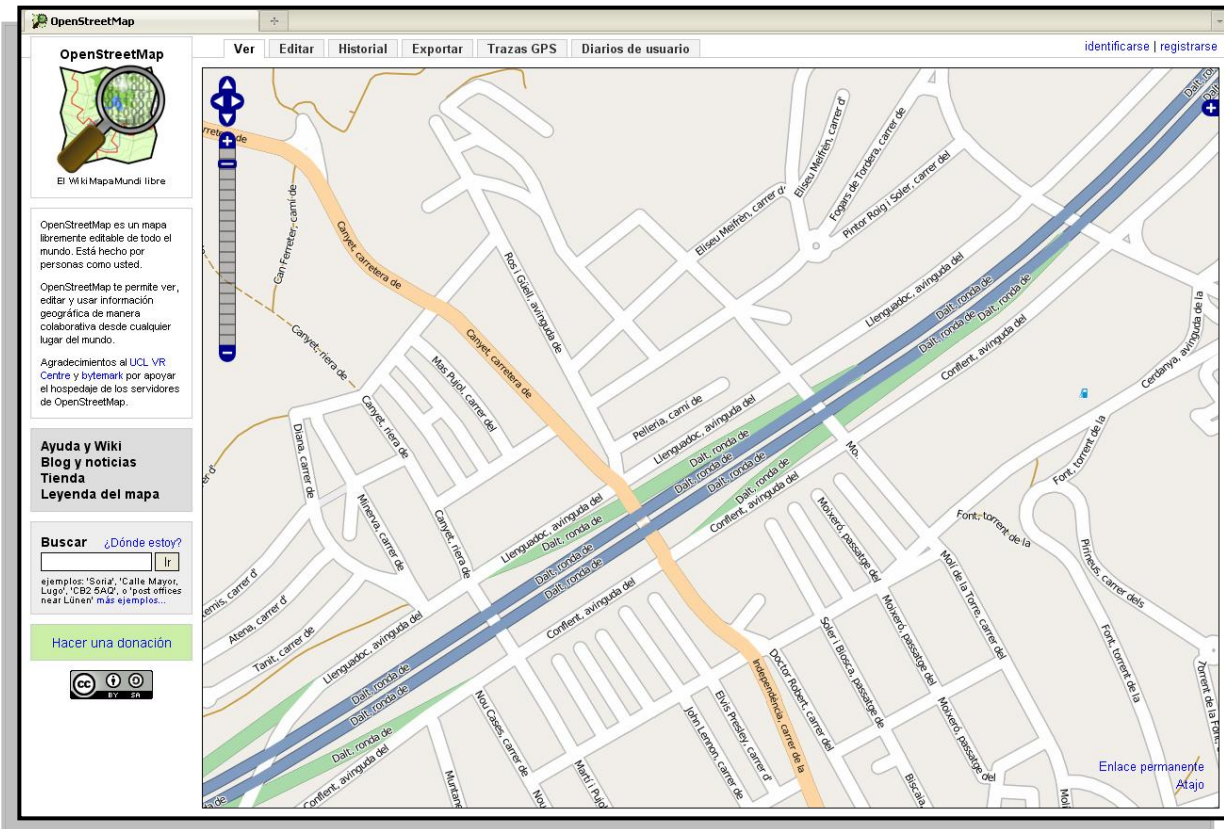


Fig 6.10. Jerarqització de les xarxes de comunicació.

7. Conclusions

7.1. Valoració del treball

Des d'un principi es va tenir en compte que era un projecte pilot, on es farien les primeres proves per migrar la informació cartogràfica de l'AMB a un format totalment diferent, sempre s'ha tingut en compte que amb aquestes proves es podria definir el que es pot realitzar en un futur i quines són les funcionalitats que es poden aprofitar millor.

Una vegada hem carregat les dades al servidor i podem editar-les i actualitzar-les al mateix temps podem establir que l'objectiu de migració i sincronització s'ha realitzat amb èxit.

Els principals problemes que hem tingut al moment de migrar la informació, són la gran mida física dels fitxers .shp un cop transformats a fitxers osm, amb una mida tant gran es fa difícil la carrega de dades als servidors geoespacionals considerant també que només s'ha de fer una vegada en tot el projecte.

Les reprojeccions dels datums, ha estat una de les tasques més complicades. En el moment de reprojectar del datum ED50 al datum WGS84 ens donava un error de localització d'uns 2-3 metres, que ha endarrerit una mica la marxa del projecte.

Les aplicacions creades aporten una nova forma d'importar dades a OSM, encara que hi ha diferents aplicacions per aquesta tasca, la que hem realitzat s'ha començat des de zero i això li dona molta importància a la personalització del sistema.

Per tal de que un futur es pugui utilitzar aquest sistema d'actualització de les dades en els municipis, s'han creat dos manuals dels diferents editors que trobem a OSM, és un manual bàsic però suficient per donar les eines al personal dels municipis per poder començar un projecte col·laboratiu com es proposa aquí.

S'ha de tenir en compte que un cop visualitzades les dades, trobem un resultat satisfactori, però en un futur es pot crear una tipologia de simbologia com a l'AMB per fer-ho el més semblant possible a la guia de carrers. Tanmateix es pot crear una base de dades amb molta més informació temàtica ja que la majoria de simbologia, només està representada per l' icona.

En resum ha estat un projecte en el que s'han assolit els objectius que s'havien demanat i des de l'Àrea Metropolitana de Barcelona ja es comença a veure les possibles sortides que li poden donar a aquesta eina, que promet continuïtat en un futur molt pròxim.

7.2. Valoració personal

En el moment de començar un projecte, et crees unes expectatives, uns dubtes i unes pors que poc a poc es van esvaïnt en el transcurs de la realització del mateix, el meu dubte principal estava sempre present en la mida del projecte en qüestió, hi havia una quantitat de treball molt gran i en uns 4 mesos escassos semblava que faríem curt i no arribaríem a la data.

Deixant de banda el temps, la meua tasca a la catalogació de la cartografia per a importar-la a PostgreSQL amb POSTGIS ha estat relativament senzilla fins que ens varem trobar amb els problemes de la reprojecció del datum, però que varem solucionar gràcies a les persones que hi havia darrera. La continuïtat de la feina a partir de la fase de catalogació, l'anàvem solucionant gràcies a que hi havia tres persones cercant quina era la solució als problemes que anaven sorgint.

A banda de tots els coneixements tècnics que hem après, m'ha fet intuir quin seria el grup ideal per realitzar Sistemes d'Informació Geogràfica en entorns web, un grup que es complementi entre els diferents tècnics que hi han estat presents en el projecte, un tècnic en SIG i Cartografia o geografia, un tècnic en informàtica de sistemes i un programador amb molta perspectiva i idees per crear noves eines.

La satisfacció més gran ha estat, a part del moment de començar la programació dels conversors de les dades d'AMB a OSM i la posterior visualització de les dades, es la instal·lació del servidor de dades OSM on m'ha permès aprendre molt i en profunditat el món del programari lliure i codi obert del programari que existeix actualment per la realització d'entorns SIG lliures en internet. T'obre les portes a altre programari que és possible utilitzar, a banda del programari comercial tan limitat al que ens tenen acostumats.

En resum ha estat un inici personal d'entrada en el món del SIG lliure bastant satisfactori i que m'ha deixat amb les ganes de saber més i obtenir més coneixements de les eines OSM i en definitiva de les eines de SIG lliure.

Per finalitzar convido a tots aquells que llegeixen aquest document a que es passin per la plana web de OpenStreetMap i comprovin la potencialitat de l'eina que tindran davant i comprovaran que no fa falta esser un tècnic especialitzat en cartografia per aportar informació a aquest sistema que es podria dir que fa molt poc temps que està en marxa.

8. Referències bibliogràfiques/web

- Referències bibliogràfiques

Carmen Muñoz-Cruzado García (2006) : “Instalación de PostgreSQL con la Extensión PostGIS como base de Datos Espacial”. Universidad Politécnica de Madrid.

Andres Herrera (2005) : “PostgreSQL. Geo...Sin Enredos”

Manuel Martín Martín : “Manual PostGIS”. Traducció del manual de Postgis de Paul Ramsey

- Referències web

Plana web de OpenStreetMap

<http://www.openstreetmap.org/>

Wikipedia de OpenStreetMap

http://wiki.openstreetmap.org/wiki/ES:Main_Page?uselang=es
http://wiki.openstreetmap.org/wiki/ES:Beginners_Guide_1.3.1

Map Features - OpenStreetMap

http://wiki.openstreetmap.org/wiki/Map_Features

PostgreSQL - Sistema Gestor de Base de Dades

<http://www.postgresql.org/>
<http://wiki.postgresql.org/wiki/Espa%C3%B1ol>

POSTGIS - Complement Geoespacial de PostgreSQL

<http://postgis.refractions.net/>
<http://wiki.openstreetmap.org/wiki/PostGIS>
<http://trac.osgeo.org/postgis/wiki/UsersWikiMain>

osm2pgsql - Importació de dades OSM a PostgreSQL

<http://wiki.openstreetmap.org/wiki/Osm2pgsql>

Osmosis - Eina d'actualització de dades OSM

<http://wiki.openstreetmap.org/wiki/Osmosis>

osmChange - Format de fitxers OSM a Osmosis

<http://wiki.openstreetmap.org/wiki/OsmChange>

Bulk_upload - Eina Java de bolcat de dades massives

http://wiki.openstreetmap.org/wiki/Bulk_upload.pl

Editor offline JOSM

<http://wiki.openstreetmap.org/wiki/ES:JOSM>

Editor online Potlatch

<http://wiki.openstreetmap.org/wiki/ES:Potlatch>

Editor SIG Quantum GIS

<http://www.qgis.org/>

Netbeans - Entorn de programació Java

<http://netbeans.org/>

<http://es.wikipedia.org/wiki/Netbeans>

Tutorial XSLT - Traducció de XML a OSM

<http://w3schools.com/xpath/default.asp>

9. Annexos

Annex I

Diccionari AMB-OSM (Map Features específic)



• **Introducció**

Per realitzar la traducció de la cartografia dels 36 municipis al format OSM, hem fet diferents taules de traducció per a les diferents carpetes on guardem els elements.

- **01_Text_Transport** : Mitjans de comunicació principals
- **02_simbologia** : Directori de serveis
- **03_Poligons_lineas** : Zones urbanes, equipaments, activitats econòmiques, parcs i jardins

L'estructura del diccionari és la mateixa en totes les taules de traducció.

CODI	Descripció
-----	-----

KEY	VALUE	Element
----	-----	

D'una banda tenim la informació de les taules, on el **codi** es el número de la taula que tenim en les carpetes i a la base de dades PostGIS, la **descripció**, ens indica el contingut temàtic de la taula.

La **Key** i el **Value** ens indica els valors que pren aquella taula en OSM, es poden tenir més d'una etiqueta, i la key de layer ens indiquen el nivell de visualització de les capes, on adquireixen un valor entre -5 i 5.

Finalment com a camp d'informació , **l'element** ens indica el tipus de dada primitiva que convertirem la taula, Node, way o way tancada.

• **Trams**




















La taula de traducció de trams, és lleugerament diferent a les altres. Dintre de la taula trams tenim un codi que ens identifica el tipus de via del tram, així doncs en aquesta taula el codi no ens indica el nom de la taula, ens separa els trams en diferents tipus de via.

Degut al nivell d'alguns tipus de via, com ara els passos elevats o els túnels, tindrem que anar jugant amb la value de la key layer, deixant en un nivell més alt o a un nivell més baix dependent del tipus de via.

CODI	Descripció
0	Vial Indefinit
7500	Vial carrers
7501	Vial carrers tunel
7502	Vial carrers construccio
7503	Vial carrers peatonals

KEY	VALUE	Element
highway layer	residential 0	
highway layer	residential 0	
highway tunnel layer	residential yes -1	
highway layer	construction 0	
highway layer	pedestrian 0	

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)
Projecte final del Màster en Tecnologies de la Informació Geogràfica (11ª Edició)

7504	Vial pas elevat	highway bridge layer	residential yes 1	
7506	Vial laterals	highway layer	residential 0	
7507	Vial carrers peatonals elevat	highway bridge layer	pedestrian yes 1	
7508	Vial carrers Impor	highway layer	residential 0	
7510	Vial autopista_autovia	highway layer	motorway 0	
7511	Vial autopista_autovia tunel	highway tunnel layer	motorway yes -1	
7512	Vial autopista_autovia Construccio	highway layer	construction 0	
7513	Vial rondes	highway layer	trunk 0	
7514	Vial rondes tunel	highway tunnel layer	trunk yes -1	
7515	Vial rondes elevat	highway bridge layer	trunk yes 1	
7517	Vial autopista_autovia elevat	highway bridge layer	motorway yes 1	
7530	Vial carretera nacional	highway layer	trunk 0	
7540	Vial carretera comarcal	highway layer	primary 0	
7550	Vial carretera local	highway layer	secondary 0	
7552	Vial carretera local construccio	highway layer	construction 0	
7560	Vial carretera enllaç	highway	trunk_link	
7561	Vial carretera enllaç tunel	highway tunnel	trunk_link yes	
7562	Vial carretera enllaç construccio	highway layer	construction 0	
7563	Vial carretera enllaç elevat	highway bridge layer	trunk_link yes 1	

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)
 Projecte final del Màster en Tecnologies de la Informació Geogràfica (11ª Edició)

7564	Vial ronda enllaç	highway layer	trunk_link 0	
7565	Vial ronda enllaç tunel	highway tunnel layer	trunk_link yes -1	
7570	Vial carretera Pista	highway tracktype layer	track grade2 0	
7580	Vial carretera Cami	highway tracktype layer	track grade1 0	
8169	Vial carrers	highway layer	residential 0	

Tots els vials tindran la Key de Name :
 i a value agafarem la concatenació dels camps
 TVIAL,XVIAL i NVIAL, es a dir el nom del carrer o vial.

• **Portals amb**

Camp	Descripció
------	------------












text	Numero del Portal
------	-------------------

KEY	VALUE	Element
-----	-------	---------

addr:housenumber	text	
------------------	------	---

• **01 text transport**

CODI	Descripció	Element
10010	aeroport	Point
10020	estacio de FGC	Point
10030	estacio de metro	Point
10040	estacio de RENFE	Point
10050	parada funicular	Point
10060	terminal d'autobus	Point
10070	parada d'autobus origen final	Point
10080	parada de tramvia	Point
10090	de tramvia blau	Point
10110	port comercial	Point
10120	port esportiu	Point

KEY	VALUE	Element
aeroway layer	aerodrom 2	
railway layer	halt 2	
railway layer	subway_entrance 2	
railway layer	halt 2	
aerialway layer	station 2	
amenity layer	bus_station 2	
amenity layer	bus_station 2	
railway layer	tram_stop 2	
railway layer	tram_stop 2	
waterway layer	mooring 2	
waterway layer	mooring 2	

• **02 simbologia**

CODI	Descripció	Element
21010	ajuntament	Point
21020	informacio	Point
21030	bombers	Point
21040	centre ITV	Point
21050	cossos de seguretat	Point
21060	edificis religiosos	Point
21070	gasolinera	Point
21080	mercat municipal	Point
21090	oficina correus	Point
21100	parquing municipal pagament	Point
21110	parquing municipal gratuït	Point
21120	punt reciclatge	Point
22010	hospital clinica	Point
22020	Centre atencio primaria	Point
23010	biblioteca	Point
23020	centre comercial	Point
23030	cinema	Point
23040	museu	Point

KEY	VALUE	Element
amenity layer	townhall 2	
tourism layer	information 2	
amenity layer	fire_station 2	
amb layer	amb:ITV 2	
amenity layer	police 2	
amenity layer	place_of_worship 2	
amenity layer	fuel 2	
shop layer	supermarket 2	
amenity layer	post_office 2	
amenity layer	parking 2	
amenity layer	parking 2	
amenity layer	recycling 2	
amenity layer	hospital 2	
amb layer	amb:CAP 2	
amenity layer	library 2	
shop	department_store mall 2	
amenity layer	cinema 2	
amenity layer	museum 2	





23050	patrimoni humanitat	Poly	historic layer	icon 2	
23060	teatre	Point	amenity layer	theatre 2	
24080	parada taxi	Point	amenity layer	taxi 2	

• **03 Poligons lineas**

CODI	Descripció	Element
4000	costa	Arc
4010	rius	Arc
4101	platja	Poly
4130	canal	Arc
4150	piscinas	Poly
4160	basses	Poly
4220	moll	Arc
4221	moll construcció	Arc
4330	cementiri	Poly
4400	aeroport	Poly
4500	ferrocarril RENFE	Arc
4510	ferrocarril FGC	Arc
4520	ferrocarril TGV	Arc
4530	teleferic	Arc
4540	tramvia	Arc
4800	bosc	Poly

KEY	VALUE	Element
natural layer	coastline -2	
waterway layer	river -3	
natural layer	beach -3	
waterway layer	canal -3	
leisure layer	water_park -2	
water_park layer	water_park -2	
waterway layer	dock 2	
waterway construction layer	dock 2	
landuse layer	cemetery -2	
aeroway layer	aerodrome -2	
railway layer	rail -1	
railway layer	narrow_gauge -1	
railway layer	rail -1	
aerialway layer	cable_car 1	
railway layer	tram 0	
natural landuse layer	wood forest -5	

Projecte pilot d'anàlisi de la coordinació entre la Guia de Carrers digital de l'AMB i el Mapa de Carrers d'OpenStreetMap(OSM)
Projecte final del Màster en Tecnologies de la Informació Geogràfica (11ª Edició)

4820	limit jardí	Poly	leisure layer	garden -3	
4830	conreu	Poly	landuse layer	farm -4	
4870	limt parc AMB	Poly	leisure layer	park -3	
4880	limit parc natural	Arc	leisure layer	nature_reserve 2	
4890	poligon parc natural	Poly	leisure layer	nature_reserve -5	
5210	alineacio edificada EC	Poly	landuse layer	residential -4	
5211	alineacio edificada EC urbanitzacio	Poly	landuse layer	residential -4	
5220	alineacio no edificada EC	Poly	landuse layer	comercial -3	
5250	edificacio interior EC	Poly	building layer	yes -2	
5410	alineacio edificada EI	Poly	landuse layer	industrial -4	
5420	alineacio no edificada EI	Poly	landuse layer	industrial -2	
5500	edif public singular	Poly	amenity layer	public_building -2	
5590	edificacio aillada	Poly	building layer	yes -2	
5600	zona esportiva	Arc	landuse layer	recreation_ground -2	
7020	limit municipal	Arc	boundary layer	administrative 2	

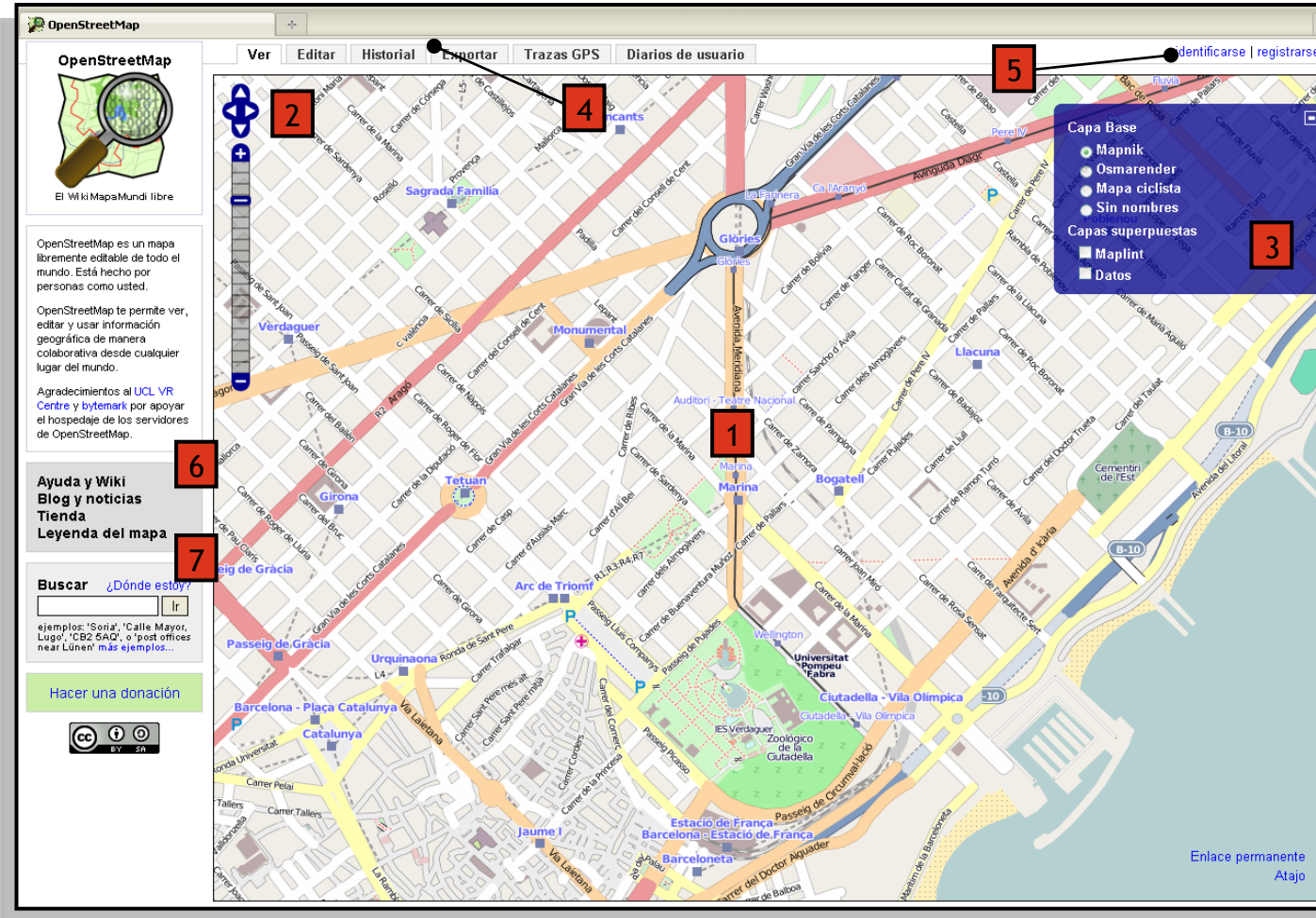
Annex II Manual bàsic de Potlatch



Índex

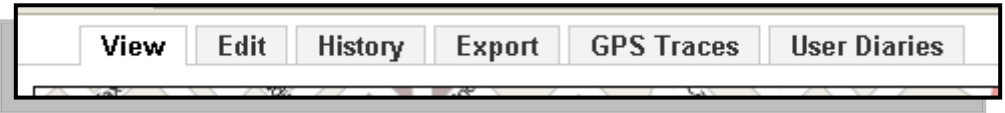
1. Vista general de la plana web de www.openstreetmap.org	93
• Funcions de les pestanyes	
2. Registre	98
3. Edició	99
• Eines de Potlatch	
• Desplaçar-se pel mapa	
• Edició	
• Etiquetes/Atributs	
• Altres	

1. Vista general de la plana web de www.openstreetmap.org



1. Vista General del Mapa
2. Eines de visualització
3. Slippy Map : Capes seleccionables
4. Pestanyes de funcions
5. Identificació d'usuari
6. Ajuda i Wiki
7. Cercador

• Funcions de les pestanyes



- **View** : ens mostra la visualització del mapa creat per OSM.
- **Edit** : Entres en el mode edició (cal estar registrat).
- **History** : Historial de les modificacions que han realitzat els usuaris.

OpenStreetMap | Changesets

View Edit History Export GPS Traces User Diaries

Welcome, DavidRG | [inbox \(0\)](#) | [logout](#)

Changesets

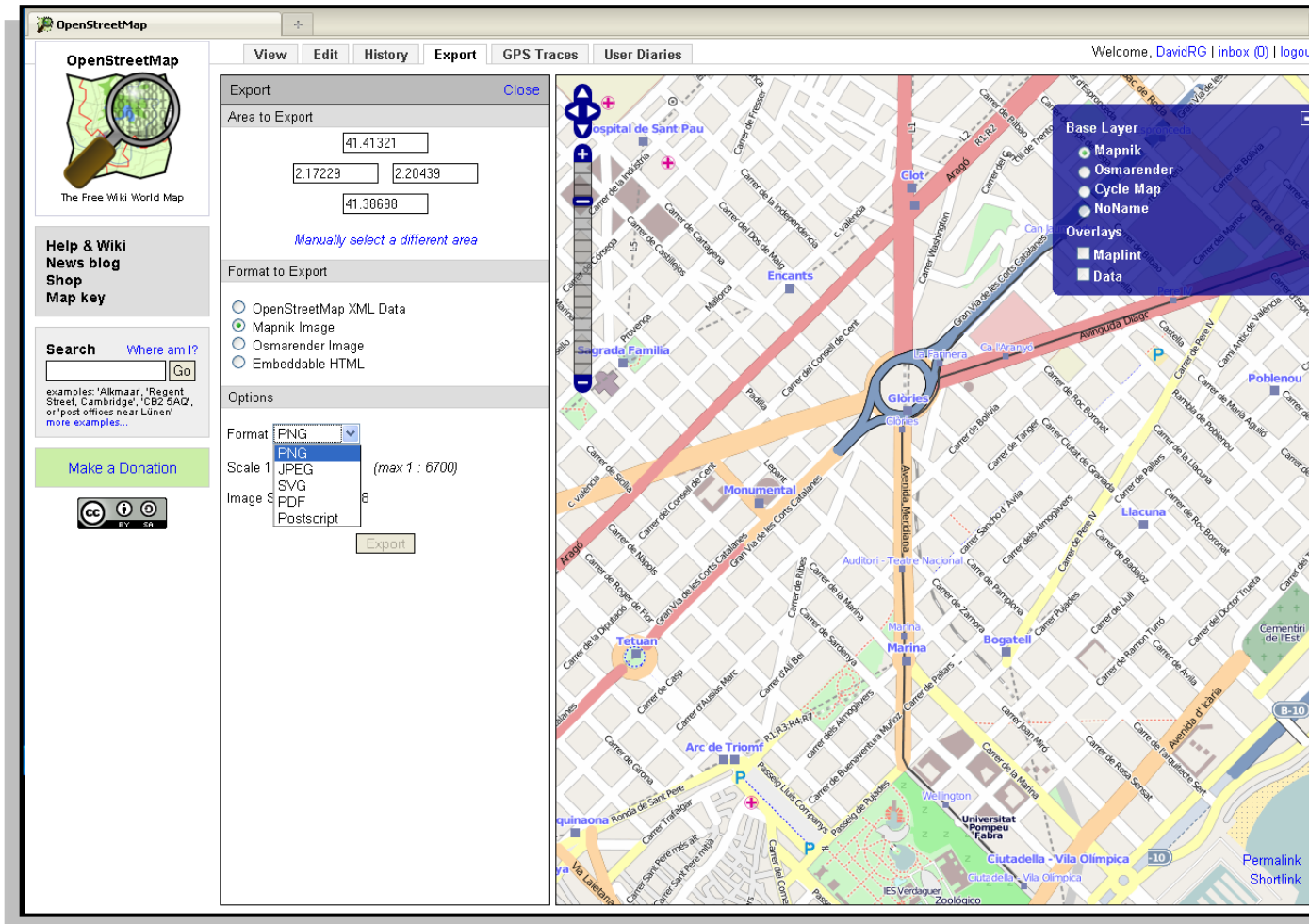
Recent changes

« Previous | Showing page 1 | Next »

ID	Saved at	User	Comment	Area
#3624102	(still editing)	hurdygurdyman	Fließrichtung berichtigt	7.835,48.306,7.940,48.372
#3624101	January 15, 2010 13:11	garl	BBOX:39.09,45.35,39.12,45.37 ADD:42 UPD:0 DEL:0	39.062,45.346,39.123,45.368
#3624099	(still editing)	g07egdo	Street sussawitsch - köstendorf - schinzengraben inserted	13.396,46.619,13.519,46.634
#3624098	(still editing)	Fips	(none)	8.644,50.164,8.644,50.165
#3624097	January 15, 2010 13:11	etiennerrando	Office Tourisme	7.358,48.079,7.363,48.082
#3624096	(still editing)	Supt_of_Printing	New road & buildings	145.345,-37.933,145.355,-37.924
#3624095	January 15, 2010 13:11	mhippeli	Nordheimer Wald	10.183,50.459,10.227,50.500
#3624094	January 15, 2010 13:11	walo	burgos	-3.965,42.464,-3.781,42.549 (big)
#3624093	(still editing)	friedemann2811	(none)	8.677,50.595,8.682,50.596
#3624092	(still editing)	hurdygurdyman	Fließrichtung berichtigt	7.799,48.329,7.821,48.341
#3624091	January 15, 2010 13:11	Mitridat	buildings added	28.857,47.044,28.872,47.051
#3624089	(still editing)	Palos	(none)	18.720,47.511,18.812,47.548
#3624088	January 15, 2010 13:10	Golovco Anatolie	Raut	27.973,47.396,28.310,47.435
#3624087	(still editing)	G1G@R@1NZ	fix name	136.377,35.361,136.377,35.361
#3624086	(still editing)	ruhri2010	(none)	7.092,51.461,7.102,51.464
#3624085	(still editing)	Mosmas	Gedenkstein	7.052,51.449,7.052,51.449
#3624084	(still editing)	moyogo	(none)	26.461,1.660,26.867,1.830 (big)
#3624083	January 15, 2010 13:09	moi_cheucle	Saint pal de Mons	4.263,45.192,4.306,45.255
#3624080	January 15, 2010 13:09	Erik Streb	haus der kita kotti e.v. mit adresse eingetragen	13.417,52.501,13.422,52.505
#3624079	January 15, 2010 13:09	Blackhex	BBOX:18.33,49.67,18.34,49.68 ADD:4 UPD:5 DEL:1	18.332,49.672,18.340,49.679

« Previous | Showing page 1 | Next »

- Export : Eina que permet importar parts del mapa en diferents formats gràfics.

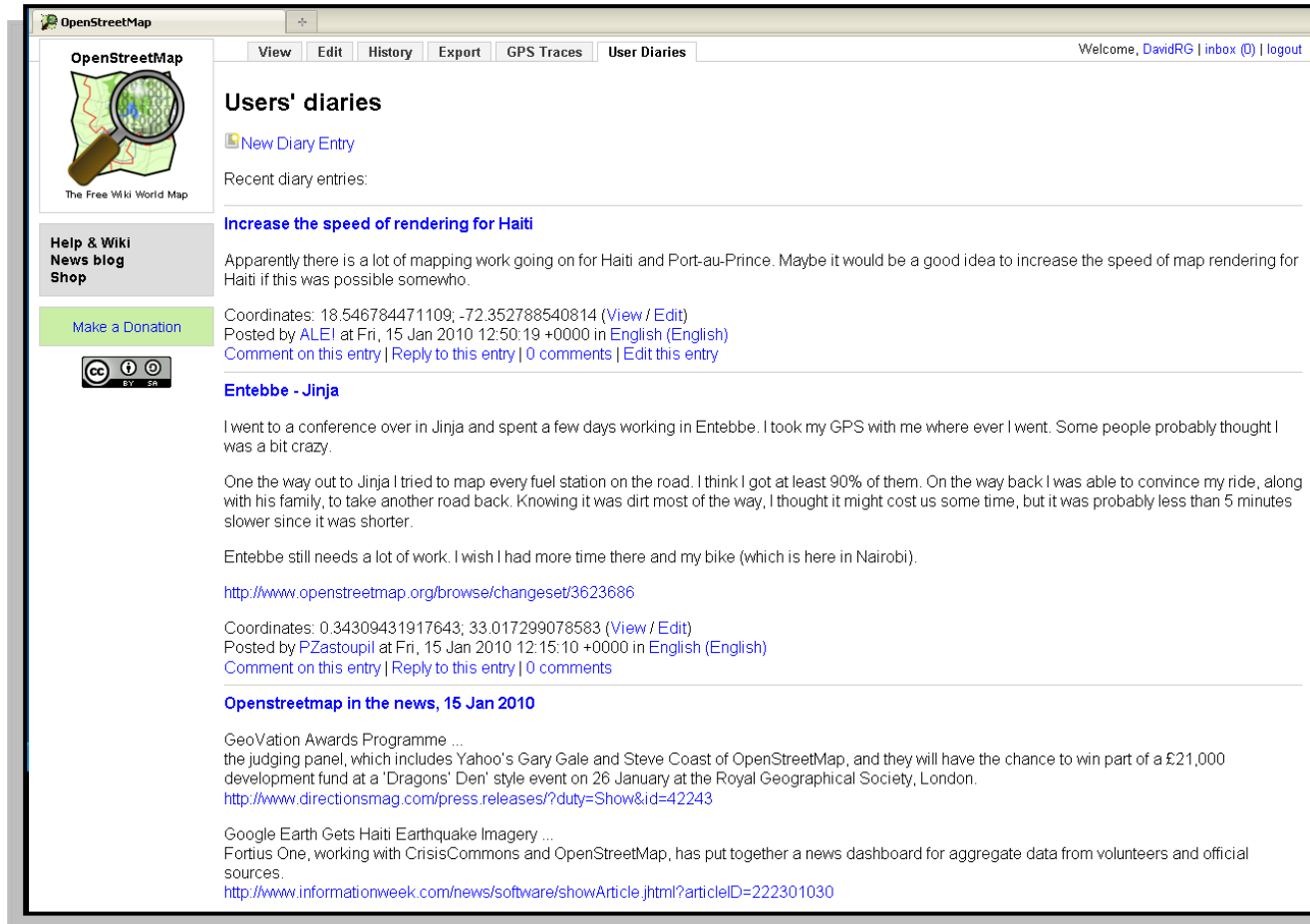


Els formats gràfics per importar son : PNG, JPEG, SVG, PDF i en Postscript

- **GPS Traces** : ens mostra les traces GPS que han creat els usuaris.



- **User Diaries** : Mini-blog de les diferents edicions i notícies dels usuaris de OSM



2. Registre

Per registrar-te Premem a “[registrar-se](#)” i ens porta a la següent pàgina, introduïm la direcció de correu electrònic i escollim un password. La informació del compte ens la envia a la direcció del mail per activar-la.

OpenStreetMap | Crear cuenta

Ver Editar Historial Exportar Trazas GPS Diarios de usuario [identificarse](#) [registrar-se](#)

Crear una cuenta de usuario

Llenar el formulario y te enviaremos un mensaje de correo rápidamente para activar tu cuenta.

Por la creación de una cuenta, usted está conforme en que todos los datos que envíe al proyecto Openstreetmap serán licenciados (no exclusivamente) bajo [esta licencia Creative Commons \(by-sa\)](#).

Dirección de correo:

Confirmar la dirección de correo:

No se muestra de forma pública (vea la [política de privacidad](#))

Nombre en pantalla:

Tu nombre de usuario público. Puedes cambiarlo más tarde en "preferencias".

Contraseña:

Confirmar contraseña:

Terminado

Una vegada estem registrats ens podem identificar, prement a “[identificarse](#)”, ens posarà el nom de usuari que haguem escollit.

Welcome, [DavidRG](#) | [inbox \(0\)](#) | [logout](#)

En aquest moment ja podeu començar a editar la cartografia de OSM.

Prement al nom de l'usuari, podem accedir a informació del nostre compte, les edicions que hem realitzat, les traces GPS que hem carregat, etc..

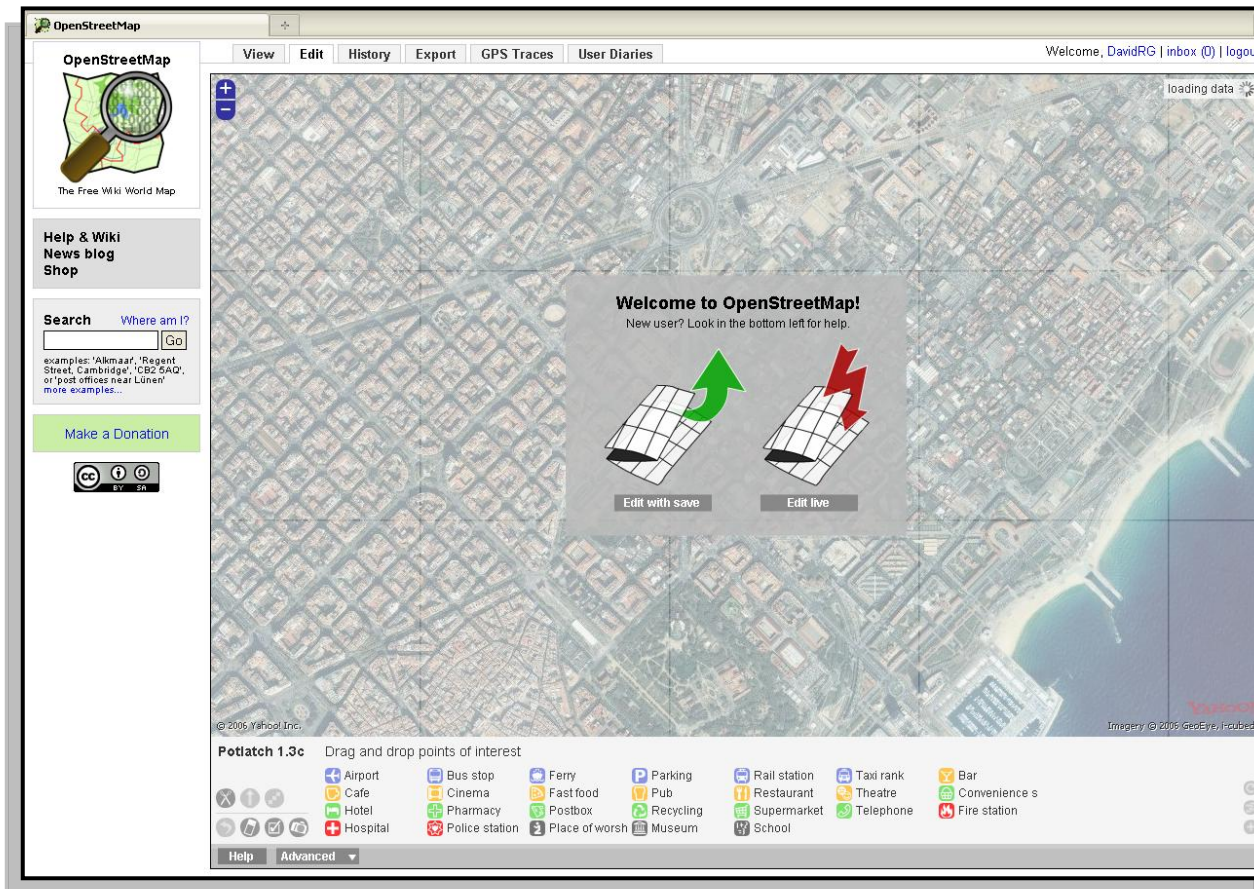
També disposem d'una bústia de missatges per comunicar canvis als altres usuaris.

Prement a “[logout](#)”, ens sortim de la compte del usuari actiu.

3. Edició

Una vegada tenim l'usuari creat podem prémer a la pestanya “**Editar**”, la finestra de visualització canviarà i es posarà la capa wms de yahoo, per ajudar a crear nous elements.

Ja estem dintre de l'editor online Potlatch, per començar a editar ens surt una finestra amb dos opcions per guardar els canvis :



- **Edit with save**

Guarda els canvis que s'hagin realitzat quan decideixis, amb un botó d'opció per guardar.

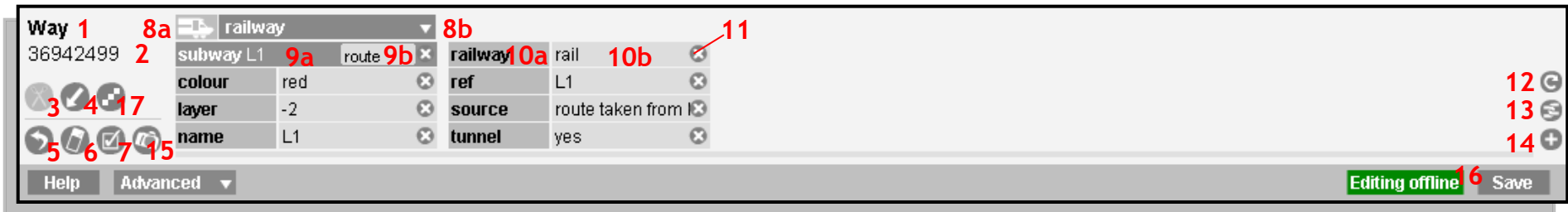
- **Edit live**

Guarda els canvis automàticament.

Una vegada seleccionat el mode d'edició ja podem començar a editar i afegir nous elements, orientant-nos amb l'ortofoto que tinguem carregada.

• Eines de Potlatch

En la part inferior de la pantalla ens trobem la interfície de les opcions que conté potlatch, hi ha un munt d'opcions per editar els nostres elements, però es van actualitzant algunes funcions.



- 1 : Tipus de dada seleccionada (Node, way, way tancat)
- 2 : Identificador de l'objecte seleccionat
- 3 : Eina per tallar ways i ways tancades (Split)
- 4 : Selector de direcció de la way
- 5 : Desfàs una acció
- 6 : Mostra les traces creades
- 7 : Mostra opcions de visualització de les wms
- 8 : Selector de plantilla : a) la classe, b) el tipus
- 9 : Etiqueta de relació : a) l'objecte seleccionat es membre d'(a) i membre de la funció, si s'escau de (b)
- 10 : Objecte etiquetes : a) la clau o key b) el valor o value
- 11 : Elimina etiqueta o relació
- 12 : Copia l'etiqueta de l'objecte seleccionat prèviament
- 13 : Obrir l'editor de relacions
- 14 : Afegir una etiqueta nova
- 15 : Càrrega fotos als objectes
- 16 : Mode d'edició
- 17 : Alinea els nodes de les ways

La funció més important a entendre es la inserció d'etiquetes en els elements, una vegada tinguem dibuixada una línia, node o línia tancada li podem atorgar de quina Classe és, per exemple si es una classe de carretera i de quin tipus és, com a primària, secundària,etc.. Així funciona amb totes les característiques físiques i no físiques, en el cas de que es tingui dubte de quina característica posar, trobem recollides totes les que hi ha per defecte al sistema OSM a la wiki de la pròpia plana web :

http://wiki.openstreetmap.org/wiki/Map_Features

• Desplaçar-se pel mapa

Mode visualització

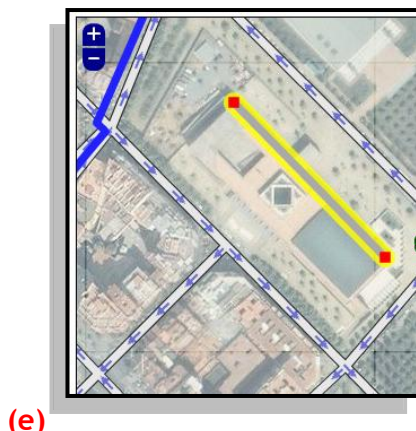
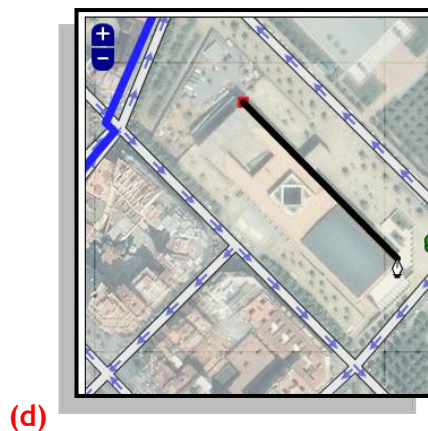
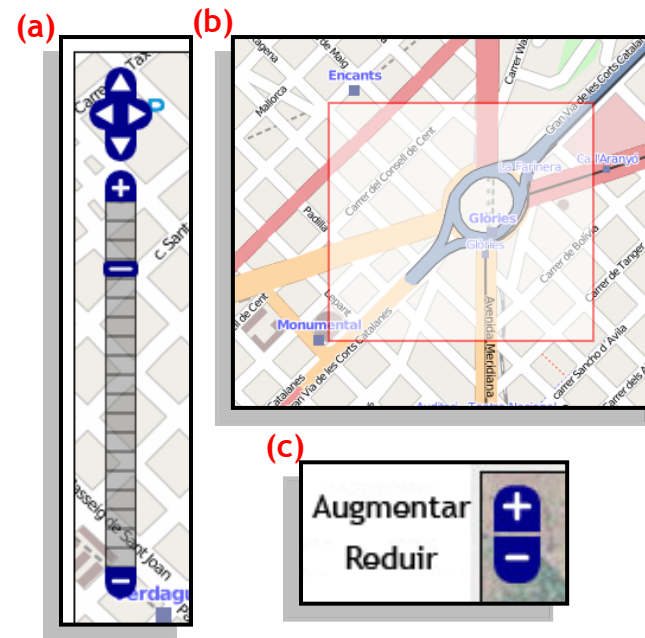
- Prement i pressionant amb el botó esquerra del ratolí.
- Augmentar o reduir el nivell de visualització amb la rodeta central del ratolí.
- Augmentar o reduir el nivell de visualització amb la icona "+,-". (a)
- Crear una finestra de augment (Ctrl + botó esquerra pressionat). (b)

Mode edició

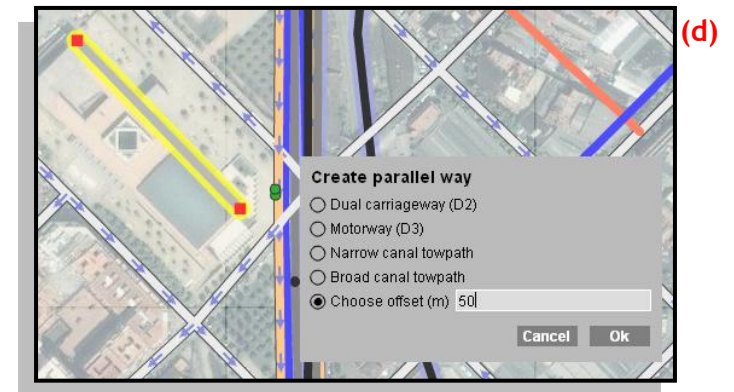
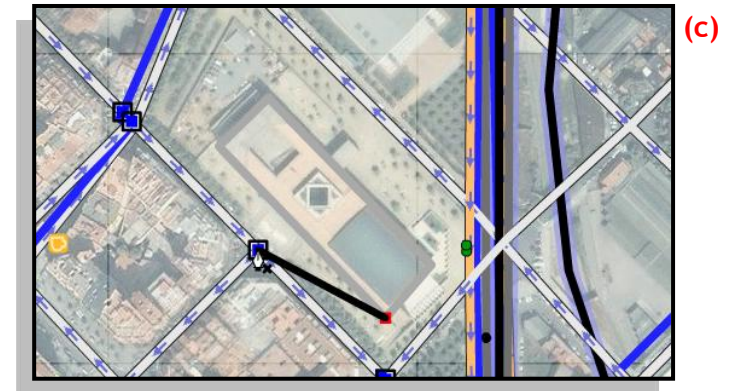
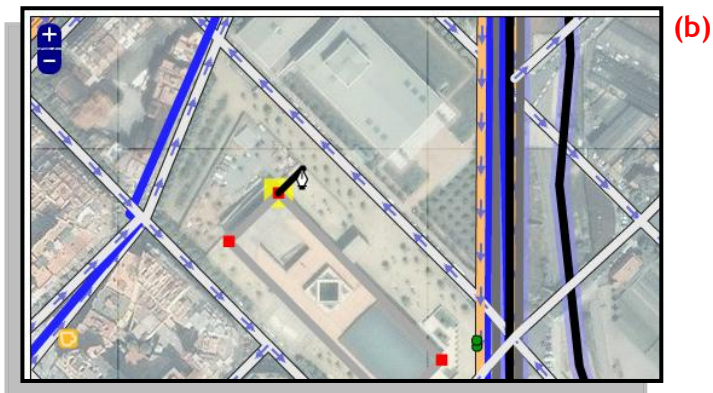
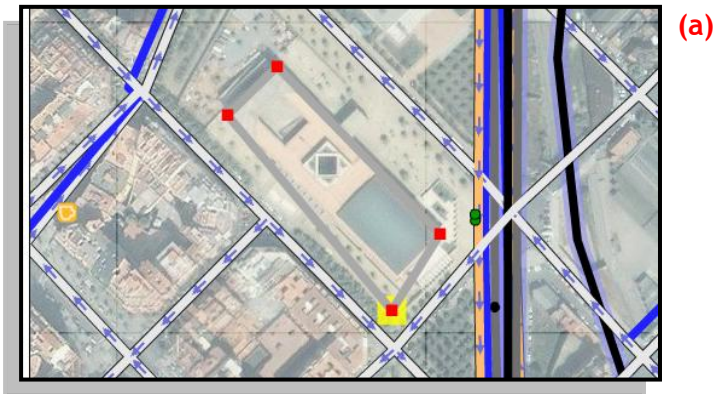
- Prement i mantenint amb el botó esquerra del ratolí.
- Augmentar o reduir el nivell de visualització amb la rodeta central del ratolí.
- Augmentar o reduir el nivell de visualització amb la icona "+,-". (c)

• Edició

- Feu clic a una àrea en blanc per iniciar una nova línia. (d)
- Un cop fet clic tenim una línia virtual, fem clic al següent punt per finalitzar-la. (e)
- Fem doble clic per terminar la línia.
- Tecla "Esc" o "Enter" per cancel·lar una edició o una nova línia.
- Fem doble clic al node inicial de la línia per crear una àrea tancada (f)



- Premem i mantenim en un node per modificar la geometria. **(a)**
- Tecla "Shift+Supr" per esborrar un element.
- Tecla "Ctrl+Z" per anar endarrere en l'edició (Undo).
- Feu doble-clic per inserir un node.
- Fem clic a un node inicial per ampliar la línia. **(b)**
- "Shift+clic" al node de una línia per continuar des del propi node.
- Ens apropem a un node i ens marca els nodes a unir. **(c)**
- Amb una línia seleccionada premeu "P" per crear formes paral·leles. **(d)**
- Tecla "X" per dividir un camí.
- Tecla "K" per bloquejar o desbloquejar un camí.

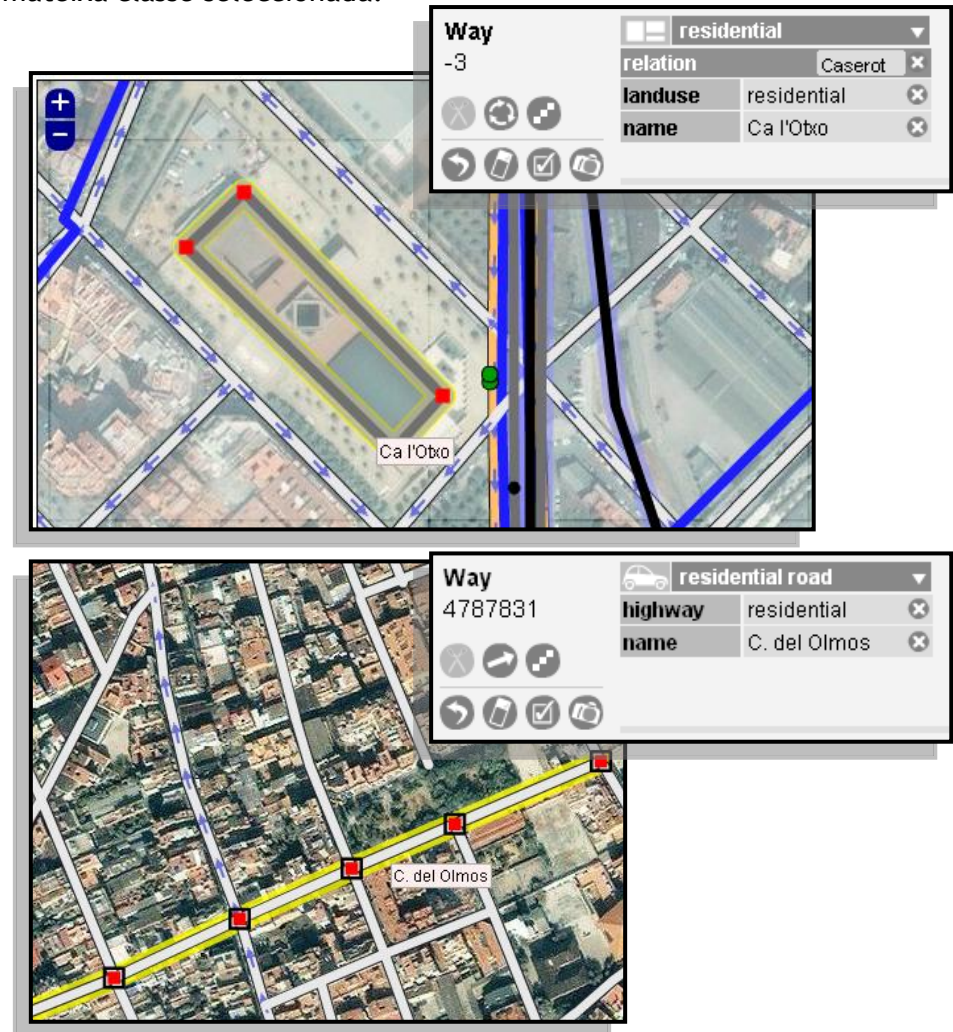


- **Etiquetes/Atributs**

- Tecla “+” per afegir una nova clau / valor.
- Tecla “R” per repetir l’última etiqueta posada.
- Tecles de l’1 al 9, per anar afegint etiquetes preestablertes de la mateixa classe seleccionada.

- **Altres**

- Tecla “S” per guardar les edicions.
- Tecla “L” per mostrar les coordenades geogràfiques.
- Tecla “M” per maximitzar la finestra del navegador.
- Tecla “Bloq maj” per afegir transparència als elements.
- Tecla “F2” per activar ortofoto de yahoo.
- Tecla “F3” per activar Mapnik Tiles.
- Tecla “F4” per activar Osmarender Tiles.
- Tecla “F5” Per activar Maplint.
- Tecla “F1” Per desactivar la ortofoto.



Annex III Manual bàsic de JOSM



Índex

1. Introducció	106
▪ Requisits del sistema	
2. Interfície gràfica	107
2.1 Barra d'eines d'edició	109
2.2 Barra d'eines OSM	114
▪ Tractament de dades OSM	
▪ Eines OSM	
2.3 Configuració de preferències	116
▪ Ajustaments de la connexió	
▪ Preferències WMS	

1. Introducció

JOSM (Java OpenStreetmap editor), és un dels principals editors amb el que contem a OpenStreetMap, hi ha altres editors com Potlatch o Merkaartoor, però en concret JOSM, està dirigit a usuaris de OSM experimentats, si vols editar només el nom d'un carrer, utilitza millor potlatch.

Amb JOSM, pots modificar un gran conjunt de dades sense necessitat de que aquestes edicions es modifiquin de forma immediata a la base de dades. Permet treballar en les zones que es creu que hi ha altres persones editant dades, amb la fusió d'aquests canvis i la solució de conflictes de l'edició.

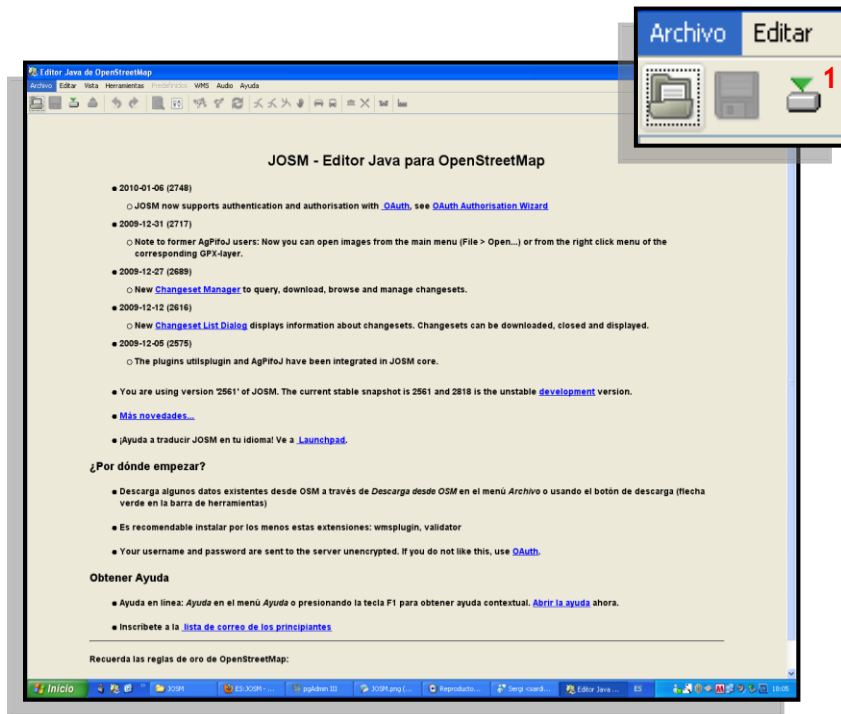
La possibilitat d'incloure nous plugins, com la càrrega de noves ortofotos o descarregar els planet.osm des de la mateixa interfície del programa el fa molt útil en quan a velocitat d'edició.

- **Requisits del sistema**
 - Funciona amb tots els sistemes operatius.
 - Java 1.5 o superior
 - 256 MB o més de RAM
 - Resolució de pantalla de 1024x768 o superior
 - Es recomana un adaptador gràfic amb acceleració 2D

2. Interfície gràfica

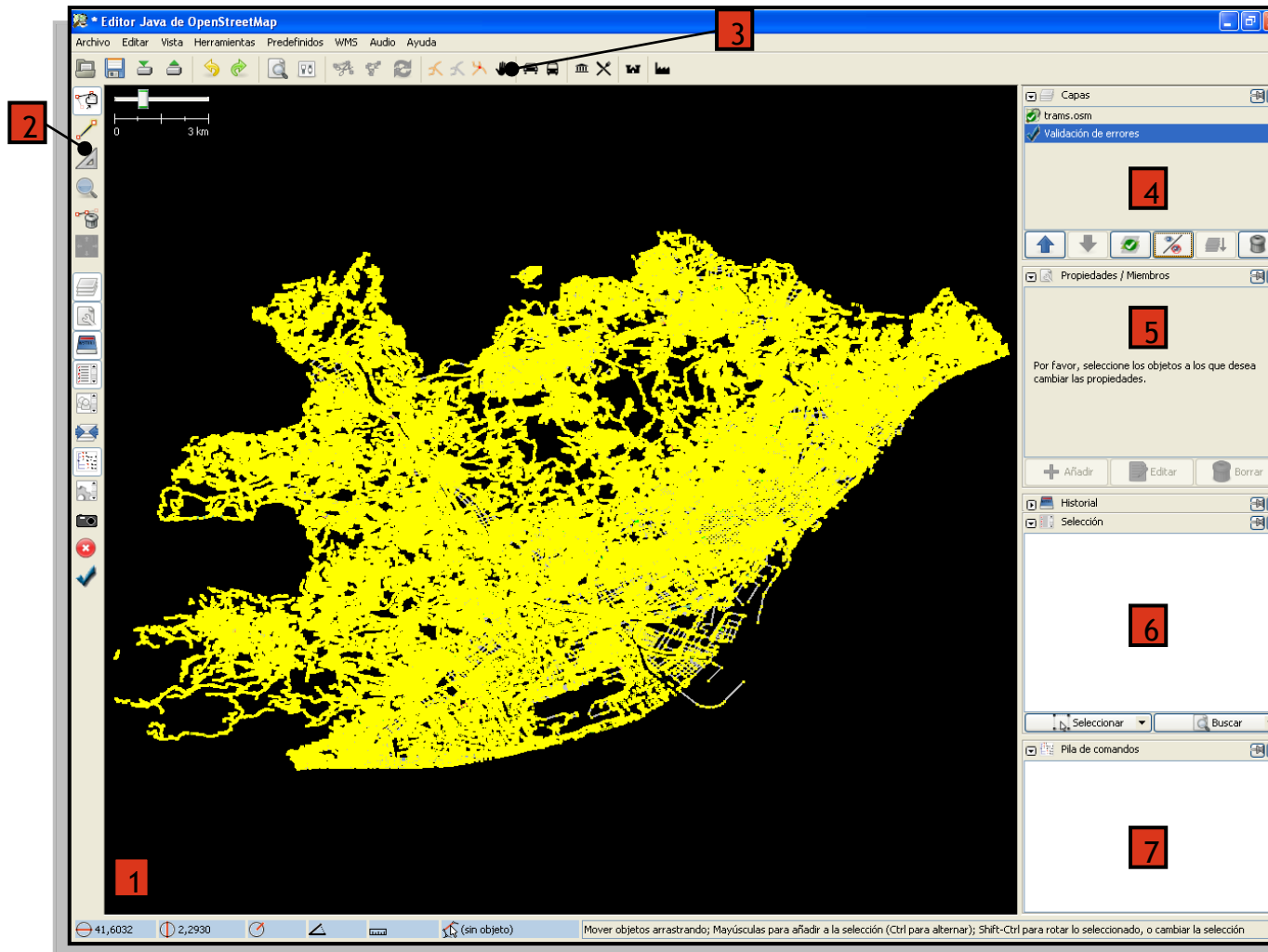
Una vegada instal·lat Java i JOSM, executem l'aplicació i ens trobem amb la següent pantalla, que ens mostra les últimes notícies d'actualitzacions que tenim per al programari. A la part superior de la pantalla ens mostra dos opcions per veure els arxius OSM :

- 1: Descarregant una part del planet OSM
- 2: Obrint un fitxer .osm



Un cop obert el fitxer ens trobem amb la següent interfície, tot seguit expliquem les següents eines que trobem :

- | | | |
|------------------------------|---|-----------------------------------|
| 1. Vista general de l'editor | 4. Llistat de capes carregades | 7. Llistat de comandes executades |
| 2. Barra d'eines de edició | 5. Llistat de selecció d'objectes (Key/Value) | |
| 3. Eines OSM | 6. Llistat de elements seleccionats | |



2.1 Barra d'eines d'edició

Podem trobar la barra d'eines al lateral esquerra de la vista general del mapa, tot seguit descrivim les principals eines de les que disposem per editar els elements de OSM :

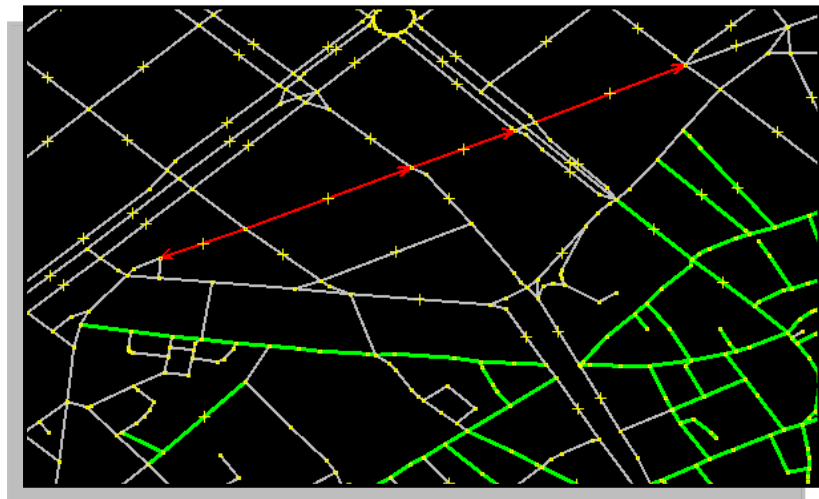
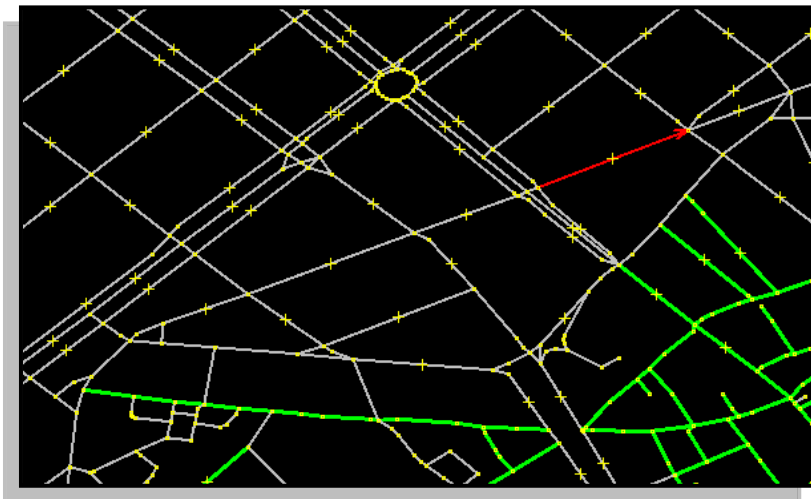


Selecció d'objectes

Aquesta icona és l'eina principal per seleccionar els elements que tinguem a la capa carregada, es pot fer una selecció múltiple clicant a diverses ways.

- Per seleccionar premem sobre qualsevol way o node.
- Un cop seleccionat l'element ens mostra quants nodes tenim seleccionats i les ways o línies que conformen aquests nodes.

Els nodes els mostra amb una creu groga, i les ways es senyalitzen amb color vermell, així com la direcció de la línia amb una fletxa del mateix color de selecció.

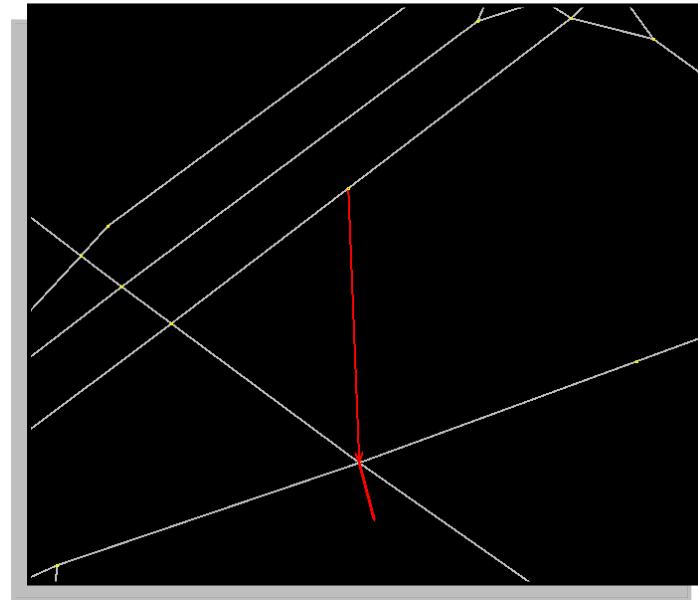
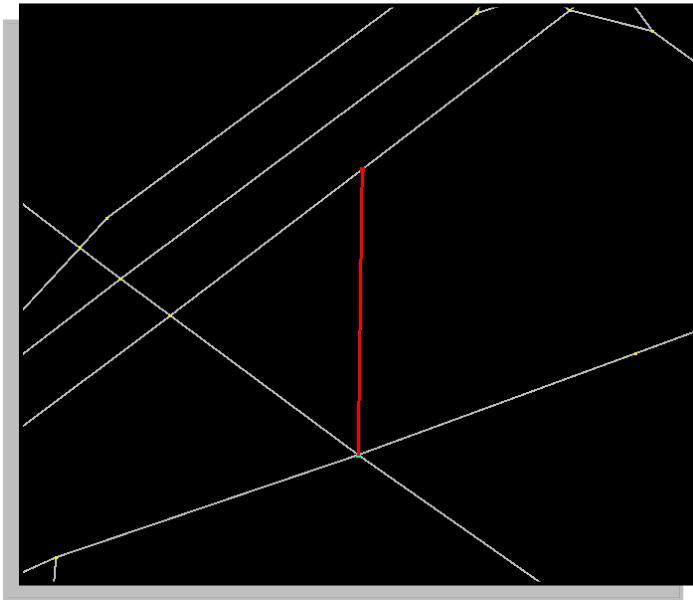




Creació d'objectes

Icona per a la creació de nous elements :

- Cliquem a un àrea qualsevol o en un node per crear una línia nova.
- Doble clic per crear un node.
- Tancar al node inicial per crear una way tancada.



Mesura

Eina de mesura, per calcular superfícies a partir del perímetre de les way seleccionades o les ways tancades seleccionades.



Augmentar o Reduir

Eina de visualització :

- Fent clic augmentem nivell de visualització.
- Augmentem o reduïm el nivell de visualització amb la rodeta del ratolí
- Tecla "Ctrl+Boto esquerra" per crear finestra de augment.

Podem trobar més Eines de visualització al *Menú > Vista*



Eliminar objecte

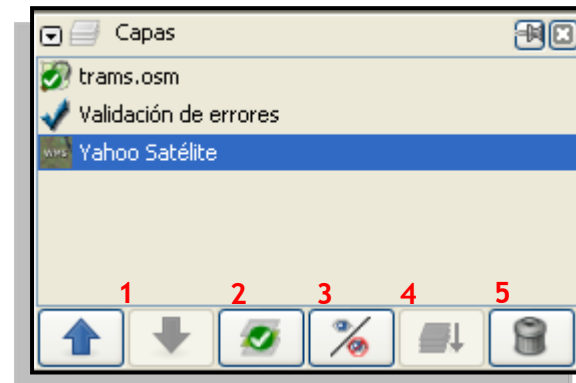
Quan tenim un o variis elements seleccionats, clicant en aquesta icona la suprimim.



Capes Carregades

Aquesta icona activa i desactiva la llista de capes .osm que tenim carregades, també ens mostra les capes wms i té quatre opcions principals :

- 1: **Ordre visualització** : Amb les fletxes pugem o baixem les capes seleccionades.
- 2: **Capa activa** : només podrem seleccionar els elements de la capa seleccionada.
- 3: **Visualització** : activem o desactivem la visibilitat de la capa.
- 4: **Combinar** : Combina capes amb la capa activa
- 5: **Paperera** : Eliminem la capa seleccionada.

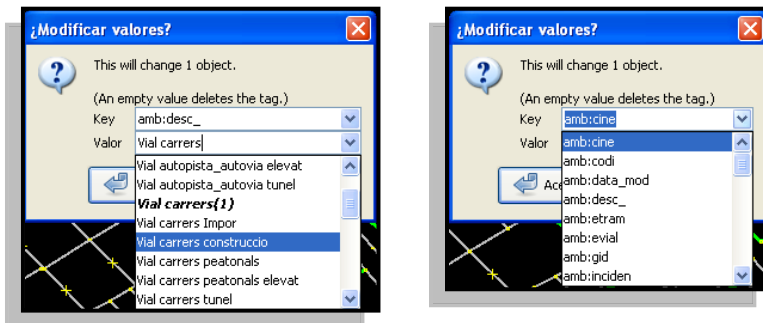




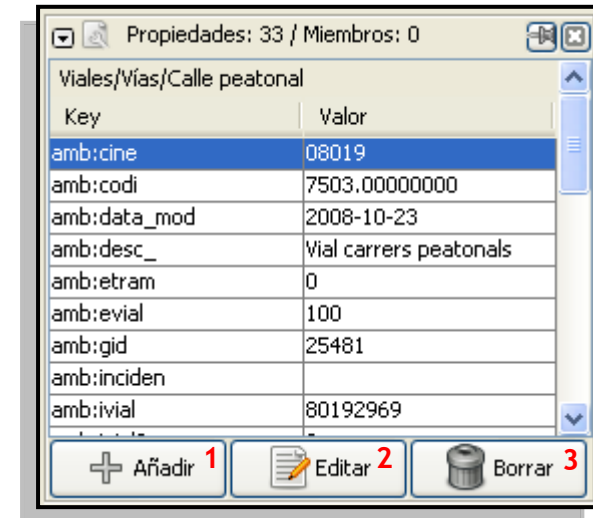
Propietats dels objectes

Aquesta icona activa i desactiva la llista de propietats (Etiquetes) amb la clau i el valor de l'element seleccionat al mapa i té tres opcions principals :

- 1: **Afegir** : Inserim una nova etiqueta, podem escollir la clau i el valor.
- 2: **Editar** : Pots modificar la clau i el valor de la tag seleccionada.



- 3: **Esborrar**: Esborrem l'etiqueta de la llista de propietats de l'element seleccionat.



Historial d'edició

Icona que activa i desactiva la llista d'edicions que anem realitzant en el transcurs de l'edició dels elements.



Seleccions d'objectes

Prement aquesta icona activem i desactivem la finestra de selecció dels elements escollits i ens Mostra la quantitat de dades que tenim seleccionades i de quin tipus són.

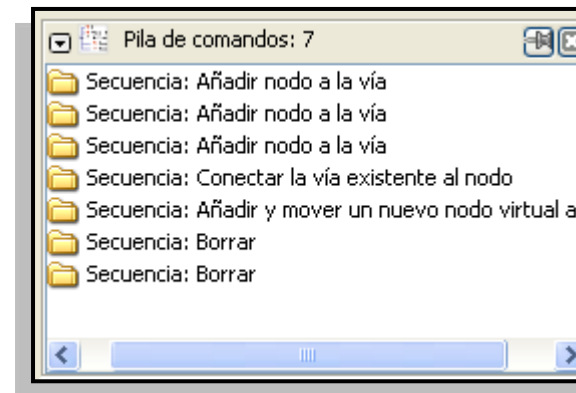
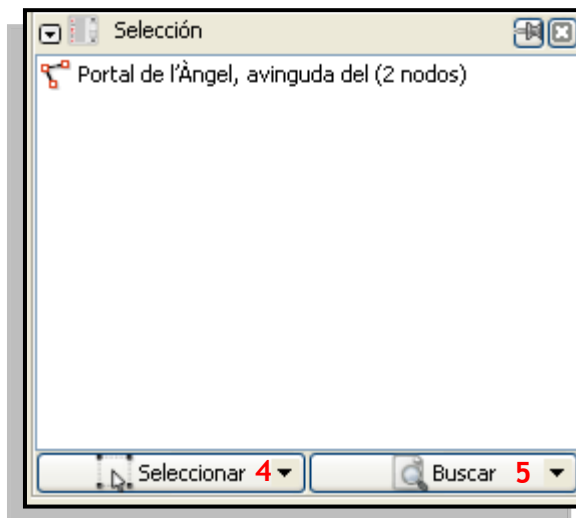
Hi ha dos opcions :

- 4: **Seleccionar** : Amb aquesta eina pots seleccionar els elements que tens actius.
- 5: **Cercador** : Et cerca per paraules claus els valors que hi ha a la capa seleccionada.



Pila de comandes executades

La icona ens mostra les comandes executades des de que s'ha obert el fitxer, et va mostrant a la llista les comandes que s'han realitzat i sobre quin element s'ha fet.



2.2 Barra d'eines OSM

Tenim la barra d'eines a la part superior de la vista general, i podem trobar icones per carregar dades i tractar les etiquetes de OSM, to seguit ho descrivim :

- **Tractament de dades OSM**



Obrir Fitxer

Podem obrir fitxers en format .osm guardats a qualsevol disc local.



Guardar Fitxer

Guarda els fitxers en format .osm de les edicions que s'hagin realitzat, els guarda als discos locals escollits.



Descàrrega de dades OSM

Icona que ens permet descarregar fitxer .osm d'una zona determinada del mapa OSM, guardat als seus servidors principals, per poder descarregar has de posar el teu nom d'usuari i la contrasenya creada per poder baixar les dades.

Les dades descarregades es poden guardar després en un arxiu .osm o carregar les dades al servidor OSM.



Càrrega de dades OSM

Icona germana de l'anterior, permet carregar les edicions modificades tant en un fitxer osm carregat des d'un disc local, com les dades descarregades dels servidors de OSM.

Per poder pujar les edicions fetes, en les opcions has de posar el nom d'usuari i la contrasenya a les preferències.



Fer i desfer

Permet fer i desfer les accions d'edició realitzades als arxius .osm.

- Eines OSM



Cercador

Mostra una finestra que ens permet fer una cerca dins del fitxer .osm, disposa de diferents criteris de cerca.




Etiquetes OSM

Icones de les principals etiquetes que proporciona per defecte els servidors OSM.



2.3 Configuració de preferències

Per accedir a la configuració OSM, podem accedir :

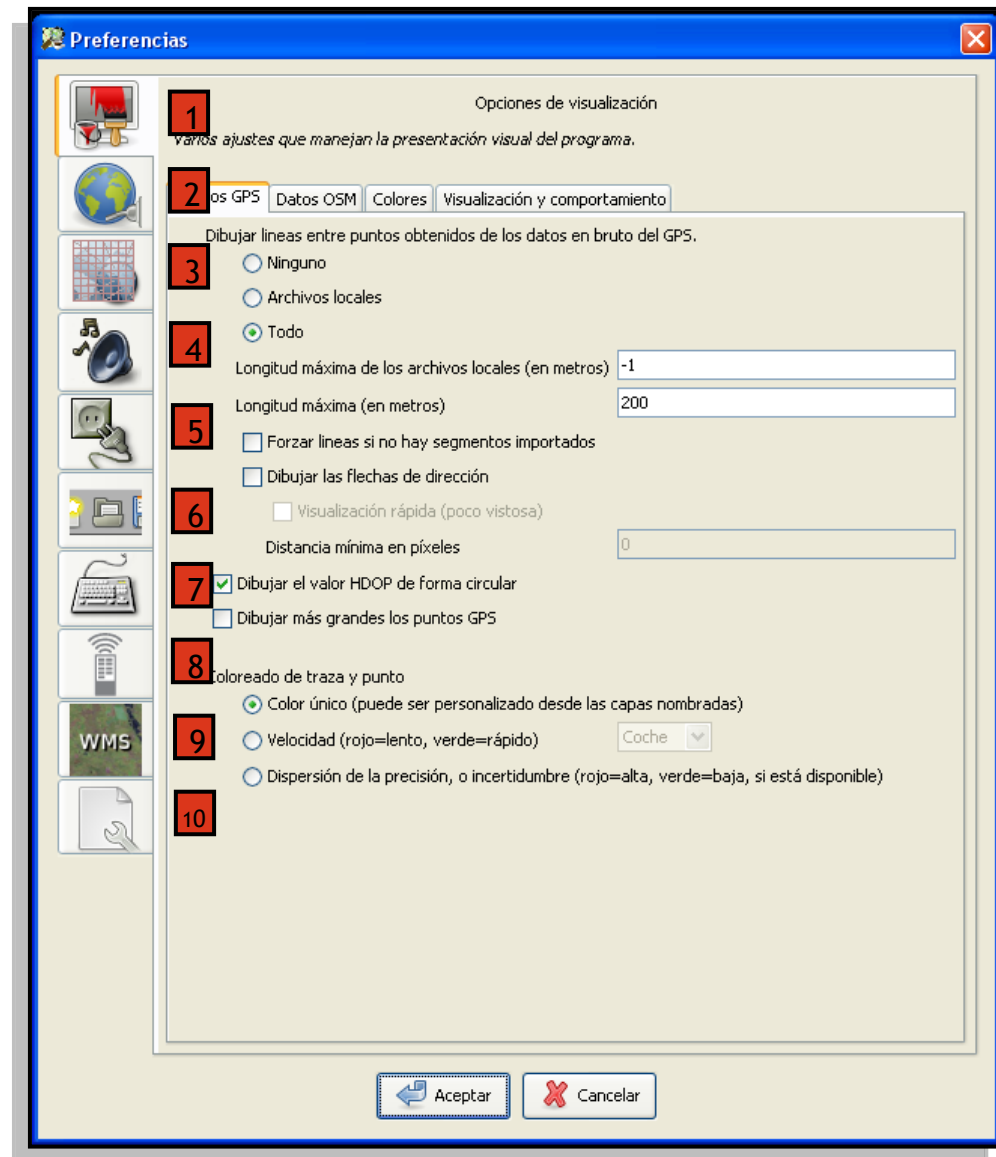
- Clicant en la icona : 

- Anant al menú : *Editar > Preferencias*

A les preferències es poden canviar moltes opcions :

1. Opcions de visualització
2. Ajustaments de la connexió
3. Ajustaments del Mapa
4. Configuració d'Audio
5. Complementos
6. Personalització de la barra d'eines
7. Dreceres de teclat
8. Control remot
9. Preferències WMS
10. Preferències avançades

A continuació expliquem les pestanyes de preferències més importants, Els ajustaments de la connexió i les capes WMS



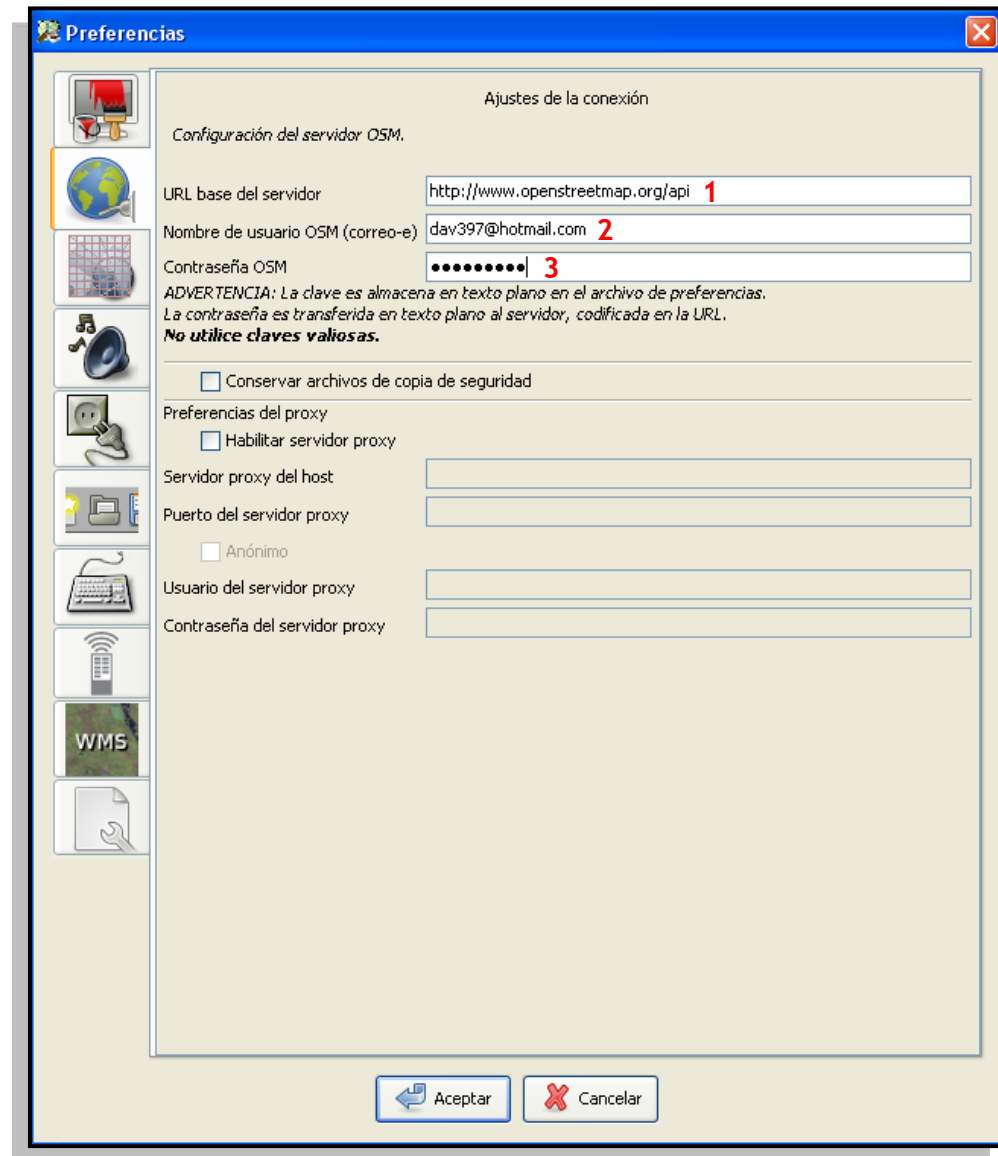
• Ajustaments de la connexió

Per poder carregar les dades al servidor tenim que dir-li una sèrie de dades :

- 1: URL : Direcció on es troba el servidor de dades OSM
- 2: Nom d'Usuari : Usuari creat en potlatch
- 2: Contrassenya: Clau de l'usuari creat en potlatch

Quan fem servir la icona de càrrega de dades, ens pujarà aquestes dades al servidor amb l'usuari i la contrasenya indicades.

S'ha d'anar amb molta cura quan pugem dades de mida molt gran, depenent del servidor pot causar errors de lectura o pèrdua d'informació.



• Preferències WMS

Per poder carregar les dades al servidor tenim que dir-li una sèrie de dades :

- 1: WMS carregades : Mostra els servidors WMS afegits
- 2: WMS disponibles : Llistat de servidors WMS

Podem posar les capes WMS que hem afegit a la llista des de el menú WMS, com mostrem en la imatge següent :

