

Llicència GPL3

Tots els drets garantits per a aquesta Llicència s'atorguen com a copyright del Programa, i es proporcionen de manera irrevocable sempre que es compleixin les condicions establertes. Aquesta Llicència afirma explícitament el seu permís il·limitat per a executar el Programa sense modificacions. El resultat de la execució d'un programa emparat està cobert per aquesta Llicència només si la sortida, pel seu contingut, constitueix un treball emparat. Aquesta Llicència reconeix els seus drets per un ús raonable o altre equivalent, tal com determina la llei del copyright.

Podeu realitzar, executar i difondre treballs emparats que no traspasseu, sense cap altra condició mentre no tinguin una altra llicència més restrictiva. Podeu traspassar els treballs emparats a tercers amb el simple objectiu que ells facin modificacions exclusivament per a vostè, o perquè us proporcionin ajuda per a executar aquests treballs, sempre que compleixi els termes d'aquesta Llicència transportant tot el material del copyright del qual no posseïu el control. Aquells que realitzin o executin els treballs emparats per a vostè han de fer-lo exclusivament en el seu nom, sota la vostra direcció i control, amb els termes que els prohibeixin realitzar còpies del vostre material amb copyright al marge de la relació amb vostè.

Podeu traspassar còpies literals del codi font del Programa tal com el rebeu, en qualsevol mitjà, sempre que publiqueu de forma clara i cridanera en cada còpia un avís de copyright apropiat; mantingueu intactes tots els avisos que afirmen que aquesta Llicència i qualsevol terme no-permissiu afegit d'acord amb la secció 7 són aplicables al codi; mantingueu intactes tots els avisos d'absència de garanties; i proporcioneu a tots els destinataris una còpia d'aquesta Llicència juntament amb el programa.

Podeu traspassar un treball basat en el Programa, o les modificacions produïdes a partir del Programa, com Codi font sota els termes de la secció 4, sempre que compleixi totes les condicions següents:

- a) El treball ha d'incloure avisos destacats indicant que vostè ho ha modificat, donant una data pertinent.
- b) El treball ha d'incloure avisos destacats indicant que ha estat realitzat sota aquesta Llicència i qualsevol altra condició afegida sota la secció 7.
- c) Heu d'aplicar la llicència al treball complet, com un tot, a qualsevol persona que tingui possessió d'una còpia. Aquesta Llicència s'aplicarà per tant, igual que qualsevol altra condició addicional de la secció 7, al conjunt complet del treball i totes i cadascuna de les seves parts, independentment de com siguin agrupades o empaquetades. Aquesta Llicència no permet ser aplicada al treball de cap altra forma, però no s'anul·larà el permís esmentat si ho heu rebut per separat.
- d) Si el treball té interfícies d'usuari interactives, cadascuna ha de mostrar Avisos Legals Apropisats; de totes maneres, si el Programa té interfícies interactives que no mostren Avisos Legals Apropisats, el seu treball no té perquè modificar-les perquè ho facin.

UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria en Informàtica

Xarxes de Consum Solidaries

Alumne/a: Jordi Farré Ramon
Dirigit per: Anna Muñoz Bollas
Co-dirigit per:

CURS 2010-11 (Febrer/Setembre)

UNIVERSITAT OBERTA DE CATALUNYA

Enginyeria

2. Dedicatòria i Agraïments

Vull dedicar especialment aquest projecte a la meva novia Maria, que m'ha animat en tot moment i m'ha donat la empenta necessària per poder acabar aquest projecte. Sense els seus ànims, ara mateix no estaria finalitzant aquest projecte.

També vull dedicar-ho als meus pares Jordi i Nuri, que sempre han confiat en mi i en les meves possibilitats, des de que vaig començar els estudis, fins a l'últim moment de la carrera.

No vull oblidar-me de l'Anna Muñoz, la meva consultora en aquest projecte de final de carrera, que m'ha orientat en tot moment i m'ha ajudat per poder assolir el final d'aquest projecte.

3. Resum

Les Xarxes de Consum Solidari (XCS) o Grups de Consum Alternatiu del territori català són associacions que treballen en l'àmbit del comerç just i el consum responsable. Defensen un comerç just amb la voluntat de transformar radicalment el sistema per tal que es pagui un preu digne al productor, que respecti el medi ambient i que es subministrin aliments sans i adequats culturalment.

El present projecte haurà d'implementar un portal web que permeti cercar les ubicacions dels diferents grups de consum agroecològic, així com dels diferents productors, de les fires i mercats de productes ecològics, dels horts urbans, etc. Des d'aquest portal també s'haurà de poder gestionar la base de dades amb la informació relacionada.

4. Índex de Continguts e Índex de figures

Índex de continguts

| | |
|---|----|
| 2. Dedicatòria i Agraïments..... | 3 |
| 3. Resum..... | 4 |
| 4. Índex de Continguts e Índex de figures..... | 5 |
| 5. Cos de la memòria..... | 7 |
| 5.1. Capítol 1, Introducció..... | 7 |
| 5.1.1. Justificació del TFC i context: punt de partida i aportació del PFC..... | 7 |
| 5.1.2. Objectius del PFC..... | 7 |
| 5.1.3. Requeriments mínims..... | 8 |
| 5.1.4. Enfocament i mètode seguit..... | 8 |
| 5.1.5. Planificació del projecte..... | 9 |
| 5.1.5. Productes obtinguts..... | 9 |
| 5.1.6. Descripció dels altres capítols de la memòria..... | 9 |
| 5.2. Capítol 2. Instal·lació de programari..... | 10 |
| 5.2.1. Instal·lació Tomcat..... | 10 |
| 5.2.2. Instal·lació Geoserver..... | 10 |
| 5.2.3. Instal·lació PostgreSQL i PostGIS..... | 11 |
| 5.3. Capítol 3. Recerca de dades..... | 12 |
| 5.3.1. Recerca de fonts de dades..... | 12 |
| 5.3.2. Recerca de dades cartogràfiques..... | 12 |
| 5.4. Capítol 4. Disseny de base de dades..... | 13 |
| 5.4.1 Disseny lògic de base de dades..... | 13 |
| 5.4.2. Disseny físic de base de dades..... | 17 |
| 5.5. Capítol 5. Importació de dades..... | 20 |
| 5.5.1. Importació a base de dades..... | 20 |
| 5.5.2. Importació a GeoServer..... | 23 |
| 5.6. Capítol 6. Creació entorn desenvolupament..... | 27 |
| 5.6.1. Instal·lació Maven..... | 27 |
| 5.6.2. Instal·lació GWT..... | 28 |
| 5.6.3. Instal·lació GXT..... | 28 |
| 5.6.4. Instal·lació GWT-Openlayers..... | 28 |
| 5.6.5. Instal·lació Spring Framework..... | 28 |
| 5.6.6. Instal·lació Eclipse..... | 28 |
| 5.6.7. Creació i configuració del projecte..... | 29 |
| 5.6.8. Estudi de llibreries..... | 30 |
| 5.7. Capítol 7. Disseny Aplicació..... | 31 |
| 5.7.1. Estàndards utilitzats..... | 31 |
| 5.7.2. Arquitectura de l'aplicació..... | 33 |
| 5.7.3. Disseny part Servidor..... | 35 |
| 5.7.4. Disseny part Client..... | 36 |
| 5.8. Capítol 8. Implementació..... | 39 |

| | |
|---|----|
| 5.8.1. Implementació Accés BBDD..... | 39 |
| 5.8.1.1. Implementació DAO amb Spring..... | 41 |
| 5.8.2. Implementació serveis web..... | 42 |
| 5.8.2.1. Implementació servei GWT RPC..... | 43 |
| 5.8.3. Implementació part Client..... | 44 |
| 5.8.3.1. Implementació GWT..... | 45 |
| 5.8.4. Implementació dels mapes..... | 45 |
| 5.8.4.1. Implementació GWT-Openlayers..... | 47 |
| 5.8.5. Implementació de la pantalla de consultes..... | 50 |
| 5.8.6. Pantalles de manteniment amb mapa..... | 54 |
| 5.9. Capítol 9. Línies de treball futures..... | 57 |
| 5.10. Capítol 10. Conclusions..... | 57 |
| 6. Bibliografia..... | 59 |
| Annex 1. Guia ràpida d'instal·lació..... | 61 |
| Annex 2. Guia ràpida de compilació..... | 63 |

Índex d'il·lustracions

| | |
|--|----|
| Il·lustració 1: Diagrama ER de base de dades..... | 14 |
| Il·lustració 2: Login GeoServer..... | 23 |
| Il·lustració 3: Creació de un nou Workspace..... | 24 |
| Il·lustració 4: Creació de un nou DataSource..... | 25 |
| Il·lustració 5: Selecció de taules a publicar en GeoServer..... | 26 |
| Il·lustració 6: Càlcul de Bounding Boxes..... | 27 |
| Il·lustració 7: Dependències del projecte XCS..... | 29 |
| Il·lustració 8: Dissenyador visual de GWT..... | 31 |
| Il·lustració 9: Arquitectura de l'aplicació..... | 34 |
| Il·lustració 10: Diagrama UML de serveis web..... | 35 |
| Il·lustració 11: Diagrama UML d'accés a BBDD..... | 36 |
| Il·lustració 12: Diagrama UML de la part Client..... | 37 |
| Il·lustració 13: Exemple de pantalla de manteniment simple..... | 38 |
| Il·lustració 14: Exemple de pantalla de manteniment amb mapa..... | 39 |
| Il·lustració 15: Diagrama UML per la implementació de un servei GWT RPC..... | 42 |
| Il·lustració 16: Capes pantalla de cerca..... | 46 |
| Il·lustració 17: Capes pantalla manteniment..... | 47 |
| Il·lustració 18: Extensió de GWT-Openlayers..... | 48 |
| Il·lustració 19: Resultat cerca filtrant per població..... | 51 |
| Il·lustració 20: Resultat al filtrar per producte..... | 52 |
| Il·lustració 21: Resultat al filtrar per esdeveniment..... | 53 |
| Il·lustració 22: Resultat al filtrar per una xarxa de consum solidari..... | 54 |
| Il·lustració 23: Manteniment de productors, mostrant les dades d'un productor..... | 56 |

5. Cos de la memòria

En aquest punt de la memòria, definirem tot el contingut específic del PFC. Començant per una breu introducció del PFC i seguint amb els aspectes propis del projecte com la instal·lació de programari, el disseny de base de dades i totes les decisions relacionades amb el disseny i la implementació de l'aplicació.

5.1. Capítol 1, Introducció

En aquest primer capítol, farem una introducció del que són les Xarxes de Consum Solidari i del que pretén aportar el nostre projecte a aquest context.

També indicarem la metodologia a seguir i la planificació establerta per la realització del projecte.

5.1.1. Justificació del TFC i context: punt de partida i aportació del PFC

Les Xarxes de Consum Solidari (XCS) o Grups de Consum Alternatiu del territori català són associacions que treballen en l'àmbit del comerç just i el consum responsable. Es parteix d'una visió integral del comerç just que engloba tot el cicle d'un producte, des de la seva producció fins a la seva comercialització a casa nostra. Defensen un comerç just amb la voluntat de transformar radicalment el sistema per tal que es pagui un preu digne al productor, que respecti el medi ambient i que es subministrin aliments sans i adequats culturalment. Consideren que productors i consumidors han de ser els que decideixin quins aliments produir. En definitiva, defensen el dret a la sobirania alimentària dels pobles. Els grups i cooperatives de consum agroecològic són una realitat cada dia més present a nivell local. I tot i que en xifres totals, es tracta d'experiències que sumen un reduït grup de persones, demostren que és possible un altre model de consum que tingui en compte criteris socials i mediambientals.

El present projecte haurà d'implementar un portal web que permeti cercar les ubicacions dels diferents grups de consum agroecològic, així com dels diferents productors, de les fires i mercats de productes ecològics, dels horts urbans, etc. No es tracta de fer només una eina de consulta que permeti filtrar la informació i mostrar-la de manera georeferenciada, sinó que des d'aquest portal s'haurà de poder gestionar la base de dades amb la informació relacionada. Aquesta informació serà majoritàriament alfanumèrica però també podrà ser gràfica, de manera que permetrà editar l'àmbit geogràfic de les granges i territoris de conreu propis d'una determinada explotació agrícola o bé dels horts urbans. La integració de la informació espacial amb les bases de dades es farà gràcies al servidor de mapes GeoServer que permet l'edició de dades geoespacionals fent servir l'estàndard Web Feature Service Transactional (WFS-T) de l'Open Geospatial Consortium (OGC). Aquestes dades, ja siguin alfanumèriques o espacionals, les emmagatzemarem en una base de dades PostgreSQL/PostGIS, que ens proporcionarà les eines d'anàlisi espacionals necessàries pel projecte.

5.1.2. Objectius del PFC

En acabar aquest projecte de final de carrera, s'espera haver assolit els següents objectius:

- Comprendre els conceptes de la tecnologia SIG i la seva metodologia.
- Conèixer l'estructura dels diferents tipus de dades amb què treballa un SIG.

- Conèixer els sistemes d'emmagatzemament estàndards, tant d'informació ràster com vectorial, i ser capaç d'ubicar la informació en les coordenades que correspongui.
- Trobar, generar i manipular dades geogràfiques.
- Saber plantejar un projecte SIG.

El projecte SIG que cal desenvolupar té els següents objectius:

- Conèixer l'estat actual de l'ecosistema d'aplicacions lliures, i en particular les que corresponen als servidors de mapes.
- Treballar amb dades d'OpenStreetMap.
- Treballar amb una base de dades espacial com PostGIS.
- Treballar amb l'estàndard WFS-T amb Geoserver.
- Implementar un visor web.

5.1.3. Requeriments mínims

Aquests son els requeriments mínims que es demanen:

- Consultes filtrades per àmbit geogràfic de XCS, productors, fires, mercats i altres events relacionats, horts urbans, etc.
- Consultes dels productors associats a una determinada XCS i mostrar la seva ubicació geogràfica.
- Consulta alfanumèrica i geogràfica dels productors segons el tipus de producte que n'ofereixen.
- Consulta dels productors que han participat en alguna fira concreta.
- Manteniment (altes/baixes/modificacions) de la base de dades alfanumèrica que relaciona tota aquesta informació i poder editar gràficament, seguint l'estàndard WFS-T, els polígons corresponents a la ubicació dels àmbits de treballs dels productors i d'alguns horts urbans.

5.1.4. Enfocament i mètode seguit

El projecte s'ha dividit en quatre grans mòduls:

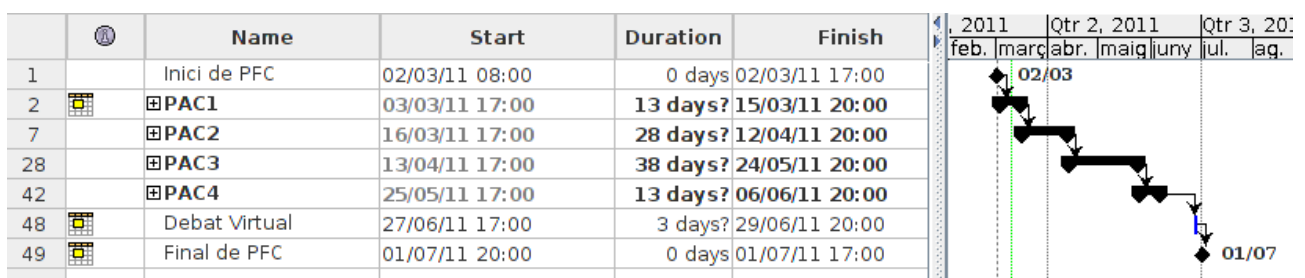
1. Pla de treball.
2. Instal·lació de programari, recerca de dades i disseny de base de dades.
3. Disseny i desenvolupament del programari.
4. Creació de la presentació virtual.

Tots els mòduls són molt importants per finalitzar correctament el projecte i tenen una gran dependència entre ells. És per això que s'haurà de donar la mateixa importància a tots ells, tot i que òbviament no tots requereixen el mateix esforç.

5.1.5. Planificació del projecte

La planificació del projecte, és la que vam indicar en el pla de treball i que a grans trets, es divideix en diferents mòduls, on cada un d'aquests mòduls corresponen al lliurament d'una PAC. L'explicació de cada un d'aquests mòduls i la seva planificació temporal és aquesta:

- PAC1** Lectura de l'enunciat i creació del pla de treball.
- PAC2** Instal·lació de programari, recerca de dades, disseny de base de dades e importació de dades.
- PAC3** Desenvolupament de l'aplicació.
- PAC4** Revisió de la memòria i creació de la presentació virtual.



5.1.5. Productes obtinguts

Els productes obtinguts d'aquest projecte són bàsicament aquests:

- Memòria del projecte. On es detallarà tota la feina que s'ha realitzat en el projecte i les decisions que s'han pres.
- Programari desenvolupat. En aquest cas es lliurarà el fitxer .war del programari desenvolupat i el seu codi font. També es lliuraran scripts de base de dades i altre tipus d'informació que es cregui pertinent.
- Manual d'instal·lació del programari desenvolupat.
- Presentació virtual. Presentació amb imatge i so, on farem un resum de la memòria del projecte i una petita demostració del programari desenvolupat.

5.1.6. Descripció dels altres capítols de la memòria

A continuació indiquem, en una breu descripció, el que seran la resta de capítols de la memòria:

- Capítol 2 – Instal·lació de programari: En aquest capítol explicarem tot el que hem fet per instal·lar el programari necessari.
- Capítol 3 – Recerca de dades: Aquí explicarem el procediment que hem seguit, per tal de trobar dades per fer un petit joc de proves. També indicarem d'on hem extret la cartografia de referència.

- Capítol 4 – Disseny de base de dades: En aquest capítol, definirem el disseny lògic i físic que tindrà la nostra base de dades.
- Capítol 5 – Importació de dades: En aquest apartat indicarem la forma en que hem importat les dades, del nostre joc de proves a la base de dades i també com hem incorporat la cartografia de referència al servidor GeoServer.
- Capítol 6 – Creació de l'entorn de desenvolupament: En aquest capítol, indicarem com hem instal·lat i configurat el nostre entorn de desenvolupament. També explicarem quina decisió hem pres en quant a la llibreria per desenvolupar el programari.
- Capítol 7 – Disseny de l'aplicació: Explicarem les decisions preses, respecte al disseny de l'aplicació, adjuntant també els diagrames UML corresponents.
- Capítol 8 – Implementació: En aquesta capítol detallarem tots els aspectes destacables de la implementació de l'aplicació.
- Capítol 9 – Línies futures de desenvolupament: Aquí explicarem tots els aspectes que es poden millorar en el projecte.
- Capítol 10 – Conclusions: Conclusions obtingudes al desenvolupar el projecte.

5.2. Capítol 2. Instal·lació de programari

En aquest capítol, descriurem el procés que hem seguit, per tal d'instal·lar tot el programari que necessitarem, tant pel desenvolupament com pel funcionament de l'aplicació.

Tot el programari s'ha instal·lat en un Sistema Operatiu Ubuntu Linux 10.10. Donem per suposat que ja disposem del JDK de Java instal·lat. (en el meu cas OpenJDK 1.6.0).

5.2.1. Instal·lació Tomcat

Per desplegar Geoserver i la nostra aplicació, necessitarem un servidor de aplicacions Java/J2EE. En aquest cas hem triat segurament el més popular, que no és un altre que Apache Tomcat. La versió que hem triat és la 7.0.11 (veure [1]).

Per procedir a la instal·lació, únicament hem de descarregar el fitxer (core) i descomprimir-lo en un directori qualsevol. Aquest directori l'anomenarem d'ara en endavant *TOMCAT_HOME*.

Dins de *TOMCAT_HOME* hi ha un directori */bin*, en el que disposem dels scripts per inicialitzar i aturar el servidor (*startup.sh* i *shutdown.sh*).

5.2.2. Instal·lació Geoserver

El servidor Geoserver l'emprarem per publicar les dades cartogràfiques, disponibles en la nostra base de dades. La instal·lació es farà en el servidor Apache Tomcat que hem instal·lat en el pas anterior.

Per aquest PFC hem triat l'última versió estable, que és la 2.1 en format web archive (veure [2]).

Per la instal·lació, tan sols hem de copiar el fitxer .war que hem descarregat, al directori *TOMCAT_HOME/webapps*. Un cop copiat iniciarem Tomcat i veurem que es genera un directori */geoserver* dins de *TOMCAT_HOME*.

Per ultim crearem un nou usuari per tal d'accedir a Geoserver. Per fer això modificarem el fitxer *TOMCAT_HOME/geoserver/data/security/users.properties* i afegirem aquesta nova línia:

```
xcs=xcs,ROLE_ADMINISTRATOR
```

5.2.3. Instal·lació PostgreSQL i PostGIS

Per la instal·lació de PostgreSQL i PostGIS en Ubuntu 10.10, utilitzarem els repositoris oficials d'aquest sistema operatiu i el seu gestor de paquets (apt-get).

Per la instal·lació de PostgreSQL 8.4, executarem la següent comanda en un terminal:

```
sudo apt-get install postgresql postgresql-client postgresql-contrib pgadmin3
sudo apt-get install postgresql pgadmin3
```

Per la instal·lació de PostGIS 1.5, executarem aquesta comanda en un terminal:

```
sudo apt-get install postgresql-8.2-postgis
```

A continuació crearem una nova base de dades (l'anomenarem xcs) amb suport PostGIS. Per fer-ho executarem aquesta comanda en una línia de comandes:

```
su postgres
createdb my_db
createlang plpgsql my_db
psql -d my_db -f /usr/share/postgresql/8.4/contrib/postgis-1.5/postgis.sql
psql -d my_db -f /usr/share/postgresql/8.4/contrib/postgis-1.5/spatial_ref_sys.sql
psql -d my_db -f /usr/share/postgresql/8.4/contrib/postgis_comments.sql
```

Per ultim, crearem un nou usuari a la base de dades, que anomenarem xcs i que és el que utilitzarem en la nostra aplicació. Per la creació del usuari, executarem aquestes comandes en un terminal:

```
adduser xcs
passwd xcs
su postgres
psql -d xcs -U postgres
```

Aquesta última comanda, obrirà una connexió amb la nostra base de dades. El que farem ara és executar la següent sentència sql:

```
CREATE USER xcs WITH PASSWORD 'xcs';
```

Podem ampliar informació sobre el procés d'instal·lació i configuració, en les referències [3], [4] i [5] de la bibliografia.

5.3. Capítol 3. Recerca de dades

En aquest capítol explicarem la recerca que hem realitzat, tant per obtenir les dades de cooperatives de consum, productors, etc., que formaran el nostre joc de proves, com les dades cartogràfiques de referència que utilitzarem en la nostra aplicació.

5.3.1. Recerca de fonts de dades

Per tal de tenir un petit joc de proves, amb el que puguem provar l'aplicació, hem fet una recerca de dades de cooperatives de consum, els seus productors, productes, algun hort urbà i alguna fira o mercat.

Per fer la recerca ens centrarem en la zona de Barcelona. En aquesta zona hem trobat dues cooperatives *Xarxa de Consum Solidari* i *El Guaret* (veure [6] i [7]).

Al fer la recerca de productors, hem localitzat la pàgina de CCPAE (Consell Català de la Producció Agrària Ecològica). En aquesta pàgina trobarem informació de productors i productes (veure [8]).

També cercarem mercats i fires, per tal de tenir-ne algun en el nostre joc de proves inicial. Hem trobat dues pàgines, en les que hi ha informació sobre mercats setmanals i fires de Catalunya (veure [9] i [10]).

Amb tota aquesta informació en tindrem prou per el nostre petit joc de proves.

La relació entre xarxes de consum i productors, així com la relació entre productors i esdeveniments (mercats o fires) és totalment fictícia, ja que no hem trobat informació prou completa en aquest sentit.

5.3.2. Recerca de dades cartogràfiques

En aquesta ocasió farem una recerca de les dades cartogràfiques de referència, que posteriorment exportarem a Geoserver.

La idea inicial, era la de cercar les dades d'una àrea limitada, com és l'àrea de Barcelona. En la pàgina web de OpenStreetMap tenim una opció d'exportació, que ens permet exportar una àrea del mapa, a un fitxer xml del tipus OpenStreetMap (extensió osm). El principal problema, és que aquesta exportació està força limitada i només ens deixa exportar una àrea força petita. Per tant tindríem que fer la exportació en varies parts, que és una mica feixuc.

Per això després d'una recerca de formes d'exportació, hem decidit utilitzar la cartografia de referència que podem trobar a Cloudmade.

Cloudmade és una pàgina web, on podem descarregar la cartografia de OpenStreetMap de tot el món en un únic fitxer o bé podem descarregar la cartografia d'una zona concreta. En aquest cas

descarregarem la cartografia de Catalunya (veure [11]).

5.4. Capítol 4. Disseny de base de dades

A partir de les funcionalitats que ha de tenir la nostra aplicació, definirem el disseny de base de dades necessari pel funcionament de la mateixa.

En aquest capítol definirem aquest disseny de base de dades, indicant el diagrama d'Entitat-Relació corresponent i posteriorment definint el disseny físic propi de PostgreSQL.

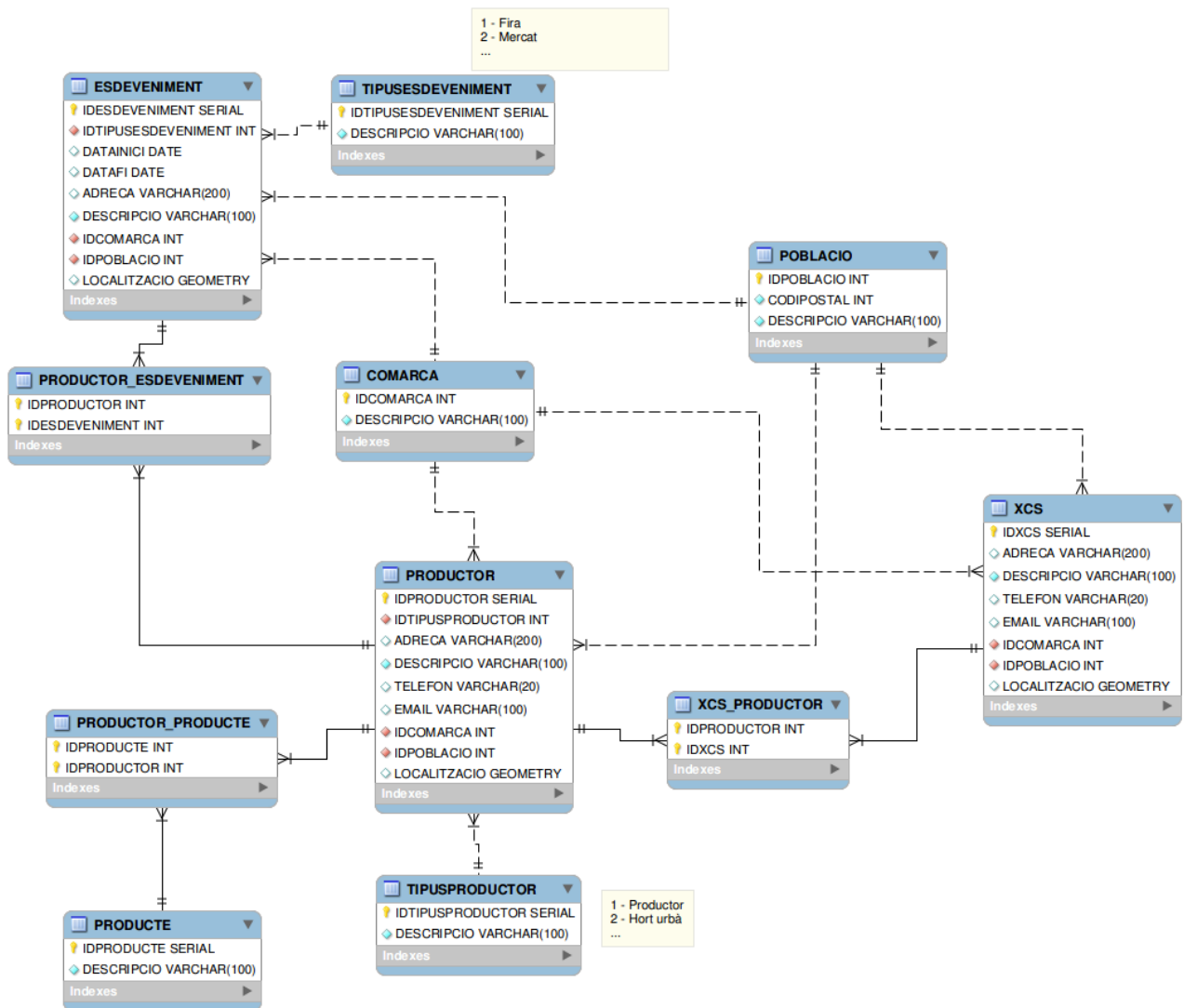
5.4.1 Disseny lògic de base de dades

Per la realització d'aquest diagrama Entitat-Relació, hem utilitzat el programari MySQL Workbench 5.2 (veure [12]).

Aquest programari està pensat per a ser utilitzar en bases de dades MySQL, però disposa d'un apartat de modelat de dades, que ens ha agradat força i que és totalment vàlid per les nostres necessitats.

Es tracta d'un programari amb llicència GPL i disposa d'un paquet propi per Ubuntu, amb el que la instal·lació és directe.

En la imatge numero 1 podem veure el diagrama corresponent al disseny lògic de la base de dades.



Il·lustració 1: Diagrama ER de base de dades

Anem a continuació a fer una descripció més detallada de cada una de les taules:

| COMARCA | | | | | |
|--|--------------|----------------------------------|----|----|------------|
| Informació de les comarques de Catalunya. Aquesta taula es carregarà d'inici i no es podrà modificar utilitzant l'aplicació. | | | | | |
| Columna | Típus | Descripció | PK | FK | Permet Nul |
| IDCOMARCA | INT | Identificador únic de la comarca | Si | No | No |
| DESCRIPCIO | VARCHAR(100) | Nom de la comarca | No | No | No |

| POBLACIÓ | | | | | |
|---|--------------|-----------------------------------|----|----|------------|
| Informació de les poblacions de Catalunya. Aquesta taula es carregarà d'inici i no es podrà modificar utilitzant l'aplicació. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDCOMARCA | INT | Identificador únic de la població | Si | No | No |
| CODIPOSTAL | INT | Codi postal de la població | No | No | No |
| DESCRIPCIO | VARCHAR(100) | Nom de la comarca | No | No | No |

| TIPUSESDEVENIMENT | | | | | |
|--|--------------|---|----|----|------------|
| Identifica els tipus de esdeveniments que es poden donar en el context de l'aplicació. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDTIPUSESDEVENIMENT | SERIAL | Identificador únic del tipus de esdeveniment | Si | No | No |
| DESCRIPCIO | VARCHAR(100) | Descripció del tipus d'esdeveniment (mercat, fira, etc) | No | No | No |

| TIPUSPRODUCTOR | | | | | |
|---|--------------|---|----|----|------------|
| Identifica els tipus de productors que es poden donar en el context de l'aplicació. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDTIPUSPRODUCTOR | SERIAL | Identificador únic del tipus de productor | Si | No | No |
| DESCRIPCIO | VARCHAR(100) | Descripció del tipus de productor (productor, hort urbà, etc) | No | No | No |

| PRODUCTE | | | | | |
|--|--------------|--|----|----|------------|
| Desa informació dels diferents productes que pot produir un productor. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDPRODUCTE | SERIAL | Identificador incremental del producte | Si | No | No |
| DESCRIPCIO | VARCHAR(100) | Descripció del producte | No | No | No |

| PRODUCTOR | | | | | |
|--|--------------|---|----|----|------------|
| Desa informació dels diferents productors que poden formar part de una xarxa de consum solidari. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDPRODUCTOR | SERIAL | Identificador incremental del productor | Si | No | No |
| IDTIPUSPRODUCTOR | INT | Referència al tipus de productor | No | Si | No |
| ADRECA | VARCHAR(200) | Adreça del productor | No | No | Si |

| | | | | | |
|--------------|--------------|--|----|----|----|
| DESCRIPCIO | VARCHAR(100) | Descripció del productor | No | No | No |
| TELEFON | VARCHAR(20) | Telèfon de contacte | No | No | Si |
| EMAIL | VARCHAR(100) | Email de contacte | No | No | Si |
| IDCOMARCA | INT | Referencia a la comarca a la que pertany | No | Si | No |
| IDPOBLACIO | INT | Referencia a la població a la que pertany | No | Si | No |
| LOCALITZACIO | GEOMETRY | Referencia geoespacial de la seva situació | No | No | No |

XCS

Desa informació de les diferents xarxes de consum solidari que tenim registrades a la nostra aplicació.

| Columna | Tipus | Descripció | PK | FK | Permet Nul |
|--------------|--------------|---|----|----|------------|
| IDXCS | SERIAL | Identificador incremental de la xcs | Si | No | No |
| DESCRIPCIO | VARCHAR(100) | Descripció de la xcs | No | No | No |
| TELEFON | VARCHAR(20) | Telefon de contacte | No | No | Si |
| EMAIL | VARCHAR(100) | Email de contacte | No | No | Si |
| IDCOMARCA | INT | Referencia a la comarca a la que pertany | No | Si | No |
| IDPOBLACIO | INT | Referencia a la població a la que pertany | No | Si | No |
| LOCALITZACIO | GEOMETRY | Referencia espacial de la seva situació | No | No | No |

ESDEVENIMENT

Desa informació dels diferents esdeveniments en els que participen els productors.

| Columna | Tipus | Descripció | PK | FK | Permet Nul |
|---------------------|--------------|--|----|----|------------|
| IDESDEVENIMENT | SERIAL | Identificador incremental del esdeveniment | Si | No | No |
| IDTIPUSESDEVENIMENT | INT | Referencia al tipus de esdeveniment | No | Si | No |
| DATAINICI | DATE | Data de inici del esdeveniment | No | No | No |
| DATAFI | DATE | Data de fi del esdeveniment | No | No | No |
| ADRECA | VARCHAR(200) | Adreça del esdeveniment | No | No | Si |
| DESCRIPCIO | VARCHAR(100) | Descripció de l'esdeveniment | No | No | No |
| IDCOMARCA | INT | Referencia a la comarca a la que pertany | No | Si | No |
| IDPOBLACIO | INT | Referencia a la població a la que pertany | No | Si | No |

| | | | | | |
|--------------|----------|---|----|----|----|
| LOCALITZACIO | GEOMETRY | Referencia espacial de la seva situació | No | No | No |
|--------------|----------|---|----|----|----|

| PRODUCTOR_PRODUCTE | | | | | |
|--|-------|-------------------------|----|----|------------|
| Relaciona productors amb els seus productes. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDPRODUCTE | INT | Referencia al producte | Si | Si | No |
| IDPRODUCTOR | INT | Referencia al productor | Si | Si | No |

| PRODUCTOR_ESDEVENIMENT | | | | | |
|---|-------|-----------------------------|----|----|------------|
| Relaciona productors amb els esdeveniment en els que ha participat. | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDESDEVENIMENT | INT | Referencia a l'esdeveniment | Si | Si | No |
| IDPRODUCTOR | INT | Referencia al productor | Si | Si | No |

| XCS_PRODUCTOR | | | | | |
|---|-------|-------------------------|----|----|------------|
| Relaciona xcs amb els productors que hi pertanyen | | | | | |
| Columna | Tipus | Descripció | PK | FK | Permet Nul |
| IDXCS | INT | Referencia al xcs | Si | Si | No |
| IDPRODUCTOR | INT | Referencia al productor | Si | Si | No |

5.4.2. Disseny físic de base de dades

En aquest apartat indicarem el disseny físic de base de dades per a PostgreSQL. Tenim tres apartats ben definits:

- Esborrat de les taules, si existeixen.
- Creació de les taules.
- Donar permís a l'usuari xcs per a totes les taules.

Podrem veure que s'han pres les següents decisions:

- Per les dades espacials utilitzarem el tipus GEOMETRY amb una dimensió de 2 (per desar coordenades x,y). S'utilitza la projecció de Mercator.
- Per les taules que es mantenen utilitzant l'aplicació, hem utilitzat un identificador de tipus SERIAL, oblidant-nos així de duplicats per clau primària, ja que és la base de dades la que

ens proporciona l'identificador.

- En les taules que es carreguen un sol cop a l'inici, l'identificador és de tipus INTEGER, ja que aquestes taules no seran modificades per l'aplicació.

El disseny físic de base de dades quedarà així:

```
-- Esborrem les taules si existeixen
DROP TABLE IF EXISTS productor_producto;
DROP TABLE IF EXISTS productor_esdeveniment;
DROP TABLE IF EXISTS xcs_productor;
DROP TABLE IF EXISTS producte;
DROP TABLE IF EXISTS productor;
DROP TABLE IF EXISTS xcs;
DROP TABLE IF EXISTS esdeveniment;
DROP TABLE IF EXISTS tipusesdeveniment;
DROP TABLE IF EXISTS tipusproductor;
DROP TABLE IF EXISTS comarca;
DROP TABLE IF EXISTS poblacio;

-- Creem les taules
CREATE TABLE comarca (
    idcomarca INTEGER CONSTRAINT pk_comarca PRIMARY KEY,
    descripcio VARCHAR(100) NOT NULL
);

CREATE TABLE poblacio (
    idpoblacio INTEGER CONSTRAINT pk_poblacio PRIMARY KEY,
    codipostal INTEGER NOT NULL,
    descripcio VARCHAR(100) NOT NULL
);

CREATE TABLE tipusesdeveniment (
    idtipusesdeveniment SERIAL CONSTRAINT pk_tipusesdeveniment PRIMARY KEY,
    descripcio VARCHAR(100) NOT NULL
);

CREATE TABLE tipusproductor (
    idtipusproductor SERIAL CONSTRAINT pk_tipusproductor PRIMARY KEY,
    descripcio VARCHAR(100) NOT NULL
);

CREATE TABLE producte (
    idproducte SERIAL CONSTRAINT pk_producte PRIMARY KEY,
    descripcio VARCHAR(100) NOT NULL
);

CREATE TABLE productor (
    idproductor SERIAL CONSTRAINT pk_productor PRIMARY KEY,
    idtipusproductor INTEGER CONSTRAINT fk_productor_tipusproductor REFERENCES
tipusproductor(idtipusproductor) NOT NULL,
    adreca VARCHAR(200),
    descripcio VARCHAR(100) NOT NULL,
    telefon VARCHAR(20),
```

```

        email VARCHAR(100),
        idcomarca INTEGER CONSTRAINT fk_productor_comarca REFERENCES comarca(idcomarca)
NOT NULL,
        idpoblacio INTEGER CONSTRAINT fk_productor_poblacio REFERENCES
poblacio(idpoblacio) NOT NULL
    );
SELECT AddGeometryColumn('productor', 'localitzacio',900913 , 'GEOMETRY', 2);

CREATE TABLE xcs (
    idxcs SERIAL CONSTRAINT pk_xcs PRIMARY KEY,
    adreca VARCHAR(200),
    descripcio VARCHAR(100) NOT NULL,
    telefon VARCHAR(20),
    email VARCHAR(100),
    idcomarca INTEGER CONSTRAINT fk_xcs_comarca REFERENCES comarca(idcomarca) NOT
NULL,
    idpoblacio INTEGER CONSTRAINT fk_xcs_poblacio REFERENCES poblacio(idpoblacio) NOT
NULL
);
SELECT AddGeometryColumn('xcs', 'localitzacio',900913 , 'GEOMETRY', 2);

CREATE TABLE esdeveniment (
    idesdeveniment SERIAL CONSTRAINT pk_esdeveniment PRIMARY KEY,
    idtipusesdeveniment INTEGER CONSTRAINT fk_esdeveniment_tipusesdeveniment
REFERENCES tipusesdeveniment(idtipusesdeveniment) NOT NULL,
    datainici DATE,
    datafi DATE,
    adreca VARCHAR(200),
    descripcio VARCHAR(100) NOT NULL,
    idcomarca INTEGER CONSTRAINT fk_esdeveniment_comarca REFERENCES comarca(idcomarca)
NOT NULL,
    idpoblacio INTEGER CONSTRAINT fk_esdeveniment_poblacio REFERENCES
poblacio(idpoblacio) NOT NULL
);
SELECT AddGeometryColumn('esdeveniment', 'localitzacio',900913 , 'GEOMETRY', 2);

CREATE TABLE productor_producte (
    idproducte INTEGER NOT NULL,
    idproductor INTEGER NOT NULL,
    CONSTRAINT pk_productor_producte PRIMARY KEY(idproducte, idproductor),
    CONSTRAINT fk_pproducte_producte FOREIGN KEY(idproducte) REFERENCES
producte(idproducte),
    CONSTRAINT fk_pproducte_productor FOREIGN KEY(idproductor) REFERENCES
productor(idproductor)
);

CREATE TABLE productor_esdeveniment (
    idesdeveniment INTEGER NOT NULL,
    idproductor INTEGER NOT NULL,
    CONSTRAINT pk_productor_esdeveniment PRIMARY KEY(idesdeveniment, idproductor),
    CONSTRAINT fk_pesdeveniment_esdeveniment FOREIGN KEY(idesdeveniment) REFERENCES
esdeveniment(idesdeveniment),
    CONSTRAINT fk_pesdeveniment_productor FOREIGN KEY(idproductor) REFERENCES
productor(idproductor)
);

CREATE TABLE xcs_productor (
    idxcs INTEGER NOT NULL,
    idproductor INTEGER NOT NULL,
    CONSTRAINT pk_xcs_productor PRIMARY KEY(idxcs, idproductor),

```

```

        CONSTRAINT fk_xproductor_xcs FOREIGN KEY(idxc) REFERENCES xcs(idxc),
        CONSTRAINT fk_xproductor_productor FOREIGN KEY(idproductor) REFERENCES
productor(idproductor)
);

-- Donem permisos totals al usuari xcs per aquestes taules
GRANT ALL PRIVILEGES ON TABLE producte TO xcs;
GRANT ALL PRIVILEGES ON TABLE productor TO xcs;
GRANT ALL PRIVILEGES ON TABLE xcs TO xcs;
GRANT ALL PRIVILEGES ON TABLE esdeveniment TO xcs;
GRANT ALL PRIVILEGES ON TABLE tipusesdeveniment TO xcs;
GRANT ALL PRIVILEGES ON TABLE tipusproductor TO xcs;
GRANT ALL PRIVILEGES ON TABLE comarca TO xcs;
GRANT ALL PRIVILEGES ON TABLE poblacio TO xcs;
GRANT ALL PRIVILEGES ON TABLE productor_producte TO xcs;
GRANT ALL PRIVILEGES ON TABLE productor_esdeveniment TO xcs;
GRANT ALL PRIVILEGES ON TABLE xcs_productor TO xcs;

-- Donem permisos també a les seqüències
GRANT ALL PRIVILEGES ON SEQUENCE esdeveniment_idesdeveniment_seq TO xcs;
GRANT ALL PRIVILEGES ON SEQUENCE producte_idproducte_seq TO xcs;
GRANT ALL PRIVILEGES ON SEQUENCE productor_idproductor_seq TO xcs;
GRANT ALL PRIVILEGES ON SEQUENCE tipusesdeveniment_idtipusesdeveniment_seq TO xcs;
GRANT ALL PRIVILEGES ON SEQUENCE tipusproductor_idtipusproductor_seq TO xcs;
GRANT ALL PRIVILEGES ON SEQUENCE xcs_idxcs_seq TO xcs;

```

5.5. Capítol 5. Importació de dades

En aquest capítol explicarem la forma en que importarem les dades, obtingudes de la recerca de xarxes de consum, a la nostra base de dades.

També indicarem com importarem les dades cartogràfiques de referencia, al nostre servidor local GeoServer.

5.5.1. Importació a base de dades

En aquest apartat definirem el script de base de dades, que utilitzarem per carregar el nostre joc de proves inicial a la base de dades. Aquestes son les dades que hem obtingut del procés de recerca d'informació de xarxes de consum solidari.

El script d'importació serà el següent:

```

-- Esborrem el contingut de les taules
TRUNCATE TABLE xcs_productor RESTART IDENTITY CASCADE;
TRUNCATE TABLE productor_esdeveniment RESTART IDENTITY CASCADE;
TRUNCATE TABLE productor_producte RESTART IDENTITY CASCADE;
TRUNCATE TABLE xcs RESTART IDENTITY CASCADE;
TRUNCATE TABLE producte RESTART IDENTITY CASCADE;
TRUNCATE TABLE productor RESTART IDENTITY CASCADE;
TRUNCATE TABLE esdeveniment RESTART IDENTITY CASCADE;
TRUNCATE TABLE tipusesdeveniment RESTART IDENTITY CASCADE;
TRUNCATE TABLE tipusproductor RESTART IDENTITY CASCADE;
TRUNCATE TABLE comarca RESTART IDENTITY CASCADE;

```

```

TRUNCATE TABLE poblacio RESTART IDENTITY CASCADE;

-- Contingut taula comarca
INSERT INTO comarca(idcomarca,descripcio) VALUES(1, 'Barcelonès');
-- Contingut taula poblacio
INSERT INTO poblacio(idpoblacio,codipostal,descripcio) VALUES(1, 08911, 'Badalona');
INSERT INTO poblacio(idpoblacio,codipostal,descripcio) VALUES(2, 08080, 'Barcelona');
INSERT INTO poblacio(idpoblacio,codipostal,descripcio) VALUES(3, 08908, 'l''Hospitalet
de Llobregat');
INSERT INTO poblacio(idpoblacio,codipostal,descripcio) VALUES(4, 08930, 'Sant Adrià de
Besòs');
INSERT INTO poblacio(idpoblacio,codipostal,descripcio) VALUES(5, 08921, 'Santa Coloma de
Gramenet');
-- Contingut taula tipusesdeveniment
INSERT INTO tipusesdeveniment(descripcio) VALUES('Mercat');
INSERT INTO tipusesdeveniment(descripcio) VALUES('Fira');
-- Contingut taula tipusproductor
INSERT INTO tipusproductor(descripcio) VALUES('Productor');
INSERT INTO tipusproductor(descripcio) VALUES('Hort Urbà');
-- Contingut taula producte
INSERT INTO producte(descripcio) VALUES ('Pa');
INSERT INTO producte(descripcio) VALUES ('Oli de Girasol');
INSERT INTO producte(descripcio) VALUES ('Cereals');
INSERT INTO producte(descripcio) VALUES ('Cervesa');
INSERT INTO producte(descripcio) VALUES ('Mel');
INSERT INTO producte(descripcio) VALUES ('Aloe Vera');
INSERT INTO producte(descripcio) VALUES ('Carn de Vedella');
INSERT INTO producte(descripcio) VALUES ('Te');
INSERT INTO producte(descripcio) VALUES ('Brou Vegetal');
INSERT INTO producte(descripcio) VALUES ('Propolis');
INSERT INTO producte(descripcio) VALUES ('Hortalisses');
-- Contingut taula xcs
INSERT INTO xcs(adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitzacio)
VALUES('Pl. Sant Agustí Vell, 15','Xarxa Consum
Solidari','932682202','ciutatvella@xarxaconsum.org',1,2,
ST_GeomFromText('POINT(242739.942039089 5069730.74132601)', 900913));
INSERT INTO xcs(adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitzacio)
VALUES('C. Padilla, 210','El Guaret',null,'cooperativa.elguaret@gmail.com',1,2,
ST_GeomFromText('POINT(241686.859656185 5073040.10395828)', 900913));
-- Contingut taula productor
INSERT INTO
productor(idtipusproductor,adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitz
acio) VALUES(
(SELECT idtipusproductor FROM tipusproductor WHERE descripcio = 'Productor'),
'Plaça Joan Cornulleda, 23','Mapryser,
S.L.','934270881','victor.quilez@mapryser.com',1,2,
ST_GeomFromText('POINT(239159.907215176 5076635.68540295)', 900913));
INSERT INTO
productor(idtipusproductor,adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitz
acio) VALUES(
(SELECT idtipusproductor FROM tipusproductor WHERE descripcio = 'Productor'),
'C. Tuset, 9','Brenntag Química, S.A.','932184404','ganton@brenntag.es',1,2,
ST_GeomFromText('POINT(239328.66756322 5071189.24377264)', 900913));
INSERT INTO
productor(idtipusproductor,adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitz
acio) VALUES(
(SELECT idtipusproductor FROM tipusproductor WHERE descripcio = 'Productor'),
'C. Casp, 6','Alta Química, S.A.','932656012','altaquimica@altaquimica.com',1,2,
ST_GeomFromText('POINT(242287.984906469 5070609.1699428)', 900913));
INSERT INTO

```

```

productor(idtipusproductor,adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitzacio) VALUES (
  (SELECT idtipusproductor FROM tipusproductor WHERE descripcio = 'Productor'),
  'Ronda Sant Antoni, 96','Jaume Bertran
Garriga', '933018037', 'info@formmistrall.com',1,2,
  ST_GeomFromText('POINT(240834.152356707 5068861.29447035)', 900913));
INSERT INTO
productor(idtipusproductor,adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitzacio) VALUES (
  (SELECT idtipusproductor FROM tipusproductor WHERE descripcio = 'Productor'),
  'C. Guitard, 70','Mielar, S.A.', '933220125', 'mielar@mielar.com',1,2,
  ST_GeomFromText('POINT(237995.50534148 5068836.07277881)', 900913));
INSERT INTO
productor(idtipusproductor,adreca,descripcio,telefon,email,idcomarca,idpoblacio,localitzacio) VALUES (
  (SELECT idtipusproductor FROM tipusproductor WHERE descripcio = 'Hort Urbà'),
  'Plaça del Pou de la Figuera','L''Hortet del Forat',null,null,1,2,
  ST_GeomFromText('POINT(242640.42241432 5069646.46386682)', 900913));
-- Contingut taula esdeveniment
INSERT INTO
esdeveniment(idtipusesdeveniment,datainici,datafi,adreca,descripcio,idcomarca,idpoblacio,localitzacio) VALUES (
  (SELECT idtipusesdeveniment FROM tipusesdeveniment WHERE descripcio = 'Mercat'),
  null,null,null, 'Mercat General (Dissabtes)',1,1,
  ST_GeomFromText('POINT(248419.46245936 5081144.67633184)', 900913));
INSERT INTO
esdeveniment(idtipusesdeveniment,datainici,datafi,adreca,descripcio,idcomarca,idpoblacio,localitzacio) VALUES (
  (SELECT idtipusesdeveniment FROM tipusesdeveniment WHERE descripcio = 'Fira'),
  '13/06/2011','14/06/2011', 'Fira Gran Via', 'Snackex',1,3,
  ST_GeomFromText('POINT(237221.834880467 5064667.98127206)', 900913));
-- Contingut taula xcs_productor
INSERT INTO xcs_productor(idxc, idproductor) VALUES (
  (SELECT idxc FROM xcs WHERE descripcio = 'Xarxa Consum Solidari'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Mapryser, S.L.'));
INSERT INTO xcs_productor(idxc, idproductor) VALUES (
  (SELECT idxc FROM xcs WHERE descripcio = 'Xarxa Consum Solidari'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Brenntag Química, S.A.'));
INSERT INTO xcs_productor(idxc, idproductor) VALUES (
  (SELECT idxc FROM xcs WHERE descripcio = 'Xarxa Consum Solidari'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Alta Química, S.A.'));
INSERT INTO xcs_productor(idxc, idproductor) VALUES (
  (SELECT idxc FROM xcs WHERE descripcio = 'El Guaret'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Jaume Bertran Garriga'));
INSERT INTO xcs_productor(idxc, idproductor) VALUES (
  (SELECT idxc FROM xcs WHERE descripcio = 'El Guaret'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Mielar, S.A.'));
INSERT INTO xcs_productor(idxc, idproductor) VALUES (
  (SELECT idxc FROM xcs WHERE descripcio = 'El Guaret'),
  (SELECT idproductor FROM productor WHERE descripcio = 'L''Hortet del Forat'));
-- Contingut taula productor_producto
INSERT INTO productor_producto(idproducte, idproductor) VALUES (
  (SELECT idproducte FROM producte WHERE descripcio = 'Oli de Girasol'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Mapryser, S.L.'));
INSERT INTO productor_producto(idproducte, idproductor) VALUES (
  (SELECT idproducte FROM producte WHERE descripcio = 'Cereals'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Brenntag Química, S.A.'));
INSERT INTO productor_producto(idproducte, idproductor) VALUES (
  (SELECT idproducte FROM producte WHERE descripcio = 'Oli de Girasol'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Alta Química, S.A.'));

```

```

INSERT INTO productor_producto(idproducto, idproductor) VALUES (
  (SELECT idproducto FROM producto WHERE descripcio = 'Pa'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Jaume Bertran Garriga'));
INSERT INTO productor_producto(idproducto, idproductor) VALUES (
  (SELECT idproducto FROM producto WHERE descripcio = 'Mel'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Mielar, S.A.'));
INSERT INTO productor_producto(idproducto, idproductor) VALUES (
  (SELECT idproducto FROM producto WHERE descripcio = 'Hortalisses'),
  (SELECT idproductor FROM productor WHERE descripcio = 'L''Hortet del Forat'));
-- Contingut taula productor_esdeveniment
INSERT INTO productor_esdeveniment(idesdeveniment, idproductor) VALUES (
  (SELECT idesdeveniment FROM esdeveniment WHERE descripcio = 'Mercat General
(Dissabtes)'),
  (SELECT idproductor FROM productor WHERE descripcio = 'L''Hortet del Forat'));
INSERT INTO productor_esdeveniment(idesdeveniment, idproductor) VALUES (
  (SELECT idesdeveniment FROM esdeveniment WHERE descripcio = 'Snackex'),
  (SELECT idproductor FROM productor WHERE descripcio = 'Brenntag Química, S.A.'));

```

5.5.2. Importació a GeoServer

En aquest apartat indicarem com hem realitzat la importació de les dades cartogràfiques de referència, al nostre servidor local GeoServer.

El primer serà executar la següent comanda en un terminal:

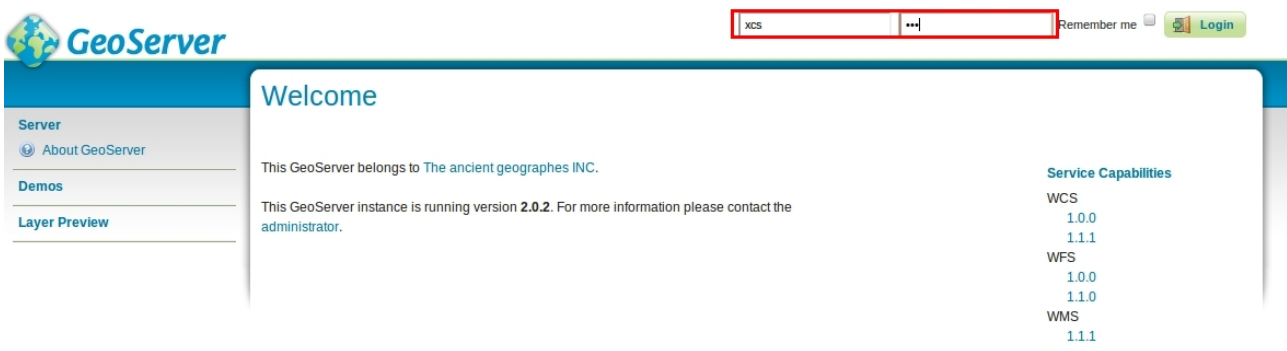
```

su postgres
osm2pgsql -E 900913 -d cataluna.osm

```

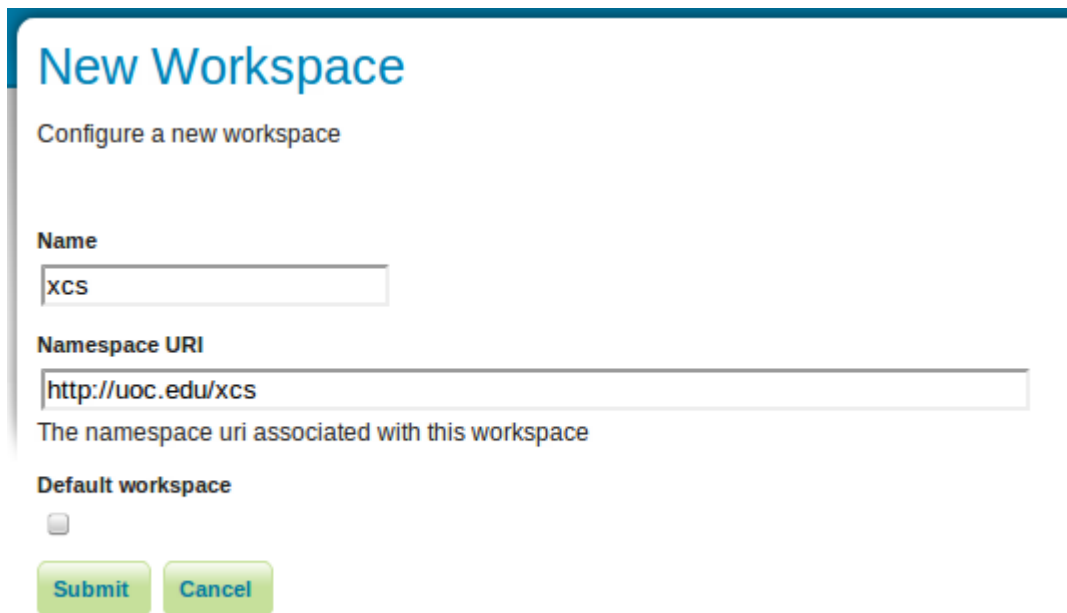
Aquesta comanda importarà les dades del fitxer osm, obtingut en la fase de recerca, a la nostra base de dades local, utilitzant la projecció Mercator, que és l'estàndard utilitzat per OpenStreetMap.

Un cop fet això, accedirem a la pàgina d'administració del nostre servidor GeoServer i ens autenticarem amb l'usuari xcs, que vam crear anteriorment. Podem veure un exemple en la imatge número 2.



Il·lustració 2: Login GeoServer

El primer pas, serà el d'anar a l'apartat “workspaces” i afegir un nou “workspace”, com el que indiquem en la imatge número 3.



New Workspace

Configure a new workspace

Name

Namespace URI

The namespace uri associated with this workspace

Default workspace

Il·lustració 3: Creació de un nou Workspace

A continuació, crearem un nou datasource, que apuntarà a la nostra base de dades PostgreSQL/PostGIS. Per això ens dirigirem a la secció “Stores” i triarem l’opció de creació d’un nou “Store”. Ens deixarà triar entre varies opcions i la nostra selecció serà PostGIS. Ja només restarà introduir les dades de connexió de la nostra base de dades. No ens oblidem de relacionar aquest nou datasource amb el workspace creat anteriorment. Podem veure un exemple en la imatge número 4.

New Vector Data Source

PostGIS
PostGIS Database

Basic Store Info

Workspace *

XCS ▼

Data Source Name *

xcs

Description

Enabled

Connection Parameters

dbtype *

postgis

host *

localhost

port *

5432

database

xcs

schema

public

user *

xcs

passwd

...

Il·lustració 4: Creació de un nou DataSource

Per ultim, anirem a la secció “Layers” i triarem l'opció d'afegir un nou Layer. Aquí tindrem que triar el nostre datasource (xcs:xcs). Un cop seleccionat el datasource, podrem veure les taules que conté la nostra base de dades. Tenim un exemple en la imatge número 5.

New Layer chooser

Add layer from

Here is a list of resources contained in the store 'xcs'. Click on the layer you wish to configure

<< < 1 > >> Results 0 to 0 (out of 0 items)

| Published | Layer name | |
|-----------|------------------------|---------|
| | comarca | Publish |
| | esdeveniment | Publish |
| | planet_osm_line | Publish |
| | planet_osm_point | Publish |
| | planet_osm_polygon | Publish |
| | planet_osm_roads | Publish |
| | poblacio | Publish |
| | producte | Publish |
| | productor | Publish |
| | productor_esdeveniment | Publish |
| | productor_producte | Publish |
| | tipusesdeveniment | Publish |
| | tipusproductor | Publish |
| | xcs | Publish |
| | xcs_productor | Publish |

<< < 1 > >> Results 0 to 0 (out of 0 items)

Il·lustració 5: Selecció de taules a publicar en GeoServer

Únicament publicarem cinc taules:

- planet_osm_line.
- planet_osm_roads.
- Esdeveniment.
- Productor.
- Xcs.

Las dos primeras taules corresponen als carrers i carreteres de la cartografia de referencia importada. Per fer la publicació, triarem la opció “Publish”. En la pantalla de confirmació de publicació, només tindrem que calcular els “Bounding Boxes”, triant les opcions “Compute from Data” i “Compute from Native Bounds”.

Les altres tres taules, corresponen a les dades de l'aplicació que contenen informació geogràfica i que necessiten ser afegides, per tal de poder treballar amb WFS-T. Per fer la publicació, seguirem el mateix procediment que en les altres dues taules.

Podem veure un exemple de publicació en la imatge número 6.

Coordinate Reference Systems

Native SRS

[EPSG:WGS84 / Google Mercator...](#)

Declared SRS

[EPSG:WGS84 / Google Mercator...](#)

SRS handling

▼

Bounding Boxes

Native Bounding Box

| Min X | Min Y | Max X | Max Y |
|--|---|---|--|
| <input type="text" value="17.789,67"/> | <input type="text" value="4.942.298,12"/> | <input type="text" value="369.981,35"/> | <input type="text" value="5.290.878"/> |

[Compute from data](#)

Lat/Lon Bounding Box

| Min X | Min Y | Max X | Max Y |
|-----------------------------------|-------------------------------------|------------------------------------|-------------------------------------|
| <input type="text" value="0,16"/> | <input type="text" value="40,523"/> | <input type="text" value="3,324"/> | <input type="text" value="42,861"/> |

[Compute from native bounds](#)

Il·lustració 6: Calcul de Bounding Boxes

Un cop fet això, ja tindrem la nostra cartografia publicada en el servidor GeoServer.

La informació per realitzar la importació de OpenStreetMap a Geoserver, l'hem extret del manual de GeoServer i altres fonts (veure [13] i [14]).

5.6. Capítol 6. Creació entorn desenvolupament

En aquest capítol instal·larem i crearem el que serà el nostre entorn de desenvolupament. També farem un petit estudi, per triar les llibreries que utilitzarem per desenvolupar la nostra aplicació.

5.6.1. Instal·lació Maven

Maven es un programari per la gestió de projectes, que ens permet gestionar les dependències d'un projecte d'una forma senzilla.

Totes les dependències que son necessàries pel projecte, es defineixen dins d'un fitxer pom.xml i és maven l'encarregat de descarregar les llibreries abans de compilar l'aplicació.

Utilitzarem la versió de Maven 2.2.1. Podem trobar l'enllaç per la seva descarrega en la referència número [21] de la bibliografia.

Per la seva instal·lació, només cal descomprimir el fitxer descarregat en el directori que vulguem.

Trobarem més informació per la instal·lació en l'enllaç de descarrega.

5.6.2. Instal·lació GWT

Google Web Toolkit és un framework de desenvolupament d'aplicacions web.

Utilitzarem aquest framework pel desenvolupament de la interfície de l'aplicació. En aquest cas, no es necessari realitzar cap tipus d'instal·lació, ja que la inclourem com a una dependència en el fitxer pom.xml de Maven, tal com podrem veure en apartats posteriors.

5.6.3. Instal·lació GXT

GXT o també conegut com a Ext GWT, és un component per a GWT que ens permetrà afegir components de ExtJS a la nostra aplicació.

Aquest complement ens permetrà construir unes interfícies més treballades i més atractives. Al igual que amb GWT, GXT no requereix cap tipus d'instal·lació, ja que únicament l'afegirem com a dependència al fitxer pom.xml de Maven.

Serà Maven l'encarregat de descarregar la llibreria e incloure-la en el nostre projecte.

5.6.4. Instal·lació GWT-Openlayers

GWT-Openlayers es un wrapper d'Openlayers per a GWT. Ens permetrà afegir mapes d'Openlayers a una aplicació desenvolupada en GWT.

Al igual que en els dos apartats anteriors, GWT-Openlayers no requereix cap tipus d'instal·lació, ja que deixarem la gestió d'aquesta llibreria en mans del programari Maven.

5.6.5. Instal·lació Spring Framework

Spring Framework proporciona múltiples serveis per aplicacions Java/J2EE, com per exemple la coneguda injecció de dependències, que gestiona les instàncies de cada classe i s'encarrega de passar les seves referències allà on calgui.

En el nostre cas, ens interessa per la part del seu suport a DAOs, ja que ens permet treballar amb la base de dades, d'una forma més fàcil que no pas amb JDBC de forma directa.

Com en els casos anteriors, només caldrà afegir aquesta llibreria com a dependència de Maven, per tal de que s'encarregui de gestionar la seva descarrega e inclusió en el projecte.

5.6.6. Instal·lació Eclipse

L'entorn de desenvolupament serà el conegut Eclipse, en la seva última versió per desenvolupadors J2EE (veure [15]).

També hem instal·lat el plugin de Maven, ja que utilitzarem maven per gestionar les llibreries del nostre projecte (veure [16]).

Per últim, instal·larem el plugin de GWT, que com veurem més endavant, és la llibreria que emprem per desenvolupar el nostre projecte (veure [17]). Instal·larem el plugin de GWT i el GWT

Designer.

Tots els plugins per Eclipse els hem instal·lat a través del gestor de plugins, que porta incorporat el mateix Eclipse.

5.6.7. Creació i configuració del projecte

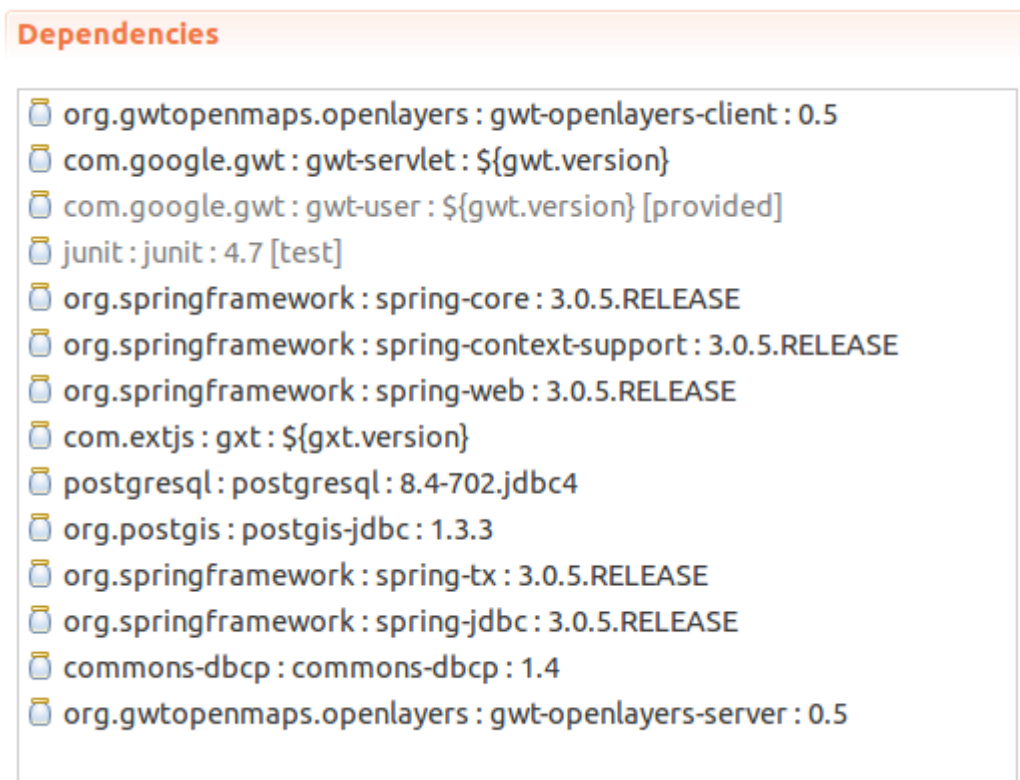
Per crear el projecte en Eclipse, crearem un nou projecte de tipus Maven i seleccionarem com a “Archetype” el tipus “gwt-maven-plugin”. Això ens generarà un nou projecte d'Eclipse, gestionat per Maven i amb les dependències necessàries per desenvolupar amb GWT.

A més de les dependències pròpies de GWT, afegirem la llibreria de PostgreSQL per accedir a la base de dades per JDBC. També afegirem les llibreries de Spring, per aprofitar la part relativa a DAOs i a JDBC.

També afegirem les dependències pròpies de GXT, que és una extensió per GWT del conegut ExtJS, que ens aportarà més elements (widgets) i de més qualitat.

Per últim afegirem les dependències a la llibreria gwt-openlayers, que integra openlayers dins de GWT i que necessitem per implementar el visor de mapes.

Totes aquestes dependències es defineixen en un fitxer pom.xml, que està inclòs dins del projecte i que utilitza Maven per afegir les llibreries que requerim al nostre projecte. Finalment les dependències definides són les que podem veure en la imatge número 7.



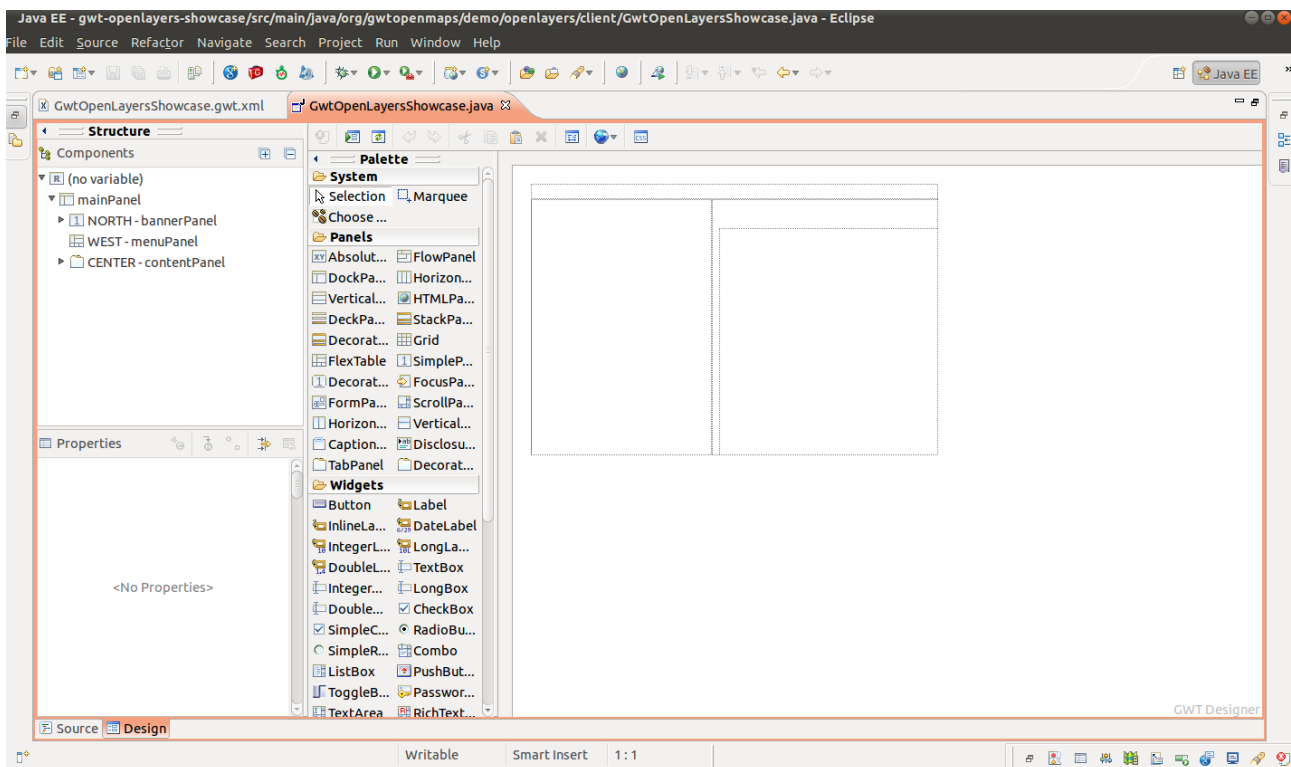
Il·lustració 7: Dependències del projecte XCS

5.6.8. Estudi de llibreries

Com podem deduir dels apartats anteriors, finalment ens hem decidit a utilitzar la llibreria GWT, per el nostre desenvolupament. Les raons d'aquesta decisió son las següents:

- El desenvolupament és més fàcil i ràpid, ja que tot es desenvolupa en Java.
- Permet comunicació Client-Servidor Ajax d'una forma molt senzilla.
- GWT transforma el codi Java en una combinació de HTML/Javascript, compatible amb la majoria de navegadors del mercat. D'aquesta forma no tenim que treballar directament amb codi HTML/Javascript.
- Disposem d'un dissenyador visual, que permet crear la pàgina de una forma senzilla i rapida.
- Tenim una integració amb openlayers, anomenada gwt-openlayers, que en una primera avaluació ens va semblar que tenia implementat tot el que necessitàvem (WMS, WFS, WFS-T). La realitat és que en el moment de la implementació, ens hem trobat amb algunes mancances, sobretot en la part de WFS-T, que ens han obligat a fer una petita extensió d'aquest projecte.
- Tenim una integració amb ExtJs, anomenada gxt, que ens aportara els components ja coneguts de ExtJs, però permetent-nos fer el desenvolupament a l'estil GWT.
- Es tracta d'un framework desenvolupat, impulsat i molt utilitzat per Google, que ha utilitzat en el desenvolupament de moltes de les seves conegudes aplicacions web, com per exemple Google AdWords i Orkut (veure referencies [18] i [19]).
- GWT és un projecte OpenSource amb la llicencia Apache i totalment compatible amb altres llicencies OpenSource.

Podem veure una captura del dissenyador visual de GWT en la imatge número 8.



Il·lustració 8: Dissenyador visual de GWT

5.7. Capítol 7. Disseny Aplicació

En aquest capítol, dedicarem un apartat als estàndards utilitzats en l'aplicació de Xarxes de Consum Solidari. També dedicarem un apartat a l'arquitectura general de l'aplicació. Per acabar, indicarem el disseny final de l'aplicació, que utilitzarem com a base per a la implementació del programari.

5.7.1. Estàndards utilitzats

En aquest apartat, donarem una descripció dels estàndards que utilitzarem en la nostra aplicació, que han estat elaborats per l'OGC.

L'OGC (Open Geospatial Consortium) és un consorci format per 421 companyies, agències governamentals i universitats. El seu objectiu és la definició d'estàndards oberts e interoperables, dins dels Sistemes d'Informació Geogràfica i de la web. Persegueix acords entre les diferents empreses del sector, que possibiliten la interoperabilitat dels seus sistemes de geoprocessament i facilitar l'intercanvi d'informació geogràfica.

En la nostra aplicació utilitzarem de forma directa, dos dels estàndards definits per l'OGC, que són l'estàndard WMS (Web Map Service) i l'estàndard WFS (Web Feature Service) i de forma indirecta utilitzarem l'estàndard GML (Geography Markup Language).

GML (Geography Markup Language)

El llenguatge GML és un tipus de gramàtica XML, que serveix per representar informació geogràfica. GML serveix per intercanviar informació entre sistemes geogràfics a través d'Internet. Els usuaris i desenvolupadors poden representar amb aquest llenguatge, conjunts de dades que continguin punts, línies i polígons.

No utilitzarem el llenguatge GML de forma directa en la nostra aplicació, si no que s'utilitza de forma indirecta al utilitzar l'estàndard WFS.

WMS (Web Map Service)

Es tracta d'un estàndard que ofereix una interfície simple HTTP per demanar informació geogràfica en forma d'imatges, obtenint aquesta informació d'una base de dades de tipus geogràfic, com pot ser PostGIS.

Una petició WMS defineix la capa i la àrea d'interès que es vol processar. La resposta a la petició són un o més mapes, retornats en forma d'imatge (JPEG, PNG, GIF, etc) que poden ser mostrades en el browser de l'aplicació. La interfície també permet especificar quan les imatges retornades tenen que ser transparents, per tal de que es puguin combinar múltiples capes.

WFS (Web Feature Service)

Es tracta d'una interfície estàndard que permet fer peticions d'informació geogràfica, utilitzant crides totalment independents de la plataforma utilitzada.

Per representar la informació geogràfica, s'utilitza el llenguatge GML, que hem explicat anteriorment.

L'especificació WFS defineix interfícies per la manipulació de dades de caràcter geogràfic. Les operacions que es poden realitzar són:

- Obtenir informació geogràfica, consultada d'una base de dades amb suport geogràfic, com pot ser PostGIS.
- Crear nova informació geogràfica en la base de dades.
- Esborrar informació geogràfica de la base de dades.
- Modificar informació geogràfica de la base de dades.

El servei WFS bàsic només permet obtenir informació de la base de dades. El servei WFS transaccional (WFS-T) també permet crear, esborrar i modificar informació.

Per tant, els serveis de tipus WFS-T, que són els que utilitzarem en la nostra aplicació, ens permeten realitzar peticions d'informació geogràfica, així com realitzar modificacions en aquesta informació (crear, modificar i esborrar). El servei retornarà un resultat en format GML, que tindrà que ser interpretat pel visor de mapes corresponent (Openlayers).

Podem veure un exemple d'un resultat GML obtingut d'una petició a un servei WFS a continuació:

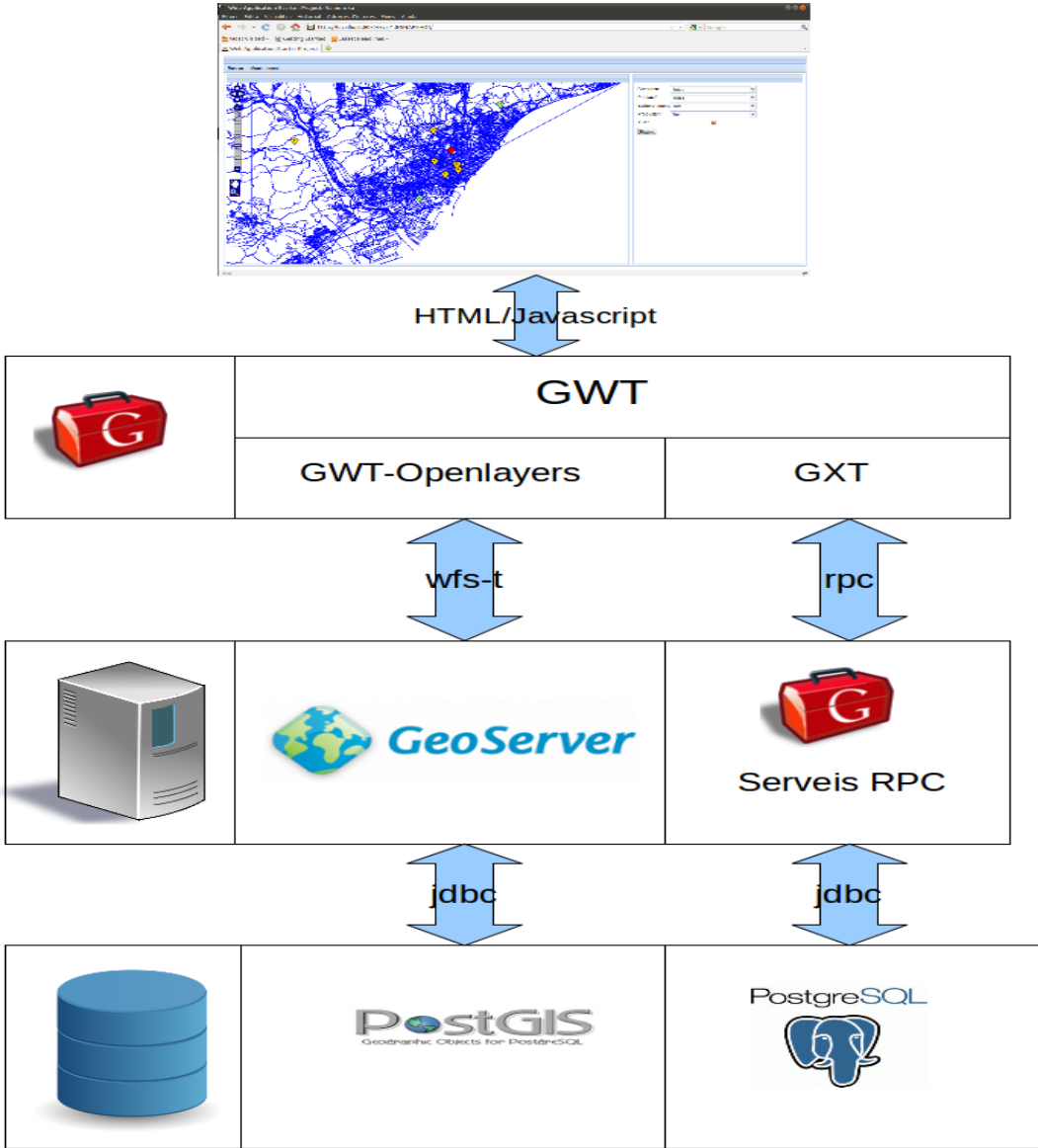

```
<topp:myDataset gml:id="myDataset.1">
  <topp:intProperty>1</topp:intProperty>
  <topp:stringProperty>one</topp:stringProperty>
  <topp:floatProperty>1.1</topp:floatProperty>
  <topp:geometry>
    <gml:Point srsName="urn:x-ogc:def:crs:EPSG:4326">
      <gml:pos>1.0 1.0</gml:pos>
    </gml:Point>
  </topp:geometry>
</topp:myDataset>
```

Podem veure com ens retorna les coordenades d'un punt concret, utilitzant la projecció 4326.

Podem trobar més informació dels serveis WMS, WFS i WFS-T en la web de l'OGC, que podem trobar en la referència [20] de la bibliografia.

5.7.2. Arquitectura de l'aplicació

En aquest apartat podrem veure l'arquitectura utilitzada en aquesta aplicació. La podem veure en la figura número 9.



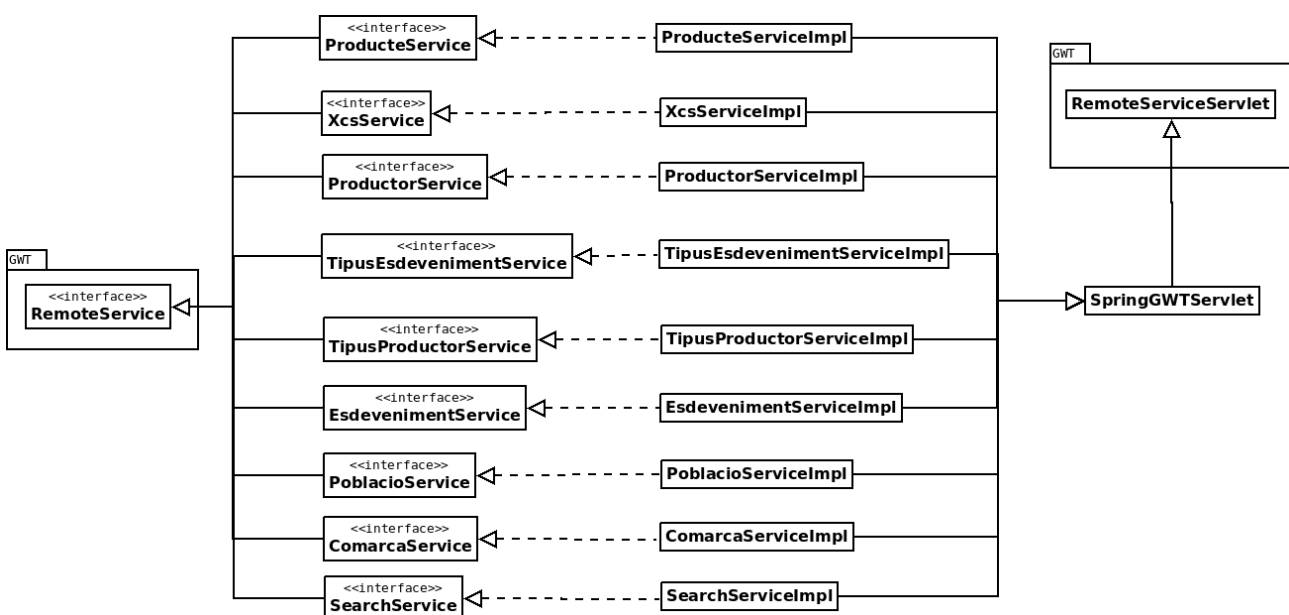
Il·lustració 9: Arquitectura de l'aplicació

A grans trets, podem diferenciar tres parts:

- Base de dades: Servidor de base de dades postgresql, amb la seva extensió PostGIS.
- Servidor: Contindrà els objectes encarregats d'accedir a base de dades i de publicar els serveis web per la comunicació amb la part Client. També tindrem el servidor Geoserver, encarregat de publicar informació geogràfica, en forma de serveis WMS i WFS.
- Client: Contindrà els objectes encarregats de mostrar tota la informació a través del navegador, accedint a la part servidor.

5.7.3. Disseny part Servidor

El primer que definirem, són les classes encarregades de publicar els serveis web, per tal de realitzar la comunicació amb el client. El disseny d'aquesta part, el podem veure en el diagrama UML de la il·lustració número 10.

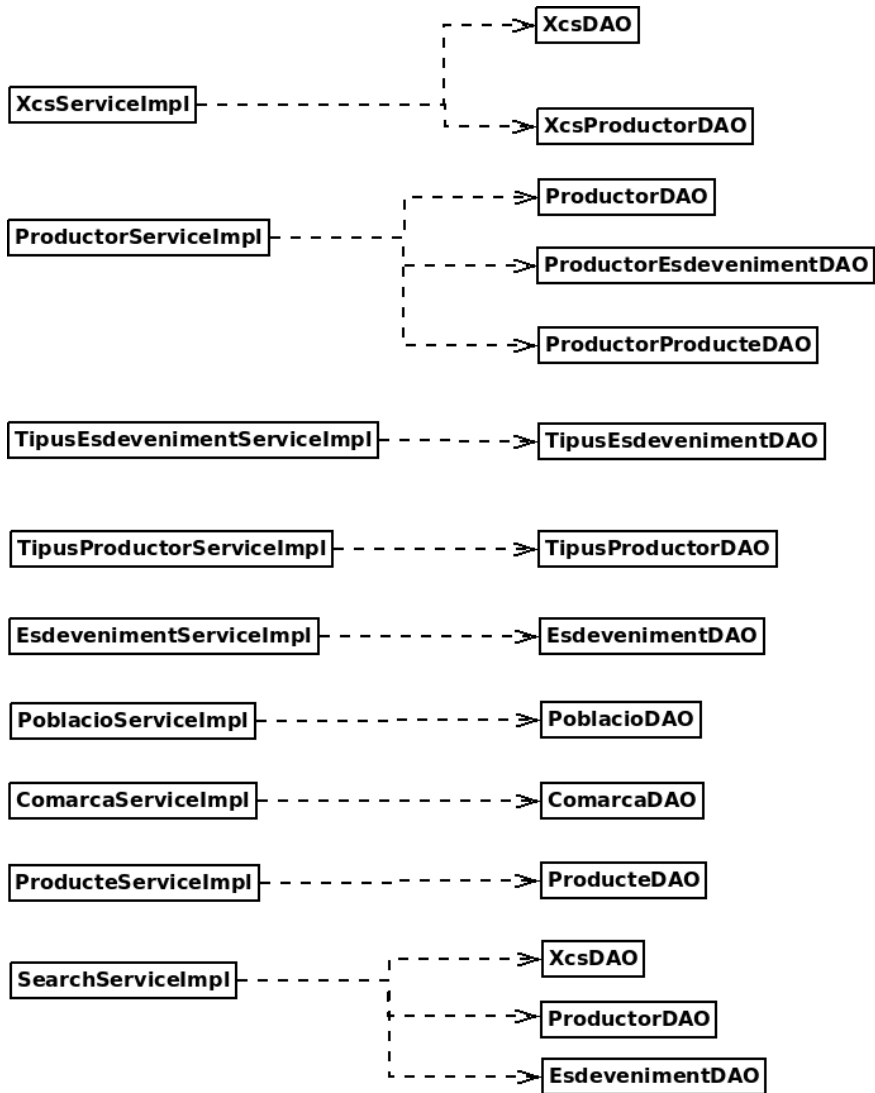


Il·lustració 10: Diagrama UML de serveis web

Podem veure que es crearà un servei web, per cadascuna de les pantalles de l'aplicació, excepte els serveis **ComarcaService** i **PoblacioService**, que no tenen pantalla pròpia, però s'utilitzen en moltes parts de l'aplicació.

Per la generació dels serveis web, s'utilitzarà la tecnologia RPC, proporcionada pel framework GWT. Donarem més detalls d'aquesta comunicació Client-Servidor en la fase d'implementació.

A continuació farem una descripció, de les classes que s'empraran per l'accés a base de dades i la relació que tenen amb els serveis que hem definit anteriorment. Podem veure el disseny d'aquestes classes en el diagrama UML de la imatge número 11.



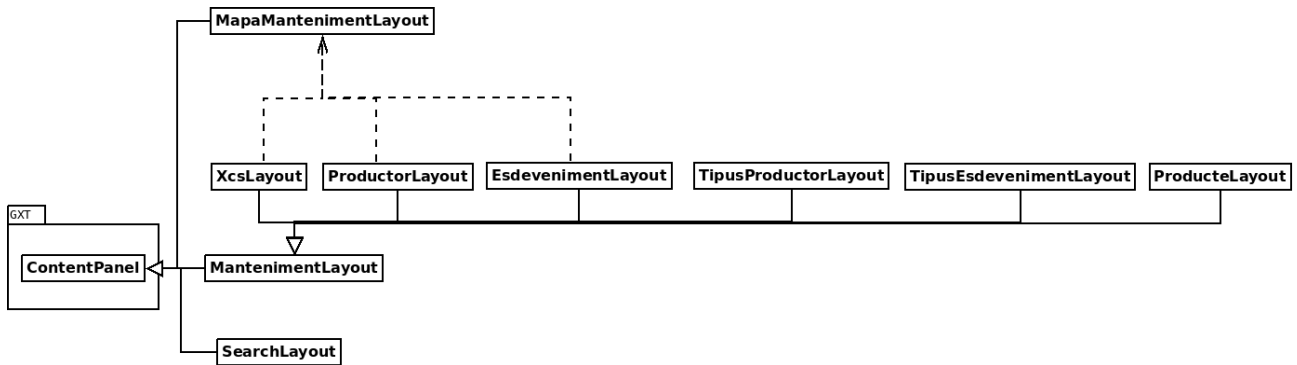
Il·lustració 11: Diagrama UML d'accés a BBDD

En aquest cas, podem veure que tenim un objecte d'accés a base de dades, per cadascuna de les taules de base de dades creades.

5.7.4. Disseny part Client

La part client de l'aplicació de Xarxes de Consum Solidari, estarà desenvolupada utilitzant el framework **GWT** i la seva extensió **GXT**. Aquest framework permet realitzar el desenvolupament

totalment en **Java**, sense necessitat de treballar directament amb HTML/Javascript. Podem veure en la figura número 12, un diagrama UML amb el disseny de classes per la part Client.



Il·lustració 12: Diagrama UML de la part Client

Com podem veure hem optat per un conjunt de classes, que estenen de la classe **ContentPanel** de **GXT**.

Per una banda, tenim la classe **SearchLayout**, que servirà per generar la pàgina de cerca de dades i mostrar la seva localització geogràfica. La pantalla de cerca és diferent a la resta i per tant estén directament de **ContentPanel**.

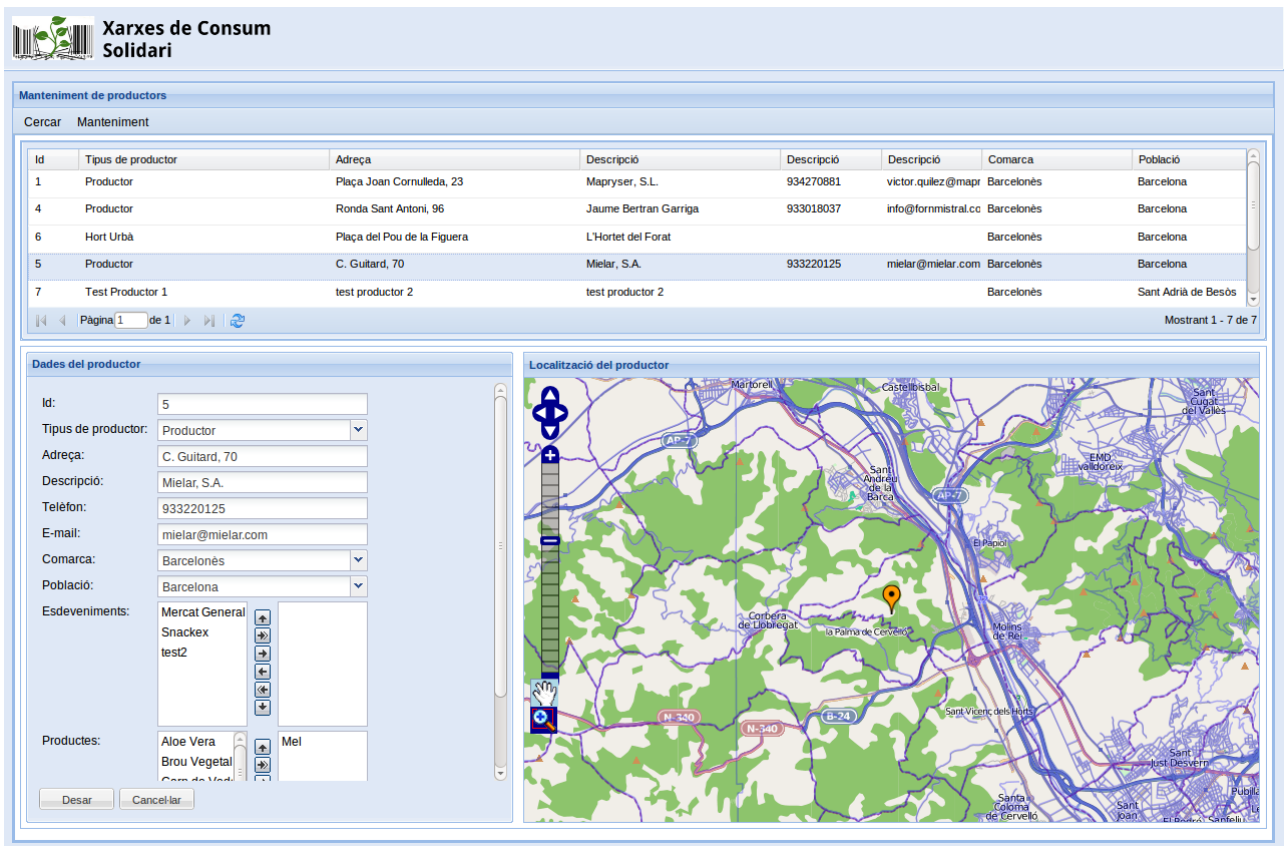
Per altra banda, tenim que la resta de classes estenen de **MantenimentLayout**. Això es així, perquè totes les pantalles de manteniment tenen una estructura similar, amb una llista dels registres existents en base de dades a la part superior i un formulari pel manteniment d'aquestes dades, en la part inferior. Podem veure un exemple d'una d'aquestes pantalles, en la figura número 13.

The screenshot shows a web application interface for 'Xarxes de Consum Solidari'. The main title is 'Manteniment de productes'. Below the title, there are two tabs: 'Cercar' and 'Manteniment', with 'Manteniment' selected. The main content area displays a table with 10 rows of product data. Each row has an 'Id' column and a 'Descripció' column, followed by a trash icon for deletion. The products listed are: 1. Pa, 2. Oli de Girasol, 3. Cereals, 4. Cervesa, 5. Mel, 6. Aloe Vera, 7. Carn de Vedella, 8. Te, 9. Brou Vegetal, and 10. Propolis. At the bottom of the table, there are navigation controls: 'Pàgina 1 de 2' and 'Mostrant 1 - 10 de 11'. Below the table is a section titled 'Dades del producte' which contains two input fields: 'Id:' and 'Descripció:'. At the bottom of this section are two buttons: 'Desar' and 'Cancel·lar'.

| Id | Descripció |
|----|-----------------|
| 1 | Pa |
| 2 | Oli de Girasol |
| 3 | Cereals |
| 4 | Cervesa |
| 5 | Mel |
| 6 | Aloe Vera |
| 7 | Carn de Vedella |
| 8 | Te |
| 9 | Brou Vegetal |
| 10 | Propolis |

Il·lustració 13: Exemple de pantalla de manteniment simple

Per altra banda, tenim una serie de classes, que fan us de la classe **MapaMantenimentLayout**. Aquesta classe conté tota la lògica necessària per mostrar un mapa amb informació actualitzable utilitzant WFS-T. Per tant, aquestes pantalles tindran un aspecte com el de qualsevol pantalla de manteniment, però afegint també un mapa creat amb Openlayers, en la part inferior. Podem veure un exemple d'una d'aquestes pantalles en la figura número 14.



Xarxes de Consum Solidari

Manteniment de productors

Cercar Manteniment

| Id | Tipus de productor | Adreça | Descripció | Descripció | Descripció | Comarca | Població |
|----|--------------------|-----------------------------|-----------------------|------------|----------------------|------------|---------------------|
| 1 | Productor | Plaça Joan Cornulleda, 23 | Mapryser, S.L. | 934270881 | victor.quelez@mapr | Barcelonès | Barcelona |
| 4 | Productor | Ronda Sant Antoni, 96 | Jaume Bertran Garriga | 933018037 | info@formmistrall.co | Barcelonès | Barcelona |
| 6 | Hort Urbà | Plaça del Pou de la Figuera | L'Hortel del Forat | | | Barcelonès | Barcelona |
| 5 | Productor | C. Guitard, 70 | Mielar, S.A. | 933220125 | mielar@mielar.com | Barcelonès | Barcelona |
| 7 | Test Productor 1 | | test productor 2 | | | Barcelonès | Sant Adrià de Besòs |

Pàgina 1 de 1

Mostrant 1 - 7 de 7

Dades del productador

Id: 5

Tipus de productor: Productor

Adreça: C. Guitard, 70

Descripció: Mielar, S.A.

Telèfon: 933220125

E-mail: mielar@mielar.com

Comarca: Barcelonès

Població: Barcelona

Esdeveniments: Mercat General, Snackex, test2

Productes: Aloe Vera, Brou Vegetal, Mel

Desar Cancel·lar

Localització del productador

Mapa de localització del productador a Barcelona.

Il·lustració 14: Exemple de pantalla de manteniment amb mapa

El fet de implementar tota la part client utilitzant **Java**, ens permet definir una jerarquia de classes per pantalles similars, que farà que repetim menys codi i augmentem el re-aprofitament.

5.8. Capítol 8. Implementació

En aquest capítol, intentarem detallar les decisions de disseny més destacables, que hem tingut que prendre en la implementació del projecte.

5.8.1. Implementació Accés BBDD

Per la implementació de les classes d'accés a base de dades (DAO), vam estudiar la utilització de diferents tecnologies:

- Hibernate.
- JDBC sense la ajuda de cap framework.
- Spring i el seu suport per DAOs.

Hibernate és un framework que facilita el guardar i recuperar objectes **Java**, definint únicament una relació entre aquests objectes i la base de dades relacional. Aquesta relació es pot definir en un fitxer xml o utilitzant anotacions.

Spring és un framework **Java**, per la construcció d'aplicacions empresarials. Aquest framework ens proporciona molts serveis, sent un dels més coneguts la injecció de dependències.

Entre els serveis que ens proporciona **Spring**, tenim el suport per la creació de objectes d'accés a base de dades (DAO). A continuació podem veure una taula, extreta de la documentació oficial de Spring, que ens indica exactament que ens proporciona i que no, aquest framework.

| Acció | Spring | Programador |
|---|--------|-------------|
| Definir els paràmetres de connexió. | | X |
| Obrir la connexió. | X | |
| Especificar la sentència SQL. | | X |
| Declarar els paràmetres i proporcionar els valors per aquests paràmetres. | | X |
| Preparar i executar la sentència. | X | |
| Preparar el bucle per recórrer els resultats (si n'hi han). | X | |
| Fer la feina per a cada iteració. | | X |
| Processar qualsevol excepció. | X | |
| Gestionar les transaccions. | X | |
| Tancar la connexió, statement i resultset. | X | |

Com a última opció, teníem el accés directe a la base de dades, utilitzant únicament **JDBC**. En aquest cas, tenim que gestionar nosaltres tot el cicle necessari per accedir a una base de dades.

Finalment l'opció triada ha estat el framework **Spring**, pels següents motius:

- **Hibernate** pot ser complicat de configurar i de posar en marxa, per tant no és molt recomanable per un projecte amb poc temps de desenvolupament, com és aquest.
- No sabem com es comporta **Hibernate** amb dades geogràfiques provinents de **PostGIS**.
- L'accés directe per **JDBC**, tot i que és simple, pot arribar a ser una mica més tediós de desenvolupar, ja que ens tenim que encarregar de coses com obrir i tancar connexions, capturar i processar les excepcions, etc.

5.8.1.1. Implementació DAO amb Spring

En aquest apartat explicarem com hem implementat un objecte del tipus DAO, per tal de donar accés a la base de dades.

La implementació d'un DAO utilitzant Spring té que seguir una serie de convencions:

1. Hem de crear una classe qualssevol, sense necessitat de que aquesta classe hereti de cap altre.
2. Ha de contindre l'anotació **@Repository** a nivell de classe, per tal de que Spring la gestioni.
3. Dins de la classe hem de tenir un atribut de tipus **JdbcTemplate**. Aquest objecte conté totes les funcions necessàries per executar sentències sql.
4. Hem de crear un mètode, que rebi com a paràmetre un **DataSource**. En aquest mètode, el que farem és crear una instància del objecte **JdbcTemplate**, passant-li com a paràmetre el objecte **DataSource**, que hem rebut com a paràmetre del mètode.
5. El mètode que hem creat anteriorment, ha de tenir l'anotació **@Autowired** a nivell de mètode. Aquesta anotació indica a Spring, que és ell l'encarregat de cridar a aquest mètode i de proporcionar-li el **DataSource** en el moment de la crida.
6. Per últim, hem de crear tants mètodes com necessitem. Cadascun d'aquests mètodes, utilitzarà l'objecte **JdbcTemplate** per executar sentències sql.
7. Per gestionar els resultats d'una consulta, podem crear un objecte del tipus **RowMapper**, que relacionarà les dades d'un registre de base de dades, amb els camps d'un objecte Java. Serà **Spring** l'encarregat d'iterar sobre els resultats obtinguts i d'utilitzar el objecte **RowMapper** creat, per tal d'anar passant els resultats a objectes Java.

A continuació, posem un petit exemple de la implementació d'un DAO, utilitzant aquesta tecnologia:

```

@Repository
public class ProducteDAO {

    private JdbcTemplate template;

    @Autowired
    public void setDataSource(DataSource dataSource) {
        template = new JdbcTemplate(dataSource);
    }

    public List<Producte> getAll() throws DataAccessException {
        String sql = "select idproducte, descripcio productedesc from producte order by productedesc asc";
        List<Producte> result = this.template.query(sql, new RowMapper<Producte>() {
            @Override
            public Producte mapRow(ResultSet rs, int rowNum) throws SQLException {
                Producte producte = new Producte();
                producte.setId(rs.getInt("idproducte"));
                producte.setDescripcio(rs.getString("productedesc"));
                return producte;
            }
        });
        return result;
    }
}

```

```

}
}

```

5.8.2. Implementació serveis web

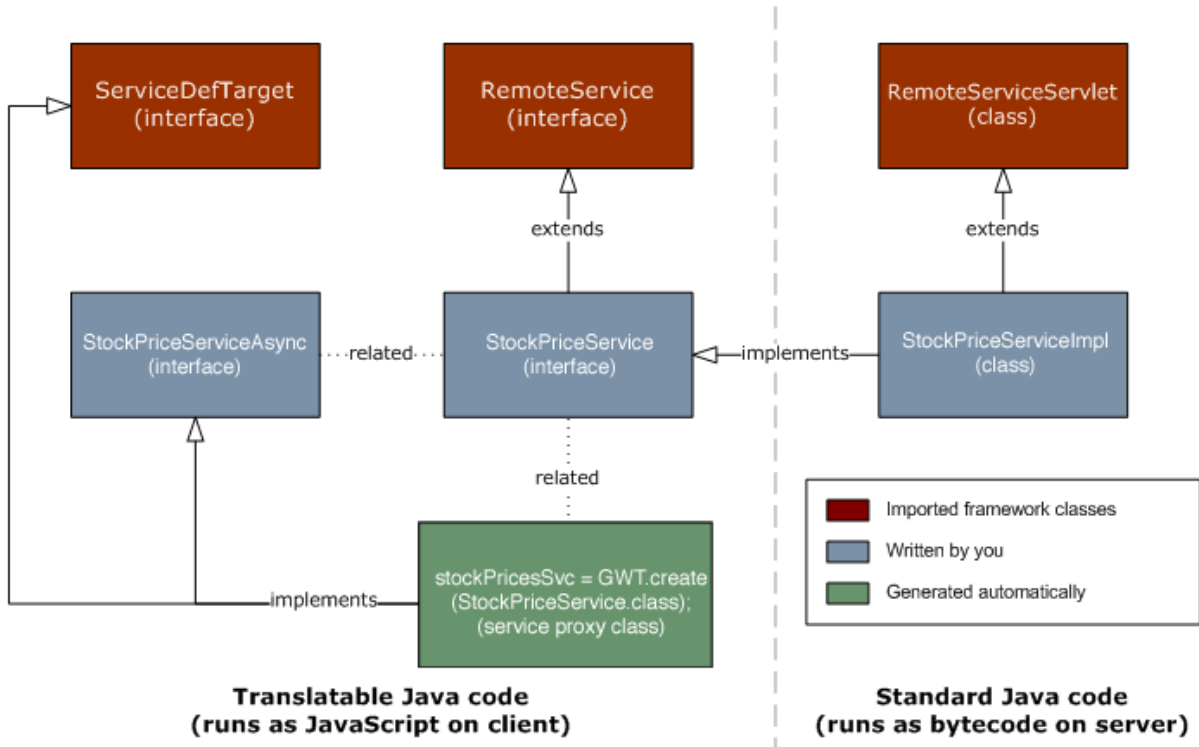
Per la implementació de serveis web, per utilitzar-los en la part client implementada amb GWT, tenim dues opcions a avaluar:

- GWT RPC.
- JSON.

GWT RPC és un sistema de comunicació, que permet intercanviar objectes **Java** entre client i servidor, utilitzant el protocol HTTP.

La implementació de **GWT RPC** està inspirada en la ben coneguda arquitectura de servlets, mentre que en el client es fa la crida a través d'un proxy auto-generat. GWT s'encarrega de serialitzar els objectes, per tal de poder enviar-los utilitzant el protocol http.

En la imatge número 15, podem veure un exemple de les classes necessàries, per implementar un servei web utilitzant **GWT RPC**.



Il·lustració 15: Diagrama UML per la implementació de un servei GWT RPC

JSON és un format de representació de dades, universal e independent del llenguatge.

Per la implementació d'aquest servei, també s'utilitzarà l'arquitectura de servlets, però en aquest cas, serem nosaltres els encarregats de transformar les dades en un objecte **JSON**.

Finalment ens hem decidit per **GWT RPC**, per els següents motius:

- Facilitat d'implementació, ja que sempre treballarem amb objectes Java i no tindrem que passar les dades a cap altre format (XML o JSON).
- Els widgets de **GWT** i **GXT** estan preparats per treballar directament amb **GWT RPC**, mentre que no es tan fàcil treballar amb **JSON**. Alguns widgets, com per exemple el encarregat de mostrar taules amb paginació, utilitzen directament un servei **GWT RPC**.

5.8.2.1. Implementació servei GWT RPC

En aquest capítol explicarem els passos bàsics, per tal de poder crear un servei web amb la tecnologia GWT RPC.

El procediment a seguir és aquest:

1. Creació d'una interfície en la part client, que hereta de la interfície **RemoteService**. Aquesta interfície ha de tenir l'anotació **@RemoteServiceRelativePath("path")**, on "path" serà l'adreça relativa del servei.
2. Aquesta interfície ha de definir els mètodes que formen part del servei web. Aquests mètodes accepten paràmetres i retornen objectes Java, sempre que siguin serialitzables.
3. Els mètodes creat llença'n una excepció del tipus **IllegalArgumentException**, que portarà encapsulada la excepció original en cas d'error.
4. Creació de la implementació de la interfície en la part servidor. Aquesta classe heretara de la classe **SpringGwtServlet**, que ens permetrà integrar **Spring** dins dels serveis **GWT RPC**. Al mateix temps la classe ha d'implementar la interfície anteriorment creada.
5. Dins d'aquesta classe podem utilitzar un o més objectes de tipus DAO. Per fer això, només hem de definir el DAO com un atribut de la classe i posar en aquest atribut l'anotació **@Autowired**, per tal de que **Spring** injecti una instància del DAO en aquest servei.
6. Crear una anotació en el fitxer **web.xml**, per tal de definir el servei com un **Servlet**.

A continuació adjuntem un petit exemple d'un servei web, implementat utilitzant la tecnologia **GWT RPC**. El primer que veurem és la interfície de la part client:

```
@RemoteServiceRelativePath("producte")
public interface ProducteService extends RemoteService {
```

```
public List<Producte> getAll() throws IllegalArgumentException;
}
```

El següent que veurem és la implementació de la interfície:

```
public class ProducteServiceImpl extends SpringGwtServlet implements ProducteService {
    @Autowired
    private ProducteDAO producteDAO;

    @Override
    public List<Producte> getAll() throws IllegalArgumentException {
        return producteDAO.getAll();
    }

    public ProducteDAO getProducteDAO() {
        return producteDAO;
    }

    public void setProducteDAO(ProducteDAO producteDAO) {
        this.producteDAO = producteDAO;
    }
}
```

Com a últim pas veurem la part d'aquest servei en el fitxer web.xml:

```
<servlet-mapping>
    <servlet-name>producteServlet</servlet-name>
    <url-pattern>/xcs/producte</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>producteServlet</servlet-name>
    <servlet-class>uoc.edu.server.service.ProducteServiceImpl</servlet-class>
</servlet>
```

5.8.3. Implementació part Client

En la part client hem utilitzat diverses tecnologies, que detallem a continuació:

1. **GWT**, que ens permetrà desenvolupar utilitzant únicament Java.
2. **GXT**, que ens permetrà utilitzar els components de **ExtJS**, però desenvolupant amb **GWT**.
3. **GWT-Openlayers**, que ens permetrà mostrar un mapa dins de components **GWT**.

5.8.3.1. Implementació GWT

A continuació farem una breu descripció dels elements necessaris per desenvolupar amb GWT:

1. Un fitxer html, que serà el punt de partida de l'aplicació. Dins d'aquest fitxer serà on **GWT** generarà tot el codi html i javascript. El nom del fitxer en aquest cas és **xcs.html**.
2. Una classe que hereti de EntryPoint, que serà el punt d'entrada de l'aplicació i des d'on carregarem tots els panells que formen part de l'aplicació. En aquest cas la classe s'anomena **xcs**.
3. Un fitxer de configuració en format xml, on indiquem a **GWT** la classe d'entrada de l'aplicació i tots les llibreries que utilitzarem com **GXT** i **GWT-Openlayers**.

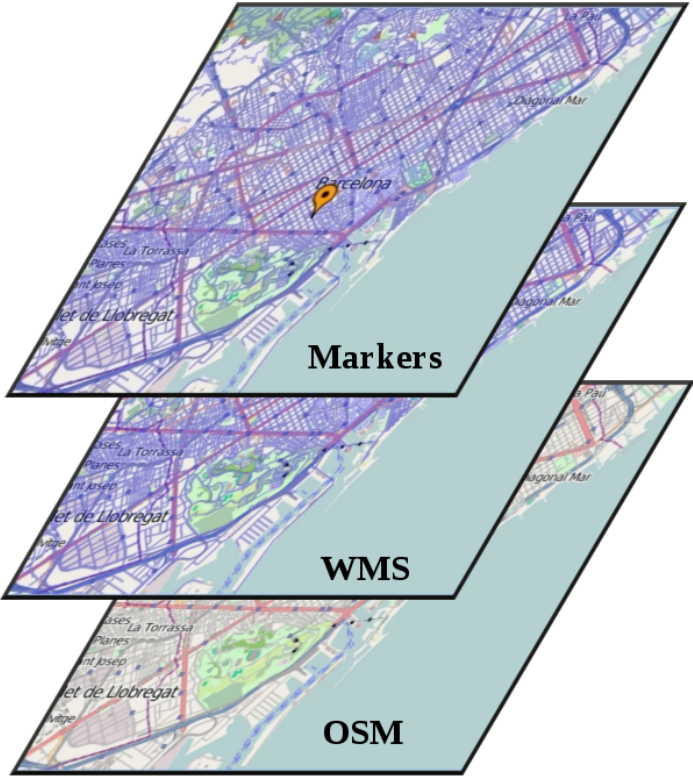
5.8.4. Implementació dels mapes

Per la implementació dels mapes, utilitzarem la llibreria opensource Openlayers. Com que aquesta llibreria no es pot utilitzar directament en GWT, utilitzarem el projecte GWT-Openlayers, que no és un altre cosa que un wrapper d'Openlayers.

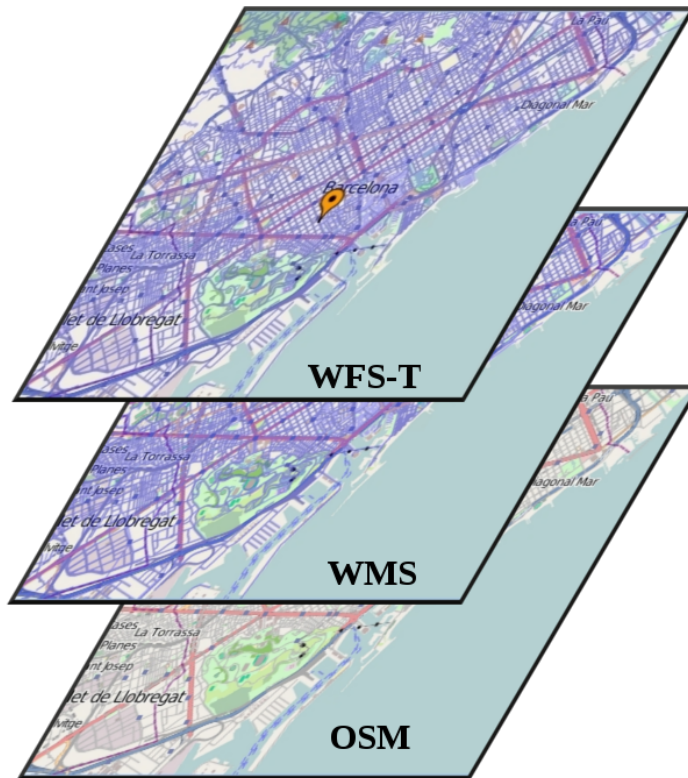
Openlayers ens permet mostrar un mapa i afegir-hi capes a diferents serveis. En el nostre cas aquestes son les capes que utilitzarem:

- OSM: Es una capa base que ens mostrarà la informació d'OpenStreetMap.
- WMS (Web Map Service): Openlayers ens proporciona una capa, que ens permet accedir a serveis web WMS i per tant utilitzarem aquesta capa, per tal d'accedir a la informació importada des d'OpenStreetMap a la nostra base de dades. Aquesta capa accedirà al nostre servidor Geoserver i obtindrà la cartografia de referència importada d'OpenStreetMap.
- WFS i WFS-T (Web Feature Service): Openlayers també ens proporciona una capa, que ens permet accedir a aquest servei web i mostrar-nos la informació en el mapa en forma de vectors modificables. Aquesta capa es la que utilitzarem per mostrar i editar la informació de les xarxes de consum, productors i esdeveniments. Aquesta capa accedirà al servidor Geoserver, per tal d'obtenir la informació geogràfica de Xarxes de Consum Solidari, productors i esdeveniments.
- Markers: És una capa de Openlayers, que no accedeix a cap servei en concret i que té l'única finalitat de permetre afegir marcadors en el mapa. Aquí som nosaltres els que tenim que accedir directament a la base de dades, per tal d'obtenir la localització de Xarxes de Consum Solidari, productors i esdeveniments, per tal de poder-los situar en el mapa.

Podem veure unes imatges representatives d'aquestes capes, en la imatge número 16, per el cas de les pantalles de cerca i en la imatge numero 17, per el cas de les pantalles de manteniment.



Il·lustració 16: Capes pantalla de cerca



Il·lustració 17: Capes pantalla manteniment

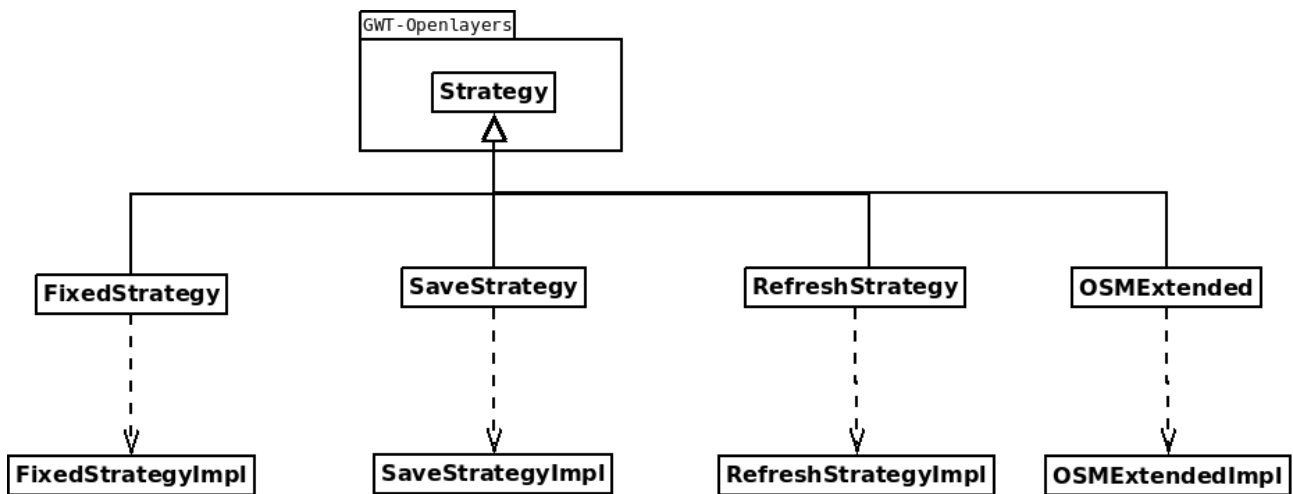
5.8.4.1. Implementació GWT-Openlayers

Per tal de poder utilitzar Openlayers dins del nostre projecte, hem utilitzat la llibreria GWT-Openlayers.

En la primera avaluació que vam fer a aquesta llibreria, semblava que teníem tot el necessari per el nostre projecte, però a la pràctica ens hem trobat que encara hi ha moltes coses sense implementar, concretament:

- La part de WFS està implementada i funciona correctament, però falta per implementar la part de WFS-T.
- La capa OSM està implementada, però no funciona.

Es per això que ens hem vist obligats a desenvolupar una extensió de GWT-Openlayers, per tal d'incloure aquestes funcionalitats que mancaven i que necessitàvem pel correcte funcionament de la nostra aplicació. Podem veure un diagrama UML amb aquesta extensió que hem realitzat, en la figura número 18.



Il·lustració 18: Extensió de GWT-Openlayers

Les classes **FixedStrategy**, **SaveStrategy** i **RefreshStrategy**, corresponen a la part de WFS-T:

- **FixedStrategy**, demana les figures al servidor utilitzant WFS només una vegada al carregar el mapa. Com que en els nostres mapes de manteniment, només volíem mostrar la figura corresponent a un registre de la base de dades, aquesta era la única Strategy que ens funcionava correctament. La Strategy BBOX ens mostrava totes les figures de la base de dades, ignorant per tant el filtre.
- **SaveStrategy**: Quan hem acabat d'afegir o modificar figures, hem d'enviar aquesta informació al servidor Geoserver, utilitzant per això el servei WFS-T. Aquesta Strategy en permet enviar les dades al servidor.
- **RefreshStrategy**: Com que estem utilitzant **FixedStrategy**, cada cop que canviem el filtre del mapa, per tal de mostrar una figura diferent, tenim que forçar a la capa WFS, per tal de que torni a demanar les dades al servidor. Aquesta Strategy en servirà per aquesta finalitat.

La classe **OSMExtended** serveix per afegir una capa OSM base al nostre mapa. Utilitzarem aquesta classe enlloc de la classe **OSM** que ja ens proporciona GWT-Openlayers, ja que no hem aconseguit que funcionés amb la classe original, ja que pel que sembla està mal implementada.

GWT-Openlayers ens proporciona una serie de classes, per la creació del mapa i de les seves respectives capes. Anem a llistar aquestes classes:

- **MapWidget**: Crea un mapa d'Openlayers dins d'un contenidor GWT/GXT.
- **Bounds**: Serveix per definir els límits d'un mapa.

- MapOptions: Classe que utilitzarem per passar configuració a un MapWidget, com per exemple el tipus de projecció.
- WMS: Serveix per afegir a un mapa, una capa per mostrar informació obtinguda d'un servei web WMS.
- WMSParams i WMSLayerParams: Classes per configurar un capa WMS d'Openlayers. Podem passar paràmetres com el format de sortida o la projecció a emprar.
- Vector: Utilitzarem aquesta classe per afegir una capa WFS-T al nostre mapa.
- WFSProtocol i WFSProtocolOptions: Definició del protocol WFS i de la seva configuració, per tal de poder construir la capa WFS.
- VectorOptions: Opcions de configuració de la capa WFS.
- Markers: Permet crear una capa en el mapa, per poder posteriorment afegir marcadors.

Anem a posar alguns exemples en forma de codi. Comencem per la creació d'un mapa, amb la seva capa OSM i WMS:

```

Bounds bounds = new Bounds(12746, 4932506, 374628, 5295759);
MapOptions defaultMapOptions = new MapOptions();
defaultMapOptions.removeDefaultControls();
defaultMapOptions.setProjection("EPSG:900913");
defaultMapOptions.setMaxExtent(bounds);
defaultMapOptions.setMaxResolutionToAuto();
MapWidget mapWidget = new MapWidget("100%", "100%", defaultMapOptions);
OSMExtended osm = new OSMExtended("osm");
WMSParams wmsParams = new WMSParams();
wmsParams.setFormat("image/png");
wmsParams.setLayers("xcs:planet_osm_line");
wmsParams.setIsTransparent(true);
WMSOptions wmsLayerParams = new WMSOptions();
wmsLayerParams.setUntiled();
wmsLayerParams.setLayerOpacity(0.4);
wmsLayerParams.setTransitionEffect(TransitionEffect.RESIZE);
WMS wmsLayer = new WMS("xcs:planet_osm_line",
    "http://localhost:8080/geoserver/wms", wmsParams, wmsLayerParams);
mapWidget.getMap().addLayers(new Layer[] {osm, wmsLayer});

```

Si volem ara afegir una capa WFS-T, el codi seria el següent:

```

WFSProtocolOptions wfsProtocolOptions = new WFSProtocolOptions();
wfsProtocolOptions.setUrl("http://localhost:8080/geoserver/wfs");
wfsProtocolOptions.setVersion("1.1.0");
wfsProtocolOptions.setFeatureType(wfsLayerName);
wfsProtocolOptions.setSrsName("EPSG:900913");
wfsProtocolOptions.setFeatureNameSpace("http://uoc.edu/xcs");

```

```
wfsProtocolOptions.setGeometryName("localitzacio");
WFSProtocol wfsProtocol = new WFSProtocol(wfsProtocolOptions);

VectorOptions vectorOptions = new VectorOptions();
vectorOptions.setProtocol(wfsProtocol);
SaveStrategy saveStrategy = new SaveStrategy();
RefreshStrategy refreshStrategy = new RefreshStrategy();
vectorOptions.setStrategies(new Strategy[] { new FixedStrategy(), saveStrategy,
    refreshStrategy });
Vector wfsLayer = new Vector(wfsLayerName, vectorOptions);
mapWidget.getMap().addLayers(new Layer[] {wfsLayer });
```

Per ultim, per afegir la capa de marcadors:

```
Markers markersLayer = new Markers("markers");
mapWidget.getMap().addLayers(new Layer[] {markersLayer });
```

5.8.5. Implementació de la pantalla de consultes

La pantalla de consultes ens mostrarà la localització de, xarxes de consum solidari, productors i esdeveniments. La pantalla de consultes es dividirà en dues parts:

- Àrea de visualització del mapa (Esquerra).
- Formulari de selecció de filtres de consulta (Dreta).

En la pantalla de selecció de filtres, tindrem els següents filtres disponibles:

- Comarca: Ens permetrà filtrar per Comarca. Aplica a Xarxes de consum solidari, productors i esdeveniments.
- Població: Ens permetrà filtrar per Població. Aplica a Xarxes de consum solidari, productors i esdeveniments.
- Producte: Ens permetrà filtrar per Producte. En aquest cas només aplica a productors.
- Esdeveniment: Ens permetrà filtrar per un esdeveniment concret. Només aplica a productors.
- Tipus Esdeveniment: Permet filtrar per tipus d'esdeveniment. Només aplica a esdeveniments.
- Tipus Productor: Permet filtrar per tipus de productor. Només aplica a productors.
- XCS: Permet filtrar per una Xarxa de consum solidari concreta. Només aplica a productors.

Com podem veure, amb aquesta combinació de criteris de selecció, podrem realitzar les consultes indicades com a requeriments mínims de l'aplicació.

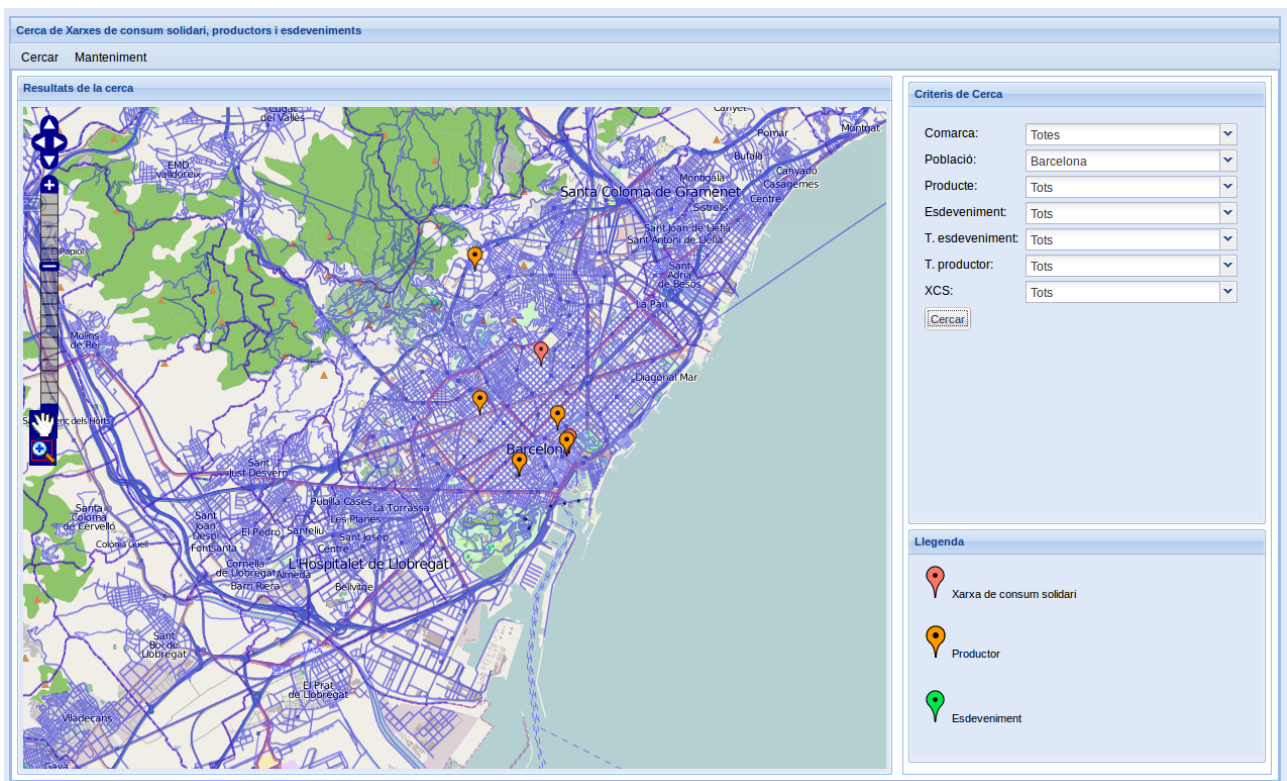
Anem de totes formes a repassar una per una les consultes requerides, indicant la sentència sql que s'executarà i mostrant una captura del resultat en pantalla.

1. Filtrar xarxes de consum solidari, productors i consumidors per àmbit geogràfic.

Per exemple si filtrem per població, s'executaran tres sentències sql, que són las següents:

```
select localitzacio, descripcio from esdeveniment where idpoblacio = ?
select p.localitzacio, p.descripcio from productor p where p.idpoblacio = ?
select localitzacio, descripcio from xcs where idpoblacio = ?
```

El resultat obtingut per pantalla serà el de la figura 19.



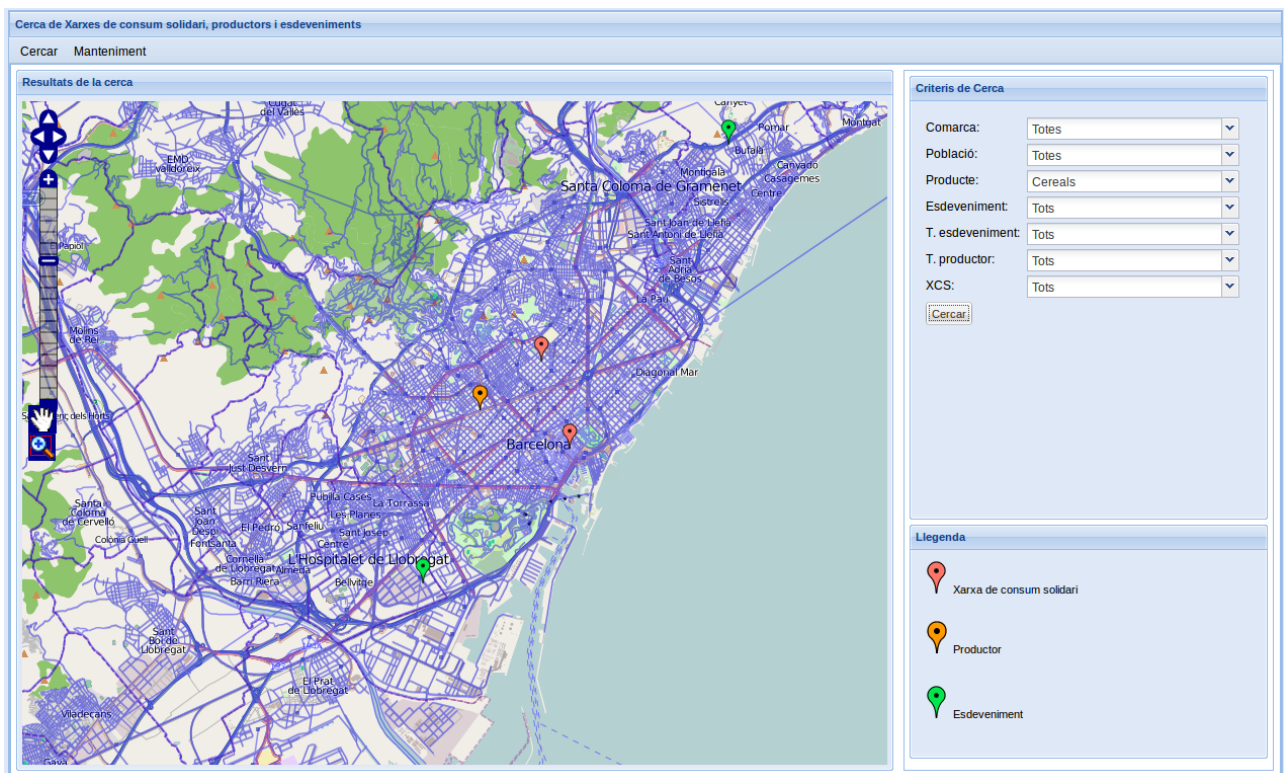
Il·lustració 19: Resultat cerca filtrant per població

2. Filtrar productors, segons el tipus de producte.

En aquest cas, al filtrar per un producte concret, necessitarem una única consulta sql, ja que només aplica a productors:

```
select p.localitzacio, p.descripcion from productor p
left join productor_producto pp on p.idproductor = pp.idproductor where
pp.idproducte = ?
```

I el resultat obtingut per pantalla en aquest cas, serà com el de la figura 20.



Il·lustració 20: Resultat al filtrar per producte

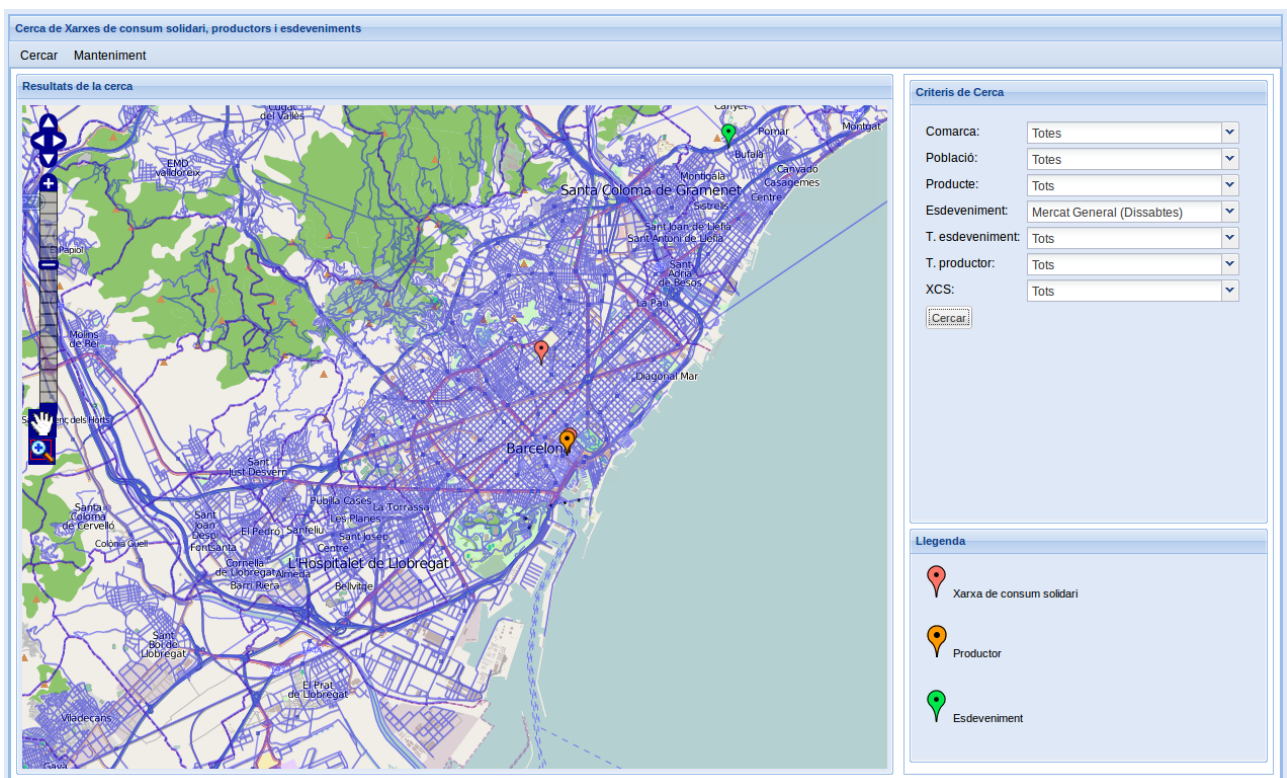
Com podem veure, les xarxes de consum solidari i els esdeveniments segueixen sortint, ja que el filtre no els afecta.

3. Filtrar productors, segons un esdeveniment concret.

Per a aquesta consulta, també necessitarem una única consulta sql, ja que novament només afecta als productors:

```
select p.localitzacio, p.descripcio from productor p
left join productor_esdeveniment pe on p.idproductor = pe.idproductor
where pe.idesdeveniment = ?
```

El resultat que ens mostrarà l'aplicació en aquest cas és el de la figura 21.



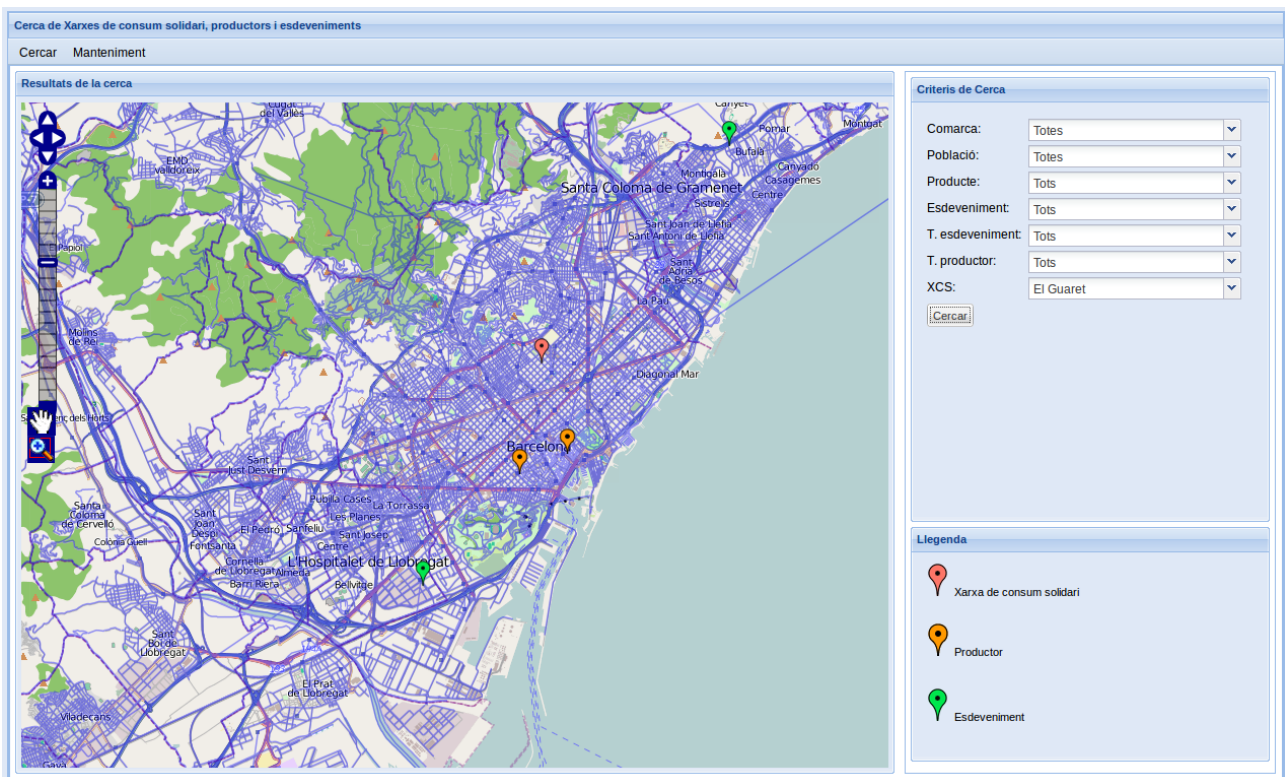
Il·lustració 21: Resultat al filtrar per esdeveniment

4. Filtrar productors que pertanyen a una xarxa de consum solidari.

Novament tindrem una única consulta sql:

```
select p.localitzacio, p.descripcio from productor p
left join xcs_productor xp on p.idproductor = xp.idproductor where xp.idxcs = ?
```

I el resultat per pantalla, és el que podem veure en la figura número 22.



Il·lustració 22: Resultat al filtrar per una xarxa de consum solidari

5.8.6. Pantalles de manteniment amb mapa

Les pantalles de manteniment amb mapa, estan formades per tres parts ben diferenciades:

- Llista del elements existents en base de dades (Superior).
- Formulari de manteniment de dades alfanumèriques (Inferior esquerra).
- Mapa de manteniment de dades geogràfiques (Inferior dreta).

La creació del mapa, en aquest cas, es realitzarà dins de la classe **MapaMantenimentLayout**, per tal de poder utilitzar el mateix codi en les diferents pantalles d'aquestes característiques.

Tan per guardar les dades alfanumèriques, com les dades geogràfiques, utilitzarem el protocol WFS-T que hem mencionat anteriorment. Bàsicament, la part on assignarem les dades alfanumèriques a una figura del mapa i enviarem les dades al servidor, es pot veure en el següent fragment de codi:

```
VectorFeature feature = mapaManteniment.getFeature();
```

```

if (feature != null) {
    if (id.getValue() != null) {
        feature.getJSObject().setProperty("state", "Update");
    }
    Attributes attributes = new Attributes();
    attributes.setAttribute("idtipusproductor",
        tipusProductor.getValue().getId());
    attributes.setAttribute("adreca", adreca.getValue());
    attributes.setAttribute("descripcio", descripcio.getValue());
    attributes.setAttribute("telefon", telefon.getValue());
    attributes.setAttribute("email", email.getValue());
    attributes.setAttribute("idcomarca", comarca.getValue().getId());
    attributes.setAttribute("idpoblacio", poblacio.getValue().getId());
    feature.setAttributes(attributes);
    mapaManteniment.getSaveStrategy().save();
    getGrid().fireEvent(Events.Attach);
    mapaManteniment.setFeatureId(0);
} else {
    MessageBox.alert(getMessages().dialleg_error(),
        getMessages().error_localitzacio_obligatoria(), null);
}

```

Bàsicament el que estem fent, és recuperar un objecte VectorFeature del mapa, que correspon a la figura que estem mostrant en el mapa actualment. A aquesta figura li assignem totes les dades del formulari i posteriorment cridem al mètode save del SaveStrategy, perquè s'envii al servidor i es guardin les dades.

Com podem veure, estem forçant el estat de la figura a update. Això ho estem fent, perquè en cas de que només modifiquem les dades alfanumèriques, no s'enviarien les dades al servidor i per tant no es guardarien els canvis, d'aquesta forma quan es tracta d'una actualització, ens assegurem de que les dades es guardin.

L'aspecte d'una d'aquestes pantalles, un cop guardades les dades, seria el de la imatge número 23.

Manteniment de productors

Cercar Manteniment

| Id | Tipus de productor | Adreça | Descripció | Descripció | Descripció | Comarca | Població |
|----|--------------------|-----------------------------|-----------------------|------------|---------------------|------------|---------------------|
| 1 | Productor | Plaça Joan Cornulleda, 23 | Mapryser, S.L. | 934270881 | victor.quelez@mapr | Barcelonès | Barcelona |
| 4 | Productor | Ronda Sant Antoni, 96 | Jaume Bertran Garriga | 933018037 | info@formmistral.co | Barcelonès | Barcelona |
| 6 | Hort Urbà | Plaça del Pou de la Figuera | L'Hortel del Forat | | | Barcelonès | Barcelona |
| 5 | Productor | C. Guitard, 70 | Mielar, S.A. | 933220125 | mielar@mielar.com | Barcelonès | Barcelona |
| 7 | Test Productor 1 | test productor 2 | test productor 2 | | | Barcelonès | Sant Adrià de Besòs |

Pàgina 1 de 1 Mostrant 1 - 7 de 7

Dades del productor

Id:

Tipus de productor:

Adreça:

Descripció:

Telèfon:

E-mail:

Comarca:

Població:

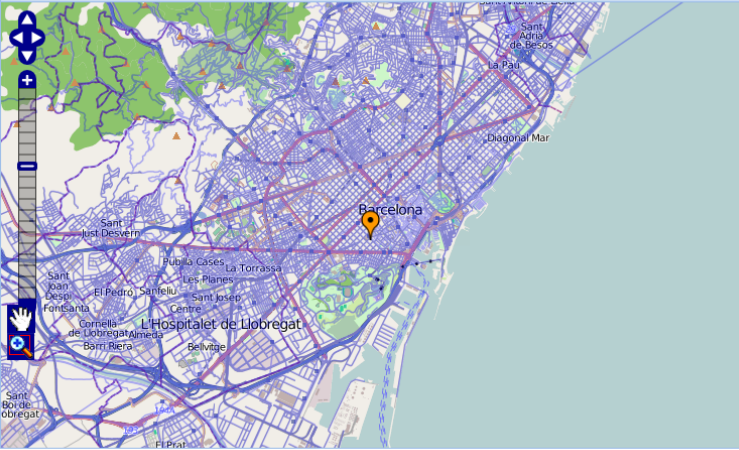
Esdeveniments:

- Mercat General
- Snackex
- test2

Productes:

- Aloe Vera
- Brou Vegetal
- Pa

Localització del productor



Il·lustració 23: Manteniment de productors, mostrant les dades d'un productor

5.9. Capítol 9. Línies de treball futures

- El framework GWT ens ha proporcionat una manera senzilla de desenvolupar una aplicació web, però en canvi hem patit bastant les mancances del projecte GWT-Openlayers, que és força incomplet i no incorpora totes les funcionalitats d'Openlayers. De cara a un futur, es tindria que estudiar la possibilitat de fer una extensió completament funcional d'aquest projecte, que no oblidem és completament Opensource.
- Ampliar el nombre de filtres en la pantalla de consultes, de forma que l'usuari pugui reduir el nombre de resultats d'una cerca d'informació i obtenir uns resultats més acurats.
- En el moment de crear o actualitzar un registre utilitzant WFS-T, no hem trobat la manera de gestionar la resposta del servidor i d'aquesta manera no podem mostrar un missatge d'error al usuari quan alguna cosa no ha anat bé. No sabem si es tracta d'una mancança de GWT-Openlayers o del mateix projecte Openlayers, però seria una millora important de cara a un entorn més real.
- Utilitzar Geowebcaché de Geoserver, per tal de reduir la carrega del nostre servidor.
- Crear una pantalla de carrega de dades massiva, partint d'un fitxer en un format concret. Això s'utilitzaria per una carrega de dades inicial, en un entorn més real.
- Crear un sistema de gestió d'usuaris per l'aplicació, de forma que puguem restringir l'accés a la part de manteniment, ja que els usuaris anònims només tindrien que poder accedir a la part de cerca d'informació.

5.10. Capítol 10. Conclusions

En general podem dir que els objectius generals i específics del projecte s'han acomplert:

- Hem après a treballar amb SIG, aprenent conceptes com les diferents projeccions a utilitzar.
- Hem dissenyat una base de dades, utilitzant tipus de dades pròpies de SIG, com per exemple el tipus Geometry de PostGIS i les funcions que hi ha per treballar amb aquest tipus de dades.
- Hem aprofundit en el coneixement del servidor Geoserver i dels serveis que ofereix (WFS i WMS), podent veure el seu funcionament i el tipus de informació que utilitzen.
- Hem après a crear visors de mapes, utilitzant una alternativa lliure com és Openlayers. Ens hem trobat amb diferents problemes deguts a la implementació GWT-Openlayers, però que al final ens ha ajudat a comprendre una mica millor el funcionament del projecte Openlayers.
- Hem importat dades directament d'OpenStreetMap al nostre servidor Geoserver.
- Hem treballat amb la base de dades PostGIS, treballant amb ella utilitzant JDBC.

- Hem après el funcionament del protocol WFS-T, aconseguint realitzar insercions i actualitzacions de registres utilitzant aquest protocol.

6. Bibliografia

[1] Apache Tomcat:

<http://tomcat.apache.org/download-70.cgi>

[2] GeoServer:

<http://geoserver.org/display/GEOS/Stable>

[3] Instal·lació de PostgreSQL 8.4 i PostGIS 1.5:

<http://www.paolocorti.net/2008/01/30/installing-postgis-on-ubuntu/>

[4] Creació de base de dades amb suport PostGIS:

<http://blog.smartlogicsolutions.com/2010/03/04/installing-postgis-1-5-0-on-postgresql-8-4-on-ubuntu-9-10/>

[5] Creació de un nou usuari a PostgreSQL:

<http://www.cyberciti.biz/faq/howto-add-postgresql-user-account/>

[6] Xarxa de Consum Solidari:

<http://www.xarxaconsum.net/>

[7] El Guaret:

<http://elguaret.wordpress.com/>

[8] CCPAE:

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

[9] Mercats Setmanals a Catalunya:

<http://www.mercatsetmanal.cat/>

[10] Fires a Catalunya:

http://www.ruralcat.net/web/guest/gecnews.module.ruralcat/-/journal_content/56_INSTANCE_8hUE/10136/1693369

[11] Cloudmade:

http://downloads.cloudmade.com/europe/southern_europe/spain/cataluna#downloads_breadcrumbs

[12] MySQLWorkbench:

<http://wb.mysql.com/>

[13] Importació de OpenStreetMap a GeoServer:

<http://gismemento.wordpress.com/2009/12/02/import-openstreetmap-data-into-geoserverpostgis/>

<http://blog.geoserver.org/2009/01/30/geoserver-and-openstreetmap/>

[14] Manual GeoServer:

<http://docs.geoserver.org/stable/en/user/>

[15] Eclipse:

<http://www.eclipse.org/downloads/>

[16] Plugin Maven per Eclipse:

<http://www.eclipse.org/m2e/>

[17] Plugin GWT per Eclipse:

<http://code.google.com/intl/ca-ES/webtoolkit/download.html>

[18] Google AdWords (desenvolupat amb GWT):

<http://adwords.google.com/>

[19] Orkut (desenvolupat amb GWT):

<http://www.orkut.com/>

[20] OGC (Open Geospatial Consortium)

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

[21] Maven Download

<http://maven.apache.org/download.html>

Annex 1. Guia ràpida d'instal·lació

Primer farem una breu descripció de la estructura de directoris del lliurament:

- bin: Conté el fitxer .war amb el programari a instal·lar.
- scripts: Scripts de base de dades.
- src: Codi font de l'aplicació.

Indiquem a continuació el programari que necessitarem, per tal de poder instal·lar la nostra aplicació:

| | |
|-----------------|---|
| Apache Tomcat 7 | http://tomcat.apache.org/download-70.cgi |
| PostgreSQL 8.4 | http://www.postgresql.org/download/ |
| PostGIS 1.5 | http://postgis.refractor.net/download/ |
| Geoserver 2.1 | http://geoserver.org/display/GEOS/Download |

Els passos a seguir son aquests:

1. Instal·lem la base de dades **PostgreSQL**. En cas d'utilitzar Windows seguim les instruccions de l'instal·lador que hem descarregat. Si utilitzem Linux podem seguir les instruccions detallades en el punt 5.2.3. d'aquesta memòria.
2. Instal·lem la seva extensió **PostGIS**. En cas d'utilitzar Windows, seguirem les instruccions indicades en la pàgina de descarrega. Si utilitzem Linux, seguirem les instruccions indicades en el punt 5.2.3. d'aquesta memòria.
3. Creem una base de dades anomenada **xcs** i un usuari amb el mateix nom, amb permisos per operar en aquesta base de dades. Per fer-ho podem seguir les instruccions indicades en el punt 5.2.3. d'aquesta memòria.
4. Obrim el programa pgAdminIII:
 1. Executem el script adjunt **creacio_taulas.sql**.
 2. Executem el script adjunt **importacio_dades.sql**.
5. Descarreguem el fitxer **cataluna.osm.bz2** de http://downloads.cloudmade.com/europe/southern_europe/spain/cataluna#downloads_bread_crums i el descomprimim en un directori qualsevol.
6. Ens situem en el directori on hem copiat el fitxer i per línia de comandes, executem la comanda **osm2pgsql -E 900913 -d cataluna.osm**. Podem veure el procediment amb més detall en el punt 5.5.2. d'aquesta memòria.

7. Descarreguem i descomprimim **Apache Tomcat** en un directori conegut. A aquest directori l'anomenarem a partir d'ara **TOMCAT_HOME**. Podem veure el procediment amb més detall en el punt 5.2.1. d'aquesta memòria.
8. Descarreguem Geoserver en versió web archive i copiem el fitxer **geoserver.war**, dins del directori **TOMCAT_HOME/webapps**. Podem ampliar informació en el punt 5.2.2. d'aquesta memòria.
9. Creem l'usuari **xcs** a Geoserver, per fer-ho podem seguir les instruccions del punt 5.2.2. d'aquesta memòria.
10. Iniciem el servidor **Apache Tomcat**, executant el script **TOMCAT_HOME/bin/startup.sh**.
11. Accedim a l'aplicació amb <http://localhost:8080/geoserver>.
12. Fem login a Geoserver amb l'usuari **xcs**.
13. Creem el workspace **xcs**. Més informació en el punt 5.5.2. de la memòria.
14. Creem el datasource **xcs**. Més informació en el punt 5.5.2. de la memòria.
15. Publiquem els següents layers. Per fer-ho podem obtenir més informació en el punt 5.5.2 de la memòria:
 1. xcs:planet_osm_line
 2. xcs:productor
 3. xcs:esdeveniment
 4. xcs:xcs
16. Copiem el fitxer **Xcs-1.2.war** que tenim dins del directori bin del nostre lliurament, al directori **TOMCAT_HOME/webapps**.
17. Accedim a l'aplicació posant la url <http://localhost:8080/Xcs-1.2> en el nostre navegador.

Annex 2. Guia ràpida de compilació

Tot i que en el lliurament es proporciona el fitxer .war amb el programari ja compilat, hem decidit fer una petita guia de com compilar el codi i generar el fitxer.war.

1. Descarreguem la versió 2.2.1 de Maven de la pàgina <http://maven.apache.org/download.html>. També serviria la versió 3 de Maven, però millor descarregar la 2.2.1 que és la que hem utilitzat.
2. Realitzem la instal·lació de Maven, seguint les instruccions que trobem al final de la pàgina de descarrega.
3. Descomprimim el fitxer .zip, que trobem dins del directori **src** del lliurament. El podem descomprimir en el directori que vulguem, a partir d'ara anomenarem al directori del projecte ja descomprimit **PROJECT_HOME**.
4. A través de línia de comandes, accedim al directori **PROJECT_HOME**.
5. Executem la comanda **mvn clean package**. Veurem que Maven ens comença a descarregar totes les dependències que necessita, pot trigar una bona estona el primer cop.
6. Un cop finalitzat, veurem el resultat de “Build Successful” i per tant ja tenim el fitxer .war generat. Trobarem aquest fitxer dins del directori **PROJECT_HOME/target**.
7. [OPCIONAL] Si volem també podem generar un informe, que entre altres coses inclou les dependències utilitzades en el projecte i un link al Javadoc. Per fer-ho només tindrem que executar la comanda **mvn site**. El informe generat el trobarem dins de **PROJECT_HOME/target/site**.