



**Universitat Autònoma  
de Barcelona**

**Disseny i desenvolupament  
d'una aplicació Android**

**"CercaJam"**

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Sistemes

realitzada per

Edgard Puig Orpinell

i dirigida per

Jordi Duran Cals

Escola Universitària d'Informàtica  
Sabadell, Setembre 2013

El sotasignat, **Jordi Duran i Cals**,  
professor de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present  
memòria ha estat realitzat sota la seva  
direcció per

**Edgard Puig Orpinell**

I per a que consti firma la  
present. Sabadell,  
Setembre de 2013

-----

----- Signat: **Jordi Duran i Cals**

El llenguatge és un procés de creació lliure, les seves lleis i principis són fixos, però la manera en què els principis de la generació s'utilitzen és lliure i infinitament variada. Fins i tot la interpretació i l'ús de paraules implica un procés de creació lliure.

*Noam Chomsky*

<b>1. Introducció .....</b>	<b>8</b>
1.1 Motivacions .....	8
1.2 Marc de treball .....	8
1.3 Objectius .....	9
1.4 Context del Projecte .....	9
1.5 Estructura de la memòria del projecte .....	10
<b>2. Estudi de Viabilitat .....</b>	<b>12</b>
2.1 Introducció .....	12
2.2. Descripció .....	12
2.2.1 Tipologia .....	12
2.2.2 Paraules Clau: .....	12
2.3 Objectius específics del projecte .....	13
2.4 Estudi de la situació actual .....	14
2.4.1 Context .....	14
2.4.2 Descripció física .....	15
2.5 Lògica del sistema .....	16
2.6 Requisits del projecte .....	16
2.6.1 Requisits funcionals. ....	16
2.6.2 Catalogació dels requisits funcionals .....	17
2.6.3 Requisits no funcionals. ....	17
2.6.4 Catalogació dels requisits no funcionals.....	17
2.7 Alternatives i selecció de la solució.....	18
2.7.1 Alternativa 1.....	18
2.7.2 Alternativa 2.....	18
2.7.3 Alternativa 3.....	19
2.7.4 Solució proposada .....	19
2.8 Planificació del projecte .....	19
2.9 Recursos del projecte .....	20
2.9.1 Recursos humans .....	20
2.9.2 Recursos materials .....	21
2.10 Dependències .....	22
2.11 Tasques del projecte.....	22

2.12 Planificació Temporal .....	25
2.13 Avaluació de riscos .....	25
2.13.1 Llista de riscos .....	25
2.13.2 Catalogació de riscos.....	26
2.13.3 Pla de contingència .....	27
2.14 Pressupost .....	27
2.14.1 Estimació cost de personal .....	27
2.14.2 Estimació cost dels recursos .....	28
2.14.3 Resum i Anàlisi .....	28
<b>3. Disseny tècnic i implementació del sistema .....</b>	<b>29</b>
3.1 Model de desenvolupament.....	29
3.2 Entorn de desenvolupament.....	30
3.3 Disseny i implementació del sistema .....	31
3.3.1 Disseny i implementació de la base de dades .....	31
3.3.2 Disseny i implementació del servei web.....	34
3.3.3 Disseny i implementació de l'aplicació Android .....	45
3.4 Diagrames del sistema.....	65
3.4.1 Diagrames de casos d'ús .....	65
3.4.2 Diagrames de seqüència.....	66
3.4.3 Diagrama d'activitats .....	68
<b>4. Test i proves.....</b>	<b>69</b>
4.1 Proves d'unitat de la base de dades .....	69
4.2 Proves d'unitat del servei web .....	70
4.3 Proves d'unitat de l'aplicació Android.....	71
4.4 Proves d'integració .....	72
<b>5. Conclusions .....</b>	<b>73</b>
5.1 Valoració d'objectius assolits.....	73
5.2 Futures línies de treball .....	74
5.2.1 Ampliacions del servei web .....	74
5.2.2 Ampliacions de l'interfície Android .....	75
5.3 Seguiment de la Planificació.....	76
5.4 Valoració personal.....	77

<b>6. Glossari</b> .....	<b>79</b>
6.1 Relació de taules .....	79
6.2 Relació d'imatges .....	79
<b>7. Bibliografia</b> .....	<b>81</b>
7.1 Documentació web .....	81
7.1 Llibres .....	82
7.2 Curs Android .....	82
<b>8. Agraïments</b> .....	<b>83</b>

## FULL DE RESUM-PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

<b>Títol del projecte:</b> CercaJam: Disseny i desenvolupament d'una aplicació Android	
<b>Autor:</b> Edgard Puig Orpinell	<b>Data:</b> Setembre 2013
<b>Tutor:</b> Jordi Duran i Cals	
<b>Titulació:</b> Enginyer Tècnic en Informàtica de Sistemes	
<b>Paraules clau:</b> <ul style="list-style-type: none"><li>• Java, Android, Geo-localització, Servei web i Base de dades.</li><li>• Java, Android, Geolocalización, Servicio web y Base de datos.</li><li>• Java, Android, Geolocation, Web Service, Databases.</li></ul>	
<b>Resum del projecte:</b> <ul style="list-style-type: none"><li>• El projecte vol desenvolupar un programari per <i>Android</i> destinat als usuaris que els interessin les jam sessions. El programa serà capaç de visualitzar les sales, cercar-les per nom o ciutat i afegir-ne una de nova. L'idea principal és que l'usuari pugui inscriure's, si queden places lliures, a la <i>jam session</i> de la sala. On també podrà ubicar-la, mitjançant l'API de Google Maps.</li><li>• El proyecto quiere desarrollar un <i>software</i> para <i>Android</i> destinado a los usuarios que les interesen las jam sessions. El <i>software</i> será capaz de visualizar las sales, realizar una búsqueda por nombre o ciudad y añadir una de nueva. La idea principal es que el usuario pueda inscribirse, si quedan plazas libres, a la <i>jam session</i> de la sala. Donde también podrán ubicarla, mediante la API de Google Maps.</li><li>• The project aims to develop software for Android for users that interest them jam sessions. The software will be able to view the jam session's clubs, search by city name or add a new one. The main idea is that the user can sign up, if there are available seats, the jam session of the club. Where you can also locate through the Google Maps API.</li></ul>	

# 1. Introducció

## 1.1 Motivacions

La major motivació que m'ha portat a realitzar aquest projecte va sorgir per la falta de recursos actuals en aquest àmbit. A l'hora de poder participar en diferents *jam sessions* de manera que puguin afegir-s'hi en línia. Fins ara, els músics anaven a les sales directament i/o parlaven amb el responsable o simplement com s'ha fet sempre parlar amb els propis músics i quan hi hagi un instrument lliure a la tarima o escenari, pujar a tocar.

Va ser llavors quan em vaig proposar el repte de solucionar aquesta necessitat realitzant una aplicació pròpia amb la que els usuaris(músics) poguessin participar a les *jam sessions*, conèixer noves sales, i sobretot interactuar amb nous participants d'una manera fàcil. A més a més, seria una bona oportunitat per a conèixer els llenguatges de programació per a mòbils.

## 1.2 Marc de treball

Una vegada definides les especificacions generals del projecte, s'han de marcar els objectius. Diferenciem les parts del projecte per a poder treballar amb més eficiència i poder integrar els diferents mòduls que el formen.

Es treballa amb un llenguatge de programació actual com és el Java i enfocar l'aplicació mòbil per cara a dispositius *Android*. També l'utilitzarem per al desenvolupament del servei web. Un dels altres llenguatges és *sql* l'utilitzarem amb la creació de la nostra base de dades i quan realitzem les *query* des de el servei web.

Una vegada planificat, es quan passem a desenvolupar plenament el projecte. A més, per a cada mòdul, es defineixen clarament les entrades i sortides amb els quals es treballaran.



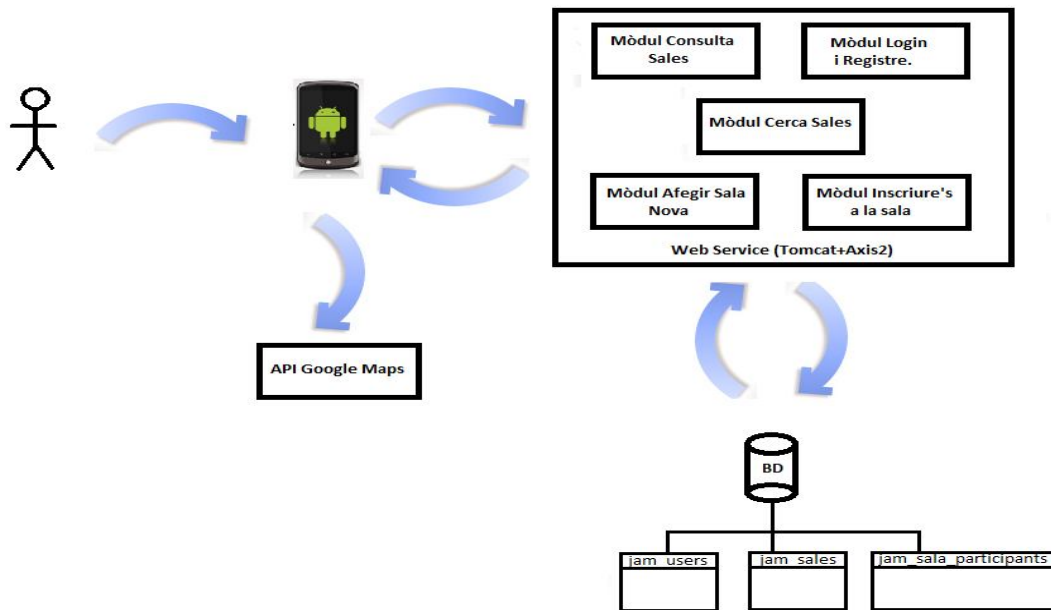
## 1.3 Objectius

Els objectius principals que pretén satisfer aquesta part del projecte són:

- Desenvolupar un servei capaç de visualitzar les respectives sales i que l'usuari pugui interactuar amb elles, amb la possibilitat d'inscriure's a les Jam Sessions.
- Desenvolupar un servei capaç d'inserir noves sales, les quals hauran de guardar la geo-localització actual on es troba l'usuari.
- Desenvolupar un servei capaç de cercar les sales per nom o població.
- Oferir una aplicació mòbil de geo-localització per a plataformes *Android*.
- Aprofundir en el coneixement del llenguatge Java.
- Aprofundir en el coneixement de l'entorn de desenvolupament *Eclipse*, la integració d'aquests amb altres mòduls i en el desenvolupament del servei web de l'aplicació.

## 1.4 Context del Projecte

Aquest projecte esta basat en el desenvolupament d'un servei web, la base de dades i la aplicació *Android*. A continuació es mostra una imatge on es representa el total del projecte i com s'han repartit els diferents mòduls que el formen.



**Imatge 1. Context del projecte.**

Els mòduls que es desenvolupen al llarg d'aquest projecte són:

-La base de dades.

-L'aplicació *Android*.

-Mòduls per a la comunicació amb la API Google Maps, per tal de mostrar la ubicació de les sales i la ubicació actual de l'usuari dins un mapa.

-El web service, el qual proporciona les funcionalitats de autenticació dels usuaris, el seu respectiu registre, retorna les dades de les respectives sales i la inserció de noves sales.

## 1.5 Estructura de la memòria del projecte

La present memòria està estructurada en vuit capítols:

**1. Introducció:** En el primer capítol es presenten les motivacions, es tracta el marc de treball, es descriuen els objectius que es volen assolir, el seu context el qual s'explica gràficament amb una imatge i finalment la estructura de la memòria.

**2. Estudi de viabilitat:** En el segon capítol s'explica l'estudi de viabilitat, fonamental per a saber si el projecte és viable realitzar-lo. Dins d'aquest estudi, podem trobar també les eines utilitzades, així com la planificació i el cost de la seva realització.

**3. Disseny Tècnic i Implementació del Sistema:** En el tercer capítol es detalla el mètode de desenvolupament utilitzat, en quin entorn es desenvolupa el projecte, es mostra la implementació del sistema mitjançant una imatge gràfica i finalment s'aprofundeix en el disseny i desenvolupament dels diferents mòduls del projecte.

**4. Fase de proves:** En el quart capítol, es documenten totes les proves realitzades per tal de verificar que l'aplicació funciona correctament.

**5. Conclusions:** En el cinquè capítol, es porten a terme aspectes del disseny de l'aplicació. Es comentarà l'arquitectura utilitzada i el nivell lògic de la base de dades utilitzada.

**6. Glossari:** En el sisè capítol s'exposa la relació de taules i imatges, on s'indicaran les pàgines on es poden trobar.

**7. Bibliografia:** En el setè capítol es llistaran els diferents direccions de pàgines web, llibres o cursos on també s'ha trobat informació necessària.

**8. Agraïments.**

## 2. Estudi de Viabilitat

### 2.1 Introducció

La intenció d'aquest apartat és la de plasmar un estudi de mercat en relació als productes actuals relacionats amb el que es vol realitzar així com de la seva possible acceptació per part dels potencials clients.

Abans de tirar endavant el projecte, s'han d'avaluar riscos i calcular els beneficis esperats per calibrar si l'esforç financer té expectatives de futur.

La nostra aplicació integra *Android* amb el servei web, el qual s'encarregarà gestionar les peticions dels usuaris *Android*, cap a la base de dades. El punt clau és proporcionar al mercat una eina capaç de respondre a una necessitat real.

### 2.2. Descripció

#### 2.2.1 Tipologia.

La tipologia d'aquest projecte té dues vessants:

- a. Desenvolupament de programari, ja que es tracta d'una aplicació que combina parts de desenvolupament Client/Servidor, desenvolupament de la base de dades, el servei web i el desenvolupament de l'aplicació *Android*.
- b. També de recerca, conté moltes de les eines de desenvolupament, les quals són desconegudes i és necessari tenir clars els seus fonaments i funcionament.

#### 2.2.2 Paraules Clau:

*Java, XML, Android, Geo-localització, SQL, Tomcat, Axis2, jam session.*

## 2.3 Objectius específics del projecte

Els objectius específics del projecte són que els usuaris de l'aplicació per a mòbils *Android* tinguin més facilitat alhora de cercar noves jam sessions, poder situar-les i sobretot poder-s'hi afegir. Per tant, poden obtenir una informació general entre d'altres.

- 1) Desenvolupar una aplicació mòbil per a sistemes *Android* utilitzant les crides a serveis web.
- 2) Desenvolupar un mòdul per a gestionar les sessions dels usuaris.
- 3) Desenvolupar un mòdul capaç de visualitzar les respectives sales i que l'usuari pugui veure la seva informació detallada i geo-localització, amb la possibilitat d'inscriure's a les Jam Sessions.
- 4) Desenvolupar un mòdul capaç d'inserir noves sales, les quals hauran de guardar la geo-localització actual on es troba l'usuari.
- 5) Desenvolupar un mòdul capaç de cercar les sales per nom o població.
- 6) Desenvolupar servei web que permeti generar peticions a la base de dades.
- 7) Aprofundir en el coneixement del llenguatge Java.
- 8) Aprofundir en el coneixement de l'entorn de desenvolupament *Eclipse*, la integració d'aquests amb altres mòduls i en el desenvolupament del servei web de l'aplicació.
- 9) Aprofundir en el coneixement del desenvolupament d'aplicacions mòbils.
- 10) Compte administrador per al manteniment de la base de dades.

Objectiu	Crític	Prioritari	Secundari
O.1	X		
O.2	X		
O.3	X		
O.4	X		
O.5	X		
O.6		X	

O.7			X
O.8			X
O.9			X
O.10	X		

Taula 1. Objectius específics

## 2.4 Estudi de la situació actual

### 2.4.1 Context

La revolució de les xarxes socials, ha donat caràcter a l'aplicació que es vol dur a terme, estan íntimament relacionats. El perfeccionament de l'accés a Internet sense cables (*wireless*) ha proliferat tota una col·lecció de dispositius portàtils capaços d'oferir a l'usuari accés a la xarxa des de pràcticament qualsevol punt.

A més, la continua connexió a les xarxes socials o a consultes d'informació de dades, provoca que la majoria de persones i, fins i tot sectors de la població que en principi s'havien quedat fora de "l'era tecnològica", disposin de dispositius mòbils tals com telèfons intel·ligents (*smartphone*) o tabletas amb connexió a Internet permanent.

Això provoca l'aparició de la que podríem considerar la vessant tecnològica en més auge de l'actualitat, el desenvolupament d'aplicacions mòbils. A partir d'aquí, tot un conjunt d'empreses i persones s'esforcen per:

1. Desenvolupar i donar a conèixer llenguatges específics per al desenvolupament d'aplicacions mòbils com es *l'Android Development Kit* en el cas de *Google Android*.
2. Estudiar la sintaxi i l'ús de les eines que aquests llenguatges proporcionen al desenvolupador d'aplicacions mòbils. És important remarcar que aquest aprenentatge és pràcticament autodidacta, a través de la xarxa i alguns llibres és on realment agafes informació.

Així doncs, ens trobem amb un mercat d'aplicacions mòbils, on qualsevol desenvolupador pot arribar a crear una aplicació i posar-la a la venda, per altre banda qualsevol usuari de la plataforma *Android* pot obtenir-la i utilitzar-la per un mòdic preu.

És, per tant, un repte doble ja que ens haurem de submergir en un entorn de desenvolupament que cada vegada està més establert en el mercat

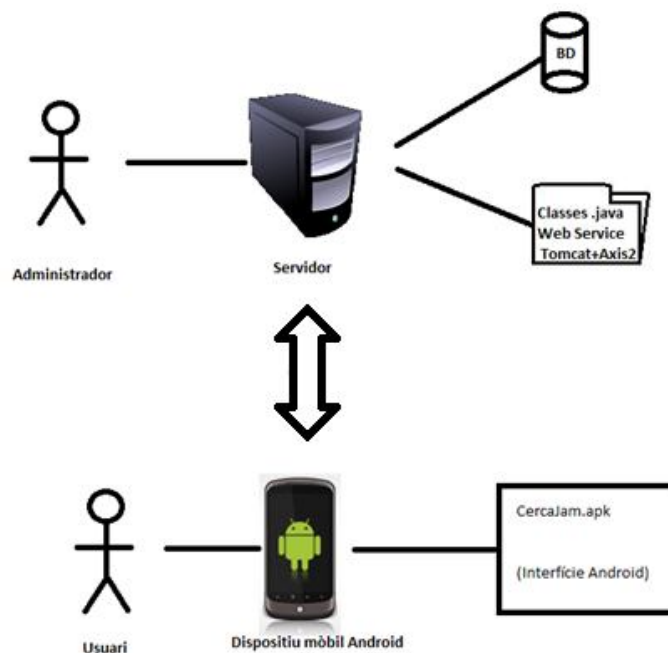
actual i haurem de ser capaços de crear un producte que ofereixi quelcom atractiu en un mercat de milers i milers d'aplicacions.

Fent un estudi sobre la situació actual podem observar que no hi ha masses aplicacions per no dir només una similar, parlant d'aplicacions per a mòbils amb sistemes *Android*. L'aplicació s'anomena "*Bandsintown*", i es tracta d'una aplicació on connectant amb les xarxes socials podem entrar a veure concerts a prop de la nostra ubicació, comprar entrades i també música en directe.

Trobem altres tipus d'aplicacions musicals, com també una aplicació anomenada "*Music Maker Jam*" per a poder crear la teva pròpia *jam session* amb diferents estils musicals i combinar els sons i compartir-la als teus amics.

### 2.4.2 Descripció física

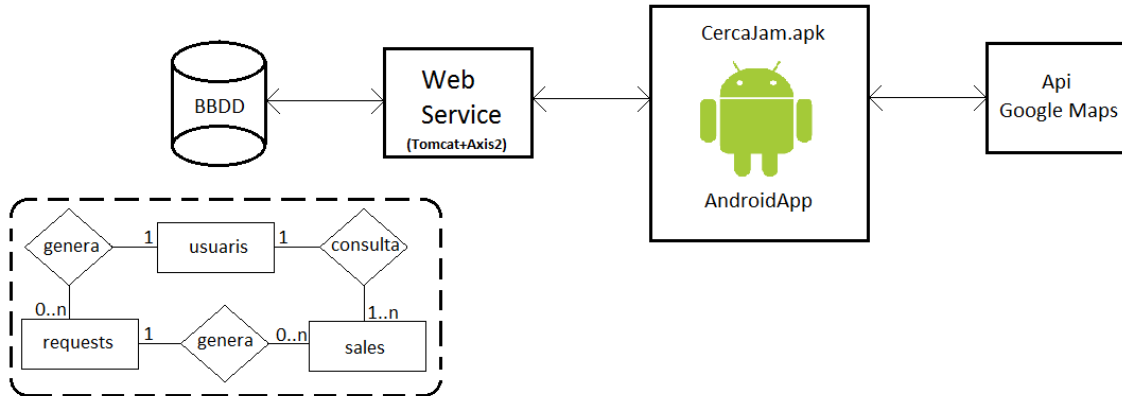
La descripció física del sistema es mostra a continuació:



Imatge 2. Descripció física del sistema

## 2.5 Lògica del sistema

La descripció física del sistema es mostra a continuació:



Imatge 3. Lògica del sistema

## 2.6 Requisits del projecte

### 2.6.1 Requisits funcionals.

1. Manteniment i gestió en la inserció de noves sales a la base de dades.
2. Manteniment i gestió en el registre de nous usuaris a la base de dades.
3. Manteniment, gestió i actualització quan els usuaris s'afegeixin a les jam sessions de les respectives sales de la base de dades.
4. Manteniment, gestió i actualització de les peticions dels usuaris a la base de dades.
5. Administració de la base de dades del sistema per part de l'administrador.
6. Control d'accés d'usuaris a l'aplicació.
7. L'aplicació ha de posicionar les sales geogràficament.
8. L'aplicació ha de permetre cercar sales via nom o població.



## 2.6.2 Catalogació dels requisits funcionals

Objectiu #	Essencial	Condicional	Opcional
1	X		
2	X		
3	X		
4		X	
5	X		
6			X
7	X		
8		X	

Taula 2. Catalogació dels requisits funcionals

## 2.6.3 Requisits no funcionals.

1. Compliment de la LOPD respecte a les dades dels usuaris.
2. El sistema ha de suportar diverses connexions d'usuaris en paral·lel.
3. Tolerància a errors.
4. La seguretat de les dades: encriptació de les dades
5. Compliment amb la normativa sobre l'estàndard XML.
6. Modular bé el codi font per a poder suportar futures ampliacions.

## 2.6.4 Catalogació dels requisits no funcionals

Objectiu #	Essencial	Condicional	Opcional
1		X	
2	X		
3		X	
4		X	
5		X	
6	X		

Taula 3. Catalogació dels requisits no funcionals

## 2.7 Alternatives i selecció de la solució

Per la manca d'aplicacions equivalents a la nostra. Faig referència de les diferents estratègies o alternatives que disposem per a desenvolupar l'aplicació.

### 2.7.1 Alternativa 1

Disseny i desenvolupament d'una aplicació per a mòbils *Android* i la creació d'un servei web. Cal a dir, que programar en *Android* no és gratuït, quan ens registrem com a desenvolupadors pagarem un únic mòdic preu de 25\$. A partir d'aquí ja podrem gestionar i pujar les nostres aplicacions al *Market d'Android*. És crearan funcionalitats totalment noves per adaptar-se a les propostes del client.

Aquesta solució ofereix funcionalitat total ja que assoleix tots els requisits i a més a més, ens ofereix la possibilitat d'ampliar el projecte en cas de canvi de decisió del client.

### 2.7.2 Alternativa 2

Disseny i desenvolupament d'una aplicació per a mòbils *Iphone* i la creació d'un servei web. Cal a dir, que programar en *Iphone* tampoc és gratuït, quan ens registrem com a desenvolupadors és un procés força laboriós i té un cost de 99\$ per any.

Ara bé, amb tot això no en tenim prou per a poder provar les nostres aplicacions a *I'iphone*, ja que necessitarem un certificat a través de la nostra conta per a poder fer-ho. També necessitem un certificat per a distribuir les nostres pròpies aplicacions. Quan l'any ha finalitzat tenim que tornar a pagar la quota de 99\$, en cas contrari, les aplicacions que tinguem penjades a l'*App Store* queden invalidades fins que realitzem el pagament. És crearan funcionalitats totalment noves per adaptar-se a les propostes del client.

Aquesta solució ofereix funcionalitat total ja que assoleix tots els requisits i a més a més, ens ofereix la possibilitat d'ampliar el projecte en cas de canvi de decisió del client.

### 2.7.3 Alternativa 3

La creació i desenvolupament d'una web. Per una banda, és molt ràpid d'implementar però no s'ajustarà a les necessitats reals demanades per l'aplicació.

Aquesta solució ens ofereix funcionalitat limitada ja que no compleix amb totes les expectatives.

### 2.7.4 Solució proposada

Veient les característiques de totes les alternatives, escollirem l'opció 1 (disseny i desenvolupament d'una aplicació per a mòbils *Android*) perquè sent la que millor s'adapta a les necessitats del client, també és l'opció més econòmica.

## 2.8 Planificació del projecte

La data de començament del projecte va quedar fixada el 5 de Desembre de 2012 i la data de finalització del mateix es planteja cap a finals de Setembre de 2013 una vegada finalitzada la presentació davant del tribunal és quan podrem considerar el cicle del projecte com acabat.

Malgrat que segons la primera planificació de desenvolupament del projecte conclouria el dia 24 de Setembre de 2013 però amb estimació d'aplaçament degut a imprevistos i/o aparició de riscos i una possible planificació massa optimista. Basant-nos en l'estimació actual la durada del projecte seria de 15 setmanes, comptem amb dos calendaris de treball diferents, un per a cada component de l'equip:

### Edgard Puig Orpinell (EPO)

1	Laborables	Festius
Dies	Dilluns a Divendres	Dissabte
Hores	12h-14h / 22h-23h	10-14h/20h-21h

Taula 4. Primer component de l'equip

Això representen:

15 setmanes \* 20 hores de dedicació setmanal = 300 hores.

### Jordi Duran Cals (JDC)

2	Laborables	Festius
Dies	Dilluns i Dimecres	Dissabte i Diumenge
Hores	19-21h	-

Taula 5. Segon component de l'equip

Això representen:

25 setmanes \* 2 hores = 50 hores.

## 2.9 Recursos del projecte

### 2.9.1 Recursos humans

Aquests són els recursos humans que intervindràn en el desenvolupament del projecte.

Nom	Correu electrònic	Càrrec	Responsabilitat	Calendari	Cost
Edgard Puig Orpinell (EPO)	<a href="mailto:edgardpuig@gmail.com">edgardpuig@gmail.com</a>	Analista Programador.	<ul style="list-style-type: none"> <li>- Anàlisis de requeriments</li> <li>- Desenvolupament del projecte.</li> <li>- Proves.</li> <li>- Documentació.</li> <li>- Implementació.</li> </ul>	1	20€/hora
Jordi Duran Cals (JDC)	<a href="mailto:jduran@deic.uab.cat">jduran@deic.uab.cat</a>	Project Manager	<ul style="list-style-type: none"> <li>- Supervisar el bon desenvolupament del projecte així com la seva viabilitat i finalització.</li> </ul>	2	40€/hora

Taula 6: Informació dels actors dins del desenvolupament del projecte.

## 2.9.2 Recursos materials

### Maquinari disponible:

Maquinari	Tipus	Característiques	Propietari	Disponibilitat	Cost	Càrrec pel projecte
Samsung Galaxy S II	Mòbil	<p><u>Processador:</u> ARM Cortex A9 dual core 1,2GHz</p> <p><u>Memòria RAM:</u> 1Gb</p> <p><u>Emmagatzematge:</u> <i>Intern:</i> 16GB-32GB <i>Extern:</i> MicroSD</p> <p><u>Imatge:</u> <i>Camara:</i> SI (3264x2448) / (1600x1200) <i>Pantalla:</i> 800x480 píxels, 16M colors SuperAmoled</p> <p><u>Connectivitat:</u> WiFi, Bluetooth, 3G.</p> <p><u>Sistema Operatiu:</u> Android O.S JellyBean 4.1.2</p>	EPO	Total	50€ (Cost al 2011)	0€
Sony Vaio SVE151G17M	Portàtil	<p><u>Processador:</u> Intel Core i5-3230M 2,60GHz (x2) 64 bits</p> <p><u>Memòria RAM:</u> 6GB DDR3 1600MHz</p> <p><u>Disc Dur:</u> 750GB SATA 5400 RPM.</p> <p><u>T. Gràfica:</u> AMD Radeon™ HD 7650M, 2 GB</p> <p><u>Connectivitat:</u> WiFi, Bluetooth.</p> <p><u>SO:</u> Windows 8</p>	EPO	Total	750€ (Cost al 2013)	0€

Taula 7: Relació i detalls del maquinari disponible.

## Programari disponible:

Programari	Llicència	Cost pel projecte
Microsoft Windows 8	Pagament.	0€*
Microsoft Office 2007 Pro amb Visio i Project	Pagament.	0€*
XAMPP	GNU Freeware	0€
Eclipse Indigo IDE per a desenvolupadors Java EE	GNU Freeware	0€
ADT (Android Development Tools)	GNU Freeware	0€
Tomcat	GNU Freeware	0€

**Taula 8: Programari disponible.**

\* En el nostre cas, ja comptàvem amb el software privatiu esmentat previ al projecte.

## 2.10 Dependències

Totes les fases es desenvolupen utilitzant un model lineal. Per tant, cada fase no comença fins que no s'ha completat la fase anterior.

En la fase de desenvolupament es preveu un model àgil de tal manera que el disseny, el desenvolupament i el test segueixin un model iteratiu.

En la fase de generació de documents es preveu al final perquè inclourà els documents elaborats durant el desenvolupament del projecte: inici, estudi de viabilitat, pla de projecte i finalment la memòria.

## 2.11 Tasques del projecte

Tasca		Subtasca		Relacions / Dependències		Recursos	Durada dins del projecte*
#	Descripció	#	Descripció	R	D		
1	<b>Fixar els objectius del projecte.</b>	1.1	Analitzar la necessitat del mercat actual i avançar- nos al client.			JDC, EPO	6h
		1.2	Estudiar el sector afectat per la necessitat.	1.1	1.1	EPO	6h
		1.3	Buscar solucions dins del mercat actual.	1.1		JDC EPO	8h
		1.4	Pluja d'idees.			JDC, EPO	3h
		1.5	Plantejar una proposta de producte per solucionar la manca.	1.4	1.3	JDC, EPO	7h

		1.6	Fixar objectius, extrems i límits.		1.5	JDC, EPO	4h
		1.7	Estudiar la possible acceptació del producte per part del client.	1.1 1.2 1.4 1.5	1.5 1.6	JDC EPO	3h
		1.8	Estudi de viabilitat.			JPC, EPO	14h
2	Fixar la base de recursos del projecte.	2.1	Marcar els suports que necessitem.		1.8	JDC, EPO	3h
		2.2	Anàlisi de sistema Operatiu; (cost, manteniment, compatibilitat...)		2.1	EPO	4h
		2.3	Programari requerit, el seu cost i adquisició.	2.2		EPO	17h
		2.4	Escollir i configurar servidor de serveis web.	2.2 2.3		EPO	17h
3	Disseny de l'aplicació.	3.1	Dissenyar el model relacional, diagrama entitat-relació.		3.1	EPO	4h
		3.2	Dissenyar la base de dades.	3.1	3.1	EPO	2h
		3.3	Disseny modular de l'aplicació.		3.2	EPO	5h
		3.4	Disseny de proves (test).	3.3		EPO	4h
		3.5	Documentació del disseny(Punt de Control).			JDC, EPO	10h
		3.6	Aprovació del disseny(Punt de Control).	3.5	3.5	JDC, EPO	10h
4	Desenvolupar.	4.1	Estudiar Java i XML per al desenvolupament d'aplicacions per a mòbils <i>Android</i> .			EPO	68h
		4.2	Preparació de l'entorn de desenvolupament.			EPO	3h
		4.3	Configurar la base de dades.		4.2		6h
		4.4	Mòdul d'adquisició de dades i funcionalitat de l'aplicació	4.3	4.2	EPO	6h
		4.5	Desenvolupament de la interfície de l'usuari		4.4	EPO	11h
		5.1	Proves unitàries			EPO	2h
		5.2	Proves d'integració	5.1		EPO	2h

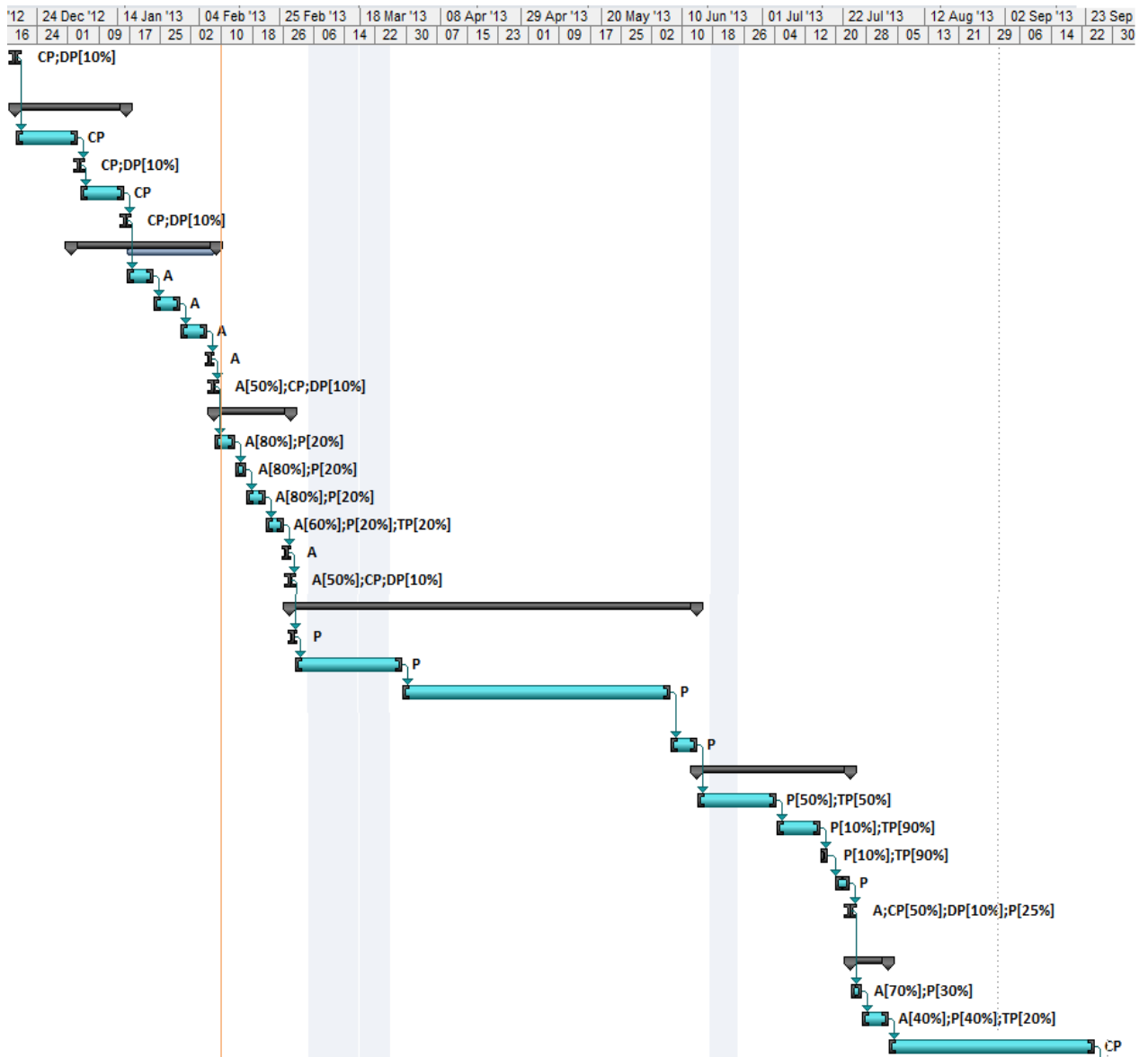
<b>5</b>	<b>Test i proves.</b>	5.3	Proves d'incidències i riscos	5.2 5.3	5.2 5.3	EPO	3h
		5.4	Documentació de desenvolupament i test			JDC EPO	2h
		5.5	Aprovació del desenvolupament i test (Punt de Control).	5.4	5.4	JDC EPO	2h
<b>6</b>	<b>Implantació.</b>	6.1	Instal·lació			EPO	2h
		6.2	Proves reals	6.1	6.1	EPO	3h
<b>7</b>	<b>Documentació de disseny, annexos i altres.</b>	7.1	Generació de documents (memòria del projecte)		1.5	EPO	60h
		7.2	Tancament del projecte		7.1	JDC EPO	1h
		7.3	Defensa del projecte		7.2	JDC	5h

**Taula 9: Tasques del projecte.**

*\*En aquesta taula hem aplicat els temps de durada de tasca segons la planificació. És a dir, tenint en compte les dependències entre tasques i les que es poden executar en paral·lel. Aleshores les hores es basen en els calendaris de treball de cada un dels actors, no en dies naturals.*



## 2.12 Planificació Temporal



Imatge 4: Planificació temporal del projecte.

## 2.13 Avaluació de riscos

### 2.13.1 Llista de riscos

- R1. Incompatibilitat entre dispositius mòbils amb diferent versió de OS Android.
- R2. Incompliment de seguretat.

- R3. Planificació optimista, no s'acaba en la data prevista, augmenten els recursos.
- R4. Manteniment de l'aplicació Android, actualització del servidor.
- R5. No es fa correctament la fase de test: desenvolupament, implantació. Manca de qualitat, insatisfacció dels usuaris, pèrdua econòmica.
- R6. Incompliment d'alguna norma, reglament o legislació: en qualsevol fase. No es compleixen els objectius, repercussions legals.
- R7. Pèrdua de totes les dades del projecte tals com el codi font, com les dades de la Base de Dades i la documentació. Tindríem que tornar a començar de nou.
- R8. Canvi de requisits: estudi de viabilitat, anàlisi. Endarreriment en el desenvolupament i resultat del projecte.
- R9. Abandonament del projecte abans de la finalització: en qualsevol fase. Pèrdues econòmiques, frustració.

### 2.13.2 Catalogació de riscos

	<b>Probabilitat</b>	<b>Impacte</b>
<b>R1</b>	Baixa	Crític
<b>R2</b>	Mitjana	Crític
<b>R3</b>	Mitjana	Crític
<b>R4</b>	Alta	Crític
<b>R5</b>	Alta	Crític
<b>R6</b>	Mitjana	Crític
<b>R7</b>	Baixa	Catastròfic
<b>R8</b>	Alta	Marginal
<b>R9</b>	Mitjana	Catastròfic

Taula 10: Catalogació dels riscos.

### 2.13.3 Pla de contingència

	<b>Solució</b>
<b>R1</b>	Desenvoluparíem una nova versió per a dispositius amb O.S <i>Android</i> anterior.
<b>R2</b>	Revisaríem el disseny per tal de garantir la seguretat a possibles atacs <i>hackers</i> i llei de protecció de dades.
<b>R3</b>	Modificació de la planificació del projecte.
<b>R4</b>	Tant el manteniment com l'actualització del servidor vindria a càrrec de l'Administrador.
<b>R5</b>	Dissenyar els test amb prou temps, realitzar tests automàtics, negociar contracte de manteniment, donar garanties, afrontar pèrdues econòmiques.
<b>R6</b>	Revisar les normes i legislació, consultar un expert, afrontar possibles repercussions penals.
<b>R7</b>	Fer còpies de seguretat periòdicament durant el desenvolupament del projecte.
<b>R8</b>	Tornar a negocia amb el client, ajornar funcionalitat, modificar planificació i pressupost.
<b>R9</b>	Sense solució.

Taula 11: Resultat d'aplicar les solucions als riscos.

## 2.14 Pressupost

### 2.14.1 Estimació cost de personal

	<b>Treball</b>	<b>Cost Total Treball</b>
<b>Cap de projecte</b>	80 h	2800 €
<b>Analista</b>	70 h	1750 €
<b>Programador</b>	130 h	2600 €
<b>Tècnic de proves</b>	20 h	200 €

Taula 12: Estimació cost de personal.

Cost Total = 7350 €

### 2.14.2 Estimació cost dels recursos

	<b>Cost amortització</b>	<b>Cost unitari</b>	<b>Període amortització</b>	<b>Període utilització</b>
<b>Amortització PC programador</b>	281,25 €	750 €	12 mesos	4,5 mesos
<b>Amortització mòbil programador</b>	6,25 €	50 €	12 mesos	1,5 mesos
<b>Amortització MSOffice</b>	40 €	120 €	12 mesos	4 mesos
<b>Amortització MSProject</b>	10 €	120 €	12 mesos	1 mes

**Taula 13: Estimació cost dels recursos.**

Cost Total: 337,50 €

### 2.14.3 Resum i Anàlisi

El cost del projecte tenint en compte tots els conceptes suma un total de:

- Desenvolupament del projecte: 7350 €
- Costos d'amortització de material: 337.50 €
- Altres costos: 220 €

Total: 7907.50 €

El cost total del projecte és elevat

## 3. Disseny tècnic i implementació del sistema

### 3.1 Model de desenvolupament

S'ha utilitzat un model de desenvolupament en cascada. Aquesta metodologia ordena les etapes del procés de manera que el inici de cada etapa ha d'esperar a la finalització de l'etapa anterior. Dins d'aquest projecte, les etapes s'han seguit han estat:

- Anàlisi de requeriments: en aquesta etapa s'analitzen les necessitats dels usuaris per tal de determinar els objectius a assolir. Aquesta etapa és de les més importants, ja que és aquí on s'ha de consensuar tot el que necessita el sistema ja que les següents etapes seguiran aquestes especificacions.
- Anàlisi del sistema: en aquest punt és on s'ha decidit i determinat quina seria la funcionalitat de cada mòdul.
- Disseny del sistema: s'han repartit els diferents mòduls del sistema entre els diferents membres del grup.
- Disseny del programa: en aquesta etapa és on s'han raonat i determinat els diferents algoritmes i les eines necessàries per a començar amb la codificació del sistema
- Codificació: és la part on s'implementa el codi font així com proves per tal de determinar possibles errors o corregir imperfeccions.
- Proves: els diferents mòduls desenvolupats s'uneixen per tal de compondre el sistema complert i verificar que tot funciona a la perfecció.
- Manteniment: finalment en aquesta etapa, es poden afegir funcionalitats extres no definides a la primera fase, de manera que es millori el sistema. També es poden detectar i corregir imperfeccions del sistema total.

D'aquesta manera, tots els errors que hagin pogut sorgir, s'han anat corregint pràcticament a la vegada.

Per una altra banda, s'ha desenvolupat un document anomenat *memòria de projecte* on es recullen totes les fases per les quals ha passat el sistema, des de que sorgeix com a idea fins que finalitza com a aplicació funcional.

### 3.2 Entorn de desenvolupament

El llenguatge de programació escollit es Java i el seu *framework* de desenvolupament es *Eclipse*. Aquest llenguatge de programació és un dels més senzills, ensenyats i utilitzat internacionalment. A més, l'*Eclipse* és un entorn de desenvolupament multi-plataforma lliure i que, per tant, està a disposició pública. Tot i que esta per a plataformes Linux, Mac OS o Windows que és el sistema operatiu que he escollit.

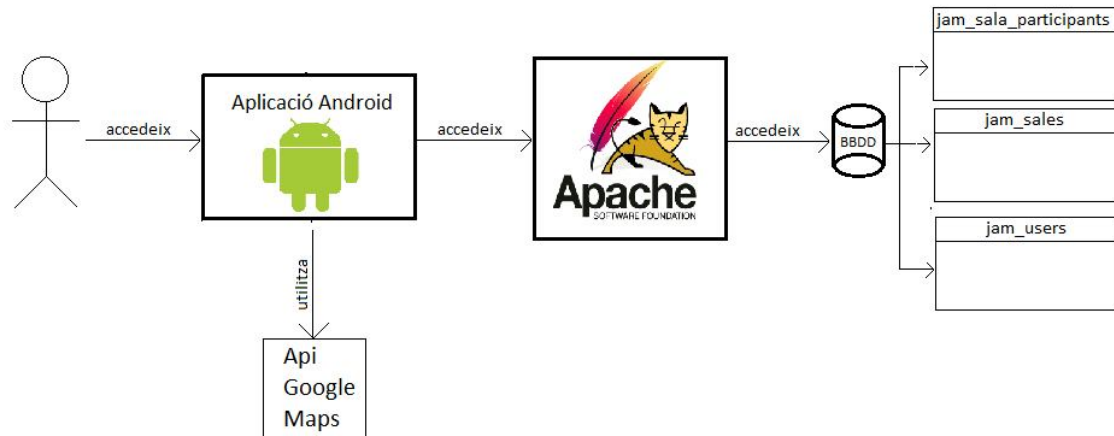
L'*Eclipse* per la seva part, potencia aquesta gran capacitat d'integració del llenguatge Java, fent possible l' utilització de milers de funcionalitats extra o complementàries a allò que s'està programant. Es val de llibreries de tipus .jar ( *Java Archive*) o fins i tot classes externes i internes que l'usuari pot importar de forma molt ràpida i fàcil.

El llenguatge de programació Java està orientat a objectes. La característica principal d'aquest llenguatge és implementar un programari de manera que les diferents tipus de dades que s'utilitzin estiguin units a les seves operacions. Així, les dades i el codi (funcions i mètodes) es combinen en entitats anomenades objectes. Un objecte pot veure's com un paquet que conté el comportament (el codi) u l'estat (dades).

Moltes vegades, canviar una estructura de dades condueix a fer un canvi en el codi que intervé sobre el mateix. La separació entre objectes coherents i independents fa més estable el disseny d'un sistema.

Per tant, es podria dir que l'objectiu es fer que grans projectes siguin fàcils de gestionar millorant també la seva qualitat i reduint el número de projecte no resolts.

### 3.3 Disseny i implementació del sistema



**Imatge 5. Implementació del sistema.**

Aquest sistema es pot extreure els següents punts:

- L'usuari pot fer login, o registrar-se com a usuari nou si encara no ho havia fet a través d'un *smartphone*.
- Pot consultar les diferents sales on es realitzin les *jam session's*.
- Pot afegir-s'hi a la sala per a participar en la *jam session*.
- Pot cercar les diferents sales per nom o ciutat que hi ha a la base de dades.
- Pot afegir noves sales a la base de dades.

#### 3.3.1 Disseny i implementació de la base de dades

He treballat amb el servidor XAMPP ja que resulta molt còmode, ràpid, lleuger i adaptable treballar amb MySQL. Funciona en varies plataformes, però la que escollirem és Windows 8. El subministrador de bases de dades escollit és el MySQL i el seu gestor principal via web *phpMyAdmin*.

A continuació faré una breu explicació del contingut de la base de dades "CercaJam", la qual conté tres taules: *jam\_users*, *jam\_sales*, *jam\_sala\_participants*.

La taula *jam\_users* es l'encarregada d'emmagatzemar la informació de l'usuari i l'estat, amb el qual sabrem si ha fet log in. També s'utilitza per a realitzar diferents tipus de control:

- *login*: quan un usuari vulgui fer login si ja està loguejat.
- *registre*: quan un usuari vulgui afegir un usuari que ja ha estat afegit a la taula.

Fent referència al contingut de la taula *jam\_users*:

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	1 <b>usuari_id</b>	int(200)			No	Ninguna	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>username</b>	varchar(20)	latin1_spanish_ci		No	Ninguna	
<input type="checkbox"/>	3 <b>password</b>	varchar(20)	latin1_spanish_ci		No	Ninguna	
<input type="checkbox"/>	4 <b>password2</b>	varchar(20)	latin1_spanish_ci		No	Ninguna	
<input type="checkbox"/>	5 <b>mail</b>	varchar(200)	latin1_spanish_ci		No	Ninguna	
<input type="checkbox"/>	6 <b>telefon</b>	varchar(15)	latin1_spanish_ci		No	Ninguna	
<input type="checkbox"/>	7 <b>loguejat</b>	tinyint(1)			No	Ninguna	

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar  Eliminar	<b>PRIMARY</b>	BTREE	Sí	No	usuari_id	2	A		
Editar  Eliminar	<b>USER_IDX</b>	BTREE	Sí	No	username	2	A		

**Imatge 6.** Taula *jam\_users* de la base de dades Cercajam.

L'ús de cada camp és el següent:

- *usuari\_id*: és un índex auto-incremental i la clau primària.
- *username*: cadena que emmagatzema el nom de l'usuari.
- *password*: cadena que emmagatzema la contrasenya de l'usuari.
- *password2*: cadena que emmagatzema la confirmació de la contrasenya de l'usuari.
- *mail*: cadena que emmagatzema el correu electrònic de l'usuari.
- *telefon*: cadena que emmagatzema el telèfon de l'usuari.
- *loguejat*: és un booleà que emmagatzema l'estat de l'usuari.



La taula *jam\_sales* es l'encarregada d'emmagatzemar la informació de les sales o locals. S'utilitza per a realitzar una consulta i cerca de les sales. També es realitzen tasques de control:

- *afegir\_sala*: quan un usuari vulgui afegir una sala nova que ja ha estat afegida a la taula.



Fent referència al contingut de la taula *jam\_sales*:

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	1 <b>sal_id</b>	int(4)			No	Ninguna	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>sal_nom</b>	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	3 <b>sal_desc</b>	varchar(500)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	4 <b>sal_provincia</b>	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	5 <b>sal_ciutat</b>	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	6 <b>sal_adreca</b>	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	7 <b>sal_categoria</b>	varchar(50)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	8 <b>sal_capacitat</b>	int(4)			No	Ninguna	
<input type="checkbox"/>	9 <b>sal_latitud</b>	double(14,12)			No	Ninguna	
<input type="checkbox"/>	10 <b>sal_longitud</b>	double(9,8)			No	Ninguna	

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
 Editar  Eliminar	PRIMARY	BTREE	Sí	No	sal_id	7	A		

**Imatge 7. Taula *jam\_sales* de la base de dades Cercajam.**

L'ús de cada camp és el següent:

- *sal\_id*: és un índex auto-incremental i la clau primària.
- *sal\_nom*: cadena que emmagatzema el nom de la sala.
- *sal\_desc*: cadena que emmagatzema la descripció de la sala.
- *sal\_provincia*: cadena que emmagatzema la província on es troba la sala.
- *sal\_ciutat*: cadena que emmagatzema la ciutat on es troba la sala.
- *sal\_categoria*: cadena que emmagatzema la categoria musical de la sala.
- *sal\_capacitat*: és de tipus enter el qual emmagatzema la capacitat total de participants a la sala.
- *sal\_longitud*: és d tipus double el qual emmagatzema la latitud on es troba la sala.
- *sal\_latitud*: és d tipus double el qual emmagatzema la longitud on es troba la sala.

La taula *jam\_sala\_participants* es l'encarregada d'emmagatzemar l'identificador de la sala i del usuari. També realitzem tasques de control:

- *afegir-se\_a\_sala*: que un usuari no es pugui afegir a una sala mes d'una vegada.

Fent referència al contingut de la taula *jam\_sala\_participants*:

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	1 <u>sal_id</u>	int(50)			No	Ninguna	
<input type="checkbox"/>	2 <u>usuari_id</u>	int(50)			No	Ninguna	

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar  Eliminar	PRIMARY	BTREE	Sí	No	sal_id	9	A		
					usuari_id	9	A		
Editar  Eliminar	usuari_id	BTREE	No	No	usuari_id	4	A		

**Imatge 8. Taula *jam\_sala\_participants* de la base de dades CercaJam.**

Per a poder obtenir el resultat, farem ús de claus foranies.

L'ús de cada camp és el següent:

- *sal\_id*: és de tipus enter i té clau forania del camp *sal\_id* que és clau primària de la taula *jam\_sales*.
- *usuari\_id*: és de tipus enter i té clau forania del camp *usuari\_id* que és clau primària de la taula *jam\_users*.

### 3.3.2 Disseny i implementació del servei web

El mòdul d'implementació d'un servei web és un dels grans reptes a implementar del projecte. La funcionalitat d'aquest servei web es ajudar-nos a interactuar amb la interfície Android i la base de dades. És el cor de l'aplicació perquè és el nexce comú per a tots els usuaris de l'aplicació.

Per tant, serà el que realment ens gestionarà totes les peticions que faran els usuaris, amb el seus dispositius mòbils amb sistema *Android*, quan vulguem consultar o inserir dades a la base de dades. Les eines utilitzades per a la creació del servei web són el projecte web dinàmic (*Dynamic Web Project*) que ofereix *Eclipse Indigo* i el plugin de *Tomcat* amb *Axis2*.

Per tal d'aconseguir tot això farem ús de *JDBC*, és una interfície que permet a un programa *Java* executar instruccions *SQL* dintre de bases de dades relacionals. Es presenta en un controlador de tipus *.jar*, per exemple per a *MySQL* el que hem utilitzat és "mysql-connector-java-5.1.24-bin.jar".

Dins d'aquest projecte tindrem diferents *packages*, per tal d'organitzar les diferents classes.

El primer *package* que ens trobem és *com.cercajam.constants*.

La seva classe és:

-Constants.class: la funció d'aquesta classe és d'organitzar millor el codi i declarar totes les constants utilitzades en el projecte dins una classe.

Els atributs associats són un conjunt de *strings* que tenen emmagatzemats un conjunt de missatges:

- *LOGIN\_INFO\_LOGAT*
- *LOGIN\_ERROR\_DADES*
- *LOGIN\_OK*
- *AFEGIRSE\_SALA\_OK*

En aquesta classe no tenim cap mètode ja que només conté constants.

En el segon *package* que ens trobem és *com.cercajam.login*.

Les seves classes són:

- Login.class: la funció d'aquesta classe és de validar amb la base de dades si el nom d'usuari i la contrasenya que ha introduït l'usuari existeixen i són correctes.

Els atributs associats són:

- *retrievedUserName*: és una variable de tipus string on emmagatzema el nom de l'usuari rebut de la base de dades.
- *retrievedPassword*: és una variable de tipus string on emmagatzema la contrasenya associada a l'usuari.
- *retrievedID*: és una variable de tipus enter on emmagatzema la ID de l'usuari rebut de la base de dades.
- *retrievedLoguejat*: és una variable de tipus enter on emmagatzema l'estat de login de l'usuari.
- *status*: és una variable de tipus string on emmagatzema la constants, que hem esmentat més a dalt, segons si el login és correcte o no.

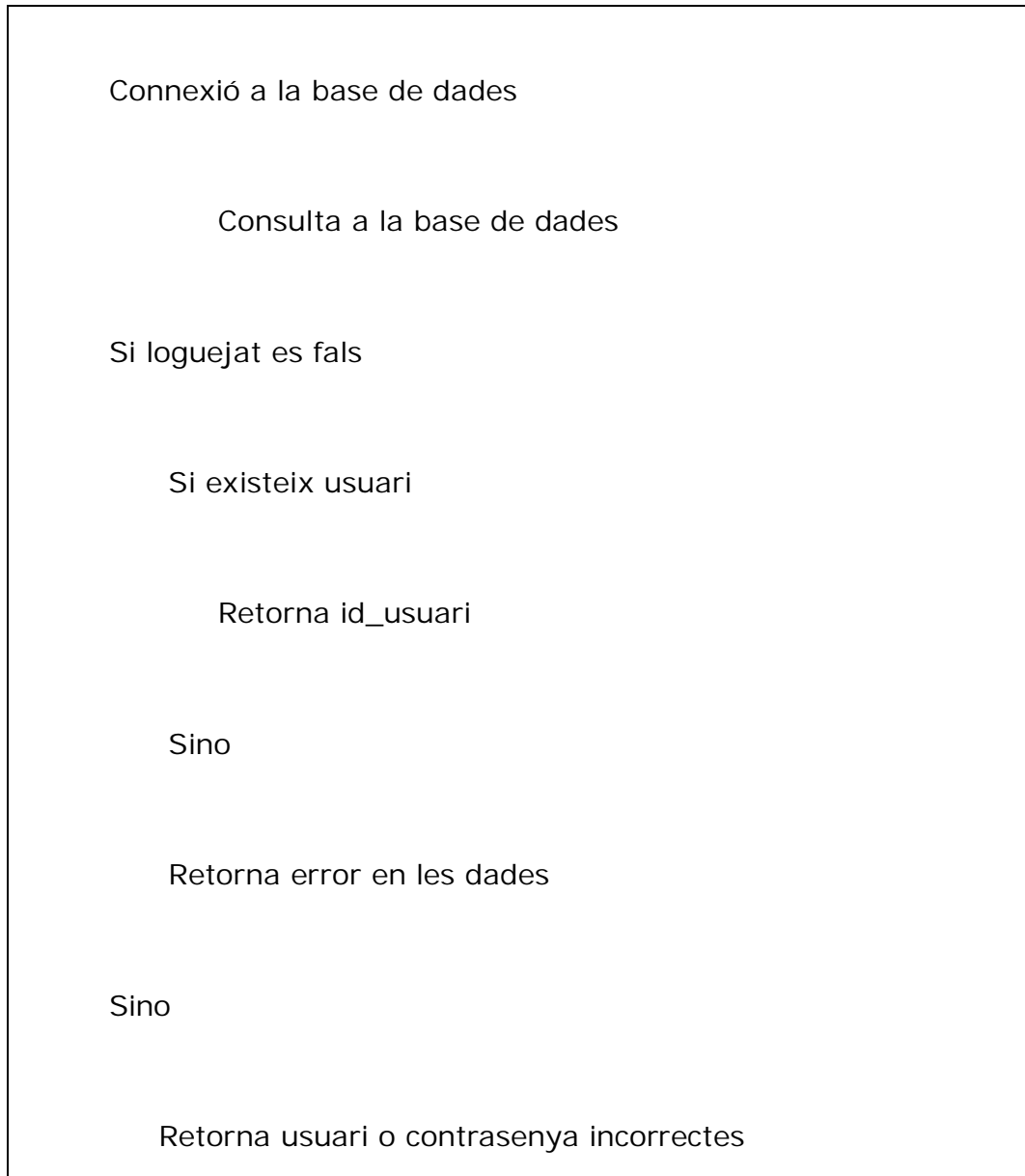
Els mètodes associats a aquesta classe són:

- *authentication(String userName, String password)*: és una funció que connecta a la taula *jam\_users* de la base de dades amb la següent *query*:

```
("SELECT * FROM jam_users WHERE username =  
"+username+"");
```

Comprovem que el nom d'usuari i el password són iguals, si és satisfactori retorna la ID de l'usuari, sino retorna una cadena amb una de les constants que hem esmentat abans.

El pseudocodi d'aquesta funció és:



- ControlLogin.class: la funció d'aquesta classe és de controlar el login amb la base de dades si el nom d'usuari i la contrasenya que ha introduït l'usuari existeixen i són correctes.

Els atributs associats són:

- *loguejatBool*: és una variable de tipus string on emmagatzema fals si l'usuari no està loguejat, en cas contrari retornarà cert.

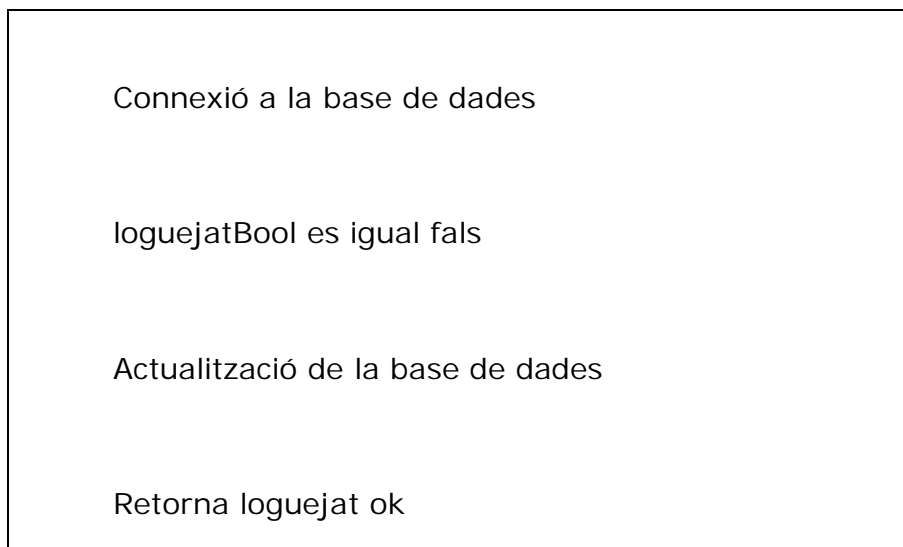
Els mètodes associats a aquesta classe són:

- *sessio(String usuari\_id, String loguejat)*: és una funció que es connecta i actualitza l'atribut loguejat a cert de la taula *jam\_users* amb la següent query:

```
("UPDATE jam_users SET loguejat = "+loguejatBool+"WHERE  
ususari_id = "+usuari_id);
```

Retorna una cadena amb una de les constants que hem esmentat abans.

El pseudocodi d'aquesta funció és:



En el tercer *package* que ens trobem és *com.cercajam.registre*.

Les seves classes són:

- *Registre.class*: la funció d'aquesta classe és de afegir un usuari nou i les seves dades a la base de dades.

Els atributs associats són:

- *retrievedUserName*: és una variable de tipus string on emmagatzema el nom de l'usuari rebut de la base dades.

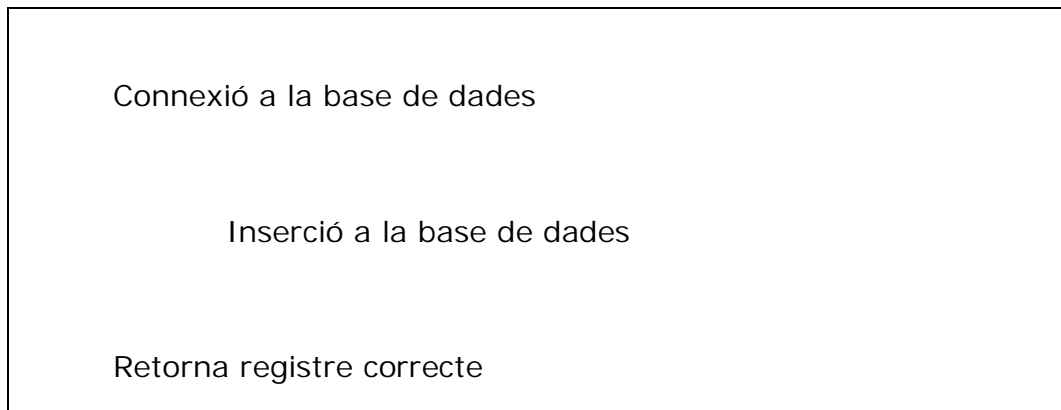
Els mètodes associats a aquesta classe són:

- *insertData(String userName, String userPassword, String userPassword2, String userMail, String userTelefon)*: és una funció que es connecta i afegeix a la taula *jam\_users* de la base de dades amb la següent *query*:

```
("INSERT INTO jam_users (username,password,password2,mail,telefon) VALUES ("'+userName+'","'+userPassword+'",'+userPassword2+'","'+userMail+'","'+userTelefon+'");");
```

Retorna una cadena amb una de les constants que hem esmentat abans.

El pseudocodi d'aquesta funció és:



En el tercer *package* que ens trobem és *com.cercajam.sales*.

Les seves classes són:

- *GestioSales.class*: la funció d'aquesta classe és consultar les sales que hi ha a la base de dades i retornar una cadena amb totes les sales.

Els atributs associats són:

- *llistaSales*: és una variable de tipus string on emmagatzema cada una de les sales separades per "&&" i el final de cada sala és "\_\_" de la base dades.
- *result*: és un objecte que s'utilitza per a la connexió amb la base de dades.

Els mètodes associats a aquesta classe són:

- `cercaSales()`: és una acció que connecta i consulta a la taula `jam_sales` de la base de dades amb la següent *query*:

```
("SELECT * FROM jam_sales);
```

Retorna la cadena `llistaSales`.

El pseudocodi d'aquesta funció és:

```
Connexió a la base de dades
    Consulta a la base de dades
Mentres existeixin sales
    Afegir sala_id a llistaSales
    Afegir nom_sala a llistaSales
    Afegir desc_sala a llistaSales
    Afegir provincia_sala a llistaSales
    Afegir ciutat_sala a llistaSales
    Afegir adreça_sala a llistaSales
    Afegir categoria_sala a llistaSales
    Afegir capacitat_sala a llistaSales
    Afegir latitud_sala a llistaSales
    Afegir longitudm_sala a llistaSales
Fi Mentres
    Retorna llistaSales
```

- `Sala.class`: aquesta classe conté tots els *getters* i *setters* de cada un dels atributs de la sala.

Els atributs associats són:

- `id`: és una variable de tipus enter on emmagatzema la id de la sala.

- *nom*: és una variable de tipus string on emmagatzema el nom de la sala.
- *desc*: és una variable de tipus string on emmagatzema la descripció de la sala.
- *provincia*: és una variable de tipus string on emmagatzema la província on es troba la sala.
- *ciutat*: és una variable de tipus string on emmagatzema la ciutat on es troba la sala.
- *adreca*: és una variable de tipus string on emmagatzema l'adreça on es troba la sala.
- *categoria*: és una variable de tipus string on emmagatzema el tipus de categoria musical de la sala.
- *capacitat*: és una variable de tipus enter on emmagatzema la capacitat total de participants de la sala.
- *latitud*: és una variable de tipus double on emmagatzema la latitud de la sala.
- *longitud*: és una variable de tipus double on emmagatzema la longitud de la sala.

Els mètodes associats a aquesta classe són:

- *getId()*: és una funció que retorna la id.
- *setId(int Id)*:
- *getNom()*: és una funció que retorna el nom.
- *setNom(String Nom)*:
- *getDesc()*: és una funció que retorna la descripció.
- *setDesc(String Desc)*:
- *getProvíncia()*: és una funció que retorna la província.
- *setProvíncia(String Província)*:



- *getAdreca()*: és una funció que retorna la adreça.
  - *setAdreca(String Adreca)*:
  - *getCiutat()*: és una funció que retorna la ciutat.
  - *setCiutat(String Ciutat)*:
  - *getCategoria()*: és una funció que retorna la id.
  - *setCategoria(String Categoria)*:
  - *getCapacitat()*: és una funció que retorna la id.
  - *setCapacitat(int Capacitat)*:
  - *getLatitud()*: és una funció que retorna la id.
  - *setLatitud(double latitud)*:
  - *getLongitud()*: és una funció que retorna la id.
  - *setLongitud(double longitud)*:
- NovaSala.class: la funció d'aquesta classe és afegir una nova sala.

Els atributs associats són:

- *capacitat*: és una variable de tipus enter i emmagatzemarà el valor de parsejar *sal\_capacitat*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base dades un valor de tipus enter.
- *longitud*: és una variable de tipus double i emmagatzemarà el valor de parsejar *sal\_longitud*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base dades un valor de tipus double.
- *latitud*: és una variable de tipus enter i emmagatzemarà el valor de parsejar *sal\_latitud*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base dades un valor de tipus double.
- *result*: és un objecte que s'utilitza per a la connexió amb la base de dades.

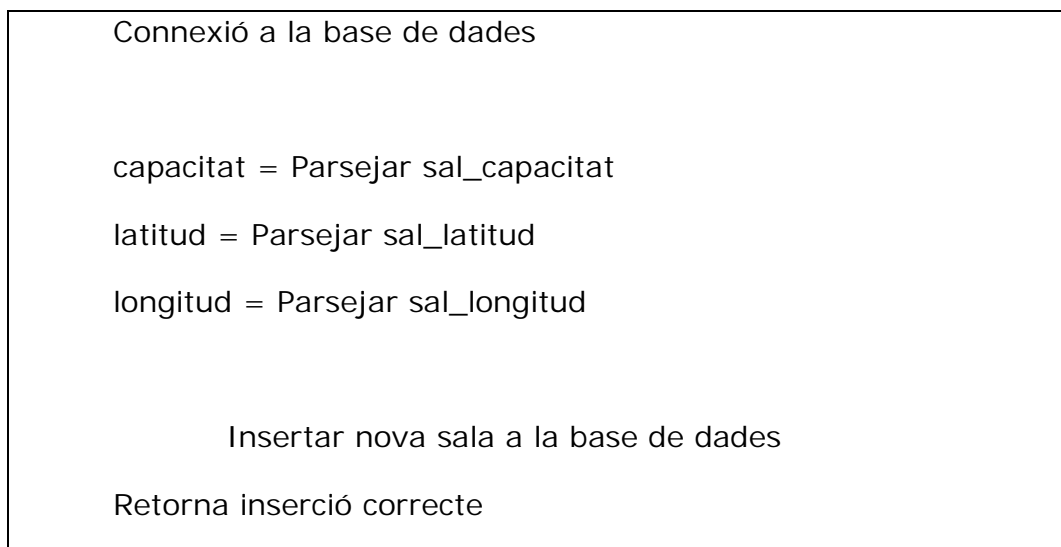
Els mètodes associats a aquesta classe són:

- *insertaSala(String sal\_nom, String sal\_desc, String sal\_provincia, String sal\_ciutat, String sal\_adreca, String sal\_categoria, String sal\_capacitat, String sal\_latitud, String sal\_longitud)*: és una funció que connecta i afegeix una nova sala a la taula *jam\_sales* de la base de dades amb la següent *query*:

```
("INSERT INTO jam_sales (sal_nom,sal_desc,sal_provincia,  
sal_ciutat,sal_adreca,sal_categoria,sal_capacitat,sal_longitud,  
sal_latitud) VALUES (" + sal_nom + "," + sal_desc + "  
" + sal_pronvincia + "," + sal_ciutat + "," + sal_adreca + "  
" + sal_categoria + "," + capacitat + "," + latitud + "  
" + longitud + "");");
```

Retorna una cadena amb una de les constants que hem esmentat abans.

El pseudocodi d'aquesta funció és:



- *AfegirseSala.class*: la funció d'aquesta classe és afegir l'usuari a la *jam session* de la sala.

Els atributs associats són:

- *sala\_id*: és una variable de tipus enter i emmagatzemarà el valor de parsejar *sal\_id*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base dades un valor de tipus enter.

- *user\_id*: és una variable de tipus double i emmagatzemarà el valor de parsejar *usuari\_id*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base de dades un valor de tipus enter.
- *result*: és un objecte que s'utilitza per a la connexió amb la base de dades.

Els mètodes associats a aquesta classe són:

- *inscriuresSala(String sal\_id, String usuari\_id)*: és una funció que connecta i afegeix el id de l'usuari i el id de la sala a la taula *jam\_sala\_participants* de la base de dades amb la següent *query*:

```
("INSERT INTO jam_sales (sal_id, usuari_id) VALUES  
("+sala_id+"",""+user_id+"");");
```

Retorna una cadena amb una de les constants que hem esmentat abans.

El pseudocodi d'aquesta funció és:

```
Connexió a la base de dades

sala_id = Parsejar sal_id
usuari_id = Parsejar usuari_id

Insertar sala_id i usuari_id a la base de dades

Retorna inserció correcte
```

- *ControlAfegirse.class*: la funció d'aquesta classe és controlar si un usuari ja s'havia afegit abans a la sala.

Els atributs associats són:

- *llistaSales*: és un *ArrayList*, és una array dinàmica que no té restriccions de capacitat i el seu tamany s'ajusta de manera dinàmica.

- *sala\_id*: és una variable de tipus enter i emmagatzemarà el valor de parsejar *sal\_id*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base dades un valor de tipus enter.
- *user\_id*: és una variable de tipus double i emmagatzemarà el valor de parsejar *usuari\_id*. Això ho realitzem perquè li arriba una cadena i hem d'enviar a la base dades un valor de tipus enter.
- *result\_nom*: és una variable de tipus booleà que emmagatzema cert si existeix el nom d'usuari a l'Arraylist *llistaSales*.
- *result\_iduser*: és una variable de tipus booleà que emmagatzema cert si existeix el id d'usuari a l'Arraylist *llistaSales*.
- *status*: és una variable de tipus string que s'utilitza per a emmagatzemar les constants que hem esmentat abans.

Els mètodes associats a aquesta classe són:

- *controlUsuari(String sal\_id, String usuari\_id, String userName)*: és una funció que connecta i consulta a la taula *jam\_sala\_participants* la qual té clau forània amb la clau primària de la taula *jam\_users* de la base de dades amb la següent *query*:

```
("SELECT usuari_id,username FROM jam_users WHERE usuari_id IN (SELECT usuari_id FROM jam_sala_participants WHERE sala_id = "+sala_id+"");
```

Retorna una cadena *status*.

El pseudocodi d'aquesta funció és:

Connexió a la base de dades

*user\_id* = Parsejar *usuari\_id*

*sala\_id* = Parsejar *sal\_id*

```
Mentres existeixi usuari_id i sala_id a sales
    llistaSales = nom_usuari
    llistaSales = id_usuari
Fi Mentre

Retorna inserció correcte
```

### 3.3.3 Disseny i implementació de l'aplicació Android

Per fer el disseny de la interfície d'usuari s'ha creat un nou projecte de tipus *Android Application* a l'entorn de desenvolupament *Eclipse*. Aquest projecte consta de diferents classes, les quals són el motor de l'aplicació. S'encarreguen del control i funcionament de la nostra aplicació. Realitzarem la connexió amb el nostre servei web fent ús del estàndard *SOAP*.

En primer lloc, s'ha de fer referència que el sistema *Android* no inclou "de sèrie" cap tipus de suport per a serveis web de tipus *SOAP*. És per això, que farem ús de la llibreria *ksoap2-android-assembly-2.5.8-jar-with-dependencies.jar*. Aquesta llibreria és un *fork*, especialment adaptat per a *Android*, de l'antiga llibreria *kSOAP2*. Aquest *framework* ens permetrà de forma relativament fàcil i còmoda utilitzar els serveis web que utilitzin *SOAP*.

Afegir aquesta llibreria al nostre projecte *Android* és molt senzill. Si tenim una versió recent del plugin d'*Android* per a *Eclipse*, copiarem l'arxiu *.jar* a la carpeta *libs* del projecte. Si la versió del plugin és més antiga, hi accedirem mitjançant el menú "*Project/Properties*" i a la finestra de propietats accedirem a la secció "*Java Build Path*". En aquesta secció, accedirem a la pestanya "*Libraries*" y polsarem el botó "*Add External JARs...*". Aquí seleccionem el fitxer *.jar* i confirmem.

Amb això ja tindriem el nostre projecte preparat per a fer ús de la funcionalitat aportada per la llibreria.

Per a poder realitzar la connexió amb el servei web mitjançant les classes creades al nostre projecte primer de tot havíem de declarar les seves constants:

- **URL**: direcció URL per a realitzar la connexió amb el nostre servei web.
  - **NAME\_SPACE**: espai de noms utilitzat en el nostre servei web.
  - **METHOD\_NAME**: nom del mètode web concret que anem a executar.
  - **SOAP\_ACTION**: Equivalent a l'anterior, el contingut de *NAMESPACE/METHOD\_NAME*.
1. creem un nou objecte *SoapObject* passant-li el *NAMESPACE* i el *METHOD\_NAME*. A aquesta petició tindrem que associar els paràmetres d'entrada mitjançant el mètode *addProperty()* al que passarem els noms i els valors dels paràmetres.
  2. El segon pas serà crear el contenidor SOAP (*envelope*) i associar-li la nostra petició. Per a fer-ho, crearem un nou objecte *SoapSerializationEnvelope* indicant la versió SOAP que anem a utilitzar ( hem utilitzat la versió *1.1*). Per últim, associarem la petició abans creada al nostre contenidor fent la crida al mètode *setOutputSoapObject()*.
  3. Com a tercer pas crearem l'objecte que s'encarregarà de realitzar la comunicació, de tipus *HttpTransportSE*, a la que passarem la URL de connexió al nostre servei web. Per últim, completarem el procés realitzant la trucada mitjançant el mètode *call()*.

Després de la crida al servei ja estem en disposició d'obtenir el resultat retornat per el mètode web cridat. Això ho aconseguim mitjançant el mètode *getResponse()*. Depenent del tipus de resultat que esperem rebre haurèm de convertir aquesta resposta a un tipus o un altre.

Ens hem d'enrecordar de concedir els permisos de accés a Internet a la nostra aplicació, afegint els permisos a l'*Android Manifest*.

Amb tot això ja tindríem preparada la crida al nostre servei web i el tractament de la resposta rebuda.

Fent referència a l'*Android Manifest* podríem dir que és un dels arxius més importants de la nostra aplicació. Aquest *XML* es genera automàticament al crear el projecte i en ell es declararan totes les especificacions de la nostra aplicació. Des de el nom de la versió del nostre programa, les activitats, els permisos, la versió d'*Android* que utilitzarem i les pantalles suportades per la nostra aplicació.

En l'apartat de notificacions del sistema o simplement notificacions nostres les farem mitjançant dos mètodes, entre d'altres.

1. `system.out.println()`: són missatges que s'envien directament a la Consola d'*Eclipse*, concretament a la pestanya *Console*.
2. `Log.d`: són notificacions que es visualitzen a la consola del *Eclipse*, exactament a la pestanya *Log*. el `.d` concretament és per enviar missatges de depuració. N'hi ha d'altres tipus, com per exemple `.e` que s'utilitzen més a per a excepcions o missatges d'error.

En l'apartat d'utilització de l'*API* de *Google Maps*, a mesura que es va anar desenvolupant el projecte *Google* va llançar la versió *v2*. Aquesta versió millora forces aspectes envers la versió anterior, entre els quals destaquem el menor tràfic intercanviat amb el servidor, la utilització de *fragments* i els gràfics en 3D.

Cal destacar, que aquesta última versió només funciona amb dispositius amb *Google Play* instal·lat. És per això, que hem fet ús d'un dispositiu mòbil amb *Google Play* i amb la mateixa versió d'*Android* que la del projecte, ja que només funciona amb dispositius mòbils i no amb la consola virtual de *Eclipse* anomenada *davlik*.

Cal destacar que a diferència de *Android*, *Google Maps* no es un software lliure, està limitat a una sèrie de condicions de servei. Podem utilitzar-lo de forma gratuïta sempre que la nostra aplicació no sol·liciti més de 15.000 notificacions geogràfiques al dia.

Per a poder utilitzar aquest servei de *Google* igual com passa quan s'utilitza des de una pàgina web, serà necessari registrar l'aplicació que l'utilitzarà. Una vegada registrada, se'ns entregarà una clau que tindrem que indicar en el *AndroidManifest*.

Realment necessitem dos claus diferents, una durant el procés de desenvolupament i un altre per a l'aplicació. La raó es que es genera un clau diferent en funció del certificat digital amb el que es firma l'aplicació. En la fase de desenvolupament les aplicacions també han de ser firmades digitalment, però en aquest cas el *SDK* utilitza un certificat especial utilitzat només en la fase de desenvolupament.

En l'apartat del disseny s'han desenvolupat diferents *layouts* o vistes que l'aplicació utilitzarà.

Un *layout* són elements no visuals destinats a controlar la distribució, posició i dimensions dels controls que s'insereixen en el seu interior. Aquests components estenen a la classe *ViewGroup*, com molts altres components contenidors, és a dir, capaços de contenir a altres controls.

Així doncs, i tenint en compte tots aquests aspectes, s'han dissenyat les vistes que a continuació s'especifiquen i amb les quals definim per complet el que seria la interfície amb la que l'usuari de l'aplicació mòbil *Android*

interactuarà. Tot plegat, un disseny senzill, intuïtiu i fàcilment implementable.

- *Vista splash.xml*

La primera vista o vista principal *splash.xml* proporciona una imatge que es mostrarà només iniciar-se l'aplicació. Per fer-ho, s'ha codificat el següent *layout*:

*splash.xml*

```
<LinearLayout > //Orientació vertical i centrat
    <ImageView>
        // Imatge a mostrar
    </ImageView>
</LinearLayout>
```

A partir del qual es genera una vista com la següent:



Imatge 9. *Layout splash.xml*.

- *Vista main.xml*

La següent vista és *main.xml* proporciona a l'usuari dos camps de text en els quals podrà introduir el nom d'usuari i la seva respectiva contrasenya. Conté dos botons, amb diferents accions.

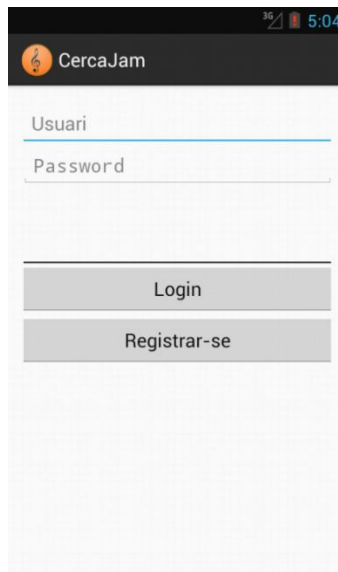
Per fer-ho, s'ha codificat el següent *layout*:



main.xml

```
<LinearLayout> //Orientació vertical i centrat
<EditText>
    // Input nom_usuari
</EditText>
<EditText>
    // Input contrasenya
</EditText>
<Button>
    // Botó de login
</Button>
<Button>
    // Botó de registre
</Button>
</LinearLayout>
```

A partir del qual es genera una vista com la següent:



**Imatge 10. Layout main.xml.**

Com podem observar tenim dos botons, per tant s'obren dos camins d'opcions:

1. *Botó Login*: Una vegada introduïts el nom d'usuari i la contrasenya, es poden produir dues situacions quan polsem aquest botó:
  - No es pot iniciar sessió correctament: davant d'aquesta opció, es mostra un missatge d'error, ja

sigui perquè falta omplir tots els camps, siguin erronis o simplement que l'usuari ja està loguejat.

- Es pot iniciar sessió correctament: davant d'aquesta opció canviarem el valor de l'atribut *loguejat* de la taula *jam\_users* a cert i ens mostrarà la vista *espera.xml*.

2. *Botó Registrar-se*: Una vegada pulsat aquest botó mostra la vista *registre.xml*

- *Vista registre.xml*

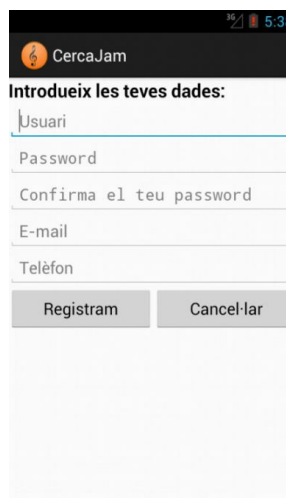
La següent vista és *registre.xml* proporciona a l'usuari cinc camps de text en els quals podrà introduir el nom d'usuari, la contrasenya, confirmar la contrasenya, el seu correu electrònic i finalment el seu telèfon.

Per fer-ho, s'ha codificat el següent *layout*:  
*registre.xml*

```
<LinearLayout> //Orientació vertical i centrat
  <EditText>
    // Input nom_usuari
  </EditText>
</LinearLayout>
<LinearLayout> //Orientació vertical i centrat
  <EditText>
    // Input contrasenya
  </EditText>
</LinearLayout>
<LinearLayout> //Orientació vertical i centrat
  <EditText>
    // Input confirma contrasenya
  </EditText>
</LinearLayout>
<LinearLayout> //Orientació vertical i centrat
  <EditText>
    // Input correu electrònic
  </EditText>
</LinearLayout>
<LinearLayout> //Orientació vertical i centrat
  <EditText>
    // Input contrasenya
  </EditText>
</LinearLayout>
<LinearLayout> //Orientació vertical i centrat
  <Button>
    // Botó de registre
  </Button>
```

```
<Button>
    // Botó de cancel·lar
</Button>
</LinearLayout>
<TextView>
    // Text mostrant missatge
</TextView>
</LinearLayout>
```

A partir del qual es genera una vista com la següent:



**Imatge 11. Layout registre.xml.**

Ens mostra un formulari i a la part inferior tenim dos botons, per tant s'obren dos camins d'opcions:

1. *Botó Registram*: Una vegada introduïts tots els camps, es poden produir dues situacions quan polsem aquest botó:
  - No es pot finalitzar el registre perquè tots els camps no estan introduïts, la confirmació de la contrasenya no coincideixi o simplement perquè l'usuari ja ha estat registrat anteriorment.
  - Es pot iniciar realitzar el registre correctament: davant d'aquesta opció és mostra la vista *espera.xml*. La qual ens enviarà a la vista *opcions.xml*.
2. *Botó Cancel·lar*: Una vegada polsat aquest botó torna a la vista anterior, per tant mostra la vista *main.xml*

- Vista espera.xml

La següent vista és *espera.xml* proporciona una vista on ens mostra un *TextView*, una imatge i una barra de progrés horitzontal. Aquesta vista s'utilitza varies vegades al llarg de l'aplicació. Normalment, l'utilitzem per exemple quan l'usuari fa log in, es registra, es vol inscriure en una sala o simplement fer log out.

Per fer-ho, s'ha codificat el següent *layout*:

*espera.xml*

```
<RelativeLayout> //Orientació vertical i centrat
  <TextView>
    // Missatge a mostrar
  </TextView>
  <ProgressBar>
    // Barra horitzontal que s'anirà carregant
  </ProgressBar>
  <ImageView>
    // Imatge a mostrar
  </ImageView>
</RelativeLayout>
```

A partir del qual es genera una vista com la següent:



Imatge 12. *Layout* espera.xml.

Com podem observar ens mostra l'esmentat abans i ens manté uns segons a l'espera, mentrestant es carrega la barra de progrés, a mostrar el següent *layout*.

- *Vista opcions.xml*

La següent vista és *opcions.xml* proporciona a l'usuari quatre opcions dos camps de text en els quals podrà introduir el nom d'usuari i la seva respectiva contrasenya. Conté dos botons, amb diferents accions.

Per fer-ho, s'ha codificat el següent *layout*:

*opcions.xml*

```
<LinearLayout> //Orientació vertical i centrat

  <LinearLayout> //Orientació vertical i centrat
    <TextView>
      // Text CercaJam
    </TextView>
    <TextView>
      // Text Benvingut
    </TextView>
    <TextView>
      // Text visualitzant nom_usuari
    </TextView>
  </LinearLayout>
  <TableLayout> // Estructura en taula
    <TableRow>
      <LinearLayout>
        <Button>
          // Botó de Mostrar Sales
        </Button>
        <TextView>
          // Text Mostrar Sales
        </TextView>
      </LinearLayout>
      <LinearLayout>
        <Button>
          // Botó de Cercar Sales
        </Button>
        <TextView>
          // Text Cercar Sales
        </TextView>
      </LinearLayout>
    </TableRow>
    <TableRow>
      <LinearLayout>
        <Button>
```

```

// Botó de Afegir Sales
</Button>
<TextView>
// Text Afegir Sales
</TextView>
</LinearLayout>
<LinearLayout>
<Button>
// Botó de Logout
</Button>
<TextiView>
// Text Logout
</TextView>
</LinearLayout>
</TableRow>
</TableLayout>

<TextView>
// Text amb el correu electrònic
</TextView>
</LinearLayout>
```

A partir del qual es genera una vista com la següent:



**Imatge 13. Layout opcions.xml.**

Com podem observar tenim quatre botons, per tant s'obren quatre camins d'opcions:

1. *Botó MostraSales*: Una vegada polsat el botó mostrarà el *layout llista\_sala.xml*.
2. *Botó CercaSales* Una vegada polsat el botó mostrarà el *layout cerca\_sala.xml*.

3. *Botó AfegirSales*: Una vegada polsat el botó mostrarà el *layout* *ubicar\_sala.xml*, una vegada ubicada la sala anirà a la vista del formulari *afegir\_sala.xml*.
4. *Botó Logout*: Una vegada polsat el botó visualitzarem el *layout* *espera.xml* el qual ens retornarà el *layout* *main.xml*.

- *Vista llista\_sala.xml*

La següent vista és *llista\_sala.xml* proporciona una llista de les sales que hi ha a la taula *jam\_users*.

Per fer-ho, s'han codificat els següents *layouts*:

*llista\_sala.xml*

```
<LinearLayout>
    <ListView>
        // Llista visualitzant les sales
    </ListView>
</LinearLayout>
```

*element\_sala.xml*

```
<LinearLayout>
    <TextView>
        // Text per al nom de la sala
    </TextView>
    <TextView>
        // Text per a la ciutat de la sala
    </TextView>
    <TextView>
        // Text per a la capacitat total de la sala
    </TextView>
</LinearLayout>
```

A partir dels quals es genera una vista com la següent:

Razzmatazz
Barcelona
Capacitat: 4
Harlem Jazz Club
Barcelona
Capacitat: 8
Luz de Gas
Barcelona
Capacitat: 5
Monasterio
Barcelona
Capacitat: 6
Sunset Jazz Club
Girona
Capacitat: 7
La Vaqueria
Tarragona
Capacitat: 12
Casal Popular de Joves
Lleida
Capacitat: 11

**Imatge 14.** *Layout llista\_sala.xml/element\_sala.xml.*

Com podem observar ens mostra en una llista totes les sales, si ens hi fixem bé abans hem declarat dos *XML* i només visualitzem una vista.

Per a mostrar més d'un atribut dins de la llista, he creat una vista alternativa anomenada *element\_sala.xml* la qual visualitza els tres atributs. Quan polsem sobre una sala de la llista, ens mostra la vista *descripcio\_sala.xml*.

- *Vista descripcio\_sala.xml*

La següent vista és *descripcio\_sala.xml* proporciona una cerca per nom o ciutat sobre la llista de sales que hem vist abans.

Per fer-ho, s'ha codificat el següent *layout*:

*descripcio\_sala.xml*

```
<LinearLayout> //Orientació vertical i centrat
<LinearLayout> //Orientació vertical i centrat
  <TextView>
    // Títol Nom de la sala
  </TextView>
  <TextView>
    // Text visualitzant el nom de la sala
  </TextView>
</LinearLayout>
<LinearLayout>
  <TextView>
    // Títol Província
  </TextView>
  <TextView>
    // Text visualitzant la província de la sala
```



```
</TextView>
</LinearLayout>
<LinearLayout> //Orientació vertical i centrat
  <TextView>
    // Títol Ciutat de la sala
  </TextView>
  <TextView>
    // Text visualitzant la ciutat de la sala
  </TextView>
</LinearLayout>
<LinearLayout>
  <TextView>
    // Títol Adreça
  </TextView>
  <TextView>
    // Text visualitzant l'adreça de la sala
  </TextView>
</LinearLayout>
<LinearLayout>
  <TextView>
    // Títol Categoria
  </TextView>
  <TextView>
    // Text visualitzant la categoria musical
  </TextView>
</LinearLayout>
<LinearLayout>
  <TextView>
    // Títol Capacitat
  </TextView>
  <TextView>
    // Text visualitzant la capacitat total de la sala
  </TextView>
</LinearLayout>
<TextView>
  // Títol Descripció
</TextView>
<LinearLayout>
  <TextView>
    // Text visualitzant la descripció de la sala
  </TextView>
</LinearLayout>
<LinearLayout>
  <Button>
    // Botó de inscriu-te a la sala
  </Button>
  <TextView>
    // Text mostrant resultat inscriu-te
  </TextView>
</LinearLayout>
<LinearLayout>
  <Button>
    // Botó de Mostrar sala al Mapa
  </Button>
</LinearLayout>
</LinearLayout>
```

A partir del qual es genera una vista com la següent:



Imatge 15. *Layout* descripcio\_sala.xml.

Ens mostra una clara descripció de la sala seleccionada, a la part inferior tenim dos botons:

1. *Inscriu-te*: quan el polsem afegirà el id de l'usuari i el id de la sala a la taula *jam\_sala\_participants* de la base de dades. Hi ha dues opcions:

- *Correcte*: Si l'usuari no s'havia afegit anteriorment, l'acció següent és visualitzar la vista *espera.xml* i després aquesta ens enviarà a visualitzar la vista *llista\_sales.xml*.
- *Incorrecte*: Si l'usuari ja s'havia afegit anteriorment o que les places ja estan cobertes, l'acció següent és visualitzar un missatge d'error a un *TextView*.

2. *Mostra-la al mapa*: quan el polsem ens mostra la vista *mapa\_sala.xml*.

- *Vista mapa\_sala.xml*

La següent vista és *mapa\_sala.xml* proporciona la geo-localització actual de l'usuari i la ubicació de la sala, utilitzant les API's de Google Maps.

Per fer-ho, s'ha codificat el següent *layout*:

mapa\_sala.xml

```
<RelativeLayout>
    <Fragment>
        // Visualitzar Google Maps
    </Fragment>
    <Button>
        // Visualitzar Sala al mapa
    </Button>
    <Button>
        // Calcular distancia entre sala i ubicació actual
    </Button>
</RelativeLayout>
```

A partir del qual ens genera la vista següent:



**Imatge 16. Layout mapa\_sala.xml.**

Ens mostra el fragment de Google Maps i a la part inferior de la vista tenim dos botons que ens ofereixen per una banda veure la ubicació de la sala al mapa, i per l'altra banda calcular la distancia entre la nostra ubicació actual i la ubicació de la sala.

- Vista cerca\_sala.xml

La següent vista és *cerca\_sala.xml* proporciona una cerca per nom o ciutat sobre la llista de sales que hem vist abans. Hi accedim a través de la vista *opcions.xml*.

Per fer-ho, s'han codificat els següents *layouts*:

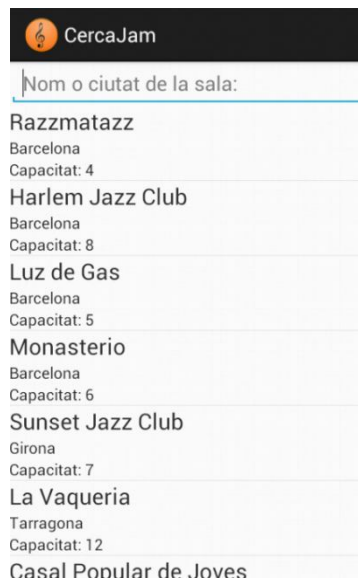
*cerca\_sala.xml*

```
<LinearLayout>
    <LinearLayout>
        <EditText>
            // Cerca la sala per nom o ciutat
        </EditText>
    </LinearLayout>
    <ListView>
        // Llista visualitzant les sales
    </ListView>
</LinearLayout>
```

*element\_sala.xml*

```
<LinearLayout>
    <TextView>
        // Text per al nom de la sala
    </TextView>
    <TextView>
        // Text per a la ciutat de la sala
    </TextView>
    <TextView>
        // Text per a la capacitat total de la sala
    </TextView>
</LinearLayout>
```

A partir dels quals es genera una vista com la següent:



**Imatge 17.** *Layout cerca\_sala.xml/element\_sala.xml.*

Com podem observar ens mostra com l'anterior vista, però ara amb un camp per realitzar una cerca per nom o ciutat de la sala. La part més atractiva ve quan a mesura que anem introduint caràcters al camp de la cerca, ens anirà mostrant les sales que coincideixin amb el nom o ciutat cercat sense confirmació.

Seguim tenint l'opció que quan polsem sobre la sala ens mostri la vista *descripcio\_sala.xml*.

- *Vista ubicar\_sala.xml*

La següent vista és *ubicar\_sala.xml* proporciona la geo-localització actual de l'usuari, utilitzant les API's de Google Maps.

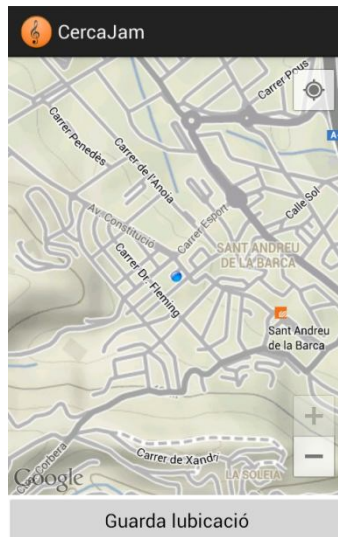
Per fer-ho, s'ha codificat el següent *layout*:

*ubicar\_sala.xml*

```
<RelativeLayout>
    <Fragment>
        // Visualitzar Google Maps
    </Fragment>
    <Button>
```

```
// Guardar posició  
  
</Button>  
  
</RelativeLayout>
```

A partir del qual es genera una vista com la següent:



**Imatge 18. Layout ubicar\_sala.xml.**

Ens mostra el fragment de Google Maps i a la part inferior de la vista tenim un botó que ens serveix per a guardar la ubicació. Quan el polsem ens mostra la vista *afegir\_sala.xml*.

- *Vista afegir\_sala.xml*

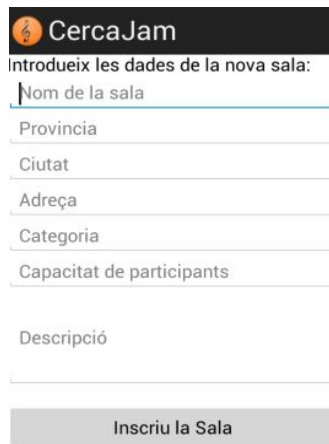
La següent vista és *afegir\_sala.xml* proporciona a l'usuari set camps de text en els quals podrà introduir el nom de la sala, la província, la ciutat, la seva adreça, la seva categoria musical, la capacitat total de la *jam session* i la seva descripció.

Per fer-ho, s'ha codificat el següent *layout*:

afegir\_sala.xml

```
<LinearLayout>
    <EditText>
        // Input nom de la sala
    </EditText >
    < EditText >
        // Input província de la sala
    </EditText >
    < EditText >
        // Input ciutat de la sala
    </EditText >
    < EditText >
        // Input adreça de la sala
    </EditText >
    < EditText >
        // Input categoria musical de la sala
    </EditText >
    < EditText >
        // Input capacitat de la sala
    </EditText >
    < EditText >
        // Input descripció de la sala
    </EditText >
</LinearLayout>
```

A partir del qual es genera una vista com la següent:



**Imatge 19. Layout afegir\_sala.xml.**

Ens mostra un formulari i a la part inferior tenim un botó. Si el polsem obtindrem dos opcions:

1. *Correcte*: quan el polsem afegirà la sala introduïda per l'usuari a la taula *jam\_sales* de la base de dades. També es visualitzarà mitjançant un *TextView* un missatge d'aprovació i ens retornarà a la vista *opcions.xml*.
  2. *Incorrecte*: quan el polsem es visualitzarà mitjançant un *TextView* un missatge d'error.
  - 3.
- *menu.xml*

Des de totes les vistes especificades anteriorment l'usuari pot, a més, fer logout mitjançant un menú. Per fer-lo, s'ha fet servir la següent estructura:

```
<menu>
  <item>
    //Opció 1 : Logout Usuari
  </item>
  <item>
    //Opció 2 : Informació sobre l'aplicació
  </item>
</menu>
```



Genera un menú molt intuïtiu i on l'usuari pot:

- *Fer logout*: quan polsem aquesta opció immediatament canviem el valor de l'atribut *loguejat* de la taula *jam\_users* a fals.
- *Veure informació sobre l'aplicació*: quan polsem aquesta opció ens mostra un *pop-up* amb la informació de l'aplicació.

### 3.4 Diagrames del sistema

Els diagrames associats al sistema que he desenvolupat ajuden a entendre millor i d'una forma molt més detallada el funcionament i la interacció de cada mòdul amb la resta del sistema en general.

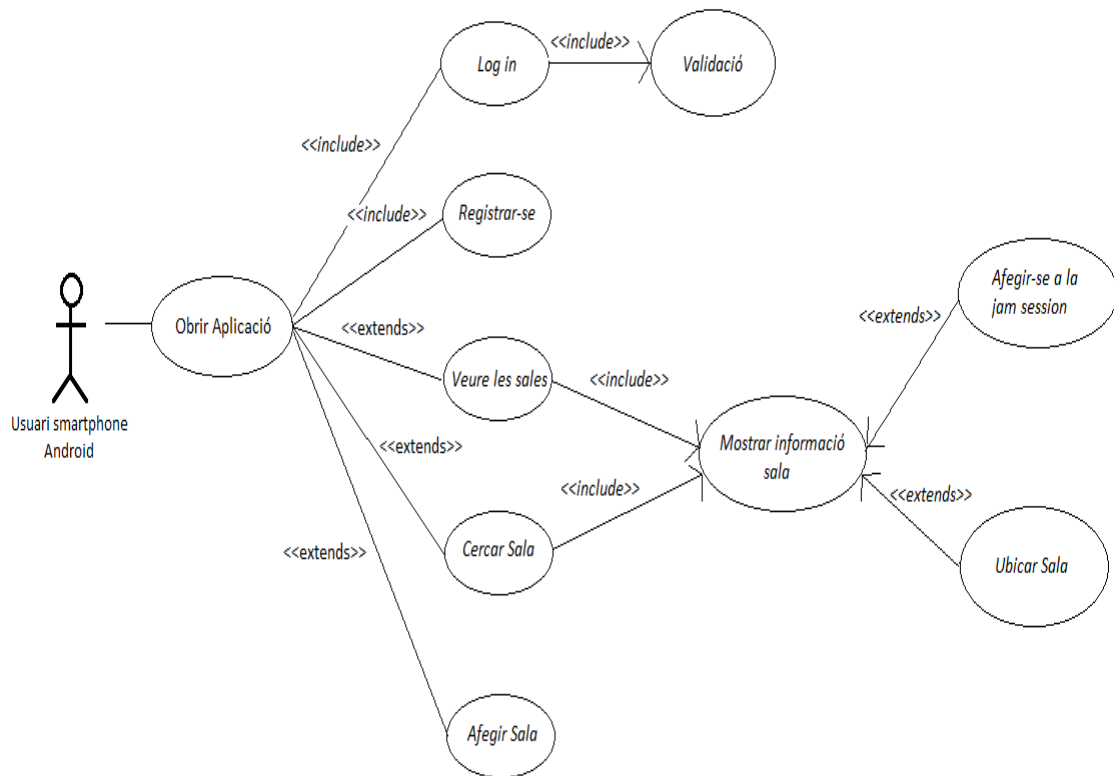
La claredat amb la que el conjunt de diagrames descriuen el sistema és també un indicatiu del correcte disseny i estructuració d'aquest i formen una part molt important del projecte en sí.

A continuació, es mostren un seguit de diagrames principals associats al sistema.

#### 3.4.1 Diagrames de casos d'ús

El diagrama de casos d'ús és el principal del sistema. Ens ajuda a identificar els diferents actors presents en el sistema i les accions que poden realitzar, de forma directa o indirecta, cadascun d'ells.

El diagrama de casos d'ús associat al sistema és el següent:

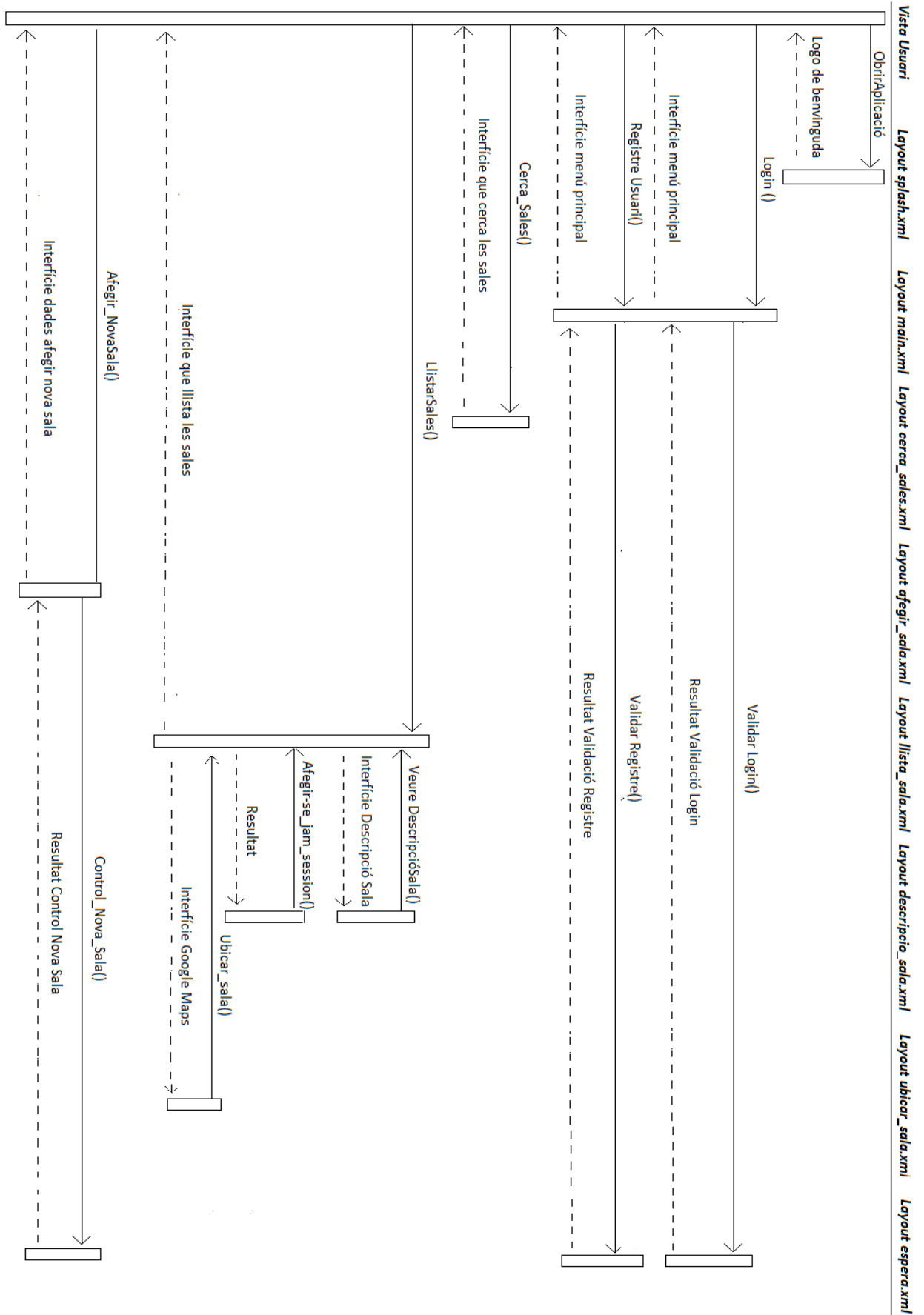


Imatge 20. Diagrama de casos d'ús del sistema.

### 3.4.2 Diagrames de seqüència

El diagrama de seqüència mostren la interacció del conjunt de classes o objectes de l'aplicació en el temps i se'n modela un per a cada cas d'ús. A més, permet veure quin és el intercanvi de missatges entre els diferents elements del sistema.

El diagrama de seqüència associat al cas d'ús és el següent:



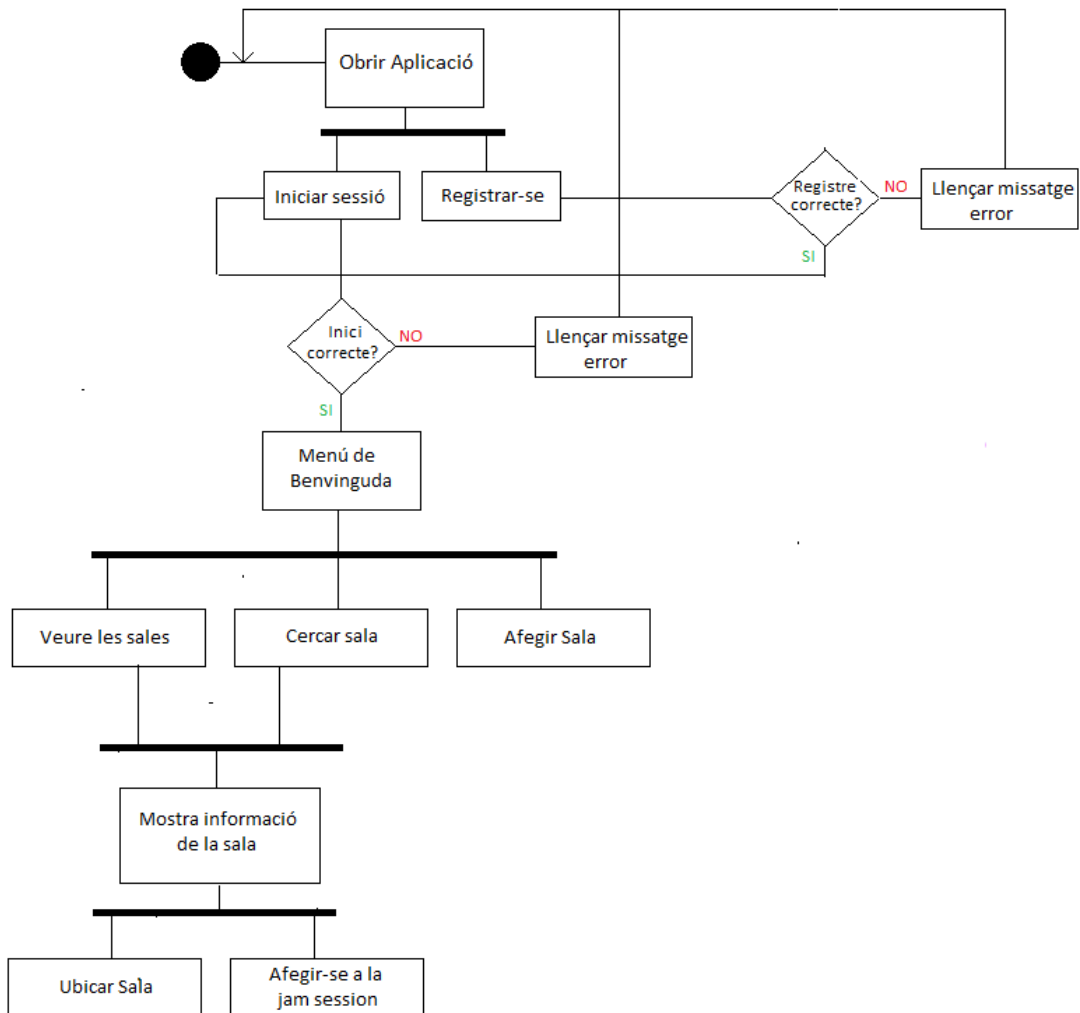
Imatge 21. Diagrama de seqüència del sistema.

En aquest cas el que podem veure és com la vista de l'usuari va canviant segons amb quin *layout* es comunica l'usuari. Observem a més, que és un accés seqüencial en el que no es pot accedir a qualsevol *layout* en qualsevol moment, sinó que s'ha de saltar d'un a un altre.

### 3.4.3 Diagrama d'activitats

Per últim, els diagrames d'activitats ens donen una idea dels diferents fluxos de treball presents al sistema. Són molt explicatius i també se'n representat un per a cada cas d'ús.

El diagrama d'activitats associat al cas d'ús és el següent:



Imatge 22. Diagrama d'activitats del sistema.

## 4. Test i proves

Per aquest apartat hem de tenir molt en compte que estem davant d'un projecte on el resultat és una aplicació per a dispositius mòbils amb sistema *Android*. Això significa que és una aplicació funcional; però encara hi ha alguns detalls que és imprescindible millorar abans de pujar la aplicació al *Market d'Android* com a aplicació final.

Aleshores doncs s'han aplicat uns criteris de testos basats estrictament en la funcionalitat del sistema. Per tant, la seguretat del sistema no queda en cap moment garantida.

S'han aplicat bàsicament proves d'unitat mòdul a mòdul per comprovar de forma independent que cadascun d'ells funciona correctament.

Paral·lelament, cada cop que un mòdul superava correctament les proves d'unitat per assegurar-ne la funcionalitat en els casos bàsics, s'han anat agrupant en l'aplicació beta final per aplicar proves d'integració pas a pas.

### 4.1 Proves d'unitat de la base de dades

A la taula següent, es poden veure les proves a les quals s'ha sotmès la Base de Dades:

#	Requeriment	Resultat esperat	Resultat obtingut	Tester
1	Accés complet a les taules del sistema a través de <i>phpMyAdmin</i> .	Poder connectar a la base de dades a través del navegador a la direcció <a href="http://localhost/phpmyadmin">http://localhost/phpmyadmin</a> a Windows 8.	Com l'espera't.	EPO
2	Connexió a la base de dades des de Java ( <i>web service</i> ).	Poder connectar a la base de dades a través de codi Java.	Com l'espera't.	EPO
3	Execució de sentències SQL des de la interfície de <i>phpMyAdmin</i> .	Executar SELECT, INSERT, UPDATE.	Com l'espera't.	EPO
4	Execució de sentències SQL des de Java ( <i>web service</i> ).	Executar SELECT, INSERT i UPDATE a través de codi Java.	Com l'espera't.	EPO
5	Exportació de tot contingut de la Base de Dades per a còpies de seguretat.	Obtenir un fitxer de sortida amb tota l'estructura de la Base de Dades per poder importar-la en cas de desastre.	Com l'espera't.	EPO

Taula 14: Proves realitzades a la base de dades.

## 4.2 Proves d'unitat del servei web

L'objectiu principal del servei web que sigui un sistema funcional, senzill i ràpid. La resta d'aspectes no menys importants com la seguretat i el disseny no jugaven un paper important en el nostre desenvolupament.

És el nostre "intermediari" entre l'aplicació *Android* i la base de dades, quan els usuaris sol·liciten peticions a la base de dades tot passa per el servei web.

A continuació es mostra una relació de les proves a les quals s'ha sotmès al servei web:

#	Requeriment	Resultat esperat	Resultat obtingut	Tester
1	Iniciar el servidor web ( <i>tomcat+axis2</i> ) a través d' <i>Eclipse</i> .	Poder accedir a la web usant un navegador a la direcció localhost:8080/CercaJamWS/servises/listServices	Com l'espera't.	EPO
2	Log in dels usuaris i registre .	Establir un control quan l'usuari intenta fer login. Controlar que hagi introduït tots els camps i siguin correctes i que l'usuari no estigui loguejat per tal connectar satisfactòriament amb la base de dades.	Com l'espera't.	EPO
3	Obtenir Sales.	Obtenir un cadena amb el contingut de totes les sales obtingudes de la taula <i>jam_sales</i> . Separades per "&&" i finalitzades per "__".	Com l'espera't.	EPO
4	Registre d'usuari	Quan un usuari es registra per primer cop, controlem que aquest usuari no consti a la base de dades. Si tot ha anat bé, les seves dades s'insereixen correctament a la base de dades.	Com l'espera't.	EPO
5	Proporcionar un sistema d'inserció del usuari a la sala.	Controlar quan l'usuari es vulgui afegir a una sala, que no s'hagi afegit abans i que no estigui plena. Per tal d'afegir-lo a la taula <i>jam_sala_participants</i> de la base de dades.	Com l'espera't.	EPO

Taula 15: Proves realitzades al servei web.

### 4.3 Proves d'unitat de l'aplicació Android

Resum de les proves aplicades a la aplicació *Android*:

#	Requeriment	Resultat esperat	Resultat obtingut
1	Proporcionar un sistema de log in i log out funcional.	Establir una vista inicial de benvinguda que obligui a connectar-s'hi per seguir.	Com l'espera't.
2	Canviar el número de punts d'interès.	L'usuari pot determinar quants punts d'interès vol visualitzar, ja siguin els 10 per defecte com qualsevol altre número.	Com l'espera't.
3	Permetre visualitzar les sales.	Un cop l'usuari s'ha registrat, pot veure les diferents sales, podent accedir a la descripció de cadascuna d'elles.	Com l'espera't.
4	Inscripció de l'usuari.	Mitjançant un botó l'usuari pot inscriure's a la <i>jam session</i> de la sala.	Com l'espera't.
5	Permetre la cerca per nom o ciutat de sales.	Mitjançant un camp de text podem cercar per nom o ciutat les sales. A mesura que anem introduint caràcters, es visualitzen les sales que hi coincideixin.	Com l'espera't.
6	Permetre a l'usuari afegir una nova sala.	Primer de tot guardar la posició actual de l'usuari mitjançant un mapa,. Quan hi estiguem d'acord farem la confirmació mitjançant un botó, ens enviarà a una vista on omplirem un formulari amb les dades de la sala.	Com l'espera't.
7	Permetre veure en un mapa una sala determinada.	Visualitzar en un mapa la posició d'un punt concret. També tenim l'opció de calcular la distància entre la nostra posició actual i la de la sala.	No superat. La distància que calculem es la distància en graus. A més quan calculem la posició actual es queda sempre a aquell punt del mapa.
8	Permetre la desconnexió de l'usuari.	L'usuari pot escollir si vol desconnectar-se de l'aplicació. Mitjançant una opció del botó físic ( <i>Menu</i> ) del dispositiu o també des de el menú principal de l'usuari.	Com l'espera't.

Taula 16: Proves realitzades a la aplicació Android.

#### 4.4 Proves d'integració

A continuació es mostra una taula amb la relació de proves d'integració que s'han realitzat al sistema:

#	Requeriment	Resultat esperat	Resultat obtingut
1	Inserció de peticions a la base de dades.	Les peticions que es generen mitjançant la pàgina web a la base de dades, el mòdul d'extracció de punts d'interès és capaç de veure-les.	Com l'espera't.
2	Lectura de punts d'interès a la base de dades per part de la interfície web.	El mòdul d'extracció de punts d'interès pot afegir nous punts d'interès a la base de dades i la aplicació web pot accedir a ells.	Com l'espera't.
3	Pujada d'arxius a la base de dades.	El mòdul d'extracció de punts d'interès pot visualitzar els arxius que es pugen a la base de dades mitjançant la interfície web.	Com l'espera't.
4	Lectura de sales a la base de dades per part de l'aplicació Android.	El mòdul d'extracció de punts d'interès pot afegir nous punts d'interès a la base de dades i el Web Service pot accedir a ells.	Com l'espera't.
5	Obtenció de la definició de la wikipèdia per a un punt.	L' aplicació <i>Android</i> ha de ser capaç de comunicar-se amb el mòdul d'extracció de definicions de la wikipèdia per tal de retornar la descripció sobre un punt concret.	Com l'espera't.

Taula 17: Proves d'integració realitzades.



## 5. Conclusions

### 5.1 Valoració d'objectius assolits

L'objectiu principal del projecte era desenvolupar una aplicació *Android*, on els usuaris es poguessin registrar i consegüentment poguessin iniciar sessió. A més, podran llistar les diferents sales on es realitzin les *jam sessions* però la part més important és que l'usuari té l'opció d'inscriure's a la *jam session* de la sala i obtenir la ubicació de la sala.

També pot realitzar una cerca per nom o ciutat i fins i tot podrà afegir una sala nova.

A continuació es mostren els objectius marcats a l'estudi de viabilitat (*apartat 2.3*), el qual consta de 10 objectius:

- 1) Desenvolupar una aplicació mòbil per a sistemes *Android* utilitzant les crides a serveis web. **(Complert)**
- 2) Desenvolupar un mòdul per a gestionar les sessions dels usuaris. **(Complert)**
- 3) Desenvolupar un mòdul capaç de visualitzar les respectives sales i que l'usuari pugui veure la seva informació detallada i geo-localització, amb la possibilitat d'inscriure's a les Jam Sessions. **(Complert)**
- 4) Desenvolupar un mòdul capaç d'inserir noves sales, les quals hauran de guardar la geo-localització actual on es troba l'usuari. **(Complert)**
- 5) Desenvolupar un mòdul capaç de cercar les sales per nom o població. **(Complert)**
- 6) Desenvolupar servei web que permeti generar peticions a la base de dades. **(Complert)**
- 7) Aprofundir en el coneixement del llenguatge Java. **(Complert)**
- 8) Aprofundir en el coneixement de l'entorn de desenvolupament *Eclipse*, la integració d'aquests amb altres mòduls i en el desenvolupament del servei web de l'aplicació. **(Complert)**
- 9) Aprofundir en el coneixement del desenvolupament d'aplicacions mòbils. **(Complert)**
- 10) Compte administrador per al manteniment de la base de dades. **(Complert)**

Com es pot observar, tot els objectius proposats en aquest projecte s'han pogut dur a terme. De manera que es pot afirmar que el projecte es viable.

Tot i que es tracta d'un prototip, un cop estigués finalitzat com a versió final, aquest es podria introduir al *Market d'Android* per tal que qualsevol usuari es pogués descarregar la aplicació amb un cost mínim. D'aquesta manera, recuperariem la inversió inicial i a més, els usuaris podrien començar a gaudir d'aquesta aplicació organitzant les seves properes *jam sessions*.

## 5.2 Futures línies de treball

En aquest apartat es pretén proposar un seguit de millores o línies futures de treball per tal d'aconseguir una aplicació final amb una funcionalitat perfecte o ideal. És per aquest motiu que es proposen ampliacions o millores vers la interfície *Android* i el servei web.

### 5.2.1 Ampliacions del servei web

El servei web que s'ha desenvolupat en aquest projecte, està dissenyat per a realitzar les connexions a la base de dades i retornar les peticions a la nostra aplicació amb sistema *Android*. Ha estat configurat mitjançant un servidor *Tomcat* amb *Axis 2* utilitzant *Eclipse Indigo*. Per tant, una possible millora del servei web seria:

1. La seguretat de les dades: aconseguir encriptar les dades que enviem des de el servei web cap a la base de dades. Un dels camps relacionat seria la contrasenya.
2. Control variable sessió: Per altre banda, un altre aspecte important a desenvolupar és l'expiració del logout. Aconseguir que la variable de sessió que utilitzem, per saber si l'usuari està a dins l'aplicació o no, pugui tenir caducitat. És a dir, que torni a estat fals (no loguejat) quan porti un cert temps d'inactivitat. Amb aquest punt, obtindríem una gran millora alhora de resoldre aquest problema. Ens solucionaria els casos quan l'usuari per qualsevol causa no hagi tancat correctament l'aplicació o per inactivitat.

Aquestes implementacions no s'han dut a terme per el dens volum de classes que s'haurien de canviar o desenvolupar

## 5.2.2 Ampliacions de l'interfície Android

L'aplicació que s'ha desenvolupat en aquest projecte, està dissenyada per a plataformes *Android*, ja que és un llenguatge de programació lliure (en aquest cas *Java*) i és gratuït. Per tant una possible millora de l'aplicació seria:

1. Aplicació multiplataforma: aconseguir que la aplicació no tan sols funcioni per a dispositius *Android*, ja siguin mòbils com tabletetes, sinó que també sigui funcional per a dispositius amb altres sistemes operatius com per exemple, *iOS*, *Blackberry* o *Windows Phone*. Per a poder implementar la aplicació i fer-la funcional per a dispositius *iOS* s'haurien de traduir totes les funcions de *Java* a *Objective C*.

De la mateixa manera que per a desenvolupar la aplicació *Android* s'ha hagut d'instal·lar el complement *ADT*, per a desenvolupar aplicacions funcionals per a terminals *Blackberry*, s'ha d'instal·lar el complement *Blackberry JDE (Java Development Environment)* i adaptar les classes *Android* a *Blackberry*. Finalment, per implementar l'aplicació a dispositius que utilitzin el sistema operatiu *Windows Phone* s'han de traduir les funcions fetes en *Java* a *C#*.

2. La conta de usuari: aconseguir que l'usuari pugui modificar i afegir una foto al seu perfil. També quan l'usuari vulgui fer log in i no recordi el seu nom d'usuari o contrasenya pugui ser sol·licitada via correu electrònic.
3. El disseny de l'aplicació Android: aconseguir millores visuals tals com implementar un nou disseny quan l'usuari ha fet log in a l'aplicació tingui les opcions en diferents pestanyes i amb un simple moviment de dit cap a la dreta o esquerra pugui canviar de *layout*, aquest mètode seria fent ús de *scroll views*.
4. Connectar amb xarxes socials: fer referència avui en dia de les xarxes socials és una realitat establerta. La possibilitat d'afegir aquesta opció a la nostra aplicació per tal de compartir informació, quan l'usuari faci ús de la nostra aplicació, a les xarxes socials.
5. Ubicació de la sala: millorar l'apartat de càlcul de distància entre la

nostra posició i la de la sala. També podríem fer ús de com arribar-hi, tal i com fa l'aplicació *Google Maps*, amb la possibilitat d'arribar-hi en cotxe o a peu i sobretot, que calculi el camí per carrers o carreteres i no traçant una línia al mapa entre els dos punts.

Aquestes implementacions no s'han dut a terme per el dens volum de classes que s'haurien de canviar o desenvolupar.

### 5.3 Seguiment de la Planificació

Les planificacions de projectes complexos solen ser optimistes i no sempre s'acaba complit amb els plans establerts al inici. En concret, la planificació del projecte s'ha vist modificada respecte a la que es va proposar a finals de l'any anterior.

Un dels principals problemes que vaig trobar-me va ser la funcionalitat alhora de realitzar la connexió al servei web des de *Android* utilitzant el protocol *SOAP*. També va ser quan en l'aplicació *Android* vam voler visualitzar totes les sales, que provenien del servei web, en un *listview* que visualitzes el seu nom, ciutat i capacitat total. I un cop superades tots aquests problemes l'últim gran repte va ser quan vaig voler realitzar la cerca per nom o ciutat.

A continuació, es mostra una taula amb la relació de tots els problemes que han anat sorgint i la seva corresponent implementació:

Problema	Solució Aplicada
Integració del <i>TomCat</i> i <i>Axis2</i> amb <i>Eclipse</i> .	<i>Documentació a la web</i> <sup>1</sup> per com fer- los funcionar.
Pas i retorn de paràmetres de la part <i>Android</i> al servei web.	S'ha solucionat fent ús només de variables tipus <i>Strings</i> , fent la seva respectiva conversió quan la variable havia de ser enviada a la base de dades per no obtenir errors.
Instal·lar el complement <i>ADT</i> en <i>Eclipse</i> .	S'ha hagut de cercar informació a la pàgina oficial de <i>Android Developers</i> per trobar la manera de que funcionés correctament.
Canvi de versió d' <i>Eclipse</i> per incompatibilitat entre <i>TomCat</i> , <i>Axis2</i> i <i>ADT</i> .	S'ha instal·lat la versió d' <i>Eclipse JEE Indigo</i> , la qual accepta aquests complements.
Canvi de versió de sistema <i>Android</i> , per incompatibilitat amb la versió del mòbil per realitzar les proves.	S'ha instal·lat la versió 4.1.2 <i>JellyBean</i> , ja que la versió anterior era <i>Gingerbread</i> 2.3.3. Amb aquest canvi s'ha de fer ús de <i>threads</i> quan connectem al servei web, ja que abans ho fèiem tot des de el <i>UI thread</i> .
Comunicació <i>Android</i> i la base de dades.	S'ha solucionat fent ús del <i>WebService</i> .

Funcionament del manifest d'Android.	Cerca d'informació sobre com funciona i com s'ha d'utilitzar.
Utilització del GPS en el dispositiu mòbil	<i>Documentació a la web</i> <sup>2</sup> sobre com accedir a la informació que et pot proporcionar qualsevol dispositiu <i>Android</i> i la inserció de una nova llibreria que utilitzarem en el nostre projecte <i>Eclipse</i> .

**Taula 18: Problemes sorgits.**

Cal a dir que gràcies a la *documentació web*<sup>1,2</sup> i al *curs Android*<sup>3</sup> he pogut assolir els meus objectius amb la planificació i sobretot en l'ensenyament dels coneixements per al desenvolupament d'aplicacions *Android* en aquest projecte, ja que ha estat un entorn nou de desenvolupament i un llenguatge pràcticament desconegut.

#### 5.4 Valoració personal

A nivell personal cal remarcar que des de el inici vaig plantejar aquest projecte no com a exercici acadèmic de final de carrera si no com a prova definitiva de les meves capacitat i habilitats sobre un escenari el més pròxim al món laboral.

La idea del projecte va sorgir quan una nit vaig anar prendre una copa a un bar, que està a prop de la feina, on habitualment fan *jam sessions*. Bocabadat amb la passió dels músics, que la majoria d'ells ni es coneixien, vaig pensar que realment podria ajudar-los a conèixer nous locals o bars, i sobretot musics amb els qui poder tocar. De manera que finalment, es van anar dibuixant els diferents mòduls que hauria d'incloure el sistema a realitzar. Ara que l'aplicació està finalitzada, estic satisfet d'haver estat capaç de portar a terme el projecte tal i com s'havia ideat inicialment.

En quant a nivell tècnic he adquirit moltes habilitats començant per el llenguatge Java. És un dels més utilitzats, simple i molt potent. Gràcies a l'esforç i dedicació en l'aprenentatge del llenguatge Java durant aquest projecte ha estat fonamental, ja que em faltaven refrescar i també adquirir nous coneixements.

<sup>1</sup> Informació extreta de la web: <http://codeoncloud.blogspot.com.es>

<sup>2</sup> Informació extreta de la web: <http://www.sgoliver.net/blog/?p=3244>

<sup>3</sup> Curs realitzat de "Desarrollo de aplicaciones Android" del Foment del Treball.

També he adquirit altres habilitats tals com treballar sota un entorn de desenvolupament diferent als vistos fins ara, utilitzar el meu portàtil com a servidor web gràcies a la utilització del complement per *Eclipse Tomcat* amb *Axis2*, refrescar conceptes de base de dades tals com *query's* i relacions, treballar en detall amb *phpMyAdmin*, aprofundir en les eines ofimàtiques, etc.

A més del que s'ha exposat anteriorment, he après a veure el que realment costa planificar un projecte, formalitzar els tràmits previs necessaris per tal de dur-lo a terme i sobretot sent responsable justificant cada pas. He après a gestionar-me el temps de dedicació juntament amb les activitats que s'havien d'anar fent. Finalment he pogut viure en primera persona el que significa ser desenvolupador, analista i cap d'un projecte.

Com a conclusió final, estic satisfet amb el treball realitzat i els objectius assolits.

## 6. Glossari

### 6.1 Relació de taules

#### Taules

TAULA 1: Objectius específics.....	13
TAULA 2: Catalogació dels requisits funcionals.....	17
TAULA 3: Catalogació dels requisits no funcionals.....	17
TAULA 4: Primer component de l'equip.....	20
TAULA 5: Segon component de l'equip.....	20
TAULA 6: Informació dels actors dins del desenvolupament del projecte...20	
TAULA 7: Relació i detalls dels recursos del projecte.....	21
TAULA 8: Programari disponible.....	22
TAULA 9: Tasques del projecte.....	22
TAULA 10: Catalogació de riscos.....	26
TAULA 11: Resultat d'aplicar les solucions als riscos.....	27
TAULA 12: Estimació cost de personal.....	27
TAULA 13: Estimació cost dels recursos.....	28
TAULA 14: Proves realitzades a la base de dades.....	69
TAULA 15: Proves realitzades al servei web.....	70
TAULA 16: Proves realitzades a la aplicació <i>Android</i> .....	71
TAULA 17: Proves d'integració realitzades.....	72
TAULA 18: Problemes sorgits.....	76

### 6.2 Relació d'imatges

#### Imatges

IMATGE 1: Context del projecte.....	10
IMATGE 2: Descripció física del sistema.....	15
IMATGE 3: Lògica del sistema.....	16
IMATGE 4: Planificació temporal del projecte.....	25
IMATGE 5: Implementació del sistema.....	31
IMATGE 6: Taula <i>jam_users</i> de la base de dades Cercajam.....	32
IMATGE 7: Taula <i>jam_sales</i> de la base de dades Cercajam.....	33
IMATGE 8: Taula <i>jam_sales_participants</i> de la base de dades.....	34
IMATGE 9: Layout splash.xml.....	48
IMATGE 10: Layout main.xml.....	49
IMATGE 11: Layout registre.xml.....	51
IMATGE 12: Layout espera.xml.....	52

IMATGE 13: Layout opcions.xml.....	54
IMATGE 14: Layout llista_sala.xml/element_sala.xml.....	56
IMATGE 15: Layout descripcio_sala.xml.....	58
IMATGE 16: Layout mapa_sala.xml.....	59
IMATGE 17: Layout cerca_sala.xml/element_sala.xml.....	61
IMATGE 18: Layout ubicar_sala.xml.....	62
IMATGE 19: Layout afegir_sala.xml.....	64
IMATGE 20: Diagrama de casos d'ús del sistema.....	66
IMATGE 21: Diagrama de seqüència del sistema.....	67
IMATGE 22: Diagrama d'activitats.....	68



## 7. Bibliografia

### 7.1 Documentació web

- Eclipse– The Eclipse Foundation open source community website, <http://www.eclipse.org/>, 2013.
- MySQL: The world's most popular open source database, <http://mysql.com/>, 2013.
- phpMyAdmin, [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php), 2013.
- Apache Friends – Xampp, <http://www.apachefriends.org/es/xampp.html>, 2013.
- Google Maps API – Google Developers, <https://developers.google.com/maps/?hl=es>,
- 2013.
- Android Developers, <http://developer.android.com/index.html>, 2013.
- Apache Tomcat – Welcome!, <http://tomcat.apache.org/>, 2013.
- Stack Overflow, is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required. <http://www.stackoverflow.com/>, 2013.
- Sgoliver Blog, web dedicada a la programació *Android*, *Java* i *.NET* <http://sgoliver.net/>, 2013.
- Codeoncloud, Random tutorials for beginners <http://codeoncloud.blogspot.com.es/>, 2013.
- Androideity, Comunitat mexicana orientada al aprenentatge i ensenyament del desenvolupament d'aplicacions *Android*, <http://androideity.com/>, 2013.
- El Android Libre, Blog d'Aplicacions, Notícies, Jocs i *Smartphones* Lliures, <http://www.elandroidlibre.com/>, 2013.
- Android Curso, Programació d'aplicacions per a mòbils, <http://www.androidcurso.com/>, 2013.
- Normativa de Projectes Enginyeries Tècniques en Informàtica, [http://www.uab.cat/Document/541/595/Normativa\\_PF\\_CNovembre2010.pdf](http://www.uab.cat/Document/541/595/Normativa_PF_CNovembre2010.pdf), 2010.

## 7.1 Llibres

- Android 4: Desarrollo de aplicaciones, Wei-Meng Lee (Anaya Multimedia, GRUPO ANAYA S.A, 2011).

## 7.2 Curs Android

- Realització i finalització del curs Android "**Desarrollo de aplicaciones Android**" ofert per el Foment del Treball del 25 de Maig de 2013 al 25 de Juliol de 2013.

## 8. Agraïments

A la meva família i la meva parella per haver estat al meu costat en tot moment, encoratjar-me per arribar fins aquí.

Als meus amics que han sabut entendre la dedicació màxima, pel poc temps que de vegades els hi he pogut dedicar, que he tingut vers al projecte.

Agrair també al director del projecte Jordi Duran i Cals, que sempre ha cregut possible la realització d'aquest projecte i m'ha dedicat tot el seu temps.

Autor del document:

Edgard Puig Orpinell

Setembre 2013