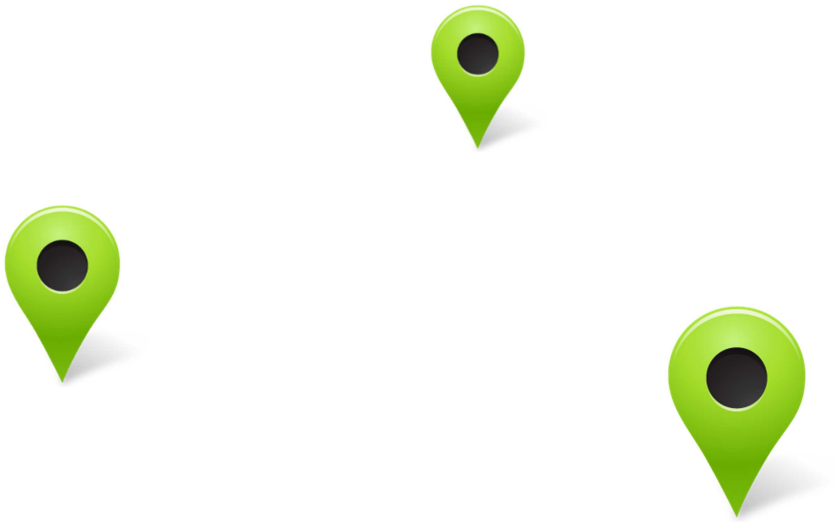
Sistema d'Informació Geogràfica

Per la localització i visualització d'espècies invasores al delta de l'Ebre



Projecte Final

Alumne: Adrià Jaime Torres

Màster en Tecnologies de la Informació Geogràfica, 15a edició

Febrer 2014

Thigis serveis ambientals

UAB Universitat Autònoma de Barcelona
Departament de Geografia

15 mtig 2013

ABSTRACT

A Catalunya, i concretament al delta de L'Ebre, existeix una problemàtica amb les espècies invasores, espècies que alteren els hàbitats naturals que colonitzen i generen impactes ecològics, econòmics i fins i tot poden arribar a afectar a la Salud humana. Aquest projecte pretén desenvolupar una eina per ajudar a controlar aquest problema. Es tracta d'un Sistema d'Informació Geogràfica que compta amb una aplicació mòbil, que amb ajuda de GPS i càmera dels dispositius Smartphone va enregistrant a partir de fitxes omplertes per l'usuari a una base de dades SQLite. Aquesta Informació serà exportada i tramitada per ser inserida a un servidor de mapes on-line (WMS) a partir de GeoServer. Un cop publicats aquests registres de l'aplicació amb la seva corresponent geometria (puntual), seran visualitzats a un visor web, programat a partir de llibreries de JavaScript com OpenLayers i Geoext. Aquest visor serà interactiu, permetent a l'usuari dur a terme accions com recerca de coordenades, establir mesures com distàncies i superfícies i consultar la informació dels registres al clicar-hi a sobre.

***

In Catalonia, and specifically in Delta de L'Ebre , there is a problem with invasive species , species that alter the natural habitats which they colonize generating environmental and economic impacts, even can affect the human health . This project aims to develop a tool to help control this problem. It is a Geographic Information System that has a mobile application, which access to the GPS and Camera of the Smartphone Device and record sheets filled by the user about the specie he has found in a SQLite database. This information will be processed and exported to be inserted to a Web Map Service (WMS) created with GeoServer . Once these records has been published with the corresponding geometry (point) will be displayed in a web viewer , which has been programed with JavaScript libraries such as OpenLayers and GeoExt . This map viewer will be interactive, allowing the user to perform actions like coordinate research, establish measures such as distances and surfaces and consult the records alphanumeric data by clicking on the displayed geometry.

SISTEMA d'INFORMACIÓ GEOGRÀFICA PER LA LOCALITZACIÓ I VISUALITZACIÓ D'ESPÈCIES INVASORES AL DELTA DE L'EBRE

1. Introducció
   - 1.1 Breu explicació del problema de les espècies invasores
   - 1.2 Objectius de l'aplicatiu proposat respecte el problema
   - 1.3 Marc institucional

2. Anàlisi de requeriments

   2.1 App mòbil:

   Entorn de desenvolupament (PC, programació Java a partir de Eclipse, dispositiu real de probes)

   Dispositius destinataris (API Requerida, perifèrics dels que ha de disposar: GPS, càmera, memòria externa, etc.)

   2.2 Transformació de dades del mòbil al visor

   2.3 Visor:
   Programat amb javascript, connexió a servidor a partir de php. Cartografia (wms utilitzats carto. base, Geoserver)

3. L'app Mòbil:

   3.1 Disseny de l'aplicació, esquema d'activitats i la relació entre elles

   3.2 Funcionalitat.
   - 3.2.1 Base de dades SQLite. (taules i relació)
   - 3.2.2 Formulari, captació de dades(, tipus de dades i perifèrics del dispositiu mòbil utilitzats, càmera GPS, etc.
   - 3.2.3 El formulari Sistema CRUD (creació, eliminació, edició de registres) i visualització en *Listview*
   - 3.2.4 Inventari d'espècies
   - 3.2.5 Exportació de dades en XML i Base de dades Sqlite

   3.3 Mostra de l'aplicació

4. Visor Web

4.1 Procés de transformació de dades de l'app a fitxers visibles a un visor cartogràfic web

4.2 Muntatge de servidor web

4.3 El visor

4.3.1 Disseny de la interfície

4.3.2 Capes (*shapes* dels registres i cartografia base)

4.3.3 Funcions (selecció de registre i visió en *pop ups* de la imatge i informació corresponent georeferenciada sobre el mapa. Mesures de distàncies i àrees

4.4 Mostra de resultats il·lustrada amb *screenshots*

5. Notes Finals

6. Bibliografia

ANNEXES:

I. Mostra de codi comentada de classe Java per Android (Formulari.java) de l'aplicació ExoticApp (codi complet al CD adjunt)

II. Codi del document Visor_delta.html comentat

III. Programa Phyton complementari, per tractar arxius XML.

1. Introducció

Aquesta és la memòria que descriu els objectius principals i el desenvolupament del projecte dut a terme a l'empresa Thigis Serveis Ambientals SL i tenint com a potencial receptor del producte l'Institut de Recerca i Tecnologia Agroalimentària(IRTA). Projecte que consta d'un Sistema d'Informació Geogràfica que integra tecnologies de recollida de dades mòbil (aplicació per *Smartphones* Android) i la seva corresponent visualització en un visor a partir d'un servidor web. Una solució tecnològica que pot ajudar a la gestió i control d'un problema com és el de les espècies invasores.

1.1 El problema de les espècies invasores

- *Espècies Invasores*

Els diferent hàbitats estan poblats per espècies que han evolucionat aïlladament, separades per barreres naturals. L'activitat humana però, ha fet que aquests obstacles esdevinguin relatius i com a resultat, les espècies es desplacen ara cap a zones més enllà de les seves àrees de distribució natural.

Aquestes espècies que han estat traslladades, intencionadament o no, cap a zones on no s'esdevenen naturalment s'anomenen espècies introduïdes o exòtiques. D'aquestes, algunes no sobreviuen, altres prosperen i poden arribar a amenaçar la biodiversitat de l'ecosistema local i fins i tot ser causants d'altres tipus d'impacte, com ara econòmics (problemes amb conreus per exemple) o bé relacionats amb la salut humana. És en aquestes condicions quan se les coneix com a espècies invasores.

Algunes característiques comuns per a identificar aquestes espècies invasores són una elevada taxa de reproducció i creixement, capacitat de dispersió, flexibilitat fenotípica (capacitat per adaptar-se fisiològicament a nous entorns) i adaptabilitat alimentària.

- *La problemàtica que representen*

Les invasions per part d'espècies foranies poden alterar l'estructura i composició de l'ecosistema natiu directament, quan existeix competència pels recursos . També afecten indirectament mitjançant canvis en els cicles de nutrients i relacions ecològiques entre les espècies locals.

Per altre banda el trajecte evolutiu de les espècies natives es pot veure alterat per exclusió competitiva. Fins al punt de que la espècie invasora acabi per complet amb les espècies originàries de l'habitat que ha colonitzat.

En síntesi, els impactes que poden causar aquestes espècies es poden classificar en els següents àmbits:

Impactes ambientals

- Pèrdua de la biodiversitat.
- Canvis en la funció dels ecosistemes.
- Canvis en els cicles dels nutrients.
- Disminució de la qualitat de l'aigua .

Impactes en la salut i el benestar de les persones

- Paràsits i malalties.
- Disminució de les oportunitats per a les activitats de lleure.

Impactes econòmics

- Interferència amb els recursos biològics que fomenten les activitats pesqueres i aqüícoles, col·lapsant les poblacions d'espècies objectiu o afectant-les amb agents patògens.
- Trastorns al turisme.
- Danys a infraestructures (emissaris, ports, boies, canals de regadiu...)
- Costos de neteja i control.
- Costos de tractament i quarantena.

Impactes Culturals

- Competència amb les espècies natives cultivades per la subsistència
- Degradació d'habitats d'importància cultural.

- *Predicció i gestió*

Les espècies poden ser introduïdes Involuntàriament (al adherir-se al casc d'un vaixell per exemple) o bé de manera voluntària (espècies alliberades per augmentar la pesca local). O bé un punt entremig, és a dir, s'introdueixen amb alguna mena de contenció però acaben escapant.

En tot cas un cop identificades les rutes d'entrada, cadascuna demana unes mesures específiques, contemplant accions per controlar, mitigar i eradicar una incursió un cop s'ha produït. És en el terreny de la prevenció però en el que s'han de fer els majors esforços, i on aquest sistema integrat de d'aplicació mòbil i visor web que es proposa en aquest projecte pot ajudar. Facilitant la geolocalització dels punts més crítics, enregistrant fotografies georeferenciades des del dispositiu mòbil dels tècnics encarregats i oferint una visió territorial del problema en un marc geogràfic concret a partir d'un visor cartogràfic web interactiu.

1.2 OBJECTIUS

- Desenvolupament d'una Plataforma SIG que permeti la captura de dades sobre el terreny (Delta de l'Ebre en aquest cas) a partir de dispositius mòbils amb una aplicació que recull informació a base de formularis i la seva corresponent base de dades on emmagatzemar la informació recollida pels usuaris.

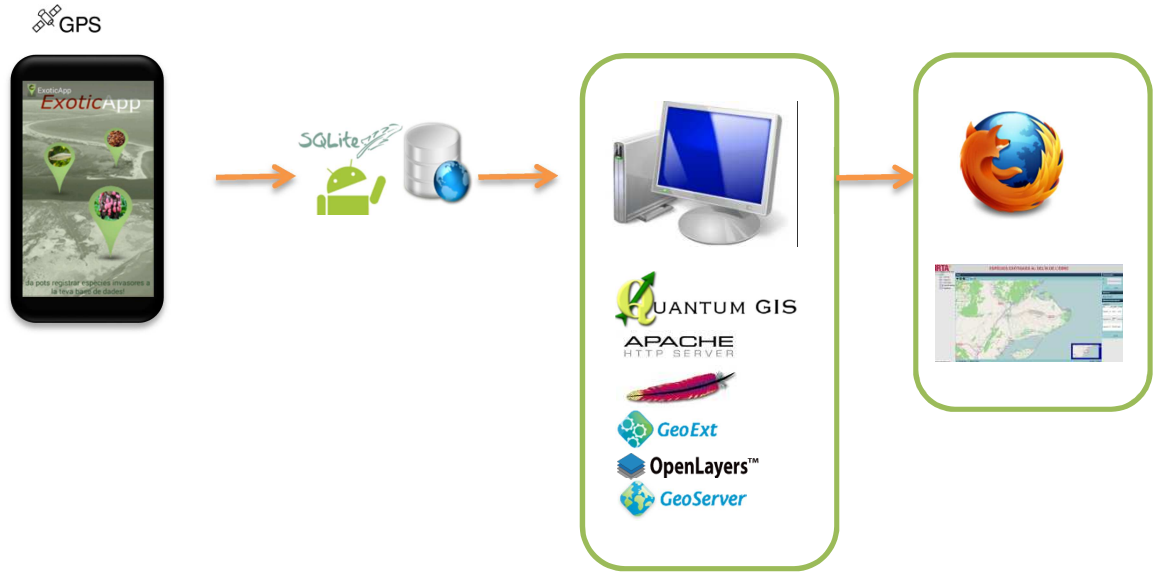
- Elaboració d'un visor web amb el qual poder visualitzar la distribució i la informació dels registres captats pels usuaris de l'aplicació mòbil, donant així un enfocament geogràfic a l'hora de determinar la problemàtica en el territori .

- Dotar als tècnics encarregats d'una eina que faciliti el control i la gestió d'un problema com és el de les espècies invasores, així com enriquir el fons documental amb un nou sistema d'informació geogràfica.

- En quant a l'abast territorial, es definirà en el marc geogràfic del Delta de l'Ebre, definint escales màximes i mínimes en aquest espai. Si s'escau el marc del visor es pot ampliar a tot el territori català.

- Desenvolupar el contingut coordinant-se amb l'IRTA, potencial receptor de l'aplicació respecte les espècies que s'han de tenir en compte, així com altres informacions que poden ser d'interès.

Per l'assoliment d'aquests objectius es proposa el següent sistema, combinant recollida de dades amb l'aplicació mòbil i la validació de dades i visor web que de moment, ha estat desenvolupat en un entorn local:



1.3 Institucions involucrades

Departament de Geografia de la Universitat Autònoma de Barcelona, com a organitzador del Màster de Tecnologies de la Informació Geogràfica(MTIG) en totes les seves edicions, des de la primera al 1998 fins l'actual (2012-2014). El màster és impartit al Laboratori d'Informació Geogràfica i Teledetecció (LIGIT) de la UAB.

15mtig 2013

Tutor assignat pel màster:

- **José Quirós Jiménez** : Geògraf i tècnic superior de suport a la recerca al LIGIT (UAB).

Thigis Serveis Ambientals S.L. Constituïda a l'octubre de 2011, aquesta societat limitada centra la seva activitat en els àmbits de l'eficiència energètica, l'educació/formació, el greening i els geoserveis. Aquesta darrera branca és la que rep l'oferiment del Laboratori d'Informació Geogràfica i Teledetecció per acollir el projecte final d'un dels alumnes del 15è Màster en Tecnologies de la Informació Geogràfica. Fet que s'estima des de Thigis com una oportunitat no comercial per a poder impulsar un projecte d'I+D. Els serveis que ofereix l'empresa es cataloguen en els quatre conceptes esmentats anteriorment:

Educació/Formació/Comunicació. Cursos i activitats de tota mena dirigits tant a centres educatius com empreses, sobre temes com la gestió de residus, estalvi energètic i energies renovables, contaminació atmosfèrica, mobilitat sostenible, recursos hídrics…

Eficiència energètica. S'ofereixen solucions per reduir l'impacte ambiental i energètic de cada projecte, oferint serveis com la certificació energètica d'edificis, plans personalitzats d'estalvi energètic: especialitzats en Pimes i centres educatius, auditories energètiques. Anàlisi i gestió de consums energètics, etc.

Greening: Assessorament per a empreses i serveis per a la seva millora ambiental en el context de la responsabilitat social corporativa.

Geoserveis Serveis que agilitzen la gestió, anàlisi i edició de les dades espacials, geo-referenciades o no, mitjançant el potencial que ofereixen tant les eines SIG com les plataformes tradicionals de producció cartogràfica. Ofereix serveis com captura i tractament de dades espacials amb eines estàndards i anàlisi i modelització espacials.

Tutor de l'entitat col·laboradora:

- **Xavier Serra Sellarès**: Oceanògraf, Tècnic Superior en Sistemes de la Informació Geogràfica i encarregat del departament de geoserveis de Thigis.

Institut de recerca i Tecnologia Agroalimentàries, és un institut d'investigació de la Generalitat de Catalunya, adscrit al Departament

d'Agricultura, Ramaderia, Pesca, Alimentació i Medi Natural, regulat per la Llei 04/2009 de 15 d'abril, del Parlament de Catalunya, que ajusta la seva activitat a l'ordenament jurídic privat.

El seu objectiu és contribuir a la modernització, competitivitat i desenvolupament sostenible dels sectors agrari, alimentari i aqüícola, al proveïment d'aliments sans i de qualitat per als consumidors i, en general, a la millora del benestar de la població

Aquesta entitat és la receptora final de l'aplicació que es desenvolupa en aquest projecte, encarregada també d'escollir i proporcionar la informació sobre les espècies invasores que es volen registrar en el marc geogràfic del Delta de l'Ebre.

2. Anàlisi de requeriments

A continuació s'explicarà quins han estat els requeriments tècnics per desenvolupar el projecte, i els que l'usuari haurà de satisfer per tal de poder utilitzar el producte final. És important assentar les bases de l'entorn de treball des del principi, fer un estudi de que és el que es vol fer i què es necessitarà, per tal de no trobar-se amb problemes al llarg del desenvolupament relacionats amb mancances de recursos o haver instal·lat versions de programari no adients.

2.1 Desenvolupament de l'Aplicació Mòbil

És una aplicació per Android, per tant és necessari instal·lar a l'ordinador un entorn de desenvolupament per aquest llenguatge (que és Java, amb una sèrie de llibreries pròpies per Android). En aquest cas s'ha optat per Eclipse. Però hi ha alternatives com Android Estudio. Ambdós són gratuïts i tenen prestacions similars.

- *Eclipse i JDK (Java Development Kit)*

Es pot descarregar desde la pàgina www.eclipse.org/downloads, compatible amb sistemes operatius Windows, iOS i Linux.
Concretament en aquest projecte s'ha utilitzat la següent Versió d SDK, que és el conjunt d'eines d'Android:

Versió: 3.7.2 Build id: M20120208-0800

(c) Copyright Eclipse contributors and others 2000, 2012. All rights reserved.

API minim 8 (Android 2.2)

API optima 17 (Android 4.2)

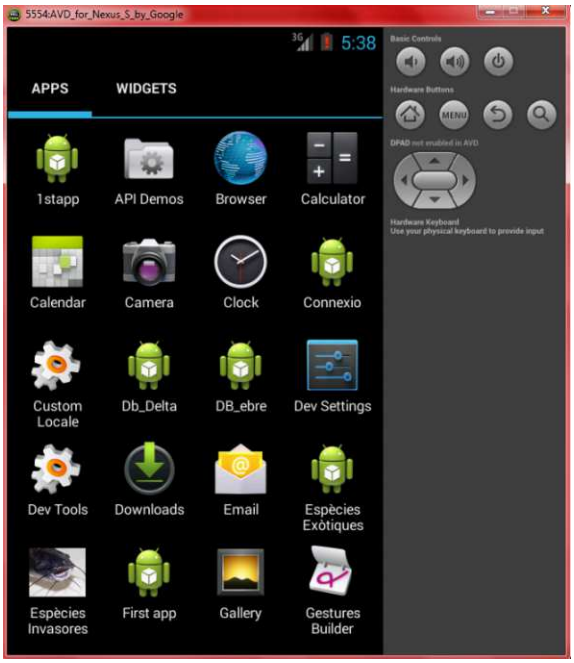
Un cop instal·lat l'executable que adquirim al descarregar el programa, és important seleccionar l'espai de treball. Per defecte s'ofereix una carpeta anomenada "workspace" que conté la carpeta personal de l'usuari. Podem especificar qualsevol altra ruta, i si la carpeta no existeix serà creada.

Hem d'entendre l'espai de treball com una carpeta que contindrà els projectes i paràmetres de configuració d'Eclipse. Es poden tenir múltiples espais de treball amb paràmetres o agrupacions de projectes diferents. En aquest cas s'ha treballat al *workspace* que es dona per defecte.

- *AVD (Android Virtual Device)*

Eclipse et permet desenvolupar AplicacionsEsense necessitat d'un dispositiu real. es tracta d'un dispositiu virtual, que emula un Smatphone. Es poden crear tants com siguin necessaris, emulant les característiques del tipus de dispositiu que es vulgui.

En el cas de tenir dispositiu real no és necessari. Però caldrà activar les opcions de desenvolupador de l'aparell. Convertint-lo en un dispositiu per depurar les aplicacions creades en Android i fer les proves necessàries.



Menú dispositiu virtual generat per Eclipse

En quant a l'usuari, necessitarà:

- un Smartphone, amb un sistema operatiu Android
- API mínima del Sistema operatiu Android 2.2, Nivel de API 8 (Froyo)
- El Dispositiu ha de comptar amb càmera, GPS i targeta de memòria externa
- 2 Mb de memòria per instal·lar l'aplicació
- Cable USB-Micro USB per exportar les dades al PC en format de base de dades, o bé en XML.
- Important: activar serveis de GPS a la configuració del dispositiu per el correcte funcionament de l'App.

2.2 Tractament de Dades

Programari necessari pel tractament de dades:

SQlite Administrator, Primer de tot un administrador de base de dades que sigui compatible amb sqlite i seguint la dinàmica del projecte, que també sigui de programari lliure que permeti exportació a XLS(coma separated vàlues). En aquest cas s'ha utilitzat el anomenat SQLite Administrator. Es pot descarregar a la pàgina *http://sqliteadmin.orbmu2k.de/.* Amb copyright del 2008, orbmu2k.

Els requisits per la versió 0.8.3.2 (versió beta) són els següents:

- És necessari un sistema operatiu Linux o Windows (versió mínima XP)
- Molt lleuger, 2 Mb de memòria necessaris.
- Programari lliure, per tant no representa cap cost, tot i que te copyright.

Les competències que te segons l'apartat de característiques de la pàgina web esmentada anteriorment són les següents:

- Crear / Modificar / Eliminar Taules per Wizard
- Crear / Modificar / Eliminar Índexs per Wizard
- Crear / Modificar / Eliminar Vistes per Wizard
- Crear / Modificar / Eliminar disparadors per Wizard
- Finalització Codi SQL que suporta àlies de taula
- Codi SQL Ressaltat
- Localització de SQL Error
- Importar dades des d'arxius CSV
- Exportar dades (XLS / CSV / HTML / XML)
- Consultes Botiga d'usuari en base de dades
- Cerca de Consultes d'usuaris
- Guardar imatges en "Blob Fields" (JPG / BMP)
- Mostra SQL de cada article de base de dades
- Migrar SQLite2 Bases de dades per SQLite3
- Intenta mantenir índexs després de modificar una taula.

Calc, és l'editor de fulls de càlcul de OpenOffice (també vàlid Microsoft Excel o altres). Programari lliure, descarregable a *http://www.openoffice.us.com/download-openoffice-free.php.*

- Compatible amb tos els sistemes operatius.
- Descarregar l'executable requereix 770 Kb
- Pot obrir documents d'altres editors com Excel.

Quantum Gis, és un SIG d'escriptori que permet generar Shapes a partir de CSV (sempre i quant l'arxiu tingui els camps adients (en els que hi hagi coordenades x,y).

Esta disponible per a MS Windows i Mac OS X. I Linux, es pot descarregar de manera completament gratuïta a *http://download.qgis.org.*

- Qgis es distribueix sota la Llicència Pública Genera (GNU)
- Versió utilitzada Quantum GIS 1.8.0(Lisboa)
- Memòria al disc necessària (Instal·lant el paquet Standard) 550 Mb

Udig , GIS d'escriptori orientat a publicacions a internet. Per generar estils per els *shapes*, exportables al servidor de mapes GeoServer) és necessari un format concret (sld).

- *User-friendly Desktop* Internet GIS
- Versió: 1.4.0b
- Aquest programa inclou software desenvolupat per Eclipse, Geotools i Apache Software Foundation.
- Memòria en disc necessària: 298 MB

Apache HTTP server, És Un Servidor web HTTP de codi obert, per plataformes Unix (BSD, GNU / Linux, etc.), Microsoft Windows, Macintosh i altres, que implementin el protocol HTTP/1.1. En el que es crearà un servidor local(localhost) per generar el visor web.

Es pot descarregar al següent enllaç *http://www.apache.org/dyn/closer.cgi.*

- Versió 2.2
- Servidor local (port 8080)

2.3 Visor Web

A continuació el programari i llibreries amb les que s'ha de comptar dins del directori en el que es prepari el servidor web, és a dir tots els següents elements hauran d'estar dins del directori d'Apache.

OpenLayers, és una biblioteca de JavaScript de codi obert sota una derivació de la llicència BSD per mostrar mapes interactius en els Navegadors web. OpenLayers ofereix accés a diferents fonts d'informació cartogràfica a la Xarxa: WMS, Mapes Comercials (tipus Google Maps, Bing, Yahoo), diferents formats vectorials, mapes d'*OpenStreetMap*, etc.

- Memòria necessària: 62,7 Mb.
- Versió 2.12

Geo Ext, serveix per complementar la llibreria Ext Js que és perfecta per crear aplicacions web, però falla en els mapes i informació epacial.

Per tant la solució és posar els mapes de OpenLayers en un panell Ext i embolicar-los OpenLayers en els components de GeoExt.

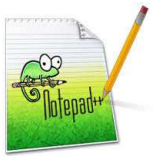
GeoExt combina els controls geoespacials de OpenLayers amb els components d'interfície d'usuari de Ext JS en un *framework* que ens permet construir aplicacions GIS d'estil similar a les d'escriptori, però en un navegador. Tot aquest programari és de codi lliure i es pot descarregar a la pàgina http://www.sencha.com/products/extjs/download. A www.geoext.org es pot trobar el complement per treballar amb Open Layers.

- Memòria necessària per ExtJs 59,7Mb + 2.6 Mb per GeoExt.
- Versió Geoext: 1.1

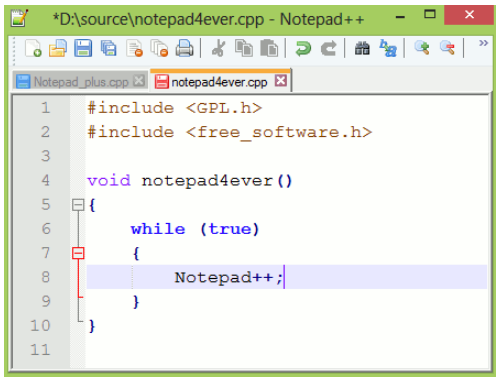
GeoServer, Es tracta d'un servidor de codi lliure escrit en Java. S'utilitza per compartir i editar dades geospacials.

Cal registrar-se per accedir a les seves funcions. Suporta un apli ventall de formats d'entrada, PostGIS, Shapefile, ArcSDE y Oracle. VFP, MySQL, MapInfo y WFS. Mentre que els formats de sortida (outputs) que ens permetran visualitzar la informació inserida son entre d'altres*, JPEG, GIF, PNG, SVG, Shapefile y GML*. En pot descarregar a *www.geoserver.org/display/GEOS/Download.*

- Versió utilitzada: 2.4.3 (versió estable, hi ha noves actualitzacions, però en fase beta)
- Memòria necessària: 65.5 Mb



Notepad ++, És un editor de text, en el qual s'hi escriurà tot el codi necessari. Hi ha innumerables editors, aquest és recomanable per el sistema d'ajuda visual que escriu cada part del codi en un color diferent en funció de les seves característiques, comptant amb aquesta ajuda per diferents llenguatges. Programa de codi lliure.



Es pot descarregar a la pàgina *http://notepad-plus-plus.org/download/*

- Versió: 6.3.2
- Memòria necessària: 11 Mb

Firefox, És un explorador web, de codi lliure dissenyat per Mozilla, una comunitat global que genera software públic i de codi obert.

Necessari per anar executant i depurant el codi que es genera amb el notepad++(en aquest cas). També compatible amb tota mena d'editors. Es pot descarregar a la pàgina de Mozilla *www.mozilla.org.*

- Versió 26.0
- Memòria 50 Mb

JavaScript, És un llenguatge de programació orientat a objectes, utilitza llibreries i s'utilitza en programació web. Tot i el seu nom, no està vinculat a Java, només n'adopta algunes convencions, però tenen una semàntica i un propòsit diferent. És el llenguatge a partir del qual es programarà el visor web.

- **PC amb el que s'ha fut a terme el projecte:**

Fabricant: Packard bell

Model: EasyNote TM86 (portàtil)

Processador: Intel i5CPU @2.27GHz // RAM : 4GB

Sistema Operatiu: Windows 7 (64 bits)

3. EXOTICAPP , l'aplicació mòbil per captar i gestionar registres d'espècies

En l'apartat 2.1 s'ha pogut veure quin és l'entorn de treball per desenvolupar l'aplicació. Ara es procedirà a detallar la creació del projecte elaborat en Eclipse, anomenat "Invasores". Primer de tot s'analitzarà el conjunt d'arxius que formen un projecte Android, que en el cas d'aquest projecte és el paquet anomenat "com.exemple.invasores".

Es tracta d'una aplicació basada en un sistema de base de dades SQLite i un formulari per anar omplint amb les dades que l'usuari vagi captant al fer treball de camp, fent us de càmera i Georeferenciació. Compta també amb un inventari per obtenir informació de les espècies a l'hora de fer treball de camp per si hi ha dubtes al trobar-te amb una. Per això s'han dissenyat una sèrie d'activitats amb interfície, es a dir, visibles per l'usuari, i una sèrie de classes auxiliars que són les que permeten el correcte funcionament de l'aplicació.

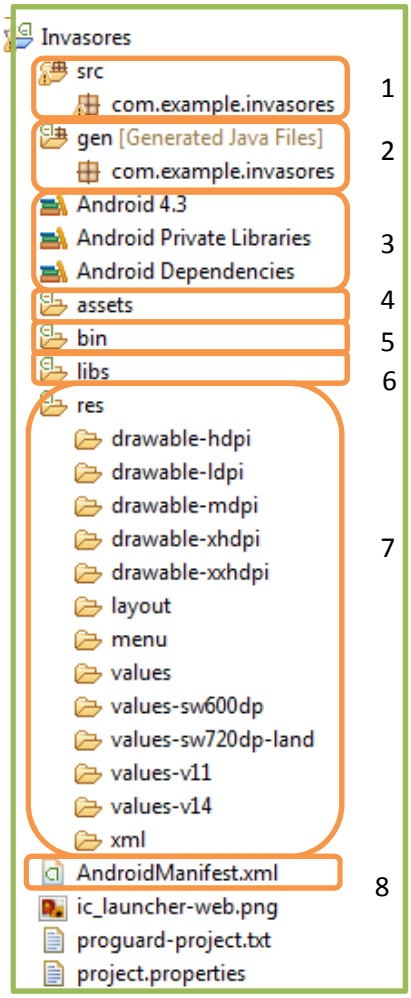


Figura 1, Fitxers Projecte Android

1. *carpeta src* : és la crpeta on es diposita per defecte tot el codi Java, per tant les classes que es vagin generant i que es veuran més endavant aniran col·locant-se en aquest directori.

2. *Carpeta gen:* és una carpeta que genera arxius Java automàticament, a diferència de ser, aquesta no és convenient modificar-a, ja que conté l'arxiu R que va indexant i identificant tots els objectes que es vagin creant a src.

3. *Llibreries Android:* són les llibreries d'Android, les que s'han descarregat desde el SDK, tot i que compatible amb les anteriors, es mostra la de l'API que s'ha establert com a principal receptor de l'aplicació que s'esta desenvolupant.

4. *Carpeta Assets:* aquí es dipositen els arxius que actuen com a recursos i que seran compilats a l'hora de generar l'arxiu final (APK) com ara bases de dades, arxius XML... però no seran indexats ni integrats a l'aplicació, el que fa el seu ús més ineficient i el seu accés és més laboriós.

5. *Carpeta bin:* Carpeta que es genera automàticament i a la que tampoc s'ha d'accedir, es la encarregada de generar la compilació del arxiu final apk.

6. *Carpeta lib:* aquí s'hi guarden les llibreries externes, les que no són pròpiament Android. És a dir que s'hi troben documents *.jar*. com per exemple les llibreries d'Open Street Maps per Android, anomenada Osmdroid.

7. *Carpeta res:* És la carpeta on hi van tots els recursos que utilitzarà l'aplicació. Es la més complexe i es subdivideix en varis subdirectoris, en funció del tipus de recurs. Al contrari de l'anterior, el contingut d'aquesta carpeta si que és indexat a l'arxiu R esmentat al punt 2.

- Existeixen també els *drawables*, on s'hi insereixen els arxius d'imatge (en funció de la qualitat hdpi(alta), mdpi(mitja), lmdi(baixa) .
- A la carpeta *Layout* hi van els XML de d disseny de les activitats que veurà l'usuari. Han d'anar vinculats a alguna de les classes de la carpeta *src*, on son els arxius Java en els que s'hi programa la seva funcionalitat.

- A la carpeta menú, també s'hi depositen arxius xml, en aquest cas els que determinen el dissenys dels menús d'opcions i contextuals que es vulguin a l'aplicació (també programats a la carpeta *src*)

- A la carpeta *Values* hi tenim els *Strings*, cadenes de text que volem que surtin a l'aplicació per tots els elements (com *textViews*, botons, dialges...) amb un id assignat. També altres components com estils i dimensions.

8. *Android Manifest:* És l'arxiu resum de l'aplicació, en la que s'han de declarar totes les activitats (vinculades a un *Layout*). També s'hi defineixen les característiques del projecte, com el nom de l'API mínima i la òptima i els permisos d'usuari per accedir a les funcions del telèfon en aquest cas els permisos necessaris son els següents:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
```
(Permís per creardocuments a la tageta SD)

```
<uses-permissionandroid:name="android.permission.READ_EXTERNAL_STORAGE"/>
```
(Permís per llegir documents de la targeta SD)

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```
(Permís per accedir al GPS)

```
<uses-permission android:name="android.permission.INTERNET" />
```
(Permís per accedir a internet, en el cas de no poder accedir al GPS s'establiran les coordenades a partir de Wi-fi, o la última posició coneguda si tampoc es pot accedir a aquesta xarxa)

3.1 Disseny Activitats

Les activitats marcades al gràfic són les d'interacció amb l'usuari, les que aquest podrà veure, mentre que la resta són les encarregades de gestionar les bases de dades. En Android la funció encarregada de relacionar activitats són els *Intents*. Aquests intents ens permeten passar d'una activitat a l'altre, declarant-los en botons, menús, *ListViews* i qualsevol objecte que ho permeti.

L'objectiu principal, és captar dades a partir d'un formulari, així com poder accedir a l'inventari abans d'anar al formulari, així doncs les interfícies que l'usuari veurà i la seva relació és la següent:

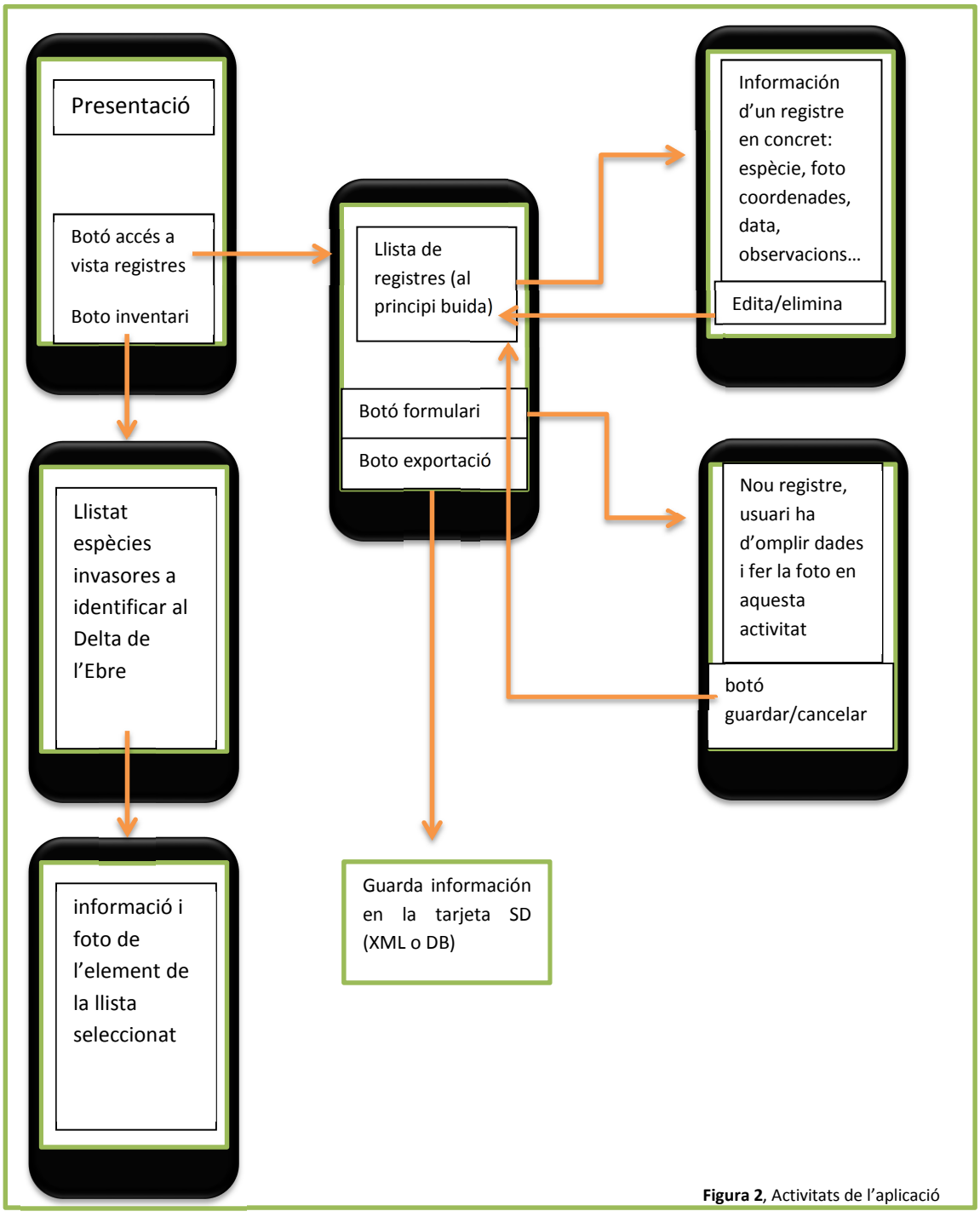
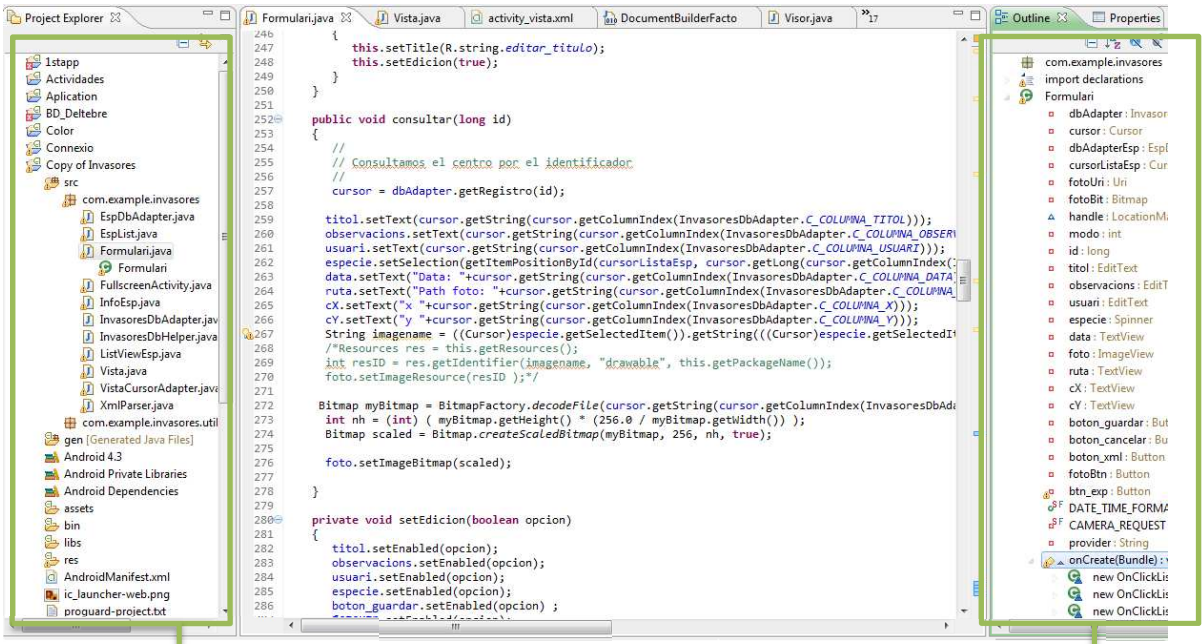


Figura 2, Activitats de l'aplicació

3.1.1 La programació de les activitats

En l'entron d'Eclipse, i per tant en Android, el llenguatge de programació és Java, un llenguatge de propòsit general, orientat a objectes i basat en classes. Aquesta estructura en classes facilita que un cop programada l'aplicació, pugui se r executada des de qualsevol dispositiu amb més facilitat i no haver de dependre de factors externs.

Eclipse compta amb un editor Java que proposa una sèrie de facilitats:



Un arbre que mostra totes les classes que s'han anant creant, a part de tots els arxius que són necessaris pe tal de dur a terme l'aplicació, explicats anteriorment(punt 3) , des d'on es poden agregar, copiar i eliminar arxius .

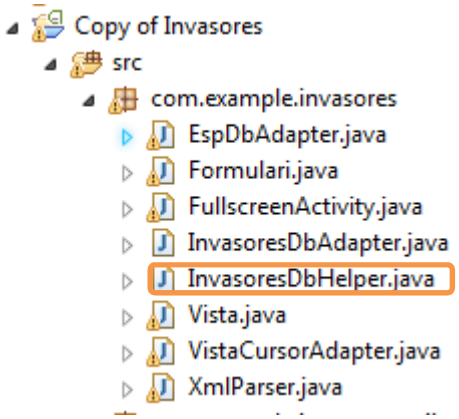
Aquest menú lateral mostra tot el que s'ha anat generant en la classe sobre la que s'estigui treballant, objectes, *listeners, intents*... Permetent una fàcil navegació a través del codi .

Figura 3, Entorn de programació Java a Eclipse

3.2 Funcionalitat

En aquest punt es detalla quin paper juga cadascuna de les classes que formen part de l'aplicació. Darrere de les activitats que es mostren en l'esquema del punt 3.1, hi ha tot una sèrie de funcions i classes auxiliars que simplifiquenla interfície en la que l'usuari intervé. Mentre l'usuari només genera i modifica registres, l'aplicació va construint una base de dades, accedint a perifèrics del dispositiu com la càmera i el GPS. També s'especifiquen una sèrie de cursors per alterar les dades i genera automàticament arxius en format DB (data Base) i XML(*eXtensible Markup Lenguage*) .

3.2.1 SQLite



S'ha comentat anteriorment que l'aplicació es basa en SQLite. A continuació es detallen les característiques i implementació d'aquest sistema. el motor de SQLite no és un procés independent que necessiti un programa de gestió de base de dades amb el que el programa es comunica. El que caracteritza la biblioteca SQLite es que s'enllaça amb el programa passant a formar part del mateix. Per això es requereix d'una classe que faci de gestor de la base de dades.

Android compta amb una classe anomenada *SqliteOpenHelper*. Per Treballar amb SQLite sempre s'ha de fer servir una classe que derivi d'aquesta, en ella s'hi crearan les taules i es faran els comandaments de creació actualització de les taules i els camps corresponents. Es a dir és la classe que fa de gestió de la base de dades.

En el cas de l'aplicació que es duu a terme en aquest projecte, es tracta de la classe *InvasoresDbHelper*. Classe en la que es genera la base de dades que necessitem a partir de comandaments SQL .

Les taules que es generen són la taula CITA, una taula que estarà buida i que s'anirà omplint amb les dades que l'usuari insereixi al formulari de l'aplicació. Un altre taula que actuarà de diccionari, on estarà tota la informació de les espècies entre les que l'usuari ha d'escollir una anomenada ESPÈCIES. Aquestes taules estaran relacionades a partir del camp id de l'espècie (esp_id).

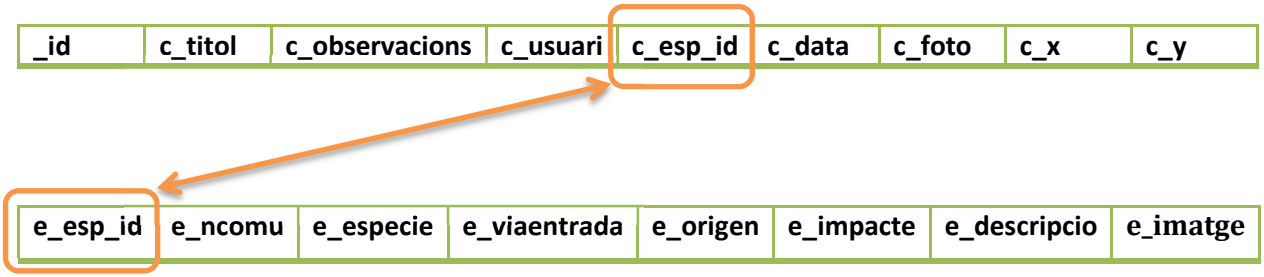
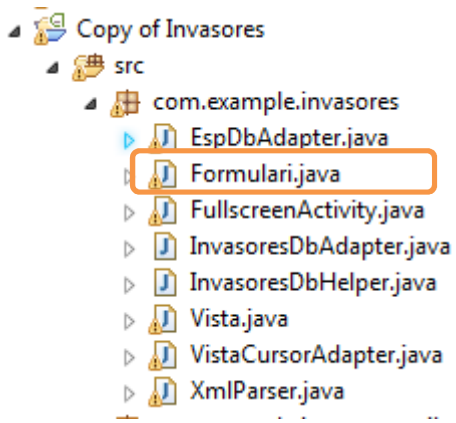


Figura 4, Relació entre taules I camps SQLite

3.2.2 El formulari, captació de dades.



La classe formulari és la que s'utilitza en el treball de camp , és la que permetrà a l'usuari anar omplint la base de dades**.** Hi ha informacions que ha d'omplir manualment l'usuari i d'altres que es generaràn automàticament el guardar el formulari com a registre.

Aquesta mateixa activitat servirà, com es veurà més endavant per visualitzar i editar registres

Informació que omplirà l'usuari:

- *Títol registre*: un camp 'edit text', en el que s'escriurà un títol que serveixi per identificar el registre de manera que l'usuari cregui convenient

- *Espècie* : a partir d'un objecte anomenat *Spinner* (un llistat extensible). Al clicar s'obre l'objecte sortiràn totes les opcions possibles d'espècies que interessa identificar. (accedeix a la taula diccionari per mostrar el nom de les espècies)

- *Observacions:* un camp 'edit text', en el que l'usuari escriurà si s'escau, les les observacions que cregui necessàries

- *Botó Foto:* Botó encarregat d'obrir la càmara del dispositiu, un cop feta la foto es mostrara una vista prèvia al mateix formulari a apartir de l'objecte "ImageView".

  La càmara s'inicia amb una cosntant anomenada ACTION_NEW_PICTURE. Obre l'intefície que el dispositiu té per defecte per fer fotos, al donar a guardar, desa la foto a la galeria també per defecte. En aquest cas s'ha programat un Image View que recupera la ruta de la foto i en genera una vista prèvia al formulari.

Informació que s'adquireix automàticament:

- *Data:* es connectarà amb el rellotge del dispositiu per adquirir la data i la hora en la q ue s'ha fet la foto

- *Coordenades:* S'adquereix a partir de de la crida d'un servei que ofereix la llibreria android , anomenat LOCATION_SERVICE. Configurat en aquest cas per que agafi la més pressissa possible, és a dir GPS. En cas de no poder accedir-hi la *Network* triangula a partir de wi-fi.

- *Ruta Foto:* genera un sting amb la ruta i nom de la foto, per tal de poder accedir-hi a l'hora de visualitzar els registres i localitzar les fotos que es vulguin exportar.

- *Id registre:* s'assigna un id numèric únic a cada registre que es vagi guardant

- *Botonera Guardar/Canclear*: El botó guardar, el que está fent és llegir la informació obtinguda als edit text i la generada automàticamnet, convertir-les en strings i inserir-les a la taula CITA, que s'anirà actualitzant en funció dels registres que es vagin guardant , editant i eliminant. El botó cabcelar, retorna a l'activitat prèvia a l'activació del formulari. Aquí va una mostra de l procediment utilitzat per registrar un dels camps(observacions):

```
reg.put(InvasoresDbAdapter.C_COLUMNA_OBSERVACIONS, observacions.getText().toString());
```

3.2.3 El Formulari, sistema CRUD

CRUD es refereix a les funcions bàsiques de gestió de base de dades (*Create, Read, Update, Delete*). Per fer això, es podria haver creta un altre activitat, però en aquest cas per optimitzar el rendiment de l'aplicació, s'utilitzarà la mateixa que per la captació de dades. A partir d'un sistema de modes.

En funció del mode en el que s'iniciï la classe formulari, tindrà una utilitat o altre. Compta amb els modes Crear, Visualitzar, Editar i Eliminar.

Create (crear): S'activa el formulari amb els camps buits i modificables, és el mode que s'utilitza per dur a terme la captació de dades explicada al punt 3.2.2.

Read(Visualtizar): El formulari accedeix a la informació del registre seleccionat, mostrant la informació de cadascun dels camps. Els que són de text és senzill, a partir d'un cursor que recorre la base de dades fins al registre seleccionat , el converteix en *string*(cadena de caràcters) i el mostra en el mateix camp en el que s'havia inserit en el mode 'Crear'.

La foto requereix un procediment més complexa. A la base de dades no es guarda la imatge en si, si no la ruta d'ubicació de la foto al dispositiu (URI). Per visualitzar-la es genera una imatge (bitmap) a partir de l'objecte bitmap Factory . Donant la possibilitat així de reproduir-la a escales de qualitat inferiors per garantir compatibilitat i optimitzar el rendiment.

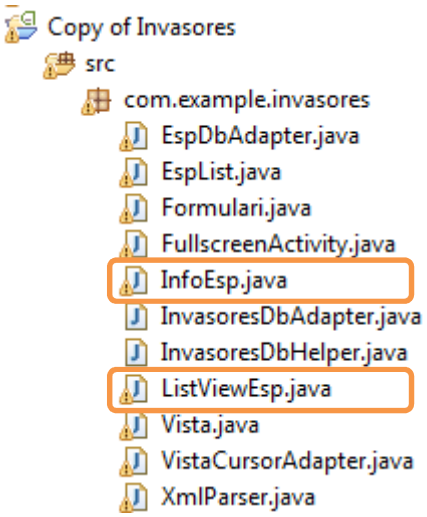
En aquest mode s'oculten els botons, i es bloqueja la possibilitat de modificar els camps 'edit Text'.

Update (editar) : el procediment és el mateix que el de visualitzar, però la opció d'edició que estava inactiva en el mode anterior (false) . En aquest cas s'ha activat (true). Això és possible generant una funció booleana en la que s'ha declarat cadascun dels elements que registren camps a la base de dades.

Delete (Eliminar): aquest no és un mode visible, simplement, seguint el mateix procediment, en comptes de mostrar-la en el formulari, elimina la informació del registre seleccionat.

3.2.4 Inventari d'espècies

Aquest apartat tracta la part de l'aplicació que fa les funcions de diccionari. Més de caire didàctic, amb informació de totes les espècies, es tracta de dues activitats, una en la que es mostra el llistat d'espècies que es volen detectar en el treball de camp i una altra que mostra informació específica de l'espècie seleccionada.



A l'activitat del llistat s'accedeix a la taula "Espècies" de la base de dades, i es genera una llista (*ListViewEsp*) a partir del camp "nom comú", mostrant un total de 10 espècies:

- Perca de Riu
- Cargol Poma
- Canya comú
- Cloïssa asiàtica
- Carpa comú
- Gardí
- Rasbora
- Cranc roig americà
- Musclo zebrat
- Silur

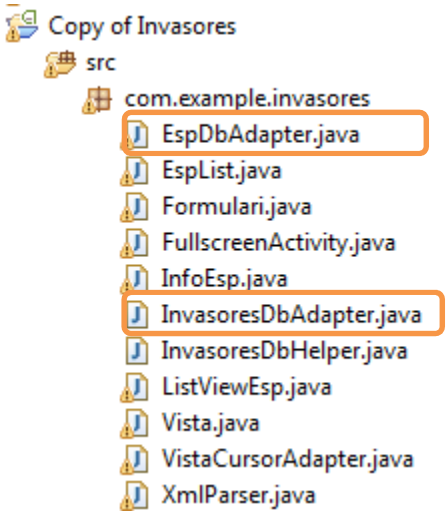
Al fer clic sobre qualsevol d'aquests elements de la llista s'obrirà una nova activitat (InfoEsp) que mostrarà la informació del registre en concret de la taula Espècies. Seguint el mateix procediment que en el cas de el mode "Visualitzar" del formulari explicat en el punt anterior (3.2.3). En aquest cas però les fotos estan emmagatzemades a la carpeta res, que com s'ha vist al punt 3 (carpeta 7) és on es guarden els recursos. És recomanable no guardar les fotos en la mateixa base de dades per optimitzar el funcionament, així que el que s'ha guardat al camp "imatge" és el nom de la imatge.

Cal esmentar aquí el sistema seguit per poder accedir a les fotos. Que ha estat diferent que en el cas del formulari (en aquest cas les imatges estan al paquet de la pròpia aplicació. El procediment normal per accedir a les imatges és senzill, es crida la funció "getResources.drawable.nom_arxiu", però en aquest cas la selecció de imatges va relacionada amb un nom guardat a una base de dades i aquest procediment no permet utilitzar variables. I per tant no es pot utilitzar un cursor per buscar el nom a la base de dades. En aquests casos, existeix el mètode "getIdentifier()". Que permet ficar-hi un *String* en el s'hi guarda la informació recollida pel cursor que recorre la base de dades.

La informació en aquesta taula, és inserida automàticament des de la primera vegada que s'obre l'aplicació. A partir de sentències SQL programades a la classe InvasoresDbHelper:

```
db.execSQL(" INSERT INTO Nom_Taula (Camp1, camp 2...camp N) VALUES (valor 1,valor 2..valor N");
```

3.2.5 Classes auxiliars:



Les classes auxiliars són aquelles que serveixen per garantir el funcionament intern de l'aplicació, classes amb les que l'usuari no interactua ni visualitza. Es el cas dels adaptadors, no és necessari generar una classe, es poden generar a cadascuna de les classes, però com s'utilitza en moltes de les activitats, per optimitzar el codi, s'ha generat una classe per cada una de les taules. I es crida des de les diferents activitats.

Un adaptador serveis per llegir i escriure en les bases de dades, en aquest cas generant variables String en les que emmagatzemar i moure les dades de la base de dades.

En aquest cas es compta amb dos adaptadors, un per la taula Cita (InvasoresDbAdapter) i un per la taula Espècies (EspDbAdapter). L'estructura és la mateixa en els dos casos, canvia la informació dels camps i el nom de les variables, evitant que siguin iguals en les dues classes per evitar confusions.

El funcionament dels adaptadors és el següent:

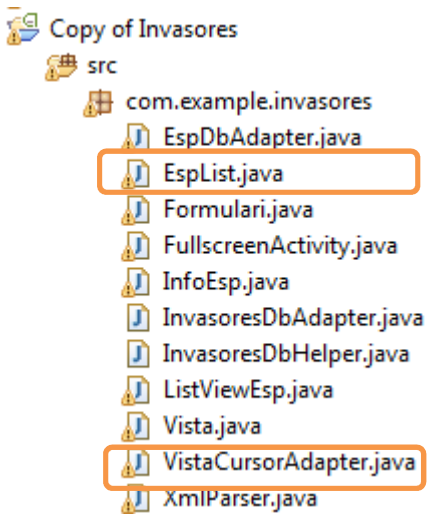
- Cal definir variables String amb el nom de la taula i de cadascun dels camps. És a dir, si un dels camps de la taula Cita és c_títol, es declara una variable com segueix:
  public static final String C_COLUMNA_TITOL = "c_titol";

- Es defineix una llista amb de les columnes de la taula per utilitzar-la en les consultes de la base de dades.

- Es defineixen els cursors, de manera que només caldrà cridar-los. Un cursor que retornarà la informació que hi ha en totes les columnes de la taula, s'ha anomenat '*getCursor*'. De manera que quan s'hi vulgui accedir des d'altres classes s'haurà de declarar *InvasoresDbAdapter. getCursor*. Un altre cursor que s'ha anomenat *getRegistre*, que retorna la els valors de totes les columnes d'una sola fila.

Això ens permetrà realitzar funcions com obtenir la llista de tots els registres, basats en el camp que es desitgi, en aquest cas a partir del títol i la data. I que un cop tinguem aquest allista, es pugui visualitzar la informació de cadascun dels registres de manera individual.

Les següents classes són cursors que permetran la visualització de les llistes, per una banda de la llista de registres que s'anirà omplint des del formulari (VistaCursorAdapter) i per l'altra banda la llista que mostra el conjunt d'espècies que l'usuari podrà registrar al formulari (EspList).



Son unes classes força senzilles, que generen una vista a partir de la connexió als adaptadors esmentats anteriorment. En aquesta classe es passa el cursor seleccionant el camp o camps que es volen visualitzar a la llista. En el cas de la llista de registres de l'usuari visualitzarem en cada ítem de la llista el títol que hagi posat i la data, mentre que en la llista de l'inventari es podrà observar el nom de la espècie.

Un cop el cursor està programat, cal que es defineixi com es vol mostrar la informació seleccionada. En aquest cas s'ha utilitzat un recurs d'Android, que dona n format determinat de *listview*. (android.R.layout.simple_list_item_1).

3.2.6 Exportació de dades

Un cop la sessió de treball de camp finalitza, hi ha la opció de exportar la base de dades en diversos formats.(XML i en Db de Sqlite3).

- ***XML:***

A partir de la taula omplerta per l'usuari a partir de registres, l'aplicació genera un arxiu xml on els camps de la taula CITA vista al punt 3.2.1, passen a convertir-se en etiquetes (tags) entre les quals automàticament s'inserirà la informació a partir d'un sistema anomenat DOM (*Document Object Model*). El model en forma d'arbre de nodes utilitzat és el següent:

```
<registres>
  <titolregistre>
    <id> </id>
    <titol> </titol>
    <observ> </observ>
    <usu> </usu>
    <espid> </espid>
    <data> </data>
    <foto> </foto>
    <cordx> </cordx>
    <cordy> </cordy>
  </titolregistre>
</registres>
```

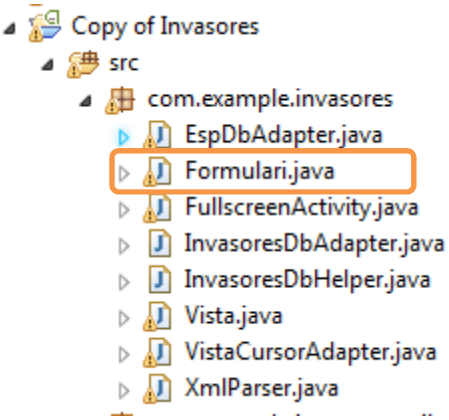
Figura 5, Estructura arxiu Invasores.xml

- <registres> és el conjunt de registres, és a dir l'equivalent a la taula.

- <titolregistre> és cadascun dels registres, és a dir les files de la columna, es repetirà tants cops com registres hagi emmagatzemat l'usuari.

- <id> i la resta de *'tags'* al mateix nivell, representen els atributs de cada registre, és a dir, són l'equivalent a les columnes de la taula.

El procés és fa a una classe auxiliar que s'ha anomenat XmlParser. En aquesta classe s'importa des d'una llibreria externa el DOM i tots els seus elements (parser, nodes...) :




```
import org.w3c.dom
```

Amb DOM, el document XML es llegeix completament abans de poder realitzar cap acció en funció del seu contingut. Això és possible gràcies al fet que, com a resultat de la lectura del document, el parser DOM torna tot el seu contingut en forma d'una estructura de tipus arbre, com s'ha vist abans, on els diferents elements de l'XML es representa en forma de nodes i la seva jerarquia pare-fill es s'estableix mitjançant relacions entre aquests nodes. De manera més gràfica, la distribució del xml generat és la següent:

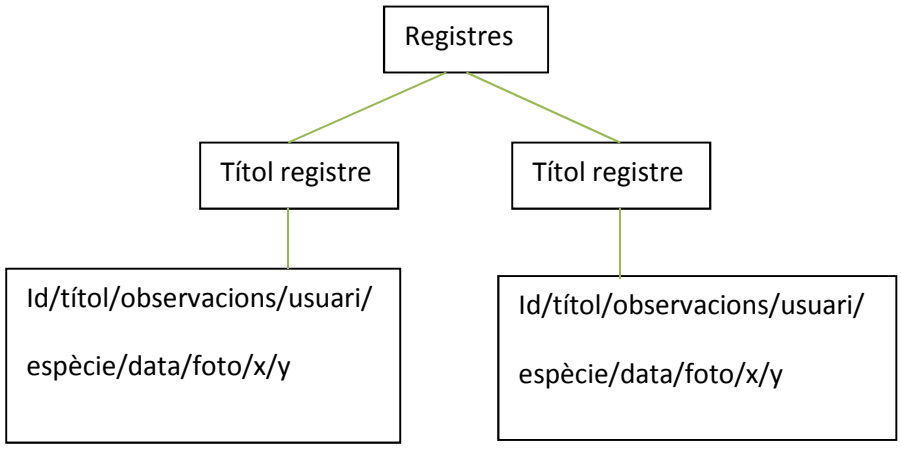


Figura 6, gràfic de la distribució dels elements de l'arxiu XML

En aquest cas però, no es llegeix d'un arxiu Xml, si no d'una base de dades que es converteix en xml. Per fer això es necessita un cursor que recorri la informació de la base de dades, adquirint la informació de cadascun dels camps. A partir d'aquest cursor, es generarà un fitxer xml, que s'anirà omplint utilitzant la estructura d'arbre fixada en el document que s'ha vist anteriorment, i copiant entre les etiquetes la informació del camp de la base de dades vinculat a les mateixes.

A partir de l'element Document Factory, es fa que aquest arxiu Xml omplert amb la informació de la base de dades, es guardi a un bloc de la memòria del dispositiu que sigui d'accés públic (la memòria externa que tingui configurat el dispositiu per defecte) de manera que l'usuari pugi extreure l'arxiu Xml al seu ordinador per poder treballar amb ell.

- ***DB SQLITE***

L'aplicació dona l'opció d'exportar la base de dades directament també. Com s'ha vist tota l'aplicació mou dades de la Base generada amb Sqlite (3.2.1). Aquesta base de dades és intransferible en un principi per l'usuari, ja que s'emmagatzema a un directori al que no es permet accedir ni des de el dispositiu ni al connectar el dispositiu a un ordinador.

Per extreure aquesta base de dades, el que s'ha fet és fer-ne una còpia i guardar aquesta còpia a la memòria externa que el dispositiu tingui configurada per defecte. Aquest procés és més senzill, no cal una classe auxiliar, amb una funció és suficient.

- ***DMMS***

Es tracta d'una perspectiva d'Eclipse (Dalvik Debug Monitor Server) que permet gestionar els dispositius amb els que s'està depurant, les aplicacions actives i tots els arxius del dispositiu. Des de dispositius reals, hi ha accés bloquejat a les dades de les aplicacions, que és la carpeta /data>data>"nom app".com. És per això que es fa la còpia a carpetes a les que si es permet l'accés. En aquest cas /storage.

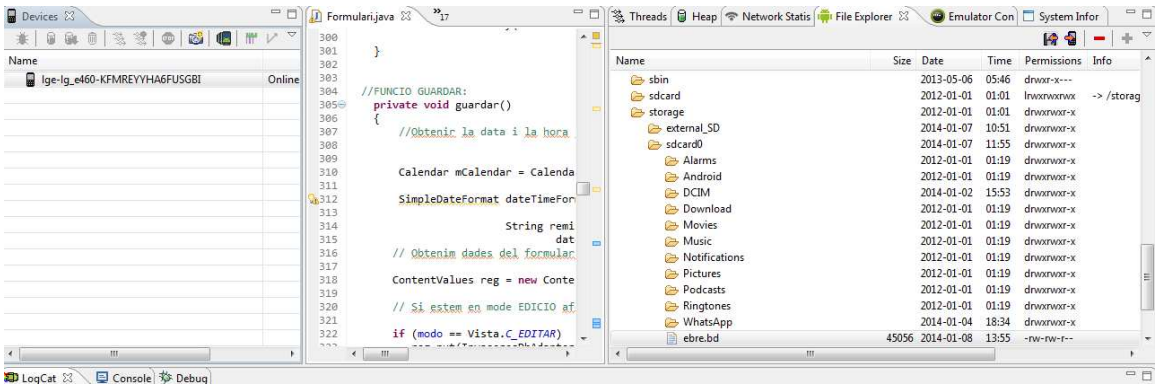


Figura 7, Entorn de control de dispositius i arxius d'Eclipse (DDMS)

Des d'aquí es pot comprovar si la informació s'ha exportat a la ubicació correcte, i en el cas de que així sigui, exportar-ho directament a l'ordinador amb el botó "*pull file from the Device*" , representat així  . O bé a l'inversa, introduïr un arxiu de l'ordinador al dispositiu amb el botó "*Push a file onto device*"  .

3.3 Els Layouts, mostra del resultat final

Les classes que formen part d'una activitat, han d'estar vinculades a un layout. Un layout és un xml en le que es dissenya la interfície de l'activitat, s'hi declaren els objectes que s'utilitzen en els documents '.java' com ara botons, *Text Views*, *Spinners*(ComboBox d'Android) , entre d'altres. (punt 3.2).

Aquests *Layouts* es guarden a la carpeta res (punt 3, directori 7). En una subcarpeta anomenada *Layouts*. S'han de guardar allà per què Eclipse dectecti correctament tots els id que s'asignen als objectes que es vulguin posar a l'interfície.

Eclipse compta amb un editor gràfic per facilitar el disseny dels layouts, tot i que es pot fer directament per codi XML. Des d'aquest editor es té accés a tots esl objectes (menú esquerre) i a les propietats de cadascun d'aquests (menú dret) mostrats a la imatge següent:

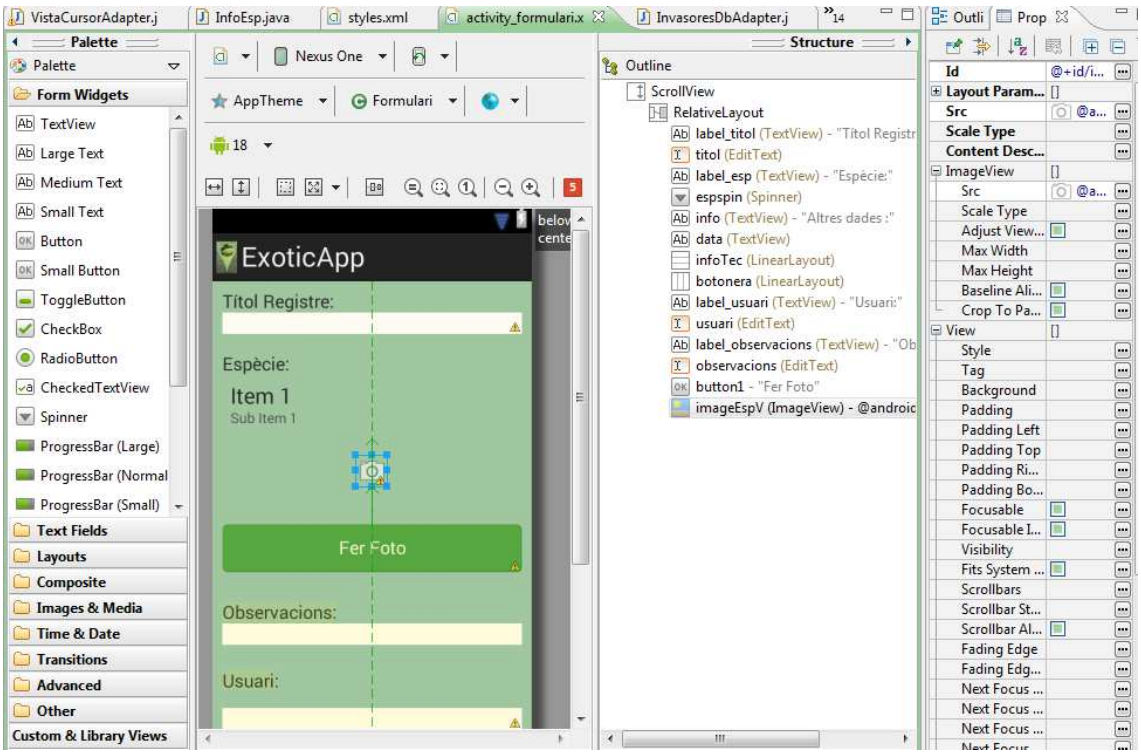


Figura 8, Entorn gràfic de disseny de layouts d'Eclipse

A tot objecte que es posi en el *Layout* se li assigna un id automàticament que es registra a l'arxiu "R.java" que és el que gestiona tots els id als que s'accedeix des de les classes vinculaades als layouts.

Hi ha diverses estructures de *Layout* en funció de la relació que es vol que tinguin els objectes que el conformen. En aquest projecte s'han utilitzat dos tipus:

- Linear Layout: Situa un objecte derrere de l'altre en posició vertical o horitzontal. Resula util per generar llistes o organitzar botons .

- Relative Layout: És més complexe, es basa en la relació entre objectes del contenidor, és a dir fer que l'objecte X estigui sota l'objecte Y, i a l'hora alineat horitzontalment amb l'objecte Z.

- **Pantalla de presentació.**

Es tracta del layout anomenat FullScreen, és la primera pantalla que l'usuari visualitza, una pantalla de presentació amb el nom de l'aplicació, un disseny de fons relacionat amb la funció de l'aplicació. Des de la que es pot accedir al inventari o bé al aformulari.

Figura9, Pantalla de presentació i botons extensibles al tocar-la

Al tocar la pantalla, es despleguen des de la part inferior de la pantalla dos botons, "Accedir a formulari" et porta a la pantalla vinculada al ListView en el que hi ha els registres de l'usuari (taula Cita de la base de dades) i "Inventari Espècies" et porta al ListView de la taula diccionari (Especies)

- **Inventari espècies**

Al prémer el botó Inventari Espècies, s'activarà el Intent punt 3.1) que trasllada a l'usuari a la següent activitat . Es tracta d'una ListActivity, en la que es mostren totes les espècies que es poden enregistrar.

Quan una llista (o qualsevol altre layout) és més llarga del que permet visualitzar una pantalla, cal inserir en el XML de disseny un objecte anomenat ScrollView, que permet que al arrossegar el dit sobre la pantalla amunt i avall aquesta vagi mostrant tota l'activitat.

Al prémer sobre qualsevol ítem de la llista, s'activarà un altre layout, que és el que permet visualitzar la informació d'una espècie en concret, es tracta d'un conjunt de TextViews (diferenciats per mida els estàtics i els dinàmics, que varien en funció del registre) i una imatge(ImageView).

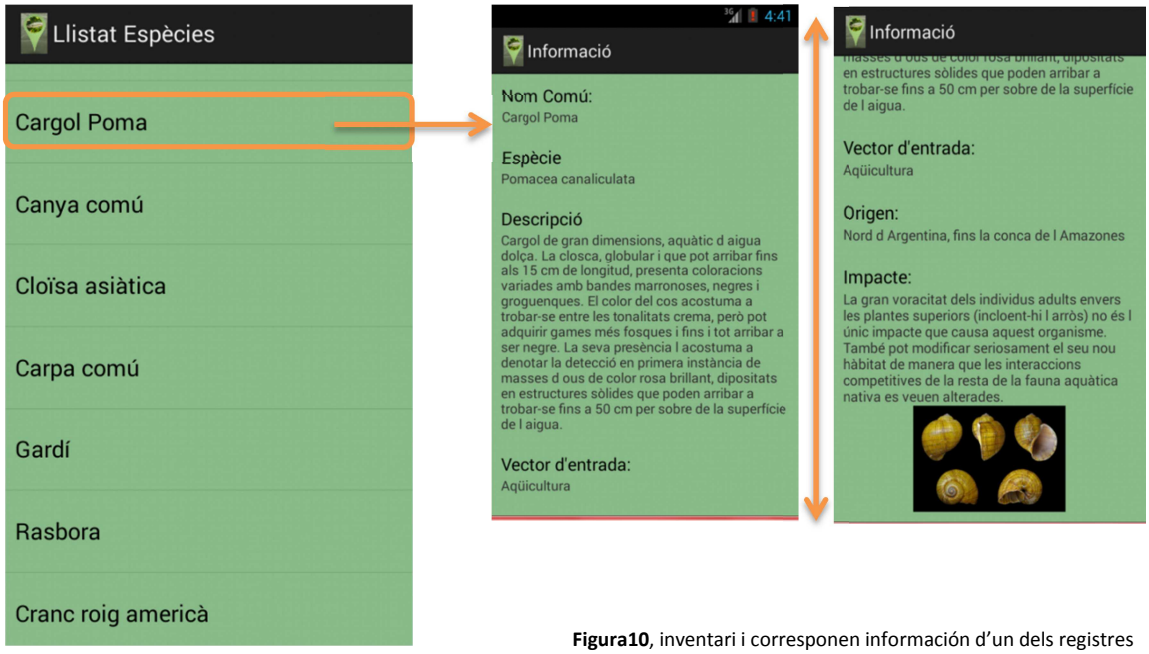


Figura10, inventari i corresponen información d'un dels registres

-Accedir a formulari, implementació del sistema CRUD:

Al prémer el botó accedir a formulari, s'inicia una vista en la que es pot veure els registres que s'han anat generant , en el cas de estar buida, es mostrarà un text que indicia com començar a registrar. (A partir de botó, o menú d'opcions)

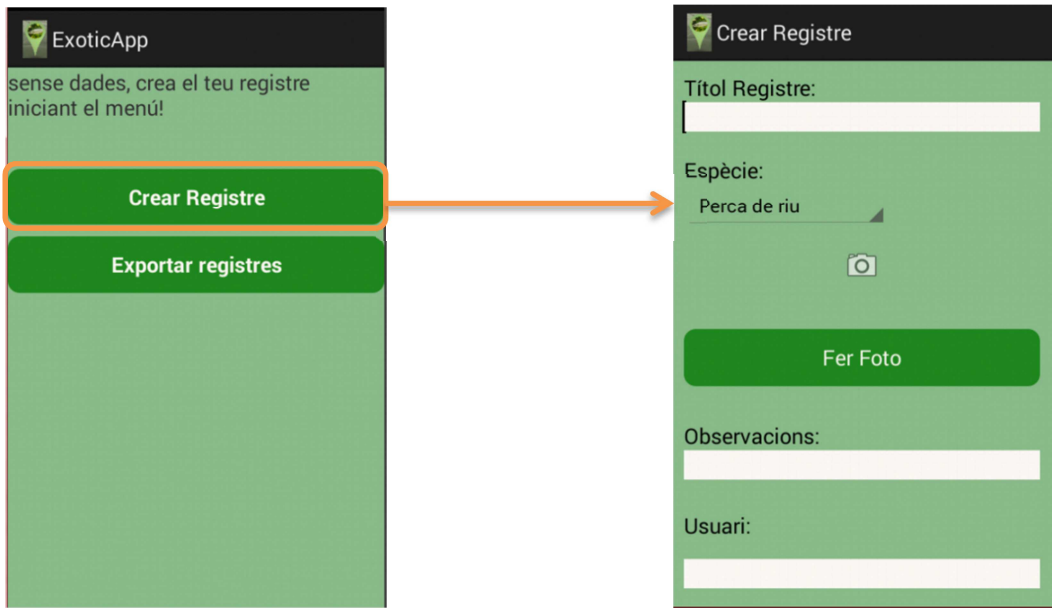


Figura11, vista registres (buida) i accés a formulari

Al prémer el botó registre surt l'activitat del formulari, l'activitat en la que l'usuari ha descriure part de la informació(títol, observacions, usuari, selecció d'espècie i foto, l'aplicació s'encarrega de recollir la resta ata, coordenades i ruta del dispositiu un s'ha emmagatzemat la foto per defecte). Al guardar el formulari, es retornarà a la vista anterior, la de la llista de registres que s'haurà actualitzat automàticament, mostrant el títol i la data i hora en el que s'ha fet. S'ha completat la primera part del sistema CRUD, la creació. Per accedir a la segona, la Visualització, només cal prémer sobre el registre creat.

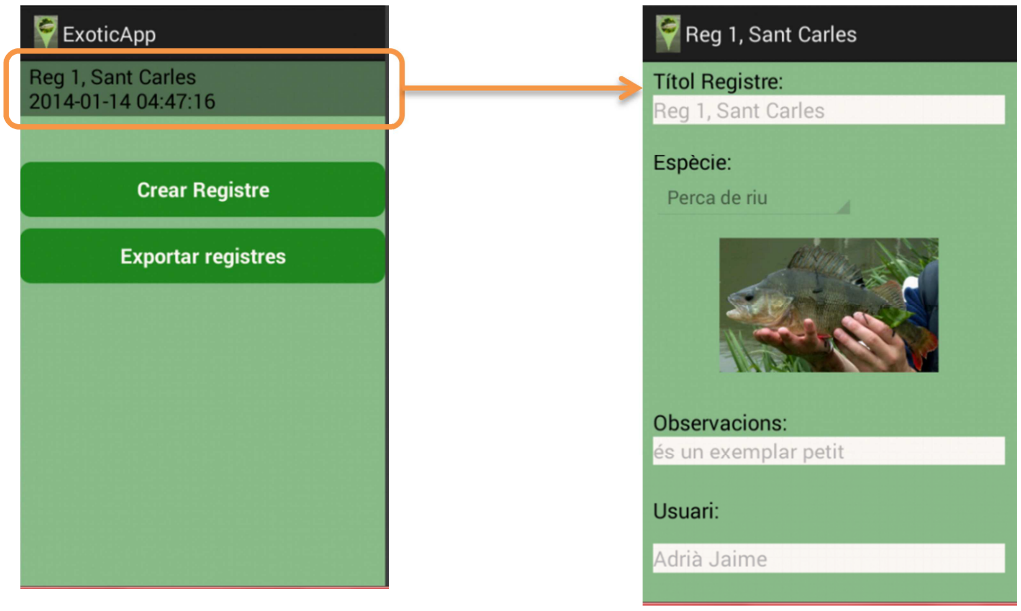


Figura12, vista registres (1 registre) i visualització d'aquest registre al prémer a sobre.

En aquesta vista, com es veu a la imatge, la informació no és modificable, esta en mode únicament de visualització. És en aquest punt en el que es pot decidir si es vol modificar quelcom dels registres ja creats. Accedint al menú (més endavant es dedica un apartat a menús d'opcions i contextuals) hi ha la opció editar registre. Al guardar els canvis amb èxit surt el missatge "registre modificat" . Actualitza automàticament la base de dades.

Figura 13, Edició i retorn a la vista de registres actualitzada.

S'ha creat, visualitzat i editat un registre. Per últim queda eliminar-los per això cal accedir al menú contextual de la llista de registres o bé un cop s'ha accedit al registre concret, al menú d'opcions. En qualsevol dels casos, s'ha programat un diàleg que sorgirà en el moment previ d'executar l'esborrada, preguntant a l'usuari si està segur de que vol fer-ho. En cas afirmatiu, es tornarà a la activitat on es visualitza el llistat de registres, eliminant automàticament el seleccionat i actualitzant la base de dades.

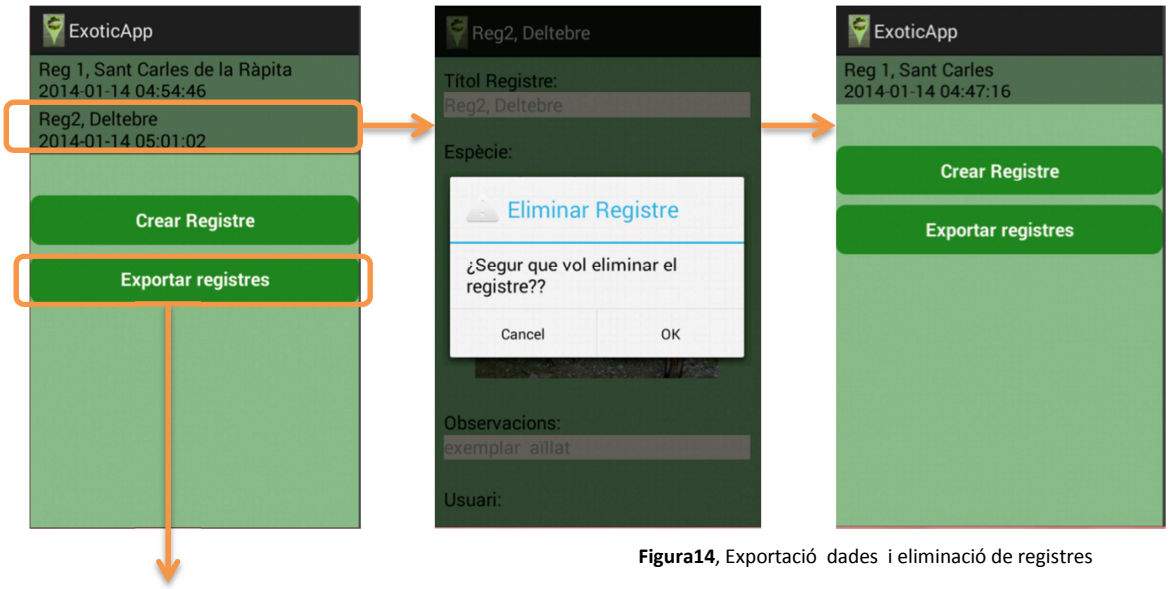


Figura14, Exportació dades i eliminació de registres

Exporta els registres a la memòria externa per defecte accessible al connectar el dispositiu al ordinador per USB. Cal assegurar-se de que s'activa el dispositiu al connectar-lo com a dispositiu d'emmagatzematge USB.

- Menús d'Opcions i Contextuals

L'aplicació compta amb una sèrie de menús d'opcions i contextuals per tal de facilitar la navegació per les diferents activitats.

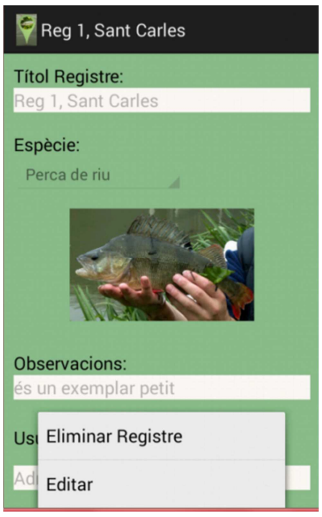


Figura15, menú d'opcions

Un menú d'opcions (prement el botó que desplegui els menús , diferent en cada dispositiu) en la vista de registres et permet crear-ne un de nou. (esquerra)En el cas d'entrar en un registre en concret, et permet editar-lo o eliminar-lo. El menú contextual sorgirà al prémer durant uns segons sobre un dels registres. Permetent la visualització, edició o eliminació d'aquest(dreta).

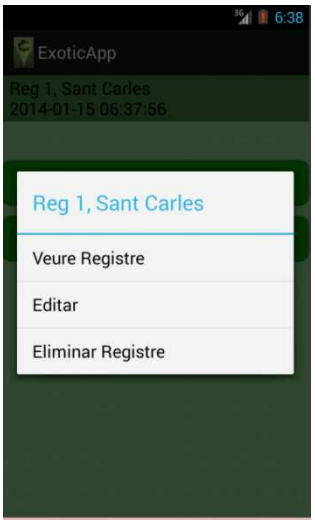


Figura16, menú contextual

- Arxius exportats (base de dades i XML)

L'aplicació genera una sèrie d'arxius per tal de poder treballar amb les dades recopilades amb el dispositiu. Es pot generar una còpia de la base de dades per tenir els registres en una taula editable des de un administrador d'escriptori, o bé un XML per tenir els registres com a recurs per alimentar altres bases de dades.

Una mostra de dos registres exportats per XML:

```
<registre>
<titolregistre>
  <id>1</id>
  <titol>Reg 1 Sant Carles</titol>
  <observ>exemplar petit</observ>
  <usu>Adrià Jaime</usu>
  <espid>7</espid>
  <foto>/storage/external_SD/DCIM/100LGDSC/CAM00050.jpg</foto>
  <data>2014-01-08 13:55:34</data>
  <cordx>2.18598</cordx>
  <cordy>41.4094</cordy>
  </titolregistre>

<titolregistre>
  <id>2</id>
  <titol>Reg2, Deltebre</titol>
  <observ>hi ha molts</observ>
  <usu>Adrià Jaime</usu>
  <espid>5</espid>
  <foto>/storage/external_SD/DCIM/100LGDSC/CAM00051.jpg</foto>
  <data>2014-01-09 11:10:48</data>
  <cordx>2.18598</cordx>
  <cordy>41.4094</cordy>
</titolregistre>
</registres>
```

Figura17, Arxiu XML exportat al PC des de l'aplicació ExoticApp

Una mostra de la visualització de la taula CITA amb 3 registres copiada directament a la memòria externa , per visualitzar aquets arxiu a l'ordinador caldrà un administrador com SQLite Administrator:

_id	c_titol	c_observacions	c_usuari	c_esp	c_data	c_foto	c_x	c_y
3	tshcb	vsnck	Selma	4	2013-12-09 12:31:29	/storage/external_SD/DCIM/100LGDSC/CAM00289.jpg	2,18394075671138	41,4093062486373
4	REG 2 SI	momotombo	Samuel	6	2013-12-09 12:33:30	/storage/external_SD/DCIM/100LGDSC/CAM00290.jpg	2,18396645312194	41,4092179920184
5	METRO	fgjbd	Jaime	4	2013-12-09 14:40:16	/storage/external_SD/DCIM/100LGDSC/CAM00291.jpg	2,18404641430168	41,4089426051865

Figura18, Registres directament exportats al PC i visualitzarts amb SQlite Administrator

4. El Visor

Per muntar un visor, primer cal preparar un servidor web, i tenir les dades necessàries transformades als formats correctes. Per tant abans d'iniciar l'explicació de com s'ha dut a terme el visor i les seves característiques, es necessari exposar el tractament de dades i la instal·lació del servidor.

4.1 Transformació de dades

Partim de la base de dades SQLite que s'han exportat de l'aplicació a l'ordinador. Primer de tot cal pasar la informació a .csv(*Coma Separated Values*) per tal de poder generar shapes des de quantumGis. Per això cal obrir l'arxiu generat per l'aplicació Ebre.db amb l'administrador que s'ha triat, en aquest cas SQLite Administrator, que permet exportar les dades de les taules al format desitjat (csv).

Un cop s'ha obtingut aquest arxiu, caldrà obrir un SIG d'escriptori, com ara Quantum Gis (punt 2.2). Des d'aquest programa, es generaran *Shapefiles* a partir de les columnes de coordenades de la taula c_x i c_y. Aquests Shapes serán de punts, un per registre i heretaran tots els camps de la taula com a atributs mostrables posteriorment en el visor interactiu.

- SQLite Administrator

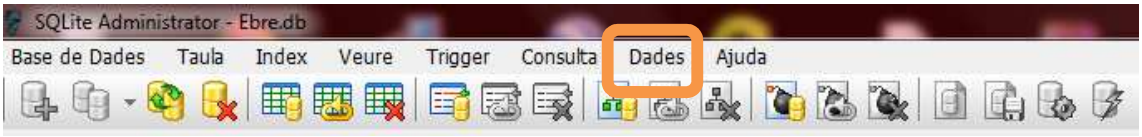
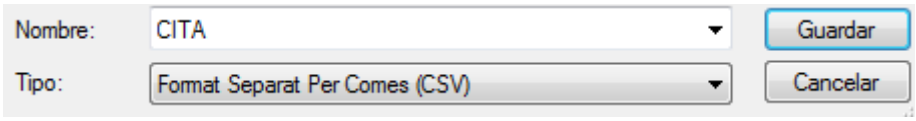


Figura19, Eines part superior SQLite administrator

Des de Dades hi ha varies opcions, s'ha de fer una exportació a CSV.



Dades>Exportar dades , Format Separat Per Comes(CSV)

Un format per representar dades, en les qeu les columnes es separen per comes (o amb el separador decimal que correspongui a cada país) i les files per salts de línies. Cal tenir en compte doncs, que els camps que continguin salt de línia o han d'estar tancats entre cometes

Guardar l'arxiu a la carpeta que s'ha assignat com a espai de treball. Arxiu al qual s'accedirà des de el SIG d'escriptori Quantum Gis (és el que s'utilitza en aquest projecte, però son vàlids altres també).

- Quantum Gis

Des de la pestanya Capa a la barra superior, accedir a 'Añadir capa de texto delimitado'.

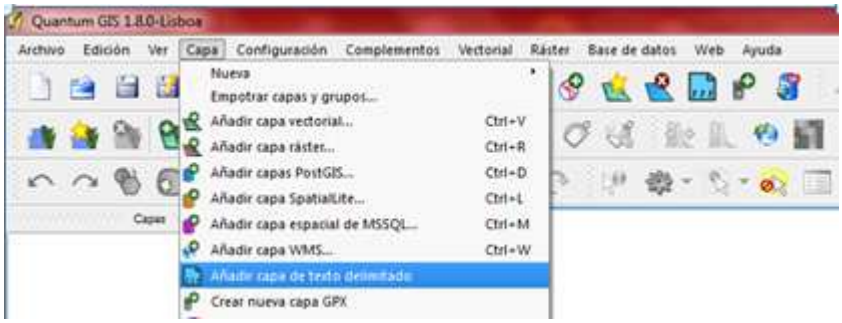


Figura21, com afegir capes a partir de CSV a Q Gis

Al fer-ho s'obrirà una finestra on cal definir:

- Nom d'arxiu csv i nom de la capa resultant
- Definir el delimitador amb els que es separen els camps, en aquest cas (;) per no confondre amb les comes dels decimals d'alguns camps.
- Els camps en els que estan definides les coordenades x,y.
- Surt una mostra dels camps, per veure com interpreta la informació i així seleccionar el separador que més ens convingui

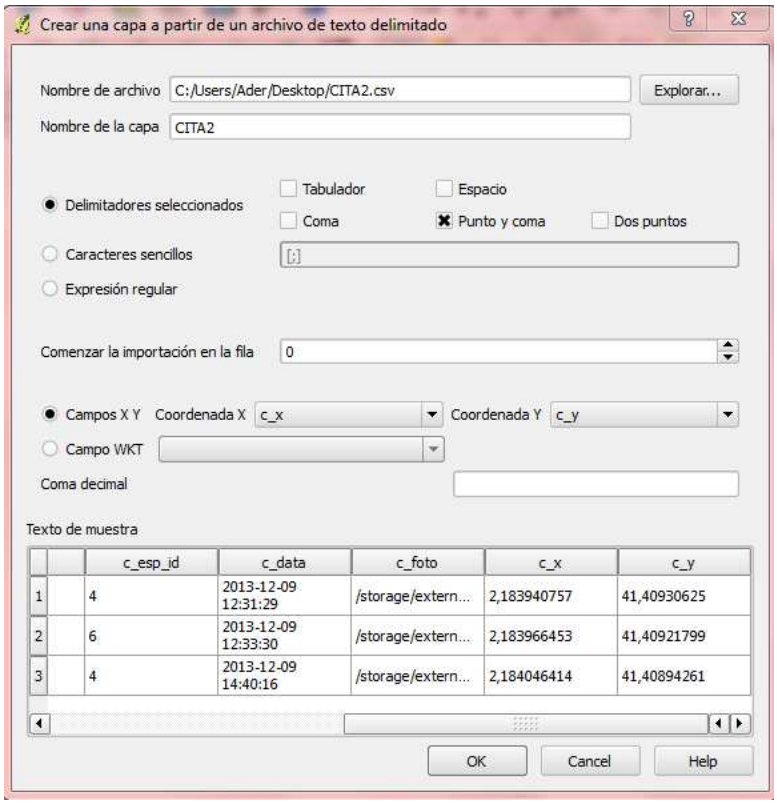


Figura22, interfície de transformació de CSV a geometria de punts

A l'acceptar ens demana que definim el sistema de referencia que es vol implementar, en aquest projecte es treballa amb ETRS89. Un cop seleccionat i confirmat, sortirà la capa resultant.

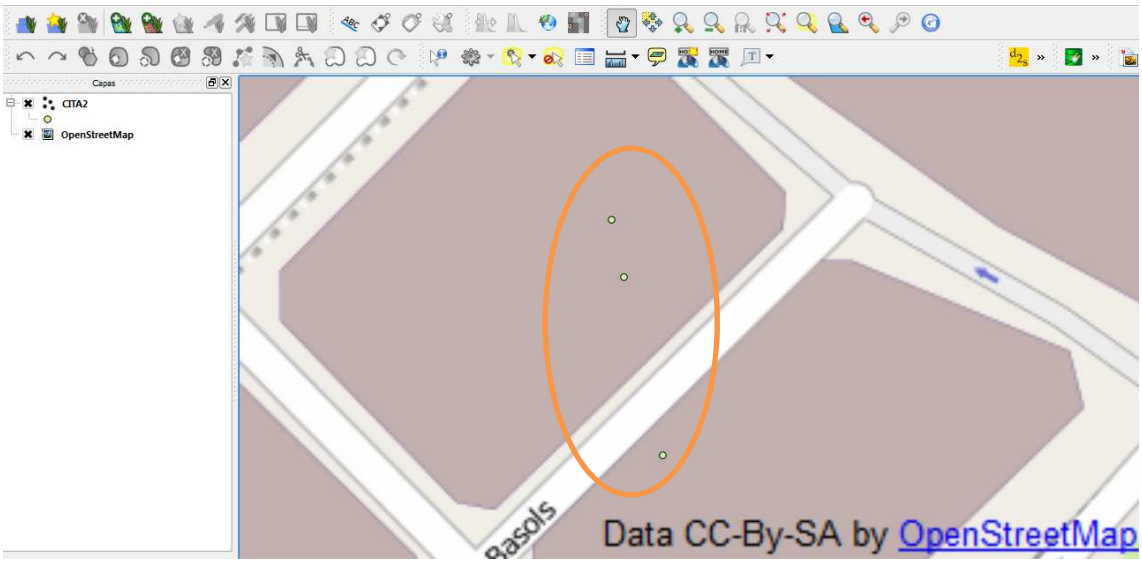
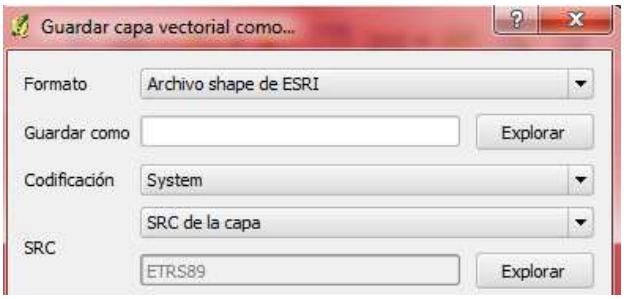


Figura23, tres registres després de la transformació, sobre una base Open Street Map

És recomanable activar algun plugg-in de cartografia base amb el que poder corroborar que els punts es generen a la zona correcte. Quantum GIS té la opció a la pestanya 'Compelemntos' de descarregar-te una sèrie de complements. En aquest cas s'ha descarregat el de Open Layers, que proporciona una sèrie de WMS amb cartografia base, (Google, Bing, i Open street Map) podent així corroborar la posició dels vectors generats.

Aquesta capa de punts, és un '*event*' es un format virtual, és a dir, no hi ha arxiu físic, per tant no es pot treballar amb ell fora del mapa en el que hagi estat generat. Això es soluciona exportant aquest '*event*' a un format que es pugui transferir, com ara un *Shapefile*.

Al fer clic sobre el botó dret sobre la capa de punts al menú de capes es desplega un menú contextual on surt la opció 'guardar como'. Es determina com a arxiu *Shape* d'ESRI. Es selecciona el directori en el que es vol guardar (recomanable tenir una carpeta per cada *shape*, ja que cada una conté varis arxius, d'aquesta manera queda el directori més ben organitzat.



Aquests *Shapes,* que han heretat tots els atributs i marquen la posició en la que s'han pres els registres, son els que s'utilitzaran per implementar el servidor de mapes on-line de GeoServer i posteriorment visualitzades des del visor web o Google Erath.

4.2 Muntatge Del Servidor Web Local.

A continuació es mostra com s'ha dissenyat el sistema d'arxius que composen el servidor des del que s'ha desenvolupat el visor web, a partir de la figura següent, es desglossarà cadascun dels seus components, per veure quin paper hi juga:

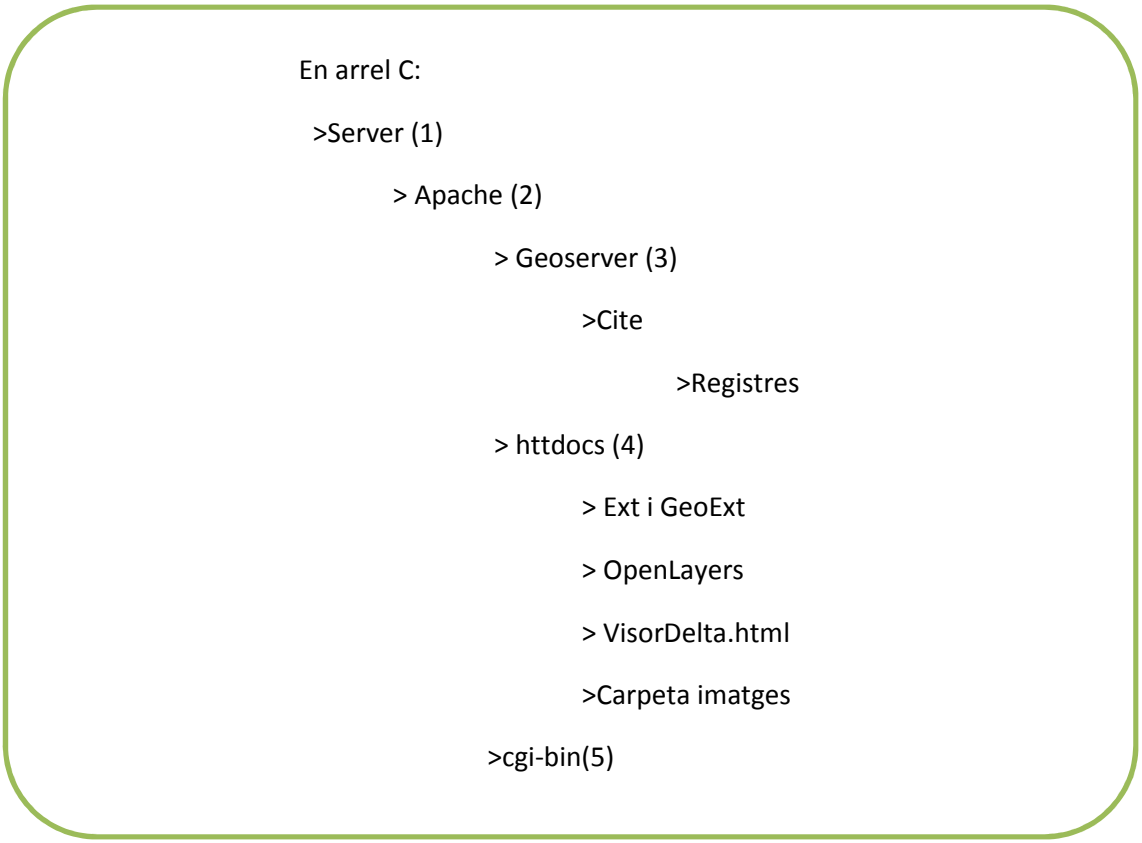


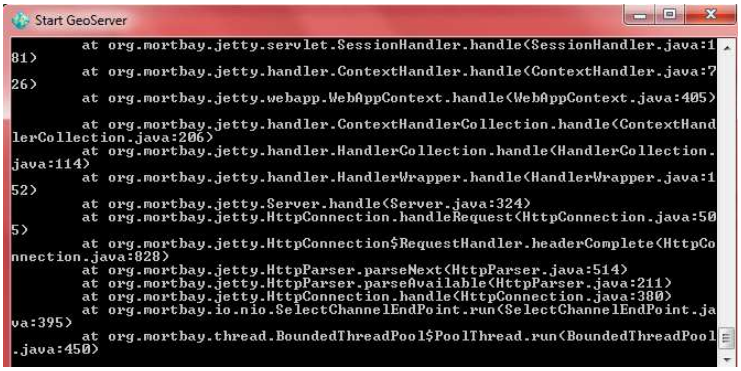
Figura24, Esquema de directoris del servidor web local

(1) És un sistema de fitxers i directoris que treballaran a partir del **localhost**. Aquest Local Host és és un nombre reservat que tenen totes les computadores per treballar a escala local, i s'ha de vincular a un port, en aquest cas '8080'. Des de localhost és des d'on s'accedeix al servidor muntat al nostre propi ordinador i des d'on s'executarà el contingut web. És recomanable si s'està en entorn Windows (en el que s'ha desenvolupat aquest projecte) descarregar el següent: ***Win32 Binary (MSI Installer)*** *apache_2.2.4-win32-x86-no_ssl.msi.*

(2) En aquesta carpeta es crearà el servidor web per al nostre localhost. El servidor web es un programa que s'executa al nostre ordinador mantenint-se a l'espera de peticions d'execució que farà el client o usuari d'internet. Aquest servidor contestarà de forma adequada entregant com a resultat o bé una aplicació web, o bé tota mena d'informació en funció dels comandaments sol·licitats.

En aquest cas s'ha instal·lat un servidor anomenat Apache (versió 2.2), Està configurat per iniciar-se automàticament quan s'inicia sessió a l'ordinador. I dins del seu directori s'hi instal·laran tots els components necessaris per fer funcionar el visor web.

(3) **Geoserver 2.4.3**, És el servidor de mapes on-line on inserirem els Shapes generats amb anterioritat de manera que siguin accessibles des del servidor web. És a dir, Geoserver permet generar Web Map Services propis. Un cop instal·lat, cal inicialitzar-lo amb la opció "Start Geoserver" que executarà una consola ses de la que podem veure com carrega tot el seu contingut, un cop ha acabat, aquest contingut serà accessible.



Compta amb una pàgina d'administració de les dades. En aquesta a pàgina, que s'obrirà des del *localhost (http://localhost:8080/geoserver/web/)* , S'ha de generar un espai de treball, un magatzem i inserir-hi les capes a les que es vol accedir des del visor a partir d' OpenLayers..

L'ordre és el següent: primer crear un espai de treball, o bé utilitzar-ne un d'existent, sigui om sigui s'ha de determinar per tal de generar les rutes del wms. El mateix amb el magatzem, en aquest cas s'ha generat un amb el nom REGISTRES. On s'hi ha inserit la capa que hem generat fent us del botó:

S'ha d'especificar un estil i una projecció. Aquesta última ha de coincidir amb la projecció del visor web i de la capa generada amb Quantum GIS, es a dir, ETRS89 UTM 31 N (EPSG 258319).

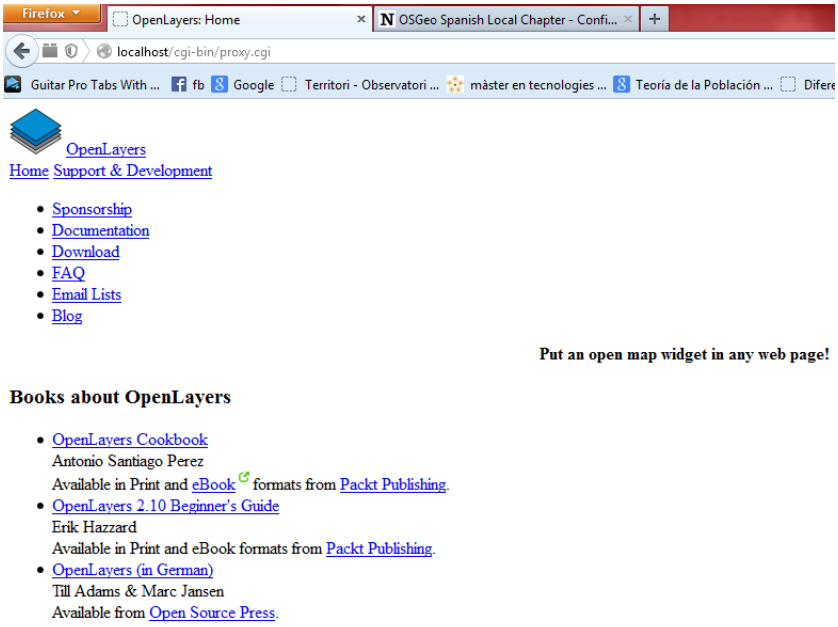
En quant a l'estil GeoServer té alguns per defecte, però accepta que s'importin d'altres sempre que sigui amb format SLD. Per generar estils propis s'ha utilitzat Udig o qualsevol altre programa que permet generar un estil i exportar-lo en aquest format. Per importar aquests estils cal registrar-los en el apartat Estilos del menú datos a la pàgina d'administració.

Un cop publicada la capa, s'hi podrà accedir es de OpenLayers com a capa a partir de la ruta wms estipulada que seguirà el patró *'http://localhost:8080/geoserver/(nom del workspace)/wms*'. En el cas d'aquest projecte serà *'http://localhost:8080/geoserver/cite/wms'*.

(4) **Htdocs,** Aquesta carpeta és en la que es guarda el codi amb el que es programa el visor, i també s'han instal·lat aquí, per facilitar rutes, les llibreries necessàries Ext, Geoext per el disseny dels menús i del panell del mapa, i Open Layers, per les funcionalitats del visor web i la càrrega de capes. En un principi, aquestes carpetes de llibreries no és necessari modificar-les, només actuen com a fonts de recursos per la programació del visor, a partir del document html que, òbviament, si que s'haurà de modificar.

(5) Cgi-bin Aquesta carpeta, es en la que s'ha de posar un arxiu Proxy. És necessari per visualitzar informacions de diferents elements en un mateix port. És un arxiu en el qeu s'ha de posar el nom del host que ha de suportar, en aquest cas localhost:8080, i estipular la ruta a la consola Phyton. Un *proxy.cgi* de mostra es pot extreure de la pàgina de preguntes freqüents de OpenLayers. Daquest arxiu només cal modificar els dos elements assenyalats i posar-lo a aquesta carpeta. Més endavant es veurà com es crida aquest arxiu des de l'arxiu en le que s'esta programant el visor.html.

Es recomana fer la proba de que el proxy.cgi és detectat posant al buscador Firefox la ruta *http://localhost/cgi-bin/proxy.cgi*. Si surt el que es mostra en la captura e pantalla següent és que està ben configurat:



En Apache, no està configurat per defecte que es pugui accedir a arxius cgi, és per això que cal accedir al arxiu httpd de la carpeta '*config*' generada automàticament al instal·lar Apache. Obrirlo en un bloc de notes i descomentar el mòdul de cgi (fig. 25)

#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule charset_lite_module modules/mod_charset_lite.so

Figura25, mòdul cgi descomentat

Guardar l'arxiu i reiniciar el servidor Apache.

4.3 El Visor

Un cop es té l'entorn llest, es pot començarà idear el visor web. Aquest visor es construirà sobre un document html que s'editarà des de notepad ++ i s'anirà depurant i controlant el seu correcte funcionament des del buscador Mozilla Firefox.

4.3.1 Disseny

La idea és la d'un visor senzill, on predomini a la pantalla el mapa que s'estigui visualitzant, procurant que la interfície del seu voltant sigui senzilla i intuïtiva:

Figura26, Esquema del Visor Web

Per generar una interfície com aquesta s'ha combinat l'ús de les llibreries ext js i GeoExt amb Open Layers. A partir del llenguatge JavaScript. La cartografia base és a partir de Web Map Services.

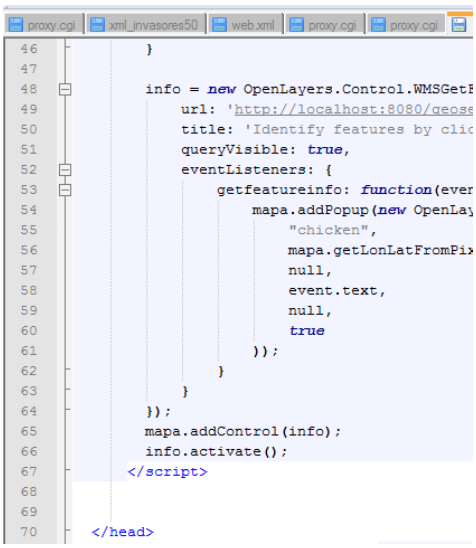
S'han utilitzar cartografies del Institut Cartogràfic de Catalunya i OpenStreetMap.

- Topogràfic de Catalunya (ICC)
- Ortofoto de Catalunya (ICC)
- Vol aeri de 1956 (ICC)
- Open Syteet Map Mapnik (OSM)

Aquestes son les capes disponibles com a cartografia base.on s'hi sobreposarà la informació del servidor Geo Server genereat anteriorment, on s'hi troben els punts amb la informació dels registres d'espècies invasores localitzats amb l'aplicació ExoticApp.

4.3.2 El Document Html

És el document a partir del qual es criden tots els objectes i Webs Map Services (*WMS*), a part de programar totes les funcionalitats que es volen atorgar al visor.



La base estructura de tot document Xml és la següent:

```
<HTML>
      <HEAD>
      </HEAD>

      <BODY>
      </BODY>
</HTML>
```

<HEAD>

Aquí és on hi va el títol de la pàgina per una banda , entre les etiquetes <title></title>
També hi situem el codi JavaScript <script type ="text/javascript"> totes les funcions i objectes que cridem d'OpenLayers i d'altres llibreries aniran entre aquests scripts.

També hi situem els links a llibreries externes o documents d'estil .

En el cas de tenir un arxiu *css* (d'estil) com en el cas d'aquest visor, anomenat estilo.css es declara de la següentmanera :

```
<link rel="stylesheet" href="estilo.css" type="text/css">
```

En el cas de vincular-se a una llibreria externa instal·lada a l'ordinador es declararà la ruta de la següent manera (en el cas d'OpenLayers):

```
<script src="Openlayers-2.12/Openlayers-2.12/OpenLayers.js" type="text/javascript"></script>
```

<BODY>

Entre les etiquetes 'body' hi va tota la estructura de contenidors (divs) de la página que s'està contruint, és a dir, on es delimiten els conteneders d'objectes que s'ha ensenyat al punt 4.3.1.) i s'assigna un id:

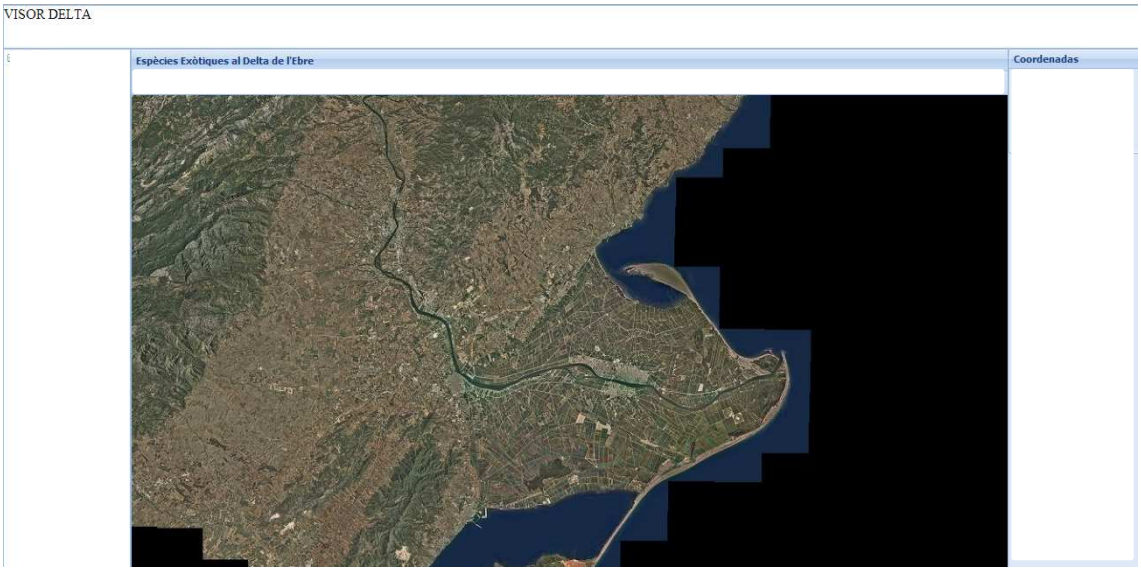
```
<div id="gxmap"></div>
```

També es pot donar format des del codi directament, no és necessari asignar un arxiu d'estil css, com a mostra, l'estil de la tipografía de la capcelera d'aquest visor s'ha fet directament amb el codi de l'arxiu html tal que així:

```
 <div  style = "font-weight: bold"id="contenidoN" align= center "margin-top= 10px"  > ESPÈCIES EXÒTIQUES AL DELTA DE L'EBRE</div>
```

4.3.3 Funcionalitat

Primer de tot, és necessari generar el contenidor on es situarà el mapa , i els contenidors dels voltants seguint l'esquema de la figura 26.



Per fer aquesta estructura, els contenedors es definiràn al <Body> del document html:

```
<div id="gxmap"></div> (és le central, on hi va el mapa)

 <div id="contenidoN"> VISOR DELTA</div> (Capcelera, títol)

 <div id="contenidoW">Oeste</div> (és on es siturà el menú/arbre de gestió de capes)
```

Es definirà la funció de cada un d'aquests a partir de scripts. El generat per contenir el mapa , el del centre i més ampli, és on hi declararem les funcions OpenLayers

```
 mapa = new OpenLayers.Map
```

i les seves corresponents propietats (tot detallat als annexes de codi)

- *càrrega de capes*

Comes veu a la figura, es pot visualitzar un mapa d'imatge satèl·lit al contenedor central, aquest mapa s'ha aplicat a partir de servidors de mapes remots, en aquest, cas del ICC. Per dur a terme aquesta connexió, és necessari una url que es pot trobar en les pàgines web dels servidors de mapes públics. Un cop localitzat el mapa, el seu nom i el seu url, es declara com a capa de OpenLayers:

```
orto5 = new OpenLayers.Layer.WMS(
                    "Orto 5m",
                        "http://shagrat.icc.es/lizardtech/iserv/ows",
                        {layers: "orto5m"}, //nom de la capa concreta del WMS
                        {isBaseLayer: true} //utlititzar-la com a cartografia base
                );
```

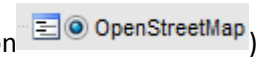
En el cas de les capes que no han de ser cartografía base, si no que s'han de sobreposar, el procediment és el mateix. Però per poder visualitzar-les sobre la cartografia base s'ha de cambiar la opció *isBaseLayer* a *false*. Per tal de poder activar i desactivar les capes sense que afecti a la que tenim com a base.

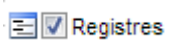
És el cas de les capes ques'utilitzen per la visualització de capes dels registres obtinguts pels usuaris des de l'aplicació móbil. Aquestes capes s'obtindràn del servidor montat prèviament amb GeoServer. I s'han de cridar com qualsevol altre wms però amb la ruta del servidor en localhost:

```
Reg= new OpenLayers.Layer.WMS(
                    "Registres", //titulo de capa
                        "http://localhost:8080/geoserver/cite/wms", //URL
                        {layers: "registres", format:'image/png',transparent:true
                {isBaseLayer: false, singleTile:true}
```

Un cop definides les capes i les seves característiques, cal afegir-les al panell del mapa, per això s'ha d'utilitzar el comandament mapa.addLayer[Noms de les capes];.

Figura27, capes carregades i arbre de capes

En aquesta figura ja es divisen altres elements, com l'arbre de capes del contenedor esquerre, on es disposen tots les capes que es poden visualitzar, veiem les bases vartogràfiques, de les quals només pot estar activa una (radio button 

) i la capa de registres, independentment de les altres pot ser activada o desactivada a partir d'un check Box 

. Aquest arbre s'ha programat a partir del objecte de Ext " *Ext.tree.TreePanel*".

- *Aplicacions del Visor*

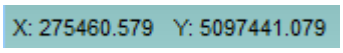
Un cop es te un mapa que visualitzar i el control de les capes, es poden inserir una gran varietat de funcions sobre aquest mapa, per tal de fer interactiu el seu contingut i poder-ne extreu-re el màxim d'informació possible.

Aquestes funcions seràn activades a partir de botons de mesús al voltant del mateix visor o bé clickant sobre elements del propi mapa.

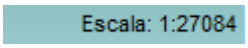
I. Barra d'eines superior del visor: És un menú on s'hi han programat una sèrie d'eines de navegació i mesura. Aquest visor conta amb els següents elemnts:

-Zoom (*OpenLayers.Control.ZoomIn*)
-Allunyar (*OpenLayers.Control.ZoomOut*)
-Extensió determinada del mapa (*OpenLayers.Control.Zoomtoextent*)
-Moure's pel mapa *(OpenLayers.Control.Navigation())*
-Càlcul superfície i distància (*OpenLayers.Control.Mesure*)

II. Barra inferior d'informació: Aquesta barra mostra informació referent al mapa

-Coordenades: a partir d'un event que permet mostrar la informació X,Y ambel simple fet de passar el ratolí per sobre del mapa, anirà mostrant constant ment la ubicació del cursor. 

A partir de: *mapa.events.register("mousemove", mapa, moverRaton)* on mousemove és l'event, mapa el contenedor del mapa, I moverRaton la funció que determina que s'agafi les coordenades del píxel sobre el que està.

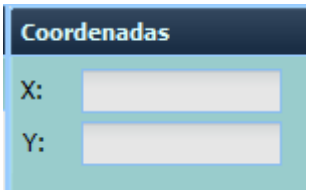
-Escala: Mostra l'escala a la que es visualitza el mapa en cada instant, es va actualitzant constantment en funció dels zooms in i uots que es facin. 

-Combo buscador de topònims: fa zoom a la zona del topònim que s'hagi escrit, aquesta funció només és útil sobre la capa base d'open Street Maps, ja que es connecta a un servidor remot que no està vinculat als servidors del ICC.

GeoNames Això funciona gràcies al projecte Geonames , una base de dades s'abast mundial amb topònims de tots els països, d'accès gratuït, es vincula al visor a partir d'un linck de referència a traves de la llibreria ext js
http://dev.geoext.org/geoext/ux/geoext.ux/ux/GeoNamesSearchCombo/lib/GeoExt.ux/GeoNamesSearchCombo.js"

III. Formularis de búsqueda i obtenció d'informació

- Recerca de coordenades, fa zoom a les coordenades x,y que s'escriguin en els camps" X:" i "Y".



Mesures: Mostra els resultats de les funcions de mesura, sigui distàncie o superfície

Element Seleccionat: Mostra la informació dels registres de les capes inserides al visor, tots els atributs heredats de l'aplicació i que habien passat al Shape, son visibles a partir de la funcio GetFeatureInfo de OpenLayers.

- Per aprofundir més en la programació d'aquestes funcions, acudir a l'Annexe II, on es troba el codi comentat d'aquest visor

4.3.4 Visor Final

Per últim, unes mostres del visor finalitzat, amb les seves diferents capes i funcions.

Primer de tot, les seves capes de cartografia base:

- Ortofotomapa del ICC

Figura28,url del wms: *http://shagrat.icc.es/lizardtech/iserv/ows*

- Mapa Topogràfic del ICC

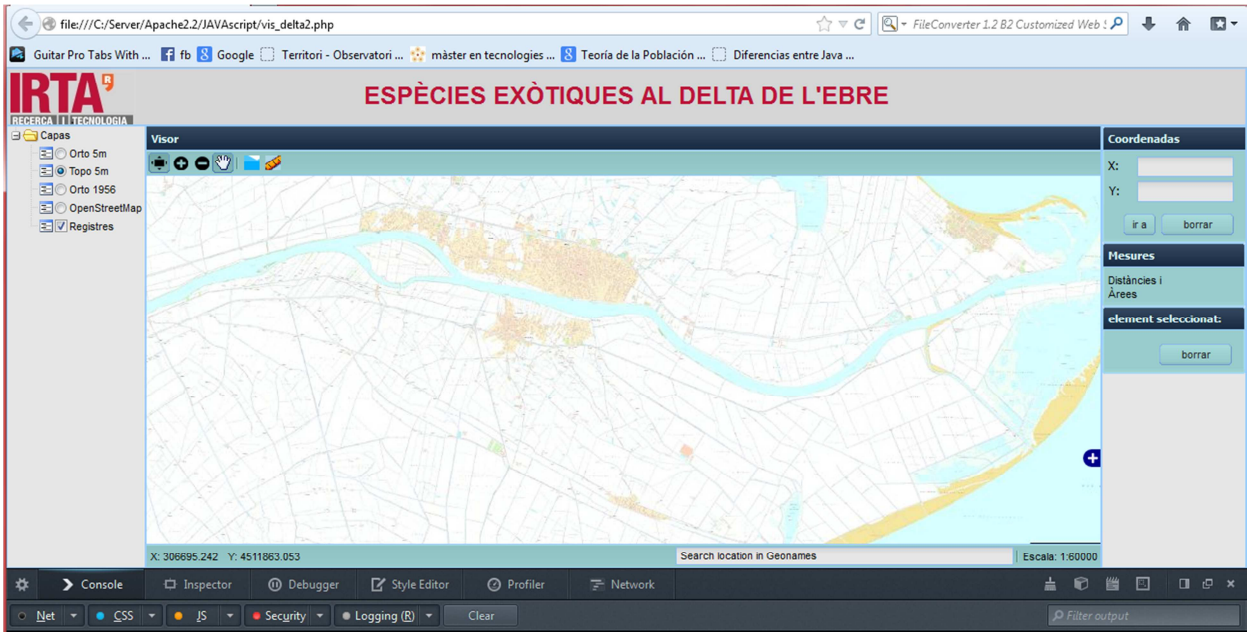


Figura29,url del wms: *http://shagrat.icc.es/lizardtech/iserv/ows*

Mapnik OpenStreetMaps

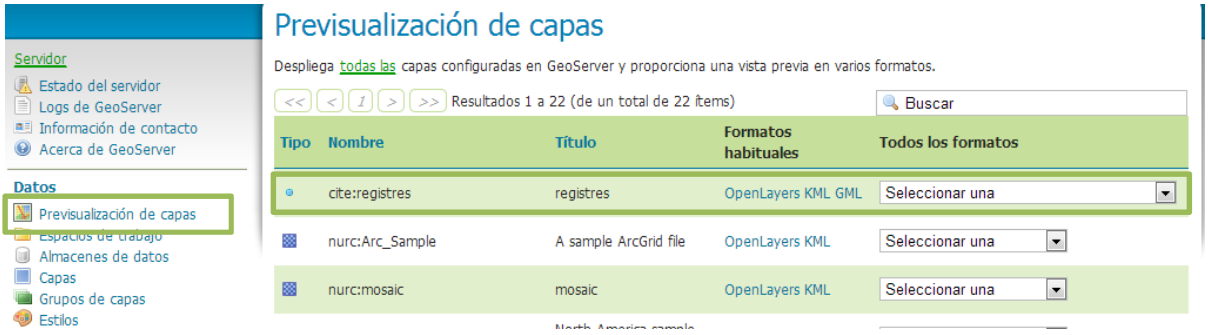
Figura30,url del wms: .OSM("OpenStreetMap")

En aquest últim es pot veure que hi ha un nou element, es tracta del mapa guia, que situa en un context geogràfic més ampli el territori que s'està observant. També, per facilitar la ubicació de la informació dels registres, al fer clic sobre un dels punts, s'ha programat un "*popUp*" que mostrarà tota la informació assenyalant el punt concret del que procedeix. Per si a la taula del formulari dret no acaba de quedar clar

Figura31, Pop Up de getFeatureInfo

- KML i Google Earth

Com a alternativa des de GeoServer també es pot exportar de *shape* a KML (*Keyhole Markup Lenguage)* , un format basat en XML dissenyat per representar dades geogràfiques. Per fer-ho caldrà anar a la pagina d'administrador de Geo Server, iniciar la nostre sessió (si no s'ha canviat, per defecte el nom es 'Admin' i l contrasenya 'geoserver'.



A l'apartat previsualització de capes, hi ha la opció KML. Al clicar-hi, es descarregarà a la carpeta predeterminada de descàrregues del pc un arxiu, que es pot obrir des de Google earth mateix. Aquest arxiu conté els punts dels registres i la corresponent informació de cadascun d'ells .

Es podrà visualitzar així des de programes com google Earth:

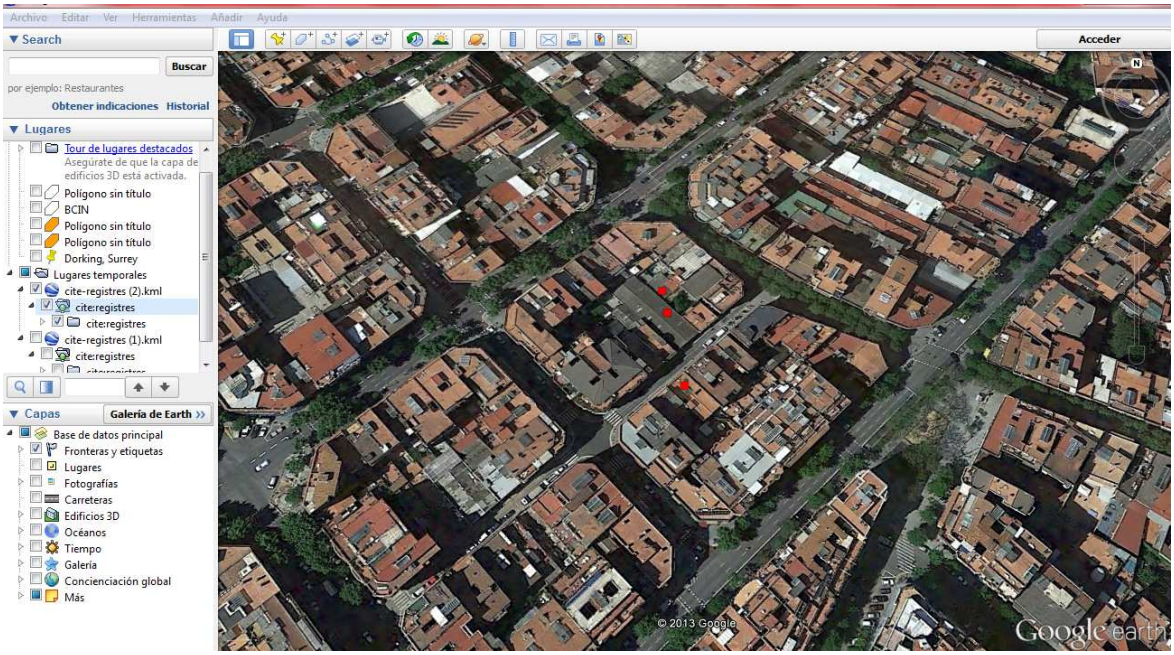


Figura32, visualització des de Google earth

Al clicar sobre els registres es mostrarà la informació de cadascun d'ells. Cal tenir el Geoserver actiu per poder visualitzar aquestes capes.

Figura33, visualització dades d'un registre des de Google earth

5. Notes finals

Aquest ha acabat sent un projecte dividit en tres blocs, com s'ha pogut veure al llarg de la memòria:

- *La programació d'una aplicació per dispositius Android*: Elaboració d'aplicació per dispositius mòbils Android per identificar les espècies invasores al Delta de l'Ebre a partir d'una base de dades i per la geolocalització d'espècimens a partir de formularis en els que es recull l'espècie, foto, coordenades i altres dades que l'usuari pot complementar.

- *Tramitació , validació i publicació de dades en un servidor web:*Conjunt de processos que permeten la transformació de les dades Sqlite a Shape a partir de coordenades x,y. I preparació d'un entorn web.

- *Visió de les dades en un visor web interactiu*: en el que es carreguen les dades de l'aplicació i Shapes a partir de el servidor de mapes Geo Server, amb funcionalitats com recerca de coordenades, topònims i mesura d'àrees i distàncies.

En general , Les tasques desenvolupades en part no han tingut a veure directament amb les estudiades al màster, cosa que es valora positivament ja que d'aquesta manera s'adquireixen nous coneixements. Concretament, tota la part referida al primer bloc. És a dir, tot el desenvolupament de l'aplicació mòbil. Com s'ha vist però, el resultat ha estat una aplicació pràctica i senzilla. Val a dir però, que dista de la idea inicial, que incorporava un visor cartogràfic.

El motiu pel que s'ha modificat ha estat que el visor requeria de cartografia online, i el marc geogràfic al que està destinada l'aplicació és el delta de l'Ebre, on les xarxes d'internet mòbil pràcticament són nul·les. Així doncs, es va optar per la visió dels registres directament en visor web, un cop fet el treball de camp.

Pel desenvolupament d'aquesta aplicació ha estat vital la ajuda trobada a traves de la web. En dos mesos sense tenir pràcticament cap mena de coneixement de programació Java i conseqüentment Android, s'ha pogut desenvolupar una aplicació amb diverses funcions: generació de base de dades, captura d'imatges i geolocalització entre d'altres, tot acompanyat d'una sèrie d'interfícies dinàmiques i amb estil propi. Tot això ha estat gràcies a la solidaritat que ho ha en el entorn web, fòrums i blogs de programadors que exposen solucions a problemes plantejats per tota mena d'usuaris de manera completament desinteressada.

D'altra banda aquest projecte final ha donat la possibilitat de posar en pràctica coneixements adquirits amb els estudis del Màster en Tecnologies de la Informació Geogràfica, en relació a la preparació de tot un entorn de desenvolupament d'un servidor de mapes web i programació d'un visor. A més s'ha hagut de posar-ho a prova en un entorn real, on no esta tot preparat perquè funcioni correctament. Començant des de zero en un ordinador que no

comptava amb el programari necessari i havent de descarregar i instal·lar el programari que ha semblat més adient, fet que ha aportat un altre vessant a l'aprenentatge.

Aquest projecte, no és un projecte 100% tancat, el sistema pot anar modificant-se en funció de les necessitats i anar incorporant millores. Es pot treballar en aspectes com ara la transferència de dades remota, sense necessitat de connectar el dispositiu mòbil a l'ordinador. O bé tractar d'automatitzar al màxim tot el procés per tal d'agilitzar-lo i fer-lo més accessible.

També altres maneres de tractar les dades, és el cas de l'XML, en el projecte l'objectiu es la visualització del registres directament. Per això no s'ha utilitzant l'arxiu que dóna la opció d'exportar l'aplicació en format XML. És Per això que es dedica un tercer annexa (III) a una de les possibilitats que es poden explotar. Es tracta de donar la possibilitat de facilitar la idea de generar una base de dades a partir del XML generat per l'aplicació .

Python

```
...
...       textValues = textValues + "," + registro.firstChild.data
...        elif returnField(registro.nodeName)=="c_y":
...
...       textValues = textValues + "," + registro.firstChild.data
...        elif returnField(registro.nodeName)=="c_idesp":
...       textValues = textValues + "," + registro.firstChild.data
...        else:
...         textValues = textValues + ",'" + registro.firstChild.data + "'"
...
...       sqlString = textInsert + "" + textValues + ");"
...
...      print sqlString
...
... convertir()
...
...
insert into cita (_id,c_titol,c_observacions,c_usuari,c_espid,c_data,c_foto,c_x,c_y) values
(3,'tshcb','vsnck','Selma','4','/storage/external_SD/DCIM/100LGDSC/CAM00289.jpg','2013-12-09 12:31:29',2.18394,41.4093);

insert into cita (_id,c_titol,c_observacions,c_usuari,c_espid,c_data,c_foto,c_x,c_y) values (4,'REG 2
SI','momotombo','Samuel','6','/storage/external_SD/DCIM/100LGDSC/CAM00290.jpg','2013-12-09 12:33:30',2.18397,41.4092);

insert into cita (_id,c_titol,c_observacions,c_usuari,c_espid,c_data,c_foto,c_x,c_y) values
(5,'METRO','fgjbd','Jaime','4','/storage/external_SD/DCIM/100LGDSC/CAM00291.jpg','2013-12-09 14:40:16',2.18405,41.4089);

insert into cita (_id,c_titol,c_observacions,c_usuari,c_espid,c_data,c_foto,c_x,c_y) values (6,'DIA D','es una
pared','Jaume','3','/storage/external_SD/DCIM/100LGDSC/CAM00293.jpg','2013-12-11 09:23:53',1.66876,42.3656);
```

Figura34, mostra de les sentències sql generades amb Phyton

Per això s'incorpora un petit programa dissenyat amb Python en l'esmentat annexa (III), que transforma la informació del XML en sentències SQL llestes per inserta la informació en una base de dades amb una taula en la que es corresponguin els camps amb les etiquetes generades per l'aplicació.

Per últim s'haurà notat, que tot i ser una aplicació destinada al delta de l'Ebre, moltes de les mostres exposades són sobre Barcelona, això és degut a que el desenvolupament del projecte ha estat desenvolupat al despatx de Thigis, situat al Clot. Totes les probes de geolocalització han estat dutes a terme allà, és per això que els *Shapes* resultants, estan ubicats en aquesta zona, i per tant les capes publicades i mostrades al visor són a Barcelona i no al marc geogràfic establert (delta de l'Ebre).

6. Bibliografia

- Programació Android:

F. Ableson, R.Sen , C. King **Android: Guia para desarrolladores,** Ed. Anaya , 2011. (Llibre de suport tècnic, útil per comprendre l'entorn de programació Android.)

www.developer.android.com (Pàgina oficial d'android, on s'hi troben tutorials gratuïts i informació sobre totes les funcions i llibreries d'objectes disponibles)

www.stackoverflow.com (Fòrum col·laboratiu d'intercanvi de preguntes i respostes sobre codi de programació)

- Informació espècies invasores

J. Andreu, J. Pino, C. Basnou, M.Guardiola i J.L. Ordóñez (CREAF). **Espècies Exòtiques a Catalunya**, resum 2012. Projecte EXOCAT.

Documentació cedida per **Thigis S.L.**

- Desenvolupament Web

www.apache.org (descàrrega i documentació sobre el servidor Apache)

www.openlayers.org (descàrrega i documentació de les llibreries OpenLayers)

www.geoext.org (descàrrega i documentació del mòdul Geoext per Ext js)

www.sencha.com (Ext js)(descàrrega i documentació d'aquesta arquitectura per JavaScript)

www.geoserver.org (Descàrrega del servidor de mapes web i documentació)

www.gis.stackexchange.com (Fòrum d'intercanvi de preguntes i respostes relacionades amb programació de sistemes d'informació geogràfica online)

Documentació del **mòdul de programació** del Màster en Tecnologies de la Informació Geogràfica (MTIG 15)

- Institucions:

www.thigis.com (Thigis Serveis Ambientals S.L.)

www.geografia.uab.es (Departament de Geografia Universitat Autònomade Barcelona)

www.irta.cat (Institut de Recerca i Tecnologia Agroalimentària)

http://ligit0.uab.es/mtig/index.htm (Màster en Tecnologies de la informació Geogràfica)