UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA TÈCNICA SUPERIOR D'INGINYERIA INFORMÀTICA

PUNTOS DE INTERÉS EN MAPAS DINAMICOS

PROJECTE FINAL DE CARRERA

Titulació:**Enginyer Tècnic en Informàtica de Sistemes**
Alumne: **Vicente Javier Mozos Pérez**
Tutor Acadèmic: **Alvaro Doménech Pujol**

ÍNDEX

Índex d'imatges

1. Introducció

Gairebé sense adonar-nos, Internet s'ha implantada en les nostres vides i ens hem acostumat a utilitzar-ho en molts camps de la nostra vida quotidiana. Cada vegada queden menys llocs per ser conquistats per aquesta tecnologia, permetent moure'ns pel món en qüestió de segons. Si a això li anem sumant que cada vegada els equips informàtics són més potents i que tota persona pot tenir accés a un, ens trobem amb que el nombre d'usuaris de la xarxa augmenta cada vegada més Internet té moltes aplicacions, així doncs podem trobar a gent que utilitza cercadors, que realitza compres per Internet, consulta guies de viatge… Molts perfils diferents d'usuaris, però tots ells tenen alguna cosa en comú: Recerca d'informació en pàgines Web dinàmiques.

Ara bé, Internet guanya molt interès quan determinades aplicacions d'organismes o empreses faciliten informació, permetent interactuar amb elles, pràcticament en temps real i estalviant-nos així temps de desplaçament als diferents llocs. És per això que, amb l'increment del nombre d'usuaris que, des de casa o oficina busquen informació, aquestes empreses o organismes decideixen posar a la xarxa informació d'interès general per a l'usuari que consulta aquesta pàgina. No obstant això i com a complement, també s'utilitza la Intranet, que a diferència d'Internet, el nombre d'usuaris que pot accedir a la informació és menor, requereix de contrasenyes i és més segura.

També hi ha hagut un gran desenvolupament en les aplicacions de lliure distribució, això permet el lliure ús de les mateixa sense haver de realitzar una gran inversió, tenint molta més llibertat d'elecció de programari.

En aquest projecte es desenvolupa una aplicació que ens permet conèixer els diferents tipus de perill que es donen en una determinada zona d'una cartografia i les persones que es veuen afectades, tot mitjançant programari, fonamentalment, lliure i utilitzant Internet com a mitjà de publicació, diferenciant dos tipus d'usuaris (autoritzats i no autoritzats) perquè puguin consultar i editar la informació o solament consultar la informació.

2. Objectius

En aquest projecte final de carrera s'ha desenvolupat una aplicació que persegueix els següents objectius:

Un servei municipal de neteja necessita una aplicació que permeta als empleats anotar i consultar els focus de contaminació existent en la ciutat de València, aqueix programa ha d'oferir la possibilitat d'accedir a ell des de qualsevol tipus de dispositiu mòbil, ordinador portàtil, "smart phone", tablet pc, etc. L'aplicació ha de permetre:

L’ objectiu d'aquest projecte és el de donar una solució, al problema de la localització de focus contaminants en la ciutat de València.
Per a açò primer es farà un estudi sobre els tipus de focus contaminants, com es van a representar en el mapa, etc. Els punts contaminants deuen estar georreferenciats. També es requereix que els empleats del la empresa encarregada de la neteja de la ciutat de València, tinguen accés a aquesta informació tant per a consultar-la com para modificar-la.

L'aplicació té les següents característiques:

- Inserció i edició de dades puntuals a través d'Internet.
- Creació d'àrees d'influència per l'usuari quan inserte o consulte un punt.
- Diferenciació de funcions de l'aplicació per usuaris autoritzats i no autoritzats.
- Disseny d'una aplicació per a tots els usuaris (experts i no experts) puguin usar-la.

Utilització de programes lliures per resoldre els problemes que es plantegin, sempre que resulti factible i fiable

Finament i com és obvi, aquest projecte es realitza per finalitzar la titulació d'Enginyer tècnica en informàtica de sistemas.

3. Descripció del problema

3.1. Petició del client

L'empresa de neteja municipal de l'ajuntament de València, ens planteja el següent problema: a causa del constant increment de població en la ciutat de València, han anat sorgint incessants focus de contaminació, alguns d'ells han de ser ràpidament esmenats. Però el problema al que s'enfronta la citada empresa de neteja, és la lentitud a l'hora de passar els avisos a la central de neteja, aquestos eren passats bé pels ciutadans, en aquesta cas en ocasions l'avís o era fals o no coincidia amb la descripció, per la qual cosa es perdia temps en aquests avisos poc exactes o bé eren els propis treballadors de la neteja que es trobaven amb els focus contaminants, però no s'informaven d'ells fins a la finalització de la jornada laboral, quan arribaven a la central.

3.2. Requeriments

L'empresa ens demana que li donem una solució al problema de localització dels focus contaminants i al problema dels retards a l'hora d'informar a la central dels mateixos. També necessita tenir actualitzat en tot moment els focus contaminants i que aquests siguen accessibles pels operaris de neteja, sent aquests mateixos els que puguen consultar mitjançant algun sistema que permeta localitzar inequívocament el lloc exacte del focus, a més de poder donar-los de baixa quan siguen esmenats.

A més l'empresa requereix que es puga portar un històric dels focus contaminants daus d'alta en la cuideu de València, per a poder fer informes, estadístiques i portar accions de millora en els serveis de neteja.

L'últim requeriment de l'empresa és que el cost de l'aplicació que de solució a les seves peticions, sigui el més ajustada possible.

3.3. Proposta de solució

D'acord als requeriments de l'empresa l'alumne pren les següents decisions:

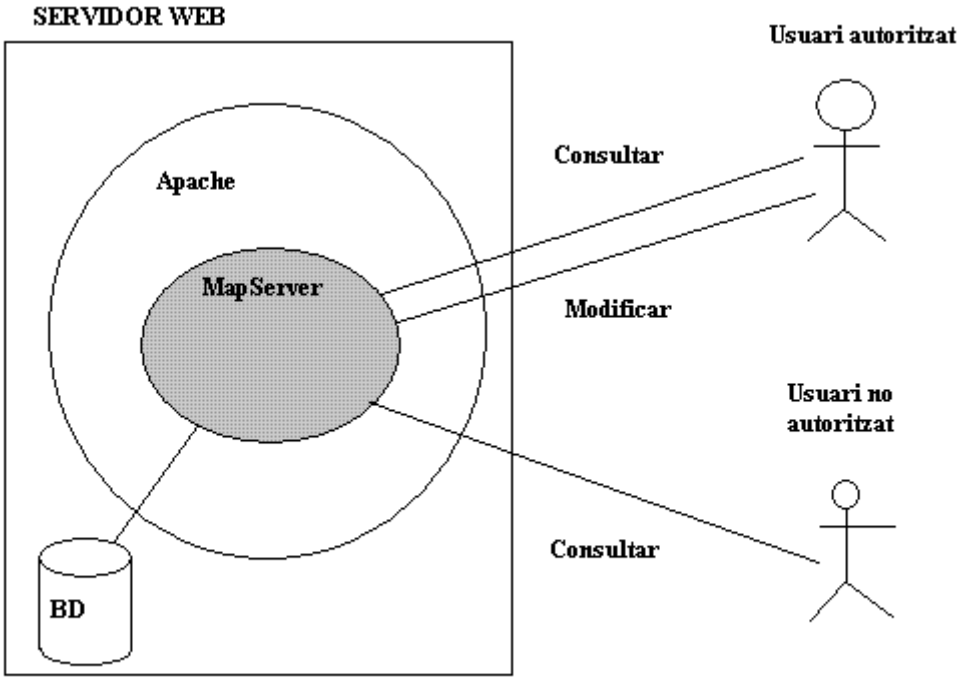
- Per respondre a la petició de tenir actualitzat i totalment accessible la informació dels focus contaminants es proposa, crear un servidor web, que al seu torn allotgi un servidor de mapes, amb la possibilitat d'inserir punts (que seran els focus contaminants), modificar-los, actualitzar-los, esborrar-los a més a més, donarà la possibilitat de tenir els focus georreferenciats, per a una major exactitud.

Opcionalment, per a donar més realisme, el servidor de mapes, podrà carregar una ortofoto, a través d'un servidor de l'Institut Cartogràfic Valencià (d'ara endavant ICV), però es podrà treballar perfectament sense aquesta.

- Per poder donar resposta a la necessitat de tenir un històric de totes les incidències detectades i esmenades, l'aplicació es recolzarà en una base de dades, per poder emmagatzemar la informació requerida per l'empresa, respecte a focus contaminant.

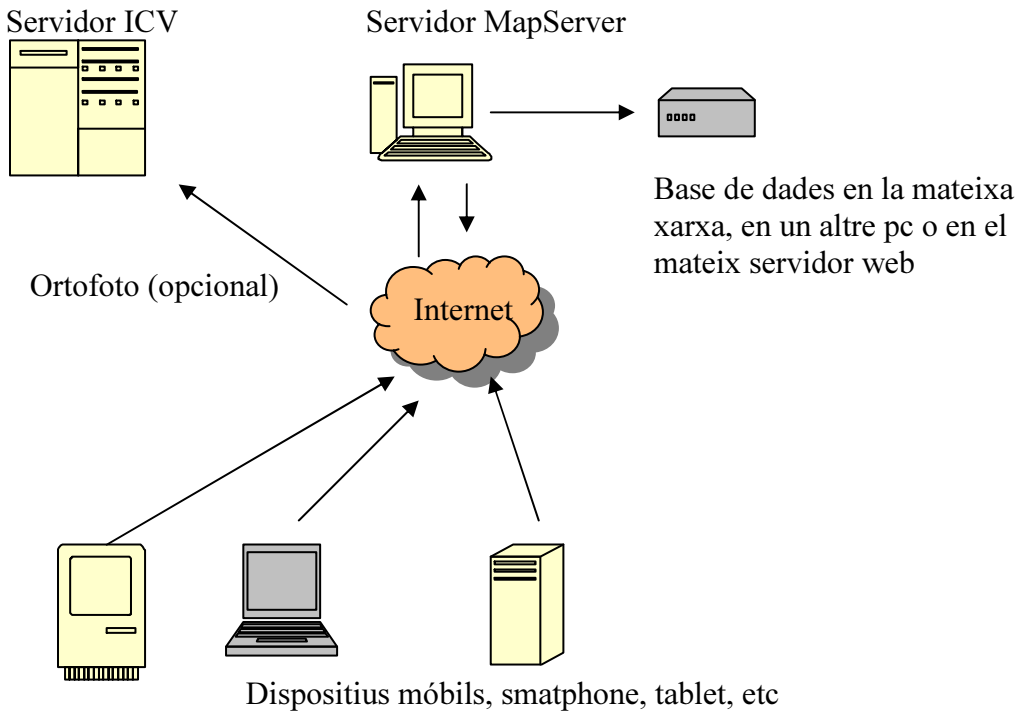
- Per a seguir les instruccions de baix cost, demanat per l'empresa, s'utilitzarà sempre aplicacions lliures, tals com: PostGis, PostGres, Mapserver, algunes d'aquestes estan recolzades per l'ICV

- Com a possible actualització de l'aplicació l'alumne proposa afegir al focus contaminant un àrea d'influència o *buffer*, amb la qual podrem conèixer que persones han estat afectades pel focus, a través d'una altra base de dades.



Imatge 1: Casos d'us

3.4. Esquema de l'aplicació



Imatge 2: Esquema d'aplicació

4. Conceptes teòrics

4.1. Introducció als sistemes d'informació Geogràfica (SIG)

Un Sistema d'Informació Geogràfica (SIG o GIS, en el seu acrònim anglès [Geographic Information System]) és una integració organitzada de maquinari, programari i dades geogràfiques dissenyada per capturar, emmagatzemar, manipular, analitzar i desplegar en totes les seves formes la informació geogràficament referenciada amb la finalitat de resoldre problemes complexos de planificació i gestió. També pot definir-se com un model d'una part de la realitat referit a un sistema de coordenades terrestre i construït per satisfer unes necessitats concretes d'informació. En el sentit més estricte, és qualsevol sistema d'informació capaç d'integrar, emmagatzemar, editar, analitzar, compartir i mostrar la informació geogràficament referenciada. En un sentit més genèric, els SIG són eines que permeten als usuaris crear consultes interactives, analitzar la informació espacial, editar dades, mapes i presentar els resultats de totes aquestes operacions.

La tecnologia dels Sistemes d'Informació Geogràfica pot ser utilitzada per a investigacions científiques, la gestió dels recursos, gestió d'actius, l'arqueologia, l'avaluació de l'impacte ambiental, la planificació urbana, la cartografia, la sociologia, la geografia històrica, el màrqueting, la logística per nomenar uns pocs. Per exemple, un SIG podria permetre als grups d'emergència calcular fàcilment els

temps de resposta en cas d'un desastre natural, el SIG pot ser usat per trobar els aiguamolls que necessiten protecció contra la contaminació, o poden ser utilitzats per una empresa per situar un nou negoci i aprofitar els avantatges d'una zona de mercat amb escassa competència.

4.1.1. Definició dels SIG

Un Sistema d'Informació Geogràfica (SIG) és un conjunt d'elements que estan relacionats d'una manera ordenada seguint unes regles que potencien la seua explotació de la informació que conté. La informació que trobem al SIG té definit un sistema de referència i té qualitats mètriques; en definitiva està posicionada a l'espai i ens dóna una idea real de les dades que es representen. Tot Sistema d'Informació Geogràfica té com a "motor" una base de dades que s'ha de construir correcta i cuidadosament (açò és una etapa molt important, que després condicionarà l'eficiència del SIG), a més a més cal un manteniment periòdic de la base de dades perquè estiga actualitzada i no es quede desfasada.

4.1.2. Aplicacions dels SIG

Els sistemes d'informació geogràfica s'utilitzen fonamentalment per obtindre resultats de determinades consultes o anàlisi que es realitzen mitjançant un estudi geogràfic i alfanumèric. Normalment es planteja un problema geogràfic i es fa un estudi, imposant condicions que s'han de complir, aleshores s'inicia el procés d'anàlisi geogràfic utilitzant ferramentes que van donant els resultats que busquem. Finalment s'obté el producte final amb el resultat de les consultes i anàlisi.

4.1.3. Components d'un SIG

Perquè un sistema d'informació geogràfica funcione correctament i s'obtinguen uns resultats fiables cal que els components que s'especifiquen a continuació participen com a conjunt aplicant el coneixement adequat.

- Usuaris dels SIG: Han de ser especialistes en el tema d'informació geogràfica, perquè es puga fer una anàlisi adequada i així es puga desenvolupar una correcta implementació dels processos d'estudi i anàlisi. Sense usuaris experts la informació es desordena i es manipula incorrectament, deixant el potencial de les eines per baix de les seues capacitats.

- Programes: Amb els programes SIG es proporcionen les eines i funcionalitats necessàries per emmagatzemar, analitzar i mostrar la informació geogràfica. Actualment, la majoria dels proveïdors de programes SIG distribueïxen productes fàcils d'usar i poden reconèixer informació geogràfica estructurada en molts formats distints.

- Equips informàtics: Els SIG funcionen en un ampli rang de tipus d'ordinadors, des d'equips centralitzats fins a configuracions individuals o de xarxa. Aquesta organització requereix d'equips específics per complir amb les necessitats cada aplicació.

- Dades: Aquest és el component més important per un SIG. Es requereixen bones dades de suport perquè el SIG puga resoldre els problemes i contestar a les preguntes de la forma mes encertada possible. Les dades geogràfiques i alfanumèriques poden obtindre's per recursos propis o obtindre's a través de proveïdors de dades però en

tot cas cal revisar-les i corregir-les dels errors que s'han anat filtrant durant el procés d'obtenció.

- Procediments: Els procediments són el conjunt de coneixements que es posen en pràctica perquè la implementació de la anàlisi siga exitosa, amb mètodes ben definits i consistents per a produir resultats correctes i que es puguen reproduir.

4.2. Capes en sistemas SIG

Les capes són una forma de diferenciar les dades. El sistema permet separar la informació en diferents capes temàtiques i les emmagatzema independentment, permetent treballar amb elles de manera ràpida i senzilla, facilitant al professional la possibilitat de relacionar la informació existent a través de la topologia dels objectes, amb la finalitat de generar una altra nova que no podríem obtenir d'una altra forma.

4.3. Sistemes de referència

Un sistema de referència o és un conjunt de convencions usades per un observador per a poder mesurar la posició i altres magnituds físiques d'un objecte o sistema físic en el temps i l'espai.

Diversos tipus de sistemes de referències:

- sistema de referència EPGS: 23030
- sistema de referència EPGS:23028
- sistema de referència EPSG:4326

5. sofware necessari

Realitzar una aplicació completa utilitzable per un navegador d'Internet, en què com s'utilitzi gvSIG com a editor cartogràfic, MapServer com a servidor de mapes, PosGIS com a base de dades geoespacial, PHP com a llenguatge de programació i Apatxe com a servidor de pàgines Web.

5.1. Elecció del sofware necessari

Per poder fer una bona elecció s'ha de conèixer els tipus de programari que actualment ens podem trobar al mercat.

- *Software* lliure.
    - Open source.
    - De domini públic.

  Protegido con *copyleft.*

  No protegit amb *copyleft.*
    - Amb licencia GPL.
    - Sistema GNU
    - Software no lliure.
    - Software semi-lliure.
- *Freeware*
- *Shareware*

No obstant això en aquest projecte ens centrarem al programari lliure i al programari no lliure, sense diferenciar dins d'aquest segon grup.

5.2. Programari lliure

Perquè un programari sigui lliure ha de complir unes característiques determinades, aquestes característiques es diuen "llibertats", que com a curiositats són les següents:

- Llibertat d'execució del programa per a qualsevol propòsit.
- Llibertat d'estudi del programa accedint al codi font.
- Llibertat de distribució de còpies.
- Llibertat de millorar el programa i publicar les millores perquè es benefici.

Llavors tindrem programari lliure si els usuaris gaudeixen d'aquestes llibertats. Podem pensar que el programari lliure és gratuït, en moltes ocasions és d'aquesta manera, però no necessàriament ha de ser-ho, és possible que es cobri un preu simbòlic en concepte de distribució o de permís.

També s'ha de saber que els usuaris de programari lliure te la llibertat d'usar-ho per a qualsevol propòsit o treball, podent modificar el codi font sense estar obligats a comunicar-ho als programadors o entitats específiques. No obstant això es recomana que totes les millores siguin comunicades a les entitats corresponents amb tal d'avançar i poder beneficiar a la resta d'usuaris.

Un avantatge molt considerable que té el programari lliure, és que normalment és que esta recolzat pels usuaris. Quan algú té algun problema, es posa en contacte amb altres usuaris o amb l'organització que ho desenvolupa, de manera que hi ha un contacte gairebé directe entre desenvolupadors, usuaris i comunitat en general. També són les usuaris que contribueixen a l'evolució del programari, gràcies a consells, reportes d'errors, fent de testers. En definitiva amb la contribució dels usuaris s'avança molt més i més ràpid.

5.3. Programari no lliure

És aquell programari que no està inclòs en l'apartat anterior, per tant no gaudeix de les llibertats. De totes les categories que hem definit anteriorment, podem agrupar el programari no lliure en:

- Programari no lliure: engloba el programari semilibre i el privatiu.
- Programari semi-lliure: programari no lliure, però els usuaris poden utilitzar-ho, copiar-ho i modificar-ho sense finalitats lucratives.
- Programari privatiu: és aquell que no és lliure ni semilibre i no es pot copiar ni distribuir ni utilitzar-ho sense autorització o llicència.
- Freeware: És el programari que es pot distribuir però no admeten modificacions, perquè el codi font no està disponible. No s'ha de confondre amb programari lliure perquè no ho és.
- Shareware: Aquest tipus de programari permet la seva distribució, però per obtenir cada còpia cal pagar una llicència. Normalment el codigoc fientes no aquesta disponible.
- Programari privat: És el programari dissenyat per a un usuari concretament, per tant és ell el que ho utilitza i el que disposa del codi font i no ho comparteix al públic.

- Programari comercial: aquell que aquesta desenvolupat per una empresa que normalment pretén obtenir una compensació econòmica

El programari que no és lliure té una manera diferent d'evolucionar i de ser consultat, ja que si hi ha algun problema s'ha de demanar directament assistència a l'empresa que ho comercialitza i en estar el codi tancat, es tanca la possibilitat als usuaris la col·laboració directa d'aquests. Com és reduït el nombre d'usuaris que tenen accés a la informació, també és major el temps d'espera de les respostes als problemes oposats.

5.4. Comparativa. Decisió final

Com actualment tant el programari lliure com el no lliure ens ofereixen moltes possibilitats, per desenvolupar aquest projecte s'ha donat prioritat al programari lliure. No obstant això, ha estat necessari l'ús de programari no lliure per a algunes coses puntuals.

Per triar un programari o altre ens hem basat a veure com era més optim i d'alguna manera el més potent o veterà. Per exemple, per desarrolar la pàgina web s'ha procurat fer-la amb programes lliures que son potents i a priori se sap com van a respondre, també és una forma de mostrar que existeixen alternatives a la metodologia que habitualment s'usa (programari no lliure).
Com s'he comentat abans, no tota l'aplicació s'ha desenvolupat amb programari lliure. El tractament de la cartografia s'ha utilitzat no programari lliure, perquè resulta una eina molt més potent, a pesar que és antic i desfasat. Ens referim a l'ortofoto, encara que es opcional.

Pel que afecta a la resta d'aplicacions creades al projecte, han estat desenvolupades amb programari lliure, perquè d'aquesta manera permetia cobrir els objectius, i a més, uns dels objectius era donar prioritat al programari lliure enfront del no lliure.

També hem d'afegir una de les condicions de l'empresa que encarrega aquest programa, que és el baix pressupost del que disposa.

5.5. Programari necessari

5.5.1. Editor de textos

Per editar i crear el codi que volem programar hem de tenir instal·lats un editor de texts. Si el treball no és molt gran es podria utilitzar el bloc de notes, però és molts mes recomanable utilitzar un editor que diferenci els tipus de codi. Existeixen molts tipus d'editors: editors de PHP, notepad++, etc.

En aquest projecte s'ha utilitzat l'editor "SciTE" perquè diferencia amb colors els tipus de codi en què estem programant. Ademas no necessita instal·lació i és molt senzill d'utilitzar.

Imatge 3: Símbol scite.

5.5.2. Apache

El servidor HHTP Apache és un programa lliure que serveix com a servidor HTTP de codi obert per a plataformes UNIX (BSD, GNI/Linux, etc), Windows i Machintosh. El servidor apatxe es desenvolupa dins del projecto HTTP Server de l' Apache Programari Foundation.

Apache té un àmplia acceptació en al xarxa: des de l'any 1996 és un dels servidors més utilitzats. L'any 2005 va ser utilitzat per un 70% dels llocs web del món.

5.5.3. Paquet MS4W

Normalment quan necessitem algun programa pensem en el com una unitat, però algunes vegades existeixen paquets que contenen tota o una part del que necessitem. En aquest cas, el paquet MS4W conté gran part del que es necessita per resoldre el problema plantejat.

- PHP.
- PHP MAPSCRIPT.
- MAPSERVER.

5.5.3.1. Mapserver

MapServer és un entorn de codi obert que s'utilitza en la creació d'aplicacions SIG en Internet o en la intranet. La funcionalitat de MapServer és visualitzar, consultar i analitzar informació geogràfica a través de les xarxa mitjançant la tecnologia IMS (Internet Map Server). Algunes de les característiques de Mapserver són:

- S'executa baix les plataformas Linux i Windows
- Els formats vectorials que suporta són ESRI Shapefiles, PostsGIS, ESRI, ArcSDE, GML, entre uns altres
- Els formats raster qué suporta són: JPG, PNG, GIF, TIFF, EPPL7 i uns altres via GDAL

El seu funcionament bàsic està configurat en un fitxer de text, que generalment té l'extensió ".map". En aquest fitxer, les dades del mapa s'organitzen en capes, al seu torn dividida en una o més classes, on en cadascuna de les quals es poden definir diferents estils visuals. Aquesta estructura permet la generació de mapes amb una definició d'estils molt flexible, que també pot dependre de l'escala del mapa.

El format eixida de MapServer, depenent de la sol·licitud, pot ser gràfic (mapa, llegenda, escala, mètriques, visió general) o alfanumèric (el resultat d'una consulta de dades alfanumèriques o espacial). L'arxiu ".map" també inclou la possibilitat de fusionar la producció d'una plantilla d'HTML MapServer, per a generar una pàgina web de lectura fàcil i agradable.

La possibilitat de ser utilitzat com a servidor de mapes per tercers programes, seguint les especificacions del OGC, o bé mitjançant la API MapScript (l'opció que s'utilitzarà en aquest projecte), ha portat a la creació d'aplicacions web basades en MapServer, per a la publicació de dades geoespaciales, com ara:

- CartoWeb
- Ka-Map
- Chameleon
- Pmapper

6. Coneixements necessaris

6.1. HTML

Aquestes sigles signifiquen HyperText Markup Language (Llenguatge de marques de hipertext) [7] i és el llenguatge per a la construcció de pàgines web. El llenguatge HTML s'utilitza per descriure l'estructura i el contingut en forma del text i de les imatges. HTML és un llenguatge que s'escriu entre etiquetes. Aquestes etiquetes van envoltades pels símbols "<>". HTML també pot tindre inclòs algun script que pot modificar l'aparença o les funcions de la pàgina web.

6.1.1. Conceptes bàsics d'HTML

El llenguatge HTML es pot crear i editar en qualsevol editor de texts bàsic, però existeixen programes que ens permeten dissenyar la pàgina web, editar codi, etc. Aquestos programes són els que normalment es solen utilitzar per a crear pàgines web. Aquestos editors es coneixen per editors WYSIWYG (What You See Is What You Get) perquè el resultat final que s'obté és el que s'ha estat veient a la vista preliminar del programa de disseny. Alguns editors són DreamWeaber, Microsoft FrontPage... No obstant existeixen uns altres tipus d'editors.

Per visualitzar una pàgina web, només cal indicar la URL (Localitzador Uniforme de Recursos) al navegador que s'ha elegit. Normalment totes les adreces d'Internet solen començar per "www", es pot dir que és el sistema d'informació propi d'Internet, i té algunes de les següents propietats [9]:

Informació per hipertext: part de la informació o de les imatges que veiem estan vinculats, i per mostrar la informació podem fer clic a l'enllaç.

Tipus d'informació: poden aparèixer a les pàgines web informació escrita, imatges, vídeos, sons...

És global: perquè podem accedir a ell des de qualsevol plataforma i utilitzant qualsevol navegador i estant en qualsevol part del món.

Pública: tota la informació pot estar distribuïda per tot arreu, a qualsevol lloc del món.

Dinàmica: en el sentit que la informació pot ser actualitzada.

6.1.2. Organització de les pàgines web

Una pàgina web ha d'estar ben organitzada. Normalment es pot fer un esborrany al principi, per exemple primerament es marquen els objectius, els continguts que tindrà la pàgina web, l'estructura que busquem, etc.

Una bona estructura és important perquè es facilita la lectura de la pàgina web, fonamentalment l'estructura depèn del contingut que tindrà la pàgina web [9].

6.1.3. Llenguatge HTML

Aquest llenguatge té la funció d'estructurar documents. La majoria dels documents tenen estructures comunes (títols, paràgrafs, llistes...) que seran definides per aquest llenguatge per mitjà d'etiquetes. Qualsevol cosa que no siga una etiqueta és part del document.

Aquest llenguatge no descriu l'aparença del disseny d'un document sinó que ofereix a cada plataforma que li done format segons la seua capacitat i la del seu navegador (grandària de la pantalla, fonts que n'hi ha instal·lades...). Per això, no hem de dissenyar els documents basant-nos en com lluïxen en el nostre navegador sinó que hem de centrar-nos a proporcionar un contingut clar i ben estructurat que resulte fàcil de llegir i entendre [9].

HTML té dos avantatges que ho fan pràcticament imprescindibles a l'hora de dissenyar una presentació web: la seua compatibilitat i la seua facilitat d'aprenentatge a causa del reduït nombre d'etiquetes que utilitza [9].

Bàsicament, els documents escrits en HTML consten del text mateix del document i les etiquetes que poden portar atributs. Açò portat a la pràctica, vindria a ser:

<*Etiqueta*> text afectat </*etiqueta*>

L'etiqueta del principi activa l'ordre i l'última (que serà la del principi precedida del signe "/") la desactiva. Generalment açò funciona d'aquesta manera, però no totes les etiquetes tenen principi i final, perquè no cal tancar l'etiqueta en tots els casos.

Els fitxers de text es converteixen en pàgines web quan salvem el document amb extensió "*.html". Aquests tipus de pàgines són pàgines estàtiques normalment, sempre que no continguen un altre tipus de codi.

6.1.4. Estructura d'un document HTML

Bàsicament són tres les etiquetes que descriuen l'estructura general d'un document HTML. Són etiquetes de filtrat perquè no canvien l'aparença del document. Aquestes etiquetes són:

<HTML>: Aquesta etiqueta limita el document HTML, indicant el començament i el final del document HTML.

<*HEAD*>: Dins d'aquesta etiqueta es troba la capçalera de la pàgina web, el contingut introductori. També trobem l'etiqueta "*TITLE*" que s'utilitzarà per identificar el contingut de la pàgina web, només cap la possibilitat d'utilitzar una única etiqueta "*TITLE*". En l'etiqueta "*HEAD*" cal no posar text que haja d'estar en el document.

<*BODY*>: En aquesta etiqueta es col·locarà tot el text de tota la pàgina web. Dins de l'etiqueta "*BODY*" podem col·locar molts tipus d'etiquetes, com poden ser "<P>" per separar paràgrafs, "<IMG>" per inserir imatges, "<HR>" per dibuixar una línia horitzontal, "<H1>" per indicar un títol d'importància 1, "<H2>" per

indicar un títol d'importància 2… i com aquestes etiquetes moltes més que podem utilitzar [6].

A la imatge següent es pot observar l'esquema bàsic que té una pàgina web:

```
<HTML>
  <HEAD>
    <TITLE>PRIMERA P&Aacute;GINA</TITLE>
  </HEAD>
  <BODY>
  </BODY>
</HTML>
```

Imatge 4: Esquema pàgina web

En la imatge següent es mostra un exemple bàsic des d'el punt de vista de l'editor de textos, on escrivim el text que programem, i la vista des d'el punt de vista del navegador d'Internet.

```
<HTML>
  <HEAD>
    <TITLE>TÍTULO</TITLE>
  </HEAD>
  <BODY>
   <HR>
    <P>Seaparación de línea.</P>
    <BR>Inserci&oacute;n de imagen <IMG SRC="./images/luna.jpg" ALT="La luna" WIDTH=100 HEIGHT=50>
    <!--ALT es un texto que en caso de que la imagen no pueda visualizarse
    en la página, saldrá.-->
  </IMG>
  </BODY>
</HTML>
```

Imatge 5: Exemple etiquetes. Punt de vista editor

L'anterior codi produeix el següent resultat:

Imatge 6: Exemple mostrat al navegador

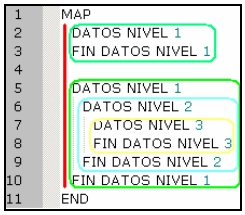
6.2. MapServer

Com s'ha dit abans, *MapServer* és un servidor de cartografia que ens permet visualitzar i consultar cartografia per Internet. La configuració del fitxer, que tindrà extensió "*.map" es mostrarà a continuació, així com la seua estructura jeràrquica d'apartats.

El fitxer "*.map" segueix una estructura jeràrquica en les diferents propietats. Cadascuna de les propietats o objectes han d'estar correctament ubicades a l'arbre

del fitxer, açò no significa que si no s'identa correctament el fitxer no funcione, però si ens ix algun tipus d'error (que sol ser habitual) serà molt complicat trobar-lo en una estructura poc ordenada. A més a més, una bona sangria ens permet visualitzar fàcilment la jerarquia del document. L'objecte principal és el tipus "*MAP*", per tant totes les característiques es col·locaran dins de les etiquetes "*MAP*" i "*END*".

Com a exemple es mostra un esquema del fitxer "*MAP*" on es mostra una correcta sangria segons el nivell de les dades que estem definint:



Imatge 7: Jerarquia fitxer "map"

6.3. PHP

PHP, que significa *Hypertext Pre-processor*, és un llenguatge de programació interpretat, un llenguatge que està dissenyat per a ser executat a través d'un intèrpret, i que dóna un aspecte dinàmic a les pàgines web [7].

PHP és un llenguatge multiplataforma orientat a web que té capacitat de connexió amb la majoria dels motors de base de dades, destacant *MySQL* y *PostgreSQL*. Un avantatge important és que és un llenguatge lliure i que permet les tècniques de programació orientat a objectes. Per programar en PHP no és necessari tindre una metodologia determinada, ni cal declarar el tipus de les variables abans d'utilitzar-les [7].

Les pàgines web que estan fetes purament amb HTML no permeten una interacció amb elles, és per açò que es necessita de codi PHP per donar un aspecte dinàmic a la pàgina web. A més a més, el manteniment de les pàgines és molt menys complicat perquè PHP pot tindre opcions d'una base de dades, i per tant només seria necessari actualitzar la base de dades. HTML per sí mateix no permet aquesta opció. És necessari que una pàgina web tinga més codi a part de HTML perquè permeta ser una pàgina web dinàmica connectant a una base de dades o un altre tipus d'informació.

6.3.1. Funcionament PHP

Des d'el nostre ordinador es carrega una pàgina al navegador d'Internet i es visualitza una pàgina web que pot ser un formulari(per a introduir dades, per exemple quan emplenem una fitxa en registrar-nos en un fòrum). En veure la pàgina web es poden omplir els camps que demana al formulari i polsar el botó que enviarà la informació que s'ha escrit a la pàgina a un servidor d'Internet i en aquest moment és quan el codi PHP és interpretat i per exemple, es podria accedir a una base de dades. El resultat s'envia al navegador, normalment una pàgina HTML.

Imatge 8: Esquema del funcionament de PHP [11]

Com PHP s'executa al servidor no és necessari que el navegador el suporte perquè és independent d'ell, però és fonamental per a un bon funcionament que el servidor suporte codi PHP.

6.3.2. Avantatges de fer servir PHP

Abans s'ha comentat que PHP permet crear pàgines web dinàmiques, però també té altres utilitats com per exemple incloure encapçalaments, peu de pàgina per a estandaritzar pàgines HTML, controlar l'accés a les pàgines web, etc.

Alguns avantatges d'utilitzar PHP són les següents:

El codi que interpreta o genera pàgines web es pot incloure al mateix arxiu (en el qual s'estiga treballant, per exemple l'arxiu de la pàgina web principal d'aquest projecte) dins d'una etiqueta HTML. És més senzill de mantenir les pàgines web que són molt dinàmiques.

És un llenguatge en què resulta fàcil d'integrar l'accés a base de dades (connectar-se a elles) en pàgines HTML, com per a presentar un formulari que es modifique.

També hi ha funcions per a mostrar estadístiques del servidor.

És un llenguatge paregut en sintaxis i gramàtica al llenguatge C, però menys complexe. Les funcions són paregudes a Perl.

Es poden definir funcions pròpies.

Es poden escriure expressions lògiques i matemàtiques.

Ens permet combinar la potència de les dades organitzats lògicament i emmagatzemats de forma persistent amb la facilitat d'ús d'un navegador.

PHP té funcions per obrir i tancar connexions a les bases de dades, per a enviar sentències SQL y rebre el resultat de la consulta.

6.3.3. Conceptes bàsics de PHP

Al fitxer on s'està programant la pàgina web inclourem el codi PHP dins del codi HTML, al mateix fitxer. La manera en que inserirem en qualsevol part del codi HTML és de la manera que es mostra [11]:

<?php

codi
?>

Quan s'han inserit les etiquetes que indiquen el codi PHP ja podem començar a programar PHP. Per poder programar es necessita declarar les variables, variables que durant l'execució del programa poden variar de contingut. A pesar que les variables s'han de declarar, no cal definir-les abans d'usar-la de manera que la mateixa variable al principi de l'execució pot tindre el valor de "3" i al final de l'execució pot tindre el valor de "hola" [11].

Com en altres llenguatges de programació (C o Javascript) PHP també té els operadors aritmètics, comparatius i lògics. D'una manera ràpida es mostren a continuació [11]:

Aritmètics: suma (+), resta (-), multiplicació (*), mòdul (%), suma un (++) i resta un (-).

Comparatius: usats per a prendre valors i poder prendre decisions, estos són els d'igual (==), distint (!=), menor que (<), major que (>), menor o igual (<=), major o igual (>=).

Lògics: són usats per a avaluar diverses comparacions, combinant els possibles valors d'estes, com "i" (&&, and), "o" (||, or), "no" (!).

També es poden utilitzar instruccions com condicionals, bucles, cadenes, etc. que ens ajudaran a filtrar unes altres instruccions que s'executaran segons les condicions o les instruccions de filtrar que estan pel codi.

A continuació es mostra com es col·loca el codi PHP i el resultat que es visualitza al navegador. L'exemple és molt bàsic però ens dóna una idea de com funciona PHP:

```
1  <html>
2    <head>
3      <title>Ejemplo de PHP</title>
4    </head>
5    <body>
6    Parte de HTML normal.
7    <BR><BR>
8    <?php
9      echo "Parte de PHP<br>";
10
11     for($i=0;$i<10;$i++) {
12       echo "Linea ".$i."<br>";
13     }
14   ?>
15   </body>
16 </html>
```

Imatge 9: Exemple codi PHP inserit en HTML

6.3.3.1. Variables per valor i per referència

Es pot treballar de dues maneres, respecte a les variables, en PHP. Una d'elles és tractar les variables per valor; açò significa que quan una variable és passada, per exemple a una funció ,per valor, el mètode rep una "còpia" del valor de la variable, i per tant, els canvis que li fem a aquesta variable dins d'aquesta funció no afecten a la variable original [21].

No obstant, quan passem per referència una variable a un mètode i es fan variacions dins del context de la funció, en aquest cas sí que afecta a la variable original [21].

Emprarem un tipus o un altre segons quines siguen les necessitats de la nostra aplicació. En el nostre cas, s'han utilitzat els dos tipus segons les funcions i segons els valors que es volien obtindre, valorant en cada cas els nostres interessos.

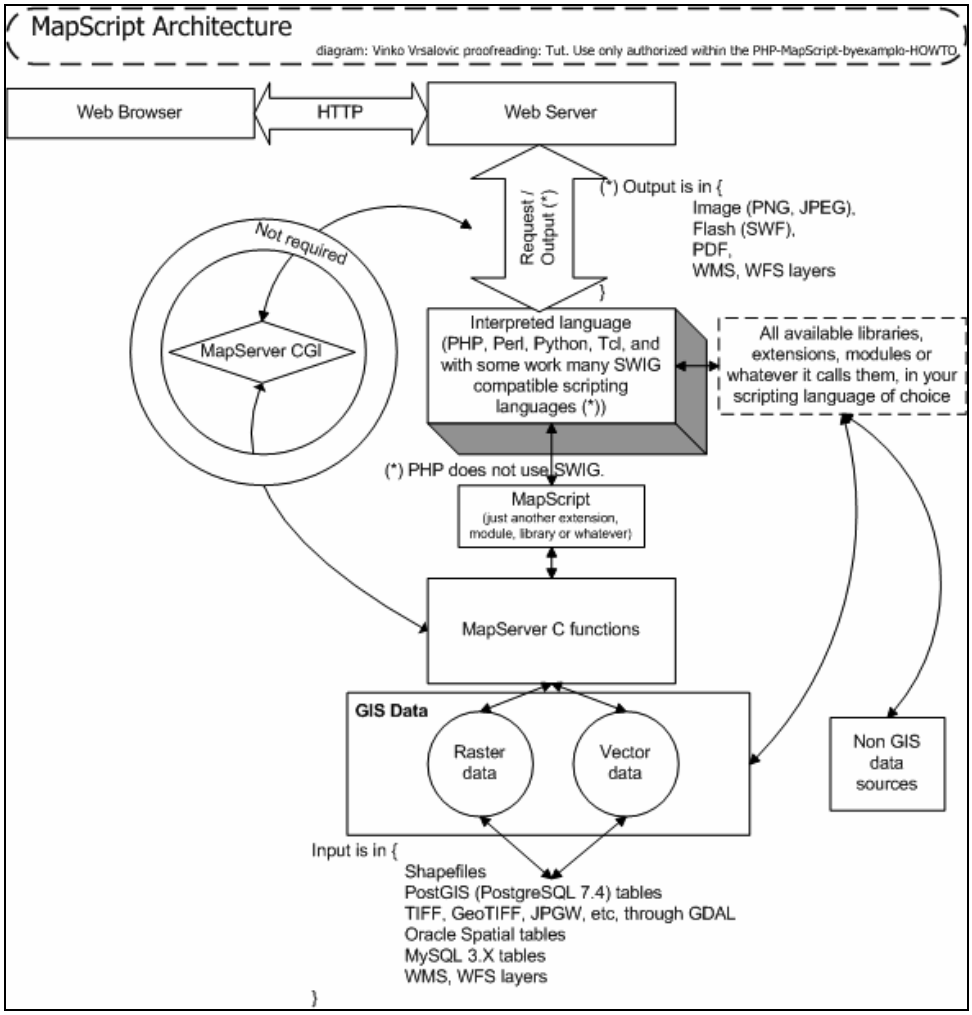
6.4. PHP MapScript

Bàsicament PHP *MapScript* és una extensió de *MapServer* que permet accedir a la seua funcionalitat des de PHP. En aquest projecte es vol fer una pàgina web dinàmica amb PHP i alhora s'ha de treballar amb cartografia, que és *MapServer* qui dóna aquesta funcionalitat; però s'ha de tenir en compte que entre PHP i *MapServer* ha d'existir un nexe comú pel qual els dos llenguatges puguen entendre's. Bé, el nexe en comú s'anomena PHP *MapScript* [12].

Però… com treballa PHP *MapScript*? La manera en que treballa s'explica a continuació. En general, quan es vol accedir a una pàgina, s'envia una petició HTML, que produïx una recerca d'una URL a través de HTTP. La petició arriba al servidor, que executarà el programa necessari per obtindre la informació, en cas que siga necessari executar un programa, clar. S'elabora la resposta que serà enviada de nou a la pàgina web.

Quan s'utilitza *MapScript*, el servidor executa un *script*, que conté informació i funcions, igual que qualsevol altra pàgina programada en un altre llenguatge. En funció del llenguatge que s'utilitza per programar s'utilitzaran unes funcions o altres, que serviran per a buscar les dades *GIS*. Aquest mòdul de *MapScript* permet fer diverses operacions amb dades espacials, des d'accedir a aquestes dades des de la web o reprojectar les dades entre altres [13].

Bàsicament aquest és el procés que segueix PHP *MapScript*; a la imatge següent es mostra de manera esquemàtica el procés:



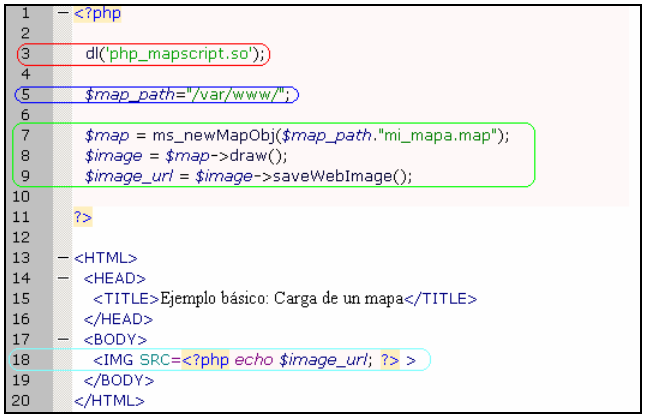
Imatge 10: Esquema procés PHP MapScript [13]

Per incloure codi *MapScript* a la pàgina web es crea l'arxiu "map" com fins ara s'ha vist; la diferència està que quan creem la plantilla HTML tindrà inclòs codi PHP *MapScript*. Abans de l'etiqueta HTML i entre les etiquetes de PHP carregarem la versió de PHP *MapScript* que tenim nosaltres, per a aquest exemple concretament s'ha utilitzat "*php_MapScript.so*". Aquesta línia no cal posar-la si l'arxiu "php.ini" (dins del pac d'instal·lació) està configurat per a carregar-lo.

A la línia 5 es crea una variable que contindrà la ruta on es troba el nostre arxiu, el que volem publicar. A la línia 7 es crea una variable, generalment nomenada "map" que contindrà un objecte, que serà el nostre arxiu "map". En aquest cas només és necessari donar-li el nom de l'arxiu, perquè li estem indicant que agafe aquest arxiu de la ruta continguda en "*map_path*".

Perquè es dibuixe el nostre mapa, es crea una variable (línia 8) en la qual s'indica que ens dibuixe la variable "map", que alhora conté l'arxiu i la ruta. Finalment s'ha de salvar eixa imatge perquè puga ser dibuixada (línia 9).

La manera de dibuixar el mapa és per mitjà de la variable que conté la imatge guardada, dins del *body* de la plantilla HTML, i dins de l'etiqueta "IMG".



```
<?php

  dl('php_mapscript.so');

  $map_path="/var/www/";

  $map = ms_newMapObj($map_path."mi_mapa.map");
  $image = $map->draw();
  $image_url = $image->saveWebImage();

 ?>

<HTML>
  <HEAD>
    <TITLE>Ejemplo básico: Carga de un mapa</TITLE>
  </HEAD>
  <BODY>
    <IMG SRC=<?php echo $image_url; ?> >
  </BODY>
</HTML>
```

Imatge 11: Càrrega d'un mapa amb PHP MapScript

Hi ha moltes eines, objectes i funcions de *MapScript* com per exemple les de zoom, llegenda o objectes de la imatge entre altres. A continuació s'anomenaran alguns exemples que s'han utilitzat per desenvolupar aquesta aplicació [12]:

De la classe *layer*, que fa referència a la capa, "*queryByPoint (pointObj point, int mode, double buffer)*" amb el qual es pot consultar un punt dins d'una cartografia que estiga georreferenciada (és refereix al posicionament amb el qual es defineix la localització d'un objecte espacial). Entre parèntesi li donem els paràmetres que li anem a passar.

De la classe de *layer* tenim el mètode "*Open ()*" que s'encarrega d'obrir una capa determinada.

De la classe de *layer* tenim el mètode "*getNumResult ()*" que pren el nombre de resultats d'una capa en l'última consulta realitzada sobre ella.

De la classe *legend*, que fa referència a la llegenda que extraurem de les dades introduïdes en l'arxiu *map*, per exemple, podem anomenar el mètode "*set (string property_name, new_value)*" que ens permet crear la llegenda.

De la classe *map*, que fa referència al mapa que estem creant, el mètode "*getsymbolbyname (string symbol_name)*", pren un objecte amb el símbol a què ens hem referit.

6.5. Base de dades

6.5.1. Què són les bases de dades

Una base de dades és, de manera rigorosa, un conjunt de dades emmagatzemades amb una estructura lògica. Per poder treballar amb les dades que emmagatzem a les bases de dades, han de tindre un ordre i estar relacionats entre sí. Aquestes característiques són les que donen coherència per poder treballar amb les dades d'una manera sistemàtica [7].

A continuació s'exposen una sèrie de conceptes relacionats amb les bases de dades que hem de conèixer-los:

Tupla: és una filera en una taula.

Atribut: és una columna en una taula.

Domini: és el conjunt de valors dels quals els atributs obtenen els seus valors.

Clau: és un atribut amb una característica de rellevància per a identificar la tupla.

Clau primària: és una clau amb valors únics, és a dir, no ocorren més d'una vegada en l'atribut.

Clau secundària: A diferència de la clau primària, no necessàriament identifica una fila d'una altra (poden existir repeticions), però serveix per a processar la informació en un ordre adequat, per a un procés en concret [22].

Cardinalitat: és el nombre de tuplas en una taula.

Grau: és el nombre d'atributs en una taula.

Relació: correspondència entre dos o més taules.

6.5.2. Programes per treballar amb bases de dades

De la lectura prèvia es pot deduir que els programes per treballar amb Bases de Dades (DBM) faciliten les funcions de:

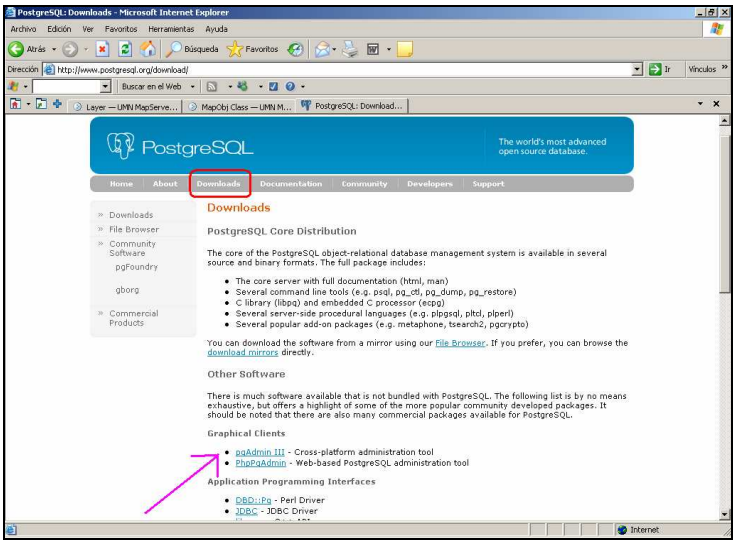
- Emmagatzemar físicament.
- Garantir consistència.
- Garantir integritat.
- Atomicitat transaccional.
- Manejar vistes a la informació.

Quan creem una base de dades s'ha de tindre en compte algunes consideracions que ens permeten fer un disseny de base de dades eficients. Algunes d'aquestes consideracions són:

- La velocitat d'accés.
- La grandària de la informació que contindrà la nostra base de dades. No és el mateix que ocupe 6 caràcters que 25.
- El tipus d'informació que inclourà cada taula de la base de dades, entre ells pot ser numèric, text, date, etcètera.
- La facilitat d'accés a la informació.
- La facilitat per a extraure la informació requerida.
- El comportament del manejador de bases de dades amb cada tipus d'informació.

6.6. PostgreSQL

Per desenvolupar aquest treball s'ha utilitzat com programa per treballar amb bases de dades *pgAdmin III*, que és un client que s'utilitza per gestionar les bases de dades, en la versió 8.0. Les últimes versions es poden trobar a la pàgina web oficial:



Imatge 12: Pàgina web oficial de *PostgreSQL*

En descarregar-lo es pot procedir a la instal·lació com s'explicarà en l'annex.

6.6.1. Llenguatge SQL

A continuació s'exposaran una sèrie de les sentències sql més bàsiques que s'han utilitzades per a aquesta aplicació. Entre elles, per exemple, tenim [20]:

CREATE: Ens dóna la possibilitat de crear nous objectes, com són taules, bases de dades, índex, operadors… Per a crear una base de dades, per exemple, s'escriurà "*CREATE DATABASE nomdb [WITH LOCATION = 'rutadb']*" o bé per a crear una taula nova escriurem "*CREATE TABLE nombre_clase (attr1 type1 [DEFAULT expression] [NOT NULL]… attrN [[CONSTRAINT name] CHECK condition1… conditionN]) [INHERITS (nombre1_clase… nombreN_clase)];*".

ALTER: Modifica objectes, canviant inclús les seues propietats. Entre aquestos tenim "*ALTER GROUP*" (afig usuaris a un grup o elimina usuaris d'eixe grup), "*ALTER TABLE*" (Modifica les propietats d'una taula) i "*ALTER USER*" (modifica la informació d'un usuari en concret), entre altres.

DROP: Esta sentència permet l'eliminació dels objectes que hem comentat amb anterioritat. Així doncs, "*DROP DB*" eliminaria una base de dades existents.

INSERT: Amb aquesta sentència podrem inserir registres en una taula de la nostra base de dades. Per exemple: "*INSERT INTO mi_tabla [(columna1, columna2…)] VALUES (expressió1, expressió2…)*".

SELECT: Selecciona tot, alguna columna o algun registre que li diguem d'una taula.

UPDATE: Aquesta sentència actualitza els valors que té una taula pels nous que li inserim.

6.6.2. Característiques de postgreSQL

Com tots els manipuladors de bases de dades, *PostgreSQL* implementa els tipus de dades definits per l'estàndard SQL3. A continuació es mostra els definits per l'estàndard SQL3:

Tipos de datos del estándar SQL3 en PostgreSQL		
Tipo en Postgres	Correspondiente en SQL3	Descripción
bool	boolean	valor lógico o booleano (true/false)
char(n)	character(n)	cadena de carácteres de tamaño fijo
date	date	fecha (sin hora)
float4/8	float(p)	número de punto flotante con precisión p
float8	real, double precision	número de punto flotante de doble precisión
int2	smallint	entero de dos bytes con signo
int4	int, integer	entero de cuatro bytes con signo
int4	decimal(p, s)	número exacto con $p <= 9, s = 0$
int4	numeric(p, s)	número exacto con $p == 9, s = 0$
money	decimal(9,2)	cantidad monetaria
time	time	hora en horas, minutos, segundos y centésimas
timespan	interval	intervalo de tiempo
timestamp	timestamp with time zone	fecha y hora con zonificación
varchar(n)	character varying(n)	cadena de carácteres de tamaño variable

Imatge 13: Tipus de dades del estàndard *SQL* en *PostgreSQL* [20]

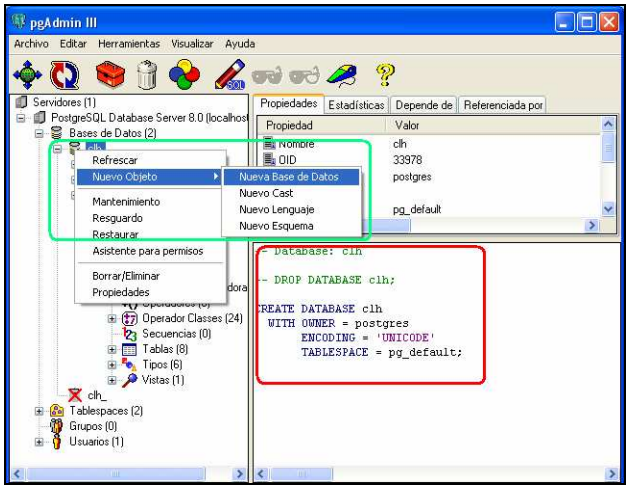
Els tipus d'extensions que es defineixen en *PostgreSQL* són les següents:

Tipos de datos extendidos en PostgreSQL	
Tipo	Descripción
box	caja rectangular en el plano
cidr	dirección de red o de *host* en IP versión 4
circle	círculo en el plano
inet	dirección de red o de *host* en IP versión 4
int8	entero de ocho bytes con signo
line	línea infinita en el plano
lseg	segmento de línea en el plano
path	trayectoria geométrica, abierta o cerrada, en el plano
point	punto geométrico en el plano
polygon	trayectoria geométrica cerrada en el plano
serial	identificador numerico único

Imatge 14: Tipus de dades estesos per *PostgreSQL* [20]

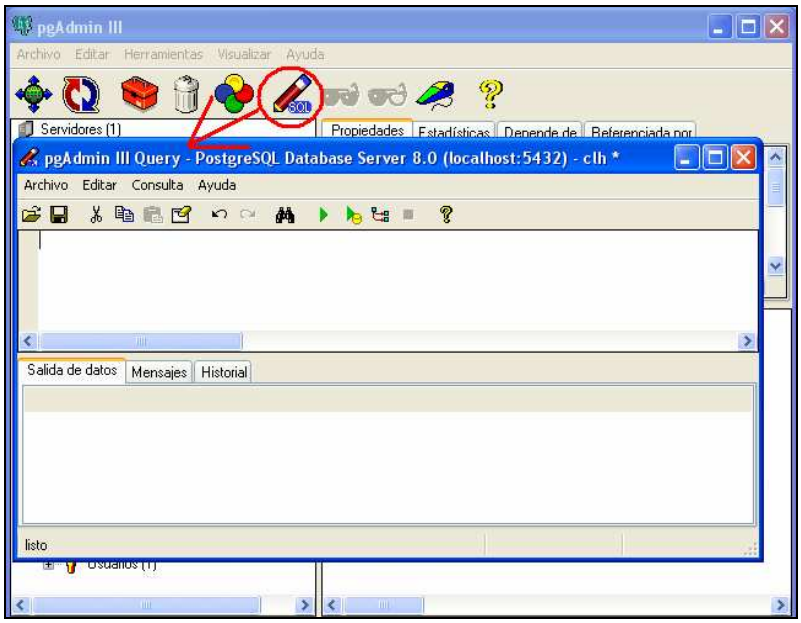
Anteriorment s'han vist vàries funcions de SQL, però quan s'està treballant amb *pgAdmin III* disposa d'un assisten que ajuda a elaborar el codi SQL.

Per exemple, si es vol crear una nova base de dades, amb el ratolí es pot crear (veure quadre verd) i al quadre roig es pot veure el codi SQL que s'ha anat creant.



Imatge 15: Creació d'una base de dades i sentències SQL

Com s'ha comentat abans, es pot fer també amb el codi SQL. Es polsa sobre el botó per escriure SQL i en la finestra que s'obri es pot escriure el codi.



Imatge 16: Finestra SQL

6.7. Javascript

Abans s'ha comentat que una pàgina web en només codi HTML és una pàgina web estàtica i per tant molt limitada per a fer una aplicació. Per dinamitzar la pàgina web s'ha utilitzat PHP, que és suficientment potent per resoldre els objectius plantejats en aquest projecte, però algunes funcions són més senzilles utilitzant "Javascript".

Javascript és un llenguatge de tipus "script" compacte, basat en objectes i guiat per esdeveniments dissenyat específicament per al desenvolupament d'aplicacions client - servidor dins de l'àmbit d'Internet [11]. Aquest llenguatge s'incrusta dins del codi HTML i les accions (demanar demanar dades, confirmacions, mostrar missatges, crear animacions, comprovar) que fa són al client.

6.7.1. Com incloure Javascript

Normalment inclourem el codi Javascript mitjançant una directiva "script" al document "HTML" amb el format següent:

<script language="Javascript"> [11]
</script>

També podem incloure al fitxer HTML un element extern amb la directiva "src", per exemple un fitxer que inclou codi Javascript, amb extensió ".js".

<script language="Javascript" src="archivo.js"> [11]
</script>

7. Solució adoptada

7.1. Preparació de la web

Per poder realitzar la web cal anar composant una sèrie d'elements (auxiliars o no) que responen a les necessitats que ens proposem, de manera que com més coses necessitem, més coses anem creant per poder apropar-nos a allò que seria el treball definitiu.

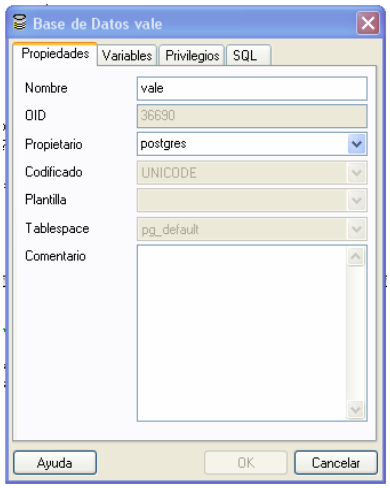
En aquest apartat es mostra com s'ha anat traçant la línia de treball, i com s'han anat solucionant els problemes que s'han anat trobant, perquè en alguns casos s'han hagut de crear passos previs a la solució final. També s'expliquen els motius pels quals s'ha fet d'aquesta manera i no d'una altra.

7.1.1. Creació de la base de dades

Creem una la base de dades, pot estar buida o no. En aquest cas hem creat una base de dades nova, per a poder emmagatzemar els focus contaminants que s'inserisquen durant l'ús de l'aplicació, també per a poder emmagatzemar les dades dels ciutadans afectats per aquests focus contaminants:

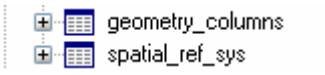
Imatge 17: Nova base de dades

La base de dades que s'ha creat presenta la següent configuració:



Imatge 18: Configuració de la base de dades

Quan creem la base de dades, en ser una base de dades espacial es creen un parell de taules amb informació espacial. Aquestes taules son la de "*geometry_columns*" i "*spatial_ref_sys*".



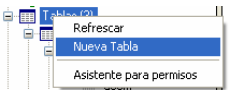
Imatge 19: Taules espacials

La taula "*geometry_columns*" conté informació de la geometria, li dóna funcionalitat espacial. La taula "*spatial_ref_sys*" conté la informació dels sistemes de referència espacials que suporta [1].

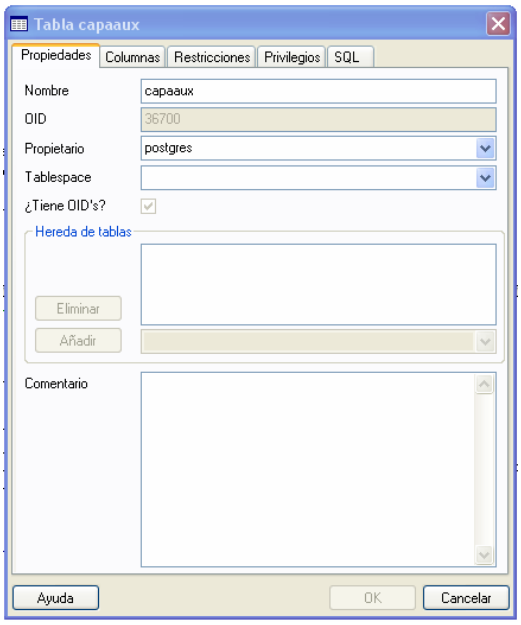
7.1.2. Capa auxiliar

Per poder fer consultes, l'objectiu de les quals és obtindre les coordenades al mapa, necessitem fer servir una capa auxiliar que ens donarà informació de les coordenades que volem consultar. La capa auxiliar no és més que un polígon de forma quadrada que abastar l'extensió del mapa a la pàgina web, per tant les coordenades de la capa auxiliar que utilitzarem ha de tenir la mateixa extensió que les coordenades que representem del mapa.

Ens disposem a crear la nova taula que contindrà la nostra capa. Creem una nova taula que tindrà la següent estructura:

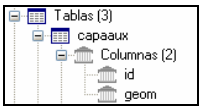


Imatge 20: Creació de una nova taula



Imatge 21: Estructura de la nova taula

La taula de la capa auxiliar tindrà dos camps: un camp que serà un identificador autoincrementable i únic (tipus de dada "*serial*") i un segon camp que serà espacial de tipus "*POLYGON*" i contindrà les coordenades de la capa auxiliar.



Imatge 22: Camps de la capa auxiliar

La sentència sql que defineix el camp "geom" és la següent:

```
select addgeometrycolumn ('capaaux', 'geom', 23030, 'POLYGON', 2);
```

Imatge 23: Sentència SQL per definir el camp geomètric

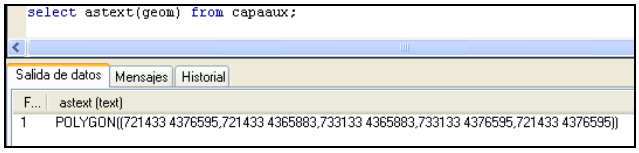
És important remarcar que ens hem de fixar en posar correctament el sistema de referència, en aquest cas el 23030.

Una vegada creat el camp geomètric podem inserir el polígon auxiliar, per poder afinar en les coordenades utilitzem *gvSIG* i prenem les coordenades que ens interessen i inserim el polígon a la base de dades amb la següent sentència:

```
insert into capaaux (geom) values
 (geometryfromtext ('POLYGON((721433 4376595, 721433 4365883, 733133 4365883, 733133 4376595, 721433 4376595))', 23030));
```

Imatge 24: Sentència de la inserció del polígon

Per comprovar si ha sigut inserit correctament, consultem la base de dades. Efectivament el polígon està correctament a la base de dades:



Imatge 25: Polígon inserit

7.1.3. Àrees d'influència

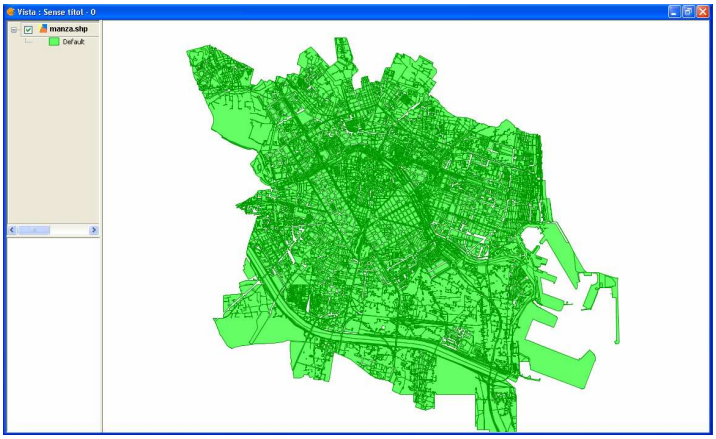
Les àrees d'influència (d'ara endavant també es diran buffers) han de crear-se en una capa de tipus *multipolygon*[1].

A l'arxiu *map* (arxiu que sosté la informació cartogràfica) no podem carregar una capa que no existeix o bé que està buida (que no té registres). Per poder visualitzar la capa de *buffers* (estiguen o no creats) hem de dibuixar un minúscul polígon a la base de dades i així poder visualitzar la capa sempre; encara que parega buida, existeix un polígon auxiliar.

7.1.4. Inclusió de les capes "shp" a la base de dades

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

A la nostra base de dades cal incloure dades cartogràfiques que originalment es troben en format *shape* (capes d'illes cartogràfiques i carrers):



Imatge 26: Cartografia visualitzada amb *gvSIG*

El procés de transformació és relativament senzill de fer per consola en Linux si el dividim en els següents apartats:

- Creem per consola una base de dades espacial (que consisteix en una base de dades que emmagatzema les coordenades dels punts inserits) buida amb la instrucció corresponent. Cal tindre en compte que per fer una base de dades espacial hem d'executar les llibreries que ens

permeten fer aquest tipus de base de dades, en el nostre cas estan ubicades a */home/usr/share/PostGIS*.

- Createdb –U postgres pr: amb aquesta ordre vam crear la base de dades, pr es el nom de la base de dades, que ha de tenir intenament.

- Createlang –U postgres plpgsql pr: amb aquesta ordre definim el llenguatge de programació per a una base de dades PostgreSQL.

- Psql –U postgres –f lwPostGIS.sql –d pr: carreguem *PostGIS* a la base de dades. Amb aquesta ordre afegim suport d'objectes geogràfics a la base de dades.

- Psql –U portares –f spatial_ref_sys.sql –d pru: car carregar aquest arxiu per poder treballar amb els sistemes de referència.

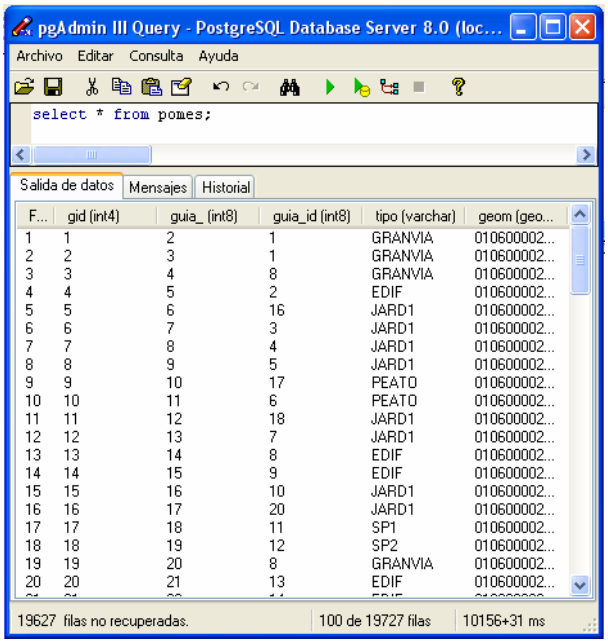
- Comprovem que la base de dades ha sigut creada satisfactòriament.

- Ara podem fer la transformació de *shape* a format *postgres*. Es passarà de format a postgres l'arxiu "poma", nomenarem al camp geomètric "geom" que tindrà el sistema de referència amb què estem treballant, el 23030, i l'arxiu es crearà al directori */home/alumno/*. Cal tindre en compte que aquesta funció no llig el prj, per això li donem el sistema de referència, si no li ho hauriem d'haver editat després. Per aquest procediment s'utilitza la següent sentència:
  - shp2pgsql manza.shp pomes -w -g geom -s 23030 > /home/alumno/pomes.sql

- Ara s'insereix a la base de dades l'arxiu *sql* creat amb la sentència:

  - psql -U postgres -f /home/alumno/pomes.sql -d bbddnom.

- Cal comprovar que tot ha anat bé fent un *select* de la nova capa creada.

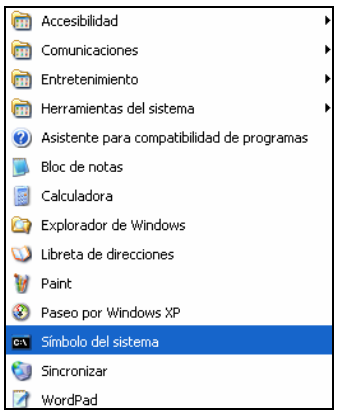


Imatge 27: Selecció de la capa "pomes"

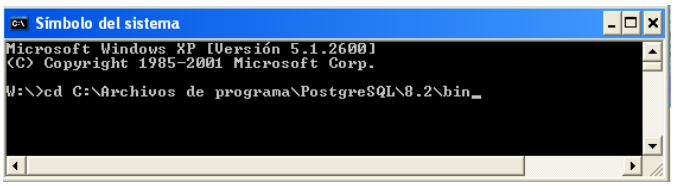
- Si tot és correcte podem fer una còpia de la taula per poder importar-la a la nostra base de dades.

Però si no tenim el sistema operatiu Linux també és senzill fer-lo amb Windows. Dividim el procés en els següents passos:

Obrim una consola de Windows i ens ubiquem al directori on està instal·lat postgres.

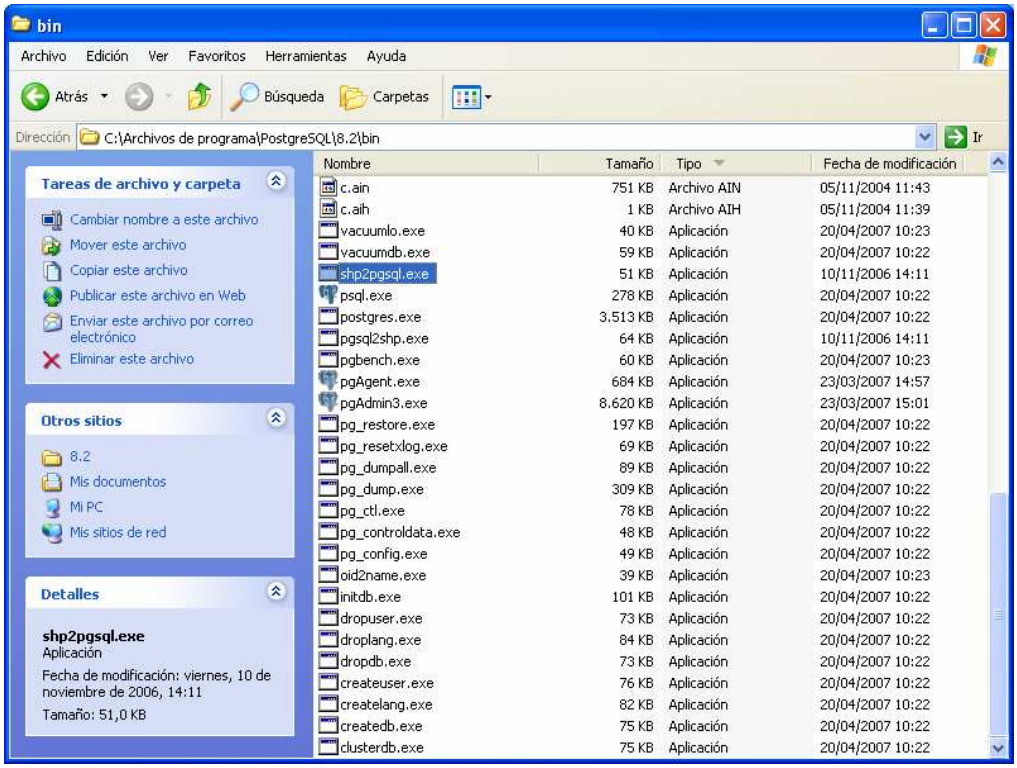


Imatge 28: Per obrir el símbol del sistema



Imatge 29: Canvi de directori de treball

Al directori on està instal·lat postgres tenim una col·lecció de funcions que podem executar. En aquest cas hem d'executar la funció shp2pgsql, que com es pot comprovar està ubicada al directori.



Imatge 30: Funcions de *PostgreSQL*

Aleshores ens queda executar la sentència sql que ens permet passar d'un format a un altre. Cal saber que la capa que es vol passar a format sql ha d'estar ubicada al mateix directori on està la funció que anem a executar. La funció que hem d'executar és la mateixa sentència que la que hem utilitzat a Linux.

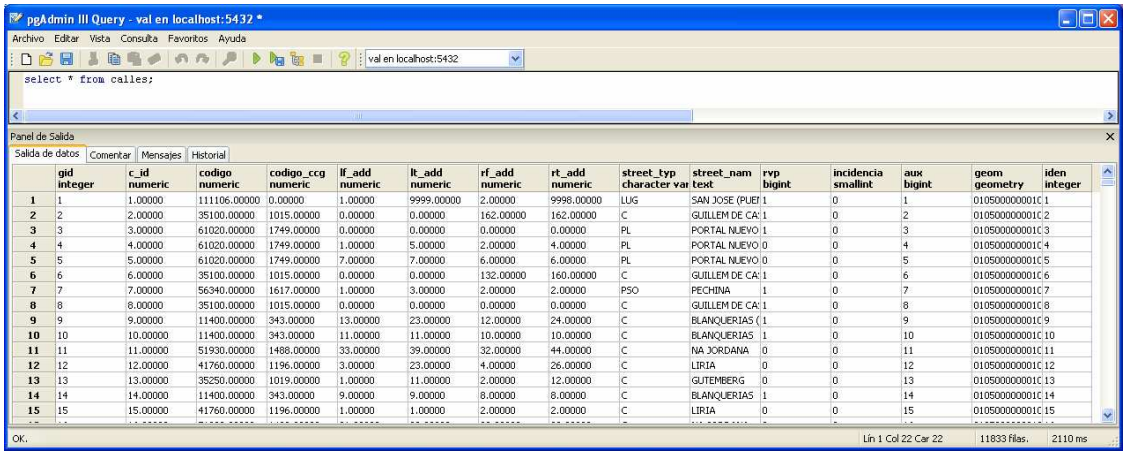
- shp2pgsql c.shp carrers -w -g geom -s 23030 > C:\Archivos de programa\PostgreSQL\8.2\bin\carrers.sql

Quan tenim l'arxiu sql ja podem inserir-lo a la base de dades. La sentència és la mateixa que hem fet servir a Linux.

Ara s'insereix a la base de dades l'arxiu *sql* creat amb la sentència:

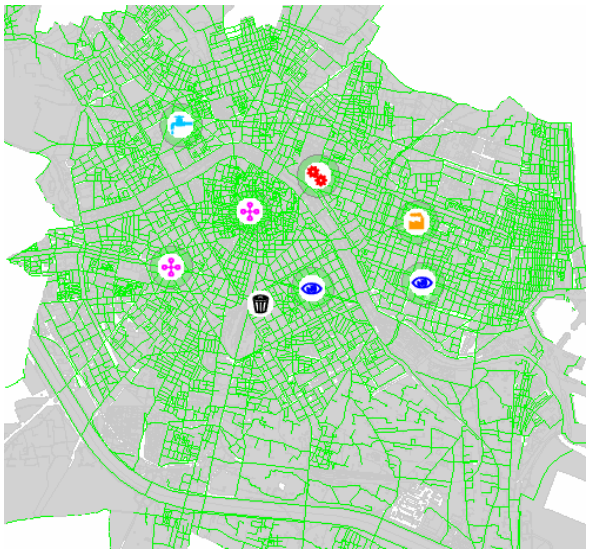
- psql -U postgres -f C:\Archivos de programa\PostgreSQL\8.2\bin \carrers.sql -d val.

Podem comprovar que s'ha inserit correctament si fem una selecció de la nova capa:



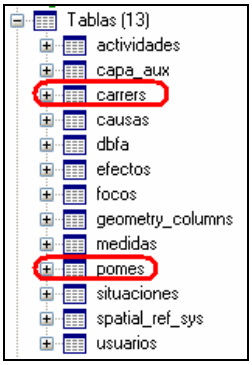
Imatge 31: Resultat de la consulta de la capa "carrers"

També es pot visualitzar la capa, de manera que el resultat (pintat en línies de color verd) es veu que és complet.



Imatge 32: Càrrega de les capes

De manera que la possible base de dades finalment queda de la següent manera:



Imatge 33: Mostra de la correcta inserció de les capes

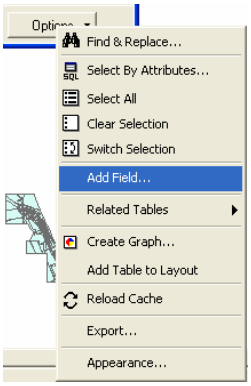
7.1.5. Capa de dades. Vinculació de dues capes

En aquest apartat s'explica com s'han vinculat les dues capes, tant la capa que ens creem nosaltres per poder completar la informació, com la capa d'illes cartogràfiques.

7.1.5.1. Preparació de la capa d'illes de cases

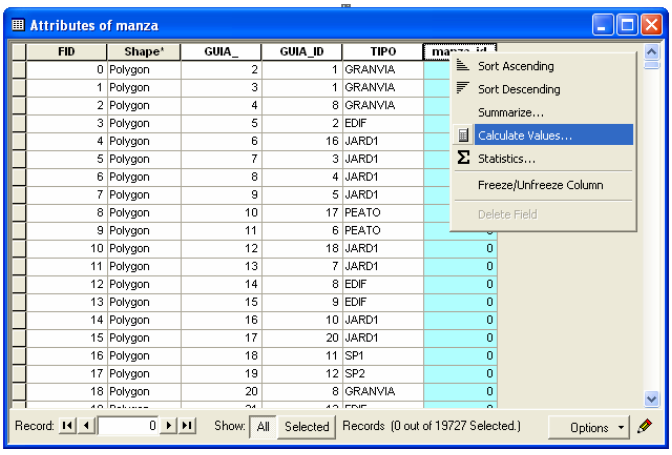
Per poder vincular dues capes hem de tenir un camp en comú. Una de les capes és la que conté les dades de les illes de cases i originalment no té cap camp que tinga informació de cada polígon de forma única. Per tant, primerament cal crear un camp de tipus numèric que identifique de manera única cada polígon. La ferramenta que s'ha utilitzat per resoldre aquest problema és *ArcGIS* perquè té una senzilla funció que en uns segons enumera cada registre de manera única.

Ara solament cal crear-nos un camp nou:



Imatge 34: Creació d'un nou camp

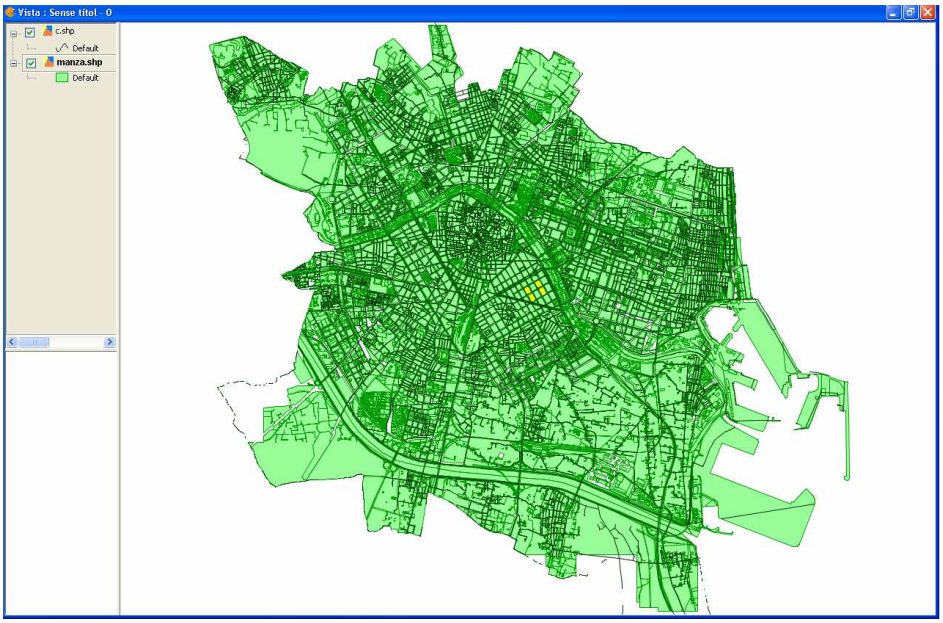
Com el camp el definim numèric, s'ompli del valor "0", però amb la calculadora d'*ArcGIS* es pot tornar a calcular els seus valors.



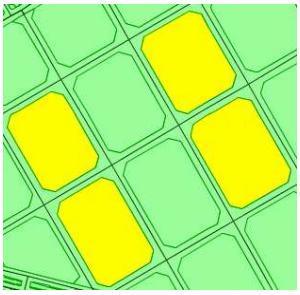
Imatge 35: Utilització de la calculadora per recalcular els valors

7.1.5.2. Capa de dades complementària

Per poder fer una consulta òptima per a l'exemple, s'han elegit quatre polígons de la capa d'illes de cases, que des d'una vista general estan ubicats al centre de la ciutat de València:



Imatge 36: Selecció dels polígons. Vista general



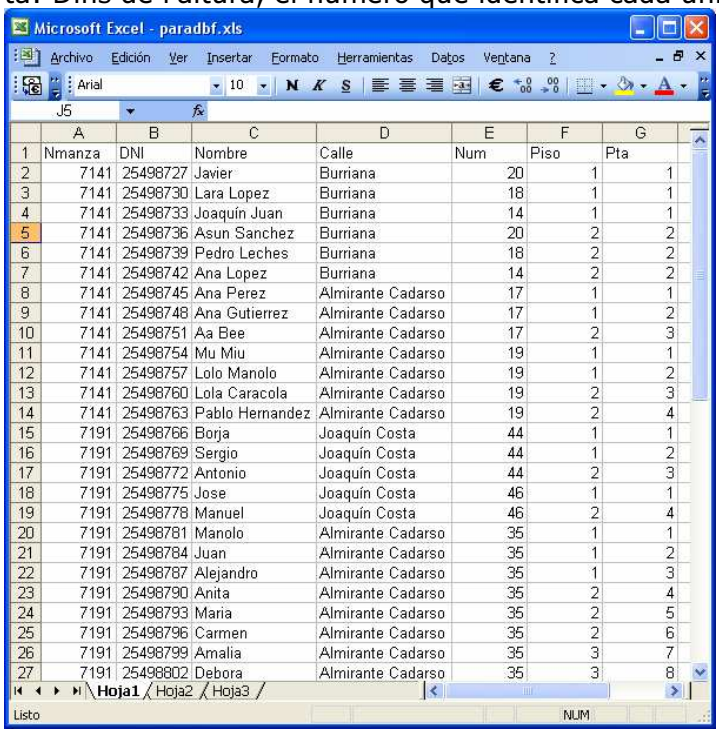
Imatge 37: Selecció dels polígons. Vista propera

Per poder afegir més informació a la nostra base de dades anem a crear una taula amb una sèrie de dades que vincularem a la capa d'illes de cases. S'utilitza un full de càlcul que tindrà els camps:

- Número d'illa de cases: o el que és el mateix, el número únic que identifica cada polígon.
- DNI: És un número triat a l'atzar, identificació única per a cada suposat habitant.
- Nom: També triat a l'atzar, identifica a cada suposat habitant.
- Carrer: Nom real, tret de fer consultes als carrers que envolten els polígons amb *gvSIG*.
- Núm: És el número de portal al carrer que estem consultant.

- Pis: Equivalent a l'altura en la qual habiten les persones.

- Porta: Dins de l'altura, el número que identifica cada unitat de vivenda.



Microsoft Excel - paradbf.xls

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Arial 10

J5

	A	B	C	D	E	F	G
1	Nmanza	DNI	Nombre	Calle	Num	Piso	Pta
2	7141	25498727	Javier	Burriana	20	1	1
3	7141	25498730	Lara Lopez	Burriana	18	1	1
4	7141	25498733	Joaquín Juan	Burriana	14	1	1
5	7141	25498736	Asun Sanchez	Burriana	20	2	2
6	7141	25498739	Pedro Leches	Burriana	18	2	2
7	7141	25498742	Ana Lopez	Burriana	14	2	2
8	7141	25498745	Ana Perez	Almirante Cadarso	17	1	1
9	7141	25498748	Ana Gutierrez	Almirante Cadarso	17	1	2
10	7141	25498751	Aa Bee	Almirante Cadarso	17	2	3
11	7141	25498754	Mu Miu	Almirante Cadarso	19	1	1
12	7141	25498757	Lolo Manolo	Almirante Cadarso	19	1	2
13	7141	25498760	Lola Caracola	Almirante Cadarso	19	2	3
14	7141	25498763	Pablo Hernandez	Almirante Cadarso	19	2	4
15	7191	25498766	Borja	Joaquín Costa	44	1	1
16	7191	25498769	Sergio	Joaquín Costa	44	1	2
17	7191	25498772	Antonio	Joaquín Costa	44	2	3
18	7191	25498775	Jose	Joaquín Costa	46	1	1
19	7191	25498778	Manuel	Joaquín Costa	46	2	4
20	7191	25498781	Manolo	Almirante Cadarso	35	1	1
21	7191	25498784	Juan	Almirante Cadarso	35	1	2
22	7191	25498787	Alejandro	Almirante Cadarso	35	1	3
23	7191	25498790	Anita	Almirante Cadarso	35	2	4
24	7191	25498793	Maria	Almirante Cadarso	35	2	5
25	7191	25498796	Carmen	Almirante Cadarso	35	2	6
26	7191	25498799	Amalia	Almirante Cadarso	35	3	7
27	7191	25498802	Debora	Almirante Cadarso	35	3	8

Hoja1 Hoja2 Hoja3

Listo NUM

Imatge 38: Estètica de la taula

Per poder integrar aquesta taula a la base de dades cal salvar-la com a tipus "dbf", i serà quan podrem treballar amb ella. Al sistema operatiu Linux ens creem una nova base de dades no espacial i buida, que li nomenarem "dbfpru".

```
alumno@linux-9z5r:~> createdb -U postgres dbfpru
Password:
alumno@linux-9z5r:~> psql -U postgres -l
Password for user postgres:
          List of databases
     Name      |  Owner   | Encoding
---------------+----------+----------
 bbddunicidad  | postgres | UTF8
 dbfpru        | postgres | UTF8
 ejercicio61   | postgres | UTF8
 postgres      | postgres | UTF8
 restau        | postgres | UTF8
 sp2           | postgres | UTF8
 template0     | postgres | UTF8
 template1     | postgres | UTF8
 test1         | postgres | UTF8
 testpgrouting | postgres | UTF8
 valpru2       | postgres | UTF8
(11 rows)
```

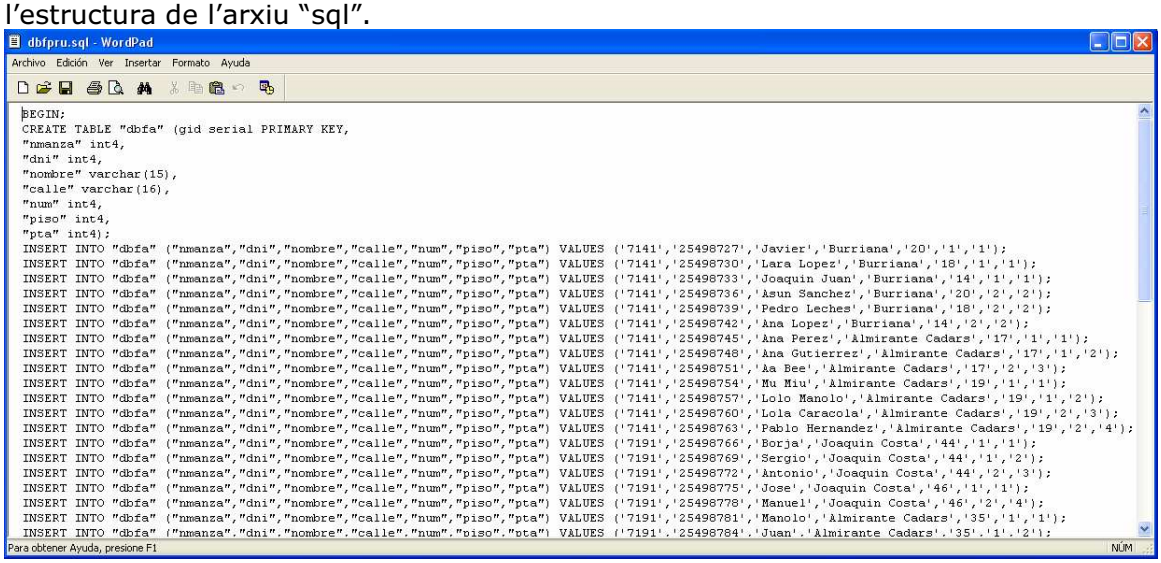
Imatge 39: Creació i comprovació de la base de dades nova

Una vegada que tenim la base de dades creada podem restaurar-la, per a poder utilitzar-la en windows. Per poder restaurar-la s'utilitza la funció "shp2pgsql" i solament haurem d'importar les dades. Utilitzem les següents instruccions:

- Per obtindre el fitxer "sql" que després importarem utilitzem la sentència: shp2pgsql paradbf4.dbf dbf -w > /home/alumno/dbf4pru.sql

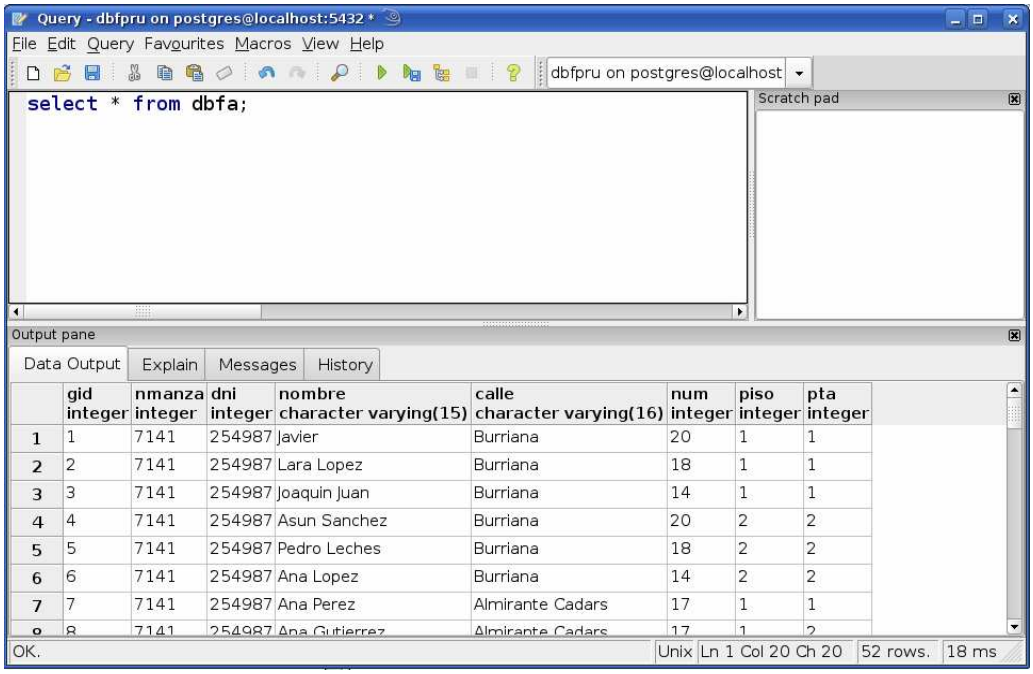
- Aquesta altra ens inclourà a la base de dades que havíem creat la taula "sql" que ens hem creat abans: psql -U postgres -f /home/alumno/dbf4pru.sql -d dbfpru

És molt important saber que aquest procés no suporta accents ni caràcters estranys, i per tant, cal filtrar. A continuació es mostra una imatge en què es veu l'estructura de l'arxiu "sql".



Imatge 40: Arxiu "dbfpru.sql"

Efectivament, quan executem la segona sentència comentada abans estem executant el fitxer "sql", replet d'instruccions que s'afegeixen a la base de dades que ens havíem creat. Per tant, tan sols queda veure gràficament que realment tots els registres s'han inserit adequadament; si fem una mostra de tots els registres de la taula, mitjançant un *select*, tenim el següent resultat.



Imatge 41: Mostra dels registres de la taula

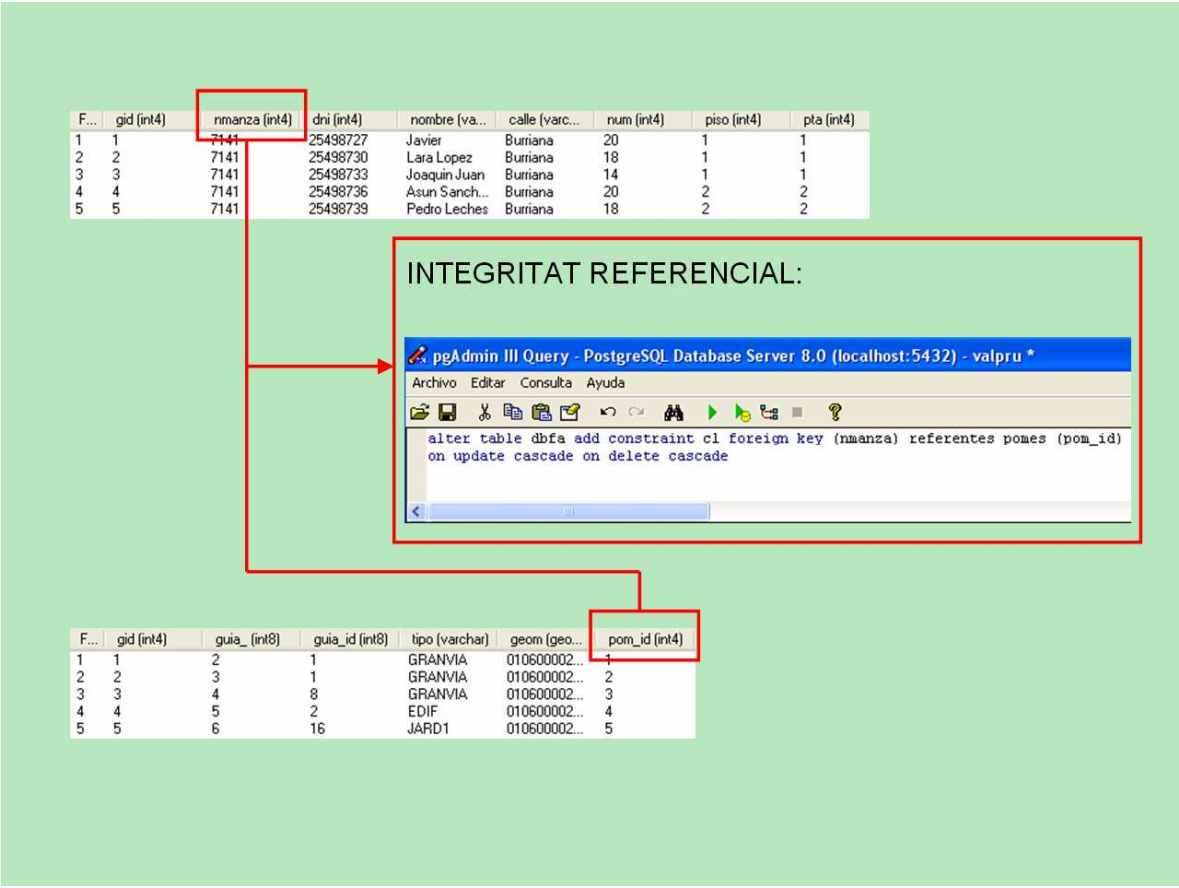
Efectivament, les dades s'han inserit bé, de manera que ja contem amb la informació dels polígons i la informació que s'afegirà.

7.1.5.3. Vinculació de les dues capes

Per poder vincular les dues capes que ens hem creat cal utilitzar la integritat referencial, que ens permet actualitzar els registres d'un camp d'una taula "A" a partir de la inclusió o esborrat dels elements d'una taula "B".

D'aquesta manera, la capa d'illes de cases té la informació de tots els polígons que conformen la ciutat de València. Cal vincular la informació de la capa que ens hem creat (la capa que hem tret a partir de la "dbf") del ciutadans, amb aquesta capa d'illes de cases per tal que si en una capa (modificable per l'administrador de la pàgina) s'elimina algun registre, que la taula s'actualitze automàticament, sense la mà de l'administrador.

En el següent esquema es mostra la vinculació dels dos camps de les dos taules a partir d'una sentència "sql" (integritat referencial). Els dos camps que es vinculen defineixen de manera única cada registre.



Imatge 42: Vinculació de les dos taules

Amb aquesta sentència "sql" queden vinculades les dos taules per dos dels camps que la formen. Per poder executar la sentència correctament, cal que els dos camps tinguen la restricció de clau primària (*primary key*); aquesta restricció exigeix que el valor d'una columna o la combinació de varies columnes no espiguen duplicats en el conjunt de files de la taula, i els valors del camp no siguen nuls [1]. La restricció de clau primària equival a les restriccions d'unicitat (*unique*) i valor no nul (*not null*).

7.1.6. Consulta

Amb la finalitat de donar una aplicació més al projecte, s'ha fet un estudi per poder aportar una solució d'una consulta complexa. En aquest cas proposem buscar

el nombre d'afectats, i les seues identitats, d'un determinat focus contaminant. Recordem que aquesta funcionalitat s'oferirà com una possible actualització de l'aplicació

Per poder solucionar aquest problema cal dissenyar una sentència sql complexa que ens relacione la capa d'illes de cases amb la capa de dades personals dels habitants de la ciutat. En aquest cas els habitants que tenim són pocs, perquè s'ha utilitzat una capa creada de manera auxiliar per fer les proves; però en general són moltes persones, i per tant registres, els que han de ser consultats. Una manera d'agilitzar el procés és fer servir els índex.

Els índex incrementen la velocitat de recerca quan les nostres taules tenen moltes files [1]. En aquest cas no es disposen de taules tan grans, però s'ha volgut fer ús dels índex per mostrar la línia de treball que es seguiria en cas que tinguérem una taula real amb milers de registres. És comú indexar aquells camps pels què anem a consultar.

Per poder aplicar-los procedirem de la següent manera:

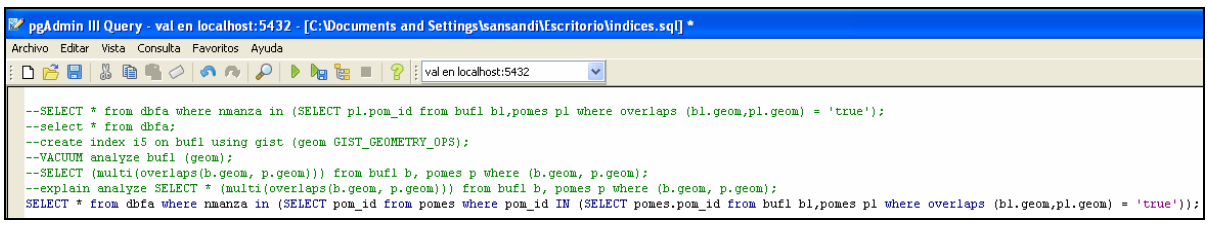
- Crearem un índex al camp geom a la taula buf1:
  - *create index i1 on buf1 usign gist (geom GIST_GEOMETRY_OPS)*
- Farem una actualització de les estadístiques perquè la taula buf1 s'adone que s'ha creat un índex, i1.
  - *Vacuum analyze buf1 (geom)*

Una volta fet açò, ja podem realitzar la consulta amb els camps implicats utilitzant els índex creats. Per saber quina consulta hem d'escriure ens hem de plantejar quin és el producte que volem obtindre. El que volem és un llistat que ens informe del nom i dni dels habitants que han estat afectats per l'àrea d'influència d'un focus contaminant.

De manera que busquem el predicat que necessitem. Podem pensar que podem aplicar un predicat *within* o *crosses*, però les descripcions d'aquestos predicats ens deixen fora alguns casos que cal contemplar també. Ens quedem definitivament amb el predicat *d'overlap*, que es descriu com "*les geometries que comparteixen una part dels seus punts i la intersecció té la mateixa dimensió que les geometries*" [1], d'aquesta manera podem incloure les illes de cases que no estiguen totalment afectades, però sí part, perquè en aquestos casos es desallotjaria tota l'illa de cases.

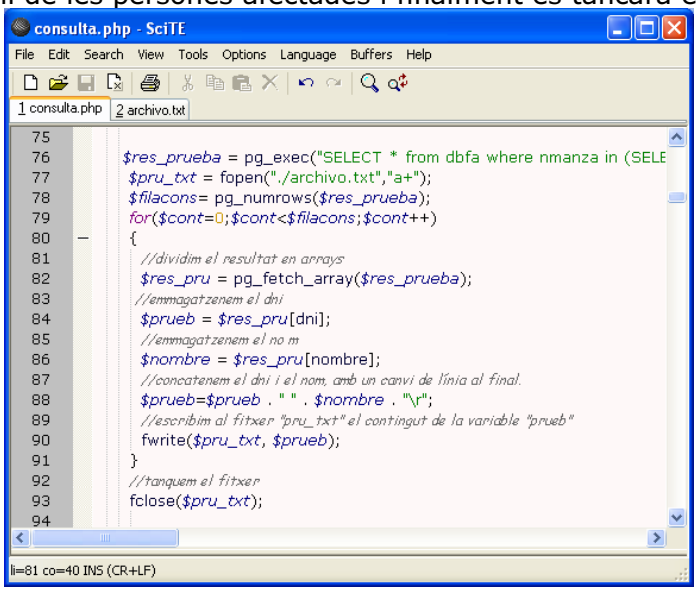
De manera que la consulta, finalment quedarà (utilitzant els índex):

- *explain analyze select * (multi(overlaps(b.geom, p.geom))) from buf1 b, pomes p where (b.geom, p.geom);*



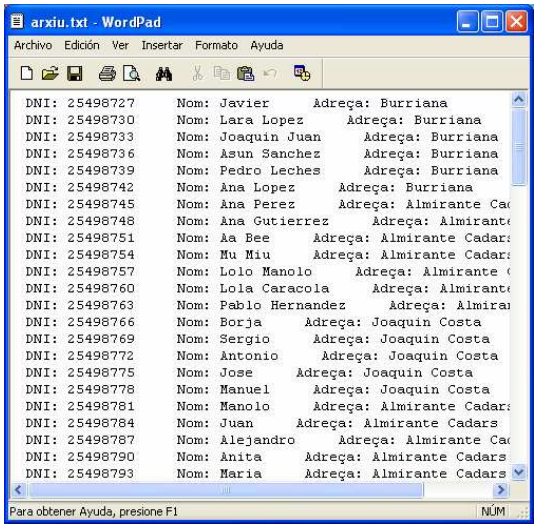
Imatge 43: Creació dels índex

Com el que ens interessa és extraure un fitxer que es puga imprimir cal crear a la web la programació corresponent, de manera que s'obrirà un fitxer, s'escriuran els noms i els dni de les persones afectades i finalment es tancarà el fitxer.



Imatge 44: Programació

Quan fem la consulta, es crea un fitxer (creat a propòsit per a aquesta comesa) en què recollim el DNI, el nom i l'adreça, aquest fitxer pot ser tractat com a dades d'entrada per una altra aplicació mitjançant el seu enviament a través de la xarxa o simplement ser impresa com un pdf, que es mostra d'aquesta manera:



Imatge 45: Resultat al fitxer de text

7.2. Desenvolupament de l'aplicació

Una volta tenim preparats tots els components per separat, s'agruparan de manera que s'anirà formant el producte final. També s'explicarà la manera en què s'ha dut a terme la programació, al menys les línies més significatives.

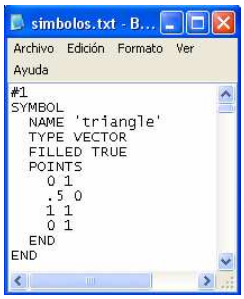
7.2.1. Elements necessaris

Per poder fer una aplicació adequada cal combinar arxius de diferent origen, biblioteques de recursos que milloren la maquetació, símbols… Per tant serà necessari comentar els diferents tipus de recursos que s'han utilitzat per fer l'aplicació en els següents apartats. No cal oblidar que també s'han d'ajuntar tots els elements que s'han estat fent per poder desenvolupar l'aplicació.

7.2.2. Biblioteca de simbologia

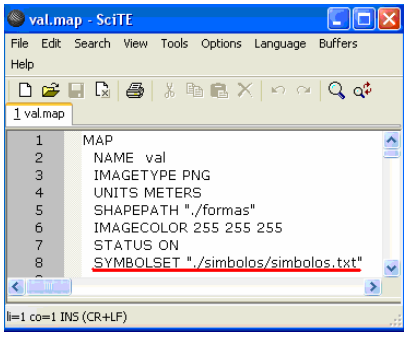
MapServer té uns símbols predefinits, (que seran els diferents focus contaminants definits per l'empresa de neteja) però aquestos són bàsics i n'hi ha poca variació. No obstant això, *MapServer* permet la visualització de símbols creats a mà, amb una configuració determinada de l'arxiu que conté les línies que defineixen els símbols.

Per poder crear símbols a mà cal definir el nom, el tipus de símbol que anem a dissenyar i les seues propietats. Finalment, cal incloure coordenades relatives on es defineix la forma i tamany del símbol. Per exemple, si volem dibuixar un triangle a la nostra pàgina web, la configuració de l'arxiu serà el següent:



Imatge 46: Definició d'un símbol

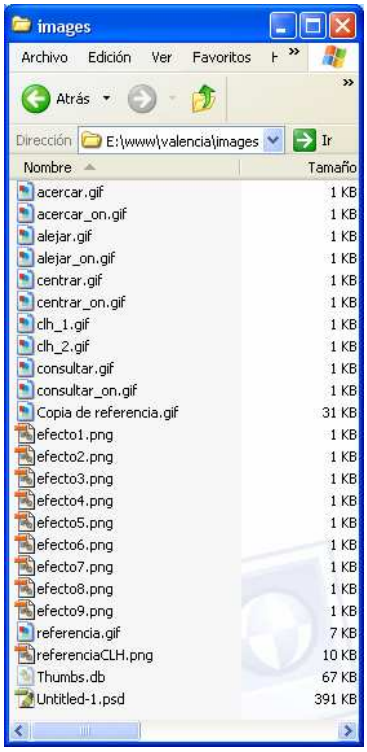
L'arxiu que es crea té l'extensió "*.txt", és un document simple de text. Cal parar atenció a la ubicació d'aquest arxiu. Ha d'estar a l'altura dels altres fitxers, i per tant al directori web. Aquesta ubicació és cridada des de l'arxiu *map*, per tant no és estrictament obligatori que siga aquesta ubicació, pot variar, però és més ordenat tindre-ho tot agrupat. A la següent imatge es mostra com es crida a aquest fitxer des de l'arxiu *map*, que es fa des de la propietat "*SYMBOLSET*":



Imatge 47: Crida a l'arxiu de símbol

7.2.3. Biblioteca d'imatges

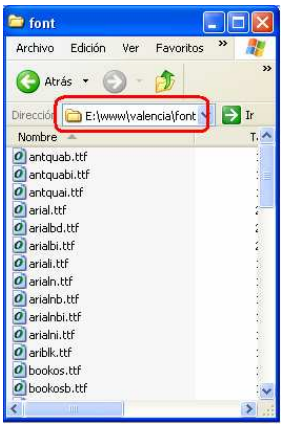
Les imatges que s'utilitzen a la pàgina web també són al directori web. Són imatges creades o retocades que s'utilitzen al llarg del desenvolupament de la pàgina web i que per tant, per tindre-les ordenades estan ubicades a un directori juntes, dins del directori "www". El directori conté la col·lecció d'imatges:



Imatge 48: Imatges i ubicació

7.2.4. Biblioteca de fonts

Per crear els servidors de cartografia i donar format a les lletres que s'estan utilitzant disposem d'una col·lecció de fonts que ja estan definides. Aquesta col·lecció està ubicada al directori web, a l'altura d'on es troben la resta d'arxius, cartografia, etc. La biblioteca de fonts té el següent contingut:



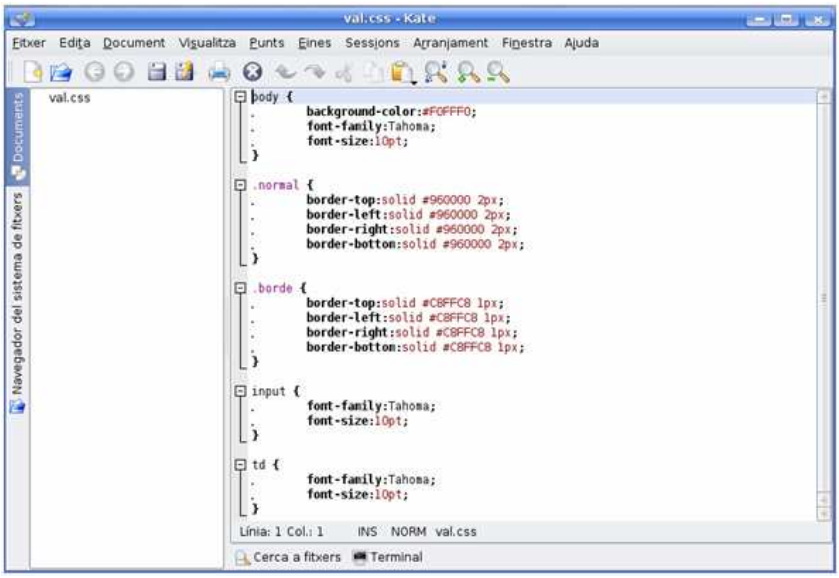
Imatge 49: Ubicació i contigut de font.

Cal fixar-se en que cada arxiu correspon a un tipus de lletra, d'ací el nom dels arxius. També cal observar la ubicació de la biblioteca, que és al directori web "www".

7.2.5. Plantilla de disseny

Quan estem dissenyant una pàgina web hem d'especificar un disseny mínim perquè tots els arxius mostren la mateixa aparença i es mantinga una visualització coherent. Per poder estalviar-nos aquestes línies, que s'haurien d'escriure una vegada i una altra, podem fer una crida des de cada arxiu que fem per construir la pàgina web a l'arxiu de disseny.

Aquest arxiu és un fitxer amb extensió ".css" compost per blocs que defineixen elements que es mostraran en les diferents pàgines que es visualitzaran. Per exemple podem definir el format del cos, el grossor d'una columna, el tipus de lletra que s'utilitzarà... es poden definir a la plantilla els elements de disseny que es consideren molt repetitius per posar-los als fitxers de les pàgines.



Imatge 50: Estructura de l'arxiu ".css"

7.2.6. Arxiu map

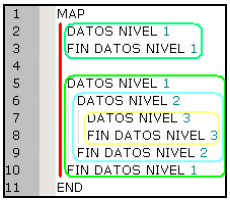
L'arxiu *map* és aquell que conté tota la informació cartogràfica que va a ser visualitzada al servidor de cartografia. En els següents apartats s'explica de manera més especificada cada apartat o cada bloc d'aquest arxiu.

Imatge 51: Arxiu "map"

7.2.6.1. Estructura de l'arxiu map

L'arxiu *map* és configurat de manera jeràrquica en cadascuna de les diferents propietats, per aquest motiu és necessari identar correctament cada línia al lloc que li correspon.

L'arxiu *map* s'edita i es treballa amb l'editor de textos "scite", tot el contingut anirà dins de les etiquetes "*MAP*" i "*END*":



Imatge 52: Arxiu jeràrquic

A l'arxiu *map* es defineixen primerament els paràmetres generals del *map*, com poden ser:

- *NAME*: Es refereix al nom del *map*, pel qual identificarem l'objecte, quan vulguem fer referència a ell des d'algun altre fitxer (en el nostre cas, des de l'arxiu "val.php").

- *IMAGETYPE*: El resultat que ens torna el servidor és una imatge, però hem d'especificar el format d'eixida.

- *UNITS*: Cal indicar les unitats en què s'estableixen les coordenades del mapa. Les unitats són necessàries per poder obtindre l'escala.

- *SHAPEPATH*: Aquesta ruta és on emmagatzemem les capes, en cas que carreguem alguna capa shape. En general és més òptim treballar amb rutes relatives.

- *IMAGECOLOR*: Es refereix al color de fons que s'indica en valors RGB, tres valors entre 0 i 255 que indiquen els valors del roig, verd i blau, respectivament.

- *STATUS*: Es refereix a si el mapa està actiu o no, es a dir, si serà visible o no.

- *SIZE*: Aquest paràmetre es refereix al tamany en píxels de la cartografia. Aquest paràmetre estarà relacionat amb la velocitat amb què es generarà la imatge. Aleshores cal no excedir-se, ni per dalt ni per sota.

- *EXTENT*: És la cartografia que va a ser representada al mapa, cal augmentar un poc les coordenades perquè no estiga pegada als límits del mapa.

- *PROJECTION*: Aquest objecte és complet i per tant s'ha d'especificar dins de les etiquetes "*PROJECTION*" i "*END*" i defineix la projecció que tenen les nostres dades cartogràfiques. Cal utilitzar el codi "EPSG" o especificar un per un els paràmetres necessaris. En el nostre cas, utilitzarem el codi "EPSG".

```
16    PROJECTION
17      "init=epsg:23030"
18    END
```

Imatge 53: Objecte "projection"

La segon part del fitxer *map* està constituïda per les capes, que poden tindre molts orígens: shp, wms, *PostGIS*, entre altres. S'explica en els següents apartats els que s'han utilitzat en aquest projecte.

7.2.6.2. Càrrega d'una capa "shp"

Per poder carregar una capa en format shape cal definir les propietats de la capa dins de les etiquetes "*LAYER*" i "*END*". Es definirà el nom, que correspon al nom que li volem donar com a etiqueta, a més a més, cal indicar els diferents paràmetres.

La definició de la capa pot ser tan complicada o senzilla com vulguem nosaltres. Podem carregar sols la capa o donar-li un estil determinat amb tamany (definit per un nombre sencer), color (indicat amb el sistema RGB), simbologia especial... A més a més, li donem un sistema de referència que en aquest cas es ED50 per a la zona 30, estem parlant de coordenades UTM.

Els paràmetres que s'utilitzen són els següents [2]:

- *LAYER*: Cadascuna de les capes que es vulga carregar, ha d'estar dins d'aquest objecte, iniciat i acabat per les etiquetes "*LAYER*" i "*END*". Cada capa té uns paràmetres mínims que la defineixen; per exemple, per carregar una capa en format "shp":

    - *NAME*: Etiqueta identificativa de la capa.

- *DATA*: Nom del fitxer "shape" que conté la informació que volem mostrar.

- *TYPE*: S'indica el tipus d'informació que es vol mostrar, pot ser:

  - *POINT*: Cartografia puntual.
  - *POLYGON*: Cartografia poligonal.
  - *LINE*: Cartografia lineal.
  - *RÀSTER*: Imatge.
  - *ANNOTATION*: Anotació.
  - *QUERY*: Consulta

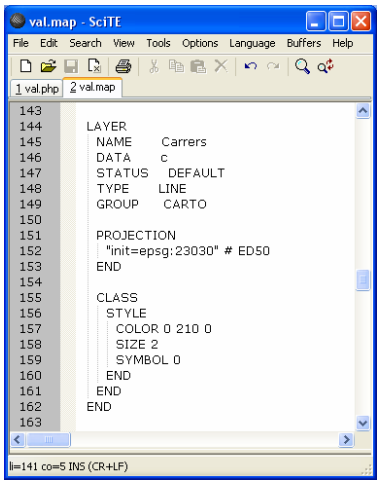
- *STATUS*: Correspon a l'estat de visualització de la capa (no del mapa). És útil per activar o desactivar capes.

- *GROUP*: És un paràmetre que ens permet agrupar vàries capes, y per tant tractar-les com a bloc.

- *PROJECTION*: És el paràmetre que ens indica la projecció en què estan emmagatzemades les dades cartogràfiques de la capa.

- *CLASS*: Aquest paràmetre defineix l'estètica que tindrà la capa.

En la següent imatge es mostra l'estructura del bloc corresponent a la càrrega de capa SHP:



Imatge 54: Capa shape

Podem visualitzar la capa en format *shape* damunt la capa wms (El format wms és un estàndard creat pel OGC (Open Geospatial Consortium) Aquest estàndard internacional defineix un "mapa" com una representació de la informació geogràfica en forma d'un arxiu d'imatge digital convenient per a l'exhibició en una pantalla d'ordinador.) perquè pugem veure els carres d'una manera més global, així com podem comprovar que realment coincideixen amb els carrers de l'ortofoto.

Imatge 55: Càrrega capa lineal shape

7.2.6.3. Càrrega d'una capa PostGIS

Per carregar una capa *PostGIS* també s'utilitzen les etiquetes "*LAYER*" i "*END*", on es defineixen les propietats de la capa en aquest apartat. A diferència de la càrrega de capes en format SHP, cal fer una connexió de tipus *PostGIS*; en aquesta connexió es defineixen els paràmetres com l'usuari, contrasenya, localització del servidor i el número del port on es connecta.

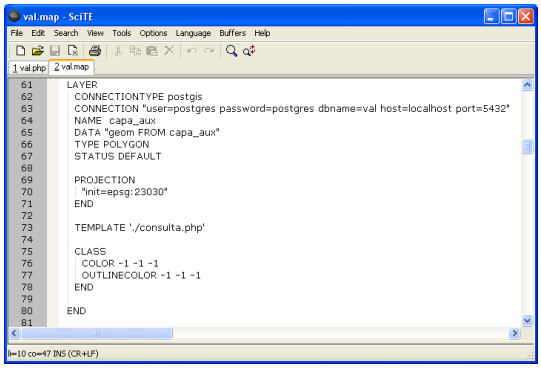
A més de tota aquesta informació, referent al tipus de connexió, cal indicar el nom de la capa a la qual fem referència, a més de tots els paràmetres com l'estat en el qual es visualitzarà la capa, el tipus de data que conté (punt, polígon...) i podem definir una simbologia entre altres propietats. En aquest cas s'ha estat utilitzant una simbologia relativament senzilla, que ens permet visualitzar les dades adequadament. També s'ha utilitzat la propietat que dóna transparència a una capa, ja que per l'ordre de visualització és més llegible.

Per carregar una capa que siga *PostGIS* tenim els següents paràmetres, que sempre aniran inclosos dins de l'objecte "*LAYER*":

- *CONNECTION TYPE*: Aquest paràmetre identifica el tipus de connexió que estem realitzant.
- *CONNECTION*: Aquesta línia s'usa per connectar-se a la base de dades, per tant, s'ha d'indicar l'usuari, la contrasenya, el nom de la base de dades, el servidor i el port. Els paràmetres han de ser exactament igual que la base de dades original.
- *NAME*: És el nom o etiqueta que li donarem a la capa.
- *DATA*: En aquest cas, aquest paràmetre indica el nom de la columna geomètrica de la taula que està inclosa a la base de dades, és a dir, la columna que conté la informació cartogràfica.
- *TYPE*: Indica quin tipus de contingut hi ha a la columna geomètrica (punts, línies, polígons…)
- *STATUS*: Indica la visibilitat de la capa.

- *PROJECTION*: Indica la projecció de la capa.
- *TEMPLATE*: En cas que la capa estiga vinculada amb una plantilla, aquest paràmetre és el que ho indica.
- *CLASS*: En aquesta variable es defineix el disseny de la capa que s'està publicant.

Per tant, si agrupen l'estructura en bloc, integrat ja a l'arxiu *map* ens queda de la següent manera:

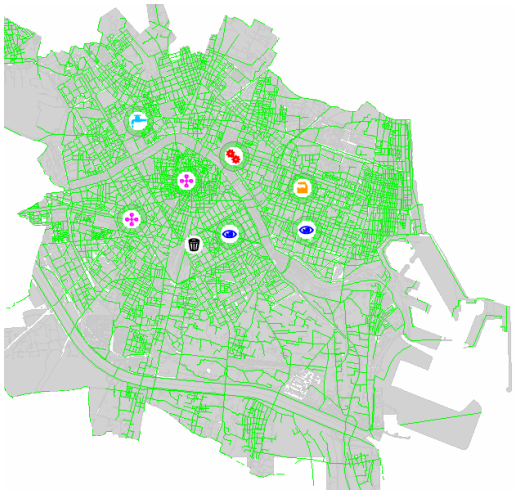


Imatge 56: Capa PostGIS

A la següent imatge tenim l'exemple de la càrrega d'una capa *PostGIS*, que com es pot comprovar no existeix variació en la visualització entre una capa *PostGIS* i una capa shape, si realment té les mateixes dades. En aquest cas, també s'ha dibuixat la capa dels carrers damunt de la capa wms, i evidentment, també coincideix.

Imatge 57: Càrrega de capa lineal *PostGIS*

A la següent imatge es carreguen les capes *PostGIS* de punts, línies i polígons:



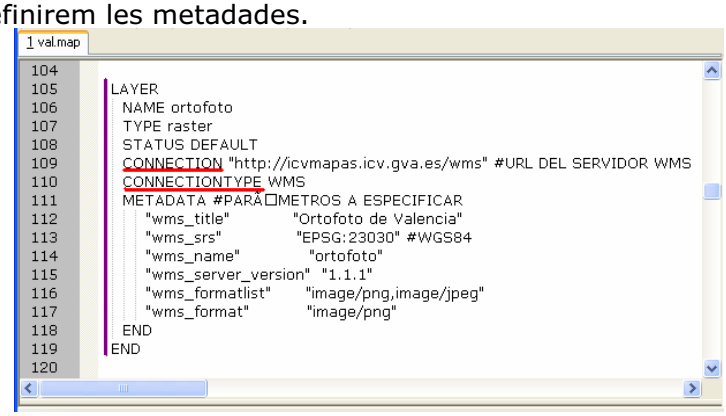
Imatge 58: Càrrega de capes de diferents tipus, *PostGIS*

7.2.6.4. Càrrega d'una capa wms

El servei Web Map Service (*WMS) definit pel OGC (Open Geospatial Consortium) produeix mapes de dades referenciades espacialment, de forma dinàmica a partir d'informació geogràfica.

Des de *MapServer* també es pot carregar remotament capes. Per exemple, en aquest cas, s'ha utilitzat una capa servida mitjançant l'estàndard WMS. Aquesta manera de carregar una capa és molt semblant a la manera que s'ha explicat a l'apartat anterior, ja que comparteixen molts paràmetres, de manera que s'explicarà el procediment però no tots els paràmetres desglossats [16]:

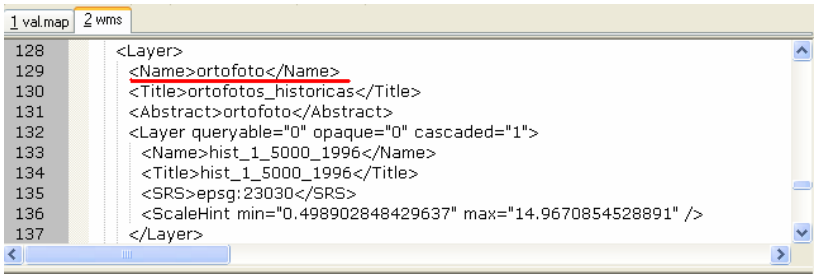
- Les dades van entre les etiquetes "*LAYER*" i "*END*" (color morat a la imatge 88). Ve definit pel nom i el tipus d'arxiu, en aquest car és "RÀSTER" perquè estem consultant una imatge. La connexió correspon a la URL del servidor que ens ha de donar la informació. El tipus de connexió respon al tipus d'estàndard, en aquest cas WMS. Finalment definirem les metadades.



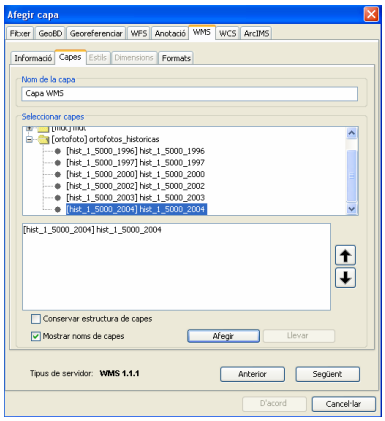
Imatge 59: Esquema capa WMS

- Cal especificar el nom exacte de la capa que anem a consultar. Aquest nom el podem saber fent una petició "*GetCapabilities*" al servidor, on tindrem accés a la informació que s'ofereix. No obstant, podem fer

servir el programari lliure "*gvSIG*" per poder fer la connexió WMS, on també podem conèixer els noms de les capes que s'ofereixen.

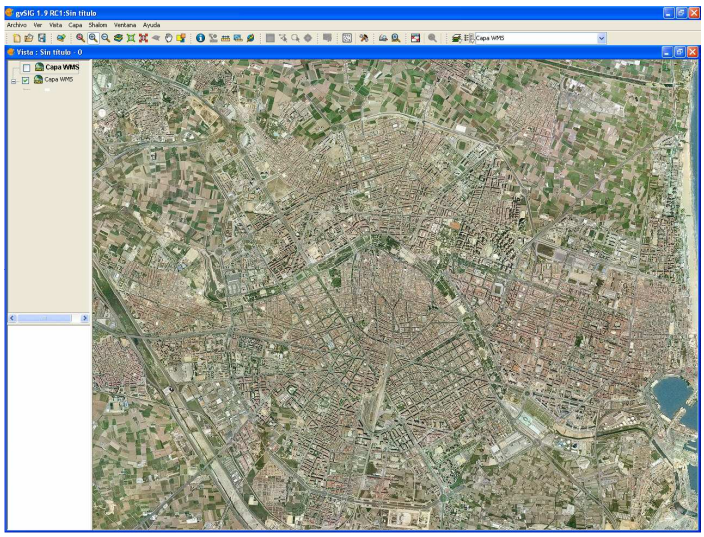


Imatge 60: Eixida *GetCapabilities*



Imatge 61: Càrrega de la capa amb gvSIG

Finalment, la capa que estem carregant té el següent aspecte utilitzant *gvSIG*:



Imatge 62: Visualització de l'ortofoto amb *gvSIG*

A la nostra pàgina tindrà l'aspecte següent:

Imatge 63: Ortofoto visualitzada a la pàgina web

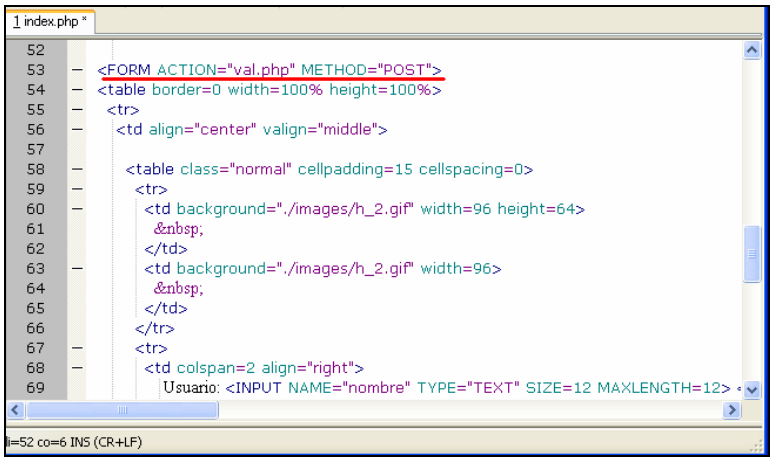
7.2.7. Arxius "PHP"

Per poder dinamitzar la pàgina web és necessari inserir el codi PHP dins del codi HTML, ja que HTML per sí mateix no dóna una activitat a la pàgina, sinó que és un element estàtic.

7.2.7.1. Arxiu índex

Aquest arxiu és el primer que es visualitza (d'ací el seu nom) i el que dóna accés a les diferents funcionalitats de les pàgines principals. És un codi senzill perquè ha de filtrar als usuaris i per tant no mostra cap tipus de dada cartogràfica, la seua funció és arreplegar tant l'usuari com la contrasenya i emmagatzemar-los.

És en aquest fitxer on comencem a relacionar-los. Per exemple, d'ací ja utilitzem l'arxiu plantilla "css", també estem enviant la informació al fitxer "val.php" gràcies al "*ACTION*" del formulari.



Imatge 64: Pàgina índex. Remarca de l'ACTION

7.2.7.2. Arxiu principal

És aquest arxiu el que suporta la càrrega més important i el qual conté la major càrrega de programació en llenguatge "php" amb la funcionalitat "*MapScript*". És necessari esmentar que per poder "enviar" la informació d'un fitxer a un altre cal iniciar amb les paraules clau "*session_start()*" (subratllat roig) i carregant la llibreria ".dll" que necessitem (en aquest cas *php_MapScript_48*) per donar la funcionalitat de "*MapScript*" al fitxer (subratllat rosa). Tot seguit l'arxiu es connecta a la base de dades amb la sentència (subratllat verd):

```
session_start();

/* Cargamos la DLL de mapscript */
dl('php_mapscript_48.dll');

$conn = pg_connect("host=localhost port=5432 password=postgres user=postgres dbname=val");
```

Imatge 65: Sentències inicials

En aquest moment, ja connectats a la base de dades podem recuperar les variables d'usuari i contrasenya que hem introduït al principi i serà en aquest moment quan s'avaluarà si l'usuari és autoritzat o no és autoritzat; l'avaluació es farà mitjançant una consulta a la base de dades i comprovant que existeix aquest registre.

També al començament del fitxer cal indicar quina és la plantilla *map* que s'utilitza, igualment cal indicar el camí on està ubicat l'arxiu. Açò és necessari perquè en aquesta part ja s'està carregant la cartografia.

```
$map_path="./";
$map_file="val.map";
```

Imatge 66: Variables relacionades amb el map

Per poder dibuixar el mapa cal crear-nos un objecte "*map*" que utilitzarà la ruta on està ubicat l'arxiu *map* i el nom de l'arxiu *map* que ha de carregar. Aquest ens permetrà carregar totes les capes que tenim a l'arxiu *map*, amb la propietat "*getLayerByName*". Quan es carregen les capes podem començar a col·locar els diferents elements que estaran ordenats espacialment a la pàgina, com poden ser la llegenda, l'escala i el mapa de referència; també les icones que accedeixen a la funcionalitat dels diferents usuaris.

Els zoom funcionen d'una manera que tots coneixem. L'icona ⊕ apropa els elements cartogràfics i l'icona ⊖ allunya els elements. Perquè no es descontrole la mesura d'allunyament i la d'apropament s'ha posat com a zoom màxim el valor de l'extensió total del mapa.

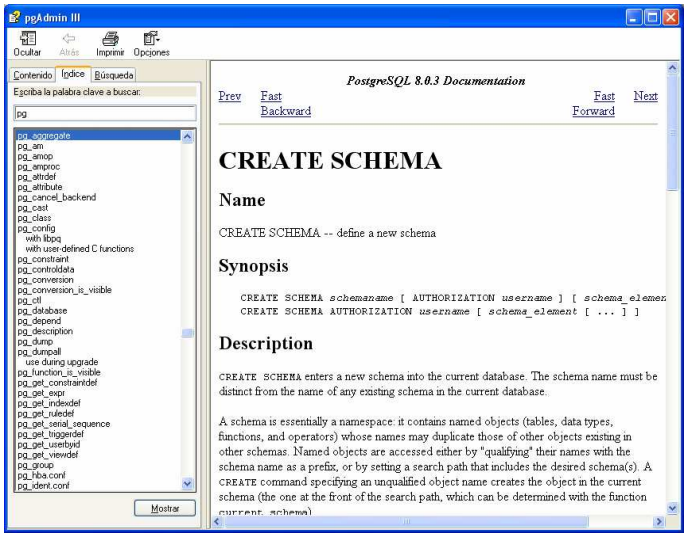
Quan l'usuari interactua amb la funció d'informació ⓘ és quan entren en joc les diferents aplicacions dels diferents usuaris: autoritzats i no autoritzats. S'envien els valors de les variables que diferencien un usuari i un altre, a més a més, també s'envien les coordenades del punt que es consultarà o bé que s'inserirà a la següent pàgina de "php", que és la de consulta.

7.2.7.3. Arxiu consulta

Les funcions que té aquest botó (consulta), és traure per pantalla un formulari que ens mostre tots els camps dels quals disposa la base de dades per poder omplir, d'aquesta manera tan senzilla podem anar emplenant la mateixa, consultar o edita-lar; així com crear un arxiu de text que ens informe de les persones afectades i que han de ser evacuades.

En aquest arxiu també hem de connectar a la base de dades, de manera que després de la connexió podrem procedir a realitzar les consultes a la base de dades. Com estem programant en un entorn de "PHP", cal buscar unes funcions en les quals pugem introduir llenguatge "SQL" des de "PHP" i així poder donar una dinamicitat a la pàgina.

Aquestes funcions "traductores" entre un llenguatge i un altre les podem trobar a l'ajuda de *pgAdmin III*, que les té arreplegades (la gran majoria) i ordenades:



Imatge 67: Funcions de pgAdmin

En el nostre cas s'han utilitzat les següents:

- *Pg_connect*: connecta a la base de dades en què volem treballar

- *Pg_exec*: Executa accions sql, com per exemple seleccionar, actualitzar i inserir dades a la base de dades.

- *Pg_numrows*: És una funció que ens dóna el nombre de registres que té una base de dades.

- *Pg_result*: assigna a una variable les dades a partir d'un identificador, resultat d'una funció *pg_exec*; és a dir, *pg_exec* genera un resultat numèric (identificador), es passa com a paràmetre a la funció *pg_result* i aquesta torna un valor, que li l'assignem a una variable.

En aquest fitxer s'han dividit les funcions a les quals cada usuari podia accedir, depenent de com accedisca, unes accions són accessibles per tots els usuaris, tals com a consulta, però unes altres solament amb accessibles per usuaris registrats, com donar d'alta nous punts o modificar-los, fent servir una variable

auxiliar que ens indica el mode en què estem en el moment d'iniciar la pàgina web, per tant es divideix en quatre blocs:

- Mode consulta: En aquest moment es recuperen els valors que estan a la base de dades i es mostren en la finestra web. En el moment de la consulta es crea el fitxer de text informatiu. Canviem al mode guardant.

- Mode guardant: S'emmagatzemen en variables els registres introduïts, bé perquè s'han editat dades o perquè s'estan inserint-ne noves, aleshores s'actualitzen les dades de la base de dades i, si no està creat, es crea l'àrea d'influència del registre puntual consultat o creat.

- Mode inserir: Aquest mode s'activa quan piquem al buit del mapa, és a dir, quan volem crear un nou registre a la base de dades. A la pàgina web ix un formulari que cal omplir i, per tant, es registrarà a la base de dades. Canviem a mode inserint.

- Mode inserint: Aquest mode insereix un nou registre a la base de dades, el crea, arreplegant els valors introduïts per l'usuari. Després de fer la inserció, canviem a mode guardant.

En línies generals, vist el codi, es mostren els diferents modes:

```
<?$modo=$_GET["modo"];
    if ($modo =="consultar"){
      //recupero los valores que contienen la base de datos sobre el punto que consulto.
      //cambio el modo a guardando, porque puedo haber cambiado algún campo y quiero que se guarden los cambios
      $modo = "guardando";
      //modo insertar, cuando pincho en el vacío del mapa, se activará este modo.
    }else if($modo =="insertar"){
      //contiene todos los campos vacíos, porque voy a insertar puntos, tendré que rellenar el formulario.
      //cambio el modo a insertando, que insertará en la tabla postgre los datos introducidos
      $modo = "insertando";
    }else if($modo =="guardando"){
      //se recuperan en las siguientes variables los campos que se ha rellenado o se han editado en el formulario
      //realizamos la actualización de la base de datos.
      pg_exec(
        "UPDATE focos SET fecha = '".$fecha."', ref_efecto = ".$efecto.",
          finalizado = ".$finalizado_bd.", coste = ".$coste.", ref_situacion = ".$situacion.",
          ref_actividad = ".$actividad.", ref_causa = ".$causa.", ref_medida = ".$medida.",
          adicionales = '".$medidas_adicionales."', previa = '".$previa."',
          resuelta = '".$resuelta."', observaciones = '".$observaciones."'
        WHERE id_foco = ".$num_foco
      );
    }else if($modo =="insertando") {
      //se recogen los valores de las variables que han sido introducidos por el usuario
      //la inserción del punto se hace mediante inert que inserta en la tabla
      pg_exec("insert into focos(id_foco, geom, fecha, ref_efecto, finalizado, coste,
        ref_situacion, ref_actividad, ref_causa, ref_medida, adicionales, previa,
        resuelta, observaciones
      ) values (
        ".$num_foco.", GeomFromText('POINT(".$_GET["geom"].")',23030),
        '".$fecha."', ".$efecto.", ".$finalizado_bd.", ".$coste.", ".$situacion.",
        ".$actividad.", ".$causa.", ".$medida.", '".$medidas_adicionales."',
        '".$previa."', '".$resuelta."', '".$observaciones."'
      )");
      //cambio a modo guardando
      $modo = "guardando";
      ?>
```

Imatge 68: Modes

7.2.8. Arxius HTML

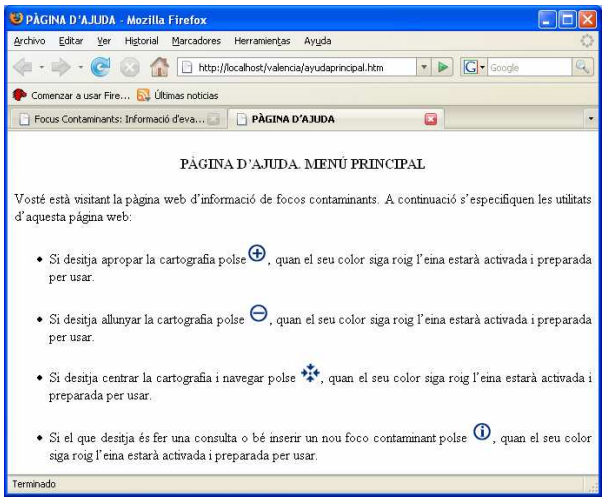
7.2.8.1. Pàgines d'ajuda

Per facilitar a l'usuari l'enteniment de la pàgina web es decideix posar a cada pas de la pàgina un enllaç que, en cas de dubte, aporte la informació necessària per poder continuar amb la navegació de la pàgina.

Per tant, n'hi ha un total de dues pàgines web d'ajuda:

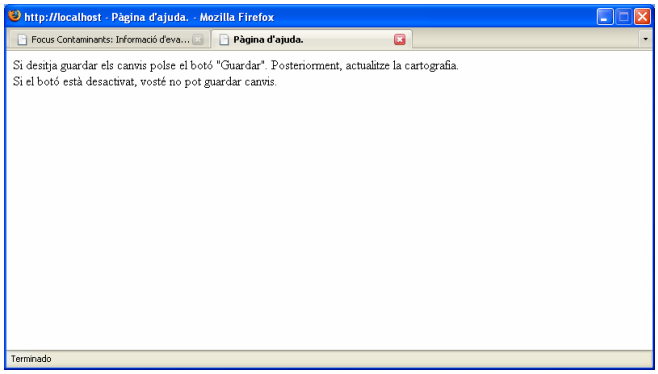
- Pàgina principal: S'explica el funcionament de les icones i el que es pot obtindre d'aquesta pàgina.

Imatge 69: Botó d'ajuda



Imatge 70: Pàgina d'ajuda principal

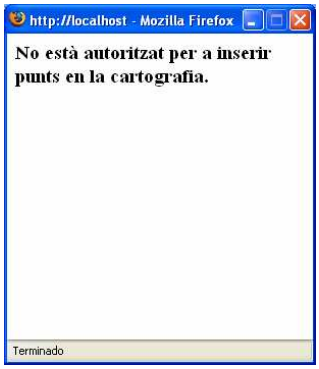
- Pàgina consulta: S'explica com s'ha d'omplir el formulari, donant algunes indicacions i explicant que un usuari autoritzat no pot fer més que consultar les dades.



Imatge 71: Ajuda de la consulta

7.2.8.2. Sense dades

Pot ser que un usuari no autoritzat intente inserir punts, i com que no està permès, s'ha tingut la consideració de crear un enllaç que mostre una pàgina informant que l'usuari no té els privilegis necessaris per poder inserir un punt. Aquesta pàgina mostra el següent aspecte:

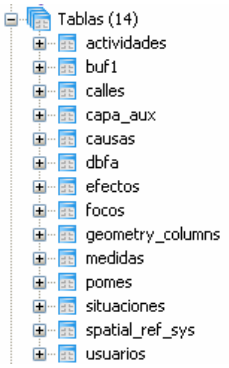


Imatge 72: Missatge no autoritzat

7.2.9. Base de dades

Passarem a comentar de què està composta finalment la base de dades.

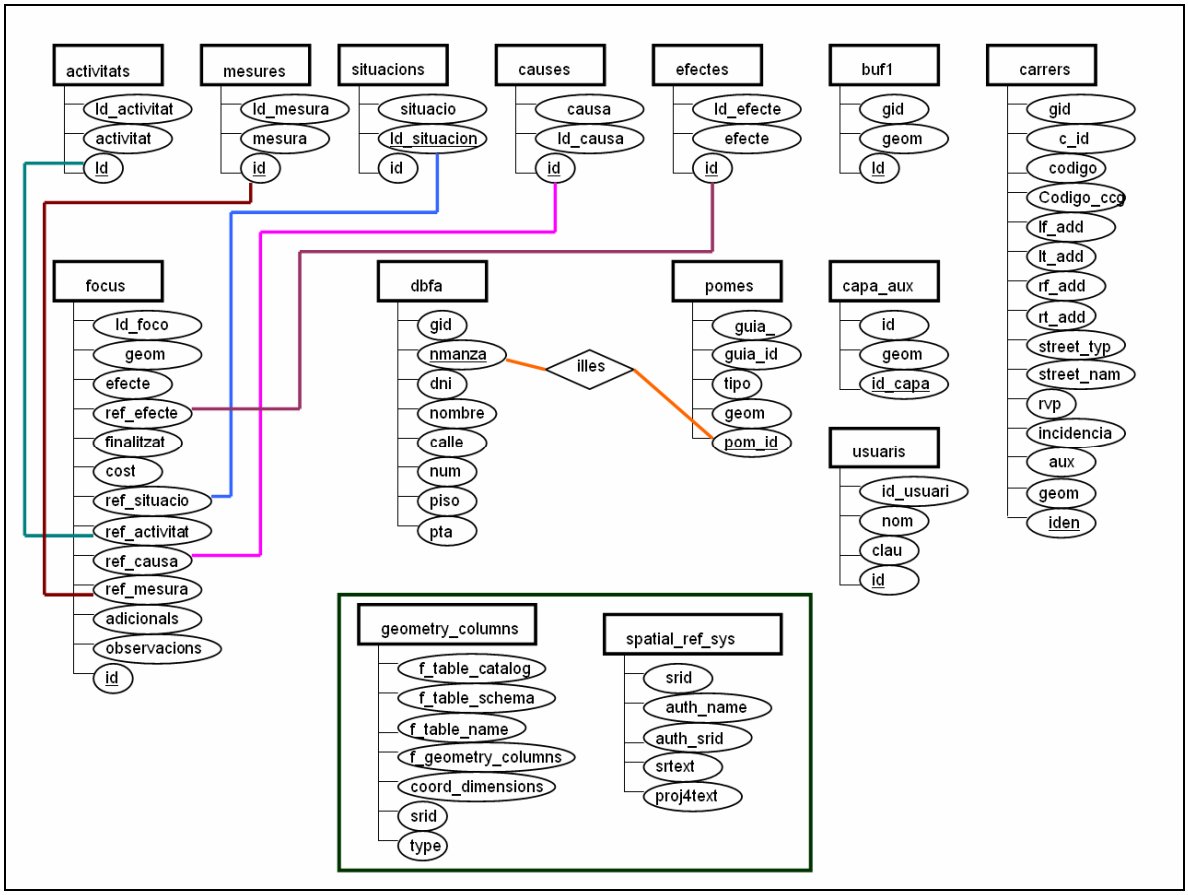
La base de dades està composta per un total de 14 taules que es mostren a continuació:



Imatge 73: Taules de la base de dades

La base de dades és espacial, i això es sap perquè tenim les taules de sistemes de referència i de geometria. Tenim un total de cinc taules geomètriques, que junt amb les taules no geomètriques ens proporcionen la informació necessària.

Quasi totes les taules tenen una informació que es complementa amb alguna informació de la resta de les taules, de manera que la taula dels focus contaminants està relacionada, per un número identificatiu, amb les taules "d'activitats", "mesures", "situacions", "causes" i efectes. El rombe simbolitza la integritat referencial entre les taules "dbfa" i "pomes".



Imatge 74: Relació de les taules

7.2.9.1. localització de la base de dades

En aquest apartat explicarem, que possibles localitzacions pot tenir la base de dades.

- local: la base de dades pots estar en el mateix servidor en l'està localitzat el servidor de mapes.
- remot: donada la possibilitat de connexió a través de la xarxa, incloent Internet o no, podem tenir emmagatzemada la base de dades en un altre ordinador, que pot estar en la nostra xarxa local, dins de l'empresa que ha contractat aquest servei o la base de dades pot estar localitzada en un altre equip havent de fer la connexió a través d'Internet.

En conclusió cada opció té els seus avantatges i inconvenients, en aquest cas s'ha decidit per l'opció local.

7.2.10. Relació de tots els elements

Quan ja hem estudiat tots els diferents components que té aquesta aplicació, cal relacionar-los perquè tot tinga sentit. A continuació s'exposarà l'esquema del raonament de l'aplicació que s'ha realitzat. En aquesta destaquen els punts forts, referits a les aplicacions que ens ofereix el projecte, com són el de consultar, inserir registres, crear àrees d'influència…

Primerament, l'aplicació (MapServer) es connecta a la base de dades, com havíem explicat en apartats anteriors. Una vegada connectat a aquesta, hi ha ja

comunicació entre el servidor i la base de dades (durant el procés de l'aplicació de tractament dels punts, són carregats en la pàgina web directament de la base de dades) que conté molta part de la informació, per tant es defineixen unes variables que determinaran el mode de la consulta i els drets de l'usuari; estes variables són les de "mode" i "autoritzat", la variable "mode" arreplega de l'arxiu "php" el mode que ha seleccionat l'usuari, és a dir, el de consulta o el d'inserció de nou punt:

- El mode "consulta" pren un formulari que carrega els valors de la base de dades corresponents als dades d'un registre determinat que seleccionem en la cartografia d'una manera visual. Si el punt no té àrea d'influència, es crearà al seu voltant en actualitzar la pàgina web.

- El "mode inserció" ens tornarà un formulari buit, en el que qual s'insereixen les dades pel teclat, aquestos camps que hem omplit s'inseriran en la base de dades en polsar el botó "guardar". En aquest cas, l'àrea d'influència es crearà en actualitzar la pàgina web i no caldrà tornar a picar al punt per crear-lo.

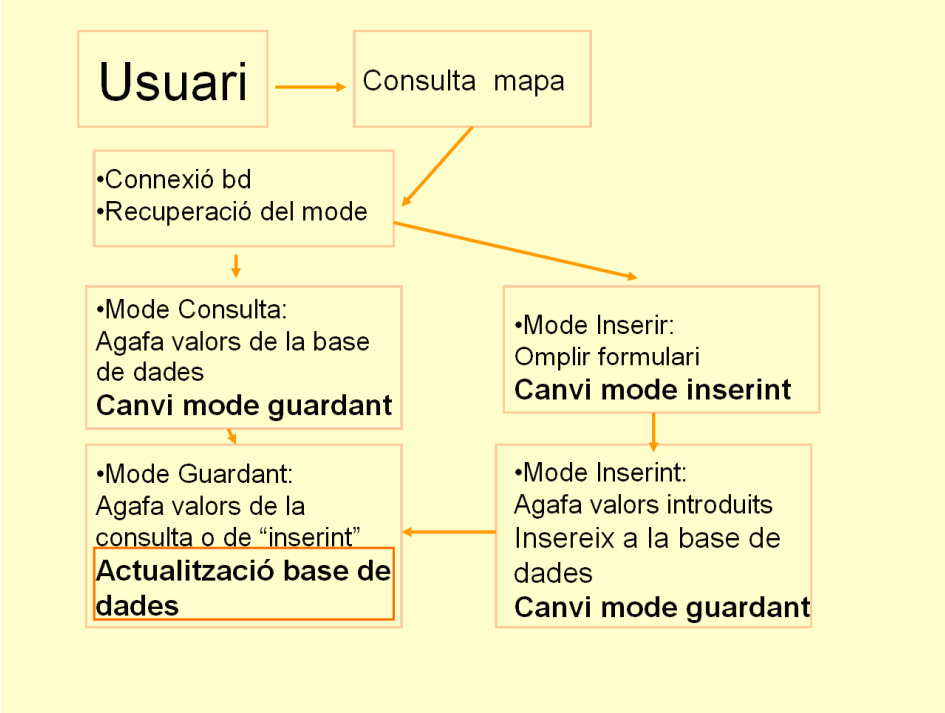
A nivell de programació, amb només dos modes és prou complicat resoldre l'aplicació, per tant s'afegeixen dos modes més: el "mode guardant" i el "mode inserint".

- El "mode guardant" inserirà o actualitzarà les dades que hàgem inserit. El que realitzarà serà un *UPDATE*.

- El "mode inserint" crearà un nou registre en la base de dades, és a dir, estem parlant d'un *INSERT*.

El recorregut que farà l'aplicació quan es trobe amb la variable mode és el següent: inicialment, "mode" ve de la pàgina inicial "php", i pot ser "consultar" o "inserir". Si és consultar, els camps apareixeran plens (dades que venen de la base de dades), i al final del codi es canviarà el mode a "guardant", perquè pot ser que s'haja editat alguna dada. Canviant el mode a "guardant" ens assegurem que els canvis es guardaran a la base de dades, sense duplicar el registre. Si el mode és "inserir", haurem d'omplir els camps, el mode canviarà a "inserint" i aquest arreplegarà les dades que hagem inclòs, llavors realitzarà l'*INSERT* quan li donem a guardar, el mode canviarà a "guardant", serà llavors quan el registre quede emmagatzemat a la base de dades.

També es crearà el registre que correspon a l'àrea d'influència, si no existeix ja, a la base de dades; així com el fitxer de text amb les persones que han de ser evacuades. D'un mode esquemàtic resulta:



Imatge 75: Esquema del procés

8. Exemple pràctic

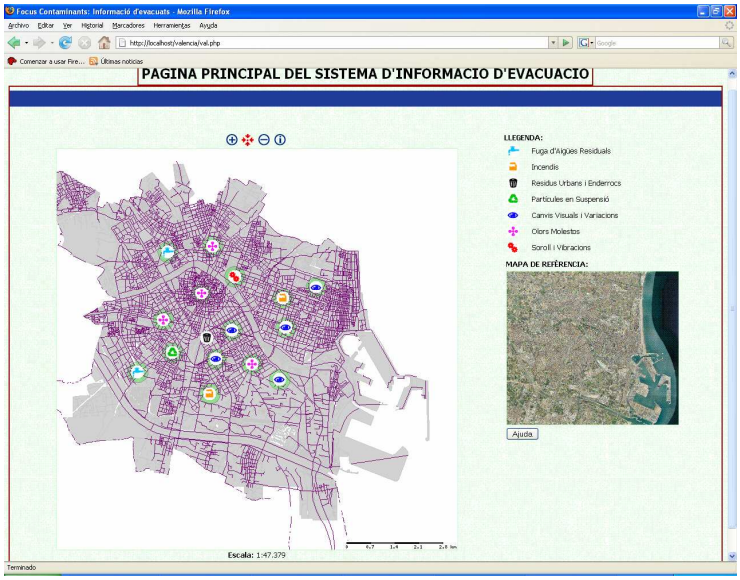
En aquest apartat comentarem i mostrarem, pas a pas, les utilitats que s'han dissenyat i les funcions de l'aplicació. Aquest apartat es realitza quan ja hem acabat tota la programació, per tant poden haver variacions de les imatges que s'han estat utilitzant com a exemple als apartats anteriors. No obstant, l'aplicació definitiva té l'aspecte que es mostra en aquest punt.

8.1. Usuari autoritzat

L'usuari inicia l'aplicació i es troba que ha de posar un nom d'usuari i una contrasenya. En escriure'ls ha de polsar el botó "Entrar":

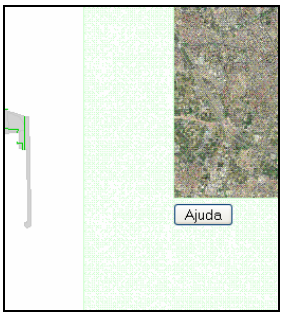
Imatge 76: Pàgina d'inici

En entrar, i com és usuari autoritzat es troba el següent aspecte:



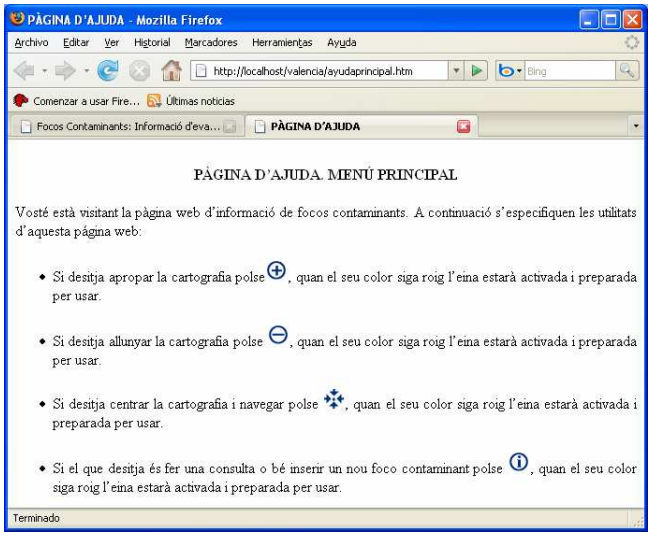
Imatge 77: Pàgina principal

Si no és un usuari que habitualment consulta la pàgina, disposa d'un botó d'ajuda que dóna la informació pertinent per poder procedir a l'ús de la taula.



Imatge 78: Botó d'ajuda

Quan s'ha polsat el botó tenim la següent informació:



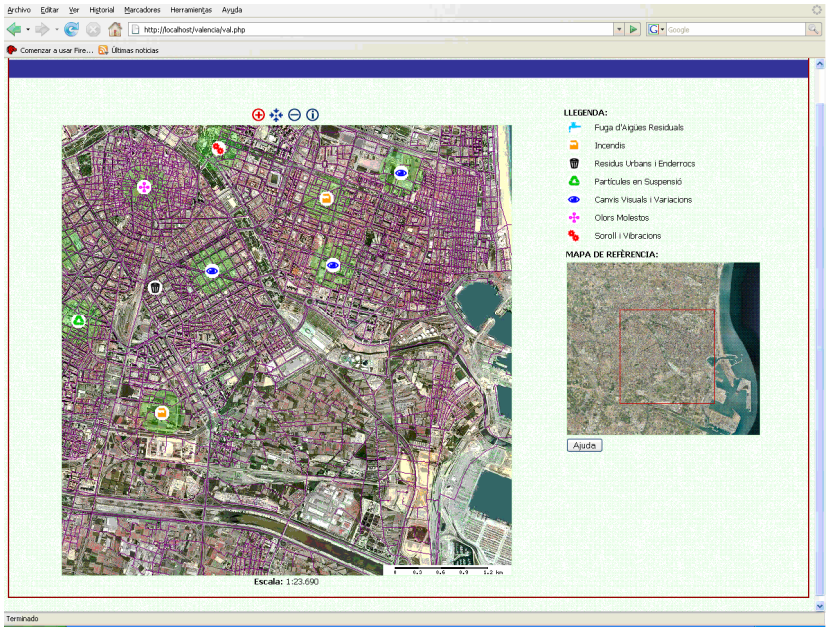
Imatge 79: Pàgina d'ajuda

Com la finestra d'ajuda s'obri en una nova pàgina, podem tancar-la i tornar a la pàgina on estaven. En un tamany més gran ix la cartografia en format imatge, en aquesta imatge apareixen dades cartogràfiques superficials (illes de cases), lineals (eixos de carrers) i puntuals (focus perillosos); a la dreta una llegenda dels punts (perquè puguen ser interpretats), i baix una imatge que representa el mapa de referència.

Si ens fixem, a la part superior, per damunt de la imatge principal, n'hi ha quatre botons:

Imatge 80: Botons

Si l'usuari polsa, de dreta a esquerra, el primer botó augmentarà el zoom de la imatge, de manera que es visualitzarà la cartografia més gran.

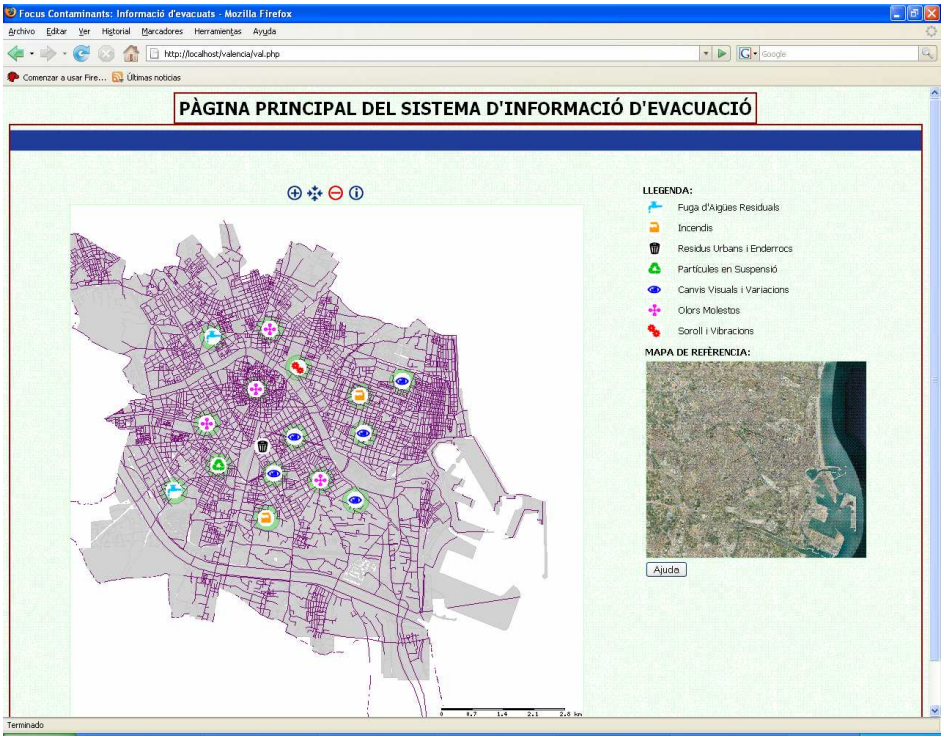


Imatge 81: Aproximació a la cartografia

En polsar el segon botó es té l'opció de navegar per la cartografia, perquè centra les dades cartogràfiques on es polsa el ratolí.

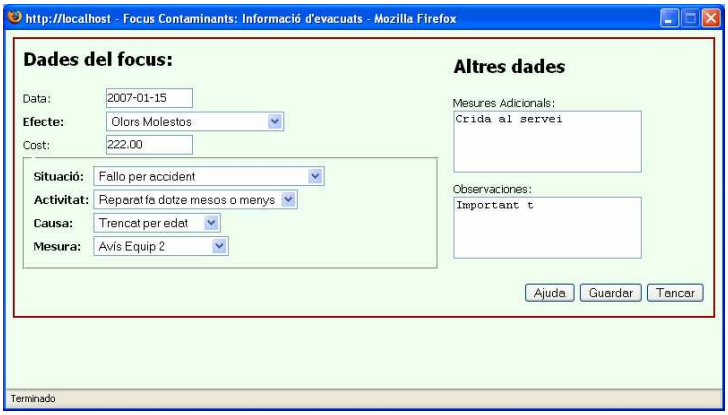
Imatge 82: Navegació

Si es polsa el tercer botó, s'obté el resultat invers al primer, es a dir, la cartografia s'allunya. No es pot allunyar més de l'extensió del mapa total.



Imatge 83: Allunyament

En provar les icones dels zoom, es pot fer la consulta. Es polsa l'icona d'informació, i en comprovar que ha canviat a color roig, polsem sobre la imatge que representa la cartografia. Podem polsar sobre un punt existent, en aquest cas ens apareixerà una finestra que arreplega les dades d'aquest punt.



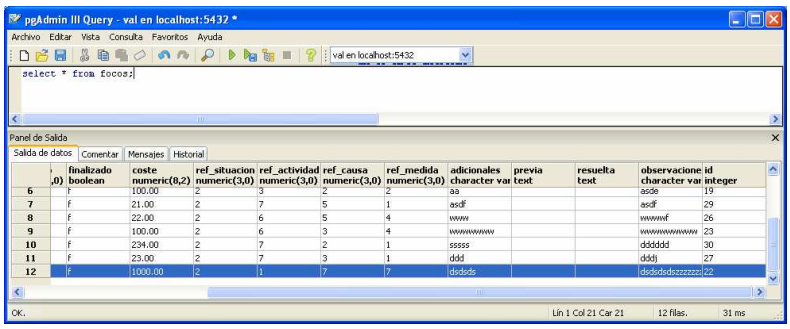
Imatge 84: Formulari de la consulta

Com l'usuari és autoritzat significa que pot editar alguna de les dades que mostra el formulari, per tant, podia canviar, per exemple, l'estat del focus. Aleshores, l'usuari realitzarà els canvis oportuns i polsarà el botó que salva els canvis.

Imatge 85: Formulari editat

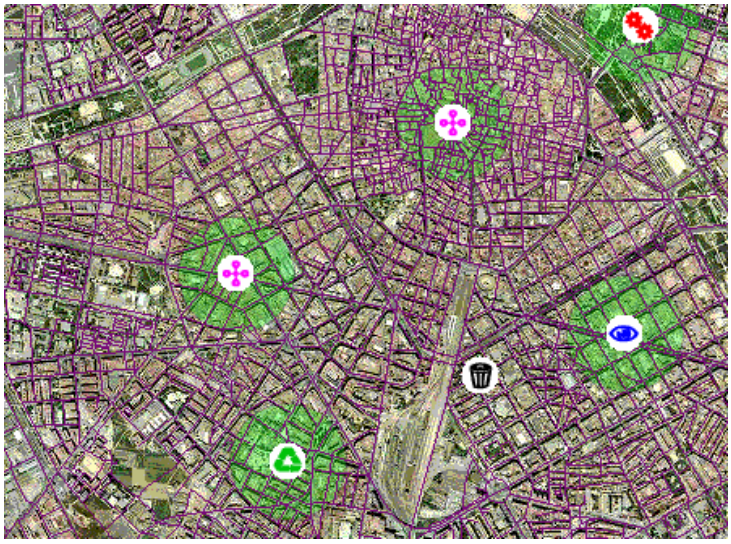
Per comprovar que tot ha anat bé i que realment s'han fet els canvis pertinents a la base de dades, cal actualitzar la pàgina principal i tornar a consultar el punt editat. En eixir el formulari es poden verificar els canvis (Imatge 115).

També es poden consultar els canvis a la base de dades directament:



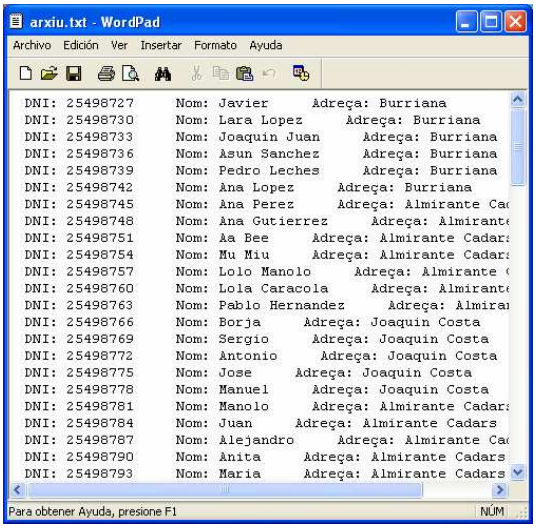
Imatge 86: Visualització en la base de dades

A més d'haver consultat el punt i editar-lo, si no n'hi havia àrea d'influència al punt, aleshores es crearà.



Imatge 87: Àrees d'influència o *buffers*

Si minimitzem els navegadors i busquem l'arxiu de text que s'ha creat, es pot comprovar que tenim la informació dels habitants afectats dins de l'àrea d'influència:

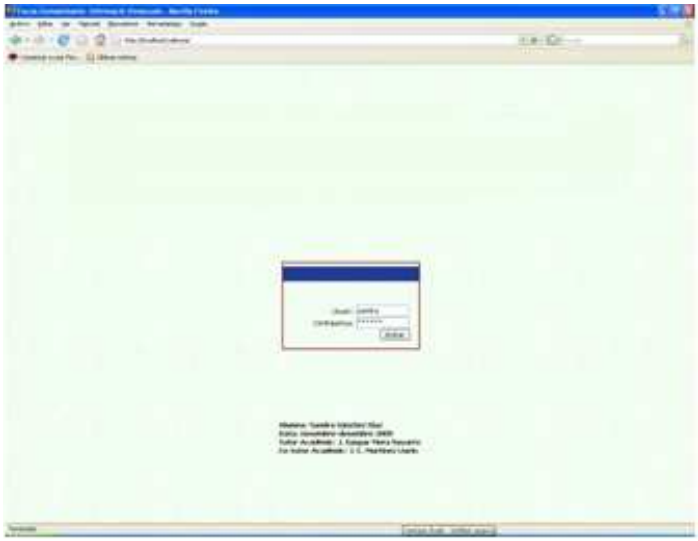


Imatge 88: Llistat dels afectats

8.2. Usuari no autoritzat

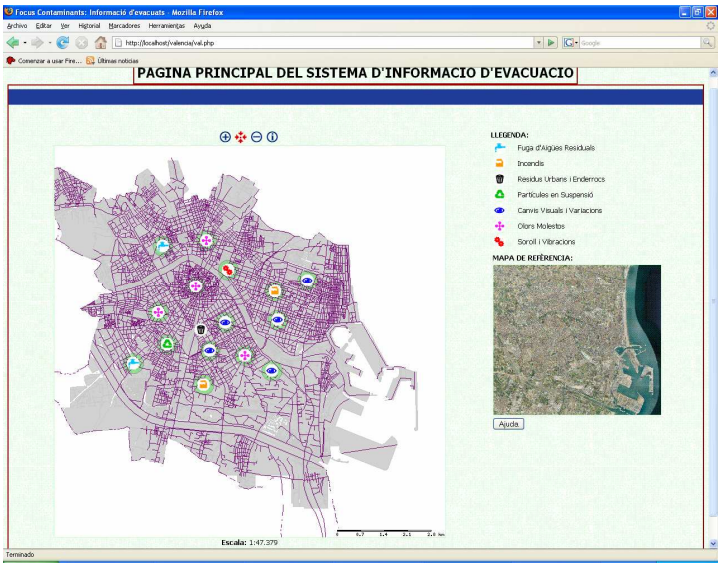
L'usuari no autoritzat té menys privilegis que l'usuari autoritzat, de manera que serà comú una part del text de l'apartat anterior; no obstant, és interessant explicar-ho tot pas a pas perquè d'aquesta manera no cal llegir l'apartat anterior per poden entendre aquest apartat.

L'usuari inicia l'aplicació i es troba que ha de posar un nom d'usuari i una contrasenya. En escriure'ls ha de polsar el botó "Entrar":



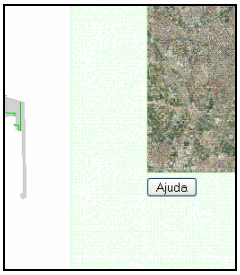
Imatge 89: Inici d'usuari no autoritzat

En entrar, i com és usuari no autoritzat es troba el següent aspecte:



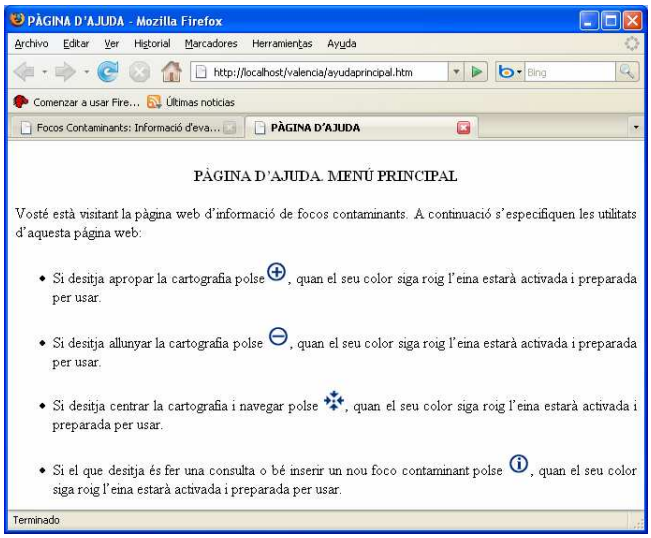
Imatge 90: Aspecte pàgina web

Si no és un usuari que habitualment consulta la pàgina, disposa d'un botó d'ajuda que dóna la informació pertinent per poder procedir a l'ús de la taula.



Imatge 91: Botó d'ajuda

Quan s'ha polsat el botó tenim la següent informació:



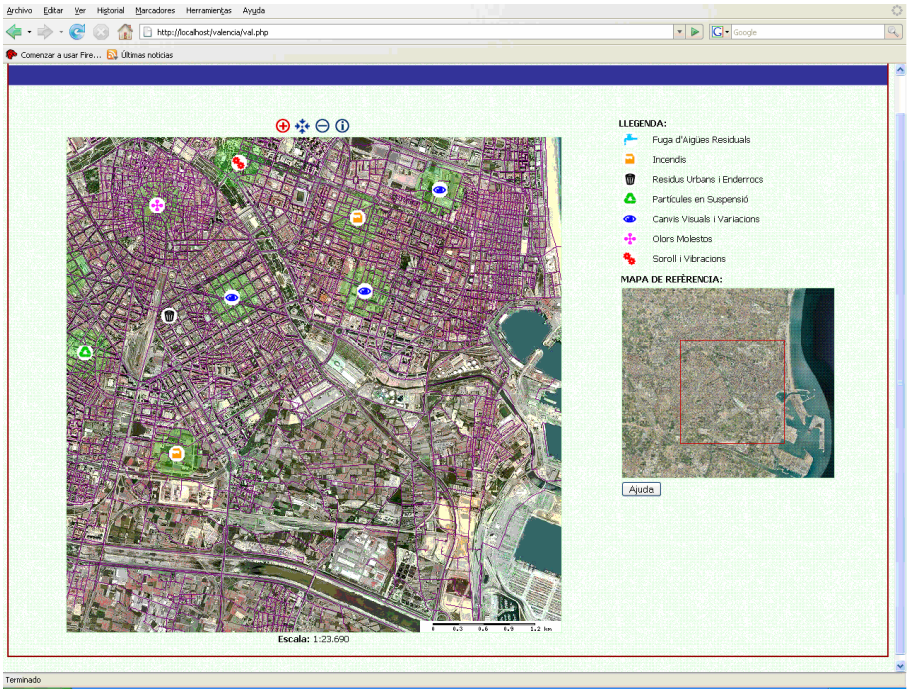
Imatge 92: Pàgina d'ajuda

Com la finestra d'ajuda s'obri en una nova pàgina, podem tancar-la i tornar a la pàgina on estaven. En un tamany més gran ix la cartografia en format imatge, en aquesta imatge apareixen dades cartogràfiques superficials (illes de cases), lineals (eixos de carrers) i puntuals (focus perillosos); a la dreta una llegenda dels punts (perquè puguen ser interpretats), i baix una imatge que representa el mapa de referència.

Si ens fixem, a la part superior, per damunt de la imatge principal, n'hi ha quatre botons:

Imatge 93: modes

Si l'usuari polsa, de dreta a esquerra, el primer botó augmentarà el zoom de la imatge, de manera que es visualitzarà la cartografia més gran.

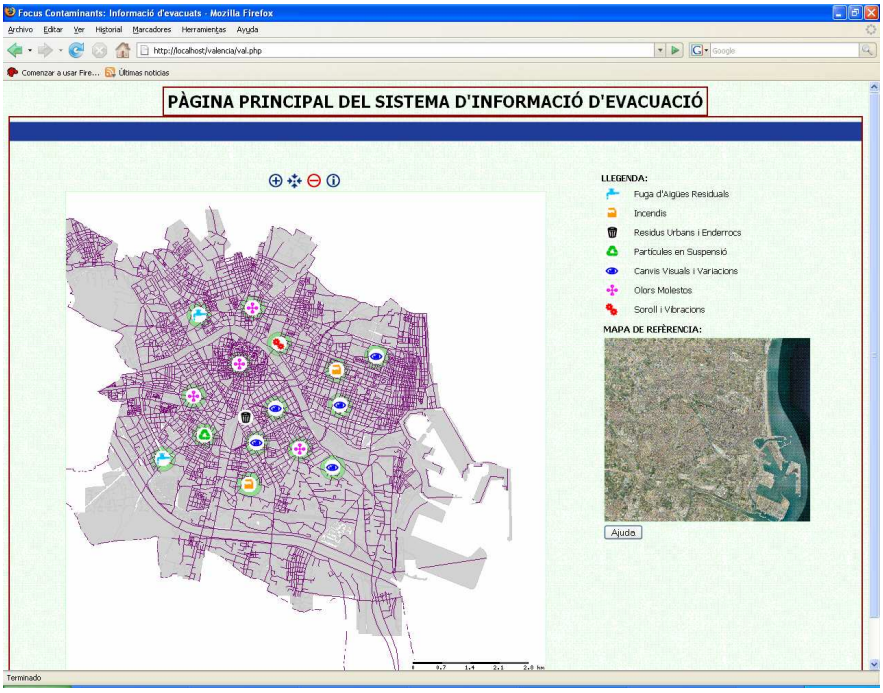


Imatge 94: Ampliació de la cartografia

En polsar el segon botó es té l'opció de navegar per la cartografia, perquè centra les dades cartogràfiques on es polsa el ratolí.

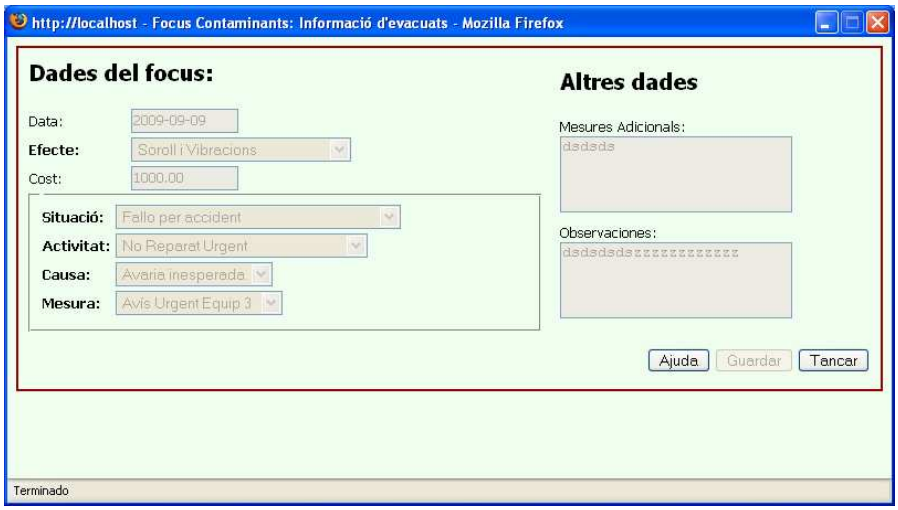
Imatge 95: Navegació

Si es polsa el tercer botó s'obté el resultat invers al primer, es a dir, la cartografia s'allunya. No es pot allunyar més de l'extensió del mapa total.



Imatge 96: Allunyament

En provar les icones dels zoom, es pot fer la consulta. Es polsa l'icona d'informació, i en comprovar que ha canviat a color roig, polsem sobre la imatge que representa la cartografia. Podem polsar sobre un punt existent, en aquest cas ens apareixerà una finestra que arreplega les dades d'aquest punt.



Imatge 97: Formulari

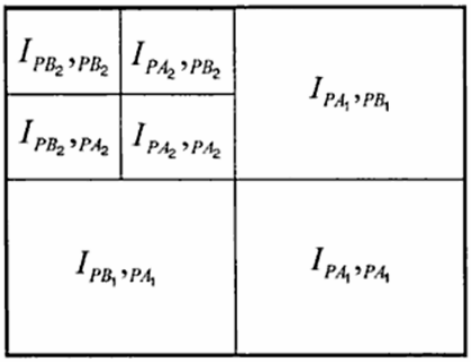
Com l'usuari és no autoritzat significa que pot consultar les dades que mostra el formulari, però no pot fer cap edició. Açò l'usuari ho identifica en veure que estan bloquejats, tant el botó com els camps consultats.

Una volta l'usuari ha consultat la informació pot tancar la finestra i tornar a visualitzar la pàgina principal.

9. Problemes trobats.

9.1. Càrrega capa wms

Un dels problemes que s'ha trobat per poder carregar la capa WMS de forma correcta és que *MapServer* té un temps d'espera limitat, el qual està per baix del temps de resposta de la connexió a la capa WMS (segons a quines escales). Açò suposa un problema perquè no es sap el temps que s'ha d'esperar, el navegador pot haver acabat de treballar i la capa no carregar-se finalment. Per poder reduir la importància d'aquest inconvenient cal augmentar el zoom que fem, perquè com que el format es de tipus ECW tindrem més detall com més zoom fem.



Imatge 98: Format ECW [4 (2002: pàg 399)]

9.2. Temps d'espera

Com a problema trobat també ha de constar que el carregar una capa WMS, i per tant augmentar la quantitat d'informació que n'hi ha al mapa, ralentitza la càrrega i el procés d'informació i de consulta; de manera que pot ser no resultara rentable carregar dades massa pesades, per exemple capes ràster amb connexió (WMS, WFS…), si volem realitzar consultes espacials o operacions amb bases de dades que contenen molts registres.

Potser amb una altra metodologia resultara més factible. Per exemple, si volem carregar moltes capes remotes podíem plantejar-nos fer servir una Infraestructura de Dades Espacials (IDEE), o utilitzant *Open Layers*…

10. Conclusions

En moltes ocasions, contràriament al que pensem, el programari lliure és més fiable i competent que el programari no lliure perquè l'usuari pot comprovar les funcions que s'estan utilitzant. Suposem que estem fent algun càlcul i no s'obté el resultat desitjat, amb un programari lliure podem consultar els tipus de funcions que s'estan utilitzant i comprovar realment que són les que es necessiten, però amb programari privatiu hauríem de començar un procés de contacte amb l'empresa, d'espera a una resposta... però mai es tindria accés a la informació del programari, i per tant no es podrien fer modificacions.

Per a la realització de tota l'aplicació s'ha utilitzat la combinació de programari lliure i programari no lliure. Sí que és ben cert que actualment el programari lliure està molt avançat i ofereix moltes possibilitats, algunes millors, algunes menys millors, però cal saber que algunes ferramentes que s'han utilitzat amb programari no lliure ha sigut perquè la ferramenta era molt més potent (topologia), o bé era molt més ràpid (crear un camp autonumèric). Algunes de les coses que s'han fet amb programari no lliure es podien haver programat amb programari lliure, però per rapidesa, seguretat i estalvi de temps s'ha utilitzat el programari no lliure.

Per realitzar aplicacions que encadenen més d'un fitxer és òptim treballar amb rutes relatives i no absolutes, ja que aquestes limiten molt l'intercanvi de l'aplicació a uns altres ordinadors i ens poden causar problemes si la ruta absoluta té espais o accents o caràcters estranys. En el desenvolupament de l'aplicació s'ha treballat sempre amb rutes relatives.

La transformació de capa de tipus "shape" (illes cartogràfiques) a la base de dades s'ha realitzat a Linux amb l'objectiu d'utilitzar coneixements que s'han adquirit en la titulació. A més a més, també s'ha buscat mostrar al lector que és compatible treballar amb Linux i Windows XP conjuntament, ja que cada sistema operatiu ens dóna unes utilitats (en moltes ocasions similars, però amb més facilitat a un en comptes d'un altre) i poden ser compatibles. Finalment s'ha utilitzat Linux per potenciar l'ús de programari lliure. No obstant, per provar que el resultat seria el mateix la capa de "carrers" s'ha transformat amb Windows XP. El resultat és exactament el mateix, però ara també coneixem una alternativa als sistemes operatius privatius.

Per poder desenvolupar el projecte s'ha necessitat de gran varietat de bibliografia, concretament en algunes parts existia poca (o nul·la) bibliografia en castellà. Fonamentalment tota la bibliografia és en anglès. Per aquesta raó s'ha redactat el projecte com un tutorial, perquè puga ser utilitzat com material de consulta en la llengua cooficial a la Comunitat Valenciana.

11. Futures línies de treball

Es podria augmentar la utilitat del projecte ampliant el nombre d'aplicacions relacionades amb la base de dades, es a dir, resulta molt útil treballar amb la base de dades a través d'Internet, sense tocar la base de dades físicament, filtrant diferents perfils d'usuaris, de manera que segons el perfil es puguen desenvolupar unes tasques o unes altres.

Per poder ampliar més el formulari es va pensar en fer una funció que recercarà els carrers de tota la ciutat de València. La funció s'ha arribat a fer però no s'ha pogut arribar a obtindre els resultats desitjats i per tant s'ha eliminat de l'aplicació. No obstant es deixen entreveure directives orientades a aplicacions com aquestes.

També es podia fer un càlcul de rutes mínimes (en temps) per Internet, relacionant bases de dades espacials, que resultaria molt útil per poder arribar d'una manera més ràpida al lloc on s'està produint una emergència. Actualment, PostGIS té un mòdul per a calcular rutes.

També cal tenir en compte la potència que té aquesta aplicació, ja que es pot aplicar a un altre tipus de necessitats, com a carreteres, ports, circuits, etc. Tan sol canviant la cartografia i la bases de dades.

12. Bibliografia

1. – Martínez Llario, J. C. (2008). *Talleres prácticos de iniciación a PostGIS (Linux y PostgreSQL)*. Valencia: Universidad Politécnica de Valencia.

2. – Coll Aliaga, E. ET ALII (2005). *Introducción a la publicación de cartografía en Internet*. Valencia: Universidad Politécnica de Valencia.

3. – Programari lliure [online] <http://www.gnu.org/philosophy/free-sw.es.html>

4. – Lerma García, J. L. (2002). *Fotogrametría moderna: analítica y digital*. Valencia: Universidad Politécnica de Valencia.

5. – *TUTORIAL DE HTML 4.0* [online] <http://www.programacion.com/html /tutorial/curso/>

6. – *MANUAL DE HTML* [online] <http://www.webestilo.com/html/>

7. – *WIKIPEDIA* [online] <http://es.wikipedia.org/wiki/>

8. – *Junta de Andalucía* [online] <http://www.juntadeandalucia.es>

9. – *Tutorial HTML* [online] <http://www.webestilo.com>

10. – *Pàgina web oficial* [online] <http://MapServer.org/>

11. – *Tutorial Javascript i PHP* [online] <http://www.desarrolloweb.com/>

12. – *PHP MapScript* [online] <http://phpexperto.blogspot.com/2007/11/php-MapScript-de-MapServer-parte-1.html>

13. – *MapTools* [online] <www.maptools.org>

14. – *PostgreSQL* [online] <http://www.PostgreSQL.org/>

15. – *Open Source Initiative* [online] <www.opensource.org>

16. – *Institut Cartogràfic de València* [online] <www.icv.gva.es>

17. –*MapServer for dummies* [online] <http://gabrielortiz.com/index.asp? Info=092>

18. – *Conselleria d' Infraestructures i Transport* [online] <http://www.cit.gva.es/ informacion-general/inf-institucional/competencias/>

19. – *Pàgina web gvSIG* [online] <http://www.gvSIG.gva.es/que-es-gvSIG/>

20. – *Postgis Manual* [online] <http://www.cp-idea.org/ nuevoSitio/documentos /tecnologia/PostGIS-manual.pdf>

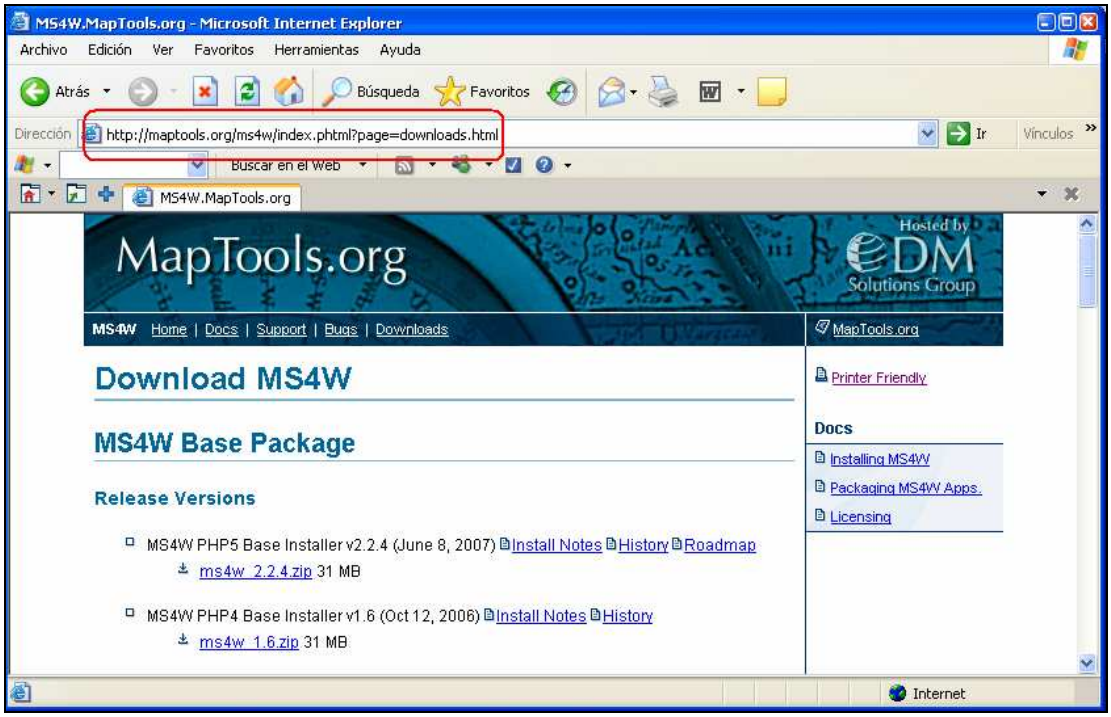
21. – *Paso de variables por referencia vs por valor [C#]* [online] <http://casidiablo.net/funciones-referencia-valor-c-sharp/>

22.- *Yahoo respuestas* [online] <http://mx.answers.yahoo.com/question/index? qid=20071029111907AA7KmOU>

13. Anexo

13.1. Guia-esquema d' instal·lació

En aquest cas hem escollit descarregar-nos el paquet d'instal·lació MS4W per ser més senzill d'instal·lar que tots els components per separat. Si entrem a la pàgina web oficial de MapServer, en l'apartat de descàrregues podem escollir el paquet desitjat, en aquest cas triem ms3w_2.1.zip



Imatge 99: web de descàrrega [10].

Com s'ha dit abans, s'instal·larà amb el paquet el conjunt d'Apatxe, MapServer i PHO entre uns altres. Normalment la instal·lació d'aquest paquet de programes no resulta una cosa senzilla; per això s'especifiquen uns passos, perquè es pugui reproduir en un futur.

- S'ha d'extraure el fitxer ".zip" al directori que desitgem, per exemple "C:/msw4", cal tenir en compte que no sigui "C:/ms4w/ms4w".
- posteriorment s'executarà l'arxiu "instal.bat" i tot seguit es reiniciarà l'ordinador. En el moment que ja s'hagi reiniciat, s'obre un navegador i comprovarem l'adreça "http://localhost/", si tot ha anat correctament llavors sortirà la pàgina principal del "msw4".
- Tot seguit es busca l'arxiu "monitor.exe" perquè aparegui el monitor de l'Apatche.
- Després es crea el directori "www", i dins donem aquest directori "temp" o "tmp".
- En aquest moment, s'ha de canviar l'arxiu "httpd.conf", concretament les llinies:

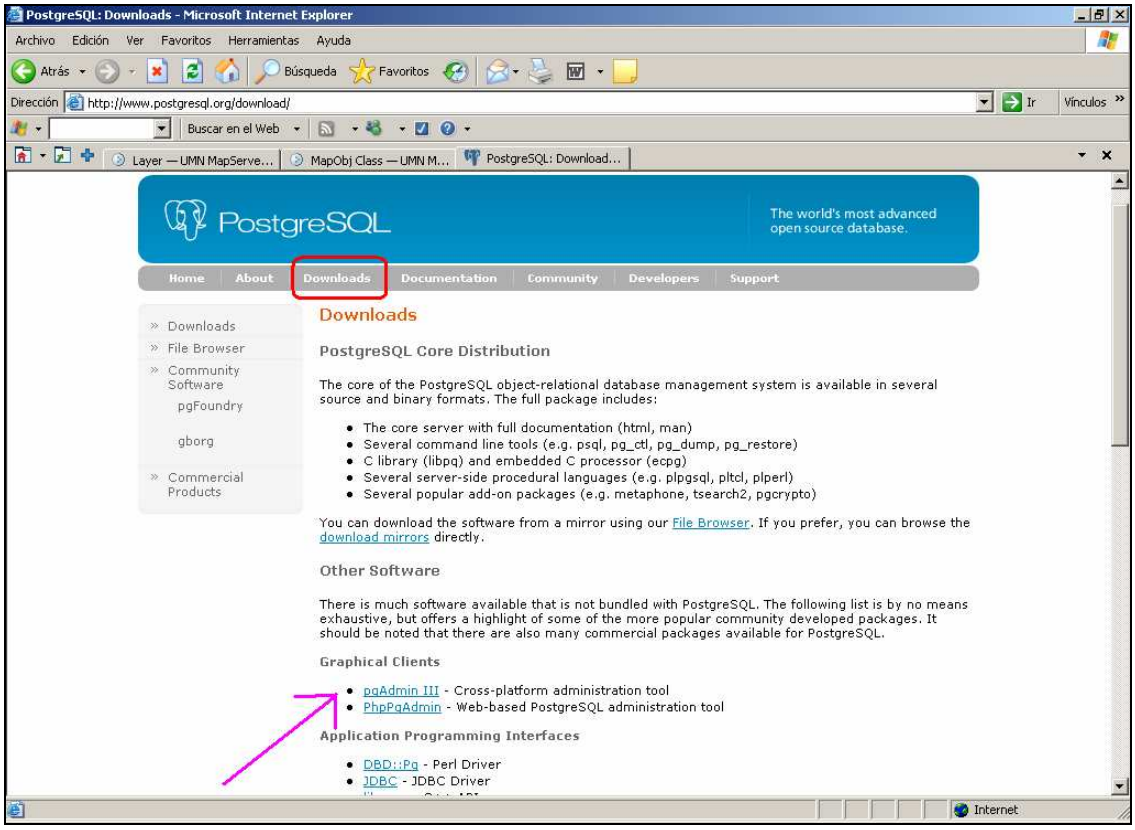
- "Document Root": Es comenta aquesta línia, es copia y es pega en la línia següent canviant la ruta on es troba el directori "www".
- "Directory": Es comentar la línia i escrivim la ruta on es troba el directori "www".
- "Alias": Es canvia l'àlies del fitxers temporals, indicant la ruta on es trova el nostre directori "temp" o "tmp".

- L'adreça web serà: http://localhost/.

En aquest punt cal remarcar que per a el desenvolupament d'aquest projecte es necessita la llibreria "php_MapScrip_46.dll" per a poder accedir a les funcionalitats de PHP MapScrip.

13.2. PostgreSQL i pgAdmin III

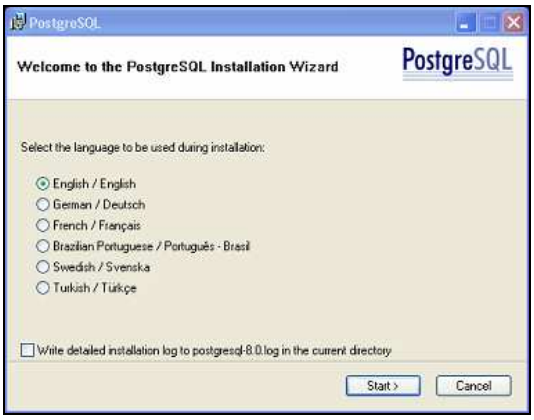
Aquest programa ens serveix per a gestionar les bases de dades. La seua descàrrega és pot fer des de la la pàgina web oficial, i la seua instal·lació es relativament senzilla.



Imatge 100: Pàgina web de *PostgreSQL* [14]

13.2.1. Guia-esquema d'instalació

En aquest programa és mes senzilla la instal·lació, de totes maneres és bona costum documentar al màxim el procés d'instal·lació perquè qui lo necessite pugui reproduir el procés. La instal·lació de PostgreSQL es executa executant l'arxiu PostgreSQL-8.0.msi del directori PostgreSQL. Triem la llengua en què volem que s'instal·li el programa.

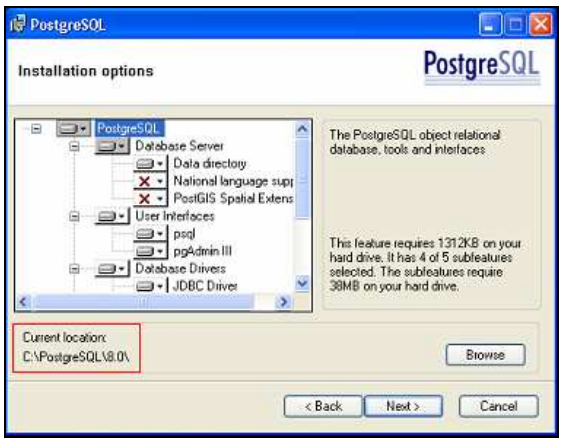


Imatge 101: Elecció de la llengua

Picarem Start i ens eixirà una advertencia que indica que si la versió d'Open SSL del nostre equip és antiga, el programa actualizarà la versió per una més recent.

Imatge 102: Opcions d'instal·lació

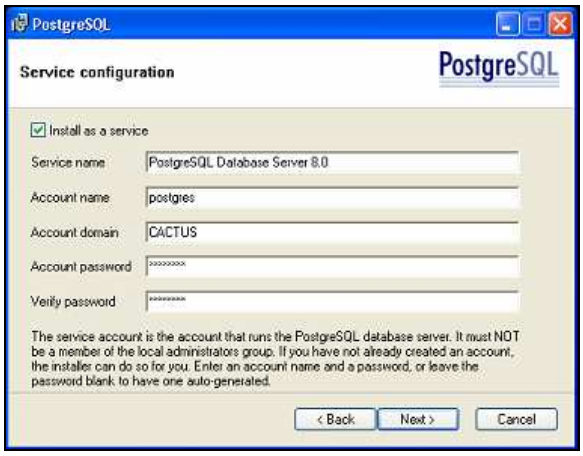
Ens conservaran totes les opcions d' instal·lació excepte la ruta d'instalació, que l'elegirem. Per exemple "C:\PostgreeSQL\8.0".



Imatge 103: Creació d'un usuari

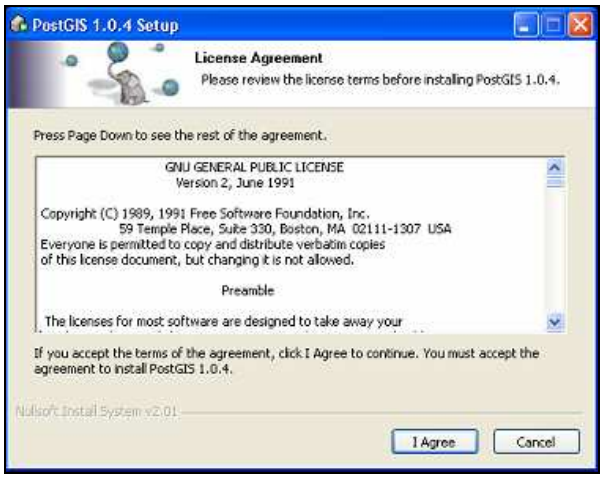
En la configuració del servei, s'establirà un compte d'usuari que no existeixi i una contrasenya. Aquest usuari no pot tenir permisos d'administrador. En aquest cas escrivim "postgres" tant en usuari com en contrasenya, però es pot utilitzar un altre nom o un usuari ja existent. Si es tria un de nou, preguntarà si es vol crear un

de nou. Contestarem que si. També pot ser que ens informi que al contrasenya és poc segura i si volem que el programa creu una per nosaltres. Contestarem que no.



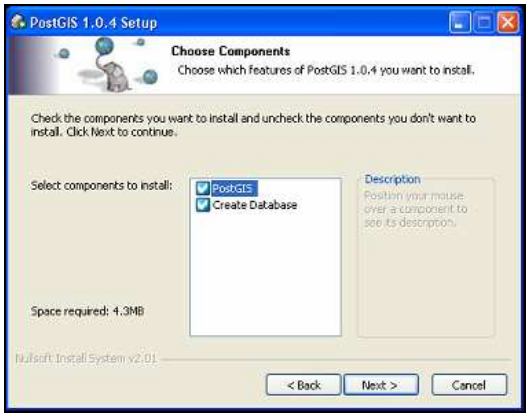
Imatge 104: Instal·lació *PostGIS*

En aquest moment ja es té instal·lada una part, però per accedir a la funcionalitat espacial (funcionalitat que ens permet, georeferenciar punts en un mapa) cal instal·lar PostGIS, que també es documenta en aquest document. La instal·lació de PostGIS s'efectua executant l'archiu PostGIS-setup-1.0.1-1.exe del directori "PostgreeSQL". Piquem I Agree.



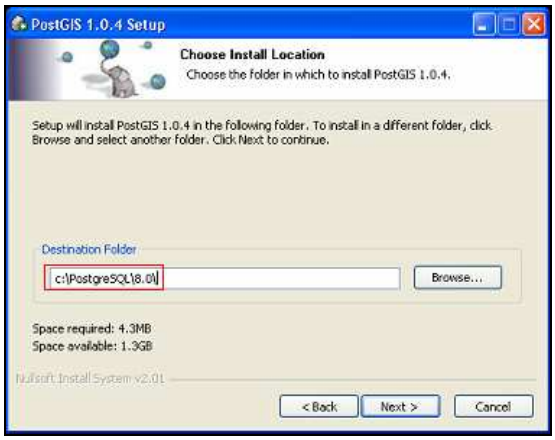
Imatge 105: Selecció d'opcions

Deixem les dos opcions seleccionades i piquem Next.



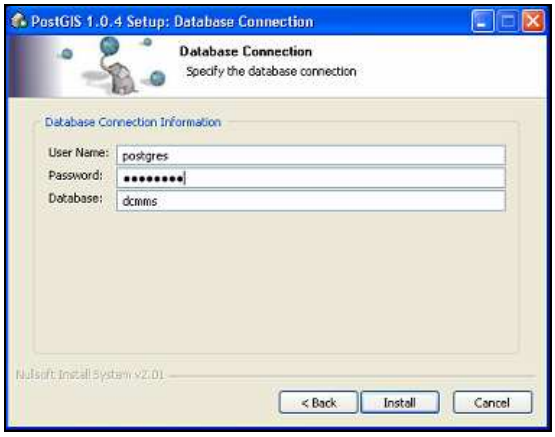
Imatge 16: Selecció del directori

Call assegurar-se d'elegir el directori on hem instal·lat PostgreSQL perquè per defecte ens apareix un altre.



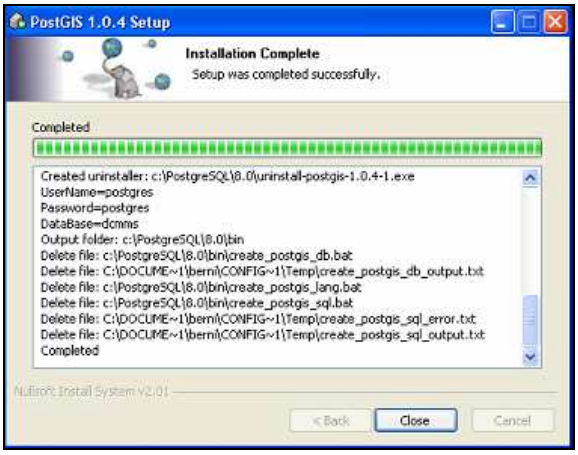
Imatge 207: Selecció del directori

A la pantalla Database connection escrivim "postgres" con usuari i contraseñilla i es deixa "dcmms" com a nom de la base de dades. Aquesta base de dades és una base de dades espacial que utilitzarem com a plantilla per crear les bases de dades.



Imatge 108: Connexió base de dades

Si tot va bé es veu la pantalla següent:



Imatge 109: Finestra d'intal·lació