

# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**E.T.I.G**

**WeDugu**

**Memòria**

**PEDRO JOSÉ MARTÍNEZ CARRIÓN**  
**PONENT: JOSEP MARIA GABRIEL SOLANILLA**

PRIMAVERA 2011



**TecnoCampus**  
**Mataró-Maresme**



## **Agraïments**

A Núria Silva Serrano i Josep Maria Gabriel Solanilla



## **Resum**

L'objectiu del present projecte es la construcció d'una xarxa social capaç de proposar plans d'oci als usuaris en funció del seus gustos i de la seva localització en el moment d'ús de l'aplicació. La xarxa social disposa de dues aplicacions clients, una aplicació web disponible des de qualsevol navegador i una aplicació per a telèfons mòbils Android. Com a nexa d'unió de les dues aplicacions client s'ha establert una base de dades comú a la qual accediran les dues aplicacions. L'aplicació web s'ha implementat seguint el patró Model-Vista-Controlador i l'aplicació Android s'ha implementat seguint les guies de la documentació oficial d'Android. El resultat son dues aplicacions totalment funcionals.

## **Resumen**

El objetivo del presente proyecto es la construcción de una red social capaz de proponer planes de ocio a sus usuarios en función de sus gustos y de su localización en el momento de uso de la aplicación. La red social dispone de dos aplicaciones cliente, una aplicación web disponible desde cualquier navegador y una aplicación para teléfonos móviles Android. Como nexo de unión de las dos aplicaciones cliente se ha establecido una base de datos común a la que acceden las dos aplicaciones. La aplicación web ha sido implementada siguiendo el patrón Modelo-Vista-Controlador y la aplicación Android ha sido implementada siguiendo las guías de la documentación oficial de Android. El resultado son dos aplicaciones totalmente funcionales.

## **Abstract**

The objective of this project is to build a social network capable of offering its users entertainment plans based on their preferences and location at the time of application's use. The social network has two client applications, web application available from any browser and mobile application for Android smartphones. As the nexus of the two client applications has established a common database which will access two applications. The web application has been implemented following the Model-View-Controller and the Android application has been implemented following the guidelines and official documentation of Android. The result is two fully functional applications.



# Índex.

Índex de figures.....	V
Índex de taules.....	VII
Glossari de termes.....	IX
1. Objectius.....	1
1.1. Propòsit.....	1
1.2. Objecte.....	1
2. Introducció.....	3
2.1. Què és WeDugu?.....	3
2.2. A qui va dirigit.....	3
2.3. Disseny funcional.....	3
2.3.1. Punt de partida.....	3
2.3.2. Utilització del producte.....	3
2.3.3. Serveis.....	4
2.4. Planificació.....	5
3. Anàlisi Tècnic.....	7
3.1. Arquitectura del projecte.....	7
3.2. Casos d'ús.....	8
3.2.1. Casos d'ús Aplicació Android.....	8
3.2.2. Casos d'ús Aplicació Web.....	13
3.3. Descripció de la base dades.....	27
3.3.1. Model de Dades.....	27
4. Tecnologies.....	37
4.1. Android vs iOS.....	37
4.2. Serveis Web.....	39
4.2.1. Web Services Description Language (WDSL).....	40
4.2.2. Simple Object Acces Protocol(SOAP).....	41
4.3. Llibreria ksoap-2android.....	42
4.4. Elecció.....	42
5. Android.....	45
5.1. Que és Android.....	45

5.2. Historia. ....	45
5.3. Característiques .....	46
5.4. Arquitectura .....	48
5.4.1. Representació de l'arquitectura.....	48
5.4.2. Aplicacions. ....	48
5.4.3. Marc de treball d'aplicacions.....	49
5.4.4. Llibreries. ....	49
5.4.5. Android Runtime. ....	50
5.4.6. Nucli de Linux. ....	50
5.5. Estructura d'una aplicació. ....	51
5.5.1. Activity. ....	51
5.5.2. Intent. ....	51
5.5.3. Listener. ....	51
5.5.4. Serveis.....	52
5.5.5. Content Provider. ....	52
5.5.6. Android Manifest. ....	52
5.6. Tipus d'aplicacions.....	52
5.6.1. Tasques. ....	52
5.6.2. Processos.....	53
5.6.2. Threads.....	53
5.7. Cicle de vida d'una aplicació.....	53
5.8. Estructura d'un projecte Android. ....	55
5.8.1. Carpeta /src. ....	56
5.8.2. Carpeta /res. ....	56
5.8.3. Carpeta /gen. ....	57
5.8. 4. Carpeta /assets.....	58
5.9. Layouts, Widtgets i altres vistes. ....	58
5.9.1. Layouts.....	58
5.9.2. Widgets i altres vistes. ....	61
5.10. Com començar a desenvolupar en Android.....	63
5.10.1. Entorn de desenvolupament. ....	63
5.10.2. Android SDK. ....	63
6. Codificació. ....	67



6.1. Android.....	67
6.1.1. Consum del servei web.....	67
6.1.1. Índex.....	69
6.1.2. Cas d'ús Log in.....	69
6.1.3. Menú principal.....	69
6.1.3. Cas d'ús mostrar perfil.....	70
6.1.4. Cas d'ús llistar amics.....	71
6.1.5. Cas d'ús consultar agenda.....	72
6.1.6. Cas d'ús demanar proposta.....	74
El cas d'ús demanar proposta consta de quatre activitats.....	74
6.1.7. Cas d'ús cercar.....	78
6.1.8. Activitat V_Base.....	81
6.1.9. Classe Sesion.....	82
6.1.10. Paquet domini.....	82
6.1.11. Paquet utils.....	82
6.2. Servei web.....	82
6.2.1. Paquet BBDD.....	82
6.2.2. Paquet WS.....	85
7. Manual d'usuari.....	87
7.1. Inici.....	87
7.2. Entrada a l'aplicació i menú principal.....	87
7.3. Perfil.....	89
7.4. Llista d'amics.....	90
7.5. Consulta agenda.....	91
7.6. Demanar proposta.....	92
7.7. Cercar.....	95
8. Proves.....	97
8.1. Log in.....	97
8.2. Perfil.....	97
8.3. Llistat d'amics.....	98
8.4. Consultar agenda.....	98
8.5. Demanar proposta.....	98

8.6. Cercar.....	99
9. Valoració econòmica.....	101
9.1 Anàlisi del projecte.....	101
9.2 Eines utilitzades.....	101
9.3 Codificació.....	102
9.4 Documentació.....	103
9.5 Hardware utilitzat.....	103
9.6 Suma total cost del projecte.....	103
10. Conclusions.....	105
11. Referències.....	107

## Índex de figures.

Figura 2.1 Diagrama de Gantt.....	6
Figura 3. 1 Esquema Arquitectura.....	7
Figura 3. 2 Diagrama de casos d'ús de l'aplicació Android.....	8
Figura 5. 1 HTC Dream.....	46
Figura 5. 2 Arquitectura Android .....	48
Figura 5. 3 Android Manifest .....	52
Figura 5. 4 Cicle de vida d'una aplicació.....	54
Figura 5. 5 Estructura projecte .....	56
Figura 5. 6 Carpeta src.....	56
Figura 5. 7 Carpeta res.....	57
Figura 5. 8 Carpeta gen.....	57
Figura 5. 9 Linear layout .....	58
Figura 5. 10 Relative layout .....	59
Figura 5. 11 Table layout.....	59
Figura 5. 12 Grid View .....	60
Figura 5. 13 Tab layout .....	60
Figura 5. 14 List View .....	60
Figura 5. 15 Date picker .....	61
Figura 5. 16 Time picker .....	61
Figura 5. 17 Form stuff.....	62
Figura 5. 18 Spinner .....	62
Figura 5. 19 Map view.....	62
Figura 5. 20 Web View .....	63
Figura 5. 21 AVD Manager.....	64
Figura 5. 22 Emulador .....	64
Figura 5. 23 Emulador executant WeDugu .....	65
Figura 6. 1 Cas d'ús perfil.....	70
Figura 6. 2 Cas d'ús llistar amics .....	71
Figura 6. 3 Cas d'ús consulta agenda.....	72
Figura 6. 4 Esquema activitats agenda .....	73

Figura 6. 5 Cas d'ús demanar proposta.....	74
Figura 6. 6 Esquema d'activitats cas d'ús demanar proposta.....	77
Figura 6. 7 cas d'ús cercar .....	78
Figura 6. 8 Esquema activitats cas d'ús cercar .....	81
Figura.6.9 Fórmula Haversine.....	84
Figura 7. 1 Índex WeDugu .....	87
Figura 7. 2 Log in.....	88
Figura 7. 3 Menú Principal.....	88
Figura 7. 4 Perfil usuari.....	89
Figura 7. 5 Missatge recordatori format dades.....	89
Figura 7. 6 Confirmació de canvis en el perfil .....	90
Figura 7. 7 Llista d'amics .....	90
Figura 7. 8 Perfil amic.....	91
Figura 7. 9 Consulta agenda.....	91
Figura 7. 10 Llista esdeveniments.....	92
Figura 7. 11 Selecció del tipus de proposta.....	92
Figura 7. 12 Llistat de propostes .....	93
Figura 7. 13 Informació proposta.....	93
Figura 7. 14 Mapa proposta.....	94
Figura 7. 15 Cercar.....	95
Figura 7. 16 Resultat de la cerca .....	96

## Índex de taules.

Taula 2. 1 Tasques.....	5
Taula 5. 1 Característiques Android .....	48
Taula 8. 1 Proves Log in .....	97
Taula 8. 2 Proves perfil.....	97
Taula 8. 3 Proves llistat amics .....	98
Taula 8. 4 Proves consulta agenda.....	98
Taula 8. 5 Proves demanar proposta.....	99
Taula 8. 6 Proves cercar.....	99
Taula 9. 1 Valoració anàlisi .....	101
Taula 9. 2 Valoració eines utilitzades .....	101
Taula 9. 3 Valoració codificació .....	102
Taula 9. 4 Valoració documentació .....	103
Taula 9. 5 Cost hardware .....	103
Taula 9. 6 Suma total cost projecte.....	103



## **Glossari de termes.**

Smartphone	Telèfon d'última generació
API	Interfície de programació d'aplicacions
JPA	Java Persistence API
MVC	Model-Vista-Controlador
IOS	Sistema operatiu mòbil d'Apple
XML	Llenguatge de marques extensible
HTTP	Protocol de transferència d'hipertext
WSDL	Llenguatge de descripció de serveis web
SOAP	Simple Object Acces Protocol
SMS	Servei de missatges curts
GPS	Sistema de posicionament global
LRU	Least Recently Used
HTML	Llenguatge de marques d'hipertext
IDE	Entorn de desenvolupament integrat
SDK	Kit de desenvolupament de software





# **1. Objectius**

## **1.1. Propòsit**

Aquest projecte es el resultat de sumar dos desitjos: aprendre a desenvolupar amb la plataforma Android i realitzar una aplicació web que aportí experiència i faciliti la inserció al món professional. Després d'una primera avaluació, va sorgir la idea de fusionar les dues idees en un sol projecte i realitzar-lo en parella, de manera que un fes l'aplicació mòbil i l'altre l'aplicació web.

L'aplicació mòbil seria una maqueta il·lustrativa dels coneixements adquirits al llarg dels mesos de desenvolupament mentre que l'aplicació web hauria de tenir un volum considerable ja que el propòsit era sintetitzar i anar més enllà del que ja s'havia après durant els tres anys.

## **1.2. Objecte**

L'objecte principal d'aquest projecte és respondre a l'eterna pregunta que sorgeix a gairebé totes les reunions d'amics: i ara què fem?

S'ha volgut desenvolupar una aplicació útil per l'usuari, que l'ajudés en el seu dia a dia, fent un ús intensiu d'una de les característiques que més exploten els smartphones, la geolocalització, que ajuda a l'usuari a conèixer més la ciutat, a obtenir informació just en el moment en que la necessita si es troba al carrer, o bé, a prendre decisions si l'usuari no té les idees del tot clares.



## **2. Introducció**

### **2.1. Què és WeDugu?**

WeDugu és una xarxa social basada en l'oferta d'esdeveniments d'oci. El seu objectiu es proposar als seus usuaris opcions atractives a partir del coneixement dels llocs a on acudeixen i dels seus gustos o rutines.

Gràcies a la localització geogràfica incorporada en els telèfons d'última generació (smartphones) i en els navegadors web més moderns, de manera fàcil es podrà saber on és l'usuari i en quina zona s'haurà de focalitzar la cerca.

La part social servirà perquè l'aplicació pugui conèixer l'entorn de l'usuari i així proposar-li opcions que hagin realitzat els seus amics i que potser l'interessin.

### **2.2. A qui va dirigit.**

La xarxa social WeDugu va dirigida a tot tipus d'usuaris majors de 18 anys. Vol aprofitar el fet de que moltes vegades la gent no sap on anar o com buscar el millor pla per divertir-se. Aquest tipus de gent perd el temps pensant i pensant, decidint amb la seva parella o amics que poden fer aquella tarda o nit, WeDugu els facilitarà una llista de propostes de coses a fer, a prop d'ells, en tant sols uns segons i només amb el requeriment d'estar enregistrat i haver omplert un perfil indicant els seus gustos.

### **2.3. Disseny funcional.**

#### **2.3.1. Punt de partida.**

El punt de partida serà la creació d'un perfil d'usuari on s'aportin unes dades bàsiques referents a preferències i professió que permetin a l'aplicació fer-se una idea de quin esdeveniment podria acceptar l'usuari.

#### **2.3.2. Utilització del producte.**

Una vegada obtingut el perfil es podria fer ús de l'aplicació mitjançant dos suports diferents:

- Aplicació per dispositius mòbils amb sistema operatiu Android.
- Aplicació Web.

### 2.3.3. Serveis.

WeDugu permetrà realitzar les següents accions:

- Gestió d'amistat: L'usuari tindrà l'opció d'afegir a la seva llista d'amics a altres usuaris amb el que tingui algun punt en comú.
- Propostes d'esdeveniments: En funció del perfil que s'anirà creant gracies a l'ús, el sistema escollirà esdeveniments disponibles que interessin a l'usuari, tenint en compte la seva situació geogràfica. Si a l'usuari l'interessa alguna de les opcions, l'acceptarà i aquesta romandrà com un esdeveniment programat en la seva agenda.
- Registre d'esdeveniments realitzats: Serà una manera de conèixer més a l'usuari i així encertar en futures propostes. Si l'usuari ha realitzat un esdeveniment en el dia d'avui, podrà enregistrar-ho a la seva agenda de la següent manera:

Introduirà el nom del lloc on ha estat i el sistema cercarà coincidències i mostrarà uns registres que s'apropin. Si el lloc no es troba entre els resultats, l'usuari podrà introduir el lloc a la base de dades i localitzar-lo.

- Planificació d'esdeveniments futurs: De la mateixa manera que en el registre d'esdeveniments ja realitzats, la planificació d'esdeveniments futurs servirà per tenir una idea més formada de l'usuari i permetrà a ell mateix disposar d'una agenda complerta que li permetrà organitzar el seu dia a dia.

El funcionament de la planificació serà el mateix que el del registre d'esdeveniments realitzats. L'usuari introduirà el lloc on vol anar i el sistema cercarà una sèrie de registres que s'apropin, si el lloc desitjat no es troba entre els resultats serà el propi usuari qui l'introdueixi i el localitzi.

Totes dues plataformes permetran a l'usuari consultar les seves amistats, accedir al servei de propostes d'esdeveniments i consultar esdeveniments programats a la seva agenda. La planificació d'esdeveniments futurs i el registre d'esdeveniments ja realitzats només estaran disponibles per a l'aplicació web.

## 2.4. Planificació.

En la taula següent es mostra la relació de les diferents tasques a realitzar per a la construcció del projecte, conjuntament amb les dates previstes per a la realització d'aquestes.

Nom de la tasca	Duració	Inici	Fi	Tasca predecessora	Recurs humà
Planificació	9 dies	dilluns 07/02/11	dijous 17/02/11		Núria;Pedro
Anàlisi(Casos d'ús)	9 dies	dilluns 07/02/11	dijous 17/02/11		Núria;Pedro
Tecnologies	9 dies	dilluns 07/02/11	dijous 17/02/11		Núria;Pedro
Disseny web	16 dies	divendres 18/02/11	divendres 11/03/11	1	Núria
Domini amb JPA	16 dies	divendres 18/02/11	divendres 11/03/11	1	Núria
Formació Android	16 dies	divendres 18/02/11	divendres 11/03/11	1	Pedro
Casos d'ús generals Web	20 dies	dissabte 12/03/11	dijous 07/04/11	5	Núria
Casos d'ús general Android	20 dies	dissabte 12/03/11	dijous 07/04/11	6	Pedro
Casos d'ús socials Web	10 dies	divendres 08/04/11	dijous 21/04/11	7	Núria
Casos d'ús socials Android	10 dies	divendres 08/04/11	dijous 21/04/11	8	Pedro
Casos d'ús agenda Web	22 dies	dimecres 27/04/11	dijous 26/05/11	9	Núria
Casos d'ús agenda Android	22 dies	dimecres 27/04/11	dijous 26/05/11	10	Pedro
Revisió final Web	6 dies	dijous 02/06/11	dijous 09/06/11	11	Núria
Revisió final Android	6 dies	dijous 02/06/11	dijous 09/06/11	12	Pedro
Documentació	92 dies	dilluns 07/02/11	dimarts 14/06/11		Núria;Pedro

Taula. 2.1 Tasques

A continuació es mostra el diagrama de Gantt de les tasques per a la realització del projecte:

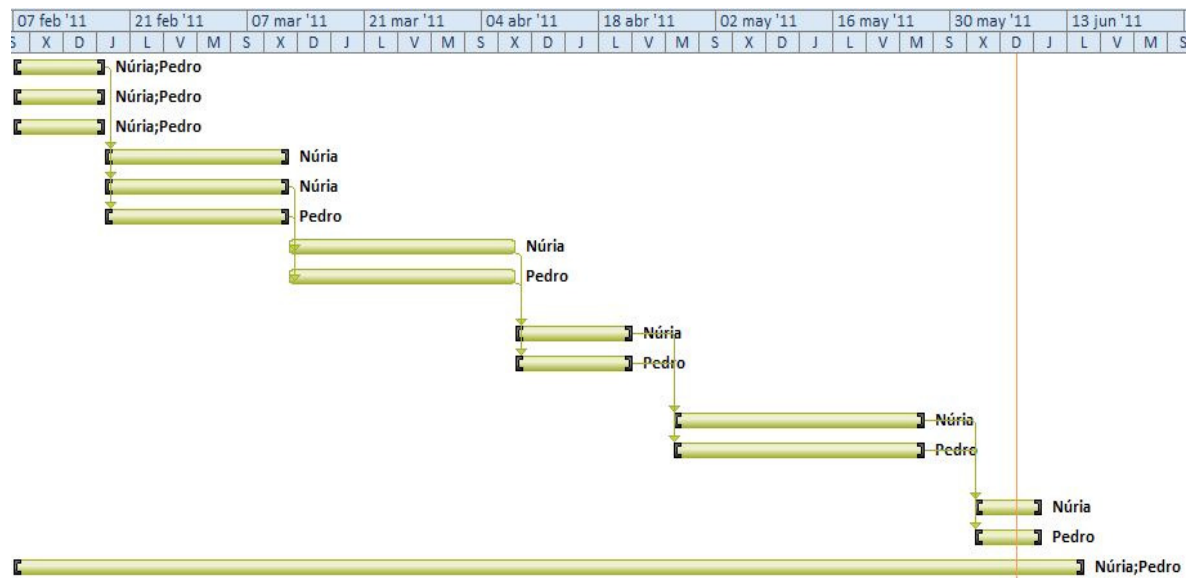


Figura 2. 1 Diagrama de Gantt

### 3. Anàlisi Tècnic.

En aquest apartat es definirà l'arquitectura sobre la qual es muntarà l'aplicació web i el servei web que proveirà d'informació a l'aplicació Android, els diferents casos d'ús que es podran realitzar tant des de l'aplicació web com des de l'aplicació Android i la base de dades on s'emmagatzemaran totes les dades.

#### 3.1. Arquitectura del projecte.

L'arquitectura del projecte serà la que es pot veure en la imatge següent:

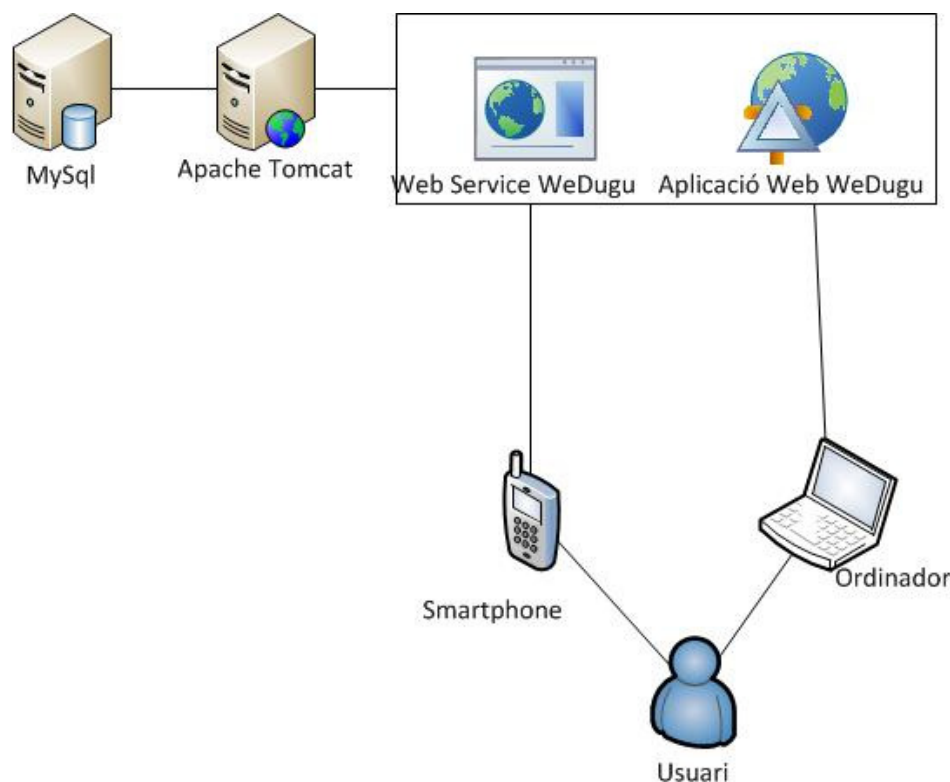


Figura 3. 1 Esquema Arquitectura

És necessari disposar d'un servidor de Base de dades on s'emmagatzemaran totes les dades referents a l'aplicació. S'ha escollit el sistema gestor de base de dades MySQL per la seva facilitat d'instal·lació i ús. Alhora la solució disposa també d'un servidor d'aplicacions Apache Tomcat que serà el contenidor tant de l'aplicació web com del servei web que proveirà a l'aplicació Android. Tots dos servidors (base de dades i aplicacions) s'executaran sobre una mateixa màquina. L'usuari farà servir tant l'aplicació web, com l'aplicació Android. En el cas de l'aplicació Android, aquesta,

internament, es comunicarà amb un servei web allotjat al servidor Apache Tomcat que, a la seva vegada es comunicarà amb el servidor de base de dades, evitant d'aquesta manera la connexió directa del smartphone amb la base de dades.

## 3.2. Casos d'ús.

A continuació es descriuran els casos d'ús que es podran realitzar des de l'aplicació Android i des de l'aplicació Web.

### 3.2.1. Casos d'ús Aplicació Android.

Diagrama de casos d'ús.

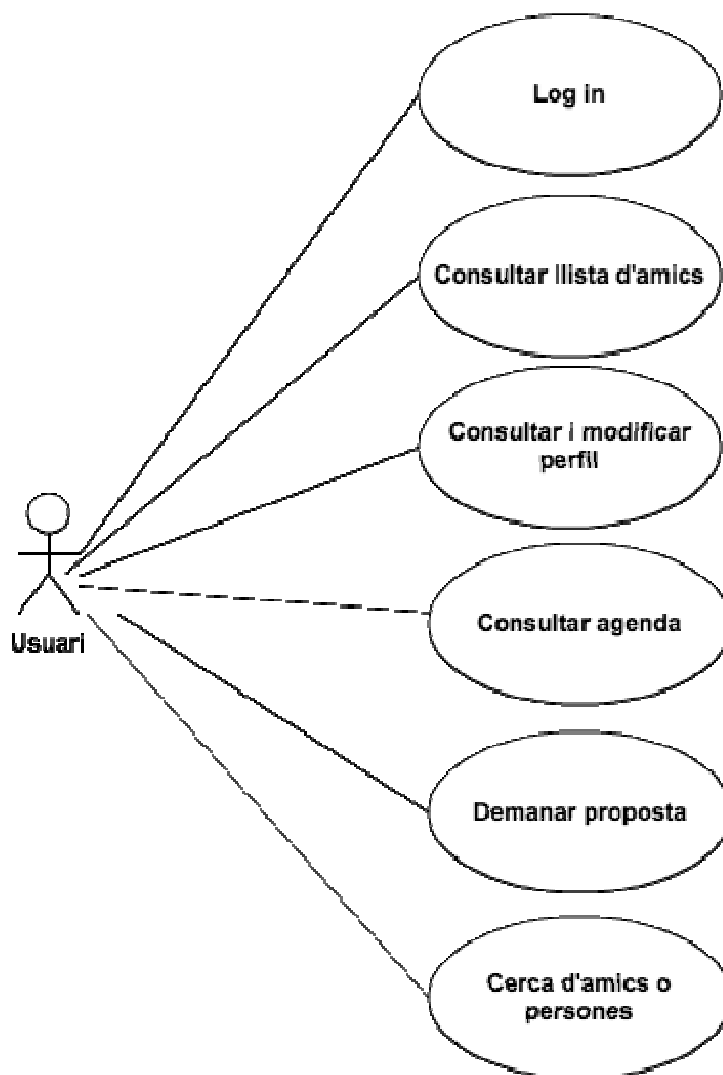


Figura 3. 2 Diagrama de casos d'ús de l'aplicació Android



**Descripcions de casos d'ús.****Log in.**

Actors: Usuari

Descripció: L'usuari vol iniciar sessió a l'aplicació.

Precondicions: La sessió no està iniciada.

Flux normal:

1. L'usuari introdueix el seu e-mail i la seva contrasenya.
2. L'e-mail existeix i la contrasenya és correcte
- 3 S'inicia sessió i es passa al menú de l'usuari.

Flux alternatiu:

2. L'e-mail no existeix o la contrasenya no són correctes
3. Es torna al pas 1.1

**Consultar llista d'amics.**

Actors: Usuari

Descripció: L'usuari vol consultar la seva llista d'amics

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem sobre el botó "Amics".
2. Es mostra per pantalla una llista amb tots els amics de l'usuari.
3. L'usuari prem sobre un element de la llista.
4. Es mostra el perfil de l'amic en que ha premut l'usuari.

Flux alternatiu:

2.1 L'usuari no té amics

2.2 Es mostra per pantalla un missatge indicant que l'usuari no té amics.

**Consulta i modificació del Perfil.**

Actors: Usuari

Descripció: L'usuari vol consultar i modificar el seu perfil.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem sobre el boto "Perfil".
2. Es mostra per pantalla el detall del perfil de l'usuari seguit d'un botó "Modificar".
3. L'usuari prem sobre "Modificar".
4. Es substitueix el detall del perfil per caixes de text omplertes amb els diferents camps del perfil.
5. Una vegada l'usuari a modificat les que creu adients prem sobre "Ok".
6. Es modifica el perfil de l'usuari i es torna al menú principal.

Flux Alternatiu:

3.1 L'usuari decideix tornar enrere i es torna al menú principal

5.1 L'usuari decideix tornar enrere i es torna al menú principal.

**Consultar agenda.**

Actors: Usuari

Descripció: L'usuari vol consultar la seva agenda

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem sobre l'opció "Agenda".
2. Es demana per pantalla el dia i el mes del qual es vol consultar els esdeveniments que es troben inclosos a l'agenda.
3. Es mostra una llista amb els esdeveniments programats per el dia indicat.

Flux Alternatiu:

- 2.1 L'usuari decideix tornar enrere i es torna al menú principal.

### **Demanar proposta.**

Actors: Usuari, Sistema

Descripció: El sistema proposa una activitat a l'usuari .

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem sobre l'opció "Propuesta".
2. Es mostra per pantalla un desplegable amb els tipus de proposta que estan disponibles i un botó per que siguin mostrades les propostes.
3. L'usuari prem sobre el botó "GO!"
4. El sistema obté les coordenades actuals en que es troba l'usuari i llença una consulta per tal d'obtenir les propostes segons el tipus indicat per l'usuari, filtrades per les preferències indicades al perfil d'aquest i la posició geogràfica en que es troba actualment.
5. Es mostra per pantalla una llista amb les propostes.

6. L'usuari prem sobre una de les propostes per obtenir més detall.
7. Es mostra per pantalla tota la informació de la proposta i dos botons, el primer per veure un mapa, el segon per incloure la proposta a l'agenda de l'usuari.
8. L'usuari prem sobre "Ver Mapa".
9. Es mostra un mapa de Google Maps on es troba marcada la posició geogràfica de l'usuari i la posició geogràfica de la proposta.
10. L'usuari torna enrere.
11. Es torna a mostrar la informació detallada de la proposta.
12. L'usuari prem sobre "Guardar".
13. La proposta és inclosa a l'agenda de l'usuari i es mostra un missatge per pantalla confirmant-ho.

#### Flux Alternatiu:

- \* Des de qualsevol punt l'usuari pot tornar a la pantalla anterior.

#### **Cerca d'amics o persones.**

Actors: Usuari

Descripció: L'usuari vol cercar usuaris entre els seus amics o entre la resta d'usuaris.

Precondicions: La sessió està iniciada.

#### Flux normal:

1. L'Usuari prem sobre el botó "Buscar" al menú principal.
2. Es mostra per pantalla un caixa de text, on l'usuari introduirà la cerca, i un botó.
3. L'Usuari introdueix la cerca i prem sobre el botó "Buscar".

4. Es mostra per pantalla una visió amb dues pestanyes. La pestanya fixada per defecte mostra els resultats de la cerca entre els amics de l'usuari actual.
5. L'usuari prem sobre la pestanya "Personas".
6. Es mostra el resultat de la cerca entre els usuaris de l'aplicació que no son amics de l'usuari actual.

#### Flux Alternatiu:

- \* L'usuari pot tornar enrere des de qualsevol punt

### 3.2.2. Casos d'ús Aplicació Web

#### Diagrama de casos d'ús.

Aquest és l'esquema dels casos d'ús implementats a l'aplicació separats per temàtiques:

#### Generals:

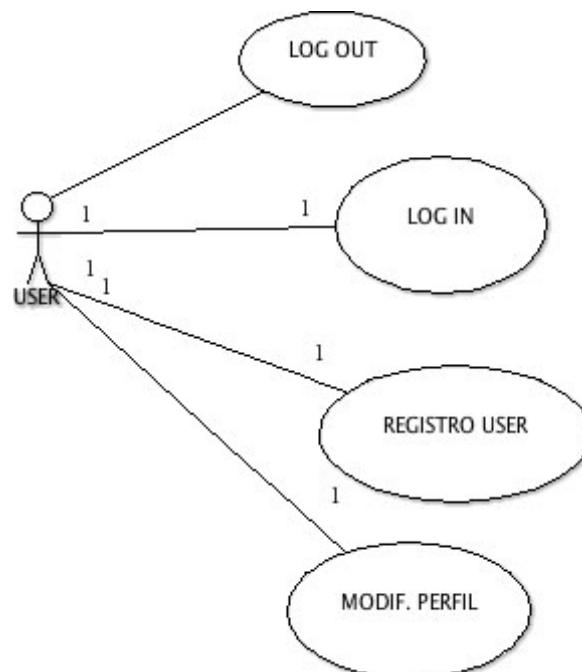


Figura 3. 3 Diagrama de casos d'ús generals

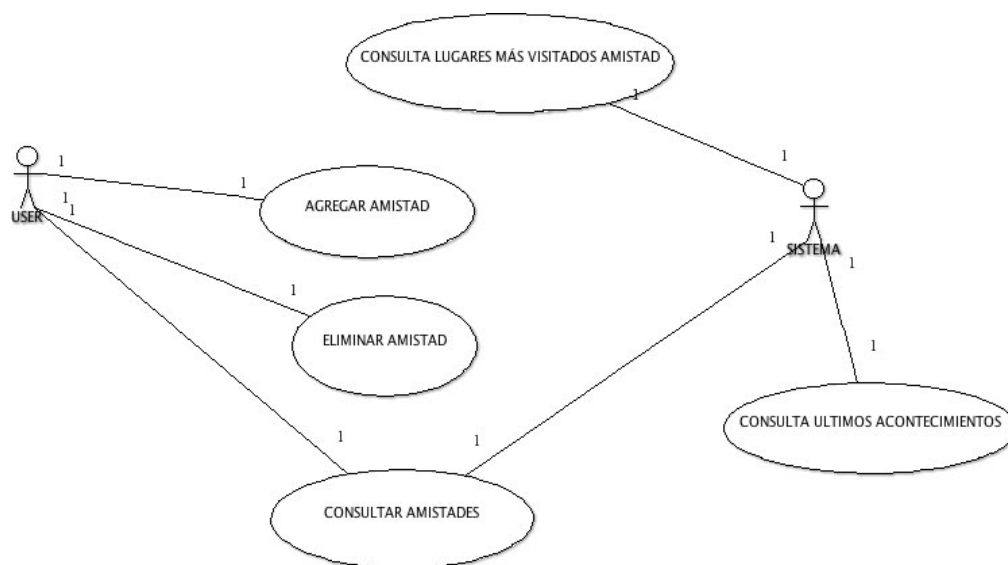
**Referents a Amistats:**

Figura 3. 4 Diagrama de casos d'ús d'amistats

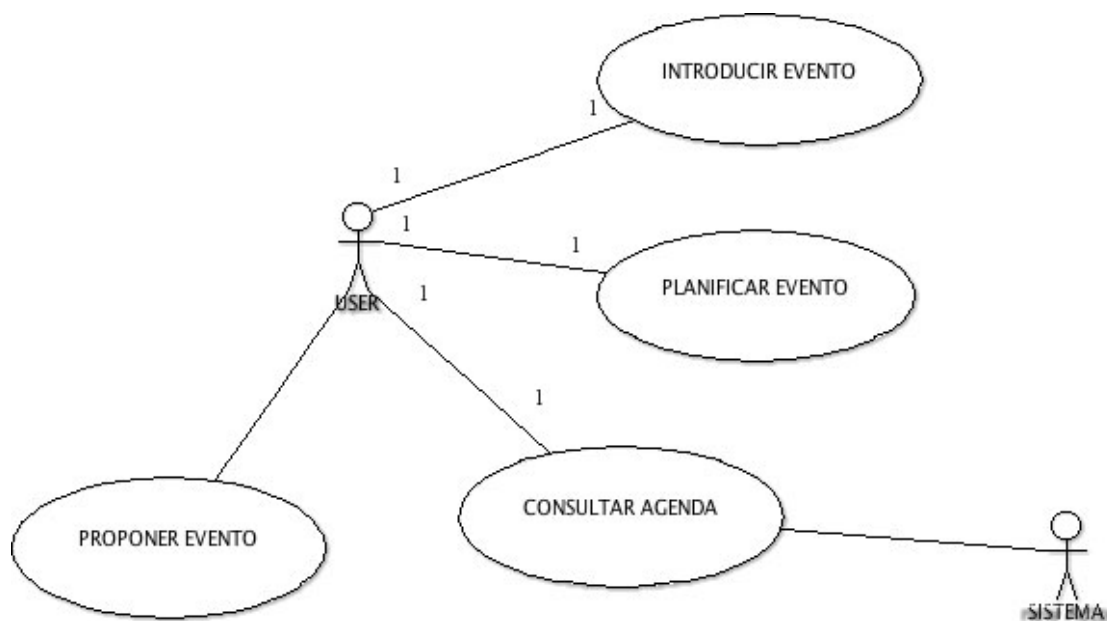
**Referents a l'agenda:**

Figura 3. 5 Diagrama de casos d'ús agenda

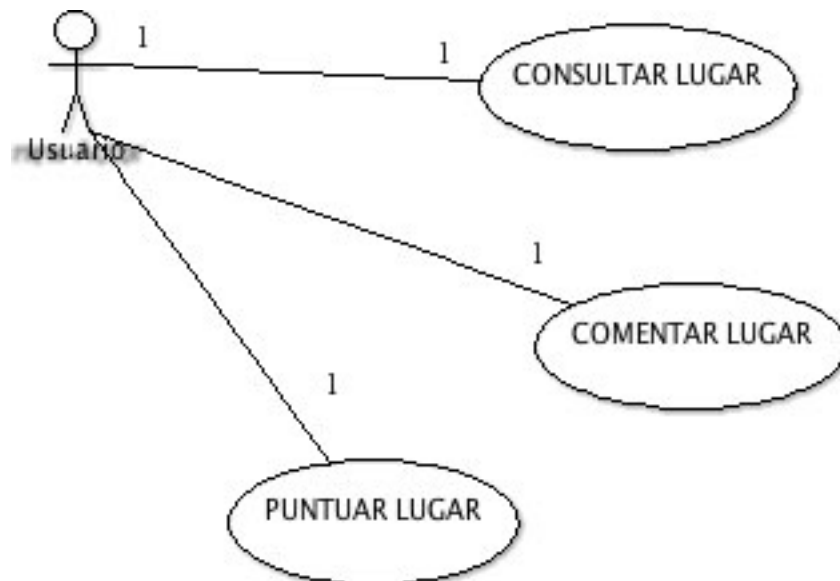
**Referents al lloc:**

Figura 3. 6 Diagrama de casos d'ús del lloc

**Descripcions de casos d'ús.****Generals:****- Log in:**

Actors: Usuari

Descripció: L'usuari vol iniciar sessió a l'aplicació.

Precondicions: La sessió no està iniciada.

Flux normal:

1. L'usuari introdueix el nom d'usuari i la contrasenya.
2. L'usuari prem al botó iniciar sessió.
3. Les credencials són correctes i el sistema el redirigeix a la pàgina principal.

Flux alternatiu:

- 3.a. L'usuari ha introduït un nom d'usuari que no existeix.
- 3.b. L'usuari introdueix una contrasenya equivocada.
- 4.a. El sistema mostra un missatge indicant que l'usuari no existeix a la Base de Dades.

4.b. El sistema mostra un missatge indicant que la contrasenya no és correcta.

- **Log out:**

Actors: Usuari

Descripció: L'usuari vol tancar la sessió a l'aplicació.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem al botó tancar la sessió.
2. El sistema elimina la sessió actual i la cookie afegida al navegador per que mantingui la sessió oberta.
3. El sistema redirigeix a l'usuari a la pàgina inicial.

- **Registro user:**

Actors: Usuari

Descripció: L'usuari vol registrar-se a l'aplicació.

Precondicions:

Flux normal:

1. L'usuari introdueix totes les dades demanades al formulari inicial.
2. L'usuari prem al botó registrar.
3. El sistema dona d'alta a l'usuari i el redirigeix a la pàgina que conté el formulari per la creació de la primera fase del perfil.
4. L'usuari omple els camps demanats i prem al botó "Siguiete" per passar a la segona fase.
5. El sistema dona d'alta les dades indicades i mostra el formulari per la creació dels estils d'usuari.
6. L'usuari afegix els estils corresponents al seu perfil i prem al botó "Siguiete".
7. El sistema dona d'alta les dades indicades i mostra el formulari per pujar una imatge de perfil.
8. L'usuari escull una imatge del seu directori d'arxius.
9. Es genera la petició al servidor que portarà la imatge i el sistema emmagatzemarà aquesta al servidor.
10. L'usuari prem al botó "Finalizar".
11. El sistema redirigeix a l'usuari a la pàgina principal amb la sessió iniciada.



Flux alternatiu:

1.a. L'usuari deixa un camp buit.

1.b. L'usuari introdueix una data de naixement que no correspon a tenir 18 anys.

1.c. L'usuari introdueix una direcció d'e-mail mal formulada.

1.d. L'usuari introdueix una direcció d'e-mail que ja està donada d'alta a la Base de Dades.

1.e. L'usuari introdueix dues contrasenyes que no coincideixen.

3.a. El sistema mostra un missatge advertint que s'ha d'informar totes les dades.

3.b. El sistema mostra un missatge advertint que s'ha de ser major de 18 anys.

3.c. El sistema mostra un missatge advertint que la direcció d'e-mail no és correcte.

3.d. El sistema mostra un missatge advertint que l'usuari ja existeix.

3.e. El sistema mostra un missatge advertint que les contrasenyes no coincideixen.

4.a. L'usuari no indica totes les dades.

5.a. El sistema passa a mostrar el formulari següent sense registrar les dades que passen a quedar buides a la Base de dades.

8.a. L'usuari no puja cap imatge al servidor.

9.a. No es genera cap petició.

-

- **Modif. Perfil:**

Actors: Usuari

Descripció: L'usuari vol modificar alguna de les dades del seu perfil.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem al botó d'edició de dades personals.
2. El sistema canvia la vista per mostrar les dades editables.
3. L'usuari introdueix els canvis necessaris.
4. L'usuari prem al botó guardar.
5. El sistema emmagatzema les dades i torna a canviar la vista a mode no editable.

Flux alternatiu:

- 1.a. L'usuari prem al botó d'edició d'estils de música.
- 1.b. L'usuari prem al botó d'edició d'estils de menjar.
- 1.c. L'usuari prem al botó d'edició d'estils de beguda.
- 1.d. L'usuari prem al botó d'edició d'estils d'activitats.
- 1.e. L'usuari prem al botó d'edició d'estils de cinema.

**Referents a Amistats:****- Consulta lugares más visitados amistades:**

Actors: Sistema

Descripció: Es busquen els llocs més visitats de la llista d'amistats.

Precondicions: La sessió està iniciada. L'usuari té amistats.

Flux normal:

1. El sistema recull la llista d'amistats de l'usuari.
2. Per cada amiat, el sistema recull la llista d'esdeveniments realitzats.
3. Un cop obtinguda la llista d'esdeveniments, el sistema calcula els 7 més visitats.
4. El sistema mostra una llista amb els 7 llocs més visitats.

Flux alternatiu:

- 3.a. Les amistats de l'usuari encara no han realitzat esdeveniments.
- 4.a. El sistema mostra un missatge indicant que encara no hi ha informació d'esdeveniments de les seves amistats.

- **Consulta últimos acontecimientos:**

Actors: Sistema

Descripció: Es genera un llistat amb els últims esdeveniments ocorreguts a l'aplicació, entre les amistats i el propi usuari.

Precondicions: La sessió està iniciada.

Flux normal:

1. El sistema recull el llistat d'esdeveniments de l'usuari de la Base de Dades.
2. El sistema suma la llista d'esdeveniments de cada usuari que és amestat a la llista anterior.
3. La llista total, el sistema l'ordena per data d'ocurrència.
4. El sistema mostra una llista amb els últims esdeveniments.

Flux alternatiu:

- 3.a. Ni l'usuari ni les seves amistats han realitzat cap esdeveniment.
- 4.a. El sistema mostra un missatge indicant que encara no hi ha informació d'esdeveniments.

- **Agregar Amistad:**

Actors: Usuari

Descripció: L'usuari vol afegir una persona com amestat.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari prem al botó "Agregar amistad".
2. El sistema mostra una finestra amb un camp per introduir el nom de la persona.
3. L'usuari introdueix el nom de la persona que busca.
4. L'usuari prem al botó "Buscar amigo".
5. El sistema mostra un llistat amb usuaris el nom del qual s'aproxima al text introduït.
6. L'usuari selecciona de la llista la persona que cercava.
7. L'usuari prem al botó "Agregar".

8. El sistema afegeix a la Base de Dades a aquesta persona com amestat seva.
9. El sistema torna a mostrar la pàgina principal on ja figura la nova amestat a la caixa d'amistats.

Flux alternatiu:

- 3.a. L'usuari no introdueix un nom.
- 5.a. El sistema mostra un missatge indicant que ha d'indicar un nom.
- 5.b El sistema no troba coincidències. Mostra un missatge indicant el fet i l'opció de realitzar una nova cerca o sortir de la finestra.
- 6.c L'usuari no selecciona cap persona.
- 6.d L'usuari prem a sobre de la persona.
- 7.d El sistema obre una nova finestra amb el perfil de la persona premuda.
- 8.c El sistema mostra un missatge advertint que ha de seleccionar la persona que vol agregar com amestat.

- **Eliminar amestad:**

Actors: Usuari

Descripció: L'usuari vol eliminar de la seva llista d'amistats una persona.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari accedeix al perfil d'una amestat.
2. El sistema mostra un missatge indicant que la persona és amestat nostre i l'opció d'eliminar-la de la nostre llista.
3. L'usuari prem al text "Quiero dejar de serlo" (per eliminar-la).
4. El sistema esborra aquesta persona de la llista d'amistats.
5. El sistema redirigeix a l'usuari a la pàgina principal.

- **Consultar amistades:**

Actors: Sistema, usuari

Descripció: Es genera un llistat amb totes les amistats de l'usuari.

Precondicions: La sessió està iniciada.

Flux normal:

1. El sistema recull de la Base de Dades les amistats de l'usuari.
2. El sistema genera una llista i en mostra 5 persones diferents cada vegada que s'executa la vista.

Flux alternatiu:

- 2.a. L'usuari no té amistats. El sistema mostra un missatge indicant-ho.
- 3.b. L'usuari vol veure la llista sencera d'amistats i prem al botó "+Muestra todos".
- 4.b. El sistema obre una finestra amb totes les amistats de l'usuari ordenades per ordre alfabètic.

**Referents a l'agenda:****- Introducir evento:**

Actors: Usuari

Descripció: L'usuari indica on ha estat.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari introdueix el nom del lloc on ha estat a la caixa "Donde has estado hoy?".
2. El sistema genera una llista de llocs que s'aproximen al text introduït per l'usuari i la mostra.
3. L'usuari selecciona de la llista el lloc que cercava.
4. L'usuari prem al botó "Ok".
5. El sistema emmagatzema el nou esdeveniment amb data d'avui.
6. El sistema mostra un missatge preguntant si vol continuar a la pàgina principal o bé consultar l'agenda.

Flux alternatiu:

- 2.a. El sistema no troba cap coincidència amb el text introduït i mostra el formulari per donar d'alta el lloc.

3.a. L'usuari selecciona el tipus de lloc.

3.a.1 El sistema mostra el formulari per donar d'alta el perfil del lloc.

3.a.2 L'usuari prem al botó "Ok".

3.a.3 El sistema mostra un mapa on l'usuari haurà d'indicar la ubicació del lloc.

3.a.4 L'usuari indica la ubicació del lloc.

3.a.5 L'usuari prem al botó "Ok".

3.b El lloc indicat és un cinema. El sistema mostra el formulari per donar d'alta el lloc afegint el camp nom de la pel·lícula.

3.b.1 L'usuari introdueix el nom de la pel·lícula.

3.b.2.a El sistema busca pel·lícules el nom de les quals s'aproximi al text introduït i genera un llistat.

3.b.2.b El sistema no troba coincidències, per tant entén que la pel·lícula s'haurà de donar d'alta.

3.b.3.a L'usuari selecciona de la llista la pel·lícula vista i prem al botó "Acceptar".

3.b.3.b L'usuari no selecciona cap pel·lícula i prem al botó "Acceptar".

3.b.3.b.1 El sistema mostra un missatge informant que l'ha de seleccionar.

5.a El sistema emmagatzema el lloc nou i l'esdeveniment a l'agenda de l'usuari.

7.e L'usuari no indica la ubicació del lloc.

8.e El sistema mostra un missatge indicant que ha de fer-ho.

- **Planificar evento:**

Actors: Usuari

Descripció: L'usuari afegeix a l'agenda un esdeveniment que realitzarà pròximament.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari tecleja sobre un dia qualsevol de l'agenda (el dia que vol realitzar l'activitat).
2. El sistema obre una finestra amb un formulari on figura la data de l'esdeveniment a planificar i la llista d'estils.
3. L'usuari escull l'estil que desitja.
4. El sistema genera un llistat de propostes segons els gustos de l'usuari per l'estil indicat i les mostra a una nova finestra.
5. L'usuari selecciona del llistat el lloc al que desitja anar.
6. L'usuari prem al botó "Añadir Evento".
7. El sistema emmagatzema l'esdeveniment a l'agenda
8. El sistema pregunta a l'usuari si vol dirigir-se a la pàgina principal o continuar a l'agenda.

Flux alternatiu:

- 4.a. El sistema no troba propostes adequades a l'usuari i mostra un missatge indicant-ho.
- 5.b. L'usuari no selecciona cap lloc del llistat.
- 7.b. El sistema mostra un missatge indicant que ha de seleccionar el lloc desitjat.
- 5.c L'usuari prem a sobre d'un lloc de la llista.
- 6.c El sistema obre una nova finestra amb el perfil del lloc demanat [cas d'ús consultar lugar].

- **Consultar agenda:**

Actors: Usuari, sistema

Descripció: Es mostren els esdeveniments emmagatzemats a la Base de Dades.

Precondicions: La sessió està iniciada.

Flux normal:

1. El sistema genera un llistat amb els esdeveniments que s'ha de realitzar pròximament (s'han planificat).
2. El sistema mostra el llistat a la pàgina principal i la pàgina de perfil de l'usuari.

Flux alternatiu:

2.a. No hi han esdeveniments pròxims programats i el sistema mostra un missatge informant-ho.

1.b. L'usuari vol consultar l'agenda.

2.b. El sistema obté tots els esdeveniments de la Base de Dades.

3.c El sistema col·loca al dia del calendari pertinent cada esdeveniment.

- **Proponer evento:**

Actors: Usuari

Descripció: L'usuari vol propostes de llocs al seu voltant que podrien interessar-li.

Precondicions: La sessió està iniciada. L'usuari ens permet obtenir la seva ubicació des del seu navegador.

Flux normal:

1. L'usuari prem al botó "No sabes donde ir o qué hacer? Accede a tus propuestas!"
2. El sistema obté la ubicació de l'usuari.
3. El sistema genera un llistat amb una proposta per tipus de lloc que es trobi dintre d'un radi de 3km i comparteixi perfil amb l'usuari.
4. L'usuari selecciona una proposta que l'interessa.
5. L'usuari prem al botó "Go!";
6. El sistema dona d'alta l'esdeveniment a la Base de Dades.
7. El sistema pregunta a l'usuari si vol dirigir-se a la pàgina principal o continuar a l'agenda.

Flux alternatiu:

3.a. El sistema no troba cap proposta per l'usuari i mostra un missatge informant-ho.



3.a.1. L'usuari decideix obtenir una llista per tipus de lloc i prem al botó "Afinar el resultado".

3.a.2 El sistema mostra la llista de tipus de llocs per l'usuari escollit.

3.a.3 L'usuari selecciona el tipus del qual vol propostes.

3.a.4.a El sistema genera una llista amb 5 propostes només d'aquest tipus de lloc i la mostra.

3.a.4.b El sistema no troba propostes per aquest tipus de lloc i mostra un missatge indicant-ho.

4.a L'usuari no selecciona una proposta de la llista.

6.a El sistema mostra un missatge indicant que ha de seleccionar el pla que accepta.

4.b L'usuari prem a sobre d'un lloc.

5.b El sistema obre una nova finestra amb el perfil del lloc [cas d'ús consultar lugar].

### Referents al lloc:

#### - Consultar lugar:

Actors: Usuari.

Descripció: Es mostren les dades referents a un lloc.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari a premut al nom d'un lloc.
2. El sistema obté les dades del lloc i la seva valoració.
3. El sistema obre una nova finestra on mostra totes aquestes dades.

Flux alternatiu:

3.a. Si el lloc encara no té valoració feta (comentaris o puntuació), el sistema mostrarà un missatge que indicarà que encara no ha estat valorat.

#### - Comentar lugar:

Actors: Usuari.

Descripció: S'afegeix un comentari a les valoracions del lloc.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari a premut al botó "Agregar un comentario".
2. El sistema obre un diàleg amb una caixa on l'usuari introduirà el comentari.
3. L'usuari escriu el comentari.
4. L'usuari prem al botó "Añade".
5. El sistema emmagatzema el comentari a la Base de Dades.
6. El sistema mostra el comentari afegit a la llista de comentaris.

Flux alternatiu:

- 3.a. L'usuari no escriu res a l'espai destinat al comentari.
- 4.a El sistema mostra un missatge informant que el comentari no pot ser buit.

- **Puntuar lugar:**

Actors: Usuari.

Descripció: S'afegeix una puntuació a les valoracions del lloc.

Precondicions: La sessió està iniciada.

Flux normal:

1. L'usuari a premut a sobre de la nota mitja del lloc o bé a sobre del número de votacions.
2. El sistema obre un desplegable amb puntuacions del 0 al 10.
3. L'usuari prem a la nota desitjada.
4. L'usuari prem al botó de tancar el desplegable.
5. El sistema emmagatzema la puntuació de l'usuari
6. El sistema re-calcula la nota mitja del lloc i la mostra.

Flux alternatiu:

- 1.a. El lloc no tenia puntuacions encara, per tant, l'usuari prem a sobre del text "Todavía no ha sido valorado" o bé el número de votacions (en aquest cas 0).
- 2.b.1 Si l'usuari havia puntuat anteriorment aquest lloc, el sistema mostra la puntuació que li va donar anteriorment.

3.c L'usuari no marca cap puntuació.

5.c El sistema tanca el desplegable sense emmagatzemar res.

### **3.3. Descripció de la base dades.**

La Base de Dades d'aquest projecte és comú per l'aplicació web i l'aplicació mòbil.

#### **3.3.1. Model de Dades.**

A continuació es passarà a descriure el model de dades de l'aplicació. El diagrama general és bastant gran, per tant, és millor descriure taula per taula les dades necessàries. Només s'explicaran les relacions a una de les taules d'una de les dos direccions per no duplicar explicacions.

**Diagrama físic:**



**Taula Usua:** És la taula on s'emmagatzemen les dades bàsiques de l'usuari.

Atributs:

- USUA\_ID: És generat automàticament i són incrementals.
- USUA\_SEXO: Sexe de l'usuari. F: Femení, M: Masculí o no informat.
- USUA\_CITY\_ORIG: Ciutat on va néixer l'usuari. Pot no ser informat.
- USUA\_PASSWORD: Contrasenya de l'usuari per accedir a l'aplicació.
- USUA\_CITY\_ACT: Ciutat on viu l'usuari. Pot no ser informat.
- USUA\_APELLIDOS: Cognoms de l'usuari.
- USUA\_PROF: Ocupació actual de l'usuari. Pot no ser informat.
- USUA\_EMAIL: Direcció d'e-mail que l'usuari utilitzarà com a nom d'usuari. Aquest camp és únic i no poden existir dos usuaris amb la mateixa direcció d'e-mail.
- USUA\_ESTU: Estudis realitzats per l'usuari. Pot no ser informat.
- USER\_NOMBRE: Nom de l'usuari.
- USUA\_TIENEFOTO: Camp que indica si l'usuari ha pujat una imatge de perfil al servidor. Valor 1: Té foto, valor 0: no té foto.
- USUA\_FNAC: Data de naixement de l'usuari. Ha de ser major a 18 anys.

Relacions:

- AGENDA. La taula usuari té una relació 1-1 amb Agenda. Un usuari té una agenda i una agenda és d'un usuari. S'identifica per l'identificador de l'agenda.
- PERFIL. Relació 1-1 amb Perfil. Un usuari té un perfil i un perfil és d'un usuari. S'identifica per l'identificador del perfil.
- AMISTAD. Un usuari pot tenir 0 o més usuaris com amestat.
- COMENT. Un usuari pot realitzar 0 o més comentaris.
- VALO. Un usuari pot realitzar 0 o més puntuacions.

**Taula Perf:** És la taula on s'emmagatzemen els gustos de l'usuari.

Atributs:

- PERF\_ID: És generat automàticament i són incrementals.

Relacions:

- PERF\_EST\_ACTIV. La taula on es relacionen els estils d'activitat escollits per l'usuari. Pot tenir 0 o varis.

- PERF\_EST\_CINE. La taula on es relacionen els estils de cinema escollits per l'usuari. Pot tenir 0 o varis.
- PERF\_EST\_BEBIDA. La taula on es relacionen els estils de beguda escollits per l'usuari. Pot tenir 0 o varis.
- PERF\_EST\_MUSIC. La taula on es relacionen els estils de música escollits per l'usuari. Pot tenir 0 o varis.
- PERF\_EST\_COMIDA. La taula on es relacionen els estils de menjar escollits per l'usuari. Pot tenir 0 o varis.

**Taula Agen:** És la taula on s'emmagatzemen els esdeveniments realitzats per l'usuari.

Atributs:

- AGEN\_ID: És generat automàticament i són incrementals.

Relacions:

- AGEN\_EVENTO. La taula on es relacionen els esdeveniments realitzats per l'usuari. S'identifica per l'identificador de l'agenda i l'identificador de l'esdeveniment. Un usuari pot realitzar 0 o varis esdeveniments.

**Taula Evento:** És la taula on s'emmagatzema el tipus d'esdeveniment realitzat i la seva data.

Atributs:

- EVENT\_ID: És generat automàticament i són incrementals.
- EVENT\_DATE: És la data quan s'ha realitzat l'esdeveniment.
- EVENT\_FECH\_HORA: És la data quan s'ha introduït aquest esdeveniment a la Base de Dades.

Relacions:

- CINE. Relació 0-varis amb la taula Cinema que indica que un esdeveniment pot ser o no un cinema i un cinema pot ser a varis esdeveniments.
- CONCIÉ. Relació 0-varis amb la taula Concert que indica que un esdeveniment pot ser o no un concert i un concert pot ser a varis esdeveniments.
- PELI. Relació 0-varis amb la taula Pel·lícula que indica que un esdeveniment pot ser o no una pel·lícula i una pel·lícula pot ser a varis esdeveniments.

- ACTI. Relació 0-varis amb la taula Activitat que indica que un esdeveniment pot ser o no una activitat i una activitat pot ser a varis esdeveniments.
- PUB\_BAR. Relació 0-varis amb la taula Pub\_Bar que indica que un esdeveniment pot ser o no una pub o bar i un pub o bar pot ser a varis esdeveniments.
- RESTAU. Relació 0-varis amb la taula Restaurant que indica que un esdeveniment pot ser o no un restaurant i un restaurant pot ser a varis esdeveniments.

**Taula Cine:** És la taula on s'emmagatzema les dades d'un cinema.

Atributs:

- CINE\_ID: És generat automàticament i són incrementals.
- CINE\_DIREC: És la direcció on es troba el cinema.
- CINE\_PREUNORMAL: Preu que ofereix aquest cinema a sessions d'un dia normal.
- CINE\_DESC: Observacions que es vulguin indicar sobre el cinema.
- CINE\_PREUREDUCIDA: Preu que ofereix aquest cinema a sessions del dia de l'espectador o bé sessions que ofereixen descompte.
- CINE\_LATIT: Coordenada latitud on s'ubica el cinema.
- CINE\_PREUFEST: Preu que ofereix aquest cinema a sessions de cap de setmana, o bé, dies festius.
- CINE\_LONGI: Coordenada longitud on s'ubica el cinema.
- CINE\_DIAREDU: Dia de la setmana que el cinema té com a dia de l'espectador.
- CINE\_NOMB: Nom del cinema.

Relacions:

- CINE\_SESION. Taula que relaciona el cinema amb les sessions que es projecten.

**Taula Sesion:** És la taula on s'emmagatzema les dades d'una sessió.

Atributs:

- SESION\_ID: És generat automàticament i són incrementals.
- SESION\_DNORMAL: Indica si la sessió és d'un dia considerat normal (ni festiu, ni reduït). Valor 0: no és un dia normal, valor 1: si és un dia normal.
- SESION\_DREDUCIDA: Indica si la sessió és d'un dia considerat tarifa reduïda. Valor 0: no és un dia reduït, valor 1: si és un dia reduït.

- SESSION: Hora a la qual es projecta la pel·lícula.
- SESSION\_DFINDE: Indica si la sessió és d'un dia considerat tarifa festiva o cap de setmana. Valor 0: no és un dia festiu, valor 1: si és un dia festiu.

Relacions:

- PELICULA. Relació 1-1 amb la pel·lícula que es projecta a aquesta sessió.

**Taula Peli:** És la taula on s'emmagatzema les dades d'una pel·lícula.

Atributs:

- PELI\_ID: És generat automàticament i són incrementals.
- PELI\_DURACION: Indica si el temps que dura la pel·lícula.
- PELI\_DATE\_INI: Indica la data quan comença a projectar-se la pel·lícula (l'estrena).
- PELI\_DATE\_FIN: Indica la data quan deixa de projectar-se la pel·lícula (sur de la cartellera).
- PELI\_DESC: Sinopsis de la pel·lícula.
- PELI\_NOMB: Nom de la pel·lícula.

Relacions:

- PELI\_EST\_CINE. Taula que relaciona la pel·lícula amb els estils amb els quals es pot classificar.

**Taula Acti:** És la taula on s'emmagatzema les dades d'una activitat.

Atributs:

- ACTI\_ID: És generat automàticament i són incrementals.
- ACTI\_LONGI: Coordenada longitud on s'ubica l'activitat.
- ACTI\_LATIT: Coordenada latitud on s'ubica l'activitat.
- ACTI\_NOMB: Nom de l'activitat.
- ACTI\_DIREC: Direcció on es realitza l'activitat.
- ACTI\_PREU: Preu de l'activitat.
- ACTI\_DATE\_INI: Data i hora quan comença a realitzar-se l'activitat.
- ACTI\_DESC: Descripció de l'activitat.
- ACTI\_DATE\_FIN: Data i hora quan finalitza l'activitat.

Relacions:



- ACTI\_EST\_ACTIV. Taula que relaciona l'activitat amb els estils amb els quals es pot classificar.

**Taula Concie:** És la taula on s'emmagatzema les dades d'un concert.

Atributs:

- CONCIIE\_ID: És generat automàticament i són incrementals.
- CONCIIE\_LONGI: Coordenada longitud on s'ubica el concert.
- CONCIIE\_LATIT: Coordenada latitud on s'ubica el concert.
- CONCIIE\_NOMB: Nom del concert.
- CONCIIE\_DIREC: Direcció on es realitza el concert.
- CONCIIE\_PRECIO: Preu del concert.
- CONCIIE\_DATE\_INI: Data i hora quan comença el concert.
- CONCIIE\_DESC: Descripció del concert.
- CONCIIE\_DATE\_FIN: Data i hora quan finalitza el concert.

Relacions:

- CONCIIE\_EST\_MUSIC. Taula que relaciona el concert amb els estils amb els quals es pot classificar.
- PUB\_BAR. Hi ha la possibilitat que el concert es realitzi a un pub o bar que existeix a la base de dades. Per tant aquesta relació indica que un concert pugui realitzar-se a 0 o 1 pub o bar i a aquest pub o bar es realitzin 0 o varis concerts.

**Taula Pub\_Bar:** És la taula on s'emmagatzema les dades d'un pub o bar.

Atributs:

- PU\_BAR\_ID: És generat automàticament i són incrementals.
- PU\_BAR\_LONGI: Coordenada longitud on s'ubica el pub o bar.
- PU\_BAR\_LATIT: Coordenada latitud on s'ubica el pub o bar.
- PU\_BAR\_NOMB: Nom del pub o bar.
- PU\_BAR\_DIREC: Direcció on s'ubica el pub o bar.
- PU\_BAR\_DESC: Descripció del pub o bar.

Relacions:

- PUB\_BAR\_EST\_MUSIC. Taula que relaciona el pub o bar amb els estils de música amb els quals es pot classificar.

- PUB\_BAR\_EST\_BEBIDA. Taula que relaciona el pub o bar amb els estils de beguda amb els quals es pot classificar.

**Taula Restau:** És la taula on s'emmagatzema les dades d'un restaurant.

Atributs:

- RESTAU\_ID: És generat automàticament i són incrementals.
- RESTAU\_LONGI: Coordenada longitud on s'ubica el restaurant.
- RESTAU\_LATIT: Coordenada latitud on s'ubica el restaurant.
- RESTAU\_NOMB: Nom del restaurant.
- RESTAU\_DIREC: Direcció on s'ubica el restaurant.
- RESTAU\_DESC: Descripció del restaurant.

Relacions:

- RESTAU\_EST\_COMIDA. Taula que relaciona el restaurant amb els estils de menjar amb els quals es pot classificar.

**Taula Est\_Comida:** És la taula on s'emmagatzema les dades d'un estil de menjar.

Atributs:

- EST\_COMIDA\_ID: És generat automàticament i són incrementals.
- EST\_COMIDA\_NOMB: Nom de l'estil de menjar.
- EST\_COMIDA\_DESC: Descripció de l'estil de menjar.

Relacions:

- Ja comentades.

**Taula Est\_Music:** És la taula on s'emmagatzema les dades d'un estil de música.

Atributs:

- EST\_MUSIC\_ID: És generat automàticament i són incrementals.
- EST\_MUSIC\_NOMB: Nom de l'estil de música.
- EST\_MUSIC\_DESC: Descripció de l'estil de música.

Relacions:

- Ja comentades.

**Taula Est\_Cine:** És la taula on s'emmagatzema les dades d'un estil de cinema.

Atributs:

- EST\_CINE\_ID: És generat automàticament i són incrementals.
- EST\_CINE\_NOMB: Nom de l'estil de cinema.
- EST\_CINE\_DESC: Descripció de l'estil de cinema.

Relacions:

- Ja comentades.

**Taula Est\_Activ:** És la taula on s'emmagatzema les dades d'un estil de cinema.

Atributs:

- EST\_ACTIV\_ID: És generat automàticament i són incrementals.
- EST\_ACTIV\_NOMB: Nom de l'estil de l'activitat.
- EST\_ACTIV\_DESC: Descripció de l'estil de l'activitat.

Relacions:

- Ja comentades.

**Taula Est\_Bebida:** És la taula on s'emmagatzema les dades d'un estil de beguda.

Atributs:

- EST\_ACTIV\_ID: És generat automàticament i són incrementals.
- EST\_ACTIV\_NOMB: Nom de l'estil de beguda.
- EST\_ACTIV\_DESC: Descripció de l'estil de beguda.

Relacions:

- Ja comentades.

**Taula Valo:** És la taula on s'emmagatzema les valoracions d'un lloc.

Atributs:

- VALO\_ID: És generat automàticament i són incrementals.
- VALO\_VALUE: Puntuació donada.

Relacions:

- CONCIERTO. Una valoració pot ser o no d'un concert i el concert tindrà varies o cap valoracions.

- ACT. Una valoració pot ser o no d'una activitat i l'activitat tindrà varies o cap valoracions.
- RESTAURANTE. Una valoració pot ser o no d'un restaurant i el restaurant tindrà varies o cap valoracions.
- PUB\_BAR. Una valoració pot ser o no d'un pub o bar i el pub o bar tindrà varies o cap valoracions.
- PELICULA. Una valoració pot ser o no d'una pel·lícula i la pel·lícula tindrà varies o cap valoracions.
- USER. Una valoració és d'un usuari i un usuari pot fer varies valoracions.
- VALO\_COMENT. Taula que relaciona la valoració amb els comentaris realitzats.

**Taula Coment:** És la taula on s'emmagatzema els comentaris.

Atributs:

- COMMENT\_ID: És generat automàticament i són incrementals.
- COMMENT\_DATE: Data quan s'ha realitzat el comentari.
- COMMENT\_ITSRESP: Camp que indica si el comentari és resposta a un altre. Valor 0: no és resposta, valor 1: si és resposta.
- COMMENT\_MNSJ: Text del comentari.

Relacions:

- COMENANT. Si el comentari és resposta d'un anterior. Relació que indica de quin comentari és resposta.
- ESCRITOR. Relació amb taula USUA que indica que un comentari és d'un usuari i un usuari pot fer varis comentaris.

**Taula sequence\_table:** És la taula on s'emmagatzema els últims índexs donats per realitzar l'assignació d'índexs automàtics.

Atributs:

- SEQUENCE\_NAME: Nom que identifica la taula de la qual és seqüència.
- SEQUENCE\_COUNT: Valor d'índexs donats.

Relacions:

No hi ha.

## 4. Tecnologies

En el mercat actual de la telefonia mòbil existeixen dos sistemes operatius que copen les primeres places en la classificació de sistemes operatius dins del sector. Aquests son iOS, el sistema operatiu propietari d'Apple per el seus dispositius mòbils com l'iPhone o l'iPad i Android, un sistema operatiu obert desenvolupat per Google i utilitzat per diferents fabricants de dispositius mòbils en els seus aparells, com podrien ser HTC, Samsung, Sony Ericsson, etc.

Per motius de seguretat els sistemes operatius per a dispositius mòbils no disposen de connectors per a bases de dades, ja que no seria segur que una aplicació client corrent en un dispositiu que es pot trobar a qualsevol lloc es connectés directament a una base de dades. Una de les maneres en que una aplicació per a un dispositiu mòbil pugui interactuar amb una base de dades remota és mitjançant els anomenats Serveis Web, que poden implementar-se en diferents llenguatges de programació i fent servir diferents eines de desenvolupament.

### 4.1. Android vs iOS

Tots dos sistemes operatius disposen d'una botiga virtual a través de la qual es distribueixen les aplicacions. En el cas de Android s'anomena Android Market i en el cas de iOS s'anomena AppStore. El número d'aplicacions disponibles a l'Android Market a finals de 2010 era de 130.000 aplicacions, havent multiplicat per sis la quantitat d'aplicacions en només un any. Per la seva banda AppStore encara disposa d'una gran avantatge sobre Android Market ja que va tancar l'any 2010 amb 300.000 aplicacions disponibles per a descarregar. L'aparició anterior en el temps de l'AppStore d'Apple justifica la diferència substancial en el número d'aplicacions disponibles.

Tots dos sistemes operatius disposen d'avantatges i d'inconvenients respecte l'altre. A tall d'exemple, s'enumeren les principals diferències entre ells:

- Android vs iOS:
  - o Avantatges:

- La disponibilitat al mercat de múltiples dispositius que el fan servir com a sistema operatiu fa que les possibilitats de creixement i expansió del mercat siguin molt altes. (Veure Gràfic 1)
- Un programador que vulgui començar a desenvolupar per Android no ha de pagar cap llicència per distribuir la seva aplicació des de l'Android Market.
- Les eines per desenvolupar amb Android son totalment gratuïtes i disponibles fàcilment a Internet.
- El Llenguatge de programació en que es programen les aplicacions per Android és Java, un llenguatge molt comú avui en dia.
- Inconvenients:
  - La gran quantitat de diferents models de telèfons mòbils que fan servir Android es alhora un inconvenient ja que cada dispositiu disposa de maquinari diferent i per tant el sistema operatiu ha de poder funcionar correctament amb qualsevol maquinari.
  - L'entorn de programació encara no està totalment acabat, com per exemple el desenvolupament de la part gràfica de les aplicacions, que s'ha de fer mitjançant arxiu XML.
- iOS vs Android:
- 
- Avantatges:
  - Ser el sistema operatiu dels dispositius mòbils de Apple és ja una avantatge, ja que el fabricant d'Estats Units disposa de molts compradors fidels a tot el mon. Tot i ser present només en dispositius Apple, la quota de mercat és molt semblant a la d'Android. (Veure figura 4.1).
  - Les eines de desenvolupament per iOS disposen d'un entorn totalment funcional, que permet al desenvolupador crear qualsevol control gràfic només arrastrant el component allà on el vol posar.

- Inconvenients:
  - Les eines per desenvolupar per iOS son de pagament malgrat que son distribuïdes per Apple amb el seu sistema operatiu Mac OS X alhora de comprar un dels seus ordinadors.
  - Per poder distribuir les aplicacions que ha creat un programador a altres dispositius mòbils d'Apple, bé passant l'aplicació a un altre telèfon o a través de la App Store s'ha de adquirir una llicència de desenvolupador de Apple que té un preu de 69€ a l'any.
  - El llenguatge de programació utilitzat per al desenvolupament de les aplicacions es Objective-C. Un llenguatge només utilitzat per Apple.

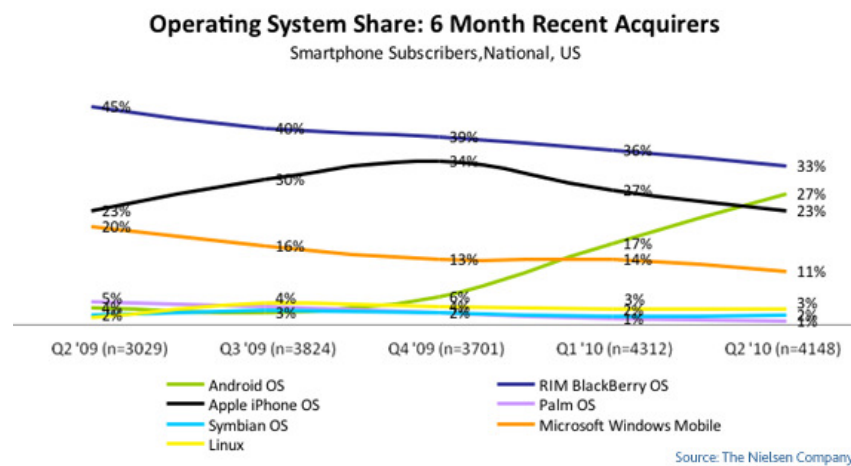


Figura 4. 1 Mercat sistemes operatius mòbils

## 4.2. Serveis Web.

Un servei web és un software que fa servir un conjunt de protocols i estàndards per intercanviar dades entre aplicacions. Diferent programari desenvolupat en llenguatges diferents i executat sobre qualsevol plataforma, pot utilitzar aquesta tecnologia per intercanviar dades entre màquines connectades mitjançant xarxes d'ordinadors com ara Internet fent servir el protocol HTTP .

Un Servei web s'encarregarà, per tant, d'interactuar amb la base de dades fent les peticions enviades per l'aplicació mòbil i retornant, en un format comprensible per aquesta, els resultats.

### 4.2.1. Web Services Description Language (WSDL).

Per descriure un servei web s'utilitza un llenguatge anomenat "Web Services Description Language" (WSDL). WSDL descriu la interfície pública dels serveis web. Està basat en XML i detalla la forma de comunicació, és a dir, els requeriments del protocol i els formats dels missatges necessaris per interactuar amb els serveis llistats en el seu catàleg. WSDL també descriu com s'ha d'utilitzar el servei web.

WSDL disposa del següents elements:

- <types>: defineix els tipus de les dades utilitzats en el missatge
- <message>: Es defineixen els elements del missatge.
- <portType>: Es defineixen les operacions permeses i els missatges intercanviats en el servei.
- <binding>: S'especifica el protocol de comunicació utilitzat.
- <service>: Conjunt de ports i direcció dels mateixos.

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-740-. -
3 <definitions targetNamespace="http://Service/" name="weduguWebSerService" xmlns="http://schemas.xmlsoap
  · www.w3.org/ns/ws-policy" xmlns:tns="http://Service/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:
  · 2004/09/policy" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsam="http://www.w3.org/2007/8
  · xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
4   <types>
5     <xsd:schema>
6       <xsd:import namespace="http://Service/" schemaLocation="WeduguWebSerService_schema1.xsd"/>
7     </xsd:schema>
8   </types>
9   <message name="saludo">
10    <part name="parameters" element="tns:saludo"/>
11  </message>
12  <message name="saludoResponse">
13    <part name="parameters" element="tns:saludoResponse"/>
14  </message>
15  <portType name="weduguWebSer">
16    <operation name="saludo">
17      <input wsam:Action="http://Service/weduguWebSer/saludoRequest" message="tns:saludo"/>
18      <output wsam:Action="http://Service/weduguWebSer/saludoResponse" message="tns:saludoResponse"/>
19    </operation>
20  </portType>
21  <binding name="weduguWebSerPortBinding" type="tns:weduguWebSer">
22    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
23    <operation name="saludo">
24      <soap:operation soapAction="saludo"/>
25      <input>
26        <soap:body use="literal"/>
27      </input>
28      <output>
29        <soap:body use="literal"/>
30      </output>
31    </operation>
32  </binding>
33  <service name="weduguWebSerService">
34    <port name="weduguWebSerPort" binding="tns:weduguWebSerPortBinding">
35      <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
36    </port>
37  </service>
38 </definitions>

```

Figura. 4.1. Arxiu WSDL descriptor de servei web



### 4.2.2. Simple Object Acces Protocol(SOAP)

SOAP es un protocol estàndard que defineix com dos objectes en diferents processos poden comunicar-se per mitjà de l'intercanvi de dades en format XML.

XML(Extensible Markup Language) és un metallenguatge extensible d'etiquetes desenvolupat per el World Wide Consortium. Dóna la possibilitat de definir la gramàtica de llenguatges específics.

L'estructura d'una petició SOAP es la següent:

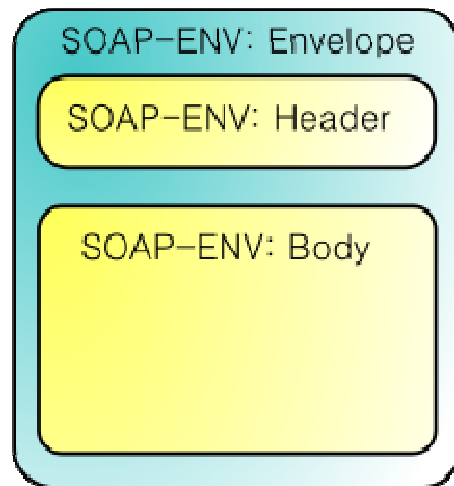


Figura. 4.4 Esquema petició SOAP

L'element arrel de qualsevol document SOAP es l'“Envelope” o sobre que conté tant el cos del missatge com qualsevol informació de capçalera que s'utilitzi per a descriure el missatge. Un missatge en SOAP només pot disposar d'un únic element “Envelope”

Dins de l'element arrel “Envelope” podem trobar dos elements:

- “Header”, és la capçalera del missatge i incorpora informació descriptiva respecte aquest. Només pot aparèixer una vegada dins del element “Envelope” i ha de ser fill directe d'aquest.
- “Body”, és el missatge pròpiament dit, la informació que es vol intercanviar.

### 4.3. Llibreria ksoap-2android

Per poder consumir el servei web des d'una aplicació mòbil es fa necessària una llibreria que proveeixi a l'aplicació dels objectes necessaris per la comunicació entre ambdós. En el cas del sistema operatiu Android aquesta és ksoap-2android.

Els principals elements d'aquesta llibreria amb els que s'ha de treballar per poder intercanviar dades en format SOAP son:

- SoapObject: Objecte Soap on s'inclourà el sobre que transportarà el missatge que es vol intercanviar.
- PropertyInfo: Objecte amb el qual s'inclouen el cos de missatge.
- SoapSerializationEnvelope: Objecte per modelar el sobre del missatge.
- HttpTransportSE: Objecte per realitzar la crida HTTP al servei web.
- SoapPrimitive: Objecte per recollir el resultat de la crida.

### 4.4. Elecció.

Després de valorar les diferents opcions pel desenvolupament d'una aplicació mòbil s'ha triat l'opció del sistema operatiu Android. Les raons per les que s'ha triat aquesta opció per davant de l'opció de iOS son les següents:

- Coneixement del llenguatge en que es desenvolupen les aplicacions. D'aquesta manera a més d'haver d'aprendre l'estructura del sistema operatiu, el conjunt de classes disponibles per al desenvolupament d'una aplicació i com interactuar amb el maquinari del dispositiu mòbil no s'haurà d'aprendre un llenguatge de programació totalment nou.
- Gratuïtat del servei de distribució de l'aplicació. Un factor clau en l'elecció del sistema operatiu Android ha estat el no haver de pagar cap llicència per desenvolupar l'aplicació i la posterior distribució d'aquesta.
- Fort increment de la quota de mercat de Android. En els últims temps el sistema operatiu Android ha augmentat la seva quota de mercat respecte els seus competidors i continua en creixement.

- Gratuïtat de les eines per al desenvolupament de l'aplicació. Les eines per a desenvolupar l'aplicació son totalment gratuïtes. Com per exemple l'entorn de desenvolupament Eclipse Helios.
- Facilitat d'ús de la llibreria ksoap-2Android. La facilitat d'ús d'aquesta llibreria i amb això la facilitat per l'intercanvi de dades entre el servidor de base de dades i l'aplicació respecte la mateixa problemàtica en iOS ha sigut un factor clau en la decisió.

Per al desenvolupament del servei web que interactuarà amb la base de dades remota s'ha escollit el llenguatge de programació Java i la eina de desenvolupament NetBeans, que disposa d'una eina eficient i de gran usabilitat per a la creació de serveis web de manera gràfica, incloent aquesta, les anotacions necessàries sobre el codi Java per a la construcció del servei web.

Per a poder consumir la informació servida per al servei web en format SOAP des de el sistema operatiu Android cal una llibreria anomenada ksoap-2Android. Aquesta llibreria proporciona un conjunt de classes mitjançant les quals es pot modelar un "Envelope" o sobre, incloure una capçalera de missatge i el missatge que es vol transportar. Alhora també proporciona un conjunt de classes per poder tractar la resposta rebuda per part del servei web també en format SOAP.



## **5. Android**

### **5.1. Que és Android.**

Android es una plataforma de programari i un sistema operatiu per a dispositius mòbils basada en el nucli de Linux, desenvolupada per Google i posteriorment per la Open Handset Alliance. Permet als desenvolupadors escriure codi en el llenguatge de programació Java que s'executarà en dispositius mòbils. Existeix la possibilitat de desenvolupar aplicacions en altres llenguatges malgrat que no estan reconeguts per Google com a oficials. Android es distribueix sobre una llicència de maquinari Apache, tractant-se, per tant, d'una llicència per a software lliure.

### **5.2. Historia.**

A l'any 2005 el gegant de la tecnologia Google va comprar Android Inc., una petita empresa de Palo Alto, Califòrnia, que era desconeguda per al món de les tecnologies si bé el seu grup de fundadors, entre els quals es trobaven Andy Rubin (co-fundador de Danger), Rich Miner (co fundador de WildFire Communications Inc.), Nick Sears (T-Mobile) i Chris White, tenia una gran experiència en plataformes web, telecomunicacions i aplicacions per a terminals mòbils. En aquells moments l'empresa es dedicava, bàsicament, a la creació de maquinari per a telèfons mòbils. Una vegada incorporats a Google, els tècnics de Android Inc van desenvolupar una plataforma per a dispositius mòbils basada en el nucli de Linux que es va promocionar entre els fabricants de terminals com un sistema flexible i actualitzable.

Els rumors sobre el llançament del sistema operatiu Android de Google i la seva intenció d'entrar a formar part del mercat de la telefonia mòbil es va incrementar a partir de 2006, quan els mitjans de comunicació van difondre notícies sobre la intenció de Google de portar el seu servei de cerca i aplicacions a dispositius mòbils.

El 5 de novembre de 2007 es va crear, amb la finalitat de desenvolupar estàndards oberts per a dispositius mòbils, la Open Handset Alliance, un conglomerat de varies companyies

entre les que es trobaven Broadcom Comporation, Nvidia, Qualcomm, Samsung Electronics, Intel, LG, Motorola i T-Mobile. i el seu primer producte va estar Android, una plataforma per a terminals mòbils construïda sobre la versió 2.6 del nucli de Linux.

El 23 de setembre de 2008, es va llançar a la venda en el mercat d'Estats Units, la primera versió d'un telèfon mòbil amb Android, en concret, el T-Mobile G1, conegut també com HTC Dream i fabricat per HTC.



Figura 5. 1 HTC Dream

### 5.3. Característiques

El Sistema Operatiu Android inclou un kernel de Linux i consta d'aproximadament 12 milions de línies de codi (3 millions en XML, 2,8 millions en C, 2,1 millions en Java i 1,75 millions en C++). A continuació s'indiquen les seves característiques més importants:

<b>Recursos gràfics</b>	El sistema és adaptable a gran quantitat de pantalles VGA, disposa de llibreries per a gràfics 2D, gràfics 3D basades en OpenGL ES 2.0 i és adaptable a les pantalles tradicionals dels smartphones
<b>Emmagatzemament</b>	Disposa d'un lleuger motor de base de dades SQLite
<b>Connectivitat</b>	Android suporta múltiples tecnologies de connectivitat GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC i WiMAX.

<b>Missatgeria</b>	SMS i MMS estan disponibles com a formes de missatgeria
<b>Suport multi llenguatge</b>	Android es troba disponible en gran varietat de llenguatges. El nombre de llenguatges s'ha multiplicat per dos a la versió 2.3
<b>Navegador web</b>	El navegador web disponible a Android està basat en el motor de codi lliure WebKit.
<b>Suport Java</b>	Encara que la majoria d'aplicacions estan desenvolupades en Java, no hi ha cap màquina virtual a la plataforma i el bytecode de Java no s'executa. Les classes Java es compilen en fitxers executables per Dalvik i s'executen en la màquina virtual Dalvik. Dalvik és una màquina virtual específicament dissenyada per Android i optimitzada per a dispositius mòbils alimentats amb bateria, amb poca memòria i CPU.
<b>Suport multimèdia</b>	Android incorpora suport per als següents formats d'àudio i vídeo: WebM, H.263, H.264, MPEG-4, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, BMP.
<b>Suport per streaming</b>	Els protocols per streaming suportats són: RTP/RTSP, HTML de descarrega progressiva i HTTP Dynamic Streaming.
<b>Suport per maquinari addicional</b>	Android pot utilitzar càmeres de vídeo, pantalles tàctils, GPS, acceleròmetre, giroscopis, magnetòmetres, etc.
<b>Multi-touch</b>	Android inclou suport natiu per a multi-touch, esta disponible per primera vegada des de l'aparició de dispositius com la HTC Hero.
<b>Bluetooth</b>	Suporta A2DP, AVRCP, enviament de fitxers, accés a agendes, diàleg per veu i enviament de contactes entre diferents terminals.
<b>Videotrucades</b>	La majoria de versions d'Android no suporta les videotrucades, però algunes versions personalitzades d'aquest sí que donen suport a través de la xarxa UMTS, com pot ser la que incorpora el Samsung Galaxy S.
<b>Multitasca</b>	Té disponible la multitasca entre aplicacions.
<b>Característiques</b>	Les cerques de veu mitjançant Google Voice estan disponibles des de la

<b>basades en veu</b>	primera versió. Les accions de veu per trucar, escriure i navegar estan disponibles des de la versió 2.2
<b>Tethering</b>	Android suporta el Tethering, el qual fa possible utilitzar el telèfon com a punt d'accés Wi-Fi.

Taula 5. 1 Característiques Android

## 5.4. Arquitectura

### 5.4.1. Representació de l'arquitectura

Els principals components d'aquesta arquitectura son:

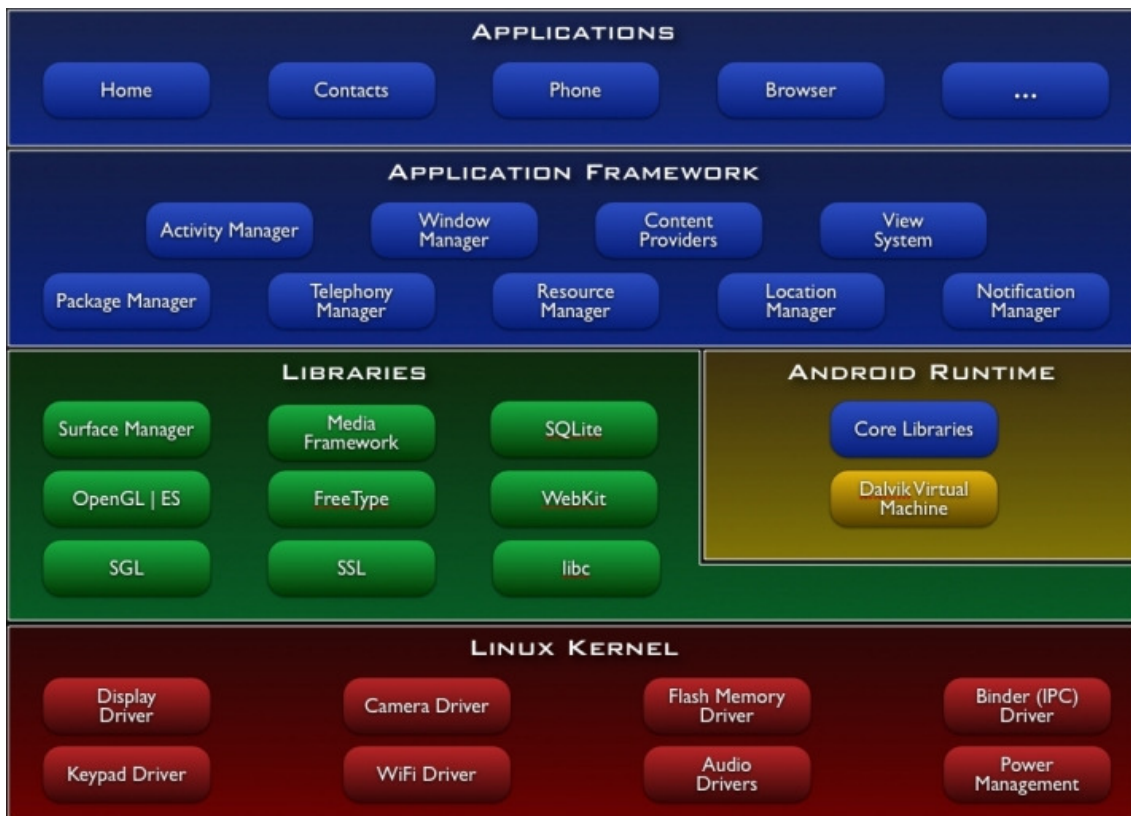


Figura 5. 2 Arquitectura Android

### 5.4.2. Aplicacions.

Les aplicacions base inclouran un client d'e-mail, un programa per SMS, calendari, mapes, navegador, contactes, etc. Totes les aplicacions estan desenvolupades amb el llenguatge de programació Java.



### 5.4.3. Marc de treball d'aplicacions.

Els desenvolupadors tenen accés al complet a les APIs del marc de treball que fan servir les aplicacions base. L'arquitectura està dissenyada per simplificar la reutilització dels components, qualsevol aplicació pot publicar les seves funcionalitats i qualsevol altre aplicació pot fer ús d'aquestes funcionalitats tenint en compte les regles de seguretat del framework.

Aquest marc de treball està format per:

- View System: Un conjunt de vistes, com poden ser, llistes, caixes de text, botons, desplegable, menús, etc.
- Content Providers: Son elements que permeten a les aplicacions accedir a informació d'altres aplicacions o compartir la seva informació amb altres.
- Resource manager: Proporciona accés a recursos que no son codi, com poden ser gràfics, cadenes de text, etc.
- Notification manager: Permet a les aplicacions mostrar alarmes personalitzades a la barra d'estat del sistema operatiu.
- Activity manager: Gestiona el cicle de vida de les aplicacions.
- Telephony Manager: Gestiona tot el referent a telefonia, com poden ser, trucades.
- Window manager: Gestiona les finestres que es troben obertes al sistema.
- Location Manager: Gestiona la geolocalització del dispositiu.

### 5.4.4. Llibreries.

Android inclou un conjunt de llibreria C/C++ usades per diferents components del sistema. Aquestes característiques son accessibles per els desenvolupadors mitjançant el marc de treball d'aplicacions d'Android. Exemples:

- System C library: Implementació biblioteca estàndard C.
- Eines de comunicació de llibreries lde basades en OpenCore
- PacketVideo, per a la reproducció i gravació de formats d'àudio i vídeo, així com arxius d'imatge, MPEG4,H.264,MP·,AAC,AMR,JPG,PNG, OGG VORBIS i OGG THEORA
- LibWebCore: un modern navegador web.
- SGL: Motor gràfic 2D
- Llibreries 3D basades en OPENGL ES1.0 API
- FreeType: Mapa de bits i vectors de la renderització de les fonts
- SQLite: Lleuger i potent motor de base de dades relacional disponible per a totes les aplicacions.

#### **5.4.5. Android Runtime.**

Android inclou un conjunt de llibreries base que proporcionen gran part de les funcionalitats disponibles en les biblioteques base del llenguatge Java. Cada aplicació Android corre el seu propi procés, amb la seva instància a la màquina virtual Dalvik. Dalvik ha estat desenvolupat per permetre que un dispositiu pugui executar diferents màquines virtuals a la vegada, de manera eficient. Dalvik executa arxius en format Dalvik Executable (.dex), el qual està optimitzat per ocupar mínima memòria. La màquina virtual està basada en registres i executa classes compilades en bytecode pel compilador de Java que prèviament han estat transformades al format .dex per l'eina "dx" inclosa.

#### **5.4.6. Nucli de Linux.**

Android fa servir Linux per proveir els serveis base del sistema com seguretat, gestió de memòria, gestió de processos, pila de xarxa i model de controladors. El nucli també actua com una capa d'abstracció entre el maquinari i la resta d'aplicacions.

## 5.5. Estructura d'una aplicació.

Dins d'una aplicació Android existeixen quatre components principals: Activity, Listener, Servei i Content Provider. Qualsevol aplicació Android està composta per algun d'aquests elements o per un conjunt d'ells.

### 5.5.1. Activity.

Les Activitats són els elements més comuns en una aplicació. S'utilitza una classe per a cada Activitat. Cal tenir present, a l'hora d'implementar-la, que aquesta classe ha d'estendre la classe base *Activity*. Cada classe mostra a l'usuari una interfície gràfica composta per diferents vistes. Cada vegada que es canvia d'interfície gràfica es canvia d'activitat. Al passar d'una activitat a una altra, la primera queda pausada i el seu estat queda enregistrat en una pila per poder tornar enrere en qualsevol moment. Per canviar d'una activitat a una altra, Android utilitza una classe anomenada Intent.

### 5.5.2. Intent.

Un Intent és un objecte que descriu quina acció realitzarà l'activitat corresponent. Les dues parts més importants d'un Intent són: l'acció que es vol realitzar i la informació necessària perquè aquesta acció es dugui a terme.

Relacionada amb la classe Intent existeix un altre anomenada IntentFilter. Aquesta és una descripció de que Intents pot gestionar una determinada Activitat.

### 5.5.3. Listener.

Els Listeners són components que serveixen per que el sistema reaccioni a esdeveniments externs tals com, prémer un botó, avis de bateria baixa, canvi de posició en el GPS, etc. Per llançar un avís no fa falta que l'aplicació s'estigui executant en aquell moment ja que Android l'iniciarà si s'activa el Listener corresponent per un esdeveniment determinat.

### 5.5.4. Serveis.

Un servei és un fragment de codi que s'executa durant un període llarg de temps i que no té necessitat de disposar d'interfície gràfica. En el cas que existeixin diferents serveis executant-se alhora es pot fixar la prioritat de cadascun d'ells.

### 5.5.5. Content Provider.

A Android les aplicacions poden guardar informació en diferents destinacions, com poden ser: fitxers de text, base de dades SQLite, etc. Per que una aplicació pugui compartir certa informació amb altres aplicacions Android fa servir la classe Content Provider. Content Provider implementa un conjunt de mètodes estàndards que fan possible a una aplicació emmagatzemar o llegir informació d'aquest.

### 5.5.6. Android Manifest.

Android Manifest es un arxiu en format XML que és imprescindible pel funcionament de qualsevol aplicació. En aquest fitxer es descriu el comportament de l'aplicació, els recursos als que té accés, els permisos dels que disposa i el llistat d'activitats que la componen. Si una activitat no està descrita a l'Android Manifest l'aplicació no la podrà executar.

Exemple d' Android Manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
  <application android:icon="@drawable/app_icon.png" ... >
    <activity android:name="com.example.project.ExampleActivity"
      android:label="@string/example_label" ... >
      </activity>
    ...
  </application>
</manifest>
```

Figura 5. 3 Android Manifest

## 5.6. Tipus d'aplicacions.

### 5.6.1. Tasques.

Una tasca és allò que l'usuari veu com una activitat i el desenvolupador veu com una successió d'activitats. Cada tasca té una activitat que es considera el punt d'entrada i aquesta ha d'estar definida a l'Android Manifest com a tal.

### **5.6.2. Processos.**

Els processos s'executen a nivell de nucli i l'usuari no és conscient de que s'estan executant. Les principals funcions dels processos son:

- Millorar l'estabilitat de les aplicacions.
- Reduir la sobrecarrega d'execucions amb la mateixa funcionalitat en diferents aplicacions.
- Ajudar al sistema a gestionar els recursos, separant l'aplicació en diferents parts que poden ser terminades independentment.

### **5.6.2. Threads.**

Android evita la creació de diferents fils d'execució d'una mateixa aplicació, sempre i quan aquest fil no hagi estat creat explícitament a l'aplicació. Per aquesta raó, com que cada aplicació només s'executa sobre un únic fil, cal evitar que l'aplicació realitzi operacions massa llargues en el temps, ja que si no pot arribar a bloquejar la resta de components del procés.

## **5.7. Cicle de vida d'una aplicació.**

La següent imatge il·lustra el cicle de vida d'una aplicació:

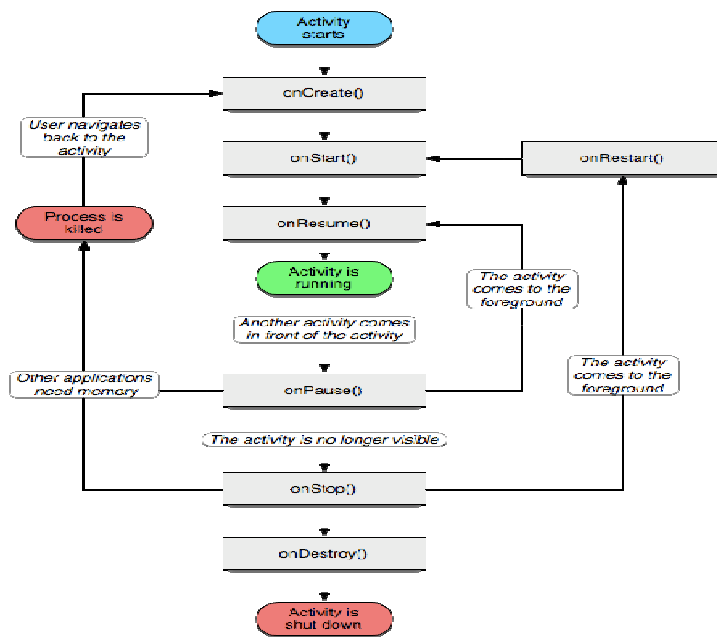


Figura 5. 4 Cicle de vida d'una aplicació

Android manté les aplicacions tant de temps com li és possible. A vegades el sistema necessita terminar els processos més vells per recuperar memòria per a processos més nous o més importants. Per determinar quin procés s'ha de mantenir i quin procés pot ser finalitzat, el sistema col·loca cada procés dintre d'un esquema de jerarquia per importància, basat en els components que s'estan executant i en el seu estat. Els processos amb menor importància són eliminats primer, després el que tenen més importància que els primers i així successivament fins alliberar els recursos del sistema.

Existeixen cinc nivells d'importància. A continuació es detallen els diferents tipus de processos en ordre d'importància essent el primer el més important i, per tant, l'últim en ser finalitzat:

- Procés en primer terme.

És un procés que és sol·licitat pel que l'usuari està realitzant en aquest moment. Un procés és considerat com de primer terme si es compleix una de les següents condicions:

- Conté una activitat amb la qual l'usuari està interactuant (es crida el mètode `onResume()`).
- Conté un servei que és necessari per l'activitat amb la qual l'usuari està interactuant.

- Conté un servei que executa en primer terme.
- Conté un servei que s'està executant en un dels retorns del seu cicle de vida.

Generalment només existeix un nombre reduït de processos en primer terme.

- Procés visible.

Aquest procés no té cap component en primer terme però es veu afectat per allò que l'usuari veu per pantalla. Un procés es considerat com visible si es compleix una de les següents condicions:

- Conté una activitat que no es troba en primer terme però que segueix sent visible per l'usuari (ha estat cridat el mètode `onPause()`).
- Conté un servei lligat a una activitat visible o en primer terme.

- Procés de servei

Un procés que executa un servei que ha estat cridat mitjançant l'instrucció `StartService()` i no pertany a cap de les dues categories anteriors.

- Procés en segon terme

Un procés que manté una activitat que no és visible per a l'usuari. Aquests processos no tenen un impacte directe sobre l'usuari i el sistema els pot finalitzar en qualsevol moment per alliberar recursos. Generalment hi ha molts processos que s'executen en segon terme i es mantenen en una llista LRU (menys recentment utilitzat) per assegurar que el procés que ha estat vist en últim lloc per l'usuari és l'últim en ser finalitzat.

- Procés buit.

Un procés que no té cap component de l'aplicació activa. L'única raó per mantenir aquest tipus de procés, és per emmagatzemament a la cache.

El rang d'un procés sempre es troba al més alt nivell possible, en base a la importància del components actius actualment en el procés. Per exemple, si un procés executa un servei i una activitat visible, és classificat com un procés visible i no com un procés de servei.

## 5.8. Estructura d'un projecte Android.

Al generar un nou projecte Android en un entorn de desenvolupament, es genera una estructura de carpetes necessàries per poder generar posteriorment l'aplicació. Aquesta

estructura serà comuna a qualsevol aplicació amb independència del seu volum i complexitat.

En la imatge següent podem veure l'estructura bàsica de directoris:

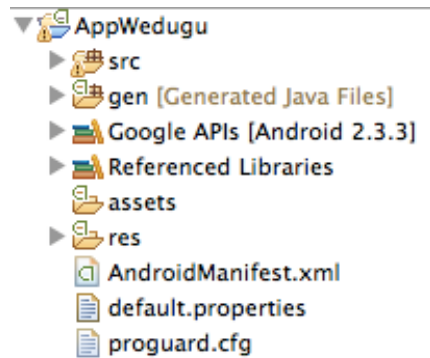


Figura 5. 5 Estructura projecte

### 5.8.1. Carpeta /src.

Conté tot el codi font de l'aplicació, codi de la interfície gràfica, classes auxiliars, etc. Inicialment l'eina de desenvolupament crearà per defecte la classe Java per l'activitat principal de l'aplicació sempre sota el paquet per defecte d'aquesta.

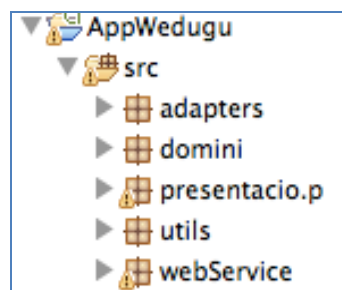


Figura 5. 6 Carpeta src

### 5.8.2. Carpeta /res.

Conté tot els fitxers de recursos necessaris per al projecte, com poden ser: imatges, vídeos, cadenes de text, etc. Els diferents tipus de recursos s'hauran de distribuir dins de les següents carpetes:

- /res/drawable: Conté les imatges de l'aplicació. Es pot dividir en: /drawable-ldpi, /drawable-mdpi i /drawable-hdpi per utilitzar diferents recursos depenent de la resolució del dispositiu.



- /res/layout: Conté els diferents fitxers de definició de les diferents pantalles de la interfície gràfica. Es pot dividir en: /loyout i /layout-land per utilitzar diferents recursos en funció de l'orientació del dispositiu.
- /res/anim: Conté la definició de les animacions utilitzades per l'aplicació.
- /res/menu: Conté la definició dels menús utilitzats en l'aplicació.
- /res/values: Conté altres recursos de l'aplicació com poden ser cadenes de text, estils o colors.
- /res/xml: Conté els fitxers XML utilitzats per l'aplicació.
- /res/raw: Conté recursos addicionals normalment en un format diferent al XML que no es troben en cap carpeta de les mencionades anteriorment.

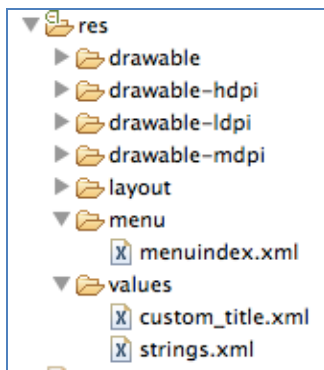


Figura 5. 7 Carpeta res

### 5.8.3. Carpeta /gen.

Conté una sèrie d'elements de codi generats automàticament alhora de compilar el projecte. El més important d'aquest es el fitxer R.java i la classe R.

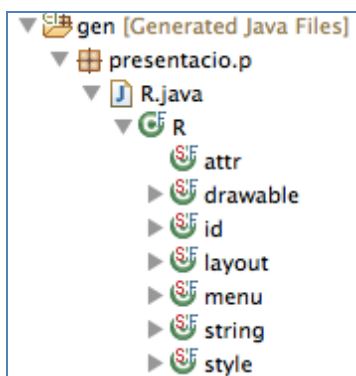


Figura 5. 8 Carpeta gen

La classe R contindrà en tot moment una sèrie de constants amb els identificadors de tots els recursos de l'aplicació inclosos a la carpeta /res/, de manera que puguem accedir fàcilment a aquests recursos des del codi de la nostra aplicació mitjançant aquesta dada.

#### 5.8. 4. Carpeta /assets.

Conté la resta de fitxers auxiliars necessaris per a l'aplicació, com, per exemple, fitxers de configuració, de dades, etc.

La diferència entre els recursos inclosos a la carpeta /res/raw i els inclosos en aquesta carpeta és que per als primers es generarà una identificació a la classe R i s'haurà d'accedir als recursos mitjançant aquesta identificació mentre que per els recursos de la carpeta /assets no es generarà cap identificació i s'accedirà a ells per la seva ruta, com qualsevol altre fitxer del sistema.

### 5.9. Layouts, Widtgets i altres vistes.

#### 5.9.1. Layouts

##### Linear Layout.

Linear Layout és una visió en grup que permet tenir elements Vista que siguin fills d'aquest en una direcció lineal, tant en vertical com en horitzontal.

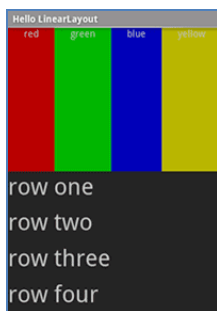


Figura 5. 9 Linear layout

##### Relative Layout.

Relative Layout és un grup de vistes que mostren per pantalla vistes d'elements en posicions relatives. La posició d'una vista pot ser especificada en relació a elements germans o en posicions relatives al àrea del Relative Layout. Un Relative Layout és una

utilitat molt poderosa per dissenyar una interfície gràfica per què pot eliminar la necessitat de més grups de vistes.

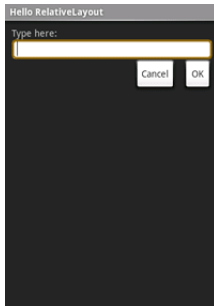


Figura 5. 10 Relative layout

### **Table Layout.**

Table Layout és un grup de vistes que mostren per pantalla el seus elements fills en forma de files i columnes.

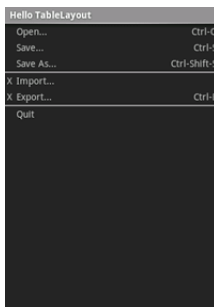


Figura 5. 11 Table layout

### **Grid View.**

Grid View mostra elements en una graella de dues dimensions en la que l'usuari es pot desplaçar cap avall i cap a dalt. Els elements de la graella son inserits automàticament, utilitzant un Adaptador.

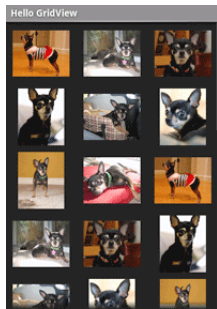


Figura 5. 12 Grid View

### Tab Layout.

Aquesta vista servirà per crear un sistema de pestanyes en les quals variarà el contingut mostrat per pantalla en funció de quina estigui seleccionada.

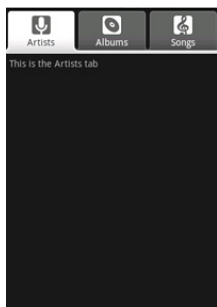


Figura 5. 13 Tab layout

### List view

List View és un grup de vistes que crea una llista d'elements sobre la qual l'usuari es pot desplaçar cap avall o cap a dalt. Els elements de la llista son carregats de manera automàtica mitjançant un adaptador.



Figura 5. 14 List View

## 5.9.2. Widgets i altres vistes.

Un Widget és una petita aplicació o programa que realitza una funcionalitat específica.

### Date picker.

Widget que permet a l'usuari seleccionar una data. Permet a l'usuari seleccionar, mes, dia i any d'una manera familiar.



Figura 5. 15 Date picker

### Time picker.

Widget que permet a l'usuari seleccionar una hora de manera còmode. Permet seleccionar hores i minuts.

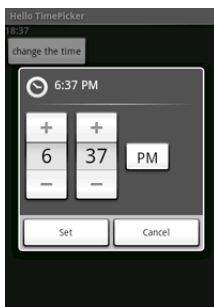


Figura 5. 16 Time picker

### Components de formulari.

Aquest és un grup de vistes que engloba totes les vistes que son necessàries per a la creació d'un formulari, com poden ser botons, camps de text, radio buttons, etc.

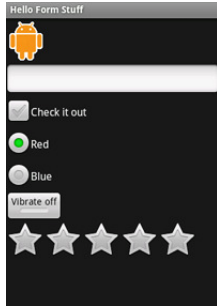


Figura 5. 17 Form stuff

## Spinner

Spinner és un widget que permet seleccionar una de les diverses opcions mostrades. Es comporta de la mateixa manera que un “select” a HTML o un “combo box” a Java.

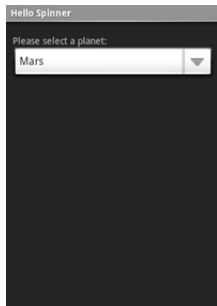


Figura 5. 18 Spinner

## Google Map View.

Utilitzant les llibreria de Google Map es pot crear un mapa personalitzat. Es poden fixar paràmetres com el zoom, sobre quin punt esta centrat el mapa, es poden localitzar i marca punts geogràfics i fins i tot es poden superposar capes com per exemple, el tràfic.

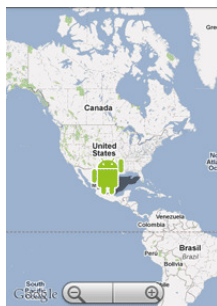


Figura 5. 19 Map view

## Web View.

La vista web permet crear una finestra per visualitzar pàgines web dintre de la mateixa aplicació.

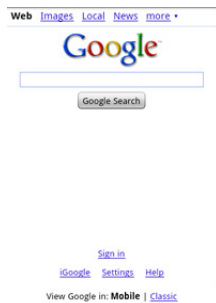


Figura 5. 20 Web View

## 5.10. Com començar a desenvolupar en Android.

### 5.10.1. Entorn de desenvolupament.

Per poder començar a desenvolupar aplicacions per Android és necessari la instal·lació d'un entorn de desenvolupament. Existeixen dos entorns distribuïts amb llicència lliure que donen la possibilitat de desenvolupar aplicacions per Android. Aquest son: Netbeans i Eclipse.

S'ha escollit Eclipse perquè el plug-in per Android està molt més desenvolupat que el de Netbeans. És recomanable instal·lar la versió Eclipse IDE for Java Developers.

Un cop instal·lat Eclipse, per poder desenvolupar aplicacions per Android s'ha d'instal·lar el plug-in necessari. Per fer-ho s'ha d'accedir a una nou font de software (<https://foss.google.com/android/eclipse>) i seguir el passos per la seva instal·lació.

### 5.10.2. Android SDK.

Una vegada instal·lat l'entorn de desenvolupament, és necessari la instal·lació de Android SDK. Una vegada descarregat i instal·lat l'Android SDK s'ha d'obrir el AVD Manager i descarregar tots els "targets" disponibles i tots els plug-in's de Google per poder desenvolupar aplicacions per a qualsevol versió de la plataforma.

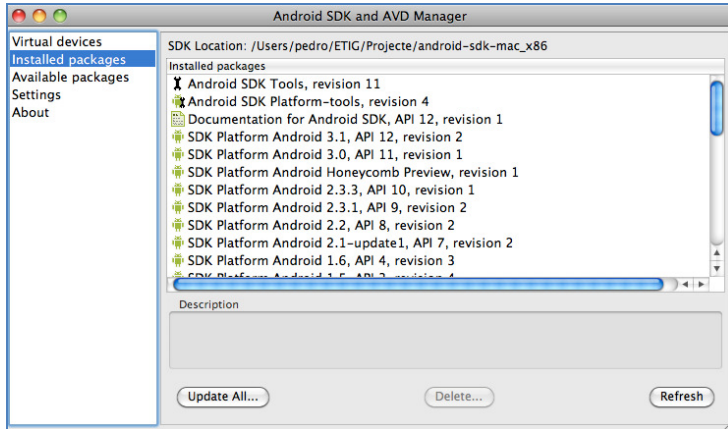


Figura 5. 21 AVD Manager

El següent pas es crear un emulador per poder provar les aplicacions. Es fa des de l'opció "Virtual devices" del AVD Manager i cal triar un nom, la versió de la plataforma, la resolució de pantalla i els recursos que es podran fer servir com, per exemple, el GPS.

Com es pot veure a la figura adjunta l'emulador generat consta d'una pantalla tàctil, un teclat qwerty i uns botons i permet fer qualsevol funcionalitat que pugui realitzar un dispositiu mòbil amb Android. Per verificar el funcionament d'una aplicació només cal executar-la al entorn de desenvolupament i automàticament s'instal·larà i executarà al emulador.

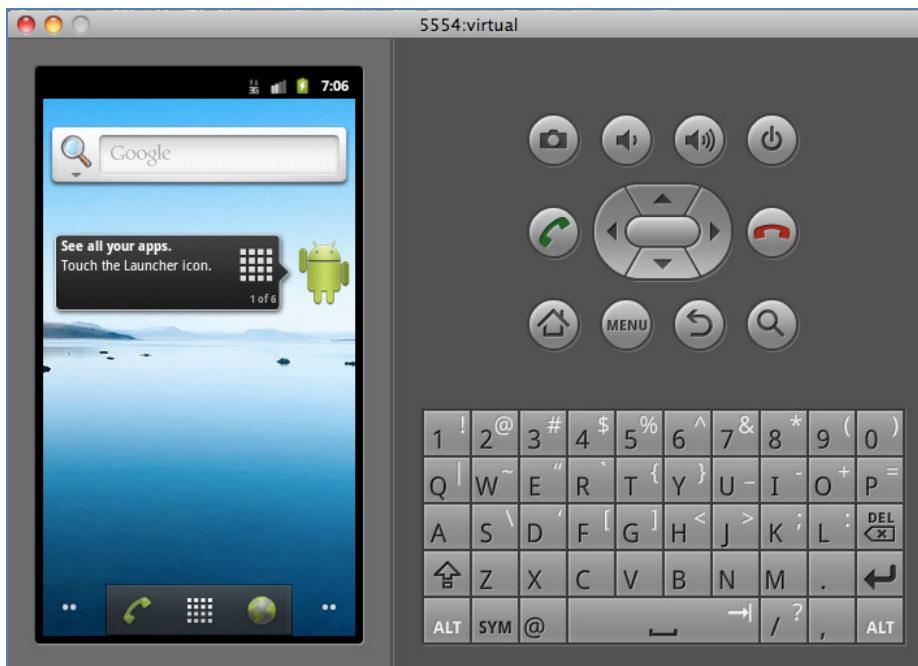


Figura 5. 22 Emulador





Figura 5. 23 Emulador executant WeDugu



## 6. Codificació.

### 6.1. Android.

Com ja s'ha vist anteriorment en un projecte Android cada cas d'ús es considera una tasca i es pot veure com un conjunt de diferents activitats. Cada activitat disposa de la seva classe Java on es descriu el seu comportament i la seva resposta als esdeveniments llançats per l'usuari i, també, d' un arxiu XML on es descriu l'aspecte d'aquella activitat, els components dels que disposa i els seus identificadors..

Alhora de començar a programar una classe Java per controlar una activitat s'ha de fer que aquesta hereti de la classe base Activity o d'una classes que, a la seva vegada, hereti d'Activity.

Els mètodes que ha de tenir aquesta classe per que la activitat funcioni son els següents:

- `public void onCreate(Bundle savedInstanceState)`: Aquest mètode es imprescindible ja que serà el mètode invocat quan es cridi a l'activitat. De fet, és el constructor de l'activitat. Dins d'aquest mètode es crida al constructor de la superclasse i s'estableix l'arxiu XML de pantalla que li correspon.. Una vegada fet l'anterior es codifica dins del mètode el comportament de la classe respecte a esdeveniments de botons, etc.
- `onPause()`, `onResume()`: Opcionalment i depenent del comportament que ha de tenir la nostra activitat, es pot implementar codi dins d'aquest mètode que serà executat quan el cicle de vida de l'activitat indiqui que ha d'evolucionar cap els estats respectius.

#### 6.1.1. Consum del servei web.

Com ja s'ha explicat anteriorment l'accés a la informació de la base de dades es farà mitjançant un servei web. Per poder consumir el servei web des de l'aplicació Android s'ha creat una classe anomenada `ConsumirWS.java`, dins del paquet `webService`. Aquesta classe disposa de diferents mètodes que, basats en els diferents mètodes del servei web, retornen la informació demandada per l'aplicació.

Els mètodes del que disposa aquesta classe son:

- `llamarMetodogetUser(String email,String password)`: Cridarà al mètode del servei web que retornarà l'identificador de l'usuari en funció del seu e-mail i contrasenya.
- `llamarMetodogetAmics(int id)`: Cridarà al mètode del servei web que retornarà la llista d'amics de l'usuari en curs.
- `llamarMetodoValidarEmail(String email)`: Cridarà al mètode del servei web que retorna si el e-mail és registrat a la base de dades.
- `llamarMetodoGetPerfilUser(int id)`: Cridarà al mètode del servei web que retorna la informació de l'usuari en curs.
- `llamarMetodoUpdatePerfil(String x...)`: Cridarà al mètode del servei web encarregat d'actualitzar la informació del perfil de l'usuari, passant-li tota la informació necessària.
- `llamarMetodogetPropuestaRestau(Double lat, Double lon, int id)`: Cridarà al mètode del servei web que retornarà propostes de restaurants.
- `llamarMetodogetPropuestaActi(Double lat, Double lon, int id)`: Cridarà al mètode del servei web que retornarà propostes d'activitats.
- `llamarMetodogetPropuestaConcie(Double lat, Double lon, int id)`: Cridarà al mètode del servei web que retornarà propostes de concerts.
- `llamarMetodogetPropuestaPubBar(Double lat, Double lon, int id)`: Cridarà al mètode del servei web que retornarà propostes de Pub/Bar.
- `llamarMetodogetPropuestaCine(Double lat, Double lon, int id)`: Cridarà al mètode del servei web que retornarà propostes de pel·lícules.
- `llamarMetodoInsertAgendarEvento(id...)`: Cridarà al mètode del servei web encarregat d'inserir l'esdeveniment a l'agenda de l'usuari, passant-li tota la informació necessària.
- `llamarMetodoGetEventosAgenda(int id, int day, int month, int year)`: Cridarà al mètode del servei web que retornarà els esdeveniments que, en un dia determinat, un usuari té a l'agenda.
- `llamarMetodoBuscar(int id, String t, String tipus)`: Cridarà a un dels dos mètodes del servei web en funció del tipus especificat com a paràmetre. Pot cridar al mètode que retorna la informació cercada entre els amics de l'usuari o al mètode que retorna la informació de la cerca entre tots els usuaris de WeDugu.

### 6.1.1. Índex.

Tota aplicació té una activitat que es considera el seu punt d'entrada. En el cas de WeDugu és l'activitat V\_Indice, implementada per la classe V\_Indice.java i el fitxer XML indice.xml. Aquesta activitat mostra per pantalla el logotip de l'aplicació i és capaç de mostrar un menú al prémer sobre el botó menú del dispositiu, amb les opcions d'entrar o sortir de l'aplicació.

Si es clica sobre l'opció "Log in", mitjançant un Intent (Element bàsic d'Android explicat en el punt 5.5.2) és llançada la següent activitat V\_LogIn. Si es clica sobre l'opció "Salir" es finalitza l'activitat cridant al mètode heretat this.finish() i com és l'activitat de punt d'entrada es finalitza tota l'aplicació.

### 6.1.2. Cas d'ús Log in.

El cas d'ús de Log in només consta d'una activitat que, a la seva vegada, consta d'una única classe Java, V\_LogIn.java, que conté el codi del seu comportament i es troba dins del paquet presentació, i d'un arxiu XML, login.xml, que descriu la pantalla mostrada a l'usuari i es troba a la carpeta /res/layout.

La classe V\_LogIn.java, en primer lloc, mostra la pantalla que li correspon i a continuació registra l'esdeveniment associat al botó d'entrar a la aplicació. Un cop enregistrat l'esdeveniment cal que la classe implementi una interfície anomenada OnClickListener que obligarà a la inclusió d'un mètode onClick(View v) per el control del esdeveniment.

Quan l'usuari prem el botó es comprova que les caixes de text associades al nom d'usuari i a la contrasenya no siguin buides i, a continuació, es crida el mètode llamarMetodoGetUser(string,string) de la classe ConsumirWS.java que retornarà l'identificador de l'usuari sempre i quan aquest existeixi. Si l'usuari no existeix a la base de dades es llançarà un missatge i en cas contrari l'activitat cridarà a la següent activitat que és l'encarregada de mostrar el menú principal mitjançant un Intent.

### 6.1.3. Menú principal.

El menú principal consta d'una sola activitat anomenada V\_MenuUsuario. La classe Java que implementa aquesta activitat és V\_MenuUsuario.java i el arxiu XML és menú.xml.

Aquesta activitat és la que dona pas als principals casos d'ús que realitza l'aplicació. La pantalla consta de 5 botons que porten a cadascun dels casos d'ús. Dins de la implementació es registren els esdeveniments associats a cadascun dels botons i quan es fa clic en algun d'ells es llença la primera activitat del cas d'ús corresponent mitjançant un Intent.

### 6.1.3. Cas d'ús mostrar perfil.

Aquest cas d'ús consta d'una única activitat anomenada V\_Perfil que és implementada per la classe Java V\_Perfil.java i el arxiu XML perfil.xml. La pantalla mostra una sèrie d'etiquetes de text, disposades verticalment, on es mostrarà la informació de l'usuari i una "Image View" que mostra la seva imatge. Si el perfil mostrat és el de l'usuari, es mostra també un botó que li permet modificar el perfil.

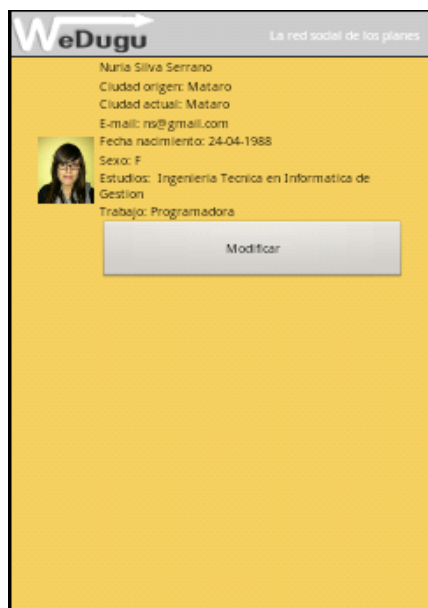


Figura 6. 1 Cas d'ús perfil

La classe V\_Perfil.java mitjançant una instància a la classe ConsumirWS crida al mètode llamarMetodoGetPerfilUser() que retornarà tota la informació necessària per omplir les caixes de text i la foto de l'usuari. Una vegada mostrades totes les dades per pantalla si, més a més, l'usuari del qual s'està mostrant el perfil és l'usuari que ha fet log in en l'aplicació també es mostra un botó "Modificar" i es registra el seu esdeveniment associat.

Quan l'usuari fa clic sobre el botó "Modificar" es substitueixen les etiquetes de text per caixes de text, on el contingut es pot modificar, amb la informació de l'usuari. També es

substitueix el botó “Modificar” per un botó “Enviar” del qual també s’enregistra l’esdeveniment clic. Per últim es mostra un missatge que recorda el format correcte en que s’ha d’informar la data de naixement de l’usuari. Quan l’usuari fa clic sobre el botó “Enviar” es comprova que el format de la data de naixement sigui el correcte i a continuació, mitjançant l’instància de la classe ConsumirWS, es crida al mètode llamarMetodoUpdatePerfil() que s’encarrega de guardar a la base de dades tota la informació modificada per l’usuari. A continuació es mostra un missatge informant de que tot ha anat correcte i es retorna al menú principal.

#### 6.1.4. Cas d’ús llistar amics.

Aquest cas d’ús és compostat per una sola activitat anomenada V\_ListaAmigos implementada per la classe V\_ListaAmigos.java i per l’arxiu XML listaamigos.xml.

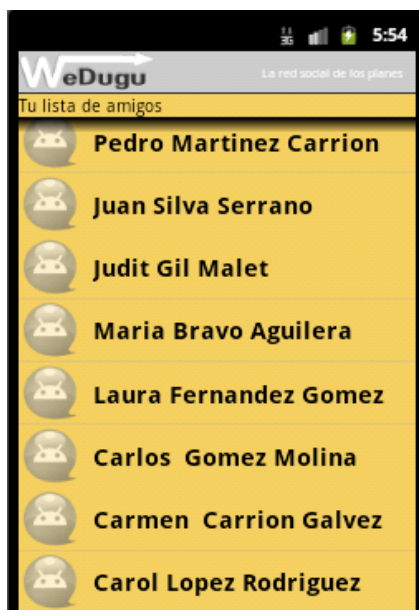


Figura 6. 2 Cas d’ús llistar amics

La classe V\_ListaAmigos.java el primer que fa és cridar, mitjançant una instància de la classe ConsumirWS, al mètode llamarMetodoGetAmics(), que retornarà la informació necessària per mostrar per pantalla, mitjançant un control ListView, la llista dels amics de l’usuari. Per poder omplir la llista es fa servir un Adapter, una classe base d’Android. Aquest Adapter s’encarrega de disposar la informació per a la llista. En aquest cas s’ha implementat una classe anomenada LazyAdapterAmigos.java que hereta de la classe BaseAdapter i que a més de carregar la informació implementa una cache per a que les imatges dels amics no s’hagin de descarregar cada vegada.

Es pot prémer sobre qualsevol dels elements del ListView.. Es registra l'esdeveniment clic sobre els elements del ListView i gràcies a un Intent es crida a la activitat V\_Perfil que s'encarrega de mostrar el perfil del amic sobre el qual s'ha realitzat el clic.

A l'arxiu listamigos.xml s'indica que existirà un ListView que, a la seva vegada, estarà format per elements descrits al fitxer items.xml.

Al fitxer items.xml es descriu que cada element estarà compost per una etiqueta de text o es mostrà el nom del amic i un ImageView per mostrar la seva imatge.

### 6.1.5. Cas d'ús consultar agenda.

El cas d'ús consultar agenda consta de dues activitats.

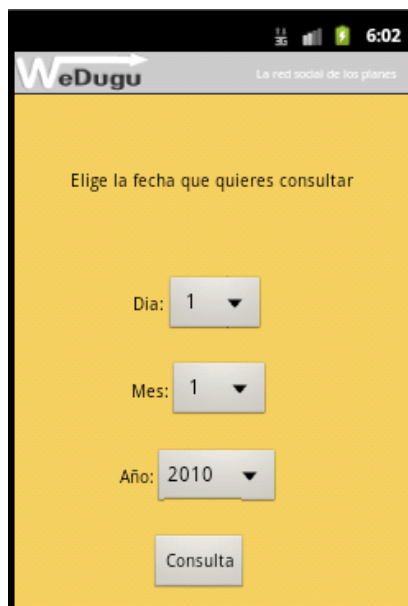


Figura 6. 3 Cas d'ús consulta agenda

#### Activitat V\_MenuConsultaAgenda.

La primera activitat és anomenada V\_MenuConsultaAgenda i és implementada per la classe V\_MenuConsultaAgenda.java i el fitxer XML consultaagenda.xml.

El fitxer consulta agenda descriu una pantalla on es mostraran tres controls spinner (Control per seleccionar una opció d'una llista) per escollir la data (dia, mes i any) dels esdeveniments de l'agenda a consultar i per un botó que executarà l'acció de consultar.



La classe `V_MenuConsultaAgenda.java` carrega els continguts dels components spinner i registra l'esdeveniment associat a l'acció de prémer el botó. Una vegada l'usuari fa clic sobre el botó, mitjançant un `Intent`, es cridarà a la segona activitat d'aquest cas d'ús, passant-li la data (dia, mes i any) que l'usuari vol consultar.

### Activitat `V_ListaEventosAgenda`.

La segona activitat que es fa servir per aquest cas d'ús és `V_ListaEventosAgenda`, és implementada per la classe `V_ListaEventosAgenda.java` i el fitxer XML `listaeventos.xml`.

Al fitxer `listaeventos.xml` es descriu una pantalla composta per una etiqueta de text i un component `ListView` on es mostraran els esdeveniments de l'usuari.

A la classe `V_ListaEventosAgenda.java` el primer que es realitza és la crida, mitjançant una instància a la classe `ConsumirWS`, del mètode `llamarMetodoGetEventosAgenda()` passant-li el dia concret pel qual l'usuari vol realitzar la consulta. Una vegada es té la informació dels esdeveniments, mitjançant un `Adapter` privat de la classe, s'omple el contingut del `ListView` descrit al fitxer XML. Si l'usuari no disposa d'esdeveniments per al dia consultat, és a dir, el mètode cridat no retorna cap informació, es mostra un missatge advertint d'aquest incident i es finalitza l'activitat.

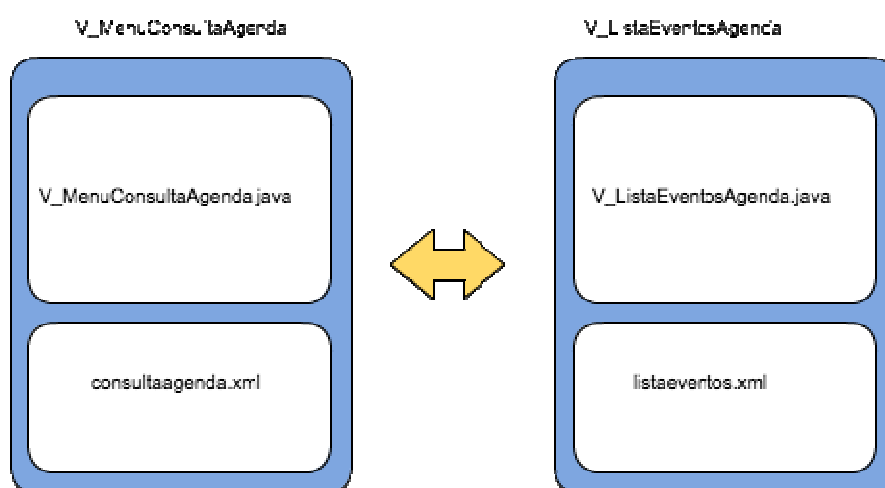


Figura 6. 4 Esquema activitats agenda

### 6.1.6. Cas d'ús demanar proposta.

El cas d'ús demanar proposta consta de quatre activitats.



Figura 6. 5 Cas d'ús demanar proposta

#### Activitat V\_MenuPropuesta.

La primera és V\_MenuPropuesta, i és implementada per la classe Java V\_MenuPropuesta.java i pel fitxer XML menupropuesta.xml.

El fitxer menupropuesta.xml descriu una pantalla amb un control spinner per escollir el tipus de proposta que vol rebre l'usuari i un botó per executar la petició de proposta.

El primer que realitza la classe V\_MenuPropuesta.java és comprovar que el dispositiu mòbil tingui el GPS activat. En cas negatiu, es mostrarà un missatge informant de la necessitat que estigui activat. Una vegada l'usuari ha acceptat el missatge, mitjançant un Intent es mostra la pantalla de preferències del terminal mòbil per tal de que l'usuari activi el GPS. L'usuari tornarà enrere i l'activitat tornarà a verificar la disponibilitat del GPS, repetint aquest procés si encara no es troba activat. Per poder implementar aquesta casulística el codi que comprova si el GPS és actiu i tot el que es realitzi posteriorment haurà d'estar inclòs en el mètode onResume(), ja que és el mètode que s'executa quan l'usuari torna després de passar per la pantalla de preferències.

Una vegada comprovada la disponibilitat del GPS la classe llença un nou fil d'execució que serà l'encarregat de capturar la posició geogràfica en la que es troba el dispositiu. El nou fil d'execució fa servir la classe base LocationManager per obtenir la posició geogràfica, mitjançant l'enregistrament dels esdeveniments de canvi de posició. Quan un canvi és detectat, els atributs de longitud i latitud de l'activitat son informats i es finalitza el registre de l'esdeveniment i el fil d'execució.

A continuació l'usuari pot escollir, en un control spinner mostrat per pantalla, de quin tipus d'activitat vol rebre la proposta. Per fer això s'han omplert les opcions del spinner amb un Adapter i enregistrar l'esdeveniment clic del botó "Go!".

Quan l'usuari fa clic sobre el botó es llença la segona activitat del cas d'ús, V\_ListaPropuestas.

### **Activitat V\_ListaPropuestas.**

La segona activitat d'aquest cas d'ús es V\_ListaPropuestas i és implementada per la classe V\_ListaPropuestas.java i el fitxer XML listapropuestas.xml.

El fitxer listapropuestas.xml descriu una pantalla on es mostra una etiqueta de text i un control ListView on es llistaran les diferents propostes.

La classe V\_ListaPropuestas.java rep la informació proporcionada per l'activitat anterior sobre el tipus de proposta que vol l'usuari i la longitud i latitud geogràfica d'aquest. En funció del tipus de proposta es cridarà un mètode o un altre de la instància de la classe ConsumirWS. Els 5 possibles mètodes cridats son:

- llamarMetogetPropuestaRestau(Double lat, Double lon, int id).
- llamarMetogetPropuestaActi(Double lat, Double lon, int id).
- llamarMetogetPropuestaConcie(Double lat, Double lon, int id).
- llamarMetogetPropuestaPubBar(Double lat, Double lon, int id).
- llamarMetogetPropuestaCine(Double lat, Double lon, int id).

El mètode cridat omplirà un vector amb les propostes rebudes pel servei web. Si el vector de propostes no és buit, és a dir, el servei web ha tornat propostes per l'usuari, es carregarà la informació de les propostes en el control ListView mitjançant un Adapter privat descrit a

la mateixa classe. Els elements del control ListView es poden prémer, per tant, s'ha d'enregistrar l'event clic sobre qualsevol dels seus elements. Quan polsi sobre un element del ListView, és a dir, sobre una proposta, es llançarà la següent activitat, V\_Propuesta, d'aquest cas d'ús.

### **Activitat V\_Propuesta.**

La tercera activitat d'aquest cas d'ús es V\_Propuesta que estarà implementada per la classe V\_Propuesta.java i el fitxer xml propuesta.xml.

El fitxer propuesta.xml descriu una pantalla que mostrarà una sèrie d'etiquetes de text on es mostrarà tota la informació de la proposta i dos botons: “Ver mapa” i “Inclure en Agenda”.

La classe V\_Propuesta.java rep tota la informació de la proposta de la classe anterior. La mostra per pantalla i enregistra els esdeveniments de clic sobre els botons “Ver Mapa” i “Inclure en agenda”.

Si l'usuari fa clic sobre “Ver Mapa” serà llençada l'activitat V\_Mapas i si fa clic sobre “Inclure en agenda” es cridarà al mètode llamarMetodoInsertAgendaEvento() de la classe ConsumirWS per inserir a l'agenda de l'usuari la proposta d'activitat.

### **Activitat V\_Mapas.**

L'activitat V\_Mapas és implementada per la classe V\_Mapas.java i el fitxer XML mapas.xml.

El fitxer mapas.xml descriu una pantalla que incorporà un control MapView.

La classe V\_Mapas.java rebrà la informació de longitud i latitud geogràfica de la proposta i de l'usuari i crearà un mapa amb controls de zoom, també situarà sobre aquest una capa per sobre del mapa on mostrarà la posició de l'usuari i de la proposta. Per realitzar aquesta funcionalitat cal implementar una classe privada, en aquest cas anomenada propuestas, que hereti de la classe base ItemizedOverlay.

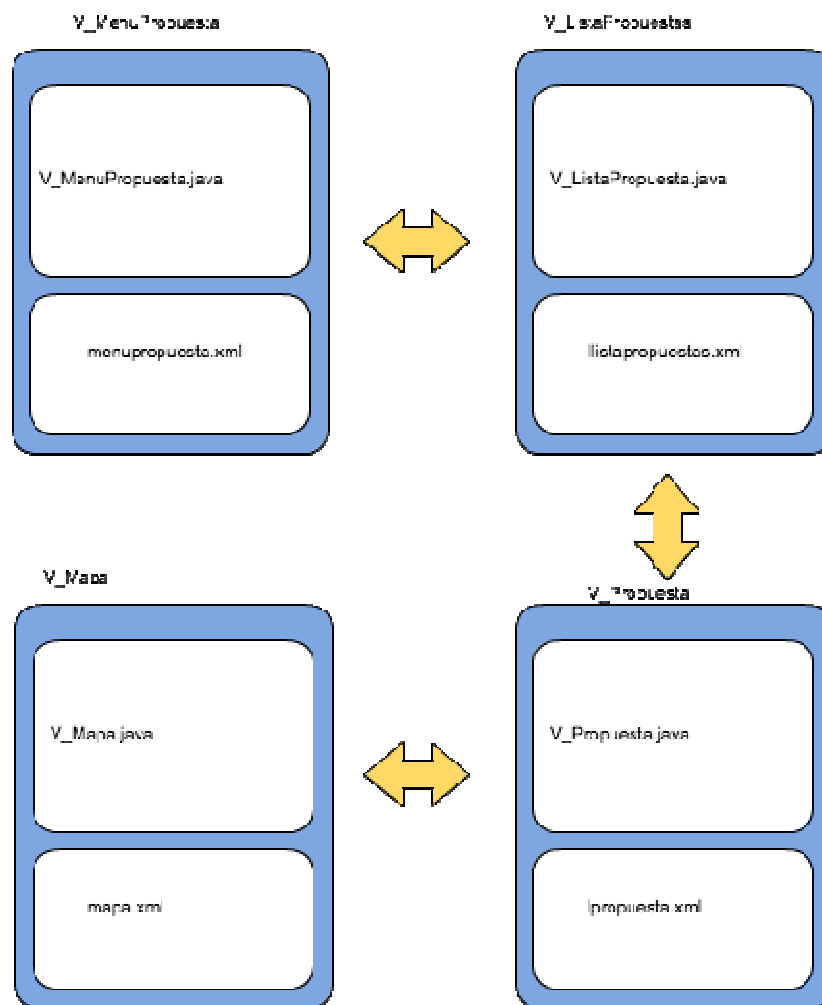


Figura 6. 6 Esquema d'activitats cas d'ús demanar proposta

### 6.1.7. Cas d'ús cercar.

El cas d'ús cercar consta de quatre activitats.

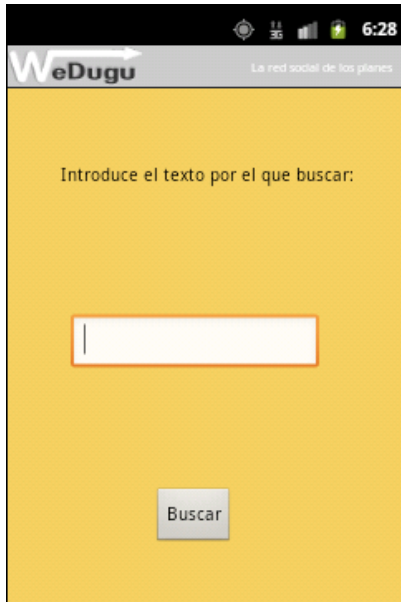


Figura 6. 7 cas d'ús cercar

#### Activitat V\_Busqueda.

L'activitat V\_Busqueda és implementada per la classe V\_Busqueda.java i el fitxer XML `busqueda.xml`.

El fitxer `busqueda.xml` descriu una pantalla on es mostra una caixa de text on introduir el criteri de la cerca i un botó per executar l'acció de cercar.

La classe V\_Busqueda.java el primer que realitza és enregistrar l'esdeveniment de clic sobre el botó. Quan l'usuari fa clic sobre el botó es comprova que hagi introduït text a la caixa i si es així mitjançant un Intent es crida a la següent activitat, V\_BusquedaTabWidget.

### **Activitat V\_BusquedaTabWidget.**

Aquesta activitat és diferent respecte les descrites anteriorment ja que no hereta de la classe base Activity sinó de TabActivity. Es fa que hereti d'aquesta classe base per crear una activitat per pestanyes. Aquesta activitat és implementada per la classe V\_BusquedaTabWidget.java i per el fitxer XML tabbusqueda.xml

El fitxer tabbusqueda.xml descriu una pantalla on es mostra un component TabWidget i un component FrameLayout que s'encarregarà de contenir les dues activitats que seran cridades per les pestanyes.

La classe V\_BusquedaTabWidget.java és l'encarregada de crear les dues pestanyes que es volen mostrar al usuari: "Amigos" i "Personas". També es registra l'esdeveniment clic sobre aquestes. Quan l'usuari clica sobre la pestanya "Amigos" es carrega dins del FrameLayout l'activitat V\_BusquedaAmigos, en canvi si clica sobre la pestanya "Personas" es carrega dins del FrameLayout l'activitat V\_BusquedaPersonas.

### **Activitat V\_BusquedaAmigos.**

L'activitat V\_BusquedaAmigos és implementada per la classe V\_BusquedaAmigos.java i el fitxer XML listabusquedaamigos.xml.

El fitxer listabusquedamigos.xml descriu una pantalla on es mostrarà un component ListView on es mostra una llista d'amics de l'usuari coincidents amb el text de la cerca.

La classe V\_BusquedaAmigos.java rebrà el text per el qual vol cercar l'usuari i cridarà, mitjançant una instància de la classe ConsumirWS, el mètode llamarMetodoBuscar(), que retornarà un vector amb els amics coincidents amb la cerca. S'implementarà una classe privada que hereti de la classe base Adapter per mostrar la informació continguda al vector dins del component ListView. Els elements del ListView no es poden prémer aquesta vegada.

**Activitat V\_BusquedaPersonas.**

L'activitat V\_BusquedaPersonas és implementada per la classe V\_BusquedaPersonas.java i el fitxer XML listabusquedapersonas.xml.

El fitxer listabusquedapersonas.xml descriu una pantalla on es mostrarà un component ListView que presentarà una llista d'usuaris de WeDugu, coincidents amb el text de la cerca, que no son amics seus.

La classe V\_BusquedaPersonas.java rebrà el text per el qual vol cercar l'usuari i cridarà, mitjançant una instància de la classe ConsumirWS, el mètode llamarMetodoBuscar(), que retornarà un vector amb els usuaris de WeDugu coincidents amb la cerca que no son amics de l'usuari. S'implementarà una classe privada que hereti de la classe base Adapter per mostrar la informació continguda al vector dins del component ListView. Els elements del ListView no son clicables aquesta vegada.



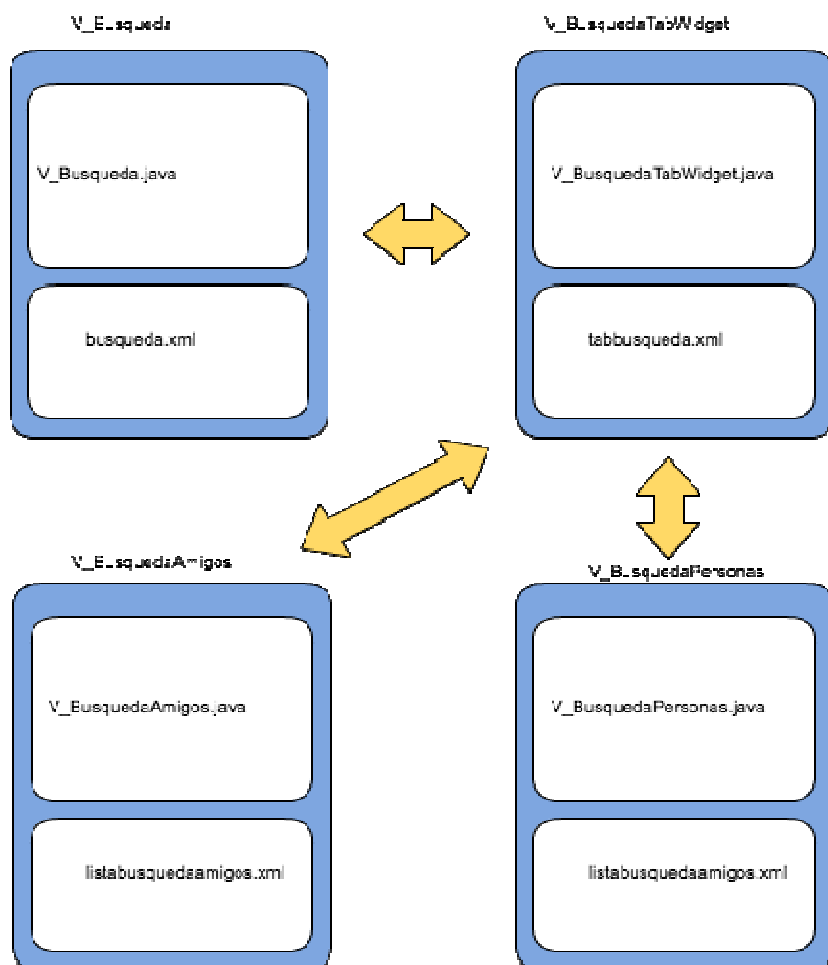


Figura 6. 8 Esquema activitats cas d'ús cercar

### 6.1.8. Activitat V\_Base.

Totes les activitats anteriorment descrites, excepte, V\_Indice, V\_Login, V\_Mapa i totes les que implementen les pestanyes hereten d'aquesta classe i no de la classe Activity directament. La funcionalitat que s'aconsegueix fent que heretin d'aquesta classe és que la resta d'activitats disposen d'una barra superior amb el logotip de l'aplicació i des del qual es pot tornar al menú principal. Aquesta activitat està implementada per la classe V\_Base.java i per el fitxer XML titulowedugu.xml

Al fitxer titulowedugu.xml es descriu una pantalla on es mostra una barra superior amb el logotip de l'aplicació, que es pot polsar i una etiqueta de text.

La classe V\_Base.java s'encarrega d'establir que el fitxer titulowedugu.xml es un títol personalitzat de l'aplicació i d'enregistrar l'esdeveniment de clic sobre el logotip. Quan

l'usuari fa clic sobre el logotip, mitjançant un intent, es torna a mostrar a l'usuari el menú principal.

### **6.1.9. Classe Sesion.**

La finalitat de la classe Sesion.java és guardar l'identificador de l'usuari que es troba identificat a l'aplicació i així no haver-lo de passar constantment de aplicació en aplicació, per a la crida de mètodes del servei web.

### **6.1.10. Paquet domini.**

El paquet domini el componen les classes Propuesta.java, PropuestaCine.java i Evento.java. Aquestes tres classes serveixen per encapsular la informació retornada per el servei web respecte propostes, propostes de pel·lícules i esdeveniments de l'usuari. Totes tres classes disposen dels seus respectius constructors, mètodes gets i mètodes sets.

### **6.1.11. Paquet utils.**

Aquest paquet conté classes d'utilitats que es fan servir per a les activitats descrites anteriorment. Per exemple, la classe ImageLoader.java, continguda en aquest paquet, s'encarrega d'implementar la caché abans mencionada amb la qual la llista d'amics no descarrega les fotografies d'aquest més d'un cop.

## **6.2. Servei web.**

### **6.2.1. Paquet BBDD.**

El paquet BBDD conté dues classes: "ConnexioBBDD" i "BD\_User".

#### **Classe ConnexioBBDD.**

La classe ConnexioBBD és l'encarregada de gestionar la connexió amb la base de dades. S'ha aplicat el patró Singleton. Consta del mètodes:

- getInstance(): Retorna una instància de la classe.
- getConnection(): Retorna una connexió a la base de dades.

- connect(): Crea la connexió amb la base de dades, definint el seus paràmetres.
- disconnect(): Tanca la connexió amb la base de dades.

### **Classe BD\_User.**

Especificació de la classe:

La classe BD\_User conté els mètodes que realitzen les diferents consultes a la base de dades per retornar la informació demanada per l'aplicació Android.

Consta dels següents mètodes:

- getUser\_Nom(String email, String pass): Retorna l'identificador de l'usuari si aquest existeix a la base de dades.
- getAmics(int id): Retorna una llista amb els amics de l'usuari.
- validarEmail(String email): Retorna l'identificador d'usuari del e-mail indicat com a paràmetre.
- setPerfilUser(int id,...): Actualitza la informació de l'usuari amb la informació passada com a paràmetre.
- getPerfilUser(int id): Retorna tota la informació referent al perfil de l'usuari especificat.
- getPropuestaRestau(String lat, String lon, int id\_user): Retorna propostes de restaurants per un usuari, en funció de les preferències guardades en el seu perfil i en un radi inferior a 3 quilometres des de la posició passada com a paràmetre.
- getPropuestaActi(String lat, String lon, int id\_user): Retorna propostes de activitats per un usuari, en funció de les seves preferències i en un radi inferior a 3 quilometres des de la posició passada com a paràmetre.
- getPropuestaConcie(String lat, String lon, int id\_user): Retorna propostes de concerts per un usuari, en funció del gustos informats en el seu perfil i en un radi inferior a 3 quilometres des de la posició passada com a paràmetre.
- getPropuestaPubBar(String lat, String lon, int id\_user): Retorna propostes de pub/bar per un usuari, en funció del gustos informats en el seu perfil i en un radi inferior a 3 quilometres des de la posició passada com a paràmetre.

- `getPropuestaCine(String lat, String lon, int id_user)`: Retorna propostes de pel·lícules per un usuari, en funció del gustos informats en el seu perfil i en un radi inferior a 3 quilometres des de la posició passada com a paràmetre.
- `insertAgendarEvento(int idUser,...)`: Afegeix a l'agenda de l'usuari com un nou esdeveniment una des les propostes retronades per algun dels mètodes anteriors.
- `getEventosAgenda(int id, int day, int month, int year)`: Retorna tots els esdeveniments que té programats un usuari per a una data en concret.
- `buscarAmigos(int id, String text)`: Retorna amics de l'usuari especificat que coincideixin amb el text passat com a paràmetre.
- `buscarPersonas(int id, String text)`: Retorna usuaris de WeDugu que no son amics de l'usuari especificat que coincideixen amb el text especificat.

En els mètodes que retornen propostes per a l'usuari especificat s'utilitza la fórmula Haversine per tal de trobar la distància entre dos punts especificant les seves coordenades geogràfiques.

### **Fórmula Haversine.**

La fórmula Haversine és una important equació per a la navegació astronòmica en quant al càlcul de la distància del cercle màxim entre dos punts sabent les longituds i latituds. És un cas especial d'una fórmula més general de trigonometria esfèrica, la llei dels semiversens, que relaciona els costats i els angles de "triangles esfèrics".

Fòrmula:

$$\text{havarsin} \left( \frac{d}{R} \right) = \text{havarsin}(\varphi_1 - \varphi_2) + \cos(\varphi_1) \cos(\varphi_2) \text{havarsin}(\Delta\lambda).$$

Figura. 6.9 Fórmula Haversine

On:

- havarsin és la funció havarsine:

$$\text{havarsin}(\theta) = \text{sen}^2(\theta/2) = (1 - \cos(\theta))/2$$

- d és la distància entre els dos punts.

- $R$  és el radi de la esfera.
- $\Phi 1$  és la latitud del punt 1.
- $\Phi 2$  és la latitud del punt 2.
- $\Delta\lambda$  és la diferència de longitud

### 6.2.2. Paquet WS.

El paquet WS consta d'una única classe anomenada Wswedugu.java que especifica el codi del servei web.

Classe WSwedugu.

La classe WSwedugu especifica el servei web, és a dir, fa servir anotacions per descriure cadascun dels mètodes dels quals disposa el servei web per retornar informació de la base de dades.

Per descriure que la classe implementa un servei web es fa servir l'anotació “@WebService” just a sobre de la declaració de la classe. Per cada mètode de la classe s'ha d'especificar que serà un mètode web, per fer això es fa servir l'anotació “@WebMethod(operation=”)” on s'especifica el nom de l'operació. A cada paràmetre dels diferents mètodes s'ha d'especificar que serà un paràmetre de l'operació del servei web, per això s'utilitza l'anotació “@WebParam(name=”)” que s'escriu just davant del paràmetre del mètode i on s'especifica el nom del paràmetre web.

La classe WSwedugu disposa del següents mètodes web:

- getUser: Crida al mètode getUser\_Nom de la classe BD\_User.
- getAmics: Crida al mètode getAmics de la classe BD\_User.
- validarEmail: Crida al mètode validarEmail de la classe BD\_User.
- setPerfilUser: Crida al mètode setPerfilUser de la classe BD\_User.
- getPerfilUser: Crida al mètode getPerfil User de la classe BD\_User.
- propRes: Crida al mètode getPropuestaRestau de la classe BD\_User.
- propActi: Crida al mètode getPropuestaActi de la classe BD\_User.
- propConcie: Crida al mètode getPropuestaConcie de la classe BD\_User.
- propCine: Crida al mètode getPropuestaCine de la classe BD\_User.
- agendarEvento: Crida al mètode insertAgendarEvento de la classe BD\_User.

- `getEventosAgenda`: Crida al mètode `getEventosAgenda` de la classe `BD_User`.
- `buscarAmigos`: Crida al mètode `buscarAmigos` de la classe `BD_User`.
- `buscarPersonas`: Crida al mètode `buscarPersonas` de la classe `BD_User`.

## 7. Manual d'usuari.

### 7.1. Inici.

Un cop arrencada l'aplicació apareixerà una pantalla amb el seu logotip. Prement el botó “Menú” del dispositiu apareixerà un menú com el mostrat a la imatge següent des d'on es podrà escollir entrar (“Log in”) o sortir (“Salir”) de l'aplicació.



Figura 7. 1 Índex WeDugu

### 7.2. Entrada a l'aplicació i menú principal.

Si es selecciona l'opció “Salir” l'aplicació es tancarà, mentre que si es polsa sobre l'opció “Log in” es passarà a la pantalla següent:

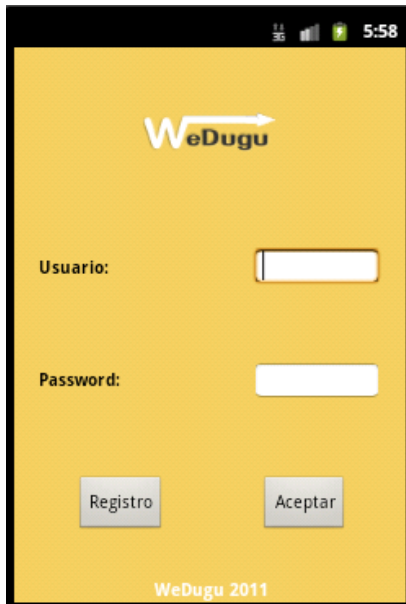


Figura 7. 2 Log in

En primer lloc cal introduir un nom d'usuari i una contrasenya vàlids i a continuació prémer el botó “Aceptar”, .Si l'usuari existeix es mostrarà un missatge informatiu i seguidament es carregarà el menú principal de l'aplicació, que permet navegar a qualsevol opció.



Figura 7. 3 Menú Principal



### 7.3. Perfil.

Per mostrar el perfil del usuari que ha entrat en l'aplicació caldrà polsar sobre el botó perfil.

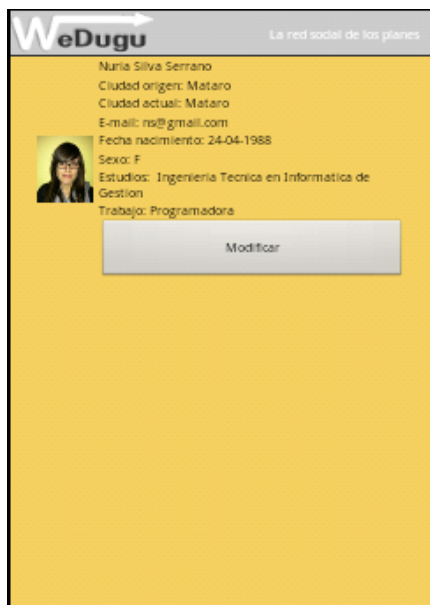


Figura 7. 4 Perfil usuari

A continuació, si es vol modificar alguna dada del perfil, s'haurà de prémer el botó "Modificar" i les dades es mostraran en caixes de text modificables. Abans de mostrar la caixes de text se li recorda a l'usuari el format en que ha d'entrar la data.

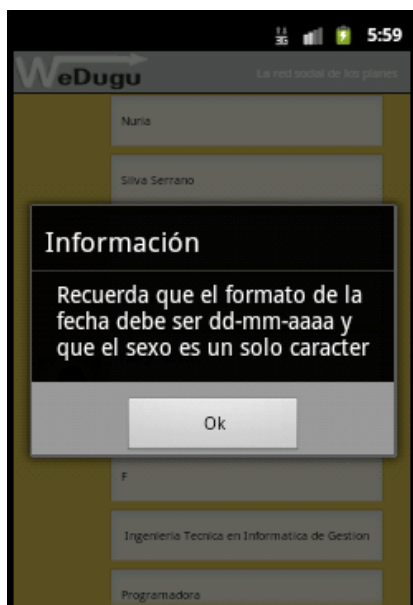


Figura 7. 5 Missatge recordatori format dades

Una vegada actualitzades les dades, prémer “Enviar” i el perfil s’actualitza, es mostra un missatge informant de l’actualització i es retorna al menú principal.

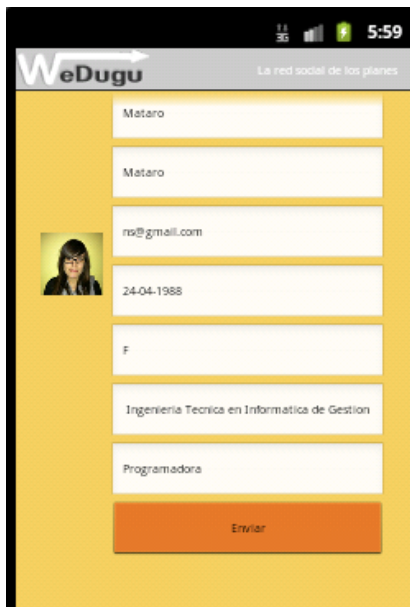


Figura 7. 6 Confirmació de canvis en el perfil

#### 7.4. Llista d'amics.

Per obtenir la llista d'amics de l'usuari s'ha de pulsar sobre el botó “Amigos” del menú principal.

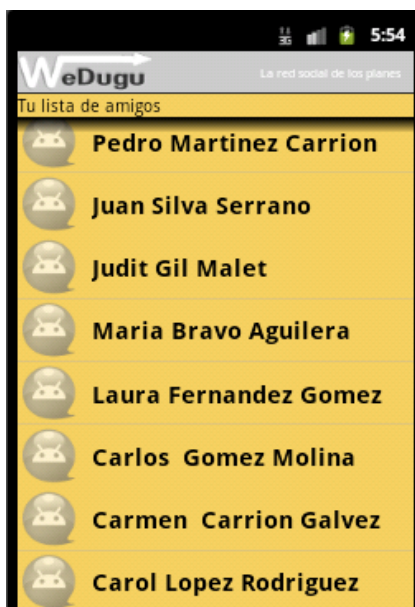


Figura 7. 7 Llista d'amics

Per accedir a les dades del perfil d'un amic només cal polsar sobre l'amic a la llista d'amics i es mostrarà la seva informació..

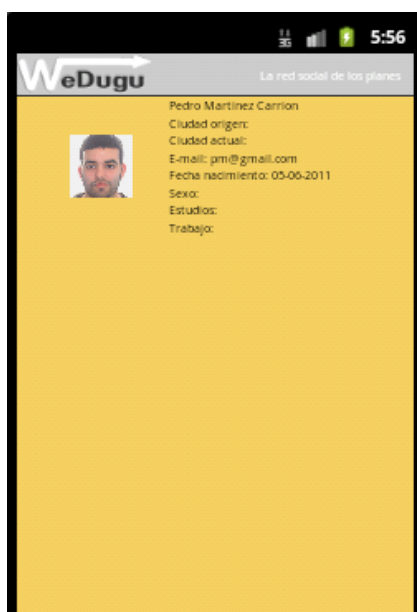


Figura 7. 8 Perfil amic

## 7.5. Consulta agenda.

Per consultar l'agenda s'ha de prémer el botó "Agenda" del menú principal.

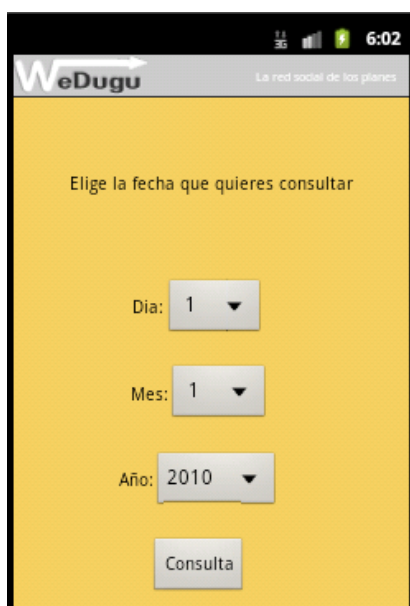


Figura 7. 9 Consulta agenda

Una vegada seleccionats dia, mes i any s'executarà la consulta prement sobre el botó "Consulta" i apareixerà una llista amb els esdeveniments programats pel dia especificat.



Figura 7. 10 Llista esdeveniments

### 7.6. Demanar proposta.

Per demanar una proposta s'ha de fer clic sobre el botó "Propuesta" del menú principal. Apareixerà una pantalla que trobarà la posició geogràfica de l'usuari i on caldrà seleccionar el tipus de proposta i fer clic sobre el botó "Go".



Figura 7. 11 Selecció del tipus de proposta

A continuació apareixerà un llista amb les propostes més properes a l'usuari.

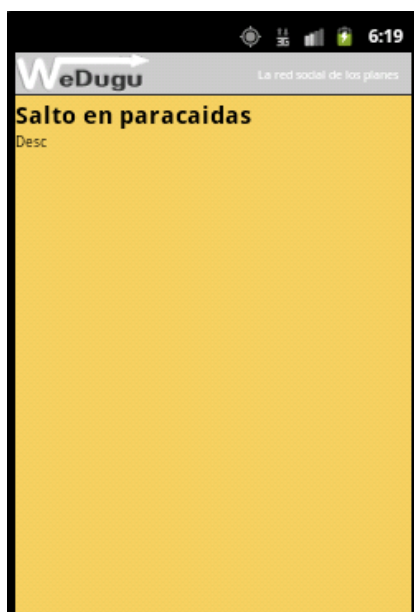


Figura 7. 12 Llistat de propostes

Per veure el detall de la proposta cal prémer sobre aquesta i es mostrarà una pantalla amb la informació. També es podrà consultar el mapa per localitzar la proposta i afegir-la a l'agenda.

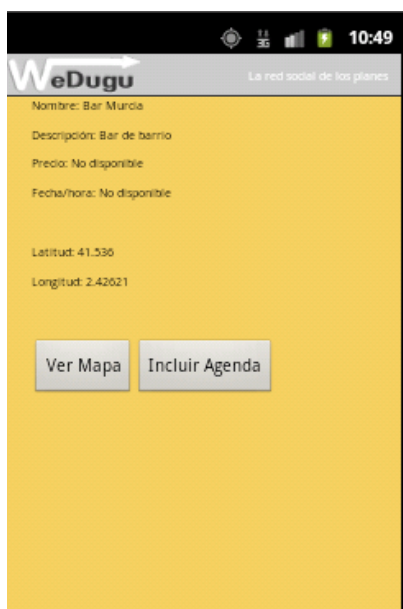


Figura 7. 13 Informació proposta

Per veure el mapa prémer sobre el botó “Ver Mapa”.

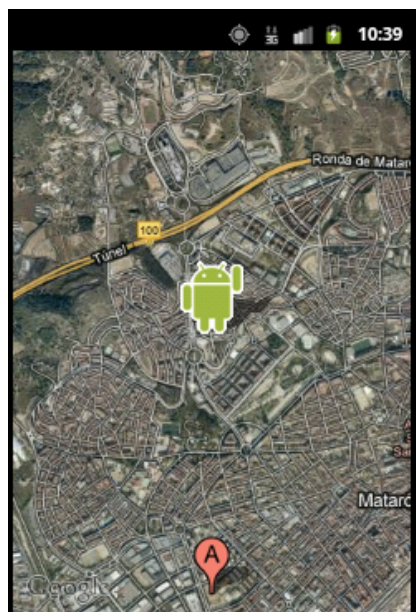


Figura 7. 14 Mapa proposta

Per incloure la proposta en l'agenda prémer sobre el botó “Incluir Agenda”.

## 7.7. Cercar.

Per cercar usuaris entre els amics o entre la resta d'usuaris de l'aplicació cal prémer sobre el botó "Buscar" del menú principal. Es mostrarà una pantalla amb un quadre de text on s'haurà d'introduir el criteri de cerca i un botó. Per executar la cerca caldrà prémer el botó.

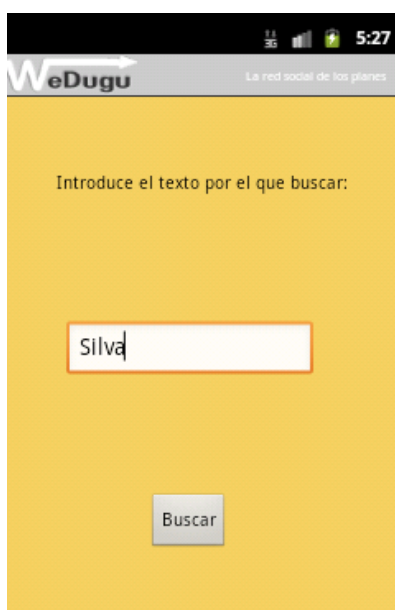


Figura 7. 15 Cercar

Una vegada realitzada l'acció de cerca, es mostrarà el resultat en una pantalla amb dues pestanyes, una mostrarà el resultat de la cerca entre els amics i l'altre entre la resta d'usuaris.

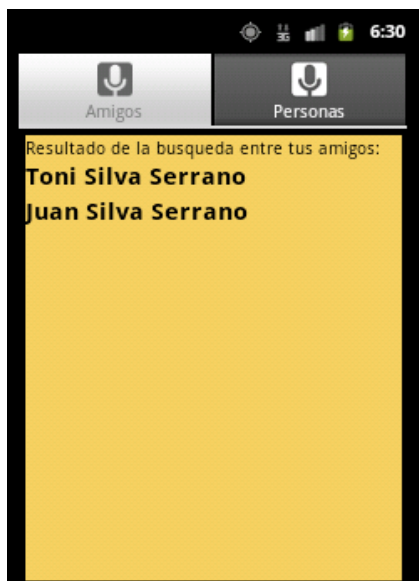


Figura 7. 16 Resultat de la cerca



## 8. Proves.

A continuació es detallen, en format de taula, les proves realitzes sobre l'aplicació especificades per els diferents casos d'ús.

### 8.1. Log in.

Acció	Resultat esperat	Prova	Resultat obtingut
Introduir un usuari no existent a la base de dades	Mostrar missatge advertint que l'usuari no existeix	Usuari: <a href="mailto:pppp@gmail.com">pppp@gmail.com</a> Contrasenya: aaa	Ok
Introduir un usuari existent a la base de dades amb la contrasenya errònia	Mostrar missatge advertint que usuari o contrasenya son incorrectes	Usuari:ns@gmail.com Contrasenya: aaa	Ok

Taula 8. 1 Proves Log in

### 8.2. Perfil.

Acció	Resultat esperat	Prova	Resultat obtingut
Consulta el perfil d'un usuari que no té dades informades	Mostrar dades només referents al accés,	Usuari: <a href="mailto:wdugu@gmail.com">wdugu@gmail.com</a> Contrasenya: wdugupasswd	Ok
Informa la data en un format erroni.	Mostrar missatge advertint de la incidència i no actualitzar perfil	Usuari:ns@gmail.com Contrasenya: nspasswd	Ok

Taula 8. 2 Proves perfil

### 8.3. Llistat d'amics.

Acció	Resultat esperat	Prova	Resultat obtingut
Consulta la llista d'amics d'un usuari sense amics.	Mostrar missatge informat que l'usuari no té amics	Usuari: prova@prova.com Contrasenya: prova	Ok

Taula 8. 3 Proves llistat amics

### 8.4. Consultar agenda.

Acció	Resultat esperat	Prova	Resultat obtingut
Consultar agenda d'un dia sense esdeveniments	Mostrar missatge informat que l'usuari no té esdeveniments.	Usuari: ns@gmail.com Contrasenya: nspasswd Data: 1-1-2010	Ok
Consulta agenda d'un dia amb esdeveniments	Llistar esdeveniments	Usuari: ns@gmail.com Contrasenya: nspasswd Data: 6-5-2011	Ok

Taula 8. 4 Proves consulta agenda.

### 8.5. Demanar proposta.

Acció	Resultat esperat	Prova	Resultat obtingut
Demanar proposta estant en ubicació sense propostes properes	Mostrar missatge informant que no hi ha propostes disponibles	Usuari: ns@gmail.com Contrasenya: nspasswd Latitud: 42,2481 Longitud: 4,12277	Ok
Tenir desactivat GPS	Mostrar missatge informant necessitat GPS i navegació preferències	Usuari: ns@gmail.com Contrasenya: nspasswd	ok
No activar GPS a les preferències després de ser enviat	Tornar a mostrar missatge	Usuari: ns@gmail.com Contrasenya: nspasswd	Ok
Estar en unes coordenades amb propostes properes	Llistar propostes	Usuari: ns@gmail.com Contrasenya: nspasswd	Ok

Veure mapa de proposta	Mapa de la proposta indicant la localització d'aquesta i de l'usuari	Usuari: ns@gmail.com Contrasenya: nspasswd	Ok
------------------------	--	---	----

Taula 8. 5 Proves demanar proposta.

### 8.6. Cercar.

<b>Acció</b>	<b>Resultat esperat</b>	<b>Prova</b>	<b>Resultat obtingut</b>
Realitzar cerca sense resultats	Mostrar missatge informant que la cerca no té resultats.	Usuari: prova@prova.com Contrasenya: prova Cerca: Silva	Ok
Realitzar cerca amb resultats	Llistar resultats de la cerca	Usuari: prova@prova.com Contrasenya: prova Cerca: Silva	Ok

Taula 8. 6 Proves cercar.



## 9. Valoració econòmica.

A continuació es detallarà la despesa econòmica que comportaria aplicar aquest projecte a la realitat.

El preu estipulat de dedicació per persona és de 20€ l'hora a la fase d'anàlisi del projecte i 10€ l'hora a la fase de desenvolupament. S'ha de tenir en compte que en aquest projecte han participat dues persones, per tant, alguns casos es multiplicaran per dos.

### 9.1 Anàlisi del projecte.

Tasca	Duració
Planificació	10h / persona
Anàlisi(Casos d'ús)	40h / persona
Estudi Tecnologies	20h / persona
<b>Total anàlisi del projecte per persona</b>	1400,00 €/persona
<b>Total anàlisi del projecte</b>	<b>2.800,00 €</b>

Taula 9. 1 Valoració anàlisi

### 9.2 Eines utilitzades.

Eina	Cost
Netbeans 6.8 (2 unitats)	Gratuït
Sistema Operatiu MAC OS X (2 unitats)	169,00 € x 2
Eclipse SDK 3.5 (1 unitat)	Gratuït
Adobe Fireworks CS5 (1 unitat)	149,00 €
Enterprise Architect 9 (Desktop Edition Standard License) (1 unitat)	93,42 € (135 \$)
Windows 2003 Server (1 unitat)	150,00 €
MySQL	Gratuït
Tomcat	Gratuït
Apache 2	Gratuït
Microsoft Office para Mac (2 unitats)	139,00 € x 2
<b>Total eines utilitzades</b>	<b>1.008,42 €</b>

Taula 9. 2 Valoració eines utilitzades

- Netbeans i Eclipse han estat utilitzats per tota la fase de codificació.
- El sistema operatiu MAC OS X, és l'instal·lat als ordinadors de treball.
- Adobe Fireworks s'ha fet servir per dissenyar les interfícies.
- Enterprise Architect s'ha utilitzat per realitzar els diagrames d'anàlisi de l'aplicació.
- Windows 2003 Server, MySQL, Tomcat i Apache 2 son les eines necessàries per el servidor de l'aplicació.
- Microsoft Office for Mac ha sigut necessari per escriure la documentació d'aquest projecte.

Es considera que aquestes eines seran utilitzades a pròxims projectes, per tant només s'ha de calcular la part proporcional a aquest. Considerant que les eines seran renovades cada any i que s'ha utilitzat durant 2 mesos l'amortització corresponent a aquest projecte serà de 168,07 €.

### 9.3 Codificació.

<b>Tasca</b>	<b>Duració</b>
Disseny web	80h (1 persona)
Creació de capa domini amb JPA	20h (1 persona)
Formació Android	80h (1 persona)
Codificació casos d'ús generals Web	80h (1 persona)
Codificació casos d'ús general Android	80h (1 persona)
Codificació Casos d'ús socials Web	40h (1 persona)
Codificació Casos d'ús socials Android	40h (1 persona)
Codificació Casos d'ús agenda Web	100h (1 persona)
Codificació Casos d'ús agenda Android	100h (1 persona)
Revisió final Web	20h (1 persona)
Revisió final Android	20h (1 persona)
<b>Total codificació del projecte</b>	<b>6.600,00 €</b>

Taula 9. 3 Valoració codificació

A les hores de codificació ja hi són incloses les hores dedicades a les proves de l'aplicació.

## 9.4 Documentació.

Tasca	Duració
Redacció	60h / persona
<b>Total documentació del projecte</b>	<b>1.200,00 €</b>

Taula 9. 4 Valoració documentació

El preu establert per temps de dedicació a la redacció de la documentació d'aquest projecte es considera el mateix que per la codificació, és a dir, 10€ l'hora.

## 9.5 Hardware utilitzat.

Equip	Cost
Macbook Pro	1.400,00 €
Macbook	840,00 €
Portàtil Ahtec Singal zw6	300,00 €
<b>Total documentació del projecte</b>	<b>2.540,00 €</b>

Taula 9. 5 Cost hardware

Considerem que la vida útil dels ordinadors és de 3 anys, comptant que la durada d'aquest projecte és de 4 mesos, apliquem el factor d'amortització al total obtingut i obtenim un cost d'amortització del hardware de 282,22 €.

## 9.6 Cost total del projecte.

Concepte	Cost
Anàlisi del projecte	2.800,00 €
Eines utilitzades	168,07 €
Codificació	6.600,00 €
Documentació	1.200,00 €
Hardware utilitzat	282,22 €
Costos indirectes	1.657,54
<b>Total cost del projecte</b>	<b>12.707,83 €</b>

Taula 9. 6 Suma total cost projecte

Els costos indirectes són els corresponents a energia, comunicacions, consumibles, lloguer, amortització de mobiliari, etc. Es considera un 15% del total del projecte.

Tenint en compte que aquest projecte engloba 2 projectes (cadascun realitzat per una persona), aquest projecte no es pot considerar excessivament car per que s'ha d'apreciar el cost de les eines que han sigut necessàries i que l'esforç ha sigut multiplicat per dos.



## 10. Conclusions.

Dins de l'aplicació Android es podrien introduir certes millores, la primera d'aquestes podria ser la millora de l'aspecte gràfic o visual de l'aplicació ja que el disseny de pantalles no és massa cridaner i vistós perquè el cost del desenvolupament de la part gràfica era molt elevat. Com a part de la millora del disseny de l'aplicació es podrien introduir alguns dels components de formulari dels quals disposa Android per a la selecció de dates, hores o caixes amb funció "autocompletar", etc.

Un altre millora ha introduir dins de l'aplicació Android seria oferir la possibilitat d'obtenir el registre en el servei des d'aquesta ja que, actualment, un usuari només es pot registrar des de l'aplicació web.

Disposant de més temps i després de conèixer més profundament el desenvolupament per Android es podria fer que l'aplicació per a dispositius mòbils realitzés tots o gairebé tots els casos d'ús que realitza l'aplicació web ja que ara només realitza els més importants, però no la totalitat d'aquests.

Una ampliació necessària perquè el projecte pogués ser comercialitzat seria l'aplicació de seguretat al servei web que proveeix d'informació l'aplicació Android. Seria imprescindible la implementació d'un sistema de seguretat en que només sabent usuari i contrasenya es pogués demanar informació a la base de dades.

Una vegada realitzat el projecte s'ha après a desenvolupar aplicacions per a dispositius mòbils amb Android, un dels mercats més emergents del moment i s'han complert les expectatives fixades a l'inici del projecte amb la construcció d'una maqueta funcional del client de la xarxa social WeDugu.



## 11. Referències.

- [1] <http://www.android-spa.com/>
- [2] <http://developer.android.com/reference/packages.html>
- [3] <http://developer.android.com/resources/faq/commontasks.html>
- [4] <http://es.wikipedia.org/wiki/Android>
- [5] [http://en.wikipedia.org/wiki/Android \(operating system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [6] <http://www.stackoverflow.com>
- [7] <http://www.sgoliver.net>
- [8] <http://www.javielinux.com/>
- [9] <http://www.androidsis.com/>
- [10] <http://www.xatakandroid.com/>