

UNIVERSITAT DE LLEIDA



ESCOLA POLITÈCNICA SUPERIOR

ENGINYERIA EN INFORMÀTICA

SISTEMES INFORMÀTICS (TREBALL FINAL DE CARRERA)

**Disseny i creació d'una xarxa social multiplataforma
basada en geolocalització**

Autor: Josep Maria Flix Rovira

Director: Ferran Perdrix Sapiña

Desembre de 2013

ÍNDEX I TAULES

CONTINGUTS

1. ÍNDEX DOCUMENT
2. ÍNDEX D'IMATGES
3. ÍNDEX DE TAULES

Índex

1.	Capítol 1. Introducció.....	7
1.1.	Motivació.....	5
1.2.	Objectius.....	6
1.3.	Estructura del document.....	7
2.	Capítol 2. Anàlisi.....	10
2.1.	Anàlisi tecnològic entorn mòbil.....	10
2.1.1.	PhoneGap. Introducció a la seva utilització.....	16
2.2.	Anàlisi tecnològic webservices.....	19
3.	Capítol 3. Desenvolupament.....	24
3.1.	Cronograma.....	24
3.2.	Previsió econòmica.....	26
3.3.	Arquitectura del sistema.....	28
3.3.1.	Diagrames d'estat de l'aplicació.....	28
3.3.2.	Descripció de funcionament de l'aplicació.....	30
3.3.3.	Funcions del servidor (Controller+Model).....	31
3.3.4.	Estructura de dades de l'aplicació.....	38
3.3.5.	Elements de seguretat de l'aplicació.....	40
3.3.6.	Aplicació de dispositiu.....	43
3.4.	Resultats obtinguts. Mini manual uVumi App.....	51
4.	Capítol 4. Conclusions i línies futures.....	58
4.1.	Conclusions.....	58
4.2.	Línies futures.....	59
	Bibliografia.....	61
	Referències bibliogràfiques.....	61
	Referències web.....	62
	Annexes.....	65
	Annex A. Funcions map.js.....	65

Índex d'imatges

Imatge 1 Usuaris amb smartphone	7
Imatge 2 Evolució de l'accés a Internet segons dispositius	3
Imatge 3 Coneixement del cloud computing en pymes espanyoles	4
Imatge 4 Multiples equips per diferents OSs suposa nous problemes	11
Imatge 5 Grau de compatibilitat dels navegadors amb HTML5 (sobre 500 pts)	12
Imatge 6 Comunicació JavaScript i Codi natiu OS	12
Imatge 7 Empaquetat d'una aplicació PhoneGap	13
Imatge 8 Arquitectura d'una aplicació PhoneGap	14
Imatge 9 OpenStreetMap. Mapa mundi. http://www.openstreetmap.org	15
Imatge 10 Llibreria JS Leafletjs. http://leafletjs.com	15
Imatge 11 Codi per l'Activity HelloWorld	16
Imatge 12 Codi HTML de la vista de l'app HelloWorld	17
Imatge 13 Aplicació HelloWorld en funcionament.....	17
Imatge 14 Uvumi App, importació de cordova.js	18
Imatge 15 Codi d'inicialització de PhoneGap	18
Imatge 16 Codi de la funció d'inicialització	18
Imatge 17 Codi de controlador de l'esdeveniment DeviceReady de PhoneGap	19
Imatge 18 Gràfica rendiment frameworks PHP	20
Imatge 19 Gràfica rendiment frameworks PHP	20
Imatge 20 Exemple <i>uVumi</i> GeoJSON	22
Imatge 21 Diagrama GANTT projecte. Software OpenProj	25
Imatge 22 Esquelet general de l'aplicació	28
Imatge 23 Gestió events de l'aplicació	29
Imatge 24 Estats toolbar de publicació.....	29
Imatge 25 Diagrama de funcionament del patró MVC	30
Imatge 26 Taula MySQL uvumi_pois	38
Imatge 27 Taula MySQL uvumi_users	39
Imatge 28 Diagrama d'estats de la funció de registre d'usuaris	40
Imatge 29 Diagrama d'estats per l'autenticació e inici de sessió	40
Imatge 30 Activació protecció XSS CodeIgniter. Arxiu config/config.php	41

Imatge 31 Activació protecció CSRF CodeIgniter. Arxiu config/config.php	42
Imatge 32 Diagrama de l'atac CSRF i mesures emprades per la protecció	42
Imatge 33 Exemple funció set per evitar SQL-Injection	43
Imatge 34 Constants d'aplicació que localitzen servidor i funcions WS. Arxiu map.js. 44	
Imatge 35 Constants d'aplicació, estableix limitacions APP. Arxiu map.js	44
Imatge 36 Declaració del plugin i de les seves crides. Arxiu map.js	45
Imatge 37 Declaració del plugin PhoneGAP. Arxiu MediaPlugin.java	45
Imatge 38 Emmagatzemament configuració. Arxiu map.js	46
Imatge 39 Funció que permet fer login i executar una funció segons el resultat. Arxiu map.js.....	46
Imatge 40 Generació de missatges estàtics dins l'aplicació. Arxiu map.js.....	47
Imatge 41 Missatges localitzats juntament amb la interfície. Arxiu map.html	47
Imatge 42 Funció de tancament de sessió, enviament crida WS de logout. Arxiu map.js	48
Imatge 43 Funció per capturar la pulsació del botó back del dispositiu. Arxiu map.html	48
Imatge 44 Inicialització del mapa de la interfície. Arxiu map.js.....	49
Imatge 45 Element HTML contenedor del mapa de l'aplicació. Arxiu map.html	49
Imatge 46 Funció boundaryRequest per obtenir els punts personalitzats. Arxiu map.js 50	
Imatge 47 Funció SQL per obtenir els punts GEOMETRY ubicats en un polígon. Server model.....	50
Imatge 48 Assignació de punts al mapa, després de la recepció. Arxiu map.js	50
Imatge 49 "Splash screen"	51
Imatge 50 "uVumi location screen"	52
Imatge 51 "Add uVumi text screen"	53
Imatge 52 "uVumi text display"	53
Imatge 53 "Zoom view"	54
Imatge 54 Icones uVumi segons contingut.....	54
Imatge 55 "uVumi photo popup"	55
Imatge 56 "Add uVumi media dialog"	55
Imatge 57 "Other options screen"	56

Índex de taules

Taula 1 Requeriments per al desenvolupament segons dispositiu	10
Taula 2 Fragmentació segons característiques i OS's mòbils	11
Taula 3 Resum de tecnologies emprades en el projecte	22
Taula 4 Valoració econòmica del projecte	27
Taula 5 Valoració econòmica del manteniment anual del projecte	27

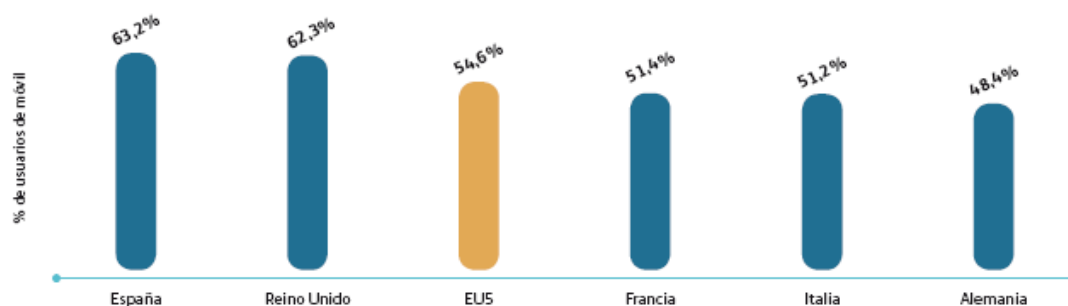
CAPÍTOL 1

1. INTRODUCCIÓ
2. MOTIVACIÓ
3. OBJECTIUS
4. ESTRUCTURA DEL DOCUMENT

1. Capítol 1. Introducció.

Actualment una tercera part de la població amb una edat de 10 anys ja utilitza la telefonia mòbil¹, de fet l'ordinador portàtil ha superat per primer cop al de sobretaula com a principal mitjà per accedir a Internet des de les llars i dos de cada tres internautes disposa d'un *smartphone* i un de cada sis disposa d'una tableta.

El telèfon mòbil actualment és una eina universal, el nombre de subscripcions anuals equival al 85% de la població mundial i en els països desenvolupats supera el 100% ja que molts usuaris tenen diverses línies, aquest fet s'ha produït gràcies a l'abaratiment dels serveis.

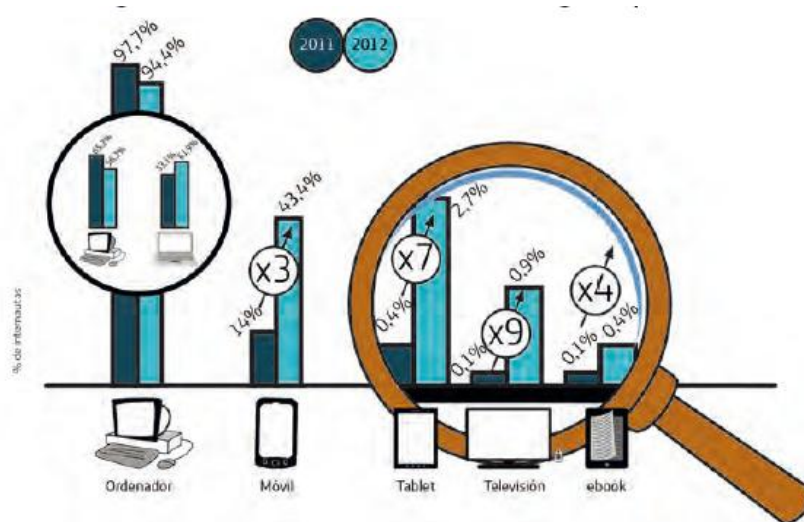


Fuente: Comscore, datos de octubre de 2012.

Imatge 1 Usuaris amb smartphone

L'any 2011 va estar l'any de l'explosió dels terminals mòbils intel·ligents o *smartphones* ja que durant aquest any varen duplicar la seva quantitat, aquesta taxa de penetració encara no ha arribat a les tabletas, encara que la previsió és que aquests dispositius tinguin un gran índex de penetració a nivell mundial, fins al punt d'arribar a posar en perill d'altres dispositius com poden esser els portàtils.

¹ eEspaña. Informe anual 2012 sobre el desarrollo de la sociedad de la información en España. Fundación Orange. [TELEFONICA2012]



Fuente: Telefónica. Datos de abril de 2012.

Imatge 2 Evolució de l'accés a Internet segons dispositius ²

Un dels grans beneficis d'aquesta multiplicitat de xarxes i dispositius d'accés ha donat la possibilitat d'accedir de forma transparent als serveis des de qualsevol lloc, encara que el fet de la multi plataforma i el multi dispositiu planteja la necessitat de fer desenvolupaments específics per a cada servei i proveir-los d'algun mecanisme que permeti un pas ràpid entre plataformes o dispositius. Per aconseguir aquesta fita, cal seguir *standards* prou robustos i prou sòlids que fonamentin el desenvolupament futur i que permeti que es pugui accedir als serveis des de qualsevol de les plataformes existents.

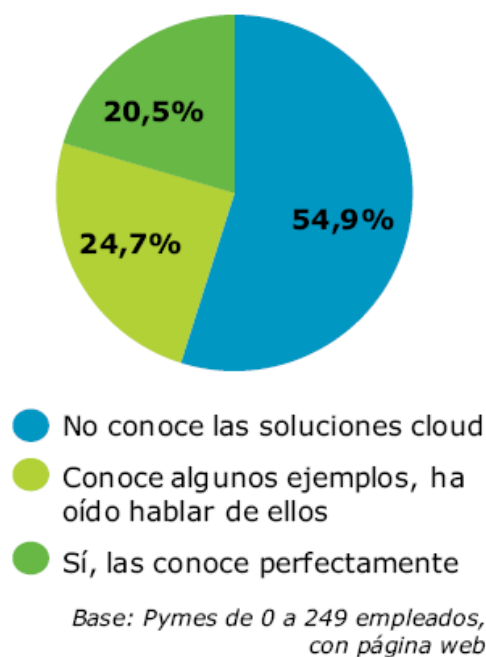
Aquesta situació queda força resolta amb l'arribada del *standard* **HTML5** que durant l'any 2012 ha experimentat una evolució cap a la generalització. De fet algunes de les característiques que s'incorporen en aquest *standard* fins ara només estaven disponibles en plataformes propietàries. Sota aquest paraigües podem trobar altres tecnologies com *CSS3*, *Javascript* amb *frameworks* com *JQUERY* o *JSON* que permeten un *standard* més universal i s'han convertit en la tecnologia de referència en el desenvolupament d'aplicacions web i fins i tot en alguns desenvolupaments mòbils.

En un món multi plataforma es fa necessària l'existència de estàndards sòlids que evitin la necessitat de desenvolupar aplicacions específiques segon el dispositiu destinatari.

² “siE[12” La Sociedad de la información en España 2012. [TELEFONICA2012]

Segons el NIST (National Institute of Standards and Technology), el *cloud computing* o computació en el núvol, és un model tecnològic que permet l'accés ubic adaptat i sota demanda en xarxa amb un conjunt compartit de recursos.

Les solucions *cloud*, aporten una sèrie d'avantatges segons el col·lectiu o sectors als que pertanyi l'usuari. Les empreses eviten assumir inversions i la solució es pot redimensionar fàcilment segons les seves necessitat, sempre disposen de la darrera actualització del software, encara aquests destacables avantatges les empreses desconeixen en gran mesura el *cloud computing*.



Imatge 3 Coneixement del cloud computing en pimes espanyoles

Segons l'estudi "La sociedad en red 2011. Edició 2012" editat pel Ministeri d'Indústria [SOCIEDADRED] el 77.5% de les empreses que coneixen de l'existència del *cloud*, no han emprat mai cap solució basada en *cloud computing*.

Per als ciutadans, el *cloud* millora l'accés als serveis personals i a la gestió de la informació, de fet el *cloud computing* ha facilitat serveis i plataformes que en poc temps s'han convertit en moviments de canvi social, els exemples més evidents són la utilització de les xarxes socials, l'emmagatzematge virtual de correus electrònics o

l'emmagatzemament de contingut multimèdia, la ofimàtica en el núvol o fins i tot continguts basats íntegrament en el núvol com la música o la televisió.

Les xarxes socials a internet sorgeix amb força i contribueix al desenvolupament de nous espais de comunicació i noves formes de fer negocis en xarxa. Les principals aportacions de les xarxes socials consisteix en donar un espai als usuaris que els permeti ampliar les relacions que ja es duen a terme en el món real.

Una de les principals claus de les xarxes socials és la immediatesa de la informació i per aquest motiu els usuaris empren aquestes xarxes per comunicar-se amb amics o familiars, compartir informació o consultar informació sobre diversos temes.

1.1. Motivació

Amb la disponibilitat dels *smartphones* les formes de compartir informació, així com la disponibilitat dels mateixos com a eina, resulta sorprenent, des de emprar el telèfon per fer una fotocòpia d'un petit document, per tal de fer-ne la consulta a posteriori, o com la utilització del telèfon mòbil per saber si una taula està en equilibri emprant el telèfon com a *nivell*, saber on tenim la farmàcia més propera perquè el telèfon mòbil incorpora GPS i plànol o saber a quina altura del nivell del mar ens trobem o cap a on està el nord.

Totes aquestes activitats que fins ara feien necessària la utilització d'altres eines o estris, actualment estan en la majoria de les butxaques de tothom i és evident que podem aprofitar-ho per compartir informació o fins i tot per emprar el núvol que abans comentàvem, com a font per deixar-nos recordatoris, fins i tot recordatoris ubicats en un espai, en un lloc.

Actualment les xarxes socials només incorporen una mesura, que és el contingut ja sigui en format text o en format multimèdia, però una cosa és clara, falta una dimensió que molts cops resulta fonamental per entendre una imatge, un text o fins i tot un vídeo, aquesta dimensió, precisament l'espacial que encara que les xarxes socials la proveeixen com una possibilitat, no és la principal del missatge.

Per exemple, una escultura de gel o un *grafiti* espectacular, un incendi, un plat en un restaurant, aquests elements com a imatge, com a vídeo, com a expressió de les sensacions que produeixen al qui les percebi, estan molt bé, però la motivació va més enllà, quan passem per davant d'un restaurant i veiem que en aquell espai, hi ha un plat que algú ha compartit prèviament. O veiem que un determinat dia, algú va publicar per allí on estem d'excursió, una escultura natural feta de gel.

Es tracta doncs de mirar la GEO localització del missatge, com a fil conductor del propi missatge i no com es fa l'enfocament actual, on la GEO localització és comporta com un element informatiu més a la informació del propi missatge (text o multimèdia).

Aquesta filosofia és similar a la que es segueix en l'aplicació *Foursquare*, encara que aquesta aplicació comporta altres elements afegits com els anomenats *check-in* i *check-out* que permeten fer un seguiment de l'usuari amb l'objectiu d'emprar-la com a eina de fidelització o estratègia de marketing. [GEOLFOURSQUARE] i [MARKFOURSQUARE].

Aquest projecte vol doncs afegir a les notícies una nova dimensió, la GEO localització, com a eina per un diari, per saber les notícies que estan succeint en un lloc determinat, directament a través dels seus lectors, o per part d'un ajuntament que vol saber amb exactitud quins desperfectes hi ha a la seva ciutat, a través dels seus ciutadans.

1.2. Objectius

Amb aquest projecte volem familiaritzar-nos amb el desenvolupament sobre entorns mòbils i amb la utilització de estàndards sòlids que permetin una aplicació multi plataforma.

Un altre objectiu a aconseguir amb aquest projecte, és el desenvolupament d'una API que permeti als usuaris de l'aplicació comunicar-se amb el cloud.

Tant mateix caldrà que l'aplicació es comuniqui de forma eficient amb el *cloud* propi per obtenir dades, com són els punts d'interès (*POIs*) o amb el *cloud* extern per obtenir els mapes, en aquest cas, emprant les *API* de *openstreetmap*.

Per aconseguir aquests objectius el projecte es dividirà fases, en la primera es perseguirà l'objectiu d'aconseguir una eina mòbil que permeti mostrar per pantalla, la ubicació del telèfon mòbil, amb els punts d'interès que es trobin en el seu radi d'acció.

En una segona fase es desenvoluparà una *API* pròpia per a l'aplicació, que permeti la consulta dels punts d'interès, la publicació dels mateixos, etc..

I per últim s'integrarà l'aplicació mòbil amb la *API* del *cloud*, per mostrar la informació en els dispositius mòbils.

1.3. Estructura del document

Aquest document es divideix en dos grans apartats, **el capítol d'anàlisi** en que s'estudien les diferents tecnologies que podrien utilitzar-se en l'aplicació i ens permetrà veure'n els seus avantatges e inconvenients. Les tecnologies que s'analitzen es veuran des de diverses vessants, en una vessant hi tindriem el telèfon mòbil, on analitzarem els *standards* a seguir per tal d'aconseguir una aplicació multi plataforma.

En una altra vessant de la fase d'anàlisi, veurem l'anàlisi de les tecnologies a emprar des del punt de vista del servidor que haurà de proveir de dades als clients mòbil, en aquesta mateix vessant es farà un anàlisi del llenguatge d'intercanvi de dades que haurem d'implementar per comunicar l'aplicació client amb el servidor.

L'altre gran apartat d'aquest document, serà **el capítol de desenvolupament**, en aquest, es detallarà la planificació seguida per implementar els serveis i desenvolupar el producte mòbil, així com també es detallaran els diferents processos duts a terme per la implantació del servidor de recursos. Comentant els elements més importants per al desenvolupament final del producte. Al final d'aquest apartat també es podran trobar algunes de les proves realitzades amb el producte funcionant sobre un *smartphone*.

En un darrer capítol de **conclusions i línies futures**, analitzarem les possibilitats de l'eina desenvolupada així com les possibles línies futures que permetrien millorar o ampliar el producte generat.

Per últim el document incorpora un apartat en que es descriuen l'opinió sobre el projecte dut a terme i les experiències recollides durant el seu desenvolupament.

CAPÍTOL 2

1. ANÀLISI TECNOLÒGIC ENTORN MÒBIL
2. ANÀLISI TECNOLÒGIC WEBSERVICES

2. Capítol 2. Anàlisi.

2.1. Anàlisi tecnològic entorn mòbil

Dins d'aquest apartat analitzarem les diferents tecnologies segons el seu àmbit d'aplicació, ja que el projecte treballa sobre dos aspectes diferents, el primer i potser el més important els dispositius mòbils i les diferents plataformes, *iOS*, *Android*, *RIM*, *BADA*, *Symbian*, *WebOS*, *Windows Phone*, etc... I per una altra banda, caldrà veure quines seran les necessitats per desenvolupar una API que permeti integrar tots els clients mòbils.

Actualment en el món dels dispositius mòbils existeix força fragmentació en tots els nivells, en la següent graella es pot observar les diferències en quan a llenguatges de programació i entorns de programació segons les plataformes mòbils més habituals.

Mobile OS	Operating System	Software/IDEs	Programming Language
iOS	Mac only	Xcode	Objective C
Android	Windows/Mac/Linux	Eclipse/Java/Android Development Tool (ADT)	Java
BlackBerry	Windows mainly	Eclipse/JDE, Java	Java
Symbian	Windows/Mac/Linux	Carbide.c++	C++
WebOS	Windows/Mac/Linux	Eclipse/WebOS plugin	HTML/JavaScript/C++
Windows 7 Phone	Windows mainly	Visual Studio 2010	C#, .NET, Silverlight or WPF

Taula 1 Requeriments per al desenvolupament segons dispositiu

Aquesta fragmentació també esta donada per la disponibilitat dels sensors dels propis dispositius mòbils, aquesta característica fa que els mateixos sistemes estiguin limitats en funcionalitats. Tal i com es pot veure en la taula següent.

	 iPhone / iPhone 3G	 iPhone 3GS and newer		 OS 4.6-7	 OS 5.x	 OS 6.0+			
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✗	✗	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✗	✗
CONTACTS	✓	✓	⚠	✗	✓	✓	✗	✓	✓
FILE	✗	✗	✓	✗	✓	✓	⚠	✗	✗
GEO LOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA (AUDIO RECORDING)	⚠	⚠	✓	✗	✗	✗	✗	⚠	✗
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✗
NOTIFICATION (VIBRATION)	✓	✓	✓	✓	✓	✓	✗	✓	✓
STORAGE	✓	✓	⚠	✗	⚠	✓	✓	✗	✗

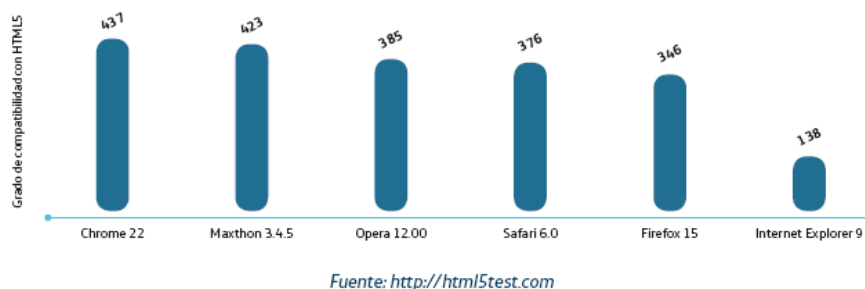
Taula 2 Fragmentació segons característiques i OS's mòbils

Aquests fets presenten una dificultat afegida a qualsevol producte mòbil, la necessitat de disposar de tants equips de desenvolupament com famílies de dispositius mòbils vulguem proveir.



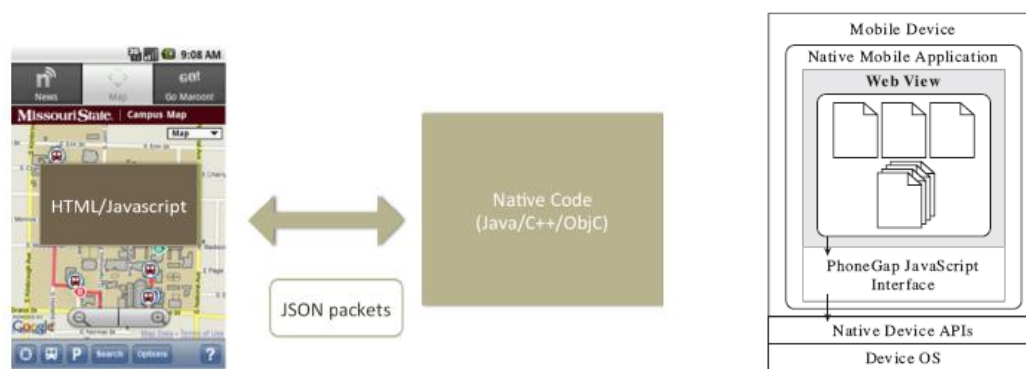
Imatge 4 Multiples equips per diferents OSs suposa nous problemes

Actualment darrera de qualsevol dispositiu mòbil podem trobar-hi un navegador i aquests utilitzen els estàndards W3C en diferents mesures, sempre subjectes al navegador en particular, aquest fet passarà a ésser el principi de la solució, fins i tot en el punt de vista dels PCs, els navegadors també segueixen el mateix estàndard, amb la qual cosa podem obtenir un producte basat en un sol llenguatge i que proveeixi a tots els dispositius, igual que fa el web amb els navegadors, basant-nos en el estàndard *HTML5/CSS3*.



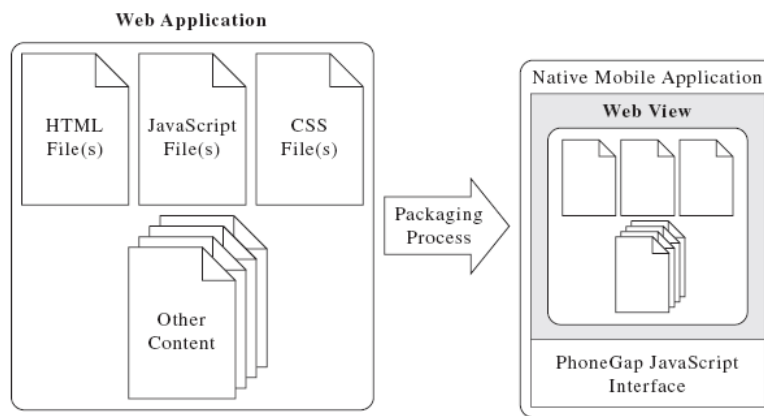
Imatge 5 Grau de compatibilitat dels navegadors amb HTML5 (sobre 500 pts)

Partint d'aquest inici i que tots els dispositius són capaços d'*embedir* una vista web, podríem obtenir comunicar una interfície *HTML/Javascript/CSS* amb el codi nadiu del dispositiu.



Imatge 6 Comunicació JavaScript i Codi natiu OS

Aquesta és la funcionalitat de *PhoneGap* (<http://www.phonegap.com>) permet el desenvolupament d'una *app* emprant com a llenguatge *HTML5/JavaScript/CSS* i mitjançant crides *Javascript* comunicar-nos amb el dispositiu i els seus sensors. Això ens permet desenvolupar una sola aplicació i de forma automàtica disposar de la mateixa *app* per a les plataformes *iPhone*, *Android*, *Blackberri* i *webOS*. *PhoneGap* permet accedir als sensors de la major part de *smartphones* actuals:



Imatge 7 Empaquetat d'una aplicació PhoneGap

Des de la mateixa *API PhoneGap* podem accedir als diferents components nadius del dispositiu, si en disposen, en cas que no disposin d'algun d'aquests elements, la mateixa *API* en gestiona l'error, per fer-ne un tractament acurat a la nostra aplicació.

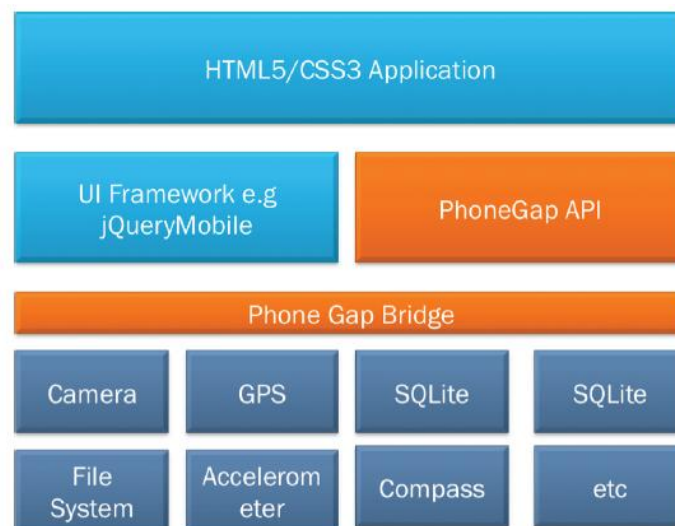
- Acceleròmetre
- Càmera
- Brúixola
- Contactes
- Arxius
- GEO localització
- Multimèdia (so, vídeo...)
- Xarxa
- Notificacions
- Emmagatzematge

A grans trets *PhoneGap API* disposa de diferents components que permeten l'accés a diferents sensors del dispositiu, tal i com es detalla a continuació.

- **Accelerometer** → Accedeix a l'acceleròmetre del dispositiu
- **Camera** → Captura una fotografia emprant la càmera del dispositiu
- **Capture** → Captura arxius multimèdia, so o vídeo emprant les aplicacions del propi dispositiu mòbil
- **Compass** → Obté la direcció del dispositiu
- **Connection** → Permet accedir a l'estat de la xarxa (wifi, 3G...)

- **Contacts** → Permet treballar amb els contactes del dispositiu
- **Device** → Permet obtenir informació específica del dispositiu
- **Events** → Permet capturar esdeveniments nadius del dispositiu a través de *Javascript*
- **File** → Permet interactuar amb el sistema de fitxers del dispositiu
- **Geolocation** → Permet obtenir les dades de GEO localització del dispositiu
- **Media** → Permet enregistrar i reproduir arxius àudio
- **Network** → Permet accedir a l'estat de la xarxa
- **Notification** → Permet activar notificacions a l'usuari (visuals, audibles, tàctils...)
- **Storage** → Permet accedir a les opcions d'emmagatzemament natives del dispositiu

Si ens veiem amb la necessitat de connectar l'aplicació amb un servei web, l'avantatge principal serà que podrem emprar les eines de *javascript*, llibreries com *jQuery* o *jQuery mobile*.



Imatge 8 Arquitectura d'una aplicació PhoneGap

Tot i les possibilitats que ofereix la pròpia *API* de *PhoneGap* podem trobar una necessitat que no estigui contemplada en el desenvolupament de la *API*, per aquest motiu existeix la possibilitat de desenvolupar *plugins* que ens permetin ampliar les funcionalitats de la *API*.

Per tant des del punt de vista tecnològic el desenvolupament de l'aplicació dins l'entorn mòbil s'ha dut a terme emprant *Phonegap* i per ajudar a la comunicació amb els serveis web que permetran la interacció entre els usuaris, s'ha emprat *jQuery Mobile*.

Dins de l'àmbit tecnològic una altra decisió important era la utilització dels mapes, actualment la referència per moltes aplicacions són els mapes que proporciona Google, aquests però tenen limitacions de llicència i d'actualització, per aquest motiu i per la filosofia de la pròpia aplicació, que és compartir, s'ha decidit utilitzar els mapes de caire lliure *openstreetmap*.



Imatge 9 OpenStreetMap. Mapa mundi. <http://www.openstreetmap.org>

Per tal de mostrar informació en els mapes en un entorn web com el que s'emprarà en l'aplicació era necessari disposar d'alguna llibreria que ens facilités la feina, si els mapes haguessin estat els propis de *google*, aquestes llibreries ja en les proporciona el propi *google*. Però en el nostre cas, va estar necessari fer una selecció de totes les llibreries que són capaces d'interactuar amb els mapes d'*openstreetmap*, en aquest cas varen estar les llibreries *leafletjs*.



Imatge 10 Llibreria JS Leafletjs. <http://leafletjs.com>

Dins les alternatives, també vaig avaluar les possibilitats que ofereix la llibreria *openlayer* [OPENLAYER210], però per la seva manca de maduresa en alguns aspectes, vaig desestimar l'opció en favor de *leafletjs* que era un producte més evolucionat, on moltes operacions geomètriques veien ja implementades i ja disposava de *plugins* per afegir de forma automatitzada marcadors, agrupar marcadors, etc.

2.1.1. PhoneGap. Introducció a la seva utilització

Per instal·lar *PhoneGap* en un entorn de desenvolupament serà necessària la descarrega del paquet que es correspondrà a la plataforma de desenvolupament (*iOS*, *Android*...) i integrar-lo dins l'entorn de programació. En el nostra cas comentarem l'entorn *Android*.

Per instal·lar *Phonegap* per *Android* serà necessari disposar de l'entorn Eclipse, tenir instal·lades i configurades les *SDK Android* i el *Android ADT Plugin*.

Quan ens descarreguem les *PhoneGap API* en el seu interior trobarem un arxiu *JAR* i un arxiu *JS* tots dos s'hauran d'ubicar en el projecte que vulguem desenvolupar en les carpetes */libs* i */assets/www* respectivament.

A partir d'aquí serà necessari modificar l'activitat principal del projecte Android per tal que es generi una activitat *DroidGAP*.

```
package org.examples.phonegap.helloworld;

import android.os.Bundle;

import com.phonegap.DroidGap;

public class HelloWorld extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

Imatge 11 Codi per l'Activity HelloWorld

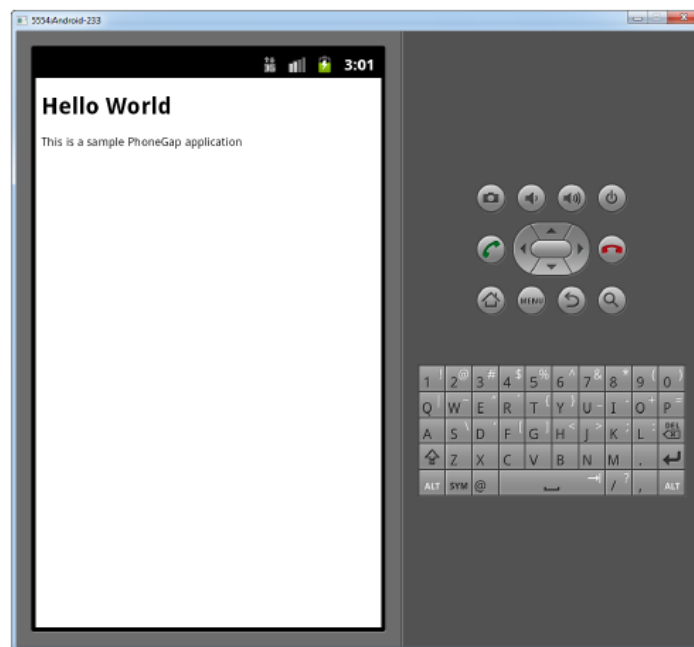
Amb aquest codi aconseguirem que l'aplicació Android s'iniciï amb el codi que es trobi dins de l'arxiu *index.html*.

HelloWorld1 Application

```
<!DOCTYPE HTML>
<html>
<head>
  <title>HelloWorld1</title>
</head>
<body>
  <h1>Hello World</h1>
  <p>This is a sample PhoneGap application</p>
</body>
</html>
```

Imatge 12 Codi HTML de la vista de l'app HelloWorld

Amb aquest codi obtindrem la següent aplicació per Android.



Imatge 13 Aplicació HelloWorld en funcionament

Amb aquest exemple no podríem accedir encara als sensors emprant les *API PhoneGap*, sinó que només podríem mostrar per pantalla una aplicació *embedida* desenvolupada en *HTML/Javascript/CSS*. Per accedir completament al potencial de *PhoneGap* seria necessari incloure la llibreria *javascript* de *Phonegap*, anomenada *cordova-x.x.x.js* segons la versió.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Uvumi</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Uvumi Android App by SIENSIS.com">
  <meta name="author" content="Josep Maria Flix Rovira">

  <!-- Styles -->
  <link rel="stylesheet" type="text/css" href="jquery.mobile/jquery.mobile-1.2.0.min.css"></link>
  <link rel="stylesheet" type="text/css" href="jquery.mobile/jquery.mobile.custom.icons.css"></link>
  <link rel="stylesheet" type="text/css" href="leafletjs/leaflet.css"></link>
  <link rel="stylesheet" type="text/css" href="leafletjs/MarkerCluster.css"></link>
  <link rel="stylesheet" type="text/css" href="leafletjs/MarkerCluster.Default.css"></link>

  <!-- JS Libraries -->
  <script type="text/javascript" charset="utf-8" src="cordova-2.4.0.js"></script>

  <script type="text/javascript" charset="utf-8" src="jquery.mobile/jquery-1.8.2.min"></script>
  <script type="text/javascript" charset="utf-8" src="jquery.mobile/jquery.mobile-1.2.0.min.js"></script>

  <script type="text/javascript" charset="utf-8" src="leafletjs/leaflet-src.js"></script>
  <script type="text/javascript" charset="utf-8" src="leafletjs/leaflet.markercluster-src.js"></script>

  <script type="text/javascript" charset="utf-8" src="app/map.js"></script>

  <script type="text/javascript" charset="utf-8">

```

Imatge 14 Uvumi App, importació de cordova.js

Un cop feta la inclusió de la llibreria només cal controlar en quin moment podem disposar de la API i per tant podem iniciar tot el procés d'accés als sensors del dispositiu.

```

</script>
</head>
<body onload="init();" >

```

Imatge 15 Codi d'inicialització de PhoneGap

```

function init() {
  $.mobile.loading('show');

  document.addEventListener("deviceready", onDeviceReady, true);
}

```

Imatge 16 Codi de la funció d'inicialització


```

var onDeviceReady = function() {
    document.addEventListener("backbutton", onBackButton);

    $.mobile.loading('show');
    if (loadConfig ()) {
        $.mobile.loading('show');
        login (function(){
            $.mobile.loading('hide');
            $.mobile.changePage("#page").

```

Imatge 17 Codi de controlador de l'esdeveniment DeviceReady de PhoneGap

2.2. Anàlisi tecnològic webservices

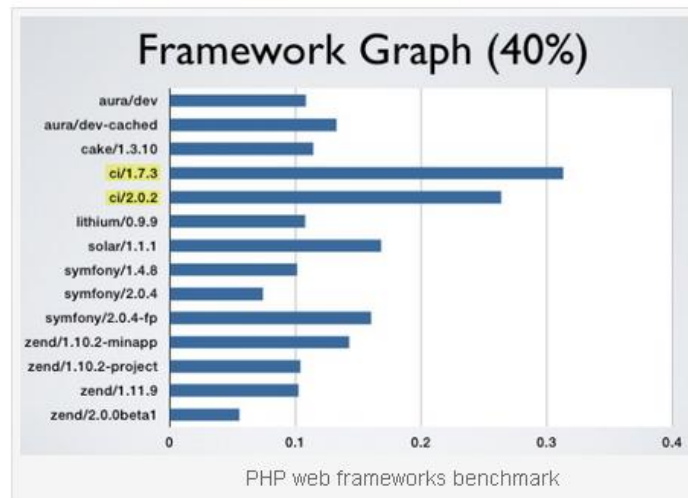
Per al desenvolupament dels *webservices* es va escollir un entorn basat en *Apache/MySQL* com a entorn de desenvolupament, des del punt de vista del servidor web, *Apache*, és un servidor de codi obert que aconsegueix els requeriments del nostre projecte i disposa multitud de mòduls que permeten adaptar-lo als diferents requeriments del projecte.

Des del punt de vista de les dades, el projecte es basa en *MySQL*, un sistema gestor de dades relacional, multi fil i multiusuari, aquest servidor està desenvolupat com a software lliure, encara que des de l'any 2009 Oracle l'ofereix sobre un sistema de llicència dual, de forma que si es vol incorporar *MySQL* en un software privatiu, cal comprar una llicència específica que permeti aquesta utilització.

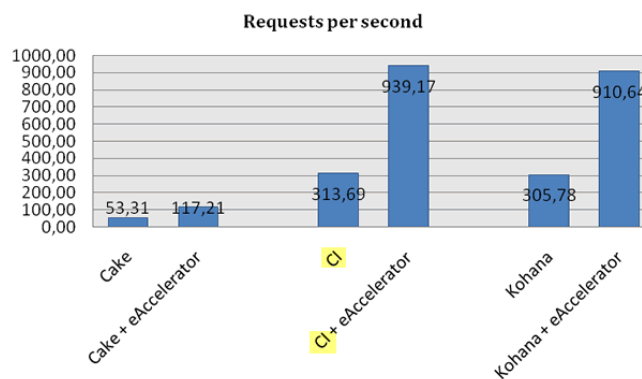
Una altra de les característiques que s'ha tingut molt en compte ha estat, que aquest gestor incorpora operacions i tipus que permeten la manipulació d'espais, per a aquest fet incorporen una classe anomenada *Geometry* [**GEOMETRYMySQL**] que incorpora objectes de tipus, punt, línia, polígon...

Com a llenguatge de programació des del punt de vista servidor s'ha emprat *PHP* i en concret un *Framework* de codi obert que permet crear llocs web dinàmics amb *PHP* i que es basa en el patró MVC (model, vista, controlador) [**PHPMVC**]. L'objectiu d'aquest *framework* és permetre que els desenvolupadors puguin dur a terme projectes més ràpidament, ja que l'entorn ja disposa de les llibreries més habituals i el propi

llenguatge permet accedir-hi de forma lògica i eficient. Existeixen diversos *frameworks* PHP però en la majoria de *benchmarks* el millor en rendiment resulta ser *Codeigniter*.



Imatge 18 Gràfica rendiment frameworks PHP



Imatge 19 Gràfica rendiment frameworks PHP

En el cas que ocupa l'actual projecte, només es desenvolupa l'entorn mòbil però en les línies futures en el moment de desenvolupar l'entorn web, utilitzaríem les característiques de les vistes.

El projecte que ens ocupa es va iniciar basant el seu desenvolupament en la filosofia REST, una tècnica àmpliament utilitzada en l'actualitat que representa l'alternativa als serveis SOAP basats en XML/HTTP.

En aquest cas els sistemes basats en REST es funcionen segons els següents fonaments:

- Un protocol client/servidor sense estat. Cada petició *HTTP* conté tota la informació necessària per comprendre la petició.
- Les operacions estan ben definides ja que el propi protocol *HTTP* ja defineix un petit conjunt d'operacions. En el nostre cas, aquest ha estat el taló d'Aquil·les i una limitació donada per la versió de *PhoneGap* no ens ha permès explotar en la seva plenitud. En les línies futures es podrà incorporar aquesta millora.
- La sintaxis és universal per identificar els recursos, ja que tot recurs té la seva pròpia *URI*

Aquestes característiques permeten el intercanvi d'informació entre recursos i servidors sense necessitar cap tipus d'infraestructura addicional. Actualment podem trobar diversos serveis basats en REST, com Twitter, Facebook, MEGA...

Era necessari comunicar l'aplicació mòbil amb la xarxa i per aquest motiu era necessari disposar d'un llenguatge d'intercanvi que facilités aquesta operació, per aquest motiu es va avaluar dos tecnologies d'intercanvi de dades, com són *XML* i *JSON*.

JSON és un estàndard obert basat en text que permet el intercanvi de dades, aquest llenguatge permet representar estructures de dades simples i llistes associatives, encara que és una derivació de *javascript* són molts els llenguatges de programació que en tenen implementacions.

La simplicitat de *JSON* ha permès la generalització de la seva utilització sobretot com a alternativa a *XML* en entorns *AJAX*. Un dels avantatges de *JSON* sobre *XML* com a format d'intercanvi de dades, és la simplicitat d'escriure un analitzador sintàctic o *parser* de *JSON*. Un altre fet que ha intervingut en la massificació de la seva utilització ha estat la ubiqüitat de *JavaScript* que està en qualsevol navegador web i això permet disposar d'un *parser* integrat.

XML també és un llenguatge de marques que ofereix moltes més eines de desenvolupament, però el fet que les estructures *JSON* siguin més compactes, que es disposi d'un *parser* integrat en el propi navegador, fa que aquesta última sigui l'opció

preferida per al desenvolupament *Ajax* i en el nostre projecte també, ja que el rendiment de l'aplicació mòbil basada en un navegador *embedit* serà major.



Imatge 20 Exemple uVumi GeoJSON

Per exemple en l'aplicació del projecte, el servidor retorna una col·lecció d'elements geogràfics basats en el estàndard *GeoJSON*, cada element de la col·lecció conté una apartat relatiu a les propietats del punt, dades de l'usuari, comentari, imatge publicada, data/hora de publicació, etc. I un altre apartat que incorpora les dades geogràfiques pròpiament dites emprant les coordenades terrestres longitud i latitud.

La taula següent ens mostra el conjunt de tecnologies emprades en el projecte en totes les vessants del mateix.

Tecnologia Mòbil	Tecnologia Servidor
PhoneGAP	PHP
HTML5	Codeigniter
CSS3	MySQL object Geometry
Javascript	JSON
jQuery	REST
jQuerymobile	API LeafletJS with GeoJSon
Ajax	
REST	
API LeafletJS	

Taula 3 Resum de tecnologies emprades en el projecte

CAPÍTOL 3

1. CRONOGRAMA
2. ARQUITECTURA DEL SISTEMA
3. RESULTATS OBTINGUTS

3. Capítol 3. Desenvolupament.

3.1. Cronograma

El projecte es divideix en les tasques següents, seguint la planificació.

- Anàlisi de requeriments
 - Anàlisi de funcionalitats
- Anàlisi tecnològic
 - Anàlisi tecnologies entorn mòbil
 - Proves entorn *PhoneGAP*
 - Anàlisi tecnologies de servidor
 - Proves entorn *Yii Framework*
 - Proves entorn *CodeIgniter Framework*
 - Proves entorn *NOCMS Framework*
 - Anàlisi de requeriments a nivell geogràfic, funcions *MySQL...*
 - Proves integració tecnologies *LeafletJS – Codeigniter*
- Desenvolupament de l'aplicació
 - Aplicació mòbil
 - Disseny aplicació mòbil
 - Desenvolupament aplicació mòbil
 - Adaptació aplicació mobil/*webservices*
 - Test aplicació amb dades estàtiques
 - Aplicació *uVumi Server*
 - Disseny suport dades
 - Desenvolupament dels *webservices*
 - Integració desenvolupament mòbil/servidor
- Test de l'APP
 - Proves de servei
 - Test de seguretat i fiabilitat
 - Proves en entorn real
- Desenvolupament de la memòria

3.2. *Previsió econòmica*

La present previsió econòmica per al desenvolupament del projecte s'ha calculat en base a la planificació realitzada i a les actuacions dutes a terme, així com en les costos de contractació d'un servidor per als *WebServices* adient per suportar una demanda important de dispositius, tant per la quantitat d'emmagatzemament, com per disponibilitat i capacitat, encara que en aquests moment més embrionaris s'ha disposat d'un servidor de característiques inferiors.

Amb tot dins de l'apartat de servidor, s'hauria de valorar la necessitat d'un servidor amb majors prestacions o possibilitats d'escalabilitat, valorant fins i tot plataformes tipus *Amazon WebServices* (AWS), ja que aquesta plataforma podria ser útil en cas créixer les necessitats de l'aplicació per un augment dels dispositius client o pel nombre de peticions dels mateixos. AWS és una plataforma que permet diferents tipus de suport a gran escala:

- **Processament de dades:** Sistemes escalables amb pagament segons consum o balanceig automàtic segons les necessitats que es defineix a la plataforma i segons les necessitats de cada moment.
- **Lliurament de contingut:** Facilita el lliurament del contingut amb una baixa latència mitjançant diverses ubicacions proveïdes per Amazon WebServices.
- **Bases de dades:** Permet gestionar dades emmagatzemades en els servidors AWS simplificant les tasques de gestió, configuració i escalat, emprant les eines del núvol de *Amazon*, caldria adaptar la capa de dades del nostre *WebService*.
- **Emmagatzemament:** AWS permet un emmagatzemament redundant que permet l'emmagatzematge i a la recuperació de qualsevol arxiu en qualsevol moment així com la extremada escalabilitat del sistema, adaptant-se a les necessitats del producte en qualsevol moment.

Amazon Web Services proveeix altres eines que poden resultar útils en multitud de productes, però en el cas d'aquest projecte cal tenir-lo en compte per futures extensions o per necessitats de requeriment, ja que AWS té un alt grau d'escalabilitat en tots els aspectes. [AWS]

DESENVOLUPAMENT I IMPLANTACIÓ				
CONCEPTE	DIES	HORES/DIA	€/HORA	TOTAL
Anàlisi funcionalitats	13	4	45	2.340,00 €
Proves entorn PhoneGAP	8	4	45	1.440,00 €
Proves Yii Framework	6	4	45	1.080,00 €
Proves entorn CodeIgniter Framework	10	4	45	1.800,00 €
Proves entorn NOCMS Framework	5	4	45	900,00 €
Anàlisi funcions geogràfiques	2	4	45	360,00 €
Test/integració API LeafletJS-Codeigniter	9	4	45	1.620,00 €
Desenvolupament aplicació phone	20	4	45	3.600,00 €
Desenvolupament WebServices	15	4	45	2.700,00 €
Adaptació APP to Webservices	5	4	45	900,00 €
			Subtotal:	16.740,00 €
EQUIPAMENT SERVIDOR				
CONCEPTE	UNITATS	COST ANUAL		TOTAL
Servidor dedicat Linux Strato	1	468		468,00 €
TOTAL COST PROJECTE			TOTAL:	17.208,00 €
			IVA (21%):	3.613,68 €
			TOTAL IVA INCLÓS:	20.821,68 €

Taula 4 Valoració econòmica del projecte

Un altre aspecte que no es pot obviar és que a part del desenvolupament del producte, el projecte tindrà associats uns costos de manteniment, no solament del servidor on s'emmagatzemaran les dades.

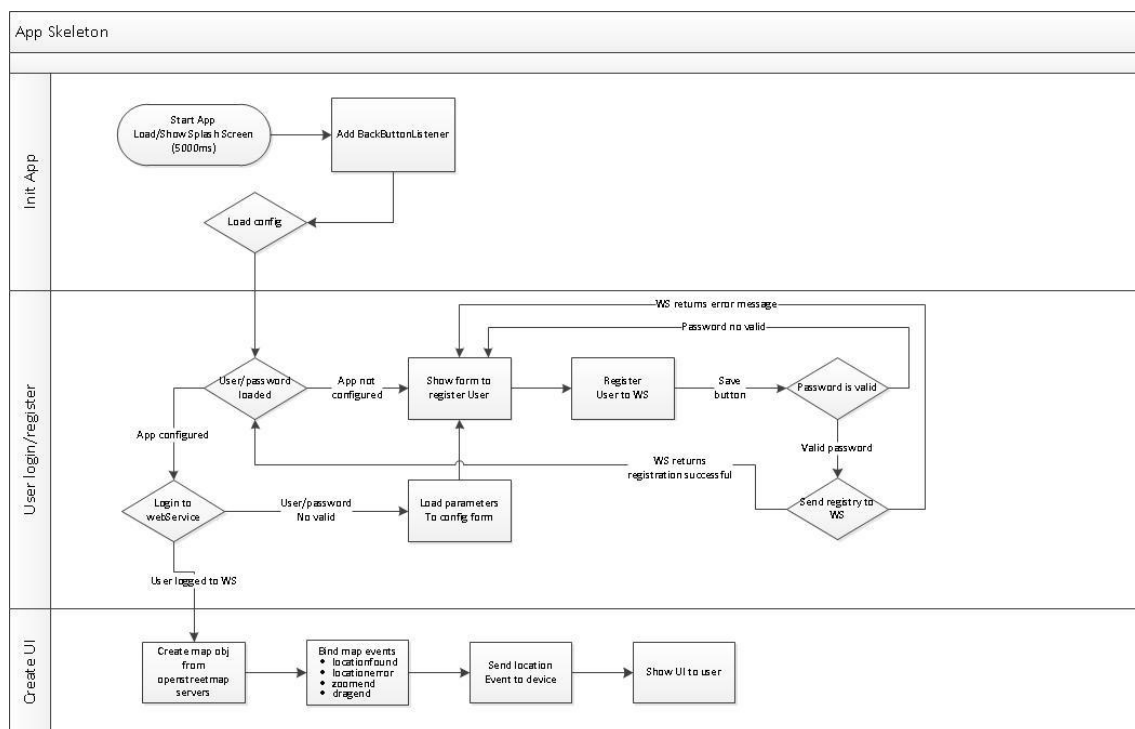
MANTENIMENT ANUAL			
CONCEPTE	UNITATS	COST ANUAL	TOTAL
Servidor dedicat Linux Strato	1	468	468,00 €
Adaptació i correcció d'errors segons el feedback dels usuaris	ND	ND	ND
Tasques de manteniment de les dades, còpies de seguretat, actualització i manteniment de servidors	ND	ND	ND
Ampliació de funcionalitats i actualitzacions de la APP per tal d'actualitzar-la segons requeriments dels usuaris	ND	ND	ND

Taula 5 Valoració econòmica del manteniment anual del projecte

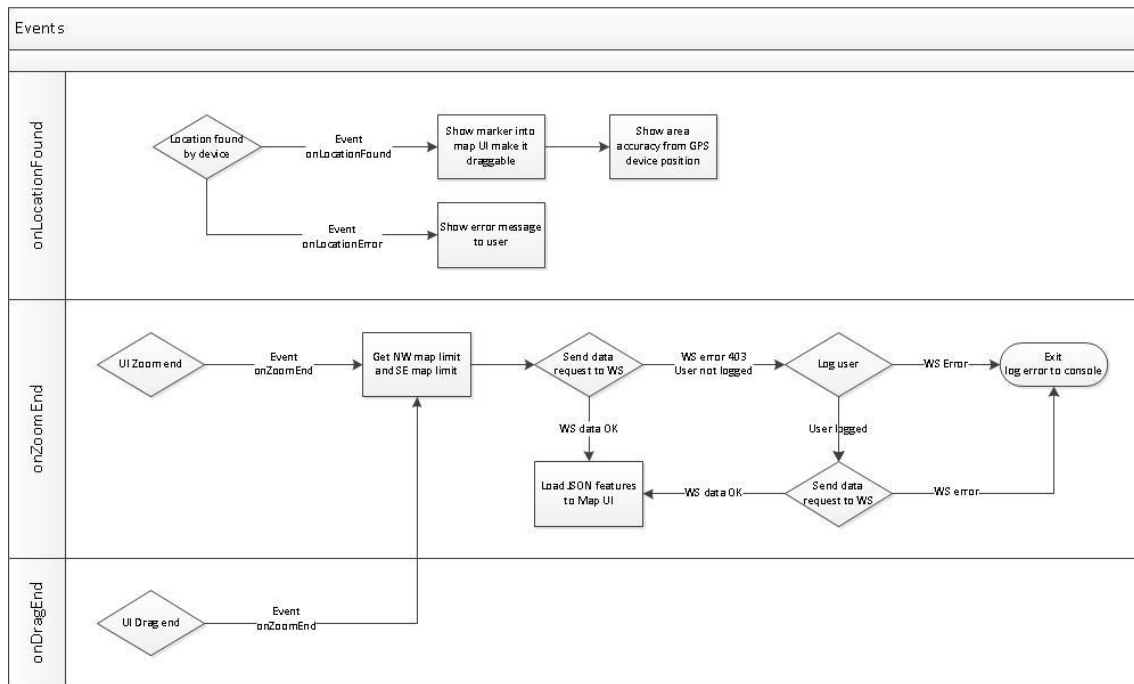
Aquestes tasques de manteniment anirien subjectes a la utilització per part dels usuaris, fins i tot, caldria tenir en compte que segons aquest ús, caldria migrar el servidor a d'altres servidors més adients o fins i tot seria necessària la disponibilitat d'un servei d'atenció al client per tal de resoldre les incidències que poguessin sorgir de la utilització de la mateixa aplicació. Per tot plegat, fer una estimació sobre les despeses associades al futur del producte resulta complicat sense disposar d'una demanda real, sense oblidar a més la necessitat d'adaptació de l'APP a les novetats socials i a les demandes dels usuaris.

3.3. *Arquitectura del sistema*

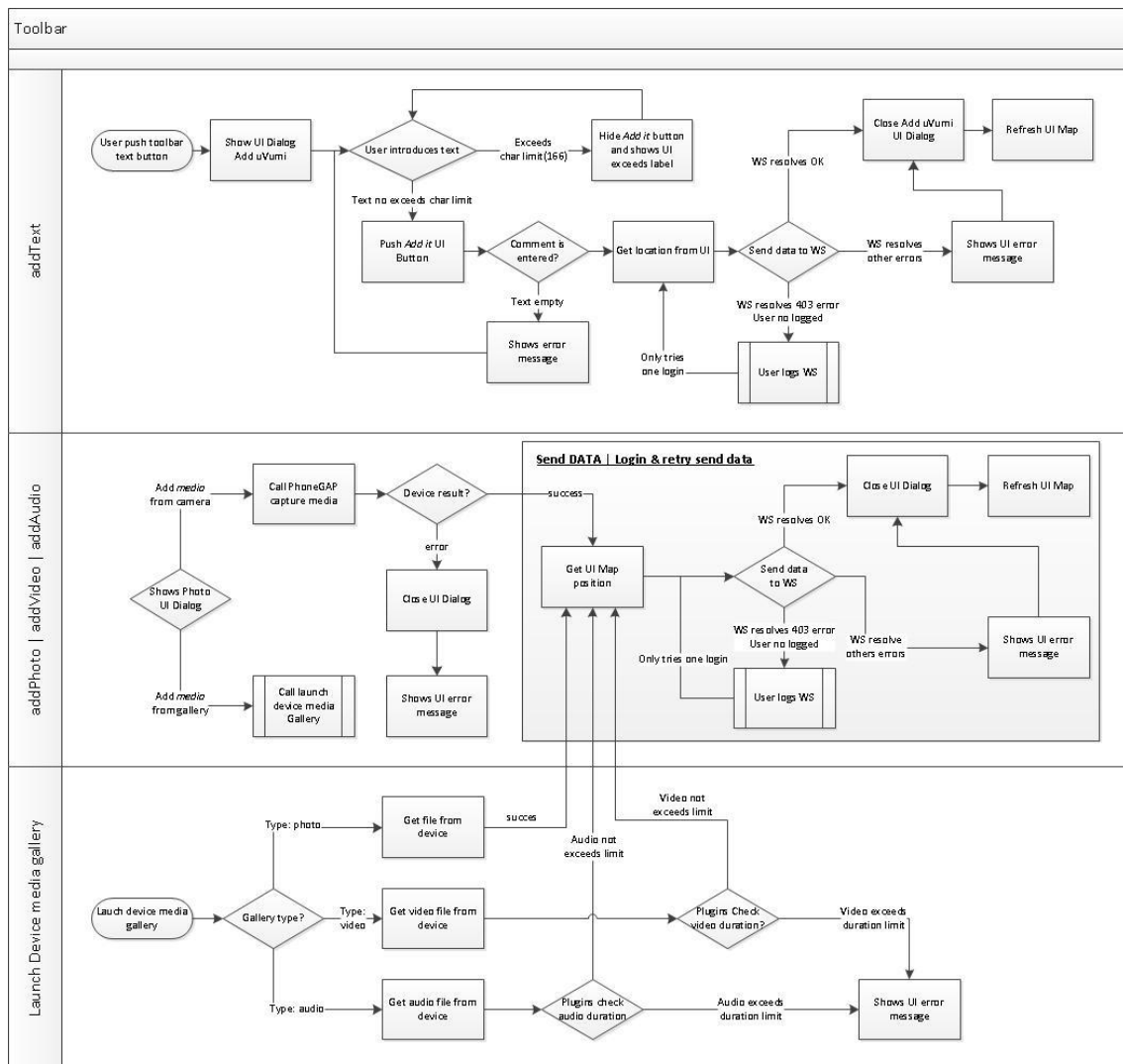
3.3.1. Diagrames d'estat de l'aplicació



Imatge 22 Esquelet general de l'aplicació



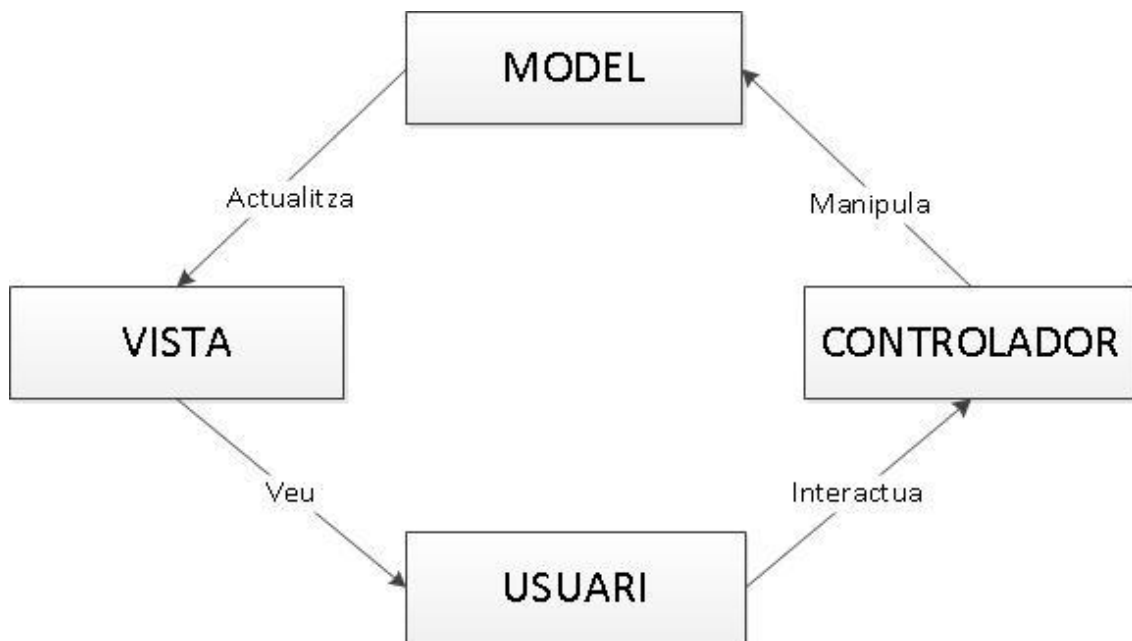
Imatge 23 Gestió events de l'aplicació



Imatge 24 Estats toolbar de publicació

3.3.2. Descripció de funcionament de l'aplicació

L'aplicació ha estat dissenyada seguint el patró de l'arquitectura del software que separa les dades de la lògica de negoci, o sigui per una banda hi ha uns components que representen la informació i per una altra la interacció amb l'usuari, aquest patró de disseny permet la reutilització de codi i la separació de conceptes que faciliten el desenvolupament de les aplicacions i el seu manteniment.



Imatge 25 Diagrama de funcionament del patró MVC

El *model* és l'encarregat de la representació de la informació amb la que opera el sistema i en gestiona el seu accés, ja sigui per fer-ne consultes com per fer-ne qualsevol modificació. El *controlador* respon als esdeveniments de l'usuari, en el nostre cas als esdeveniments de dispositiu i fa les crides al model per si cal modificar alguna dada, fent d'intermediari entre la vista i el model. La *vista* en canvi presenta la informació a l'usuari de forma adient, en el nostre cas la vista es presenta en la banda del dispositiu, mitjançant l'aplicació *Phonegap* desenvolupada emprant *HTML* i *javascript*.

3.3.3. Funcions del servidor (Controller+Model)

CONTROLLER public function registeruser ()	
Paràmetres:	Cap
Descripció:	<p>Aquesta funció obté les dades enviades mitjançant <i>POST</i> de l'aplicació mòbil i després de comprovar-ne la integritat afegeix la conta d'usuari dins la <i>BBDD</i>.</p> <pre> \$this->form_validation->set_rules('username', 'Username', 'required alpha_numeric is_unique[uwumi_users.username]'); \$this->form_validation->set_rules('password', 'Password', 'required min_length[6]'); \$this->form_validation->set_rules('email', 'eMail', 'required valid_email'); \$this->form_validation->set_rules('realname', 'Real name', 'required alpha'); </pre>
Retorna:	<p>Retorna una estructura <i>JSON</i> formada per tres camps: status, server i options.</p> <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i>. <ul style="list-style-type: none"> ○ 200 si no s'ha produït cap error ○ 400 en cas d'error de validació o error d'inserció a la base de dades • <i>server</i>: Missatge descriptiu del retorn per part del servidor, iniciat per OK o per ERR. • <i>options</i>: En aquest cas el retorn està format per: <ul style="list-style-type: none"> ○ <i>username</i>: usuari que s'ha intentat afegir o afegit, en cas que es produeixi un error de validació de les dades, aquesta informació no es retorna en la resposta al client ○ <i>opResult</i>: Missatge descriptiu més concret de l'operació realitzada i del resultat de la mateixa

CONTROLLER public function login ()	
Paràmetres:	Cap
Descripció:	<p>Obté les dades d'usuari (usuari i contrasenya) mitjançant post i després de comprovar-ne la integritat, les valida a la base de dades i inicia sessió per al dispositiu demandant.</p>
Retorna:	<p>Retorna una estructura <i>JSON</i> formada per tres camps: status, server i options.</p> <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i>. <ul style="list-style-type: none"> ○ 200 si no s'ha produït cap error i si l'usuari ha fet login correctament ○ 403 usuari o contrasenya errònia ○ 400 error de validació de les dades • <i>server</i>: Missatge descriptiu del retorn per part del servidor, iniciat per OK o per ERR segons sigui el cas

	<ul style="list-style-type: none"> • <i>options</i>: Aquest camp només es retorna en els casos 200 i 403 i està format per: <ul style="list-style-type: none"> ○ <i>username</i>: usuari que ha intentat fer login ○ <i>date (200)</i>: data d'inici de sessió ○ <i>password (403)</i>: contrasenya amb la que s'ha intentat el login
--	--

CONTROLLER public function logout ()	
Paràmetres:	Cap
Descripció:	Aquesta funció finalitza la sessió de l'usuari actual per al dispositiu sol·licitant
Retorna:	Retorna una estructura <i>JSON</i> formada per tres camps: status, server i options. <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i>. <ul style="list-style-type: none"> ○ 200 conforme s'ha eliminat la sessió del servidor • <i>server</i>: Missatge descriptiu del retorn per part del servidor, iniciat per OK • <i>options</i>: Està format per: <ul style="list-style-type: none"> ○ <i>username</i>: usuari que ha finalitzat la sessió

MODEL public function getBoundary (lat1, lon1, lat2, lon2, [byUser])	
Paràmetres:	<ul style="list-style-type: none"> • <i>lat1 , lon1</i>: Coordenada terrestre corresponent a la posició superior esquerra del rectangle a obtenir les dades • <i>lat2 , lon2</i>: Coordenada terrestre corresponent a la posició inferior dreta del rectangle a obtenir les dades • <i>[byUser]</i>: Paràmetre opcional que permet filtrar només les dades d'un sol usuari, si no s'especifica retorna per qualsevol usuari
Descripció:	Aquesta funció retorna una seqüència <i>GeoJSON</i> amb les dades dels punts d'interès que es troben dins del rectangle passat per paràmetres
Retorna:	En cas que es produeixi un error, retorna una estructura <i>JSON</i> formada per tres camps: status, server. <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i>. <ul style="list-style-type: none"> ○ 400 No existeix l'usuari o el rectangle de coordenades no està correctament definit ○ 403 No existeix sessió d'usuari per al dispositiu sol·licitant • <i>server</i>: Missatge descriptiu del retorn per part del servidor, iniciat per OK o ERR segons sigui necessari En cas que no s'hagi produït cap error retorna una col·lecció

	<p>d'elements que descriuen el punt a ubicar en el mapa en format <i>GeoJSON</i>, aquest punt esta format per:</p> <ul style="list-style-type: none"> • <i>Coordenades</i>: latitud i longitud • <i>Usuari</i>: Usuari que ha publicar el punt • <i>Tipus</i>: Tipus de punt que es tracta, text, vídeo, àudio • <i>Data</i>: Data de publicació • <i>Hora</i>: Hora de la seva publicació • <i>Icona</i>: Icona a mostrar dins l'aplicació • <i>html</i>: Codi html a mostrar al fer clic sobre el punt <pre> \$features[] = array("type" => "Feature", "properties" => array("thumb" => \$fileicon, "comment" => \$htmlComment, "username" => \$row['username'], "type" => \$row['type'], "date" => \$sData, "time" => \$sHora, "lon" => \$row['lon'], "lat" => \$row['lat'],), "geometry" => array("type" => "Point", "coordinates" => array(\$row['lon'], \$row['lat']))); </pre>
--	---

CONTROLLER public function addtext ()	
Paràmetres:	Cap
Descripció:	<p>Mitjançant <i>POST</i> s'obtenen les dades del dispositiu i després de validar-ne la seva correctesa es publiquen. Aquestes dades són:</p> <ul style="list-style-type: none"> • Coordenades geogràfiques de la ubicació del comentari • Comentari a publicar
Retorna:	<p>Retorna una estructura <i>JSON</i> formada per tres camps: status, server i options.</p> <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i>. <ul style="list-style-type: none"> ○ 200 S'ha afegit correctament el comentari al mapa compartit ○ 400 S'ha produït un error en la validació de les dades enviades pel dispositiu ○ 403 No existeix sessió d'usuari per al dispositiu sol·licitant • <i>server</i>: Missatge descriptiu del retorn per part del servidor, iniciat per OK o ERR segons sigui el cas • <i>options</i>: Només s'envia en el cas 200 i està format per: <ul style="list-style-type: none"> ○ <i>username</i>: usuari que ha inserit el comentari ○ <i>comment</i>: comentari que s'ha publicat ○ <i>lat i lon</i>: coordenades geogràfiques on s'ha publicat el missatge

	<pre> \$result = array('status' => '200', 'server'=> 'OK. Comment added', 'options' => array('lat' => \$lat, 'lon' => \$lon, 'comment' => \$comment, 'username' => \$username,),); </pre>
--	--

CONTROLLER public function addmedia (type)	
Paràmetres:	<p>El paràmetre <i>type</i> indica a la funció quin tipus de missatge es va a publicar:</p> <ul style="list-style-type: none"> • audio • video • photo
Descripció:	<p>Aquesta funció publica en el mapa compartit un arxiu multimèdia, el format d'aquest arxiu ve indicat per paràmetres, la resta de dades es reben mitjançant <i>POST</i>:</p> <ul style="list-style-type: none"> • Coordenades geogràfiques del punt • Arxiu multimèdia
Retorna:	<p>Retorna una estructura <i>JSON</i> formada per tres camps: status, server i options.</p> <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i>. <ul style="list-style-type: none"> ○ 200 S'ha afegit correctament l'arxiu multimèdia al mapa compartit ○ 400 S'ha produït un error en la validació de les dades enviades pel dispositiu: <ul style="list-style-type: none"> ▪ format d'arxiu no vàlid, tipus d'arxiu no suportat ▪ s'ha excedit la grandària màxima de publicació d'arxiu ▪ falten les coordenades d'ubicació geogràfica per a l'arxiu <pre> // Check AJAX validity passed to WS \$valid = \$this->form_validation->required(\$lat) && \$this->form_validation->required(\$lon) && \$this->form_validation->required(\$username); elseif (\$type == "photo") { \$config['allowed_types'] = 'jpg'; \$config['encrypt_name'] = TRUE; \$config['xss_clean'] = TRUE; } elseif (\$type == "video") { \$config['allowed_types'] = '3gp mp4'; \$config['max_size'] = \$this->UWI_MAX_FILE_SIZE; \$config['encrypt_name'] = TRUE; \$config['xss_clean'] = FALSE; } </pre>

	<pre>elseif (\$type == "audio") { \$config['allowed_types'] = 'amr m4a'; \$config['max_size'] = \$this->UWI_MAX_FILE_SIZE; \$config['encrypt_name'] = TRUE; \$config['xss_clean'] = FALSE;</pre> <ul style="list-style-type: none"> ○ 403 No existeix sessió d'usuari per al dispositiu sol·licitant ○ 503 Altres errors produïts a l'emmagatzemar l'arxiu en el servidor per part del <i>webService</i> • <i>server</i>: Missatge descriptiu del retorn per part del servidor, iniciat per OK o ERR segons sigui el cas
--	--

CONTROLLER public function thumbs ([imgname], [imgext])	
Paràmetres:	Aquesta funció rep com a paràmetre, el nom de la imatge i la extensió de la mateixa, la única permesa és jpg
Descripció:	La funció retorna la imatge demanada que es troba emmagatzemada en el servidor, en cas que es faci una sol·licitud incorrecta o l'arxiu no existeixi, el servidor retorna una imatge d'un comentari <i>uVumi</i> buit
Retorna:	<p>Retorna una imatge buida si no existeix o es produeix un error de petició o la imatge demanada.</p> <pre>if (\$imgname==null \$imgext==null !file_exists(\$nombre_archivo)) \$this->emptyThumb(); else { ■ ■ ■</pre> <p>Tot evitant els controls de cache que es produeixen en els navegadors, inserint en la capçalera de retorn <i>http</i>, els codis de caducitat pertinents</p> <pre>header("Expires: Tue, 03 Jul 2001 06:00:00 GMT"); header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT"); header("Pragma: no-cache");</pre>

CONTROLLER private function checkLoggedUser (msg)	
Paràmetres:	Es passa per paràmetre el missatge d'error que s'ha d'incloure a l'estructura de retorn que s'enviarà posteriorment al dispositiu
Descripció:	Aquesta funció comprova si s'ha iniciat sessió en el servidor prepara l'estructura de dades que es retornarà al dispositiu de l'usuari
Retorna:	<p>Retorna una estructura <i>JSON</i> formada per tres camps: status, server. Aquesta estructura no es retorna directament al dispositiu sinó que es retorna a la funció del servidor que la crida per a que gestioni correctament l'enviament.</p> <ul style="list-style-type: none"> • <i>status</i>: Codi <i>HTTP</i> 403

	<ul style="list-style-type: none"> <i>server</i>: El missatge que ha rebut la funció per paràmetre <pre> private function checkLoggedUser (\$msg) { if (\$this->session->userdata('loggedin')) return null; else { \$this->output->set_status_header('403'); \$result = array('status' => '403', 'server' => \$msg,); return \$result; } } </pre>
--	--

CONTROLLER private function emptyThumb ()	
Paràmetres:	Cap
Descripció:	Retorna <i>on-the-fly</i> una imatge per defecte amb la forma de comentari <i>uVumi</i>
Retorna:	<pre> private function emptyThumb() { header('Content-Type: image/png'); \$dest = imagecreatefrompng(\$this->UVI_MEDIA_FOLDER . '/thumb.png'); imagealphablending(\$dest, false); imagesavealpha(\$dest, true); imagepng(\$dest); imagedestroy(\$dest); die(); } </pre>

CONTROLLER private function thumb (filename)	
Paràmetres:	Com a paràmetre es passa a la funció el nom de l'arxiu a fer servir a modificar i a enviar al dispositiu
Descripció:	Aquesta funció agafa la imatge passada com a paràmetre de la seva ubicació en el servidor i després de redimensionar-la <i>on-the-fly</i> i adaptar-la a la forma de comentari <i>uVumi</i> s'envia al dispositiu
Retorna:	Retorna una imatge amb la forma/grandària de comentari <i>uVumi</i> <pre> // Redimensionar \$image_p = imagecreatetruecolor(\$ancho, \$alto); \$image = imagecreatefromjpeg(\$filename); imagecopyresampled(\$image_p, \$image, 0, 0, 0, 0, \$ancho, \$alto, \$ancho_orig, \$alto_orig); </pre>

3.3.4. Estructura de dades de l'aplicació

Taula *uvumi_pois*

Aquesta taula conté les dades de cadascun dels punts que es mostren en el mapa, segons la descripció següent:



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Default	Comment
1	ID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	
2	POI	GEOMETRY			<input type="checkbox"/>	No default	
3	username	VARCHAR	45		<input checked="" type="checkbox"/>	NULL	
4	comment	VARCHAR	166		<input checked="" type="checkbox"/>	NULL	
5	svr-date	TIMESTAMP			<input type="checkbox"/>	CURRENT_TIMESTAMP	
6	type	TINYINT	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0-text 1-photo 2-video 3-sound

Imatge 26 Taula MySQL *uvumi_pois*

Nom del camp	Descripció
id	Codi identificador de cada punt (<i>Point Of Interest</i>) que s'emmagatzema a la <i>BBDD</i> .
poi	Es tracta d'un camp de tipus <i>GEOMETRY</i> que emmagatzema la localització del punt. Aquest tipus s'ha afegit a partir de la versió 5.0 dels servidors <i>MySQL</i> i permet fer cerques emprant funcions geomètriques, punts dins d'una àrea, per distància...
username	Aquest camp conté el nom d'usuari que ha publicat el comentari
comment	Emmagatzema el comentari de text o el nom de l'arxiu que s'hagi publicat, això està determinat pel tipus de comentari que es publiqui
svr-date	Data de publicació del comentari, aquesta data s'estableix automàticament per part del servidor
type	Tipus de missatge que s'ha publicat: <ul style="list-style-type: none">• <i>Tipus 0</i>: comentari de text• <i>Tipus 1</i>: imatge / fotografia• <i>Tipus 2</i>: arxiu de vídeo• <i>Tipus 3</i>: arxiu de so

Taula *uvumi_users*

Aquesta taula emmagatzema la informació necessari per a la identificació i autenticació dels usuaris, segons la taula descriptiva que es mostra a continuació.

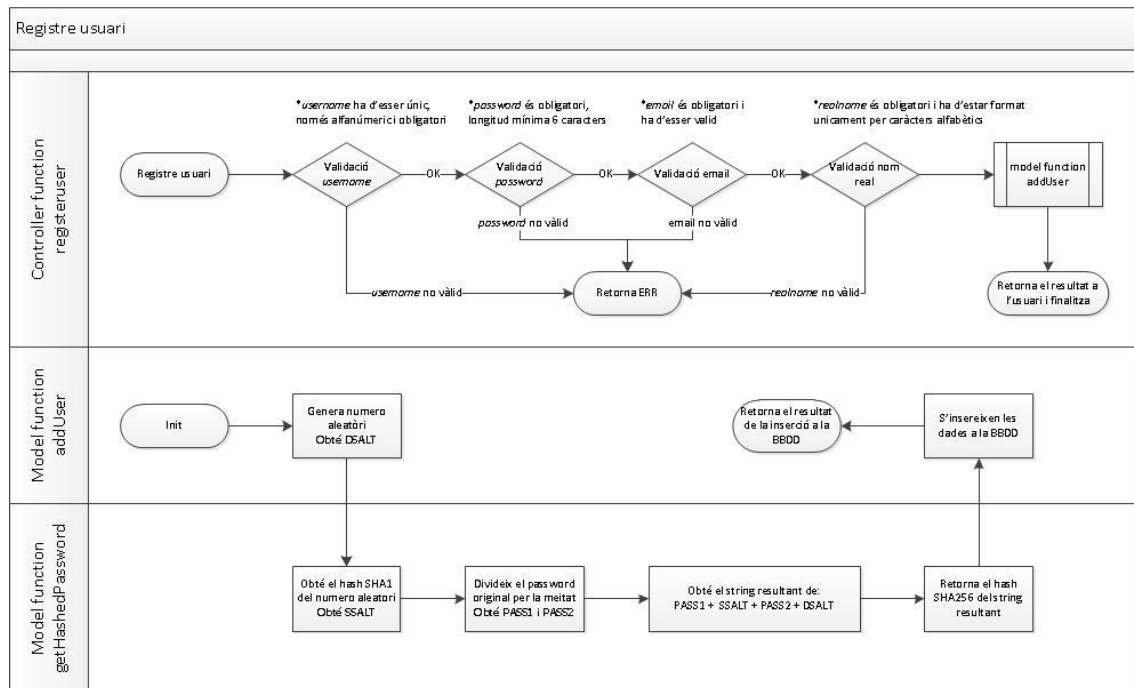
#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Default	C
	ID	INT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	
	2 username	VARCHAR	45		<input type="checkbox"/>	No default	
3	email	VARCHAR	45		<input type="checkbox"/>	No default	
4	password	VARCHAR	45		<input type="checkbox"/>	No default	
5	real_name	VARCHAR	45		<input checked="" type="checkbox"/>	NULL	
6	salt	VARCHAR	45		<input checked="" type="checkbox"/>	NULL	
7	active	TINYINT	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	

Imatge 27 Taula MySQL uvumi_users

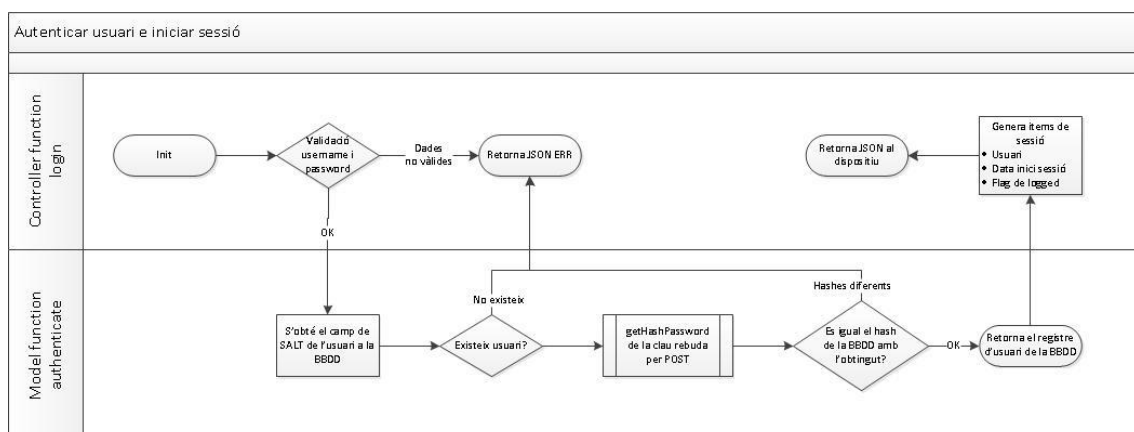
Nom del camp	Descripció
id	Identificador per als usuaris, per fer-ne les operacions típiques d'administració
username	Nom d'usuari emprat per a l'inici de sessió en el dispositiu
email	Adreça de correu electrònic per la validació de l'usuari com a persona i no com a màquina, dins de les millores de l'aplicació caldria afegir una validació d'aquesta adreça a l'hora de fer el registre d'un nou usuari
password	Contrasenya emprada per l'usuari per a la seva autenticació dins de la xarxa
real_name	Nom descriptiu de l'usuari
salt	Codi generat a l'hora de fer el registre per tal d'emmagatzemar de forma segura la contrasenya de l'usuari en format <i>HASH</i>
active	Permet mantenir la informació relativa a un usuari sense eliminar-lo, en versions futures ens permetria la desactivació de l'accés a un usuari particular

3.3.5. Elements de seguretat de l'aplicació

L'element més important de la base de dades, són els elements d'identificació de l'usuari, com són el seu nom d'usuari i la contrasenya. Per evitar que la contrasenya pugui ésser extreta per força bruta de la base de dades, s'ha emmagatzemat fen-ne un *hash* modificat segons es mostra en el següent diagrama.



Imatge 28 Diagrama d'estats de la funció de registre d'usuari



Imatge 29 Diagrama d'estats per l'autenticació e inici de sessió

Atacs XSS

Per tal de protegir l'aplicació d'atacs de tipus *CSS* (*Cross-site scripting*) ja que en el fons es tracta d'una aplicació web, s'han emprat les eines que proveeix el *Framework* emprat en el desenvolupament del *Web Service*, *CodeIgniter*. Aquest tipus de vulnerabilitat permet a un usuari malintencionat, inserir codi *JavaScript* o un altre llenguatge de script similar, permetent segrestar informació de l'usuari (claus, sessions, cookies...) aquesta opció activa la comprovació automàtica de les dades que arriben al *WebService* mitjançant *HTTP_GET* o *HTTP_POST*.

```
/*
|-----
| Global XSS Filtering
|-----
|
| Determines whether the XSS filter is always active when GET, POST or
| COOKIE data is encountered
|
*/
$config['global_xss_filtering'] = TRUE;
```

Imatge 30 Activació protecció XSS CodeIgniter. Arxiu config/config.php

Atacs CSRF

Un altre dels problemes habituals de les aplicacions basades en web, és l'atac *CSRF* (*Cross-site request forgery*) aquest tipus d'atac permet que una aplicació maliciosa envii al servidor una petició oculta mentre l'usuari està fent una altra activitat. Per evitar aquesta tècnica d'atac s'emprarà les eines que proporciona *CodeIgniter* que insereixen un camp a tots els formularis que s'envien i es comproven en el moment de rebre una petició, si aquest no coincideix, es retorna error.

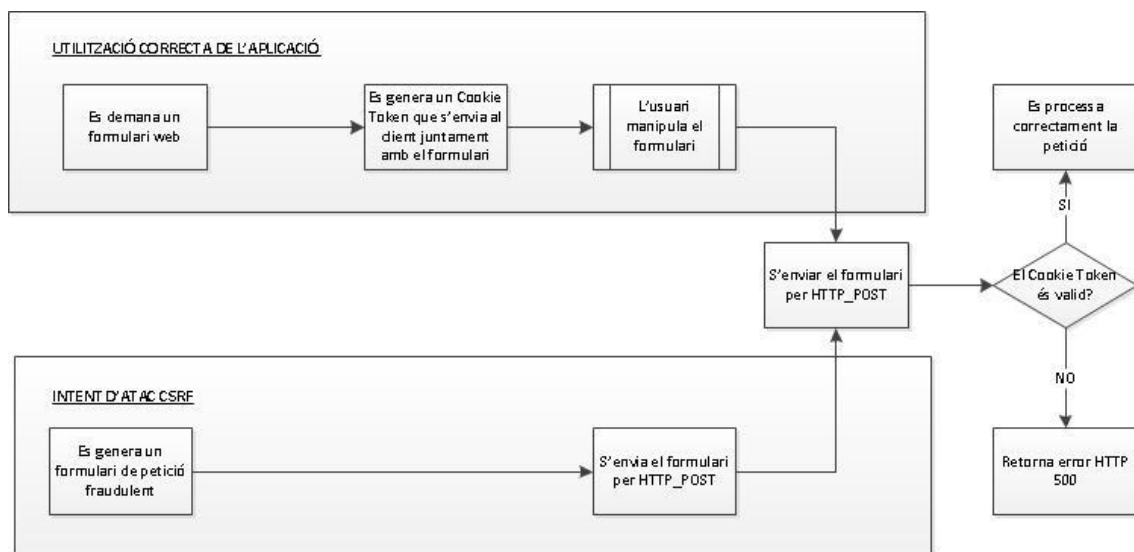
```

/*
|-----
| Cross Site Request Forgery
|-----
| Enables a CSRF cookie token to be set. When set to TRUE, token will be
| checked on a submitted form. If you are accepting user data, it is strongly
| recommended CSRF protection be enabled.
|
| 'csrf_token_name' = The token name
| 'csrf_cookie_name' = The cookie name
| 'csrf_expire' = The number in seconds the token should expire.
*/
$config['csrf_protection'] = TRUE;
$config['csrf_token_name'] = 'csrf_uvumi';
$config['csrf_cookie_name'] = 'csrf_uvumi_cookie';
$config['csrf_expire'] = 7200;

```

Imatge 31 Activació protecció CRSF CodeIgniter. Arxiu config/config.php

En el següent diagrama es mostra quin és el funcionament de la protecció *CSRF* que porta implementada el *framework* emprat.



Imatge 32 Diagrama de l'atac CSRF i mesures emprades per la protecció

Atacs SQL-INJECTION

Aquest atac es caracteritza per intentar inserir a l'aplicació seqüències de codi *SQL* dins del propi codi *SQL* programat, d'aquesta forma s'altera el funcionament normal de l'aplicació i s'aconsegueix executar codi maligne.

Per evitar aquest extrem s'ha emprat en totes les consultes *SQL* les instruccions de *CodeIgniter* que avaluen els caràcters de control (les cometes, els operadors...) i els substitueixen pel seu equivalent en cas que sigui necessari.

```
$this->db->set("comment",$text);  
$this->db->set("username",$username);  
$this->db->set("type",$type);  
$this->db->set("POI", 'geomfromtext("POINT(' . $lon . ' ' . $lat . ')"',  
$this->db->insert("uvumi_pois");
```

Imatge 33 Exemple funció set per evitar SQL-Injection

3.3.6. Aplicació de dispositiu

L'aplicació ha estat desenvolupada emprant el *Framework PhoneGAP* que s'ha descrit dins l'anàlisi tecnològic de l'aplicació, de forma que els llenguatges de programació emprats han estat *HTML5* i *javascript*, per tal de facilitar l'accés als elements de la interfície, s'ha emprat les llibreries de *JQuery*. Alhora també ha estat necessari ampliar les funcionalitats de *PhoneGAP* emprant *plugins* que ha estat necessari desenvolupar-los amb *Java*, e incorporar-los en el projecte.

L'aplicació esta formada basicament per:

- **socialpressActivity.java:** Aquest arxiu conté l'aplicació *Android* desenvolupada amb *java* que mostra el navegador *embedit* amb l'arxiu *map.html*
- **map.html:** Aquest arxiu conté la interfície d'usuari i només conté aquelles funcions que tenen a veure amb la interacció amb l'usuari, aquesta separació s'ha dut a terme per tal de facilitar la traducció en properes versions del producte.
- **map.js:** Aquest arxiu conté totes les funcionalitats que tenen a veure amb la comunicació i representació de dades provinents del *WS*. Per facilitar la configuració de l'aplicació aquest arxiu conté la declaració de constants que tenen a veure amb el funcionament global de la mateixa.
- **MediaPlugin.java:** Aquest arxiu conté el plugin que ha estat necessari desenvolupar per a ampliar les funcions de *PhoneGAP*, en concret s'ha afegit una funció que permet obtenir la durada d'un arxiu de vídeo/àudio. Aquesta

funció ha estat necessària per evitar la publicació d'arxius massa grans al servidor i evitar-ne la saturació del mateix.

```
1  var DEBUG = false;
2  var WS_LOCAL_URL = "http://uvumi.local/app/index.php";
3  var WS_REMOTE_URL = "http://www.siensis.com/app/index.php";
4
5
6  var WS_CONTROLLER_GET_UVIS = "/uvumi/getboundary/";
7  var WS_CONTROLLER_LOGIN = "/uvumi/login/";
8  var WS_CONTROLLER_LOGOUT = "/uvumi/logout/";
9  var WS_CONTROLLER_GET_USERINFO = "/uvumi/getuserinfo/";
10 var WS_CONTROLLER_ADD_UVITEXT = "/uvumi/addtext/";
11 var WS_CONTROLLER_UPLOAD_UVIMEDIA = "/uvumi/addmedia/";
12 var WS_CONTROLLER_THUMBS = "/uvumi/thumbs/";
13 var WS_CONTROLLER_CHECKUSERNAME = "/uvumi/userexists/";
14 var WS_CONTROLLER_REGISTER = "/uvumi/registeruser/";
```

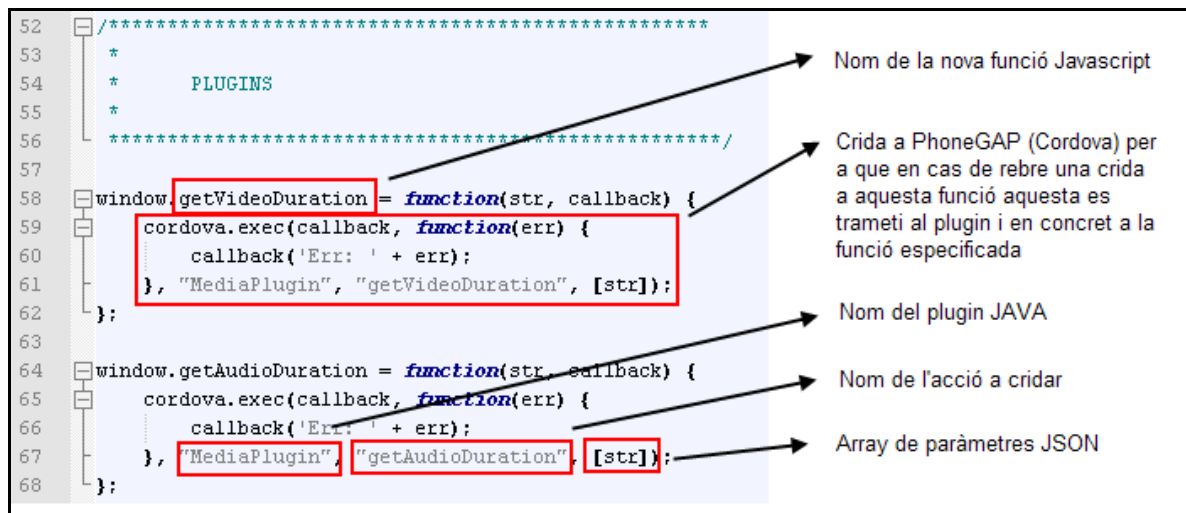
Imatge 34 Constants d'aplicació que localitzen servidor i funcions WS. Arxiu map.js

```
16 var HTTP_NOT_FOUND = 404;
17 var MAX_ITEMS_TO_UPLOAD = 2;
18 var DEFAULT_MAX_ZOOM = 18;
19 var DEFAULT_MIN_ZOOM = 8;
20 var MAX_BUBBLE_CHARACTERS = 166;
21
22 var MAX_DURATION_VIDEO = 15; // Duration for a video upload
23 var MAX_DURATION_AUDIO = 15; // Duration for a video upload
24
25 var UVI_TYPE_TEXT = '0';
26 var UVI_TYPE_PHOTO = '1';
27 var UVI_TYPE_VIDEO = '2';
28 var UVI_TYPE_AUDIO = '3';
```

Imatge 35 Constants d'aplicació, estableix limitacions APP. Arxiu map.js

Per poder emprar les funcions del *plugin* que s'han tingut de desenvolupar, primer és necessari declarar les funcions dins l'aplicació *PhoneGAP* com si es tractes d'una nova crida *Javascript*, tal i com es mostra a continuació.

En aquesta captura es pot veure que es crea una funció en línia que s'anomena *getVideoDuration* i *getAudioDuration* mitjançant una crida a *Cordova* (*PhoneGAP*) es crida al *plugin* anomenat *MediaPlugin* i es passa el nom de l'acció a executar i un array amb tots els paràmetres amb format *JSON*.



Imatge 36 Declaració del plugin i de les seves crides. Arxiu map.js

Dins l'arxiu *MediaPlugin.java* s'ha declarat el *plugin* que conté la funció *execute* que es l'encarregada de despatxar les crides a les accions del *plugin*. La funció rep per paràmetres, el nom de l'acció a desenvolupar, els arguments en format *JSON* i un apuntador al context de retorn.

```

17  public class MediaPlugin extends CordovaPlugin {
18
19      @Override
20      public boolean execute(String action, JSONArray args, CallbackContext callbackContext)
21          throws JSONException {
22          if (action.equals("getVideoDuration") || action.equals("getAudioDuration") ) {
23              String filePath = args.getString(0);
24              MediaPlayer player = new MediaPlayer();
25              try {
26                  player.setDataSource(filePath);
27                  player.prepare();
28                  int iTime = player.getDuration();
29                  callbackContext.success(Integer.toString(iTime/1000));
30              } catch (IOException e) {
31                  callbackContext.error("Error: loading media file: " + filePath);
32              }
33          }
34          return false;
35      }
36  }

```

Imatge 37 Declaració del plugin PhoneGAP. Arxiu MediaPlugin.java

La configuració de l'aplicació mòbil s'emmagatzema emprant les possibilitats de *HTML5* que permet emmagatzemament local.

```

304  /**
305   * This function gets a parameter name from phonegap app storage
306   *
307   * @author Josep M. Flix
308   * @version 1.0 [15/04/2013]
309   * @param name -> Config parameter name
310   * @returns {string} -> Returns parameter value or
311   *                    and empty string if not exists parameter name
312   *
313   */
314  function getConfigValue (name) {
315
316      var value = window.localStorage.getItem(name);
317
318      if (value==null) return "";
319      else return value;
320  }

```

Imatge 38 Emmagatzemament configuració. Arxiu map.js

Per tal de controlar les caducitats de sessió, s'ha desenvolupat una funció que primer fa *login* i en cas que aquest es dugui correctament, llavors crida a la funció demanada, així aconseguim tornar a fer *login* en cas d'una caducitat de sessió.

```

213  /**
214   * This function logins into CONTROLLER platform whith user and password
215   * set in global variables and then it calls functions func_ok o func_err
216   *
217   * @author Josep M. Flix
218   * @version 1.0 [15/04/2013]
219   * @param func_ok -> [optional] Function called if user login
220   * @param func_err -> [optional] Function called if and error occurs when user login
221   *
222   */
223  function login (func_ok,func_err) {
224
225      $.mobile.loading('show');
226
227      console.log ("Login: " + cfgSzUser);
228      console.log ("Password: " + cfgSzPass);
229
230      var urlPostUvi = wsURL + WS_CONTROLLER_LOGIN;
231
232      console.log (urlPostUvi);
233
234      var svrPost = $.ajax({
235          type: "POST",
236          dataType:"json",
237          data: {
238              username: cfgSzUser,
239              password: cfgSzPass,
240          },
241          url: urlPostUvi,
242          cache: false
243      });

```

En cas que es produeixi un error al fer login, es crida aquesta funció

Si s'aconsegueix fer login, s'executa aquesta funció

S'executa la funció corresponent, segons el resultat de fer login

Imatge 39 Funció que permet fer login i executar una funció segons el resultat. Arxiu map.js

Per tal de permetre una traducció de tota la interfície sense cap necessitat de revisar el codi de la pròpia aplicació, les cadenes de caràcters generades pel codi de l'aplicació *PhoneGAP* s'han emmagatzemat dins de l'arxiu d'interfície *map.html*.

```
333 function validatePassword(txtPass,errLabel,btnSave) {  
334  
335     var p1 = $("#"+ txtPass).val();  
336     var p2 = $("#"+ txtPass + "2").val();  
337  
338     if (p1!=p2) {  
339         $("#" + errLabel).html (ERR_PASSWORD_NO_MATCH);  
340         $("#" + errLabel + "2").html (ERR_PASSWORD_NO_MATCH);  
341         $("#" + btnSave).hide();  
342     }
```

Imatge 40 Generació de missatges estàtics dins l'aplicació. Arxiu *map.js*

```
28 <script type="text/javascript" charset="utf-8">  
29     var ERR_PASSWORD_NO_MATCH = "* password not match";  
30     var MSG_CONFIGURATION_SAVED = "Configuration saved";  
31     var ERR_CONFIGURATION_OP = "Error configuration operation";  
32     var ERR_UVUMITEXT_NOT_SET = "[uVumi text required]";  
33     var ERR_VIDEO_TOO_LONG = "Video exceeds " + MAX_DURATION_VIDEO + "s. ";  
34     var ERR_AUDIO_TOO_LONG = "Audio exceeds " + MAX_DURATION_AUDIO + "s. ";  
35     var ERR_UNKNOW_CAPTURING = "An error occurred during capture: ";  
36     var ERR_USER_EXISTS = "* Username already exists";  
37     var ERR_CONFIRMATION_PASS = "confirmation error";  
38     var ERR_EMPTY_REG_FIELDS = "empty fields, all are mandatory";  
39     var ERR_USERNAME_EXISTS = "username already exists";
```

Imatge 41 Missatges localitzats juntament amb la interfície. Arxiu *map.html*

L'aplicació per evitar robatori de sessions i el mal us de la mateixa, tanca la sessió de forma automàtica quan es crida l'opció o quan es surt de l'aplicació mitjançant el botó *back*.

```

359 function exitApp () {
360     var urlRequest = wsURL + WS_CONTROLLER_LOGOUT;
361
362     $.mobile.loading('show');
363
364     var svrPost = $.ajax({
365         async: true,
366         type: "POST",
367         dataType: "json",
368         url: urlRequest,
369         cache: false
370     });
371
372     svrPost.done(function(data, textStatus, jqXHR) {
373         console.log ("SRV: " + data.server + " (" + data.options.username + ")");
374
375         $.mobile.loading('hide');
376         navigator.app.exitApp(); // Exit APP after close server session
377     });
378
379     svrPost.fail(function(jqXHR, textStatus) {
380         console.log ("ERR code: " + jqXHR.status); // get status code 200, 500, 404
381         console.log ("ERR desc: " + jqXHR.statusText); // get error description
382         console.log (jqXHR.responseText);
383
384         $.mobile.loading('hide');
385         navigator.app.exitApp(); // Exit APP without close server session
386     });
387 }

```

Imatge 42 Funció de tancament de sessió, enviament crida WS de logout. Arxiu map.js

L'aplicació captura la pulsació del botó *back* del dispositiu per poder fer la crida a la funció del *logout* del *webService* o tornar un pas enrere en cas necessari.

```

68 function onBackButton (e) {
69     console.log("back button press");
70     if ($.mobile.activePage.is('#page')){
71         e.preventDefault();
72
73         exitApp();
74     }
75     else {
76         navigator.app.backHistory()
77     }
78 }

```

Imatge 43 Funció per capturar la pulsació del botó back del dispositiu. Arxiu map.html

La llibreria emprada per la visualització del mapa permetia la utilització de diferents fonts de mapes, google, openstreetmap, entre altres, però es va escollir el darrer tal i com s'ha explicat en els apartats d'anàlisi per qüestions de llicenciamnt i actualització de dades.

```

533 function initLeaf () {
534
535     map = L.map('map');
536
537     markers = new L.MarkerClusterGroup({ maxClusterRadius: 80,
538                                         spiderfyOnMaxZoom: true,
539                                         spiderfyDistanceMultiplier: 2,
540
541     overlayMaps = {
542         "Uvumis": markers
543     });
544
545     L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
546         maxZoom: DEFAULT_MAX_ZOOM,
547         minZoom: DEFAULT_MIN_ZOOM,
548         attribution: '&copy; uVumi'
549     }).addTo(map)

```

Imatge 44 Inicialització del mapa de la interfície. Arxiu map.js

La variable *map* que conté el mapa que es representarà dins la interfície gràfica i que és simplement un contenidor de tipus *html-div* tal i com es mostra a la Imatge 45 a banda per la inicialització del mapa, és necessari definir una capa on s'emmagatzemaran els punts provinents del WS, aquesta capa està dotada d'un agrupament de punts i per tant cal inicialitzar aquest agrupament anomenat *markers*. Per últim cal mostrar el mapa que s'ha de carregar d'alguna de les fonts de mapes que he comentat anteriorment, d'això se n'encarrega la funció *L.tileLayer* que inicialitza la llibreria per a descarregar en cas necessari les capes del mapa, segons la ubicació i el zoom que hi hagi sobre el mapa.

```

185 </div><!-- /header -->
186 <div data-role="content" style="padding:0px;">
187     <div id="map" ></div>
188 </div>

```

Imatge 45 Element HTML contenedor del mapa de l'aplicació. Arxiu map.html

Per tal de recuperar el mínim d'informació necessària, o sigui només els punts que estiguin a la vista del mapa del dispositiu, s'ha requerit crear una funció anomenada *boundaryRequest* que sol·licita al WS que envii els punts que hi hagi dins del mapa, tal i com es veu a Imatge 46.

```

646 function boundaryRequest () {
647     var nwLimit = map.getBounds().getNorthWest();
648     var selimit = map.getBounds().getSouthEast();

```

Imatge 46 Funció boundaryRequest per obtenir els punts personalitzats. Arxiu map.js

```

91 $SQL = "SELECT X('POI') as lon, Y('POI') as lat, 'username', 'comment', 'type', 'svr-date' as dd FROM 'uvumi_pois' "
92 " WHERE MBRContains(GeomFromText('Polygon(($x1 $y1, $x2 $y1, $x2 $y2, $x1 $y2, $x1 $y1))'),POI);";

```

Imatge 47 Funció SQL per obtenir els punts GEOMETRY ubicats en un polígon. Server model

Per tal de calcular els punts continguts dins del polígon, en el servidor s'ha emprat les noves funcions espacials de *MySQL* que permeten fer cerques sobre punts.

Cada cop que es reben nous punts pel mapa, cal ubicar-los, assignar-los una icona i generar un diàleg *popup* que mostri el contingut del punt, tal i com es mostra a continuació.

```

811 function addJSONFeaturesToMap (geoJsonData) {
812     geoJsonLayer = L.geoJson(geoJsonData, {
813         onEachFeature: function (feature, layer) {
814             var ico_file;
815             switch (feature.properties.type) {
816                 case UVI_TYPE_TEXT:
817                     ico_file = 'img/uvumi-poi.png';
818                     break;
819                 case UVI_TYPE_PHOTO:
820                     ico_file = wsURL+ WS_CONTROLLER_THUMBS + feature.properties.thumb;
821                     break;
822                 case UVI_TYPE_VIDEO:
823                     ico_file = 'img/uvumi-poi-video.png';
824                     break;
825                 case UVI_TYPE_AUDIO:
826                     ico_file = 'img/uvumi-poi-audio.png';
827                     break;
828                 default:
829                     ico_file = 'img/uvumi-poi.png';
830                     break;
831             }
832             var uvumi_ico = L.icon({
833                 iconUrl: ico_file,
834                 iconSize: [52, 64],
835                 iconAnchor: [18, 63],
836             });
837             layer.setIcon(uvumi_ico);
838             layer.bindPopup(feature.properties.date + ' - ' + feature.properties.time + '<br/>' +
839                 feature.properties.comment + '<br/><b>(' +
840                 feature.properties.username + ')</b>', {maxWidth: 200});
841         }
842     });
843     markers.clearLayers();
844     markers.addLayer(geoJsonLayer, true);
845     map.addLayer(markers);
846 }

```

Imatge 48 Assignació de punts al mapa, després de la recepció. Arxiu map.js

Dins “Annex A. Funcions map.js” es pot trobar el codi de totes les funcions de l’aplicació que permeten la comunicació amb el WS.

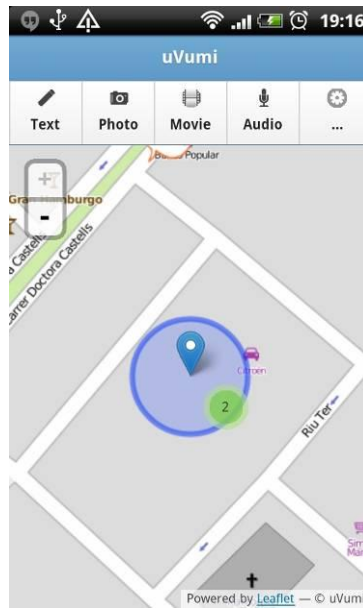
3.4. Resultats obtinguts. Mini manual uVumi App.



Imatge 49 "Splash screen"

La Imatge 49 "Splash screen" apareix en el moment d’iniciar l’aplicació, durant el temps que apareix aquest element, l’aplicació està fent les tasques corresponents a la pre-carrega de l’aplicació:

- Obtenció de les dades d’usuari (usuari/contrasenya), en cas que aquestes no estiguin disponibles, l’aplicació mostra un formulari que permet que l’usuari es pugui registrar
- Localització del dispositiu, mitjançant *GPS*, *AGPS* o la senyal de telefonia
- Inicialització del mapa, obtenint els *layers* segons la posició on està ubicat l’usuari
- Obtenció dels punts adjacents a la ubicació de l’usuari i representació gràfica dels mateixos



Imatge 50 "uVumi location screen"

La pantalla Imatge 50 "uVumi location screen" apareix en el moment que l'aplicació ja es troba disponible per l'usuari, després de tot el procés de carrega. Aquesta pantalla es divideix en dos zones: una barra d'eines ubicada en la part superior de la pantalla i la resta es correspon a una vista mapa, centrada a la ubicació de l'usuari si aquesta ha estat possible precisar-la. El zoom només es possible gestionar-lo mitjançant els botons que apareixen superposats en el propi mapa, per una qüestió d'incompatibilitats amb la *API* de representació del plànol i el propi *SSOO Android*.

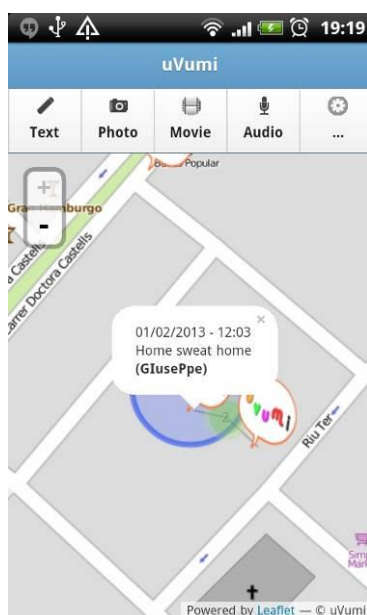
En el plànol es representen tots els punts que estan emmagatzemats en el servidor, per facilitar-ne la representació si n'hi hagués un volum molt elevat, s'agrupen mitjançant un cercle de color verd on s'indica el nombre de punts que s'estan agrupant. Tal i com es pot veure en la Imatge 50 que s'agrupen dos *uVumis*.

L'aplicació detecta la precisió amb la qual s'ha obtingut la ubicació de l'usuari i aquesta es representa mitjançant un cercle de color blau, on el centre, es la ubicació obtinguda per l'APP i el cercle té el radi de la precisió retornada pel propi dispositiu.



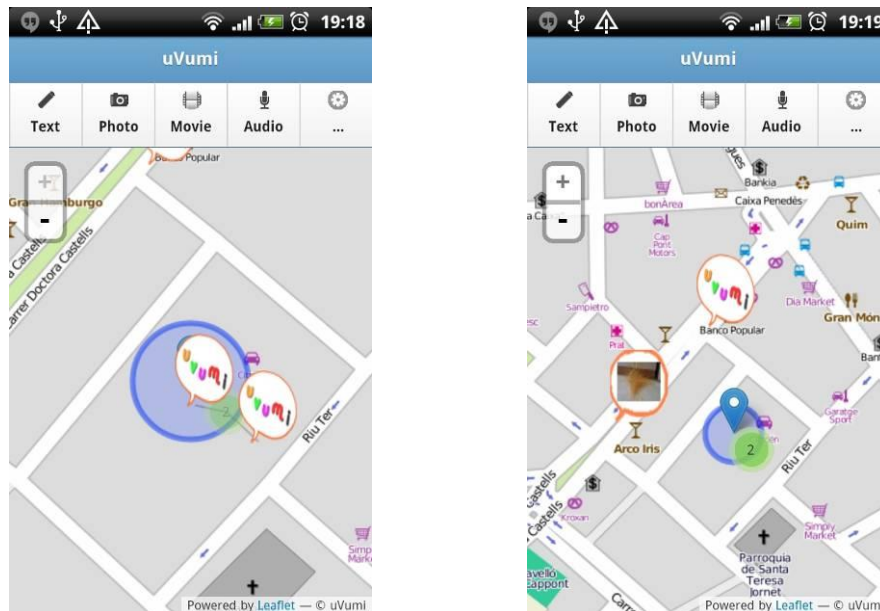
Imatge 51 "Add uVumi text screen"

Quan un usuari vol inserir un nou *uVumi* en el servidor, pot afinar la ubicació desplaçant el punt de localització de la pantalla i posteriorment inserir el text mitjançant la pantalla de la Imatge 51 "Add uVumi text screen", aquest diàleg ens permet publicar un text associat a la ubicació de l'usuari, amb el límit màxim de 166 caràcters, en cas de superar aquesta limitació, l'aplicació ens mostra el text en vermell i desactiva la possibilitat d'enviar-lo al servidor. Mitjançant el botó "*Add it!*" l'usuari envia el text i la seva ubicació al servidor per la publicació i difusió del mateix.



Imatge 52 "uVumi text display"

En la pantalla Imatge 52 “uVumi text display” l’usuari pot veure el text que es troba darrera d’un punt, mitjançant un globus, en ell es mostra el missatge publicat per l’usuari, la data/hora de la seva publicació, així com l’alies d’aquell que l’ha publicat.



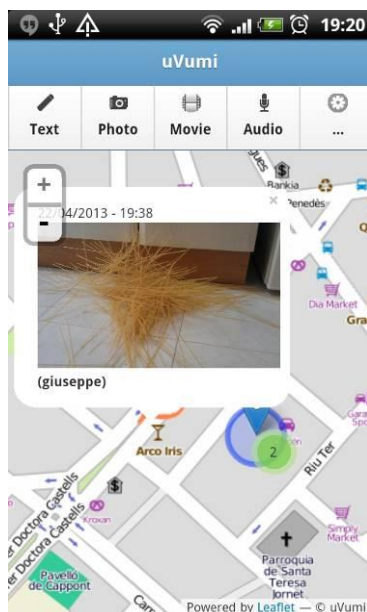
Imatge 53 "Zoom view"

L’aplicació mostra segons el zoom i el tipus de punt, un globus diferent, en Imatge 53 "Zoom view" podem observar a més que si el zoom obliga a mostrar molts punts, per evitar que aquests es superposin, es mostra agrupats, així com en els punts fotogràfics, l’aplicació mostra una miniatura de la fotografia publicada.



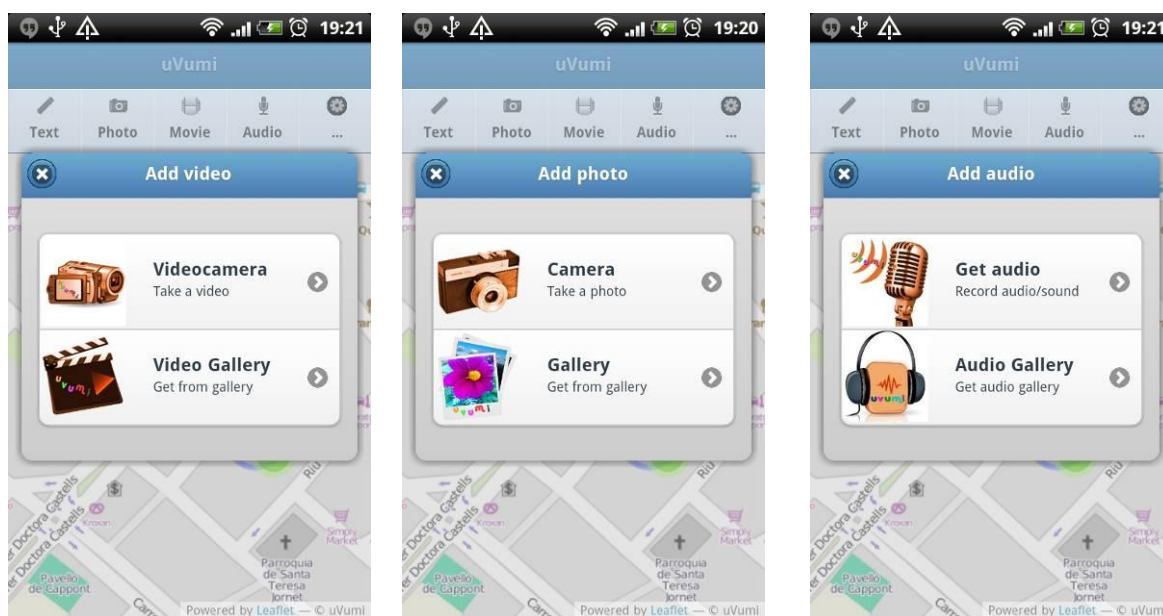
Imatge 54 Icones uVumi segons contingut

La Imatge 54 mostra les icones que es mostren segons el contingut: fotografia, vídeo, àudio i text. En la versió definitiva la icona corresponent a “uVumi Foto” ha estat substituïda per una miniatura de la fotografia publicada, això permet una visualització més ràpida del contingut.



Imatge 55 "uVumi photo popup"

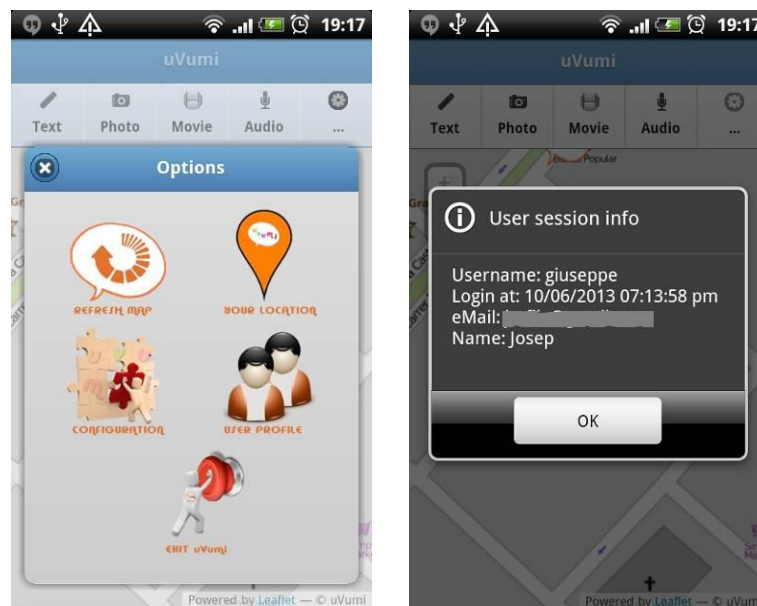
En el moment que l'usuari fa *click* en la miniatura d'un *uVumi*, aquest es mostra mitjançant un globus *popup*, tal i com es mostra en Imatge 55 "uVumi photo popup" en aquest cas a més, com que es tracta d'un arxiu fotogràfic, si l'usuari fa clic sobre la fotografia, aquesta es mostra a pantalla completa.



Imatge 56 "Add uVumi media dialog"

Quan l'usuari vol inserir un comentari multimèdia, en el plànol, segons la seva tipologia apareix un diàleg que li permet escollir la procedència de l'element multimèdia tal i com es pot veure a Imatge 56 "Add uVumi media dialog". L'element multimèdia podem

capturar-lo a través del propi dispositiu o anar-lo a buscar a la galeria multimèdia del dispositiu.



Imatge 57 "Other options screen"

L'usuari pot en tot moment accedir a altres opcions de menú tal i com es mostra en la Imatge 57 "Other options screen", dins aquest apartat, l'usuari pot fer les tasques següents:

- **Refresh map:** Actualitzar el mapa i descarregar de nou els layers del servidor així com els punts que en ell es troben.
- **Show location:** Demanar a l'aplicació que localitzi al dispositiu sobre del plànol, tal i com ja es fa de forma automàtica a l'iniciar l'aplicació.
- **Configuration:** Configurar les dades d'usuari, el nom d'usuari, la contrasenya, l'adreça de correu electrònic i el nom descriptiu del mateix.
- **User profile:** Mostra per pantalla les dades de la connexió actual, tal i com es veu en l'exemple Imatge 57
- **Exit uvumi:** Sortir de l'aplicació, encara que aquesta opció també es pot realitzar mitjançant el botó *back* de la majoria de dispositius *Android*.

CAPÍTOL 4

1. CONCLUSIONS

2. LÍNIES FUTURES

4. Capítol 4. Conclusions i línies futures

4.1. *Conclusions*

El primer a remarcar és que s'ha assolit gran part dels objectius que s'havien establert als inicis d'aquest projecte. Hem pogut obtenir una aplicació funcional que mostra informació que es publica en el *cloud*.

Dins dels objectius i eixos principals definits del projecte teníem la creació d'una aplicació multi plataforma per a mòbil, objectiu àmpliament assolit amb la utilització de la plataforma *PhoneGAP* projecte inicialment lliure que actualment es troba en mans de la companyia *Adobe* i amb la utilització d'altres estàndards com *HTML5/CSS*.

Dins del segon gran objectiu del projecte teníem la creació d'una *API* que permetés altres desenvolupaments que aprofitessin les funcionalitats creades en el *cloud* per a l'aplicació mòbil, aquest objectiu ha estat assolit amb una única mancança que caldria resoldre en les línies futures, que seria l'adaptació de la *API* a la comunicació via *REST* i l'adaptació de l'aplicació mòbil, aquesta mancança s'ha produït per un problema en el *framework PhoneGAP* que en el moment del desenvolupament encara no estava resolt tal i com hem comentat en l'apartat d'anàlisi i desenvolupament.

Amb tot s'ha pogut avaluar el funcionament de l'aplicació mitjançant diversos dispositius *android* i s'han publicat punts en diversos llocs, Lleida, Girona, Tarragona, Barcelona i aquests es mostren correctament, tal i com es pretenia al definir els eixos d'aquest projecte: creació d'una *API* que permetés a l'usuari publicar i obtenir punts d'interès segons la seva ubicació.

En definitiva el projecte aconsegueix crear una aplicació per entorn *Android*, que permet als usuaris interactuar entre ells i que podríem emmarcar dins les aplicacions socials amb *GEO* localització, com pot ésser *foursquare*.

4.2. *Linies futures*

Pel que fa al futur d'aquest projecte, serà necessària adaptar-la a les noves necessitats i les noves tendències que es vagin marcant dins de la telefonia mòbil i els *smartphones* en general. De la mateixa forma caldrà adequar-la als nous contextos de comunicació que estan sorgint arrel de les xarxes socials.

Algunes de les idees que es podrien desenvolupar en la millora d'aquest projecte podrien ser:

- Generació d'un entorn web que permeti als usuaris accedir al seu contingut via navegador, emprant qualsevol altre dispositiu, per exemple un PC
- Adaptació a d'altres tipus de dispositius, *iphone*, tablets, etc...
- Eliminar la necessitat de la inscripció o registre, mitjançant una inscripció automàtica a través del número de telèfon associat al dispositiu, validant-lo amb SMS o similar, tal i com ja es fa en altres aplicacions.
- Permetre que l'usuari de l'aplicació es pugui subscriure al contingut d'un determinat usuari o d'un determinat contingut.
- Permetre que l'usuari pugui rebre alertes sobre contingut interessant quan aquest hi estigues prop, o fins i tot segons una distància que es pugui configurar.
- Permetre a l'usuari filtrar contingut segons el seu interès, això minimitzaria el nombre d'elements a mostrar en el mapa.
- Adaptar la *API* a les característiques de *WebServices REST* i adaptar l'aplicació mòbil per a emprar aquests *WS*.

BIBLIOGRAFIA

Bibliografia

Referències bibliogràfiques

[HTML5GEO] “HTML5 Geolocation. Bringing Location to Web Applications”. Anthony T. Holdener III. Ed. O’Reilly. 2011 ISBN: 978-1-449-30472-0

[JQUERYMOBUP] “jQuery Mobile. Up and Running. Using HTML5 to Design Web Apps for Tablets and Smartphones”. Maximiliano Firtman.. Ed O’Reilly. 2012. ISBN: 978-1-449-39765-4

[JQUERYDESIGNERS] “jQuery for Designers. An approachable introduction to web design in jQuery for non-programmers. Beginner’s Guide”. Natalie MacLess. Ed. PACKT. 2010. ISBN: 978-1-84951-670-9

[OSMAPCARTOGRAPHER] “OpenStreetMap. Be your own Cartographer”. Jonathan Bennet. Ed. PACKT. 2010. ISBN: 978-1-847197-50-4

[OPENLAYER210] “OpenLayers 2.10 Create, optimize, and deploy stunning cross-browser web maps with the OpenLayers JavaScript web-mapping library. Beginner’s Guide”. Erik Hazzard. Ed. PACKT Publishing. 2011. ISBN: 978-1-849514-12-5

[PGAPESSENTIALS] “PhoneGap Essentials”. Wargo, John M. Ed. Addison Wesley. 2012. ISBN: 978-0-321-81429-6

[PGAPBEGINNING] “Beginning PhoneGap. Mobile Web Framework for Javascript and HTML5”. Rohit Ghatol, Yogesh Patel. Ed. Apress. 2012.

[PGAPBEGINNING2] “Beginning PhoneGap”. Thomas Myer. Ed. Wiley. 2012. ISBN: 978-1-118-15665-0

[PHPMVC] “Pro PHP MVC. Everithing you need to know about using mvc with PHP in a single reference”. Chris Pitt. Ed. Apress. 2012. ISBN: 978-1-4302-4164-5

Referències web

[AWS] Amazon Web Services. <http://aws.amazon.com/es/> (Consulta: octubre 2013)

[FOURSQUAREEMPRESAS] “Foursquare para Empresas. Marketing por Geoposicionamiento.” Bernat Comas

<http://www.blogtrw.com/2010/05/foursquare-para-empresas-marketing-geoposicionamiento/>

(Consulta: maig 2013)

[GEOLFOURSQUARE] “Geolocalizacion, movilidad y recomendación. A proposito de foursquare”. 13 abril de 2010. CoolInsights

<http://www.iabspain.net/wp-content/uploads/downloads/2012/05/Geolocalizacion-movilidad-y-recomendacion-A-proposito-de-foursquare.pdf>

(Consulta: maig 2013)

[GEOMETRYMYSQL]

<http://dev.mysql.com/doc/refman/5.0/es/gis-class-geometry.html>

(Consulta: maig 2013)

[MARKFOURSQUARE] “El potencial de Foursquare como herramienta de recomendación y conversión”. Web Puromarketing, article marketing digital.

<http://www.puromarketing.com/10/13680/potencial-foursquare-como-herramienta-recomendacion-conversion.html>

(Consulta: maig 2013)

[OPENSTREETMAP] <http://www.openstreetmap.org>

[**ORANGE2012**] “eEspaña. Informe anual 2012 sobre el desarrollo de la sociedad de la información en España”. Fundación Orange.

http://fundacionorange.es/fundacionorange/analisis/eespana/e_espana12.html

(Consulta: maig 2013)

[**OPENPROJ**] OpenProj projecte Open Source de software de planificació de projectes per entorns d'escriptori, desenvolupament de diagrames Gantt i diagrames PERT.

<http://sourceforge.net/projects/openproj/> (Consulta: Juny 2013)

[**PENCILPROJECT**] Pencil project és un software open-source per la creació de prototipus, diagrames... <http://pencil.evolus.vn>

[**PHPFRAMBENCH1**]

<http://blog.websitesframeworks.com/2013/03/web-frameworks-benchmarks-192/>

(Consulta: maig 2013)

[**PHPFRAMBENCH2**]

<http://pr0digy.com/codeigniter/benchmark-update-static-cake-codeigniter-kohana/>

(Consulta: maig 2013)

[**REST**] http://en.wikipedia.org/wiki/Representational_state_transfer

(Consulta: maig 2013)

[**SOCIEDADRED**] “La sociedad en red. Informe anual 2011. Edició 2012”. Secretaria de estado de telecomunicaciones y para la sociedad de la información. Ministerio de industria, energia y turismo.

<http://www.ontsi.red.es/ontsi/ca/estudios-informes/informe-anual-la-sociedad-en-red-2011-edici%C3%B3n-2012>

(Consulta: maig 2013)

[**TELEFONICA2012**] “siE[12” La Sociedad de la información en España 2012.

http://www.fundacion.telefonica.com/es/que_hacemos/noticias/detalle/10_01_2013_esp_2430

(Consulta: maig 2013)

ANNEXES

1. ANNEX 1. FUNCIONS MAP.JS

Annexes

Annex A. Funcions map.js

```
/**
 * Show current user info. Function test to server controller
 * info about current username and shows as alert.notification
 * message.
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @returns {none}
 */
function showUserInfo()

/**
 * This function publish a comment into uvumi. If user is not logged, then
 * functions calls login function to login and recall this function. The parameter
 * permits stop infinite propagation
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param stopPropagation
 * - This parameters only permits a unique call from login function
 *
 */
function addComment (stopPropagation)

/**
 * This function logins into CONTROLLER platform whith user and password
 * set in global variables and then it calls functions func_ok o func_err
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param func_ok -> [optional] Function called if user login
 * @param func_err -> [optional] Function called if and error occurs when user login
 *
 */
function login (func_ok,func_err)
```

```

/**
 * This function loads config from phonegap app storage
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @returns {Boolean}
 *      -> Returns true if parameters are loaded correctly, false otherwise
 */
function loadConfig ()

/**
 * This function gets a parameter name from phonegap app storage
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param name -> Config parameter name
 * @returns {string} -> Returns parameter value or
 *      and empty string if not exists parameter name
 */
function getCfgValue (name)

/**
 * This function validates password introduction, that txtPass and txtPass2 are equals
 * if not, error message is show in errLabel and btnSave are disabled
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param txtPass
 *      -> ID for input password field. Used to get value and set style (password-field)
 * @param errLabel
 *      -> ID for label to show error message and set style (password-match-error-label)
 * @param btnSave
 *      -> ID for button to save values into config phonegap app (button-save-config)
 */
function validatePassword(txtPass,errLabel,btnSave)

/**
 * This functions logouts closing session from CONTROLLER platform
 * and exits APP closing it
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @returns {none}
 */
function exitApp ()

```



```

/**
 * This function save username and password into phonegap config storage.
 * In debug mode it save servermode, LOCAL or REMOTE
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param userfield -> ID for input text that contains username
 * @param passfield -> ID for input password that contains user password
 * @param serverfield -> ID for select that contains server mode, LOCAL or REMOTE
 * @param formname -> ID for form element into config page
 * @param homebtnname -> ID for button that links to MAP PAGE into app
 *
 */
function saveConfig(userfield,passfield,serverfield,formname,homebtnname)

/**
 * This function clear all configuration parameters from phonegap storage
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param userfield -> ID for input text that contains username
 * @param passfield -> ID for input password that contains user password
 * @param serverfield -> ID for select that contains server mode, LOCAL or REMOTE
 * @param formname -> ID for form element into config page
 * @param homebtnname -> ID for button that links to MAP PAGE into app
 *
 */
function clearConfig(userfield,passfield,serverfield,formname,homebtnname)

/**
 * This function loads Configuration PAGE UI
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 *
 */
function loadConfigPage()

```

```

/**
 * This function write characters left to MAXIMUM into comment textbox and
 * sets the style for save/send button and shows a char counter
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param obj -> object that gets input characters
 * @param nCharacters -> MAX characters permitted into obj input textbox
 * @param btnName -> ID for button to save/send message into obj
 * @param countName -> ID for label that counts characters left
 *
 */
function textLimit(obj,nCharacters,btnName,countName)

/**
 * This function initialize leaflet map UI, with: user location, requests data for
 * boundary view and sets: ZOOM level, spiderfy markers, drag user location function
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 *
 */
function initLeaf ()

/**
 * This function adds a control into map UI to show all data if uvumi data are
 * filtered for a username
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 *
 */
function addControlToMap ()

/**
 * This function reloads map, getting data for current map boundary
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 *
 */
function refreshMapUI()

```

```

/**
 * This function request uvumi-data for controller after zoom it,
 * to show into new boundary map
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param e -> event
 */
function onZoomEnd(e)

/**
 * This function request uvumi-data for controller after drag map,
 * to show into new boundary map
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param e -> event
 */
function onDragEnd(e)

/**
 * This function gets boundary map (NW and SE) and request data to CONTROLLER
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 */
function boundaryRequest ()

/**
 * Function launched when user location is found
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param e -> geolocation event
 * @see
 * http://docs.phonegap.com/en/2.4.0/cordova_geolocation_geolocation.md.html#geolocationSuccess
 */
function onLocationFound(e)

```

```

/**
 * Function launched when user location causes and error.
 * Functions shows error message.
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param e -> error returned by device
 * @see
 * http://docs.phonegap.com/en/2.4.0/cordova\_geolocation\_geolocation.md.html#geolocationError
 */
function onLocationError(e)

/**
 * This function adds a marker into user current location, make it draggable
 * and draws a circle around user location with geolocation precision radius
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param radius -> geolocation precision radius
 */
function setLocationMarker (radius)

/**
 * Function launched when user location ends a drag operation, to
 * save the new user position
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param e -> Dragend event
 */
function onUserLoccationDragEnd (e)

/**
 * This function gets device location and shows into MAP UI, setting zoom and position
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 */
function refreshLocation ()

```

```

/**
 * This function saves current latitude/longitude into form object,
 * before send to CONTROLLER
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param frm
 *      -> Form object that contains hidden input to save current latitude and longitude
 *
 */
function getLatLon(frm)

/**
 * This functions request data to CONTROLLER with boundray limits
 * and try to connect if function gets an UNAUTHORIZED error message.
 * to evite infinite propagation, function uses onlyone parameter
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param limits
 *      -> string that contains boundary coords format "NW-lat/NW-lon/SE-lat/SE-lon"
 * @param onlyone
 *      -> this parameter stops login propagation
 *
 */
function requestData(limits,onlyone)

/**
 * This function adds GEO-JSON Features to the map. Binding a popup
 * with elements and changing
 * the icon according feature properties type.
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param geoJsonData -> JSON array returned by CONTROLLER
 *
 */
function addJSONFeaturesToMap (geoJsonData)

```

```

/**
 * This function launches device capture, the parameter determines wich type
 * of capture device will do it. ALLMEDIA type is used to audio capture,
 * otherwise it determines error.
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param iType -> Phonegap enum type
 *         (Camera.MediaType.PICTURE / Camera.MediaType.VIDEO / Camera.MediaType.ALLMEDIA)
 */
function launchCapture (iType)

/**
 * This function is launched when device has captured a video or an audio
 * file. The function checks if audio or file exceeds maximum duration for audio
 * or video file and it uploads to the platform
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param mediaFiles -> Array of files captured, even video or audio
 * @param iType -> Phonegap enum type
 *         (Camera.MediaType.VIDEO | Camera.MediaType.ALLMEDIA)
 */
function onCaptureSuccess(mediaFiles, iType)

/**
 * This function is lauched when an error occurs in the video o audio capture process
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param error -> Error produced in the capture process
 * @param iType -> Phonegap enum type
 *         (Camera.MediaType.VIDEO | Camera.MediaType.ALLMEDIA)
 */
function onCaptureError(error,iType)

/**
 * This function is lauched when an error occurs when user captures a photo
 * with the device
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param error -> Error returned by phonegap into device
 */
function onCapturePhotoError(error)

```

```

/**
 * This function launches a gallery selection to the user interface.
 * MediaType.ALLMEDIA is user for audio gallery capture. For video an audio,
 * the function checks if exceeds maximum duration, after this check it uploads to
 * the platform, photos uploads directly without any check
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param iType -> Phonegap enum type
 *         (Camera.MediaType.PICTURE | Camera.MediaType.VIDEO | Camera.MediaType.ALLMEDIA)
 */
function launchCaptureGallery (iType)

/**
 * This function is launched when user takes a photo successfully, to upload
 * to the platform
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param mediaFiles -> Array of phonegap photo captures
 */
function onCapturePhotoSuccess(mediaFiles)

/**
 * This function uploads content (audio, video, photo) to the platform over
 * controller and recorder server response, if error is "not authorized/not logged"
 * this function tries to login and resend the file, to not propagate infinitely
 * this propagation on use stopPropagate parameter
 *
 * @author Josep M. Flix
 * @version 1.0 [15/04/2013]
 * @param mediaFile -> File to upload into platform
 * @param WS_TYPE -> {string} ['audio','video']
 * @param stopPropagate
 *         -> This parameter stops propagation, after login to platform if is needed
 */
function uploadFile(mediaFile,WS_TYPE, stopPropagate)

```