



EPS

Escola Politècnica

UdG

Superior

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: SocialSong

Document: Memòria

Alumne: Marc Amorin Casañas

Director/Tutor: Ignacio Martín Campos

Departament: Informàtica i Matemàtica Aplicada

Àrea: LSI

Convocatòria (mes/any): Juny/2014

1. Introducció	5
1.1 Motivacions	6
1.2 Propòsit i objectius del projecte.....	6
2. Estudi de viabilitat	7
2.1 Cost Desenvolupament	7
2.2 Cost Producció.....	8
3. Metodologia	9
4. Planificació	11
4.1 Planificació inicial	11
4.2 Iteracions de Programació	12
5. Marc de treball i conceptes previs	15
5.1 Arquitectura Client-Servidor	15
5.2 Web Services	15
5.2.1 REST	15
5.2.2 JSON	16
5.3 Aplicació single-page	16
5.4 Xarxes socials	16
5.5 Historia de les xarxes socials	16
5.6 Classificació xarxes socials	18
5.6.1 Xarxes socials verticals	18
5.6.2 Xarxa social web	18
6. Requisits del sistema	19
6.1 Requeriments funcionals.....	19
6.1.1 Usuari no registrat.....	19
6.1.2 Usuari registrat.....	19
6.1.3 Membre Fundador i administrador.....	20
6.1.4 Membre bàsic.....	20
6.2 Requeriments no funcionals	21
6.2.1 Client	21
7. Estudis i decisions	22
7.1 Servidor	22
7.1.1 Java EE	22
7.1.2 Servidor d'aplicacions.....	23
7.1.3 Base de dades.....	23
7.1.4 API REST	24
7.2 Aplicació client	25
7.2.1 Frameworks.....	25
7.3 Entorn de treball	26

7.3.1	IntelliJ IDEA 13.....	26
7.3.2	Bitbucket	26
8.	Anàlisi i disseny del sistema	27
8.1	Descripció general	27
8.2	Actors	27
8.2.1	Usuari no registrat.....	29
8.2.2	Usuari registrar.....	31
8.2.3	Membre de grup fundador/Administrador	47
8.2.4	Membre de grup bàsic	54
8.3	Model de dades inicial	61
8.4	Disseny d'aplicació	62
8.4.1	Arquitectura global.....	62
8.4.2	Servidor	62
8.4.3	Aplicació client	89
9.	Implementació i proves.....	97
9.1	Hibernate	97
9.2	Backbone.....	99
9.2.1	AMD	99
9.2.2	RequireJS	99
9.2.3	Estructura	99
9.2.4	Configuració	100
9.5	SoundCloud API.....	102
9.5.1	Configuració	102
9.5.2	Utilització.....	102
9.6	YouTube API	103
10.	Implementacions i resultats	104
10.1	Implementació del servidor JBoss local.....	104
10.2	OpenShift	105
10.3	Resultats.....	106
10.3.1	Registre	106
10.3.2	Editar	107
10.3.3	Perfil d'usuari	108
10.3.4	Publicació	109
10.3.4	Comentari.....	110
10.3.5	Crear un grup	111
10.3.6	SoundCloud	112
10.3.7	Invitar a membre.....	113
10.3.8	Youtube	115

10.3.9 Buscar	116
10.3.10 Seguir a un grup	117
10.3.11 Publicacions relacionades amb un hashtag.....	118
11. Conclusions	119
11.1 Objectius	119
11.2 Aprenentatge	119
12. Treball futur	120
13. Bibliografia	121
14. Annexos.....	123
14.1 Condicions del servei.....	123
14.1.1 Política d'ús	123
14.1.2 Galetes	125
14.2 Documentació completa de API REST	126

1. INTRODUCCIÓ

En els últims anys el creixement dels dispositius intel·ligents han fet que la connexió a internet pugui estar a l'abast de tothom en tot moment i en qualsevol lloc, això a provocat que aplicacions com Spotify, YouTube, Facebook o Twitter, entre altres, incrementin els seus usuaris actius, ja que són eines les quals la gran part de la població les coneix i sap fer-les servir.

El poder que tenen les xarxes socials no ha passat desapercbut per aplicacions com SoundCloud o Spotify que han volgut transformar de mica en mica l'aplicació original en una xarxa social per captar major nombre de clients.

YouTube, tot i no ser una plataforma orientada a la música, avui en dia és l'aplicació que té més influència a la música. L'aplicació comprada per Google, en el Octubre del 2006 per 1.650 milions de dollars, és una fàbrica d'èxits. Artistes com Justin Bieber, PSY o Gotye han vist incrementada la seva popularitat de la nit al dia gràcies a YouTube. L'aplicació de Google té una comunitat de 1.000 milions d'usuaris únics al mes i és utilitzada per més del 37% de la població mundial de internet.

En una comparativa entre YouTube i Spotify, mitjançant la cançó "Thrift Shop", podem veure quina de les dos aplicacions té més impacte sobre la música. La cançó del grup americà va tenir 433 milions de reproduccions del vídeo oficial abans de l'octubre del 2013, en el mateix període de temps l'aplicació sueca va tenir 150 milions de reproduccions, el que equival a només al 34,64% de les reproduccions de YouTube.

Els grans beneficiats d'aquests canvis són els consumidors d'aquestes plataformes i la música, aquesta última es dona a conèixer "gratuïtament" i és una gran avantatge per la expansió dels gèneres menys coneguts i per els grups que comencen, que veuen en aquestes noves aplicacions socials una eina per donar a conèixer la seva música i poder arribar a tothom amb major facilitat.

1.1 MOTIVACIONS

La meva motivació per crear aquest projecte és conèixer amb profunditat el món del desenvolupament d'aplicacions web, un fenomen que segons la meua opinió tindrà gran importància en els pròxims anys, la creació d'una API REST i el coneixement de les eines que actualment es fan servir per aquest desenvolupament. La creació d'una API REST fa que tinguis flexibilitat per desenvolupament del client, et permet poder desenvolupar una aplicació Android, iOS i/o web.

1.2 PROPÒSIT I OBJECTIUS DEL PROJECTE

L'objectiu principal del projecte és desenvolupar una aplicació web, concretament una xarxa social, orientada a la música, que tindrà com objectiu difondre la música i donar a conèixer nous grups de música als usuaris de l'aplicació web.

Els usuaris podran crear lliurement el seu grup, i una vegada creat podran posar vídeos, cançons i els pròxims esdeveniments.

La xarxa social compta amb un mecanisme de publicacions, les quals es poden posar vídeos de YouTube, cançons de SoundCloud i hashtags per dinamitzar l'ús i la cerca d'esdeveniments fàcilment.

2. ESTUDI DE VIABILITAT

Per desenvolupar una aplicació web i una API REST no es necessita una gran infraestructura i els costos d'aquesta no són gaire elevats, per la qual cosa fa que el cost inicial per començar a desenvolupar l'aplicació sigui assequible.

Les eines inicials per desenvolupar aquest projecte han sigut:

- Ordinador un MacBook Pro amb sistema operatiu OS X Mavericks.
- Software per poder programar IntelliJ IDEA 13.
- Navegador FireFox amb l'aplicació FireBug.

No es van necessitar eines extres per el desenvolupament del projecte.

Els costos inicials per posar en producció una aplicació són baixos, ja que les tarifes de *hostings* són baixes, com a contrapunt si l'aplicació creixés en número d'usuaris actius en línia els costos de producció es veurien incrementats de forma considerable per poder donar un òptim servei als clients.

2.1 COST DESENVOLUPAMENT

Per calcular el cost del desenvolupament farà servir la següent formula:

$$Cost = cost\ inicial + cost\ extra + cost\ programació$$

El cost inicial ha sigut zero, ja que abans de començar a programar disposava de totes les eines per començar.

$$Cost\ inicial = 0\ €$$

Durant el desenvolupament del projecte no s'ha tingut que fer cap despesa econòmica extra.

$$Cost\ extra = 0\ €$$

En el cost de programació és on hi ha la despesa econòmica, el desenvolupament del programari ha sigut mitjançant només un programador. La formula per el cost de desenvolupament és:

$$Cost\ programació = (hores\ x\ nProgramadors) \times cost\ hora$$

Segons la meua experiència desenvolupant l'aplicació web he estat una mitja de 6h al dia, des del dia 17/03/2014 i el dia de finalització del desenvolupament, 16/05/2014. En total han sigut 41 dies, excloent festius i caps de setmana, que donen un total de 246h i el cost d'un programador junior és aproximadament 7,5€/h.

$$\text{Cost programació} = (246 \times 1) \times 7,5 = 1.845 \text{ €}$$

Per tant el cost total és:

$$\text{Cost} = 0 \text{ €} + 0\text{€} + 1.845 \text{ €} = 1.845 \text{ €}$$

2.2 COST PRODUCCIÓ

Per poder posar l'aplicació web en producció necessitem un servidor d'aplicacions, he analitzat el cost que suposaria posar la xarxa social a OpenShift de RedHad.

OpenShift dona 3 opcions per començar, la que millor s'ajustaria a l'aplicació és la opció *Silver*. Aquesta té uns costos per mes, en la Unió Europea, de 15€/mes, el que equivaldria a 180€/any de despesa fixa.

Tant mateix, hi han costos variables com el numero de connexions, si volem ampliar la memòria del servidor, en la opció *Silver* és de 6 GB.

3. METODOLOGIA

En el projecte s'ha seguit una metodologia *Agile* en el procés de desenvolupament del programari. Aquesta forma de treballar dona una gran avantatge ja que en tot moment es fa el que dona més valor a l'aplicació. (Figura 1)

1. Triar el projecte
2. Estudi i decisions
 - a. S'ha fet una cerca de noves eines per el desenvolupament de software i de metodologies que facilitessin la programació.
 - b. Una cop finalitzada la cerca de les noves eines de treball, s'han comparat amb el *frameworks* ja coneguts per triar els més idonis per el desenvolupament del projecte.
3. Creació d'un llistat amb les tasques més importants de l'aplicació web.
4. Desenvolupament del programari. S'ha dut a terme un procés iteratiu, els quals consta el següents apartats, una vegada finalitzat la iteració es comença de nou.
 - a. Feedback amb l'usuari de l'aplicació per tal que pugui aportar noves tasques per millorar la xarxa social.
 - b. Planificació de la nova iteració, incloent les tasques aportades per l'usuari. Es tria un conjunt de tasques per tal de desenvolupar durant la iteració, que en el meu cas és d'una setmana, i es prioritzen aquelles que aporten més valor al producte.
 - c. Anàlisi de l'estat de l'aplicació per adoptar la millor solució per les noves tasques.
 - d. Desenvolupar les tasques seleccionades.
 - e. Proves per verificar el bon funcionament de les tasques.
 - f. Un cop verificat el seu bon funcionament, s'integren amb els altres mòduls del programari, i seguidament s'inicia una nova iteració.
5. Pujar l'última versió al servidor de producció.

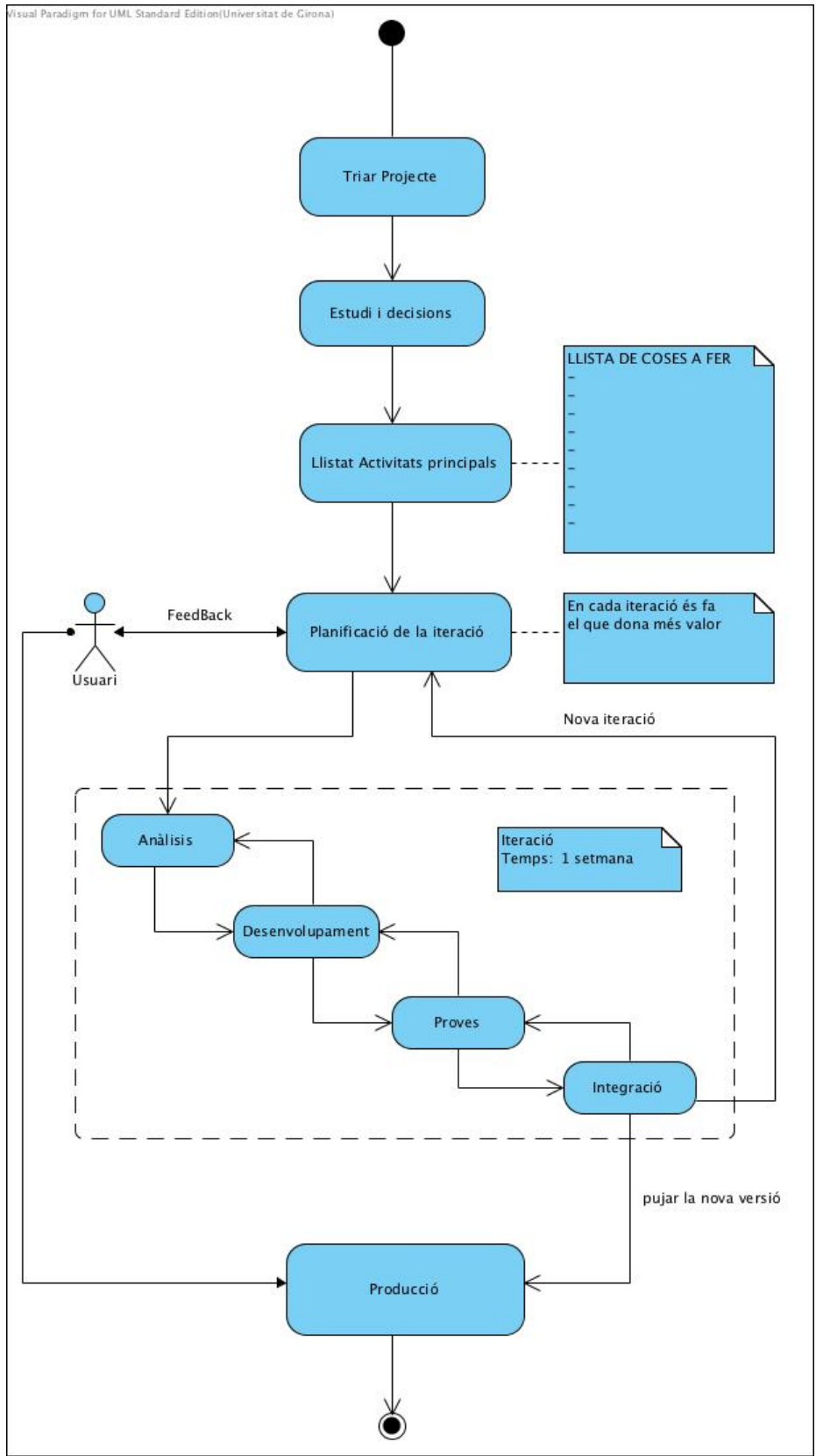


Figura 1, metodologia.

4. PLANIFICACIÓ

El projecte va començar a dissenyar-se a principis del mes de març, les iteracions van començar a mitjans de març, exactament el dia 17.

La etapa de desenvolupament es va dividir en 8 iteracions, de durada d'una setmana, cada una d'elles té diverses tasques planificades, les quals es van decidir en la planificació del dilluns i/o divendres anterior de la setmana.

Abans de començar a programar es van estudiar les eines Backbone i Bootstrap, les quals són frameworks per HTML5 i JavaScript, i es va deixar per més endavant l'estudi de les APIs de YouTube i SoundCloud.

La documentació es va anar fent progressivament durant tot el projecte, amb anotacions del que es feia a cada etapa. D'altra banda el cos de la documentació es va començar una vegada finalitzat la part de programació, per poder tenir una visió amb més experiència sobre el projecte.

4.1 PLANIFICACIÓ INICIAL

La planificació inicial va ser, 2 setmanes de planificació, 8 iteracions d'una setmana (5 dies laborables) per el desenvolupament, i la memòria començar-la després de setmana santa.

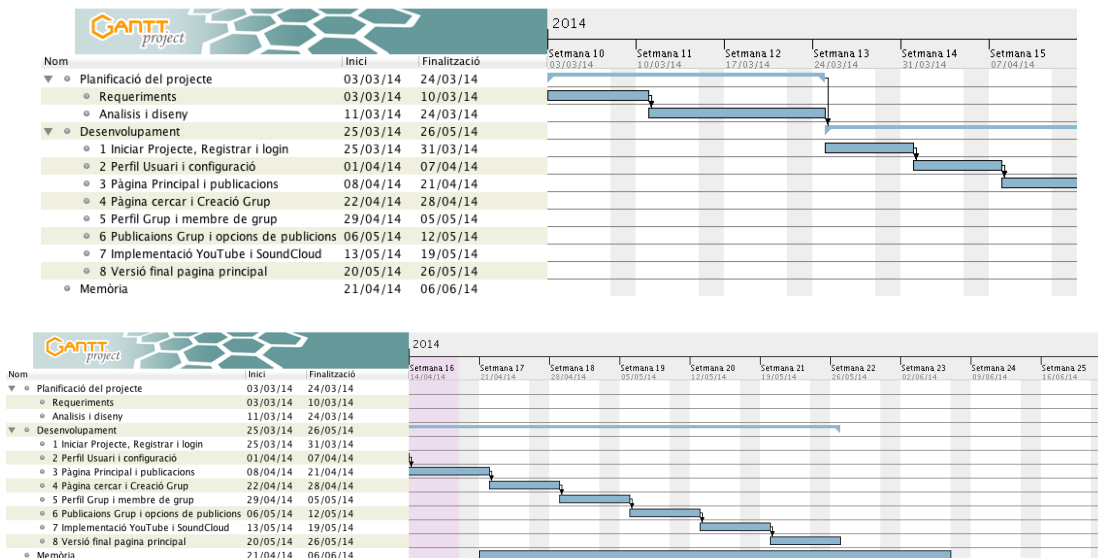


Figura 2, diagrama de Gantt

4.2 ITERACIONS DE PROGRAMACIÓ

En cada iteració s'han fet les tasques que més valor donaven a l'aplicació web, també es van corregir problemes de codi i/o visuals.

Durant aquesta etapa es van anar trobant noves funcionalitat, de gran ajuda, a les eines triades per fer el projecte, que van provocar un millor ús i un estudi amb més profunditat dels frameworks.

També es va estudiar el funcionament de les APIs de YouTube i SoundCloud, aquesta última requeria registrar l'aplicació per poder interaccionar amb el seu servei.

4.2.1 ETAPA DEL 17/03/14 AL 21/03/14

Tasques realitzades:

- Creació del repositori Bitbucket.
- Crear projecte en IntelliJ IDEA.
- Configuració servidor local JBoss.
- Creació de la Base de Dades.
- Implementació de Backbone.
- Registre, *login* i *logout*.

4.2.2 ETAPA DEL 24/03/14 AL 28/03/14

Tasques realitzades:

- Primera versió del perfil.
- Primera versió de la vista general.
- Buscar usuaris.
- Botó seguir.

4.2.3 ETAPA DEL 31/03/14 AL 04/04/14

Tasques realitzades:

- Mostrar i modificar informació de l'usuari.
- Primeres versions del mòdul de publicacions (fer una publicació i mostrar-la).
- Mostrar vídeo de YouTube en publicacions.

4.2.4 ETAPA DEL 7/04/14 AL 11/04/14

Tasques realitzades:

- Correcció del problema “número de amics en comú”.
- Correcció del problema “millora de les vistes amb Bootstrap”.
- Implementació de missatges informatius.
- Creació d'un grup.

4.2.5 ETAPA DEL 21/04/14 AL 25/04/14

Tasques realitzades:

- Creació de la vista del perfil d'un grup.
- Mostrar i modificar la informació de perfil.
- Treure i convidar membres del grup.
- Correcció de bugs menors.

4.2.6 ETAPA DEL 28/04/14 AL 02/05/14

Tasques realitzades:

- Implementació de notificacions.
- Buscar grups.
- Implementació de SoundCloud.
- Publicació de cançons per part de usuaris i grups.
- Llista de cançons dels grups.

4.2.7 ETAPA DEL 05/05/14 AL 09/05/14

Tasques realitzades:

- Seguir i deixar de seguir a un grup.
- Publicacions per part d'un grup.
- “M'agrada” d'un usuari a una publicació.
- Comentar una publicació.
- “M'agrada” d'un usuari a un comentari.
- Compartir una publicació.
- Llistar fàcilment usuaris, publicacions i grups.
- Implementació del sistema de hashtags.
- Llista de publicacions relacionades amb un hashtag.
- Correcció de bugs menors.

4.2.8 ETAPA DEL 12/05/14 AL 16/05/14

Tasques realitzades:

- Llistar els hashtags més populars del dia.
- Mostrar el top 4 de vídeos més populars del dia.
- Posar la foto de perfil.
- Posar la foto de grup.
- Buscar hashtags.
- Millora de les vistes.
- Canvi en la pantalla *search*.

5. MARC DE TREBALL I CONCEPTES PREVIS

En aquest apartat seran detallats els conceptes necessaris per poder entendre el funcionament d'una xarxa social.

5.1 ARQUITECTURA CLIENT-SERVIDOR

Client-servidor és un model d'aplicació distribuïda en la que les tasques es distribueixen entre els proveïdors de recursos o serveis, el servidor, i els demandants d'aquests, clients. El client realitza peticions les quals l'aplicació servidor respon.

En aquesta arquitectura la capacitat de procés està repartida entre els clients i els servidors, això dona una gran avantatge d'organització i centralització de la lògica de negoci i de gestió de la informació.

La separació entre client i servidor és una separació lògica, el servidor no s'executa necessàriament en un sistema totalment diferent que el client.

La xarxa client-servidor és una xarxa de comunicació on els clients estan connectats a un servidor que posa a disposició del client diversos recursos, els quals estan a disposició del client sempre que són sol·licitats.

5.2 WEB SERVICES

Un servei web és una tecnologia que utilitza un conjunt de protocols i estàndards que serveixen per intercanviar dades entre aplicacions. Els serveis web tenen com avantatge que diferents aplicacions, i desenvolupades en diferents llenguatges, poden utilitzar un servei web per la comunicació entre elles. Això és gracies a l'ús d'estàndards oberts, les organitzacions com OASIS i W3C són els responsables de l'arquitectura i reglamentació dels serveis web.

5.2.1 REST

REST, *REpresentation State Transfer*, és un tipus de servei web que es recolza en l'estàndard de HTTP. Ens permet crear serveis i aplicacions que poden ser utilitzades per qualsevol client que entengui HTTP, el que fa més fàcil el desenvolupament d'aplicacions.

El seu funcionament és senzill, mitjançant una URL, única, li podem fer peticions amb els mètodes HTTP, i respon les peticions amb objectes XML, JSON, HTML, etc.

Mètodes HTTP:

- GET: Consultar informació.
- POST: Crear recursos.
- PUT: Editar recursos.
- DELETE: Eliminar recursos

5.2.2 JSON

JSON, *JavaScript Object Notation*, és un format lleuger per intercanviar dades. JSON és un subconjunt de la notació literal de objectes de JavaScript que no requereix l'ús de XML.

Gràcies a la seva simplicitat s'ha generalitzat el seu ús, especialment com alternativa a XML en AJAX. Una de les avantatges de JSON sobre XML com a format d'intercanvi de dades és la seva simplicitat a l'hora de analitzar-lo sintàcticament.

5.3 APLICACIÓ SINGLE-PAGE

En una aplicació *single-page (SPA)* es carrega tot el contingut, ja sigui HTML, CSS, JavaScript, etc. només una vegada, els continguts es van mostrant dinàmicament a la pàgina, segons sigui necessari, generalment com a resposta d'accions dels usuaris. La pàgina no es carrega en cap moment, ni tampoc transfereix el control a una altra pàgina, poden proporcionar la percepció i la usabilitat de diferents pàgines lògiques. L'ús d'una sola pàgina implica una comunicació dinàmica amb un servidor.

5.4 XARXES SOCIALS

Una xarxa social són estructures socials compostes per un grup de persones i/o entitats relacionades entre elles, aquestes relacions poden ser d'amistat, professional, interessos comuns, entre altres.

L'estructura d'una xarxa social està composta per un graf, on els nodes són les persones o entitats i les arestes són relacions entre els nodes. A partir d'aquest principi, una xarxa social pot operar en molts nivells, les arestes poden tenir diferent criteri.

5.5 HISTORIA DE LES XARXES SOCIALS

Les xarxes socials són un servei modern, amb poca historia, i en els últims anys s'ha incrementat el seu èxit, i s'ha massificat el seu ús entre la gent, tot i que el seu origen és de fa més d'una dècada.

Ofereixen als usuaris un lloc comú on poder interactuar fàcilment, això en gran part és possible perquè els usuaris poden connectar-se mitjançant un dispositiu mòbil, la qual cosa està marcant la tendència actual.

A continuació una mica d'història de les xarxes socials per anys, i esdeveniments que han influït a les xarxes socials:

1994: Es funda **GeoCities**, una de les primeres xarxes socials tal i com les coneixem ara. L'idea era que els usuaris creessin les seves pròpies pàgines web, i allotjades segons en el barri on vivien els usuaris.

1995: TheGlobe.com dona la possibilitat als seus usuaris de posar les seves experiències en internet i compartir-les amb altres persones amb interessos semblants.

1997: Microsoft llança **AOL Instant Messenger**.

1997: Sixdegrees.com, web que permetia la creació de perfils i llistat de amics.

2002: Es llança **Friendster** pioner en la connexió online de amics reals. Arriba als 3 milions de usuaris en 3 mesos.

2003: Inauguració de **MySpace**, clon de *Friendster*, creada per una empresa de màrqueting *online*, la primera versió va ser feta en 10 dies.

2004: Facebook, feta originalment per connectar estudiants universitaris. El seu llançament va tenir lloc a la Universitat de Harvard, 19.500 estudiants es van apuntar durant el primer mes.

2006: S'inaugura la primera xarxa de *microblogging* **Twitter**.

2008: Facebook avança a *MySpace* com a xarxa social líder en visitants únics al mes.

2011: *Facebook* té 600 milions d'usuaris per tot el món, *MySpace* 260 milions, *Twitter* 190 milions.

2014: *Facebook* aconsegueix més de 1.000 milions de usuaris, *Twitter* 241 milions.

5.6 CLASSIFICACIÓ XARXES SOCIALS

Les xarxes es poden classificar de diverses maneres:

- Segons el públic, l'objectiu i la temàtica.
- Segons la relació entre usuaris.
- Segons la localització.
- Segons la plataforma.

5.6.1 XARXES SOCIALS VERTICALS

Són aquelles que tenen una temàtica definida i el seu objectiu és congregar, entorn a la temàtica definida, un col·lectiu concret.

Segons la seva especificació es poden classificar entre:

- Xarxes socials verticals professionals, estan dirigides a generar relacions professionals entre els usuaris.
- Xarxes socials verticals d'Oci, l'objectiu és congregar usuaris amb el mateix interès d'oci.
- Xarxes socials verticals mixtes, ofereixen tant als usuaris com a les empreses un entorn específic per activitats professionals i personals.

5.6.2 XARXA SOCIAL WEB

La seva plataforma de desenvolupament està basada en una estructura típica de web.

6. REQUISITS DEL SISTEMA

6.1 REQUERIMENTS FUNCIONALS

En aquest apartat es descriuen els serveis que oferirà l'aplicació web, sense tenir en compte la seva implementació.

En l'aplicació es pot distingir 4 tipus de usuaris.

- Usuari no registrat
- Usuari registrat
- Membre fundador o administrador
- Membre bàsic

6.1.1 USUARI NO REGISTRAT

Registrar: L'usuari no registrat pot registrar-se, per dur a terme aquesta activitat l'usuari tindrà que posar el seu nom, cognom, adreça electrònica, i la contrasenya.

6.1.2 USUARI REGISTRAT

Autenticació: L'usuari podrà entrar a l'aplicació web, una vegada registrat, posant l'adreça electrònica i la contrasenya correctament.

Editar la informació de l'usuari: L'usuari pot modificar la seva informació personal, tanmateix podrà decidir si la informació és: privada, dirigida als seguidors o a tothom.

Seguir i deixar de seguir a un usuari i/o grup: Podrà seguir i deixar de seguir als usuaris i als grups.

Buscar usuaris, grups i hashtags: L'usuari pot buscar fàcilment usuaris, grups i hashtags, els resultats es mostraran en pantalla, si el resultat són usuaris o grups tindrà una breu descripció juntament amb la foto, en el cas dels hashtags es mostraran els més semblants al input cercat.

Publicacions: Les publicacions dels usuaris o grups seguits i les pròpies es mostraran, l'usuari podrà fer publicacions amb vídeos de YouTube o SoundCloud, tanmateix podrà comentar les publicacions, posar "m'agrada" als comentaris i/o publicacions, i compartir-les. Aquestes només poden tenir 140 caràcters.

Creació d'un grup: L'usuari pot crear grups de música.

6.1.3 MEMBRE FUNDADOR I ADMINISTRADOR

El fundador i l'administrador poden fer tot el que fa un usuari bàsic.

Convidar i expulsar als membres del grup: El fundador i l'administrador poden enviar sol·licituds per formar part del grup, també es pot canviar el rol que tenen els usuaris i expulsar-los.

Posar cançons: Poden posar cançons de SoundCloud.

Crear esdeveniment: El fundador i els administradors poden crear esdeveniments.

6.1.4 MEMBRE BÀSIC

Publicacions: Fer publicacions amb vídeos de YouTube o SoundCloud, tanmateix podrà comentar les publicacions. La publicacions només poden tenir 140 caràcters.

Comentaris: Fer comentaris amb nom del grup. Els comentaris només poden tenir 140 caràcters.

Modificar la informació: Es pot modificar la informació del grup.

6.2 REQUERIMENTS NO FUNCIONALS

6.2.1 CLIENT

El client no té pràcticament cap restricció, qualsevol ordinador amb connexió a internet i un navegador que suporti HTML5, pot utilitzar l'aplicació web. L'única desavantatge que té utilitzar frameworks relativament nous, o simplement utilitzar HTML5, és que hi han navegadors, la majoria de ells versions antigues, que no són compatibles amb les noves tecnologies. Un exemple són les versions de Internet Explorer de Windows XP, i no és un col·lectiu minoritari. Per donar una estadística dels navegadors que poden suportar l'aplicació. Aquesta gràfica mostra els navegadors que són compatibles amb l'API *JSON parsing*, una eina essencial per la comunicació amb el servidor. Com es pot comprovar les versions inferiors a Internet Explorer 8.0 no suporten l'aplicació web, és el mateix cas que les versions inferiors de Safari 5.1.

■ = Supported ■ = Not supported

# JSON parsing - other		*Usage stats: Global								
Method of converting JavaScript objects to JSON strings and JSON back to objects using <code>JSON.stringify()</code> and <code>JSON.parse()</code>		Support:	92.63%							
Show all versions	IE	Firefox	Chrome	Safari	Opera	IOS Safari	Opera Mini	Android Browser	Blackberry Browser	IE Mobile
								2.1		
								2.2		
								2.3		
			31.0					3.0		
	8.0		32.0			3.2		4.0		
	9.0		33.0					4.1		
	10.0	28.0	34.0	5.1				4.2-4.3	7.0	
Current	11.0	29.0	35.0	7.0	21.0	7.0	5.0-7.0	4.4	10.0	10.0
Near future		30.0	36.0		22.0					
Farther future		31.0	37.0		23.0					
3 versions ahead		32.0	38.0							

informació extreta de <http://www.caniuse.com>

7. ESTUDIS I DECISIONS

En aquest apartat s'expliquen les decisions que s'han adoptat pel desenvolupament del projecte.

7.1 SERVIDOR

7.1.1 JAVA EE

Java EE o Java Enterprise Edition és una plataforma de programació Java per desenvolupar aplicacions basades en el llenguatge Java. Aquestes aplicacions funcionen sobre servidors d'aplicacions, en aquest cas JBoss AS7.1.1. Java EE té una gran quantitat de llibreries. Per la creació de l'API REST volia fer-la amb Java, per la qual cosa vaig decidir fer servir Java EE. Aquesta plataforma té un ús massificat, hi ha molta informació sobre la programació amb Java EE. Per la part de persistència de dades també es fa servir, el framework de treball, Hibernate, implementació de l'API JPA de Java EE. La decisió de fer servir Java EE és per la gran informació que es troba per fer-lo servir, és una eina usada per grans empreses.



7.1.1.1 Hibernate ORM

Per la capa de persistència necessitava interaccionar amb la base de dades MySQL per poder agafar dades i modificar-les. Hibernate és una bona eina, desenvolupada per RedHat, que ens permet utilitzar la base de dades des de Java, implementa l'API JPA de Java EE. És una eina que mapeja els atributs d'una taula relacional a objectes en Java i dona molta facilitat a la lectura com d'escriptura d'una base de dades mitjançant la programació en Java.

Les raons per utilitzar Hibernate ORM, és per la seva utilització en altres projectes tant en àmbit professional com acadèmic, també és una implementació de l'API JPA per part del servidor d'aplicacions que es fa servir.



7.1.1.2 RESTEasy

RESTEasy és una eina creada per JBoss que implementa l'API JAX-RS de Java EE. Serveix per la creació d'aplicacions Java RESTFull.

La raó per utilitzar-lo és el seu ús a nivell acadèmic i professional, i com Hibernate és una implementació de l'API JAX-RS per part de JBoss.



7.1.2 SERVIDOR D'APLICACIONS

Per desenvolupar el projecte s'ha optat per un servidor local JBoss AS7.1.1 de RedHat. L'experiència amb la utilització d'aquest servidor a decantat la decisió en fer-lo servir per aquest projecte.

La decisió de fer servir JBoss és perquè és un servidor d'aplicacions Java EE. JBoss permet tenir l'aplicació servidor, i la client en un mateix servidor. La decisió també ve lligada per el seu ús en projectes acadèmics.



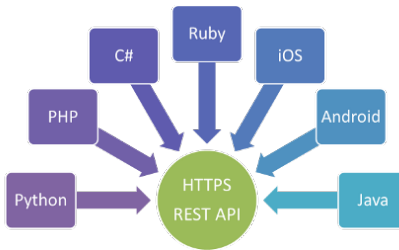
7.1.3 BASE DE DADES

MySQL va ser elegit com a sistema gestor de base de dades per ser una eina lliure i l'ús és gratuït. Es va pensar des del principi en l'ús d'una sistema NoSQL però es necessitava tenir un coneixements previ que el temps material per crear el projecte no proporcionava.



7.1.4 API REST

En la capa del servidor s'ha optat per fer servir una API REST, aquesta opció dona un gran ventall d'oportunitats per crear una aplicació client. La utilització d'un servei API REST fa que en l'aplicació client pugui haver diferents arquitectures, iOS, Android, web, entre altres. Les aplicacions client només tindrien que fer peticions al servidor per demanar dades, tanmateix poden fer peticions per modificar dades de la base de dades del servidor i aquests canvis es veurien afectats als altres clients independent de la arquitectura d'aquest.



7.2 APLICACIÓ CLIENT

En la part client s'ha optat per fer una aplicació web. Com a altres alternatives hi havien aplicació Android o iOS. En aquest cas es va optar per una aplicació web ja que qualsevol persona amb un dispositiu connectat a internet i una navegador pot fer servir l'aplicació. Una altre raó de pes va ser aprendre a desenvolupar una aplicació web, ja que tinc una nulla experiència amb la programació web, més concretament amb el llenguatge JavaScript.

JavaScript

7.2.1 FRAMEWORKS

7.2.1.1 jQuery

jQuery és una eina indispensable a l'hora de programar en JavaScript, dona una gran ventall de possibilitats i fa molt més còmode la programació amb aquest llenguatge. Moltes llibreries destinades a JavaScript necessiten jQuery, així que és una eina essencial. També jQuery té molts de *widgets* de gran utilitat per la interfície gràfica i un *sub-framework*, jQueryUI, destinat també a la interfície gràfica, el qual no he utilitzat per el projecte.



7.2.1.2 Backbone

Un dels grans problemes d'una aplicació web és la organització i la estructura que ha de seguir. Backbone és una eina de treball de les més completes i utilitzades per resoldre aquesta mancança. Proporciona les eines suficients per fer una aplicació *single-page*. Backbone et proporciona directrius per organitzar l'aplicació web i eines per fer un MVC, model-vista-controlador.

Per poder utilitzar Backbone s'ha de fer servir 2 llibreries més: `_underscore` i `requireJS`.

`_underscore` té un paper important a l'hora de carregar dinàmicament parts de la interfície. El mecanisme és senzill, en l'aplicació web tens uns *templates* que gracies a `_underscore` pots posar-los en la interfície, substituir-los i recarregar-los fàcilment.

RequireJS és opcional, però molt recomanable, t'ajuda a importar altres mòduls de l'aplicació, templates, llibreries, entre altres coses, a un mòdul concret.



7.2.1.3 Bootstrap

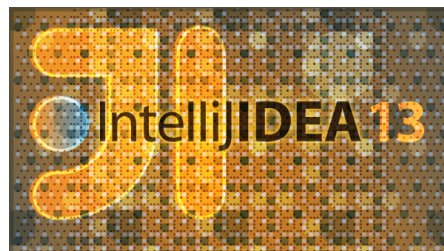
Bootstrap és un framework creat i utilitzat per Twitter en el qual té plantilles fetes dels elements d'una pàgina web per tal de facilita el disseny web. Un dels grans problemes a l'hora de visualitzar l'aplicació en un navegador web és la mida del monitor, Bootstrap proporciona un control sobre la mesura de la finestra i mostrar el contingut de la manera més oportuna segons els píxels de la pantalla, això resol en gran mesura el problema de la mida en que es mostra la interfície.



7.3 ENTORN DE TREBALL

7.3.1 INTELLIJ IDEA 13

L'entorn de treball per programar a sigut IntelliJ IDEA 13 i el navegador FireFox amb l'aplicació FireBug. Les dos aplicacions ja les tenia instal·lades al meu ordinador personal i són les que utilitzo des de 3r de carrera per programar amb Java.



7.3.2 BITBUCKET

Per el control de versions del projecte s'ha fet servir l'eina Bitbucket amb GIT.



8. ANÀLISI I DISSENY DEL SISTEMA

En aquests capítol hi haurà una descripció general de l'aplicació, identificarem els actors que participen, mostrarem el diagrama de casos d'us de cada actor i la fixa de cada cas d'us. Finalment presentarem el disseny de l'aplicació.

8.1 DESCRIPCIÓ GENERAL

L'Aplicació que volem desenvolupar en aquest projecte és una xarxa social orientada a la música on els usuaris puguin interaccionar amb altres usuaris i grups, amb publicacions i comentari d'aquests, d'un màxim de 140 caràcters, posant m'agrada a publicacions i/o comentaris o compartint-los. En les publicacions es podrà posar enllaços de YouTube i SoundCloud que el sistema detectarà, i posarà el vídeo o cançó en la publicació, també es podrà posar hashtags per dinamitzar i trobar temes de "moda" en aquell moment.

Els usuaris podran crear grups de música lliurement. Els grups tindran la seva pròpia pàgina de perfil, també podran fer publicacions i/o comentaris, posar vídeos, cançons, entre altres coses. També disposaran d'un apartat per poder posar els seus esdeveniments més importants.

8.2 ACTORS

A continuació s'identifiquen els actors de l'aplicació. Un actor és una entitat externa (persona, sistema, subsistema, etc.) que interactua amb el sistema interpretant un determinat rol o estat.

En l'aplicació es poden distingir 4 actors clarament diferenciats. Els 4 tipus de usuaris que poden existir en aquesta aplicació són: usuari no registrat, usuari registrar, membre bàsic i membre administrador.

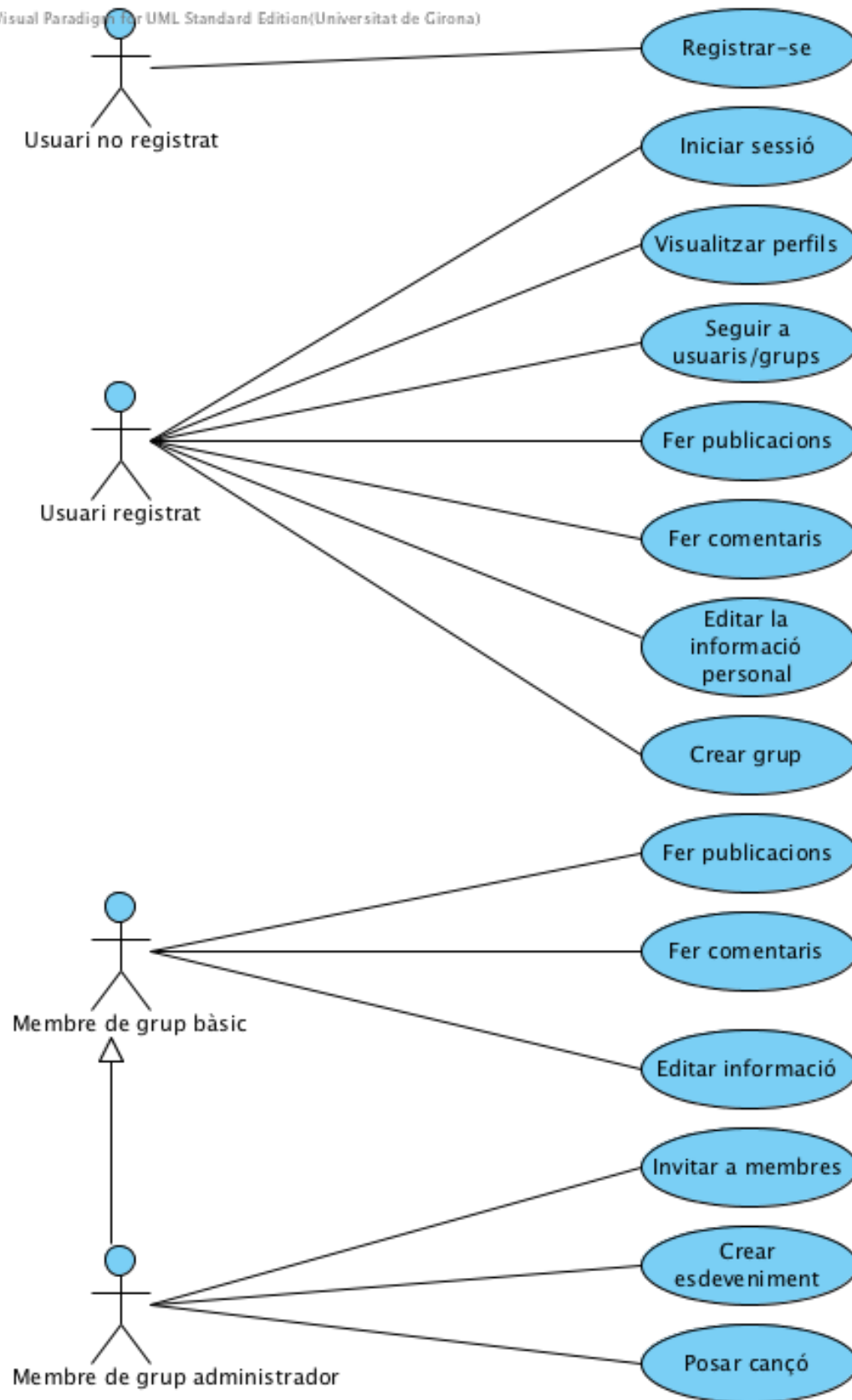


Figura 3, Diagrama de casos d'us.

8.2.1 USUARI NO REGISTRAT

L'usuari no registrat que no pertany en la comunitat de l'aplicació. En aquesta situació només pot registrar-se.

8.2.1.1 Registrar

L'usuari tindrà que registrar-se si vol pertànyer a la comunicat de usuaris de l'aplicació. Per dur a terme aquest procés, ha d'anar a la pàgina de autenticació i omplir les dades mínimes que es requereix: nom, cognoms, adreça electrònica i una contrasenya (la contrasenya es repetirà dues vegades, i a de tenir com a mínim una lletra majúscula, una lletra minúscula i un número). Una vegada omplerts els camps l'usuari tindrà que donar al botó *Enviar*, en el cas que no hi hagi error en el formulari per registrar-se aquest accedirà directament dins l'aplicació, en cas contrari se li notificarà l'error.

Cas d'us	Registrar
Descripció	Un usuari ha de registrar-se per poder formar part de la comunitat de SocialSong.
Actor	Usuari no registrat.
Precondició	Cap.
Flux Principal	<ol style="list-style-type: none">1. L'usuari inicia el programa.2. El sistema carrega la pàgina d'autenticació.3. L'usuari omple el formulari de registre.4. L'usuari prem el botó "Enviar".5. El sistema comprova les dades6. El sistema registre l'usuari i carrega la pàgina principal.
Fluxos alternatius	<ol style="list-style-type: none">6. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	S'ha crear un usuari nou amb les dades aportades per l'usuari.

Diagrama d'activitat: Registrar

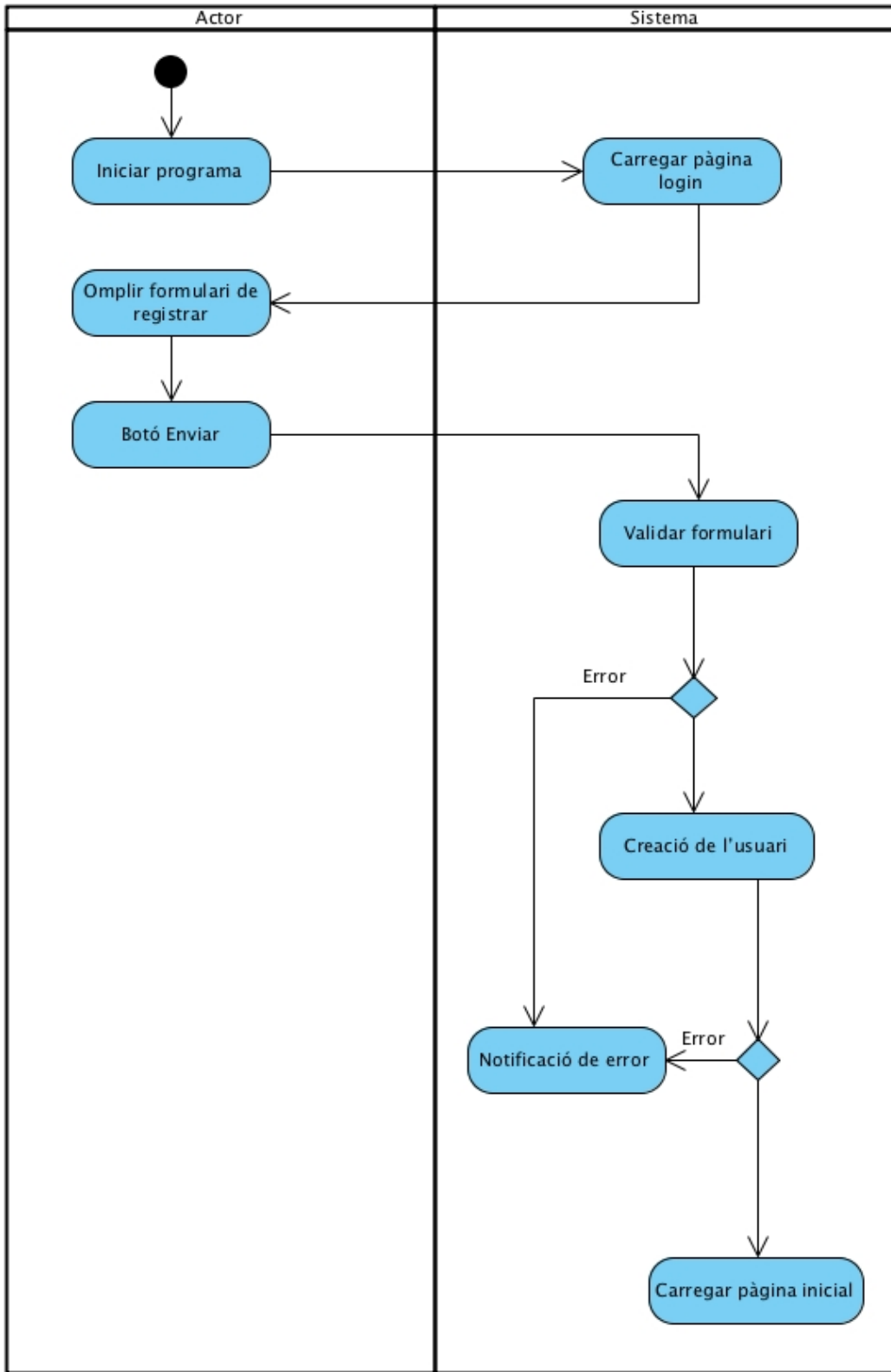


Figura 4, Diagrama d'activitat: registrar Creació de l'usuari

8.2.2 USUARI REGISTRAR

L'usuari registrat és aquell que està registrat en el sistema. Les activitats més importants que pot fer en aquesta situació són: visualitzar perfils, fer publicacions i comentaris, modificar la informació i crear un grup.

8.2.1.1 Autenticació

L'usuari, prèviament registrat, per entrar a l'aplicació tindrà que omplir el formulari de *login*, una vegada omplert, amb l'adreça electrònica i la contrasenya, tindrà que clicar el botó "*Iniciar Sessió*". Una vegada fet aquest procediment si l'adreça electrònica i la contrasenya coincideixen entrarà directament a l'aplicació, en cas contrari se li notificarà que hi ha un error.

8.2.2.2 Visualitzar perfils

L'usuari podrà visualitzar els perfils d'altres usuaris i grups. Hi han moltes maneres d'arribar als perfils, alguns exemples són buscant el nom en la barra superior de l'aplicació o simplement clicant sobre el nom de l'usuari o grup en una publicació, en aquesta última opció es veurà reflectit l'enllaç que dirigeix al perfil.

8.2.2.3 Seguir usuaris i grups

Per seguir un grup o un usuari és un procés simple. En la pantalla de perfil es podrà veure fàcilment el botó per seguir el grup o usuari en qüestió. També hi ha una altre via per fer seguir un grup, en la barra superior de l'aplicació podem buscar el nom de l'usuari o grup, i en el resum dels resultats trobades hi ha el botó de seguir.

8.2.2.4 Fer publicacions

L'usuari podrà fer publicacions des de la pàgina principal, o des de qualsevol perfil, es té que tenir en compte que si fem una publicació des de qualsevol altre perfil que no sigui el nostre, la publicació estarà dirigida a l'usuari o grup al qual pertany el perfil. La publicació només podrà tenir 140 caràcters, dintre de aquests caràcters es poden posar enllaços i hashtags. Si aquests links són de YouTube o de SoundCloud es mostrarà una visualització en la publicació, en cas que hi hagin més d'un enllaç, es mostrarà només el primer.

8.2.2.5 Fer comentaris

En una publicació es pot comentar, entre altres opcions. L'usuari per poder dur a terme un comentari tindrà que clicar el botó *respondre* de la publicació desitjada. En el comentari només pot tenir 140 caràcters. Una vegada fet el comentari, per poder publicar-lo, es pot prémer la tecla *enter* o el botó enviar.

8.2.2.6 Editar informació personal

Per editar la informació general o els paràmetres de privacitat del perfil s'ha d'anar a la pantalla de configuració, per accedir s'ha de fer clic sobre la icona, representada per un engranatge, de la barra superior de l'aplicació. La pantalla de configuració està dividida en seccions, en cada una de elles té atributs els quals podem canviar el contingut i les opcions privacitat (privat, amics i públic).

8.2.2.7 Crear grup

Els usuaris poden crear grups, en l'apartat de configuració, a l'última secció trobarem un botó el qual ens permetrà crear el grup. Una vegada clicat s'obrirà un *modal* amb el formulari per crear un grup. El formulari s'ha de omplir amb el nom de grup, estil de música, data de fundació del grup, la imatge i si volem convidar a més membres. La informació proporcionada es podrà modificar posteriorment si es desitja.

Cas d'us	Autenticació
Descripció	L'usuari ha d'iniciar sessió per poder entrar a l'aplicació.
Actor	Usuari registrat.
Precondició	L'usuari té que estar registrat.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari inicia el programa. 2. El sistema carrega la pàgina d'autenticació. 3. L'usuari omple el formulari d'autenticació. 4. L'usuari prem el botó "Iniciar sessió". 5. El sistema comprova les dades. 6. El sistema inicia sessió de l'usuari i carrega la pàgina principal.
Fluxos alternatius	6. En cas d'error: El sistema mostra l'error a l'usuari.
Postcondició	Sessió de l'usuari iniciada.

Cas d'us	Visualitzar perfils
Descripció	L'usuari podrà veure els perfils de tots els usuaris i grups.
Actor	Usuari registrat.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari clica sobre el nom de l'usuari. 2. El sistema mostra el perfil de l'usuari clicat.
Fluxos alternatius	Cap.
Postcondició	Es mostra el perfil.

Cas d'us	Seguir usuaris o grups
Descripció	L'usuari podrà seguir o deixar de seguir als usuaris i els grups que desitgi.
Actor	Usuari registrat.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari clica el botó de seguir de l'usuari o grup. 2. El sistema crea el vincle de "Seguir". 3. El sistema recarrega el perfil.
Fluxos alternatius	3. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	Es mostrarà el perfil de forma pertinent .

Cas d'us	Fer publicacions
Descripció	<p>Els usuaris poden fer publicacions des de qualsevol perfil i en la pàgina principal. Aquestes publicacions tindran com a màxim 140 caràcters.</p> <p>Si la publicació és feta des de la pàgina principal o des del perfil propi, la publicació estarà dirigida a tothom.</p> <p>Si la publicació a sigut fet des de una altre perfil, estarà destinada a l'usuari o grup del perfil.</p>
Actor	Usuari registrat.
Precondició	<p>Sessió iniciada.</p> <p>Seguidor de l'usuari o grup.</p>
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari es posa sobre la caixa de publicacions. 2. L'usuari escriu el missatge. 3. L'usuari prem el botó "Enviar". 4. El sistema publica el missatge.
Fluxos alternatius	<ol style="list-style-type: none"> 4. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	Crea la publicació.

Cas d'us	Fer comentari
Descripció	Els usuaris poden fer comentaris des de qualsevol perfil i en la pàgina principal. Els comentaris tindran com a màxim 140 caràcters.
Actor	Usuari registrat.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari clica sobre l'opció respondre de la publicació. 2. El sistema mostra la caixa per respondre. 3. L'usuari escriu el comentari. 4. L'usuari prem el botó "Enviar". 5. El sistema comprova el comentari. 6. El sistema publica el comentari.
Fluxos alternatius	6. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	Publicació del comentari.

Cas d'ús	Editar informació personal
Descripció	L'usuari podrà canviar la seva informació personal i la privacitat sobre aquesta.
Actor	Usuari registrat.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari va a la pàgina de configuració. 2. Prem el botó "modificar" de l'apartat desitjat. 3. L'usuari modifica les dades. 4. L'usuari prem el botó guardar. 5. El sistema comprova les dades. 6. El sistema modifica les dades. 7. Actualitzar dades.
Fluxos alternatius	8. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	Modificació de les dades.

Cas d'us	Crear grup
Descripció	Creació d'un grup per part d'un usuari.
Actor	Usuari registrat.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari va a la pàgina de configuració. 2. L'usuari prem el botó "modificar" de l'apartat "Grup". 3. L'usuari prem el botó de "Crear usuari". 4. El sistema mostra el formulari per crear grup. 5. L'usuari omple el formulari. 6. L'usuari prem el botó "Crear". 7. El sistema comprova les dades. 8. El sistema crea el grup i carrega la pàgina de perfil del grup.
Fluxos alternatius	8. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	Crear grup.

Diagrama d'activitat: Autenticació

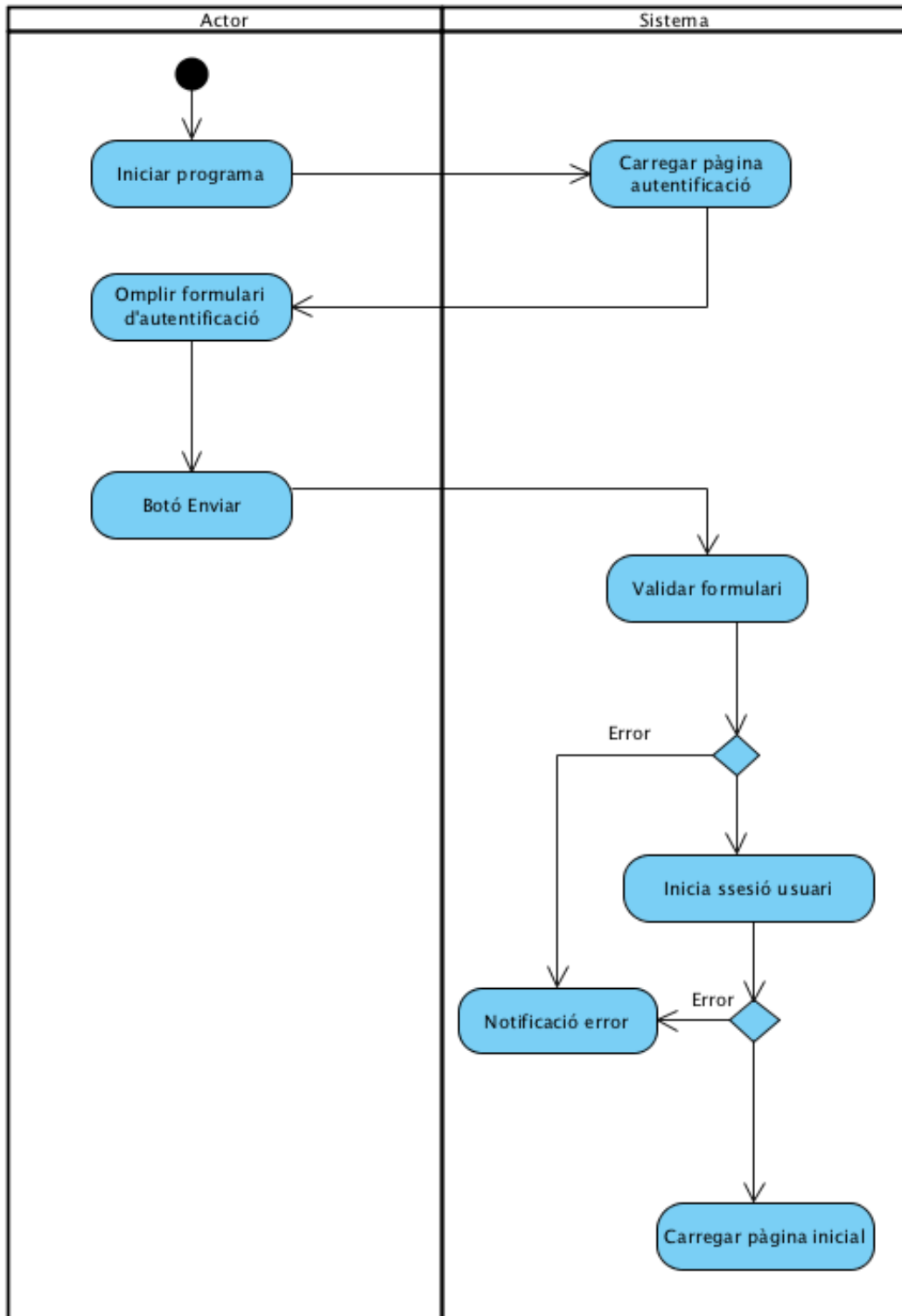


Figura 5, Diagrama d'activitat: autenticació

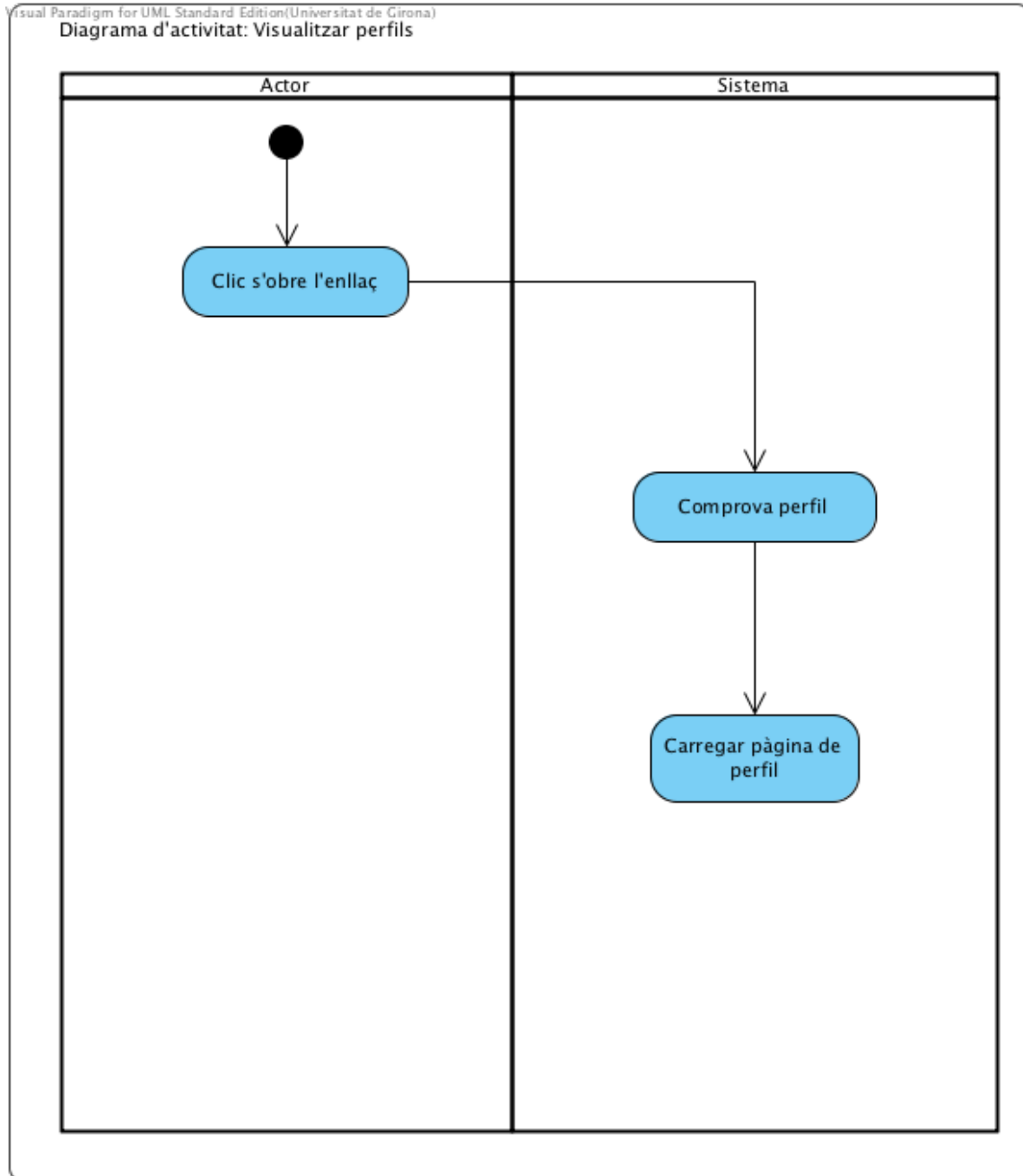


Figura 6, Diagrama d'activitat: visualitzar perfil

Diagrama d'activitat: Seguir usuari o grup

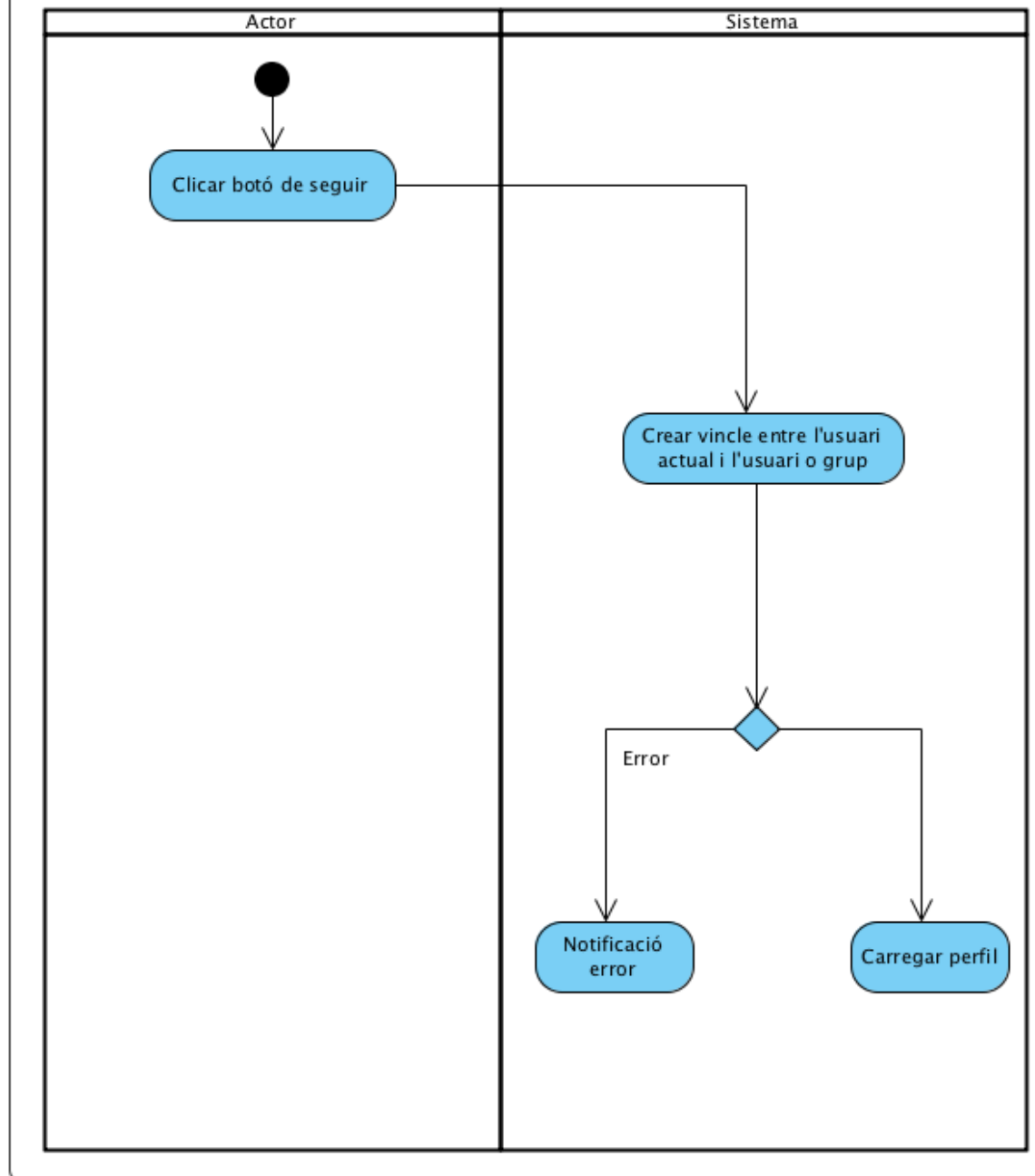


Figura 7, Diagrama d'activitat: seguir a usuari o grup

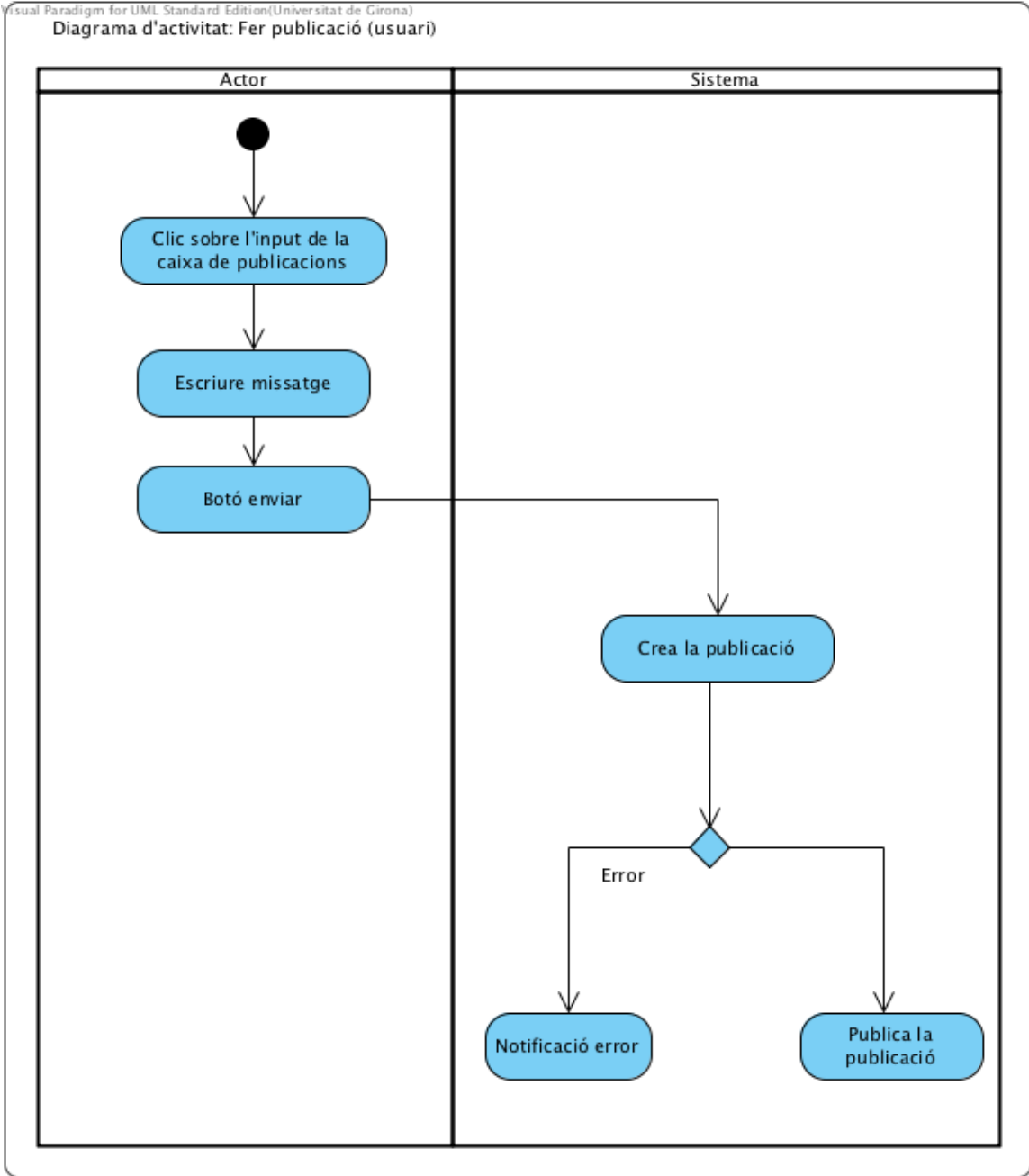


Figura 8, Diagrama d'activitat: fer publicació

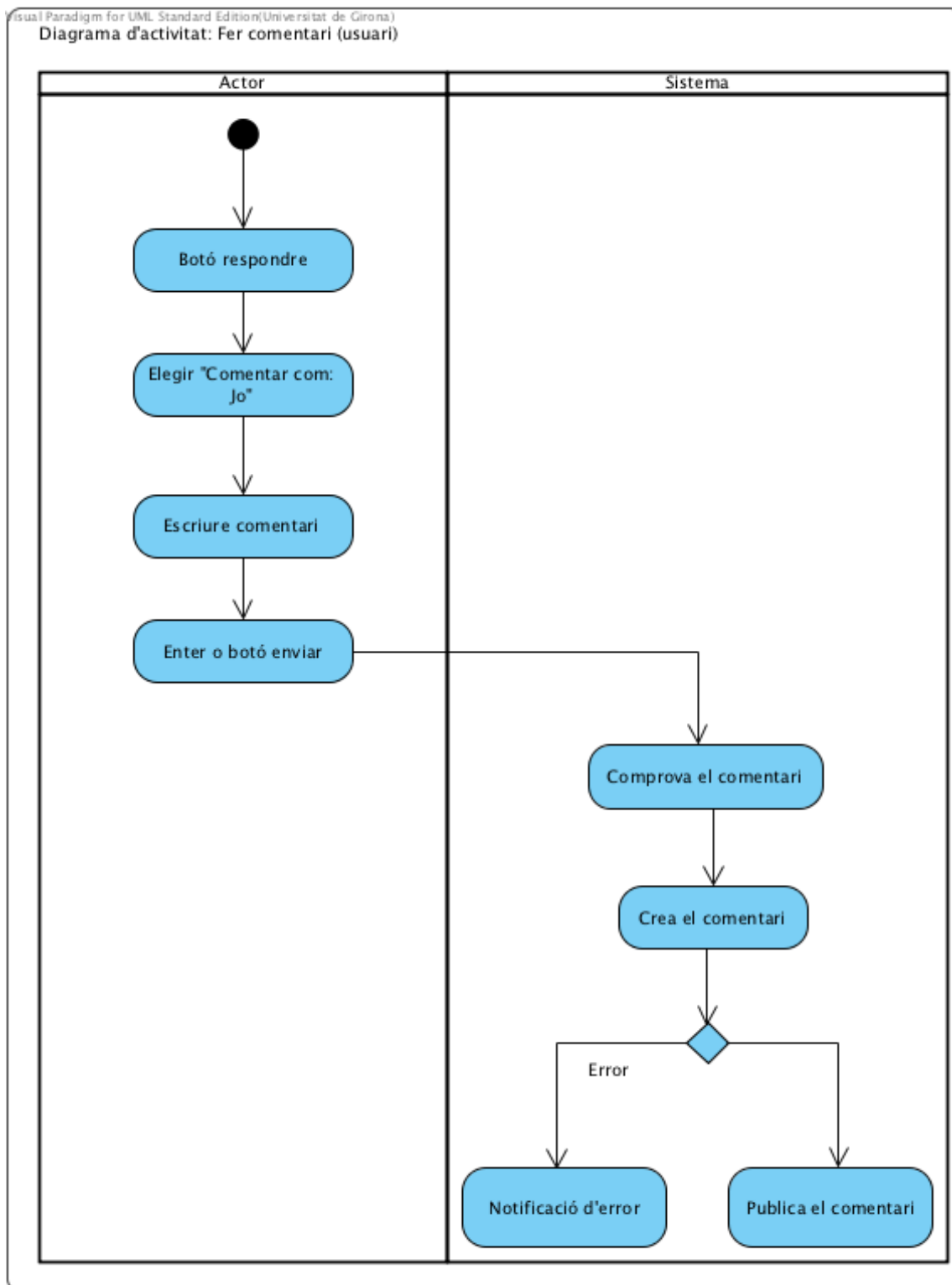


Figura 9, Diagrama d'activitat: fer comentari

Diagrama d'activitat: Editar Informació (usuari)

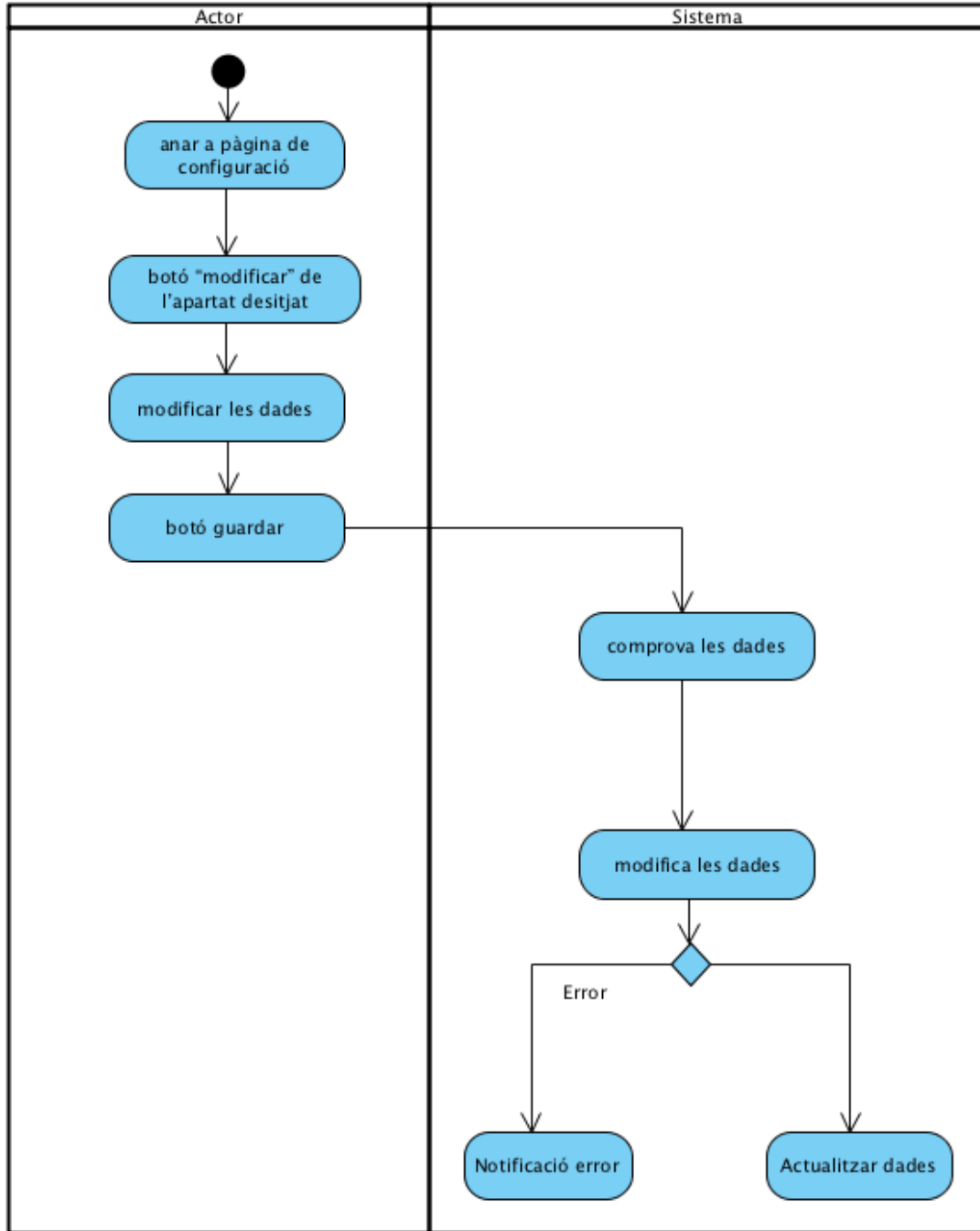


Figura 10, Diagrama d'activitat: Editar informació

Diagrama d'activitat: Crear grup

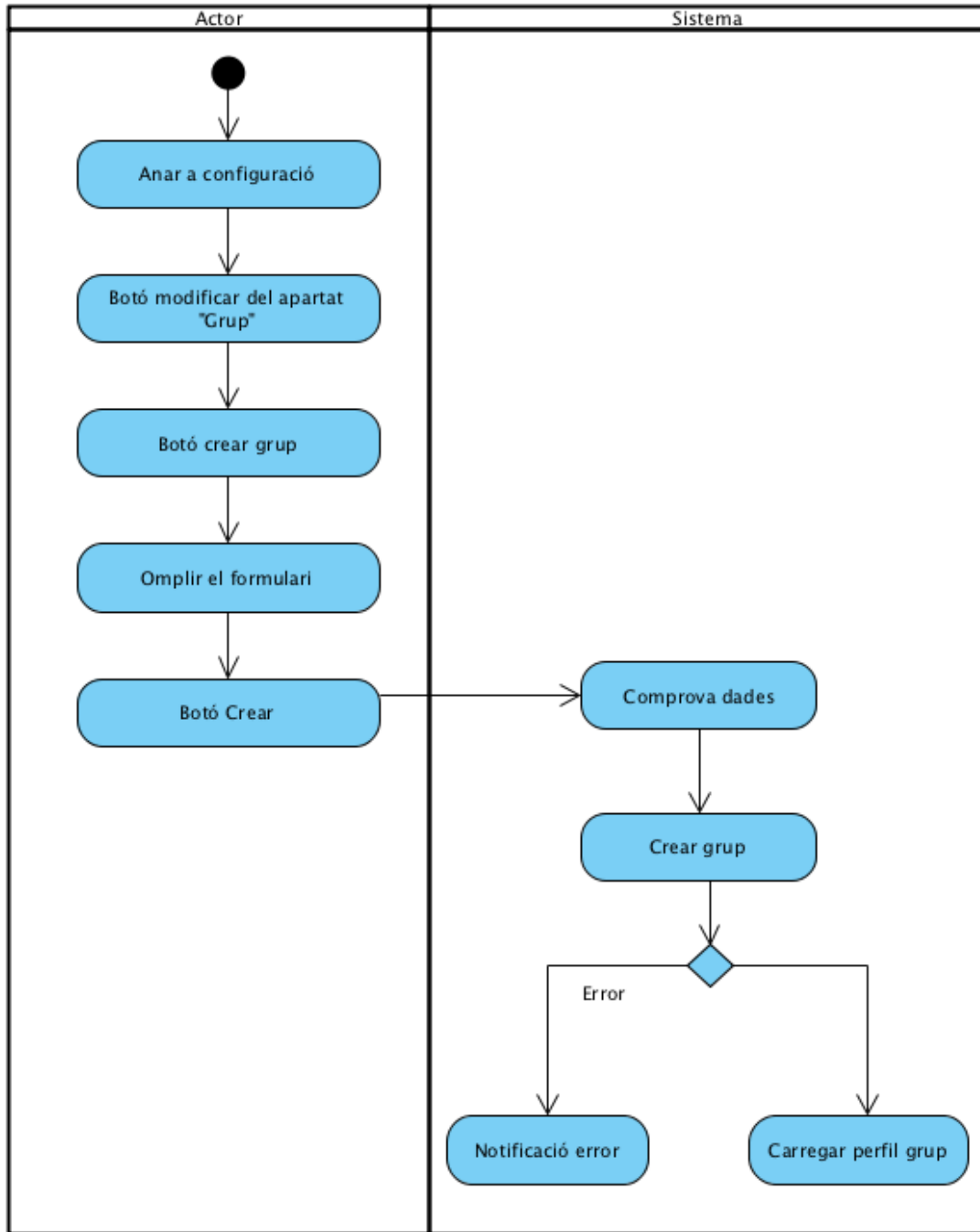


Figura 11, Diagrama d'activitat: Crear grup

8.2.3 MEMBRE DE GRUP FUNDADOR/ADMINISTRADOR

8.2.3.1 Convidar membre

Per poder tenir tots els membres de grups, el fundador i els administradors poden convidar als altres membres del grup, per poder convidar-los cal que l'usuari que vol convidar sigui seguidor de l'usuari que vol convidar. Una vegada convidat es pot canviar el rol del grup (cantant, guitarrista...), canviar de tipus de membre (bàsic o administrador). Una vegada invitat se li notificarà a l'usuari i ell tindrà que respondre la invitació positivament o negativament, en cas positiu l'usuari esdevindrà membre del grup, en cas negatiu s'anul·larà la invitació.

8.2.3.2 Crear esdeveniments

En la pàgina de perfil del grup hi ha un apartat de pròxims esdeveniments, els usuaris administradors i el propi fundador poden crear-ne nous. Es té que tenir en compte que només es mostren els 3 esdeveniments més propers.

8.2.3.3 Posar cançó

Per poder compartir cançons el grup compta amb un apartat a la dreta del perfil de grup, on els administradors i els fundadors poden posar l'enllaç de la cançó de SoundCloud, una vegada introduïda es mostrarà en gran, les altres es poden clicar per poder-les escoltar. Aquesta cançó també apareixerà com a publicació per donar l'oportunitat als usuaris de posar m'agrada o comentar-la.

Cas d'ús	Convidar membre
Descripció	Convidar a un membre per formar part del grup.
Actor	Membre de grup administrador o fundador.
Precondició	Sessió iniciada. Seguir a l'usuari a convidar.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari va a la pàgina de perfil del grup. 2. L'usuari clica sobre la icona d'engranatge. 3. El sistema mostra el formulari. 4. L'usuari clica sobre l'apartat de "Membres". 5. L'usuari clica sobre el botó "+". 6. L'usuari escriu el nom. 7. L'usuari fa doble-clic sobre el nom de l'auto-completar. 8. L'usuari clica sobre la icona de l'avió de paper. 9. El sistema envia la invitació a l'usuari.
Fluxos alternatius	9. En cas d'error: El sistema notifica l'error.
Postcondició	Crear invitació a membre.

Cas d'ús	Crear esdeveniments
Descripció	Crear esdeveniments del grup.
Actor	Membre de grup administrador o fundador.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari va a la pàgina de perfil del grup. 2. L'usuari clica sobre "Crear nou esdeveniment". 3. El sistema mostra el formulari. 4. L'usuari emplena el formulari. 5. L'usuari clica el botó "Enviar". 6. El sistema comprova les dades. 7. El sistema crea l'esdeveniment. 8. Actualitza el llistat d'esdeveniments
Fluxos alternatius	8. En cas d'error: El sistema mostra un missatge d'error.
Postcondició	Crear esdeveniment.

Cas d'us	Posar Cançó
Descripció	Compartir cançons de SoundCloud.
Actor	Membre de grup administrador o fundador.
Precondició	Sessió iniciada. Membre del grup.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari va a la pàgina del perfil de grup. 2. L'usuari escriu l'enllaç el input de l'apartat "Cançons". 3. L'usuari prem enter. 4. El sistema crea la cançó. 5. El sistema actualitza la llista de cançons.
Fluxos alternatius	4. El sistema mostra un missatge d'error.
Postcondició	Compartir cançó.

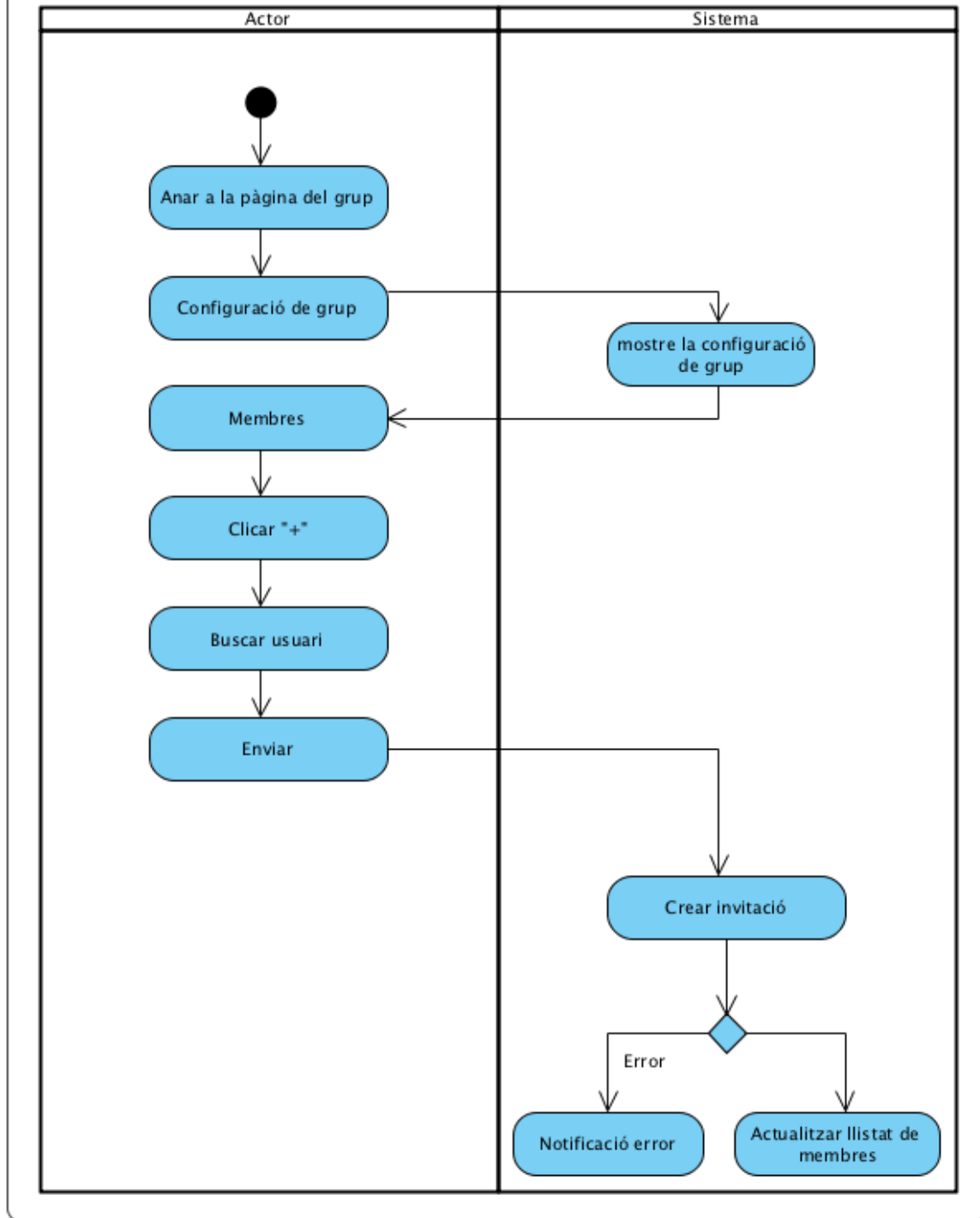


Figura 12, Diagrama d'activitat: Convidar membre

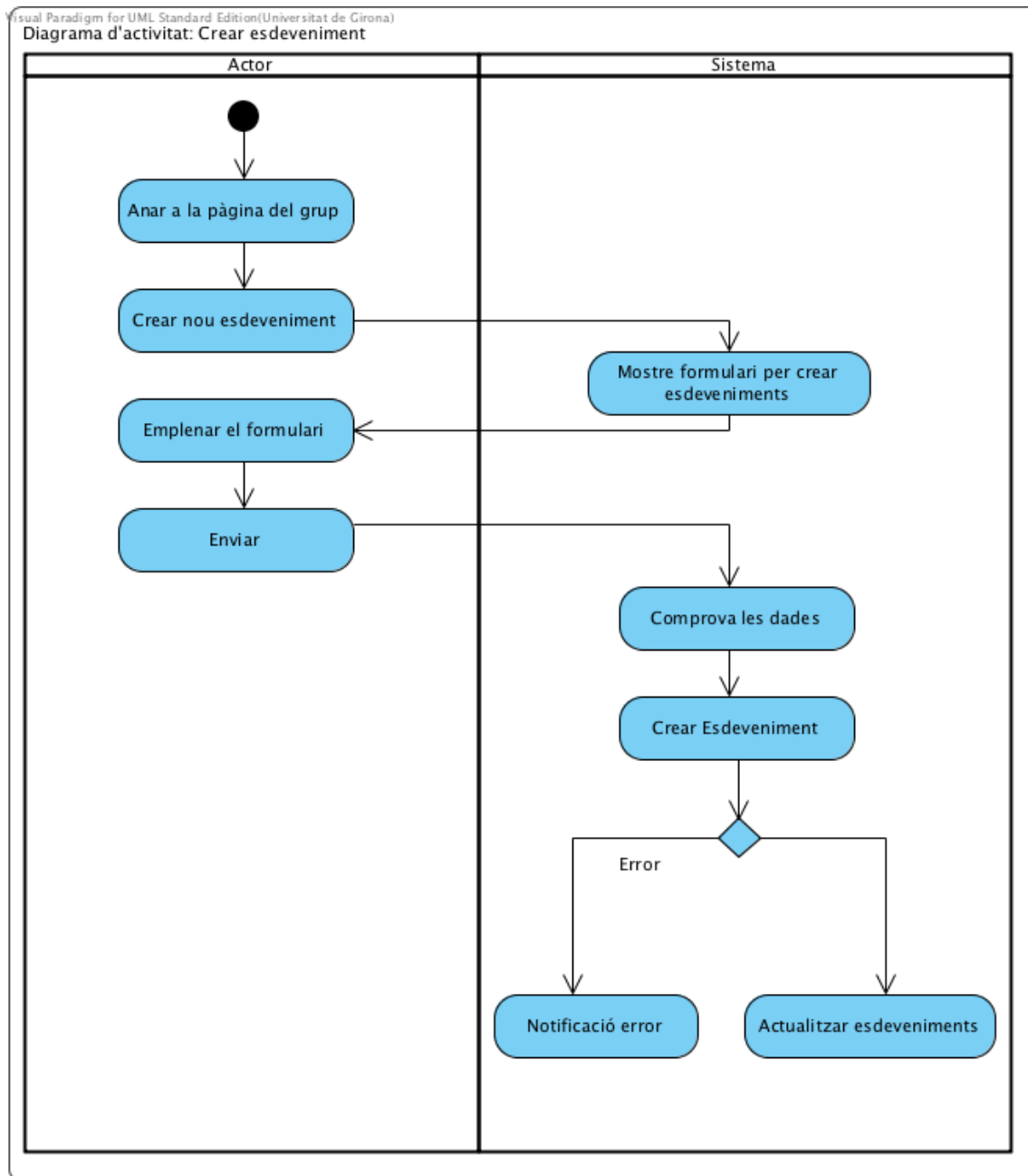


Figura 13, Diagrama d'activitat: Convidar membre

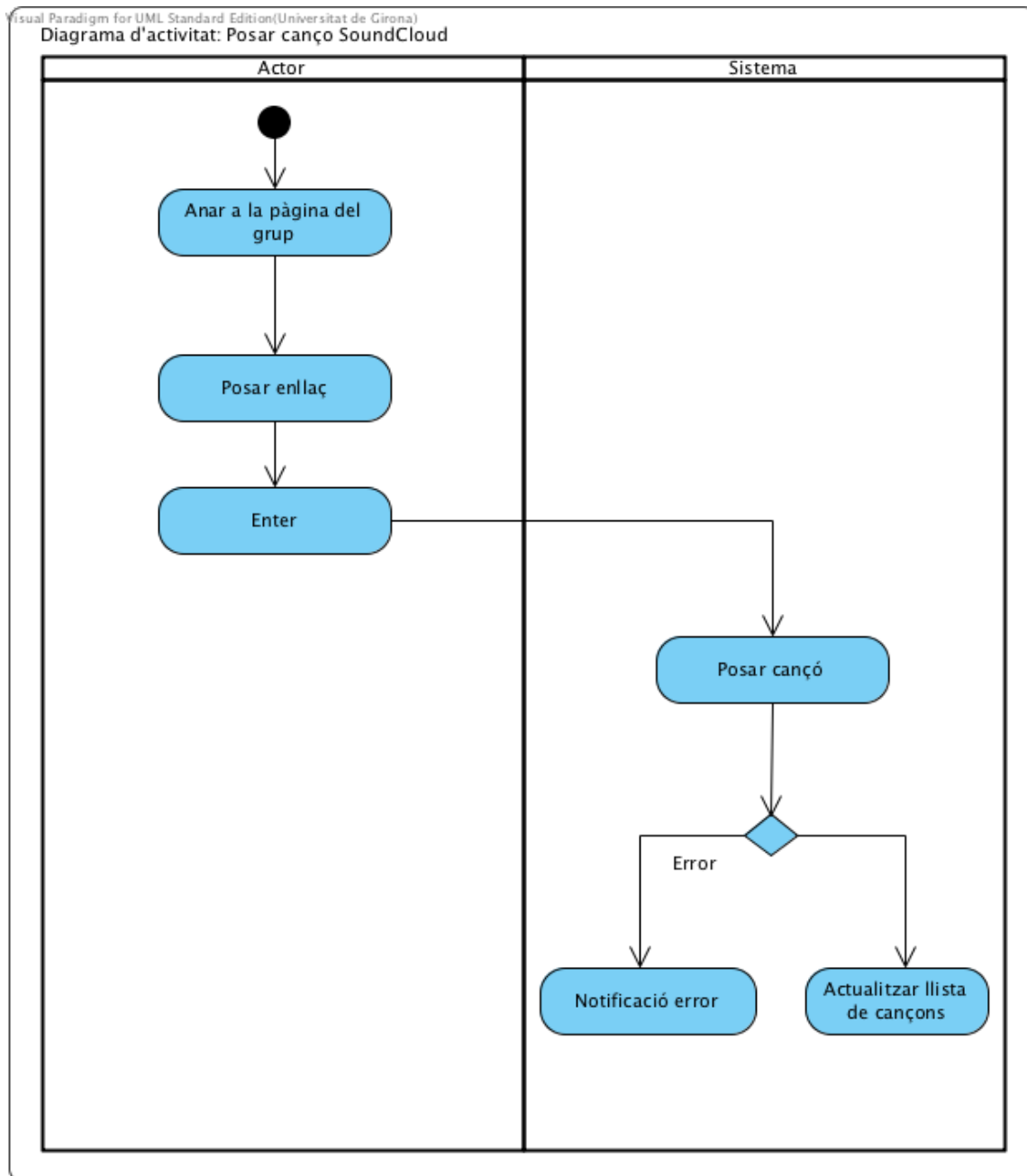


Figura 14, Diagrama d'activitat: Posar cançó

8.2.4 MEMBRE DE GRUP BÀSIC

8.2.4.1 Fer publicació

Els membres d'un grup poden fer publicacions a través del grup. Les publicacions es poden fer des de la pàgina principal, perfil d'usuari o grup. A diferència de fer publicacions per un usuari en la pestanya "*publicar com*" s'elegeix el nom del grup.

8.2.4.2 Fer comentari

Per major interacció entra la comunitat de la xarxa social, els grups (o membres del grup) poden fer comentaris a les publicacions. Per fer un comentari l'usuari membre del grup haurà de clicar a "respondre", posar a "respondre com" el nom de grup i escriure el missatge seguidament tocar "enter" o el botó d'enviar.

8.2.4.2 Editar informació del grup

Per poder completar el perfil del grups o modificar errades es pot canviar la informació del grup. Per dur a terme aquesta tasca s'ha d'anar a configuració del grup, des de la pàgina de perfil del grup, i en la pestanya "General" es pot modificar: nom, estil de musica, data de fundació, descripció i imatge de perfil.

Cas d'us	Fer publicació
Descripció	<p>Els membres de grup poden fer publicacions des de qualsevol perfil i en la pàgina principal. Aquestes publicacions tindran com a màxim 140 caràcters.</p> <p>Si la publicació a sigut feta des de la pàgina principal o des del perfil propi, la publicació estarà dirigida a tothom.</p> <p>Si la publicació a sigut feta des de un altre perfil, aquesta publicació estarà destinada a l'usuari o grup del perfil.</p>
Actor	Usuari registrat.
Precondició	Sessió iniciada.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari es posa sobre la caixa de publicacions. 2. L'usuari escriu el missatge. 3. L'usuari elegeix el nom del grup a "publicar com". 4. L'usuari prem el botó de "Enviar". 5. El sistema publica el missatge.
Fluxos alternatius	<ol style="list-style-type: none"> 5. El sistema mostra un missatge d'error.
Postcondició	Creació de la publicació.

Cas d'us	Fer comentari
Descripció	Els membres de grup poden fer comentaris des de qualsevol perfil i en la pàgina principal. Els comentaris tindran com a màxim 140 caràcters.
Actor	Usuari registrat.
Precondició	Sessió iniciada. Membre del grup.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari clica sobre l'opció respondre de la publicació. 2. El sistema mostra la caixa per respondre. 3. L'usuari escriu el comentari. 4. L'usuari selecciona el nom del grup en "Comentar com:" 5. L'usuari prem el botó "Enviar". 6. El sistema comprova el comentari. 7. El sistema publica el comentari.
Fluxos alternatius	7. El sistema mostra un missatge d'error.
Postcondició	Publicació del comentari.

Cas d'ús	Editar informació del grup
Descripció	Els membre poden modificar la informació del grup.
Actor	Usuari registrat.
Precondició	Sessió iniciada. Membre del grup.
Flux Principal	<ol style="list-style-type: none"> 1. L'usuari va a la pàgina de perfil del grup. 2. L'usuari clica sobre la icona d'engranatge. 3. El sistema mostra el formulari. 4. L'usuari modifica les dades. 5. L'usuari prem el botó de "Guardar". 6. El sistema comprova les dades. 7. El sistema guarda les modificacions.
Fluxos alternatius	7. El sistema mostra un missatge d'error.
Postcondició	Modificació de les dades.

Diagrama d'activitat: Fer publicació (grup)

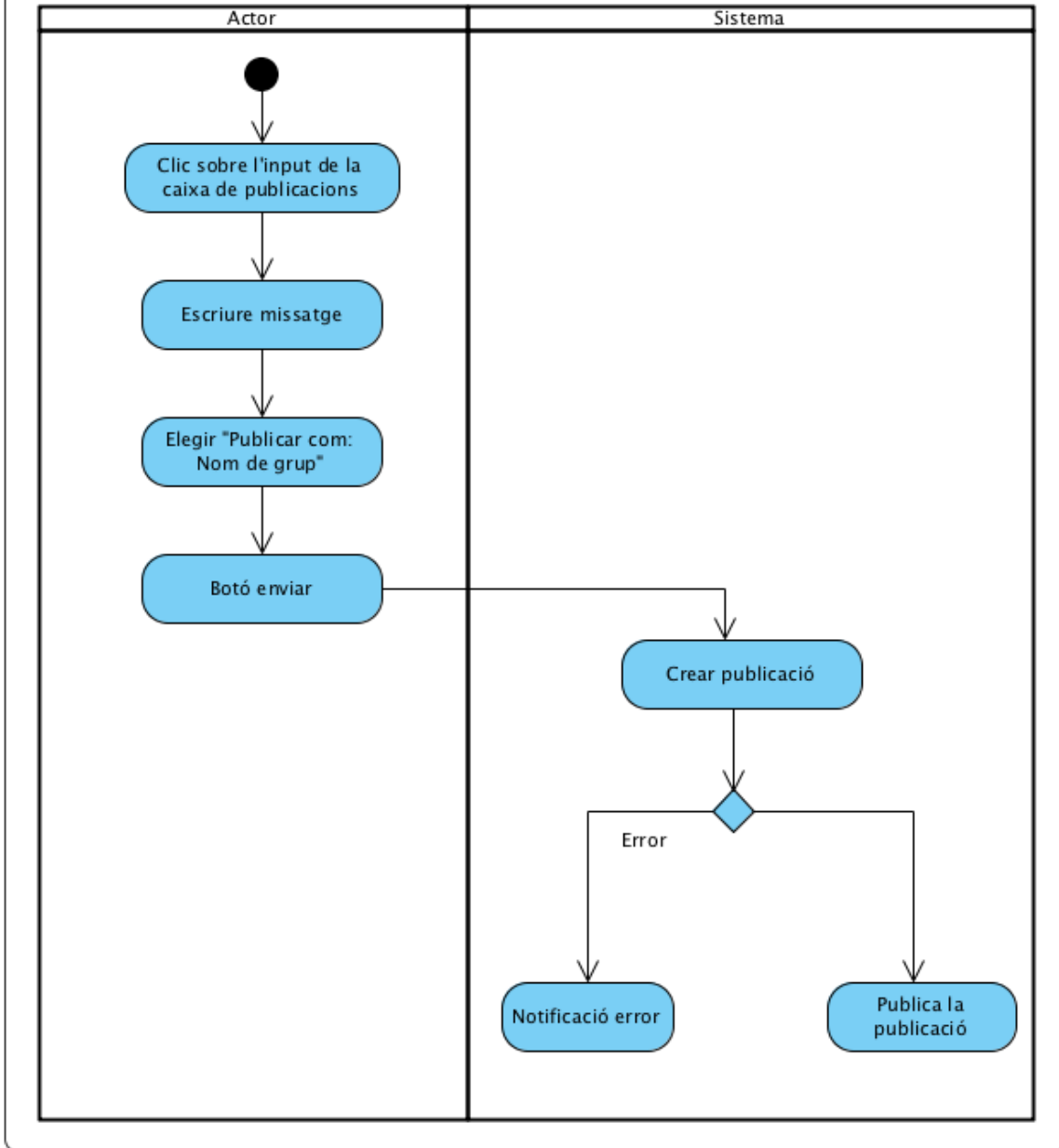


Figura 15, Diagrama d'activitat: Fer publicació

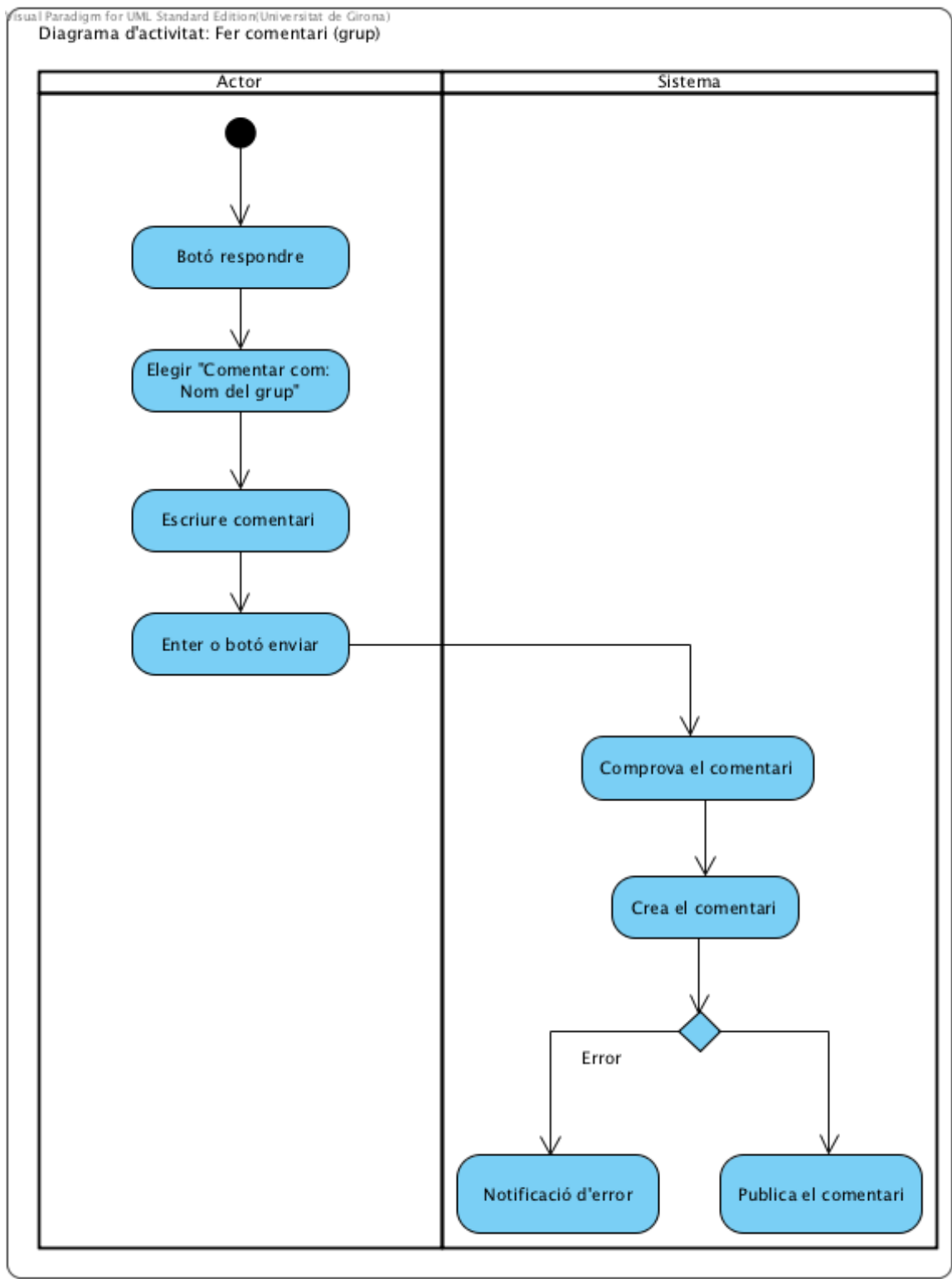


Figura 16, Diagrama d'activitat: Fer comentari

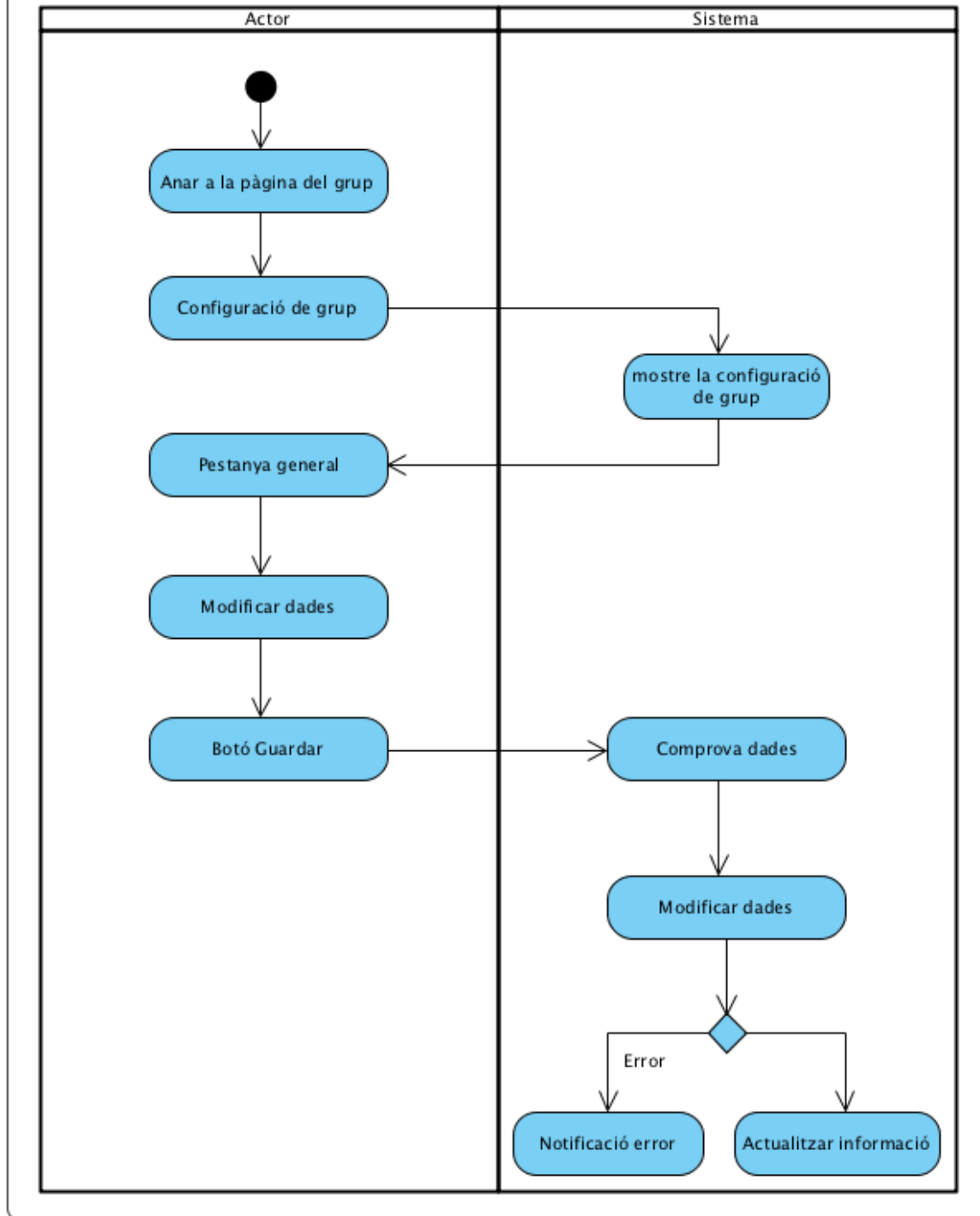


Figura 17, Diagrama d'activitat: Editar informació de grup

8.3 MODEL DE DADES INICAL

El model de dades inicial per desenvolupar l'aplicació és el següent:

Visual Paradigm for UML Standard Edition(Universitat de Girona)

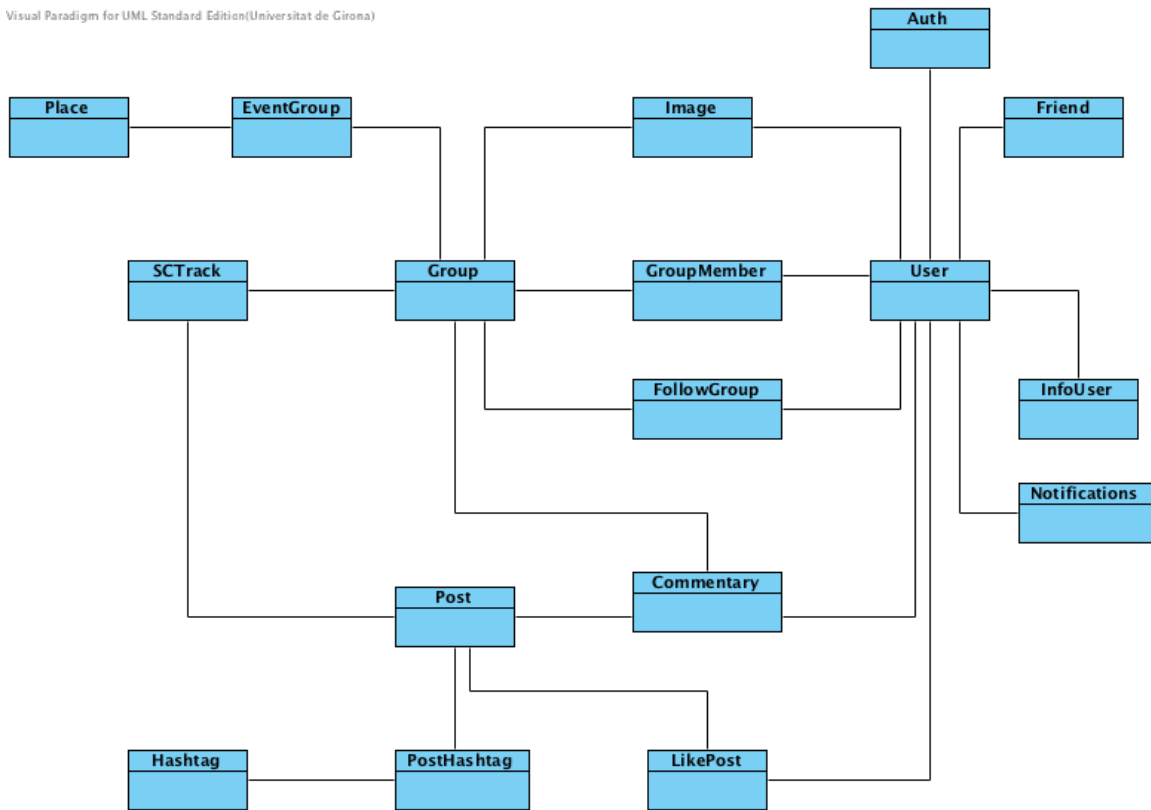


Figura 18, model de dades inicial

8.4 DISSENY D'APLICACIÓ

8.4.1 ARQUITECTURA GLOBAL

El projecte està dividit en dos parts, servidor i aplicació web. El servidor és l'encarregat de proporcionar les dades a l'aplicació web i interacciona amb la base de dades MySQL. Per altre banda l'aplicació web s'encarrega de proporcionar al client la interactivitat amb els altres usuaris. A continuació es representarà de manera general. (Figura 19)

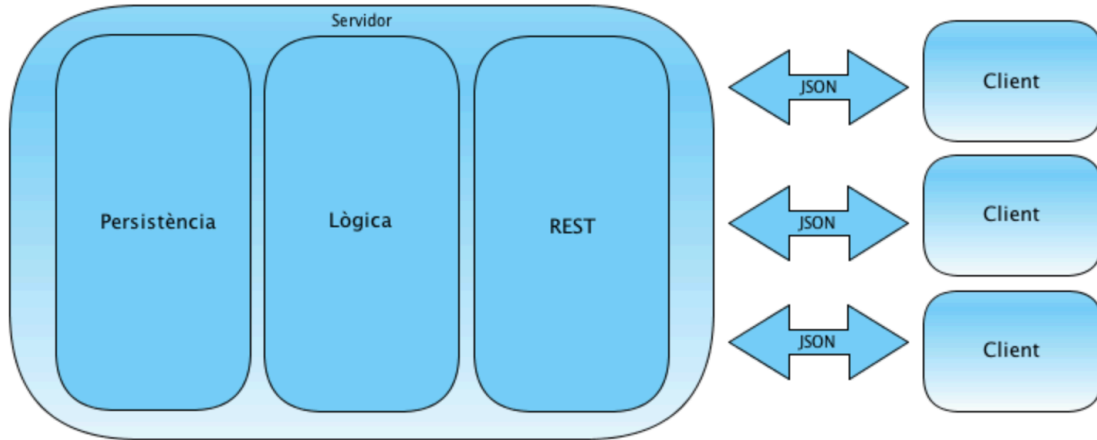


Figura 19, disseny global

8.4.2 SERVIDOR

El disseny del servidor és simple, està dividit en 3 capes, capa de persistència, capa de lògica i la capa de servei web. La capa de persistència té com a objectiu interaccionar amb la base de dades. La capa de lògica, que està en permanent contacte amb la capa de persistència i la capa de servei web, conté tota la lògica de l'aplicació. La capa de servei web és la que proporciona a l'aplicació client tota les dades. A continuació es mostra el diagrama de paquets i l'explicació de cada capa.

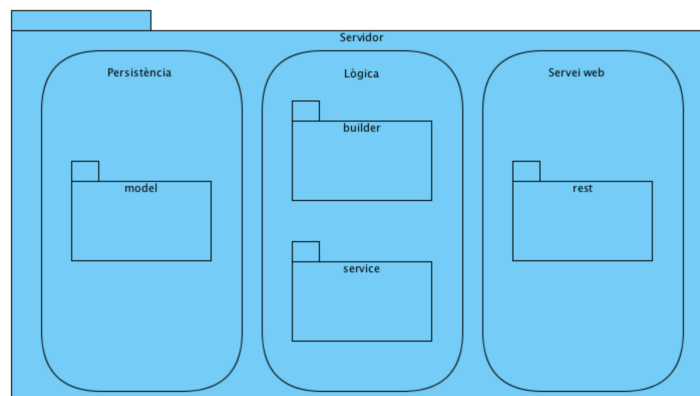


Figura 20, diagrama de paquet

8.4.2.1 Capa de persistència

La capa de persistència s'ha fet amb el framework Hibernate, que encapsula la interacció amb la base de dades. Per poder dur a terme això s'ha "replicat" en el paquet *model* la base de dades en classes Java. Hibernate et permet treballar amb classes Java com objectes de la base de dades.

A continuació es mostra les classes Java que s'han utilitzat com a models de les taules de la base de dades.

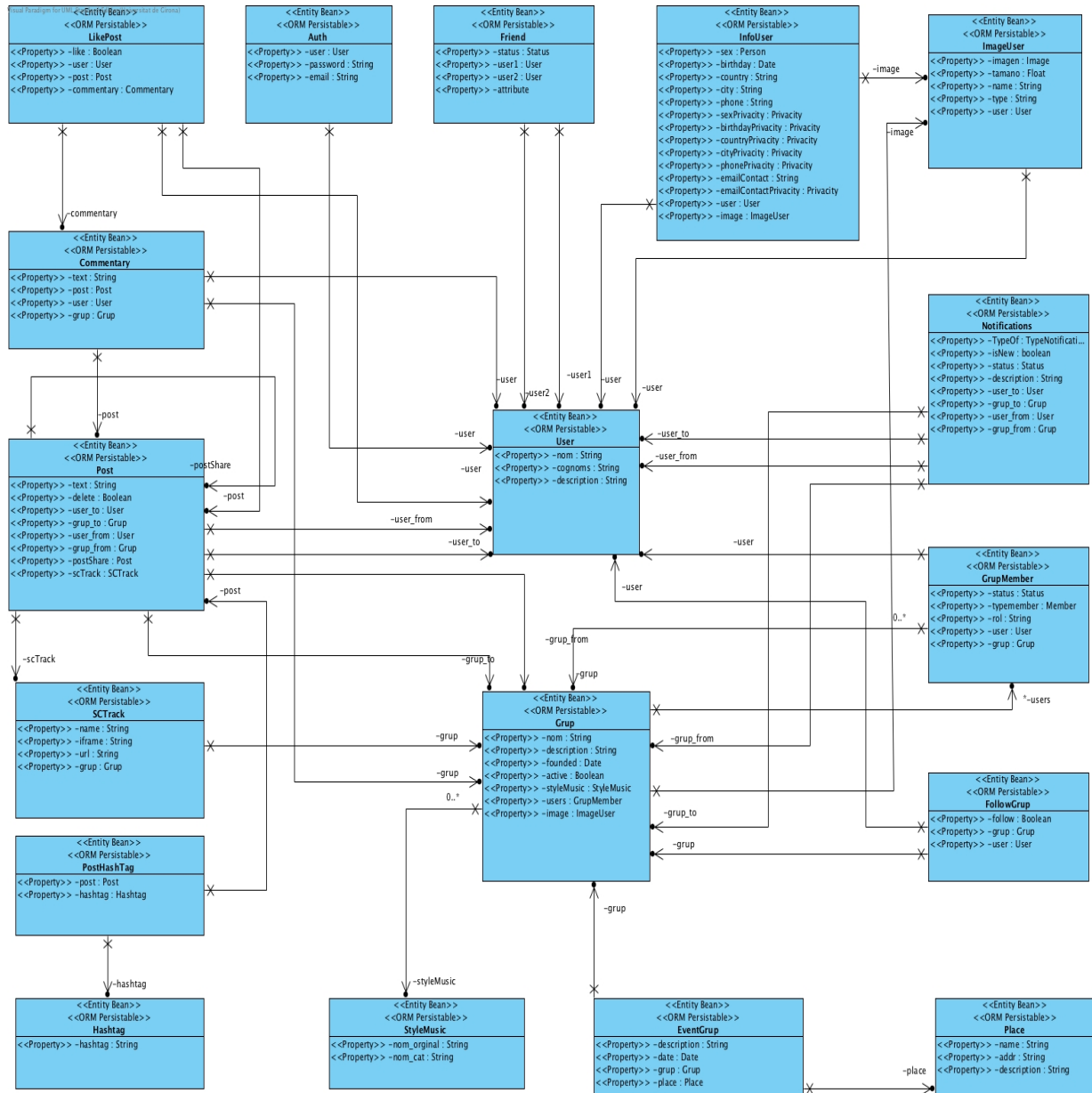


Figura 21, models java de la base de dades

Definició de les taules

User: Taula que conté la informació bàsica d'un usuari; nom, cognom i descripció.

Auth: És la taula que conté la informació del registre; direcció electrònica i contrasenya.

InfoUser: Conté tota la informació de l'usuari i la seva privacitat a l'hora de mostrar-la al públic.

Group: Taula que conté la informació bàsica per un grup; nom, descripció, data de fundació i la imatge.

Post: Taula per les publicacions, té els atributs *text* i *delete*. L'atribut *text* conté el missatge de la publicació, i l'atribut *delete* per saber si l'usuari a eliminat el comentari.

Commentary: Conté tots els comentaris dels usuaris.

EventGrup: És la taula que conté les esdeveniments del grup. Conté la descripció de l'esdeveniment, la data i el lloc on es farà.

Place: S'utilitza per guardar el lloc on és fan els esdeveniments.

Image: S'utilitza per emmagatzemar les imatges de perfil dels usuaris i grups.

Notifications: S'emmagatzema totes les notificacions que rep un usuari. L'atribut *isNew* s'utilitza per saber si l'usuari a vist la notificació.

Hashtag: S'utilitza per emmagatzemar els hashtags que han sortit en algun moments en les publicacions.

SCTrack: És guarden les cançons dels grup.

Friend: Taula que surt de la relació ManyToMany entre dos usuaris. S'utilitza per controlar si un usuari segueix a un altre usuari.

LikePost: Quan un usuari li agrada un comentari o una publicació es guarda en aquesta taula. També serveix, gràcies al seu atribut *UpdateAt*, per saber quines cançons són les més populars del dia.

FollowGroup: El funcionament és semblant que *LikePost*, però en aquest cas és quan un usuari segueix a un grup de música.

GroupMember: Guarda la relació que hi ha un usuari amb un grup, si és membre o no, i de quin tipus és, fundador, administrador o bàsic.

PostHashtag: Encarregat de guardar la relació entre els hashtags i les publicacions la seva utilitat bàsica és per llista les publicacions que hi han d'un determinat hashtag i quins hashtags són els més populars del dia, amb més publicacions en les últimes 24 hores.

A continuació es mostrarà la base de dades que crea Hibernate amb les classes Java mostrades anteriorment.

Visual Paradigm for UML Standard Edition(Universitat de Girona)

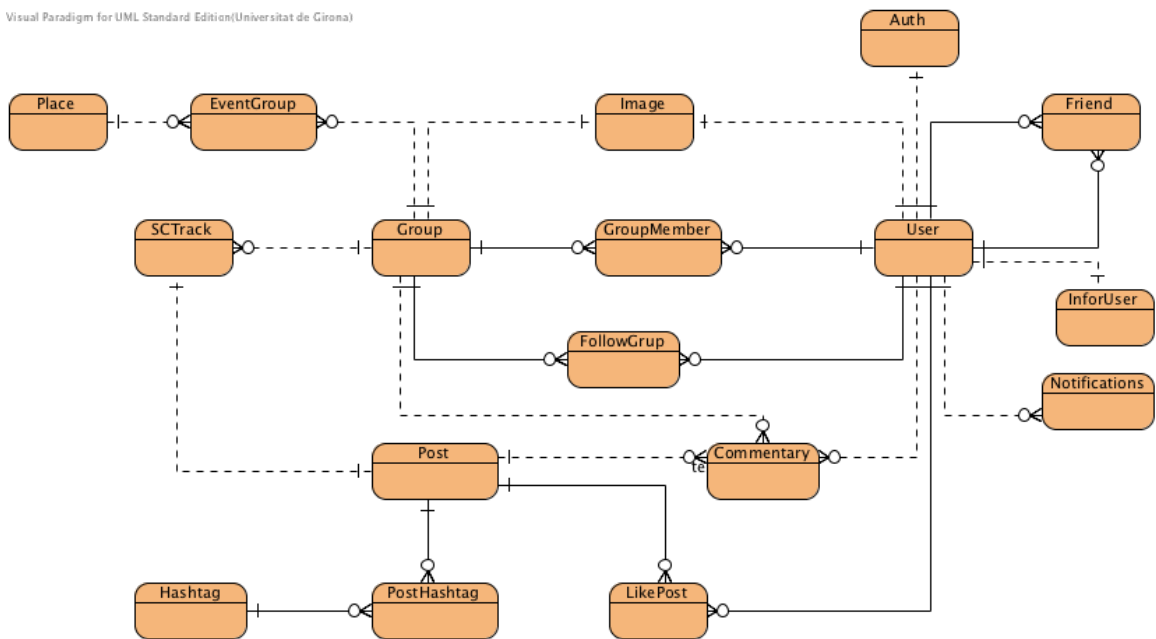


Figura 22, taules de la base de dades

8.4.2.2 Capa de lògica

La lògica està tota en els paquets *Builder*, que conté la majoria de processos com crear objectes que s'envien per l'API REST o processos complicats, i *Service*, que conté totes les crides a la base de dades.

A continuació es mostra les classes del paquet *Service* i del paquet *Builder*.

Diagrama de classes de service

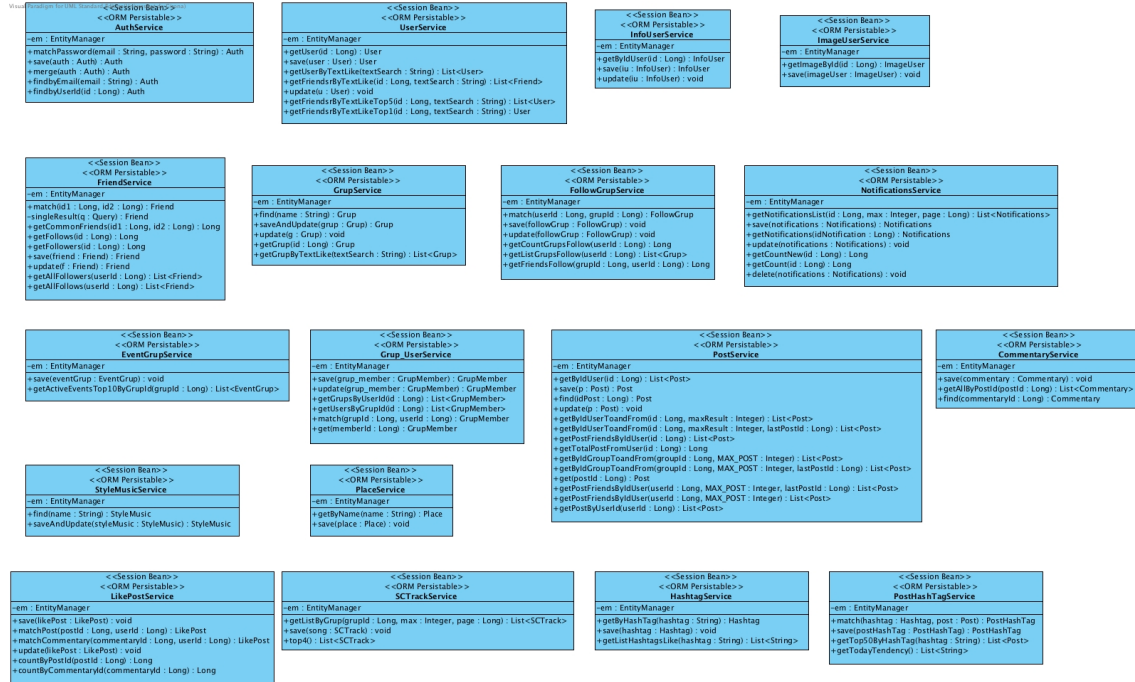


Figura 23, diagrama de classes del paquet service

Diagrama de classes del paquet Builder

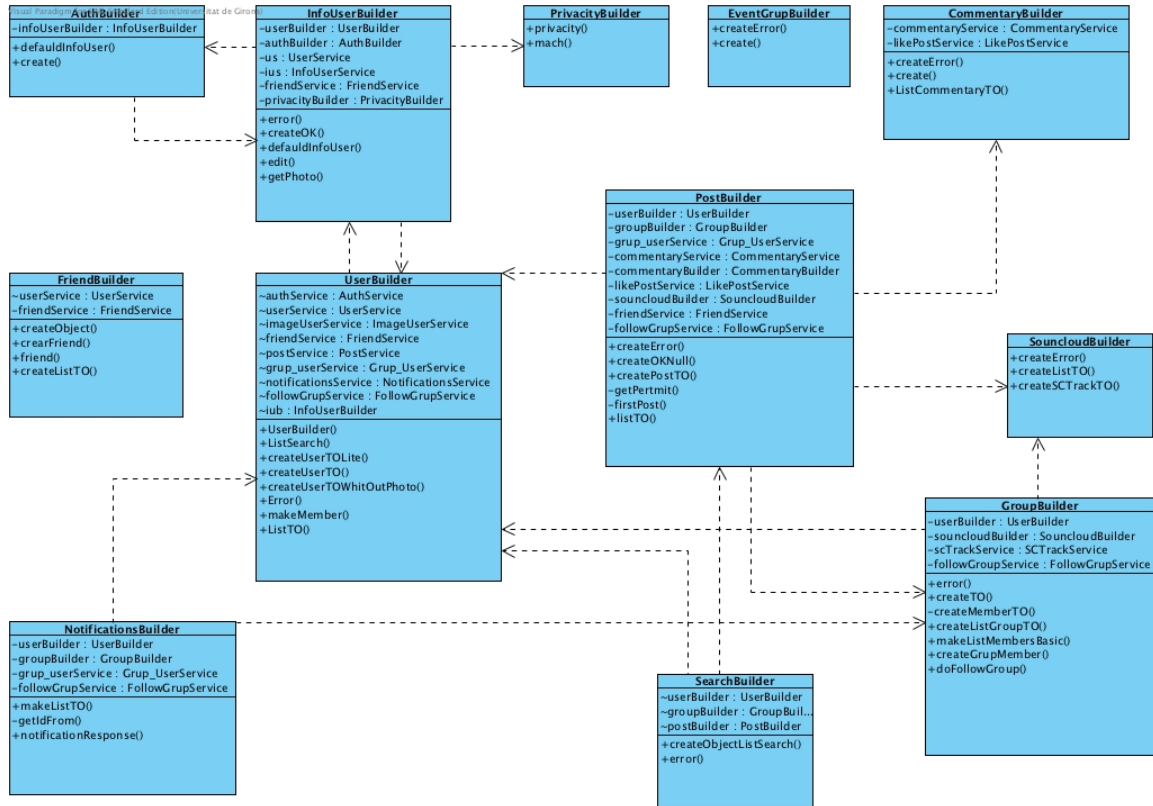


Figura 24, classes del paquet Builder

8.4.2.3 Capa de servei web

La capa de servei web és la més important, l'API REST proporciona tota la informació necessària a les aplicacions clients, en el cas d'aquest projecte a l'aplicació web. A continuació es mostra l'API REST i el diagrama de classes del paquet *rest*.

Els recursos que són privats es necessita l'autenticació de l'usuari en cas contrari es notificarà de l'error.

8.4.2.3.1 Documentació API REST

L'informació completa de l'API REST està al apartat 2 de l'índex.

Autenticació

Consultes sobre l'autenticació de l'usuari actual.

Recurs	Descripció
GET rest/auth/init Públic	Per saber si hi ha un usuari amb la sessió iniciada.
POST rest/auth Públic	Per iniciar sessió amb un usuari.
POST rest/auth/registre Públic	Registre d'un usuari.
POST rest/auth/logout Privat	Tancar sessió.

User

Informació bàsica d'un usuari.

Recurs	Descripció
GET rest/user/my Privat	Retorna la informació bàsica de l'usuari autenticat.
POST rest/user/my/infoLite Privat	Retorna una versió reduïda de la informació de l'usuari.

Informació de l'usuari

Informació completa d'un usuari.

Recurs	Descripció
GET rest/infouser/user/:id Privat	Retorna la informació completa de l'usuari desitjat.
GET rest/infouser/my Privat	Retorna la informació completa de l'usuari autenticat.
PUT rest/infouser/my Privat	Modificació de la informació d'un usuari autenticat.

Notificacions

Informació sobre les notificacions d'un usuari.

Recurs	Descripció
GET rest/notifications/my/:page Privat	Retorna una llista de 8 notificacions de l'usuari autenticat.
PUT rest/notifications/:id Privat	Per modificar una notificació.

Seguidors

Informació sobre amistats dels usuaris.

Recurs	Descripció
GET rest/friend/:id Privat	Retorna la relació que té l'usuari autenticat amb l'usuari (amb id = :id)
POST rest/friend/:id Privat	Per fer que l'usuari autenticat segueixi a l'usuari amb id = :id.
PUT rest/friend/:id Privat	Per actualitzar la relació entre l'usuari autenticat i l'usuari amb id = :id.
GET rest/friend/followers/:id Privat	Retorna una llista d'usuaris que són seguidors de l'usuari desitjat.
GET rest/friend/follows/:id Privat	Retorna una llista d'usuaris que segueixen a l'usuari desitjat.

Grup

Informació sobre els grups de música.

Recurs	Descripció
GET rest/group/:id Privat	Retorna la informació del grup desitjat.
POST rest/group Privat	Per la creació d'un grup.
PUT rest/group/:id Privat	Per la modificació del grup desitjat.
POST rest/group/member/:id Privat	Creació de invitació a un membre per ser membre de l'usuari especificat.
PUT rest/group/member/:id Privat	Modificació de la informació del membre de grup especificat.
DELETE rest/group/member/:id/:memberId Privat	Per eliminar el membre del grup.
GET rest/group/favourites/:id Privat	Retorna una llista de grups favorits de l'usuari desitjat.
PUT rest/group/follow/:id Privat	Per que l'usuari autenticat segueixi (o deixi de seguir) al grup especificat.

Esdeveniments

Informació sobre els esdeveniments d'un grup.

Recurs	Descripció
POST rest/event Privat	Creació d'un esdeveniment .

SoundCloud

Cançons SoundCloud.

Recurs	Descripció
POST rest/soundcloud Privat	Per posar una cançó.
GET rest/soundcloud/top4 Privat	Retorna una llista de les 4 cançons més populars de les últimes 24 hores.

Publicació

Informació sobre les publicacions de usuaris i grups.

Recurs	Descripció
POST rest/post Privat	Creació d'una publicació.
DELETE rest/post/my/:id Privat	Borra la publicació de l'usuari autenticat.
GET rest/post/my Privat	Retorna una llista de totes les publicacions de l'usuari autenticat.
GET rest/post/user/:id/:lastId Privat	Retorna una llista de 16 publicacions de l'usuari desitjat.
GET rest/post/:id Privat	Retorna la publicació desitjada.
GET rest/post/hashtag/:hashtag Privat	Retorna el llistat dels últimes 50 publicacions que tenen el hashtag desitjat.
GET rest/post/group/:id/:lastId Privat	Retorna una llista de 16 publicacions del grup desitjat.
GET rest/post/friends/:id/:lastId Privat	Retorna una llista de publicacions dels usuaris i grups que segueix l'usuari desitjat.
POST rest/post/share Privat	Per compartir la publicació desitjada.

Comentari

Informació sobre els comentaris.

Recurs	Descripció
POST rest/commentary Privat	Creació d'una comentari.

Like

Informació sobre els "m'agrada" dels comentaris i publicacions.

Recurs	Descripció
POST rest/like Privat	Per posar "m'agrada" de l'usuari autenticat a la publicació o comentari especificat.
PUT rest/like Privat	Modificació del vincle "m'agrada" de l'usuari autenticat a la publicació o comentari especificat.

HashTag

Informació sobre els hashtags.

Recurs	Descripció
GET rest/hashtag/tendency Privat	Retorna una llista amb el nom dels 10 hashtags més populars de les últimes 24h.

Cerca

Informació sobre cerques de usuaris, grups, hashtags.

Recurs	Descripció
GET rest/search/:text Privat	Retorna una llista de usuaris, grups, hashtags que tenen relació amb el text especificat.
GET rest/search/friends/:text Privat	Retorna una llista de usuaris seguits, per l'usuari autenticat, que el nom de aquests contingui el text especificat.

Diagrama de classes del paquet REST

Diagrama que conté totes les classes de la REST.



Figura 25, diagrama de classes del paquet rest

8.4.2.3.2 Diagrames de seqüència del servidor

En aquest apartat es tractarà el procés que segueixen les peticions, les més importants, a l'API REST d'un usuari al sistema, que presenten una complexitat elevada pel que fa a les tasques. Així doncs, es farà el seguiment de les classes i mètodes que es fan servir a través de diagrames de seqüència per fer més entenedor aquest procés.

Registre d'un usuari

Per registrar un nou usuari a la base de dades, el primer que es fa, és comprovar que no hi hagi cap usuari en el nostre sistema amb la mateixa adreça electrònica, una vegada es comprova això es crea un objecte *Auth* i *User* amb d'informació que ha donat l'usuari en el formulari, seguidament es crea un objecte *InfoUser* per defecte i una notificació del sistema cap a l'usuari. Si a sigut correcte s'enviarà una objecte amb missatge de "OK", en cas contrari se li notificarà quin ha sigut l'error.

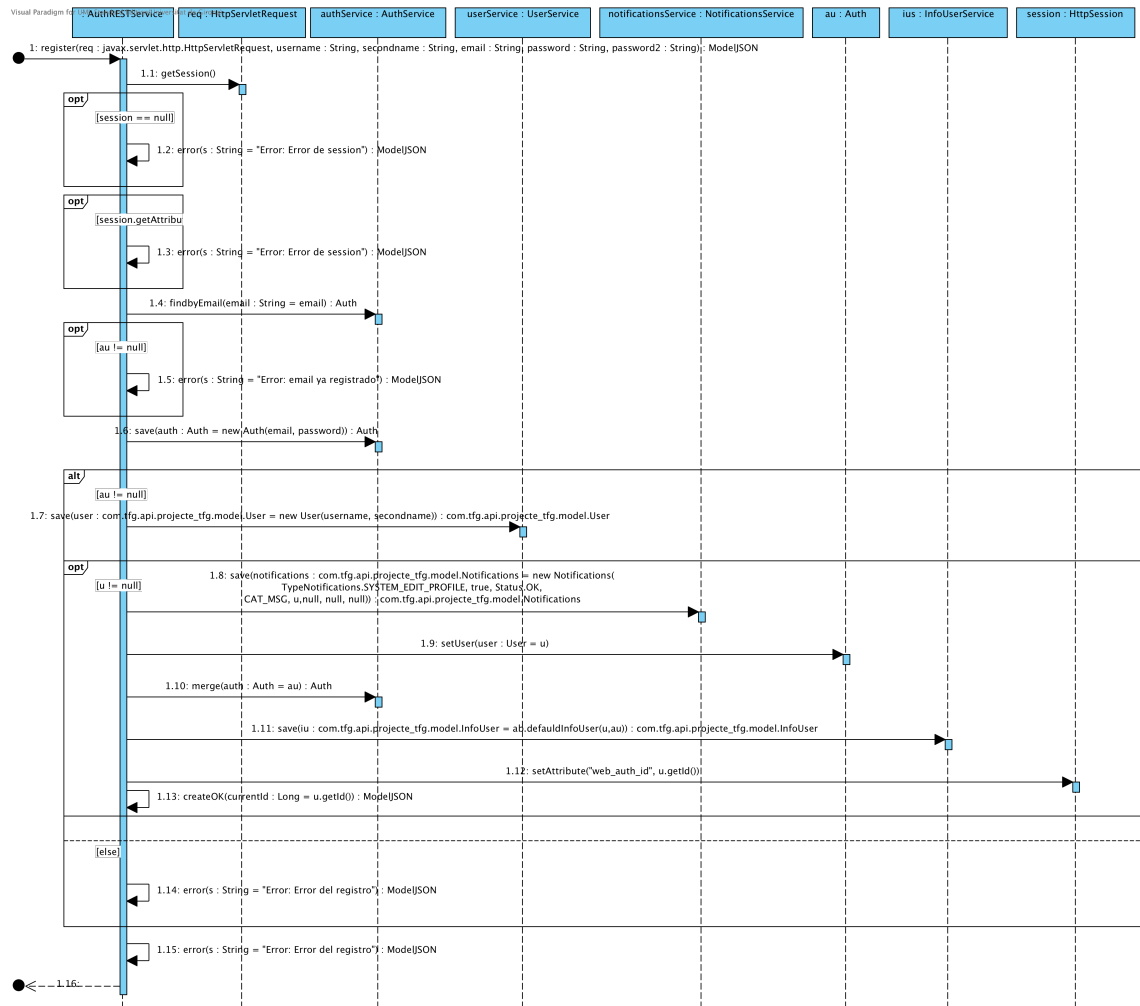


Figura 26, diagrama de seqüència registrar usuari

Autenticació d'un usuari

Per autenticar un usuari es comprova que no hi hagi cap sessió iniciada, una vegada fet això, es mira que coincideixi l'adreça electrònica i la contrasenya, si tot va bé creem una sessió amb l'identificador de l'usuari i enviem un missatge de "OK", en cas contrari enviem un missatge de error.

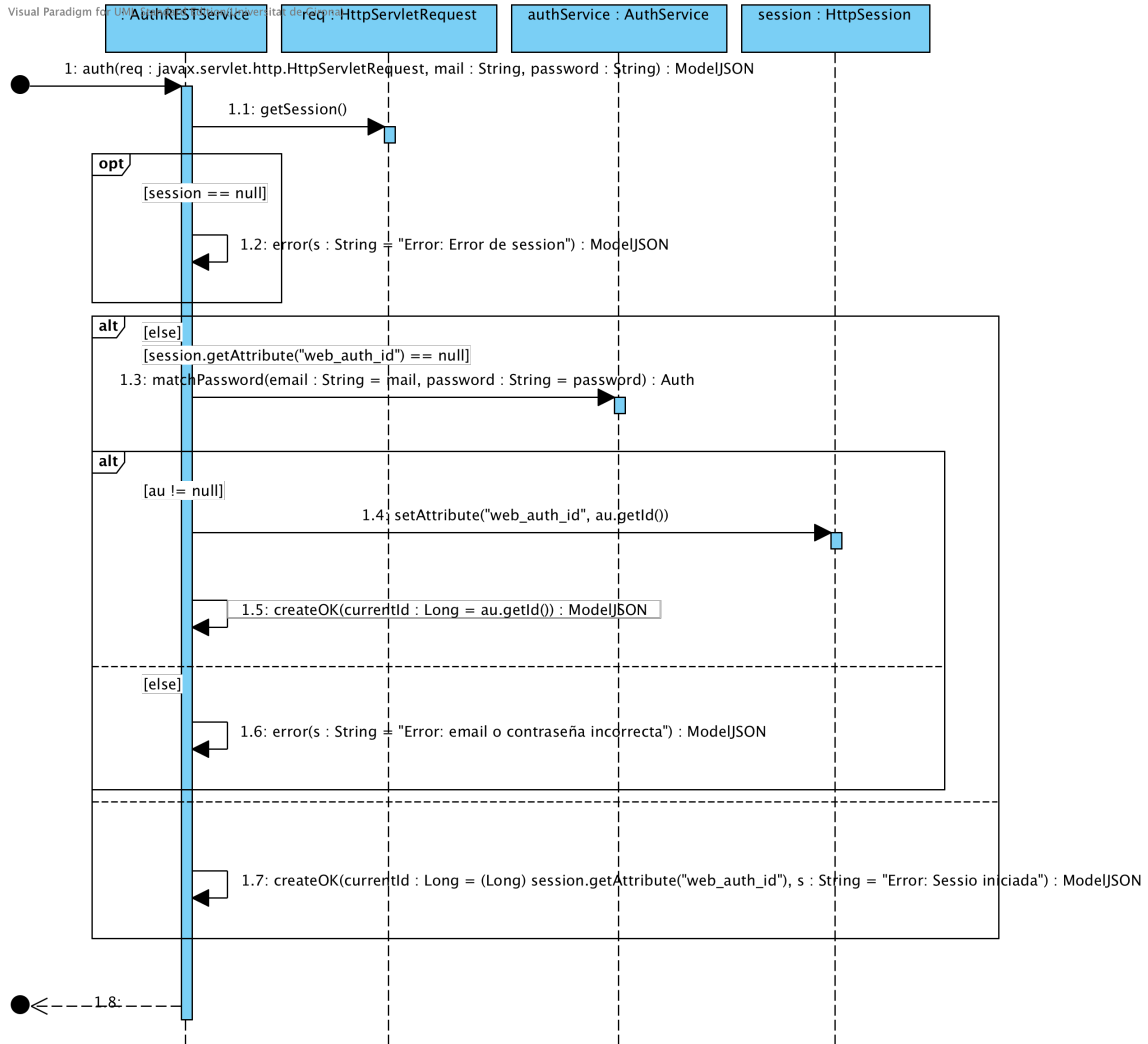


Figura 27, diagrama de seqüència autenticar a un usuari

Creació d'una publicació

Per fer una publicació es comprova que hi ha una sessió iniciada, seguidament es guarda a la base de dades la publicació, després es comprova el hashtags que hi han a la publicació, i finalment es crea una notificació d'avis al destinatari (si hi ha) de la publicació, si no hi hagut cap problema s'envia la publicació, en cas contrari, s'envia una notificació de error.

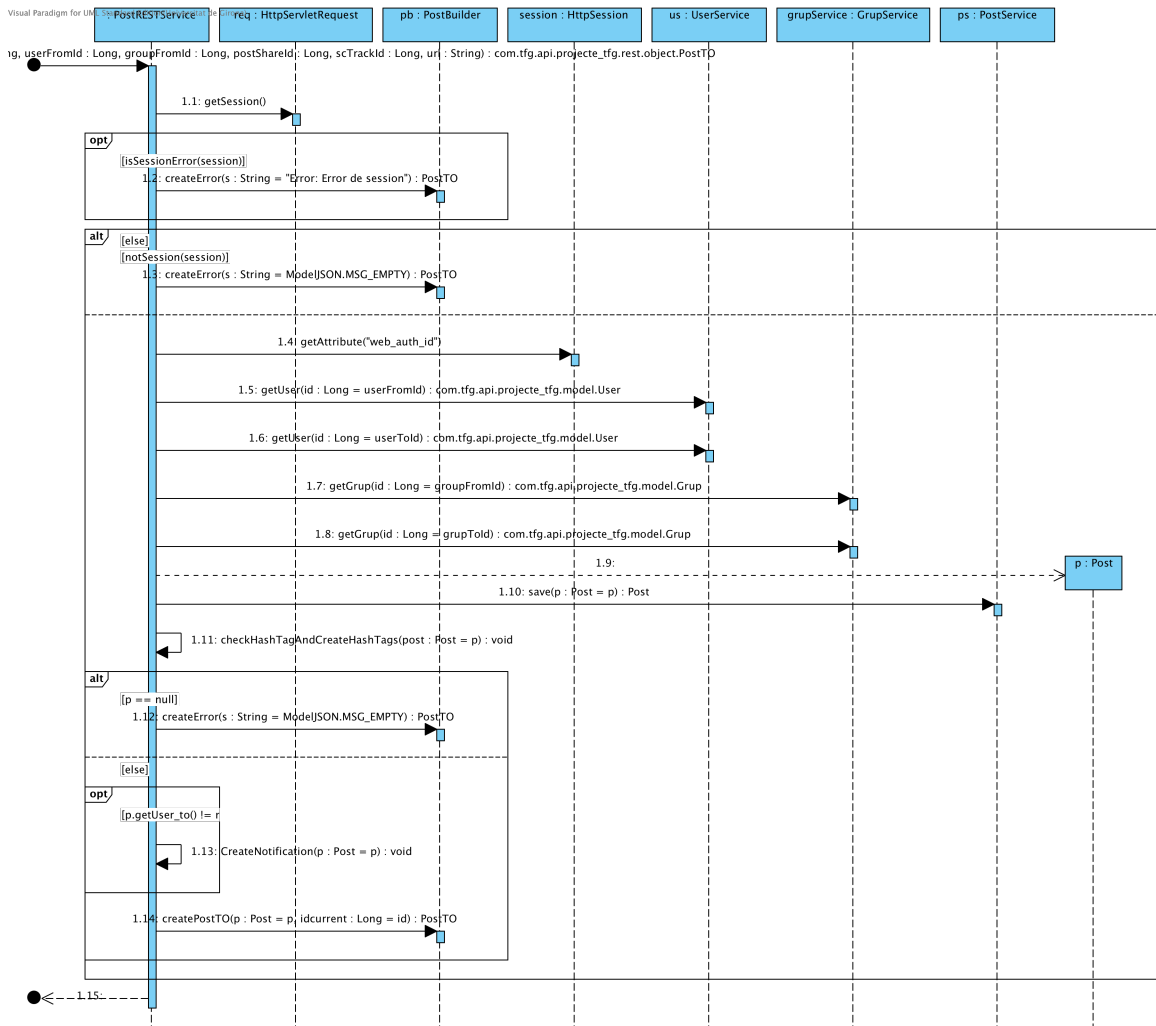


Figura 28 diagrama de seqüència creació de publicació

Creació d'un comentari

Per crear un comentari el sistema comprova que hi ha una sessió iniciada, el post, i l'usuari o grup que fa la publicació. A continuació es crea el comentari i el guarda, si tot a anat bé, s'envia el comentari, altrament se li envia una missatge de error.

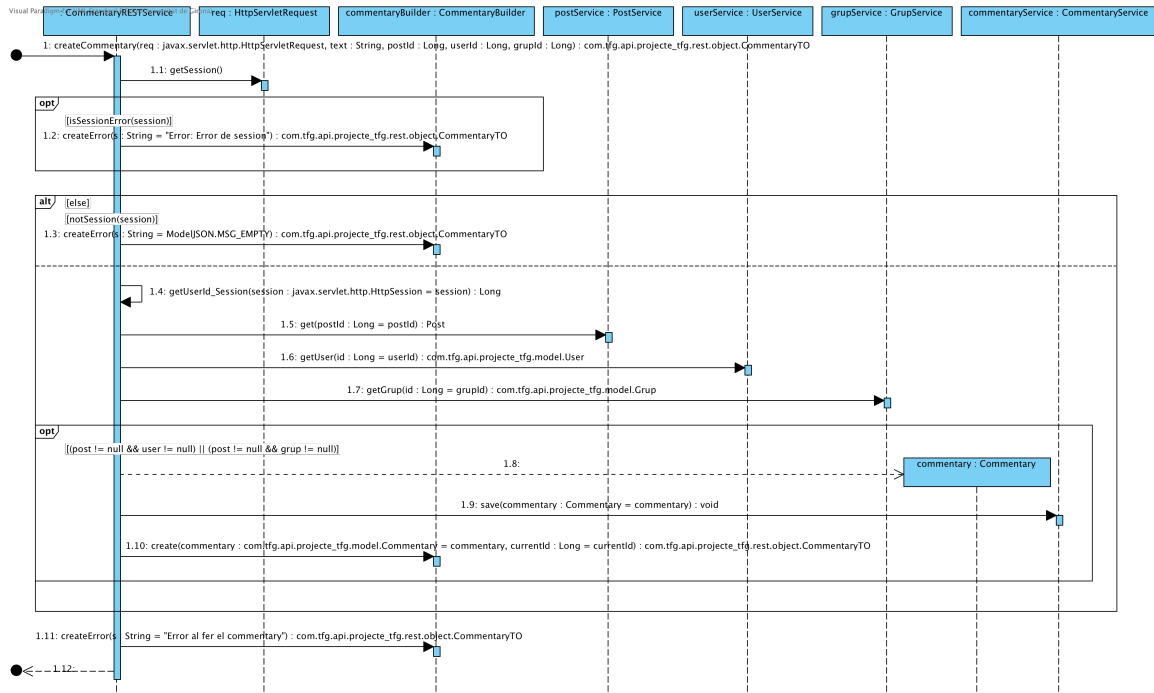


Figura 29, diagrama de seqüència creació d'un comentari

Publicacions d'un usuari

Per obtenir totes les publicacions fetes i dirigides per un usuari en concret, s'ha d'estar autenticat, després es cerca l'usuari, seguidament si l'usuari no existeix i/o si l'usuari autenticat no el segueix es retorna un missatge de error, en cas contrari, s'envia una llista amb les 15 ultimes publicacions.

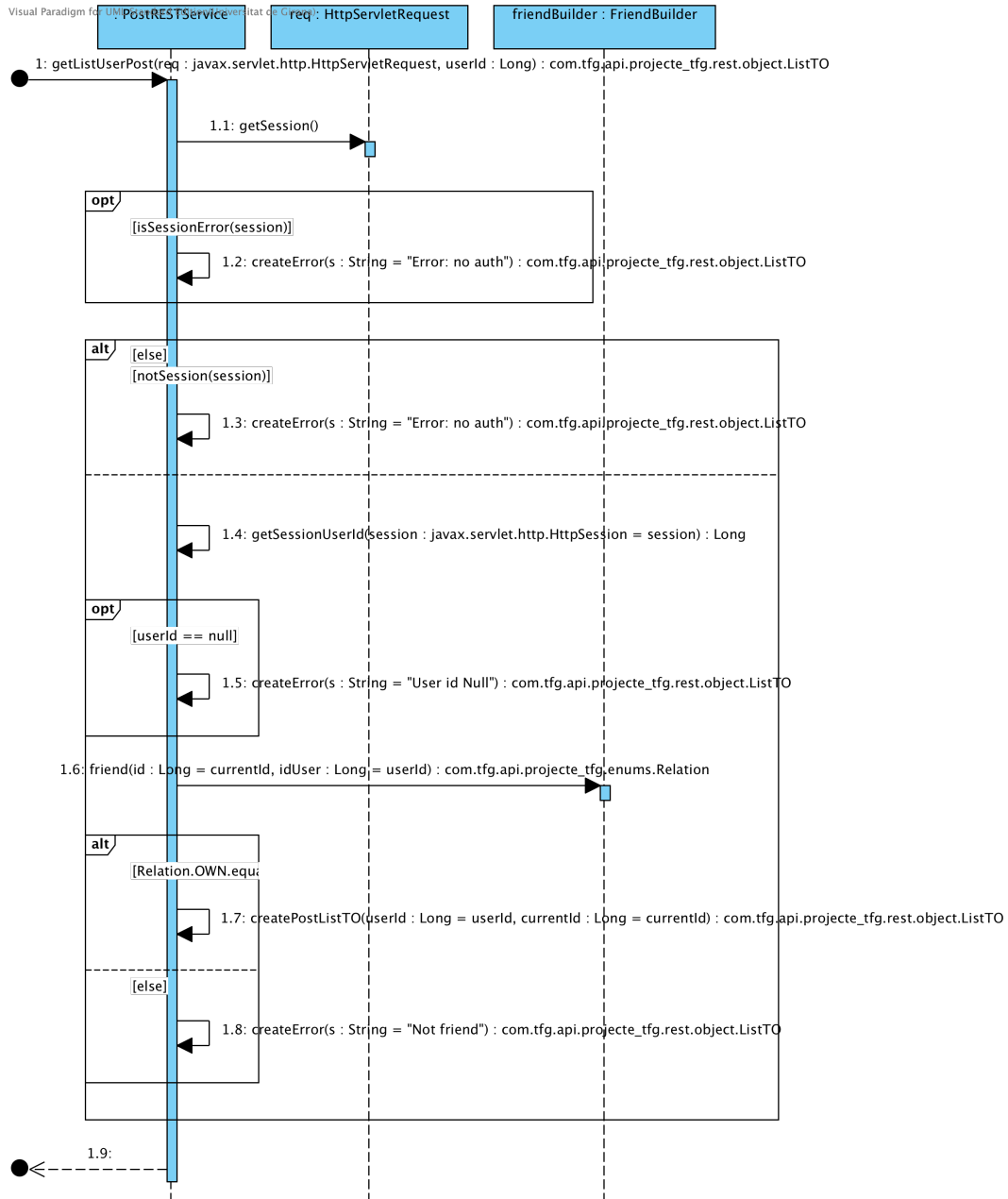


Figura 30, diagrama de seqüència publicacions d'un usuari

Creació d'un grup

Per crear un grup s'ha de tenir la sessió iniciada, i que el nom del grup que es proposa no existeixi. Una vegada comprovada la sessió i la no existència d'un grup amb el mateix nom, es crea el grup amb la informació donada per l'usuari. Seguidament una vegada creat el grup, es crea una notificació a l'usuari creador de creació de grup correcte, a continuació es crea una invitació als membre per saber si vol formar part del grup.

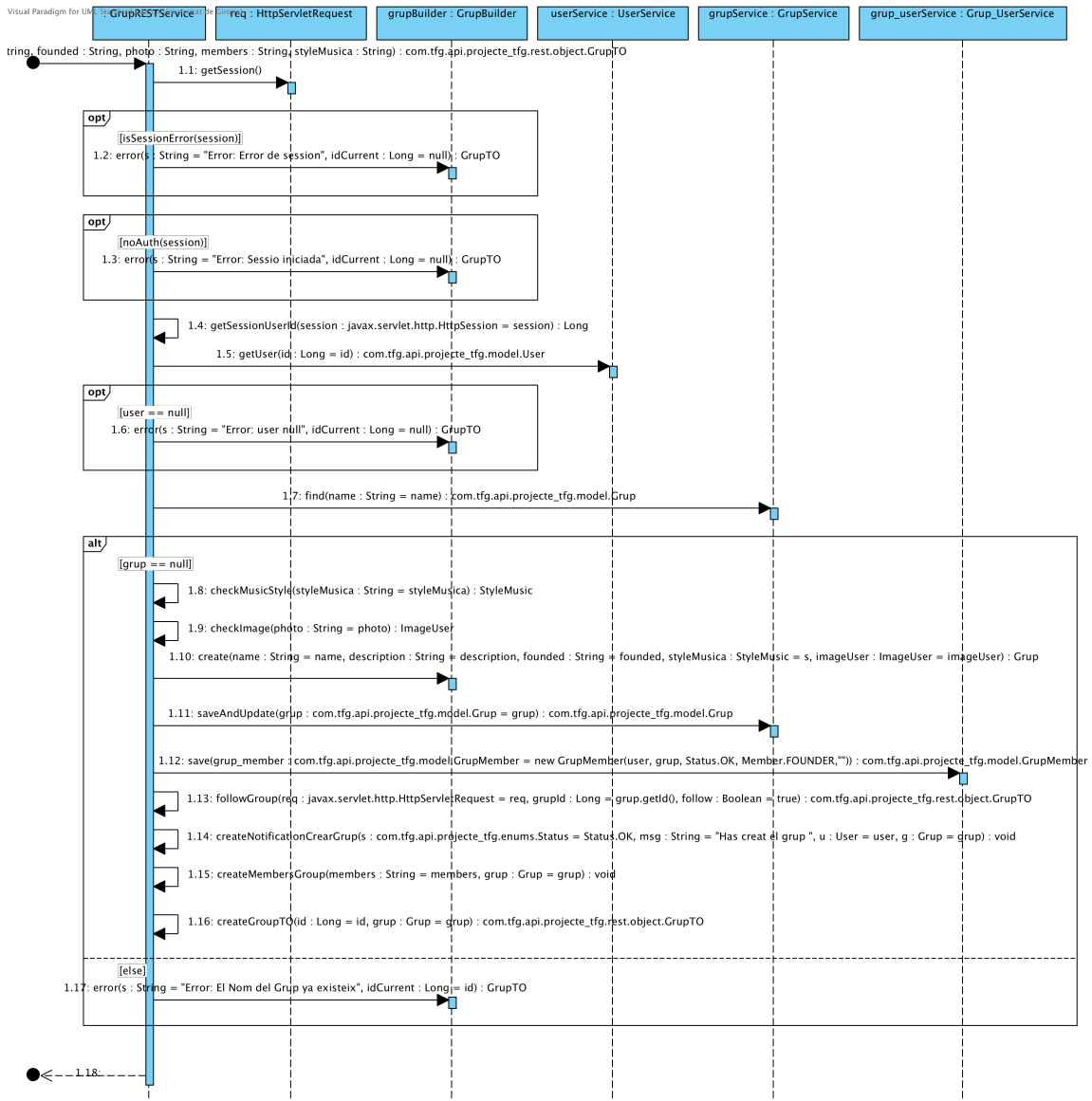
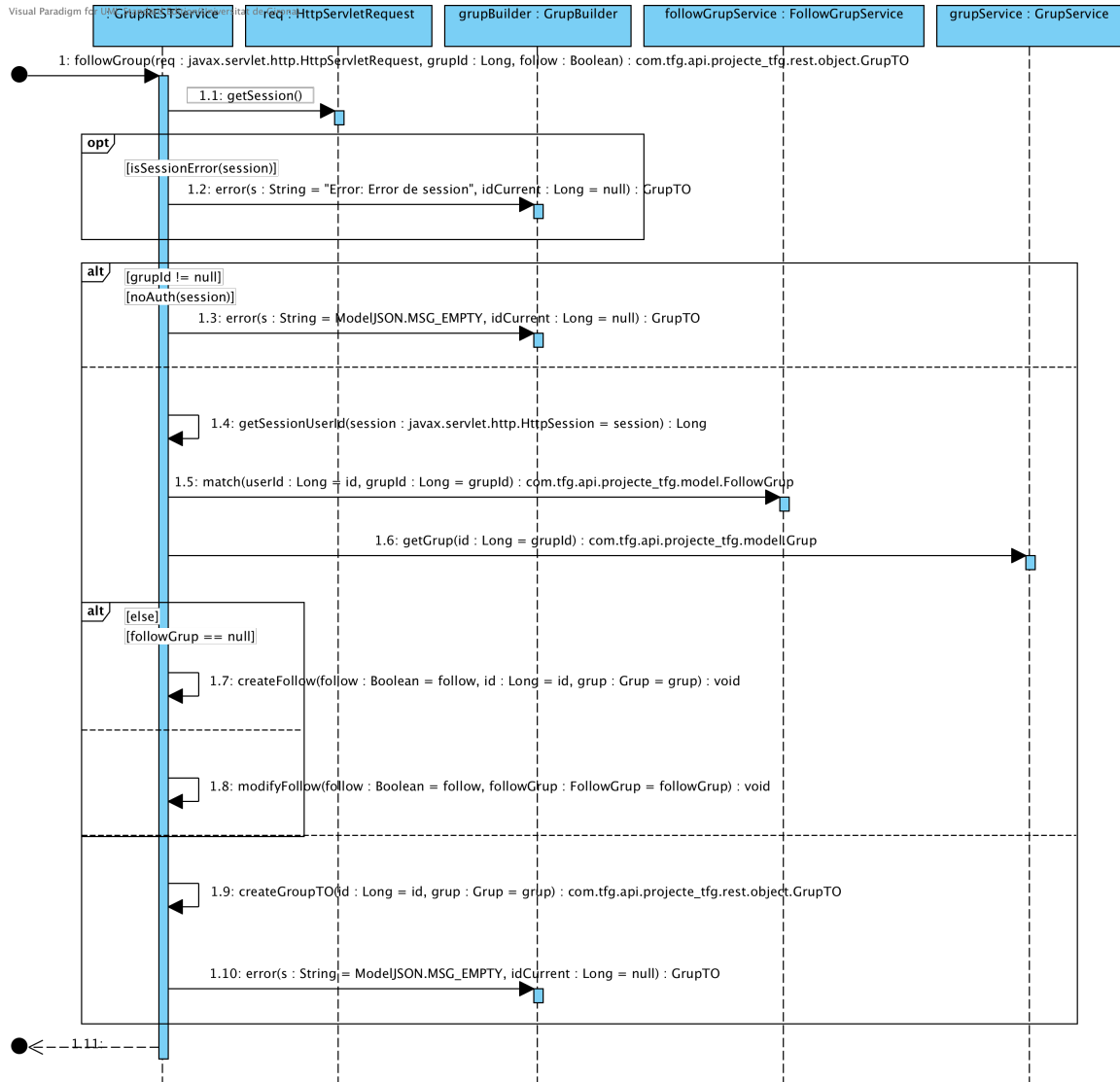


Figura 31, diagrama de seqüència de creació de grup

Seguir a un grup

Per seguir a un grup, l'usuari ha de estar autenticat, es comprova que el grup existeixi i es crea (o es modifica) el vincle entre l'usuari i el grup, es retorna la informació del grup si el procediment a sigut satisfactori, en cas contrari, s'envia un missatge de error.



Modificar informació usuari

Només es pot editar la informació d'un mateix, per tant s'ha de estar autenticat, seguidament es modifiquen els atributs. Una vegada fet això s'envia la informació si no hi hagut cap problema, altrament, s'envia l'error.

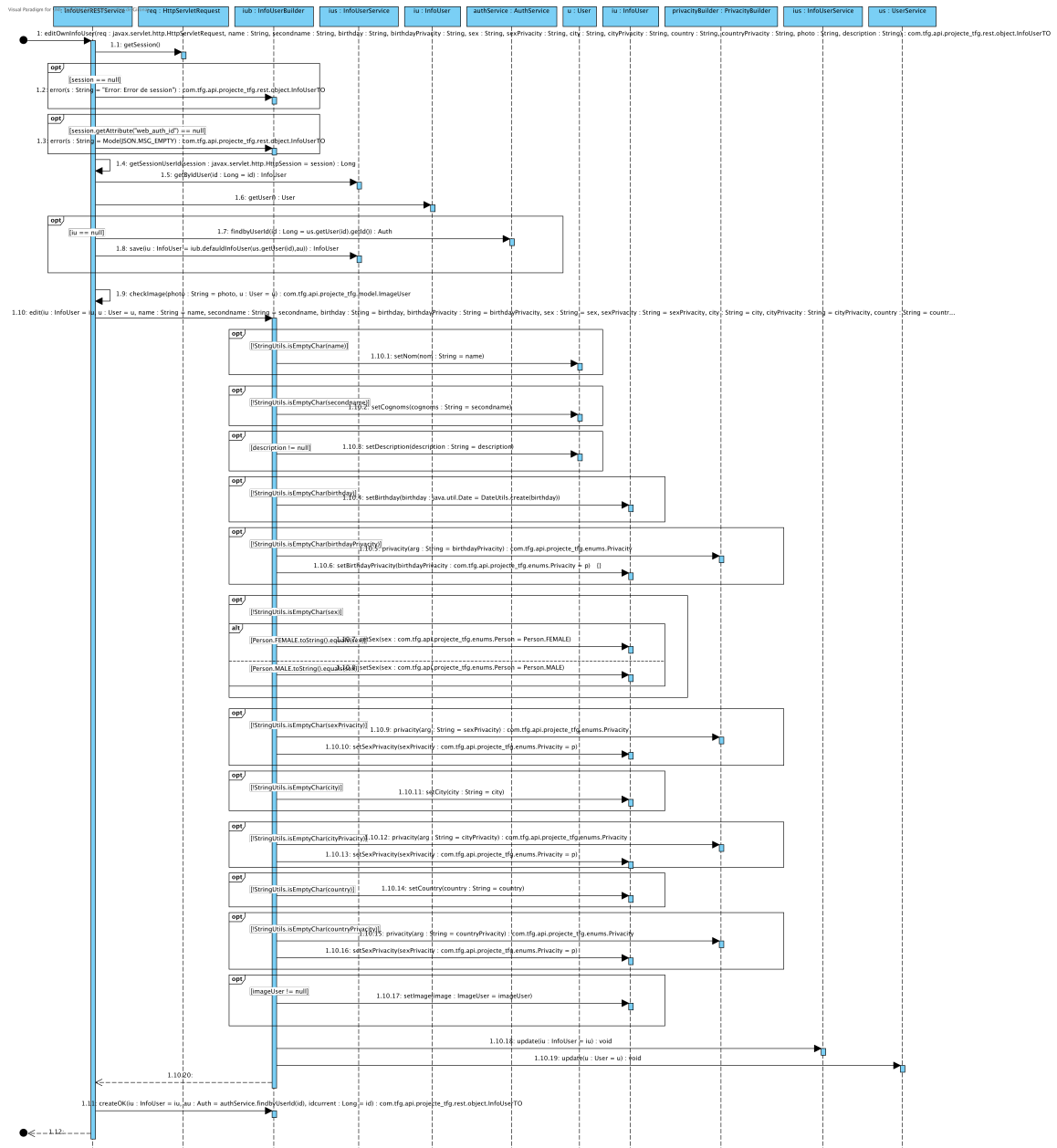


Figura 33, diagrama de seqüència: modificació de la informació d'un usuari

Afegir membre de Grup

Per invitar a usuari al grup, tenim que crear un invitació i esperar la resposta de l'usuari convidat. Per fer una invitació s'ha d'estar autenticat, després de comprovar que té la sessió iniciada comprova si l'usuari i el grup existeix, si és correcta la validació es crea la invitació.

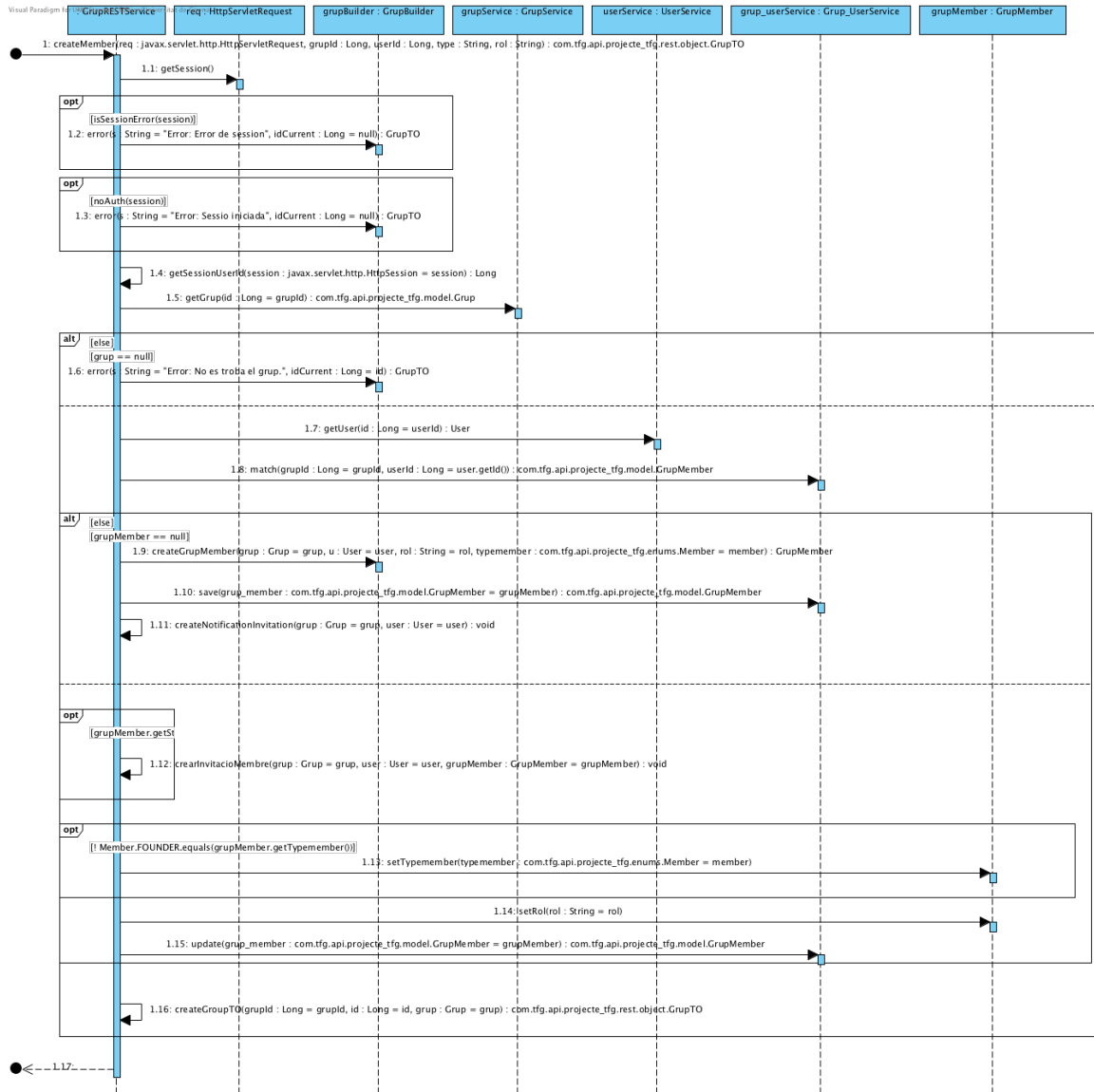


Figura 34 diagrama de seqüència afegir membre

Acceptar invitació

Per acceptar una notificació s'ha d'estar autenticat, seguidament es canvia l'estat de la notificació, si la resposta ha sigut positiva es posa l'estat de la notificació en "OK", en cas contrari "NO", una vegada fet això, s'accepta (o no) la invitació.

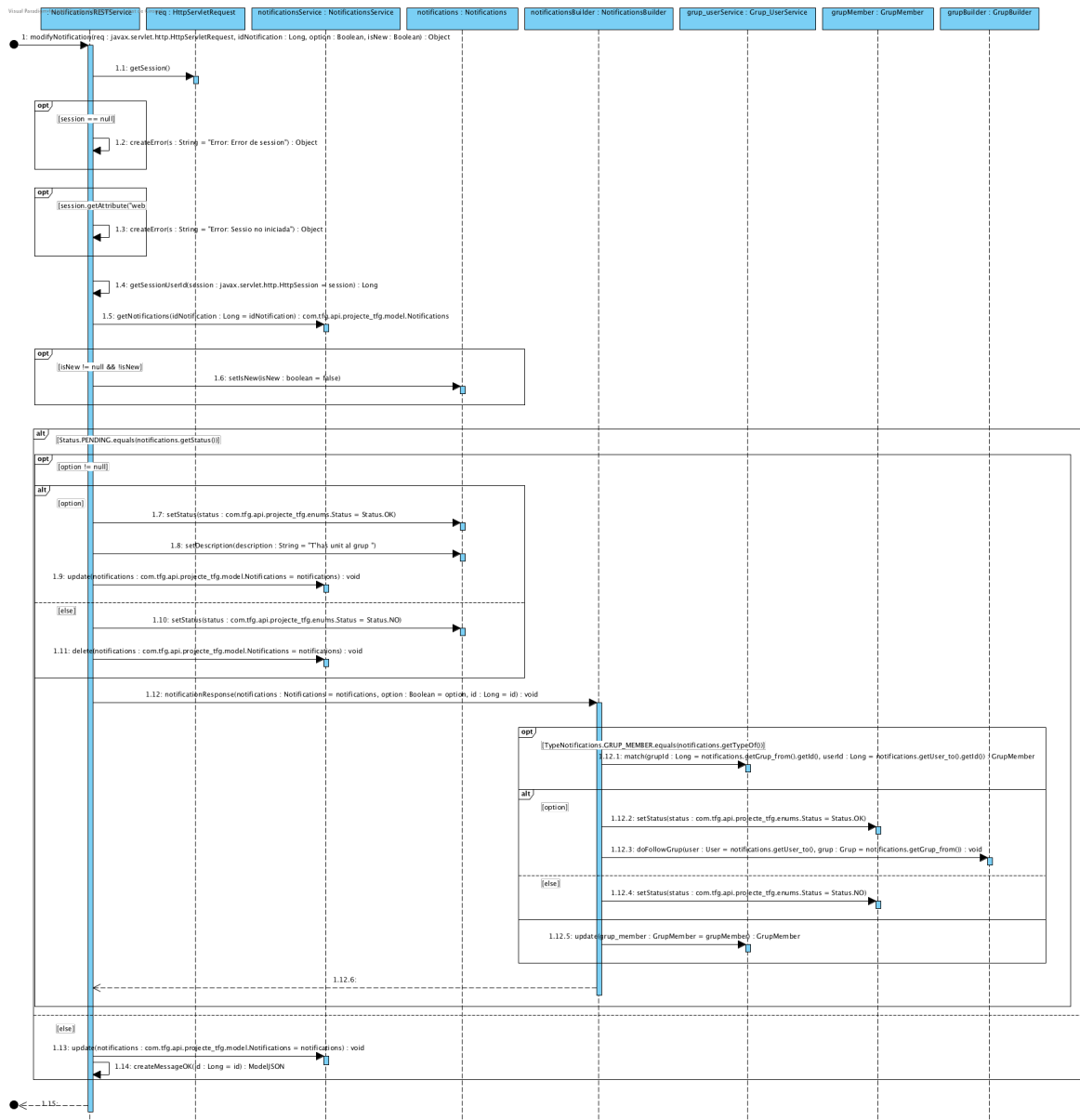


Figura 35, diagrama de seqüència acceptar una invitació

Posar cançó SoundCloud

Per posar una cançó del grup es comprova que l'usuari estigui autenticat, i que sigui fundador del grup o membre administrador, en cas positiu es crea la cançó, també es crea una publicació amb només la cançó. Finalment, s'envia una llista amb les últimes 5 cançons si tot a sigut correcta, d'altre banda, envia un missatge de error.

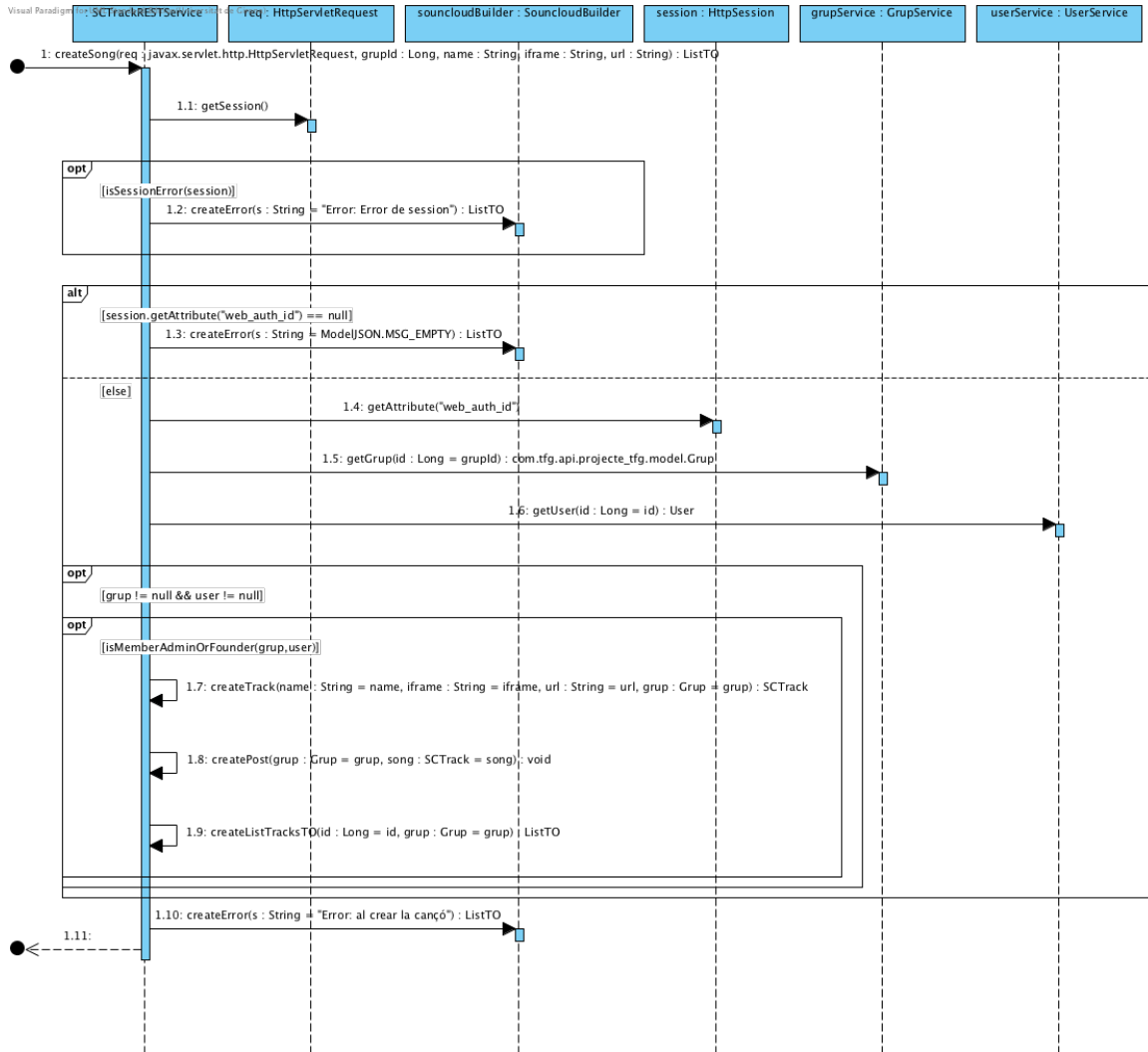


Figura 36, diagrama de seqüència crear una cançó SoundCloud

Crear esdeveniment

Per crear un esdeveniment l'usuari ha de estar autenticat, seguidament es comprova que l'usuari és fundador o membre administrador del grup, en cas afirmatiu es crea l'esdeveniment, en cas negatiu es notifica de l'error.

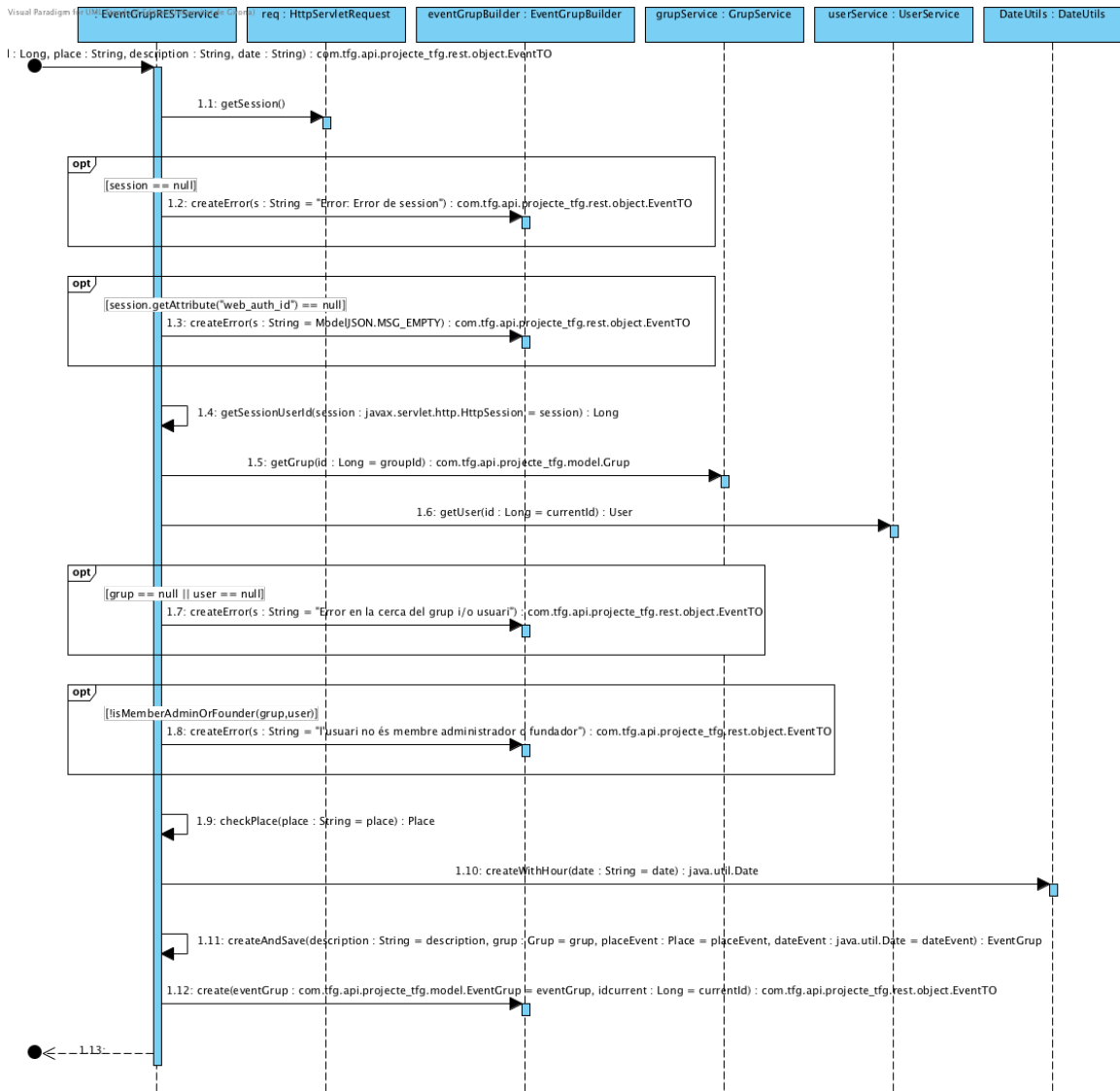


Figura 37 diagrama de seqüència crear un esdeveniment

8.4.3 APLICACIÓ CLIENT

L'aplicació client utilitza com a eina principal el Backbone, que ens ajuda a construir l'aplicació web amb el patró de MVC.

Backbone ens dona les eines necessàries per poder aplicar un patró de vista-controlador, en el qual intervenen les Backbone.Views (Vistes) i els Backbone.Model (models), o col·leccions de models. En les vistes podem posar tota la lògica per tractar esdeveniments i accions del l'usuari per tal de mostrar dinàmicament el contingut, per altre banda els models contenen, si es vol, la lògica de negoci implementada.

Les vistes poden tenir associat un model, en el qual també es pot tractar els canvis, un exemple és associar a diferents vistes un mateix model, i tractar el canvi d'un atribut per actualitzar aquest atribut en totes les vistes, com podria ser un canvi de "nom" en un model que conte tota la informació d'un usuari.

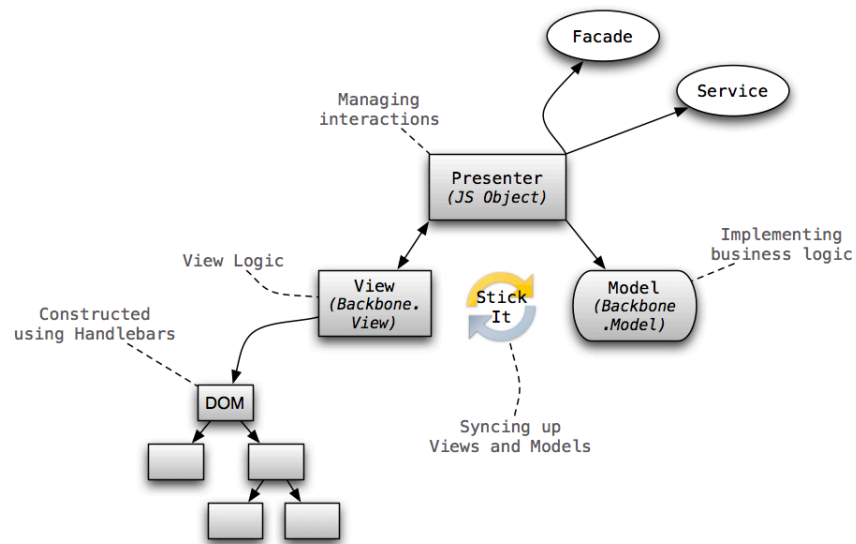


Figura 38, Backbone

A continuació es mostra el disseny de les pantalles de l'aplicació i la jerarquia de les vistes més importants.

8.4.3.1 Disseny de les pantalles

8.4.3.1.1 Pantalla d'autenticació

The screenshot shows a web browser window titled "Pàgina autenticació". At the top, there are three input fields: "Email" (containing "Adreça electrònica"), "Contrassenya" (containing "Contrassenya"), and an "Enviar" button. Below this, there is a "Registrar" form with the following elements:

- Two input fields for "Nom" and "Cognoms".
- An input field for "Adreça electrònica".
- Two input fields for "Contrassenya".
- A checkbox labeled "Acceptar condicions d'ús i galetes."
- An "Enviar" button.

Figura 39, disseny: pàgina d'autenticació

8.4.3.1.2 Pantalla principal

The screenshot shows a web browser window titled "Pàgina principal". At the top, there are several navigation and utility elements:

- Buttons for "Inici", "Perfil", "Grups" (with a dropdown arrow), "Buscar" (with an input field), "Configuració", and "Tancar".

The main content area is divided into several sections:

- "Cançons populars": A large empty rectangular box.
- "Publicacions": A large empty rectangular box on the left side.
- "Publicació": A section on the right side containing an input field with the placeholder text "Escriure publicació...".
- "Notificacions": A large empty rectangular box on the right side.
- "#Populars": A large empty rectangular box at the bottom right.

Figura 40, disseny: pàgina principal

8.4.3.1.3 Pantalla perfil usuari

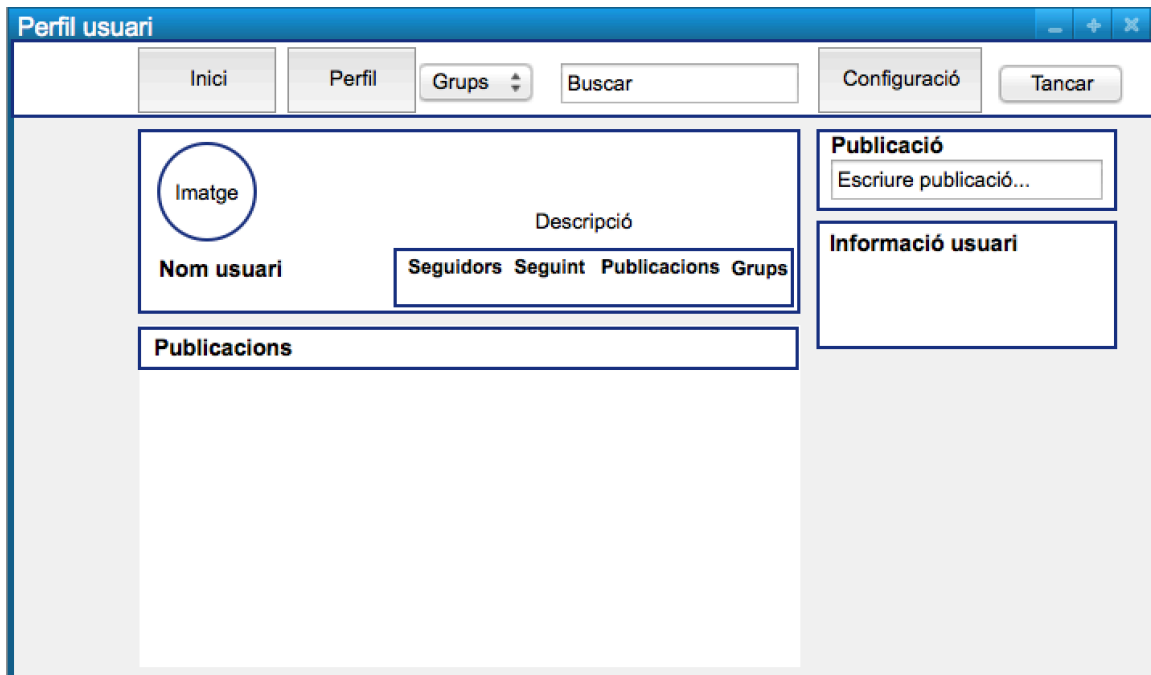


Figura 41, disseny pàgina perfil d'usuari

8.4.3.1.4 Pantalla perfil grup

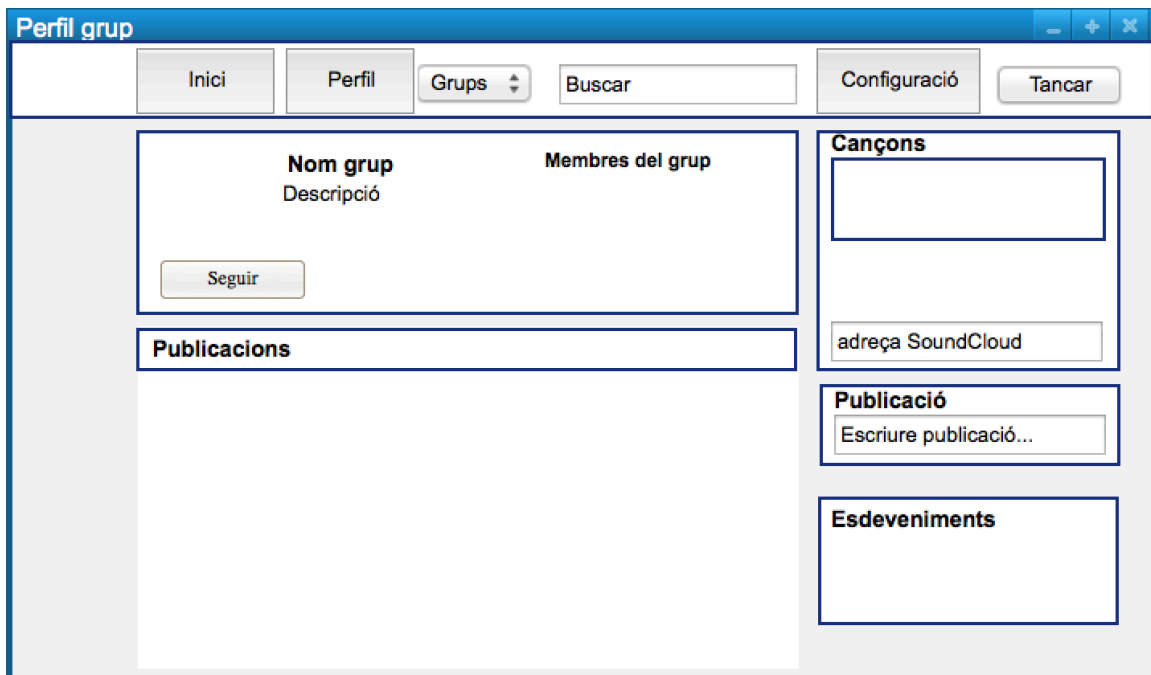


Figura 42, disseny pàgina perfil de grup

8.4.3.2 Jerarquia de vistes

8.4.3.2.1 Vista autenticació

És la vista encarregada de registrar un usuari i iniciar sessió d'un usuari. Si l'autenticació o el registre és fan correctament aquesta envia a la pàgina principal de l'aplicació.

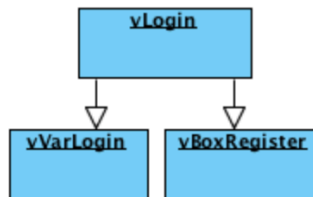


Figura 43, Vista autenticació

8.4.3.2.2 Vista cercar

Per poder mostrar aquesta vista s'ha d'enviar la cadena de caràcters que es vol buscar, la vista de cerca s'encarrega de mostrar els resultats d'aquesta cerca.

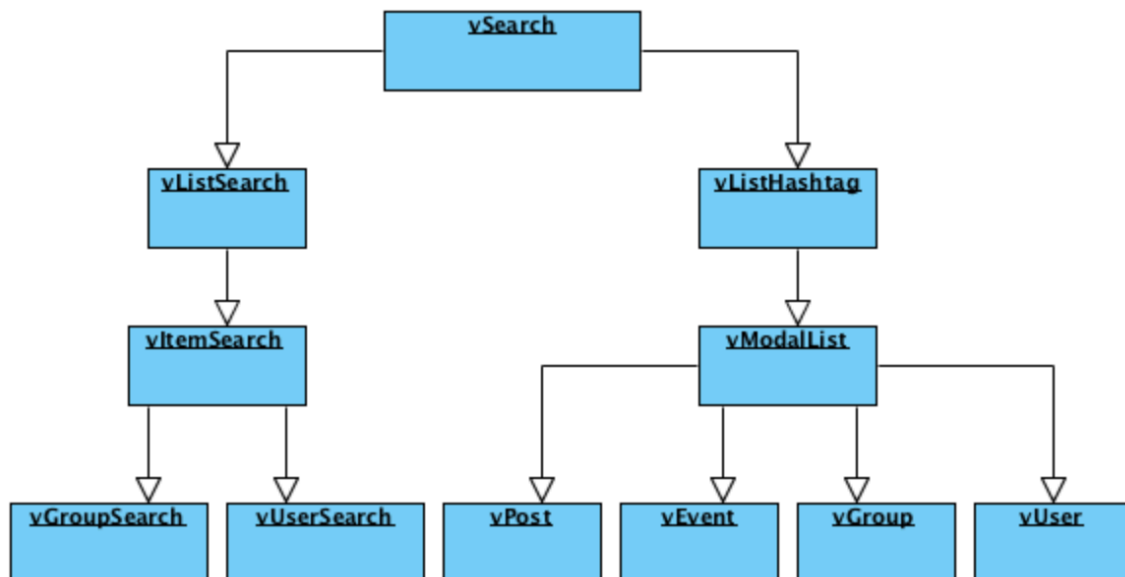


Figura 44, vista cercar

La vista *vListSearch* mostra el contingut de la cerca en el bloc gran de la dreta de la pantalla, per el contrari *vListHashtag* mostra la cerca de hashtags, els quals poden ser clicats, gracies a la vista *vModallist*, per veure les publicacions que contenen aquests hashtags.

8.4.3.2.3 Vista perfil usuari

La vista del perfil de l'usuari és de les més completes en quant a subvistes. S'encarrega de mostrar la informació de l'usuari del perfil, de poder fer publicacions dirigides a l'usuari i mostrar les publicacions.

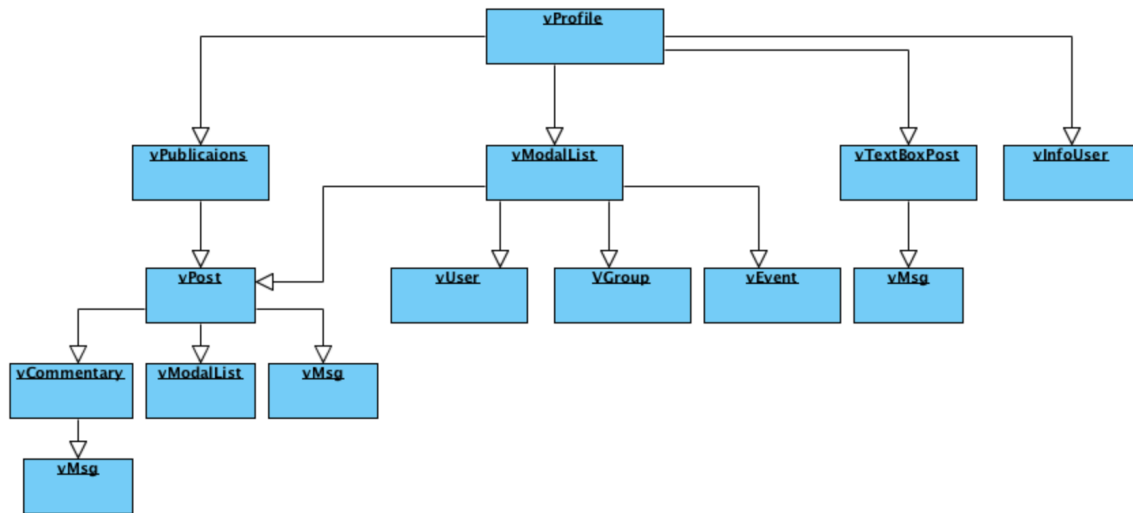


Figura 45, vista perfil usuari

La vista de **vPost** s'encarrega de mostrar una publicació i de la llista de comentaris, i les opcions de les publicacions, **vCommentary** és semblant a **vPost** però amb comentaris.

La vista **vTextBoxPost** s'encarrega de fer una publicació i actualitzar la llista de publicacions (**vPublicacions**).

vInfoUser s'encarrega de mostrar la informació de l'usuari.

La vista **vModalList** és la vista, re-usable, que mostra una llista en un modal, semblant a un *popup*. Les vistes **vUser**, **vGroup** i **vEvent** mostren la informació bàsica de usuaris, grups i esdeveniments, respectivament, en un petit *layout*.

vMsg són uns missatges que es mostren a la cantonada esquerra superior de la pantalla, serveix per notificar l'usuari de possibles errors o missatges d'accions completades satisfactòriament.

8.4.3.2.4 Vista principal

La vista principal mostra tota la informació sobre les publicacions dels usuaris i grups que segueixes, de les tendències actuals i de les notifikacions.

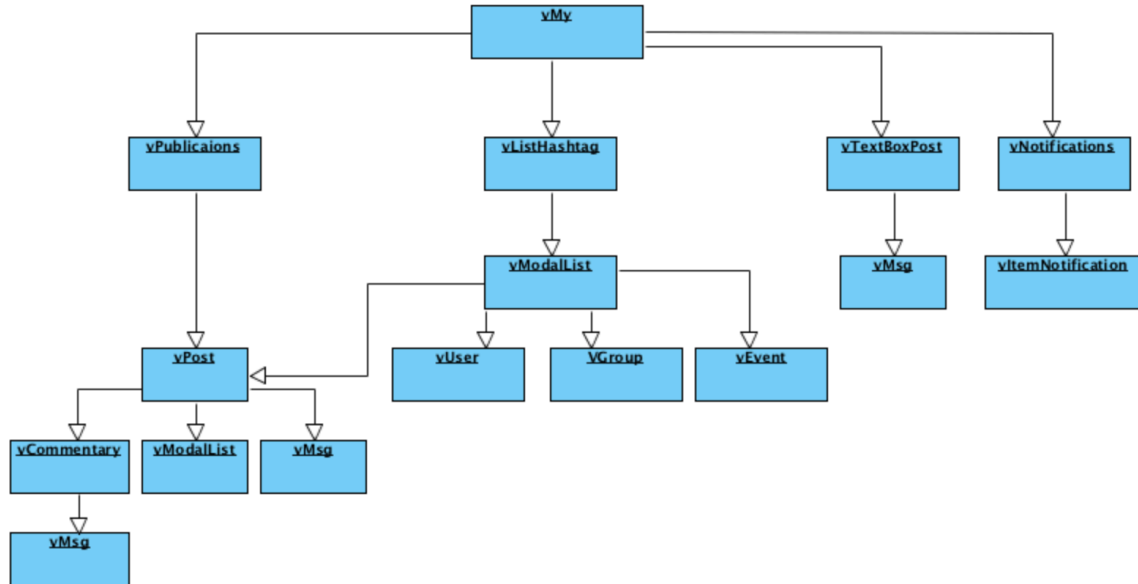


Figura 46,vista principal

En el cas de la vista principal *vPublicacions* mostra les publicacions fetes i/o dirigides per usuaris o grups que segueix l'usuari, en el perfil de usuaris i grups només mostra les publicacions fetes i/o dirigides al propietari del perfil.

vListHashtags mostra els 10 hashtags més populars de les últimes 24 hores, té com a principal avantatge, que clicant sobre un dels hashtags, mostrar els últimes 50 publicacions amb el hashtag clicat.

vNotifikacions mostra les ultimes 4 notifikacions, per poder veure totes les notifikacions hi ha la opció de "Veure més". *vItemNotifikacion* és la vista d'una notifikació en particular.

8.4.3.2.5 Vista perfil grup

Té la mateixa funció que la vista de perfil de l'usuari, però en aquest cas mostra la informació del grup, les publicacions, les cançons del grup i els seus esdeveniments més pròxims.

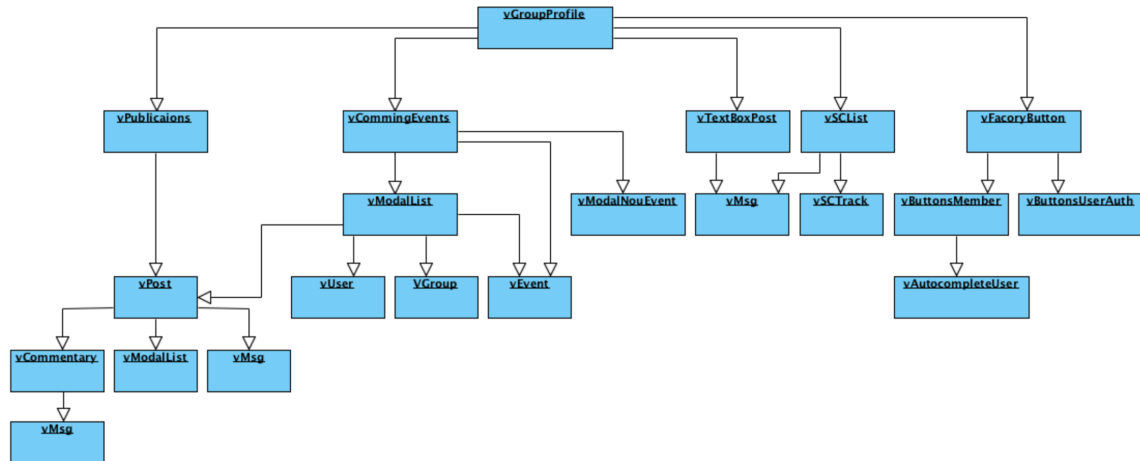


Figura 47, vista perfil de grup

La vista *vCommingEvents* mostra els pròxims 4 esdeveniments del grup, també es pot visualitzar tots els pròxims esdeveniments amb fent clic a “Veure més”. Permet crear nous esdeveniments al fundador i als membres administradors del grup.

vSCList mostra les últimes 5 cançons que el grup a publicat. També permet als membres administradors i a fundador afegir més cançons.

vFacoryButton mostra els botons de la capçalera del perfil. Si no ets membre del grup mostra el botó de seguir (*vButtonsUserAuth*), en cas contrari, si ets membre del grup mostra les opcions de configuració del grup (*vButtonsMember*).

vAutoCompleteUser és una vista que ajuda a la cerca de usuaris.

8.4.3.2.6 Vista configuració usuari

La vista de configuració de l'usuari permet a l'usuari canviar la seva informació personal i crear un grup de música. Està dividit en 4 parts:

- Descripció i foto personal: permet modificar la descripció de l'usuari i canviar la seva foto de perfil.
- Dades personals: permet canviar el nom, cognoms de l'usuari, data de naixement i el sexe.
- Adreça: per la modificació de la ciutat i país on viu.
- Grup: permet la creació d'un grup.

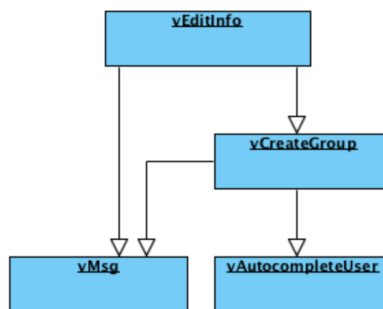


Figura 48, vista configuració usuari

vCreateGroup permet crear un grup, indicant el nom, la data de fundació, estil de musica, descripció i els membres que el componen.

8.4.3.2.6 Vista barra superior

La barra superior és una botonera per navegar per l'aplicació, té els botons per anar a la pàgina principal, al perfil de l'usuari actual, configuració de l'usuari i per tancar la sessió actual, també té una entrada per cercar i anar a la pàgina de cerca.



Figura 49, vista barra superior

9. IMPLEMENTACIÓ I PROVES

En aquest apartat s'explicarà amb més detall les implementacions de Hibernate, La configuració del framework Backbone, les APIs de YouTube i SoundCloud.

9.1 HIBERNATE

Per fer el mapeig de les classes en taules de la base de dades s'han d'incorporar diverses anotacions per tal de dotar de diferents propietats als objectes o tipus bàsics que volem representar i associar a la taula de la base de dades. Aquestes anotacions s'afegeixen a sobre de l'atribut que creem i podem trobar les següents:

- `@Entity`: declara que la classe és una entitat.
- `@EmbeddedId`: declara l'identificador de la nostra entitat , que és la clau primària de la taula.
- `@Column`: s'utilitza per indicar diferents atributs de la columna, com és el nom real de la base de dades (`name`), indicar si aquest camp conté valor únic o no (`unique = false/true`), si pot contenir un valor null (`nullable = false/true`), la longitud del camp (`length`) o fins i tot indicar si aquest camp es pot modificar (`updatable = true/false`).
- `@JoinColumn`: permet indicar que es fa referència a un camp d'una altra taula des d'aquesta. Li hem d'especificar a quina columna de l'objecte que hem declarat fa referència (`referencedColumnName`) i també pot tenir els mateixos atributs que l'anotació `Column`.
- `@ManyToOne`: defineix que tenim una relació de molts a un.

Un tros de codi d'exemple seria:

```
//File: Auth.java
@Entity
public class Auth extends Model implements Serializable {

    @NotNull
    @Column(name = "EMAIL", length = 50,unique = true)
    private String email;

    @XmlTransient
    @NotNull
    @NotEmpty
    @Column(name = "PWD", length = 50)
    private String password;

    @OneToOne( fetch = FetchType.EAGER)
    @JoinColumn(name = "IdUSER", referencedColumnName="ID")
    private User user;
```

Per fer les consultes a la base de dades, Hibernate ens proporciona l'objecte `EntityManager` que s'encarrega de les operacions de les base de dades, com a mètodes més destacables tenim els següents:

- `createQuery(String)`: on li passem una sentència amb `String` i ens retorna un objecte `Query`, amb el qual podem fer operacions com: `getResultList`, `getSingleResult`.
- `persist(Object)`: li passem per paràmetre l'objecte que volem guardar a la base de dades.
- `merge(Object)`: li passem l'objecte modificat que volem actualitzar a la base de dades.
- `Find(class, Long)`: Li donem la id del objecte i la classe que pertany l'objecte que volem buscar per la seva id i ens el retorna.

Un tros de codi d'exemple seria:

```
public class AuthService {

    @PersistenceContext
    private EntityManager em;

    public Auth matchPassword(String email, String password) {

        Query q = em.createQuery("select au from Auth au where au.email=:email");
        q.setParameter("email", email);

        Auth au = (Auth) q.getSingleResult();
        return au.getPassword().equals(password) ? au : null;

    }

    public Auth save(Auth auth) {
        em.persist(auth);
        em.flush();
        return auth;
    }

    public Auth findByEmail(String email) {

        Query q = em.createQuery("select au from Auth au where au.email=:email");
        q.setParameter("email", email);

        try {
            return (Auth) q.getSingleResult();
        }
        catch (Exception e) {
            return null;
        }

    }

}
```

9.2 BACKBONE

Per organitzar millor l'aplicació amb Backbone he utilitzat la tecnologia AMD (*Asynchronous Module Definitions*) amb la llibreria requireJS.

9.2.1 AMD

AMD és per carregar codi modular de manera asíncrona, per fer un símil amb el Java és per fer les importacions d'altres classes. Això proporciona una millor estructura del codi JavaScript ja que per el mateix no té cap.

9.2.2 REQUIREJS

RequireJS és una llibreria que ens proporciona AMD i molt utilitzada per la sinergia que proporciona amb Backbone.

9.2.3 ESTRUCTURA

L'aplicació client està estructurada de la manera que recomana Backbone, amb aquesta estructura és *single-page*, tot s'organitza amb vistes i *templates*, també podem crear objectes *models*, a continuació la estructura de l'aplicació.

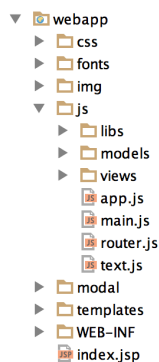


Figura 50, estructura de l'aplicació web

En la carpeta *views* estan les vistes generals de l'aplicació com: perfil de grup, perfil usuari, pàgina de configuració, entre altres. Per altre banda en la carpeta *templates* i *modal* està el "html" dinàmic que podem anar posant o treien segons vulguem, també hi han les carpetes *css*, *font* i *img* que són les típiques de les pàgines web.

Backbone et dona una eina que és molt útil per la navegació de la web, aquesta està configurada en l'arxiu *router.js* et deixa definit unes rutes per defecte per poder anar a la vistes més generals de l'aplicació, per exemple la ruta "host/#group" anirà a la vista de perfil de grup. Aquest mecanisme és molt útil per la navegació entre "pàgines" de l'aplicació web, també et permet passar dades com pot ser, seguint l'exemple anterior "host/#group/:id" i aniria al perfil del grup amb id = :id. Per tant crea un VC (Vista-Controlador).

9.2.4 CONFIGURACIÓ

Per a la configuració de Backbone + requireJS, s'ha de fer un *import* del mòdul principal de l'aplicació des de l'única pàgina html.

```
<!doctype html>
<html lang="en">
<head>
  <title>SocialSong</title>
  <script data-main="js/main" src="js/libs/require/require.js"></script>
  <script src="http://connect.soundcloud.com/sdk.js"></script>
  <link rel="stylesheet" type="text/css" href="/css/jquery.datetimepicker.css" />
</head>
<body>
<div id="varApp"></div>
<div id="contextApp"></div>
<div id="modalApp" class="modal fade"></div><!-- /.modal -->
</body>
</html>
```

Una vegada fet la importació del *main*, s'ha de configurar. Per la configuració s'ha de posar les llibreries bàsiques de la l'aplicació de la següent forma:

```
// Filename: main.js
require.config({
  paths: {
    jquery: 'libs/jquery/jquery-2.1.0.min',
    datetimepicker: 'libs/jquery/jquery.datetimepicker',
    underscore: 'libs/underscore/underscore-min',
    backbone: 'libs/backbone/backbone-min',
    bootstrap: 'libs/bootstrap/bootstrap.min',
    text: 'text',
    utils: 'libs/utils/utils'
  }
});
require(['router'], function(Router){

  var router = new Router;

});
```

Una vegada posat el nom i les rutes de les biblioteques externes que farem servir, iniciem l'aplicació amb el mòdul *Router*, és l'encarregat de fer el MVC, en ell se li posen les rutes de la nostre aplicació i la vista la qual pertany.

En el cas d'aquesta aplicació web, s'inicia amb la vista login, es pot veure en la funció *initialize* de l'arxiu router.

En aquest arxiu també es pot comprovar com es carrega els mòduls, els hi posem la ruta en i el nom en la funció *define*, en el primer paràmetre la ruta dels mòduls, només cal posar el nom del mòdul si està definida en el *main*, i en el segon paràmetre es la funció, on tenim la lògica del mòdul, que rep per paràmetre les modules definits anteriorment.

```
// Filename: router.js
define([
  'jquery', 'underscore', 'backbone', 'bootstrap',
  'views/vLogin', 'views/vMy', 'views/vProfile',
  'views/vSearch', '/vEditInfo', 'views/vGroup'
], function($, _, Backbone, Bootstrap, Login, My, vProfile, vSearch, vEditInfo, vGroup){
  var AppRouter = Backbone.Router.extend({
    routes: {
      "": "showMy",
      "login": "showLogin",
      "my": "showMy",
      "profile/:id": "showProfileById",
      "group/:id": "showGroupById",
      "search/:text": "showSearch",
      "edit": "showEdit"
    },
    initialize : function(){
      var login = new Login();
    },
    //functions here
  });
  Backbone.history.start();
  return AppRouter;
});
```

9.5 SOUND CLOUD API

Per poder utilitzar l'API de SoundCloud prèviament exigeixen un registre l'aplicació, en el qual ens dona un codi per poder fer les consultes al seu servei web.

9.5.1 CONFIGURACIÓ

L'únic que cal per fer servir l'API de SoundCloud, apart de registrar-se, és la llibreria que ens proporcionen, que la tenim que importar en la nostra aplicació, per la comoditat he fet la importació directament a la pàgina *html*.

```
<!doctype html>
<html lang="en">
<head>
  <title>SocialSong</title>
  <script data-main="js/main" src="js/libs/require/require.js"></script>
  <script src="http://connect.soundcloud.com/sdk.js"></script>
  <link rel="stylesheet" type="text/css" href="/css/jquery.datetimepicker.css" />
</head>
<body>
<div id="varApp"></div>
<div id="contextApp"></div>
<div id="modalApp" class="modal fade"></div><!-- /.modal -->
</body>
</html>
```

9.5.2 UTILITZACIÓ

Per obtenir la cançó tenim que tenir la adreça de la cançó a SoundCloud, que ens la proporcionarà l'usuari.

La llibreria que proporciona SoundCloud conté l'objecte *SC* que ens dona les opcions per demanar l'*iframe* que tenim que posar en el codi HTML. El tros de codi per obtenir la cançó en el format que hi ha a l'aplicació és el següent.

```
getIframeSoundCloud : function(track_url,_div){

    SC.initialize({
      client_id: "0faxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
    });
    //var track_url = "https://soundcloud.com/greenday/12-the-forgotten";

    SC.oEmbed(track_url, {
      auto_play: false,
      show_comments: false,
      sharing : false,
      show_artwork: true,
      maxheight: 200}, function(oEmbed) {
      _div.prepend(oEmbed.html)
    });
}
```

L'objecte SC el tenim que inicialitzar amb el codi que ens proporciona SoundCloud, seguidament cridem a la funció *oEMbed*, que té 3 paràmetres, el primer la adreça de la cançó, el segon una llista amb els paràmetres que volem, i el tercer la funció que farem servir, en la funció mostrada es fa un *prepend* (*append* però al principi) al *div* passat per paràmetre, del *iframe* que ve en el atribut *html* de l'objecte *oEmbed*.

9.6 YOUTUBE API

Per tenir un vídeo de YouTube és relativament senzill envers a SoundCloud.

Per obtenir el vídeo s'ha de tenir la adreça del vídeo a YouTube, seguidament s'ha de agafar la ID del vídeo, que està en l'adreça. Exemple:

```
https://www.youtube.com/watch?v=Soa3gO7tL-c
```

El codi del vídeo d'aquesta adreça és "Soa3gO7tL-c", el codi sempre està ratera de "watch?v=".

Una vegada obtingut el codi, l'únic que s'ha de fer és el següent *iframe*. L'atribut *src* del *iframe* ha de ser: `src="http://www.youtube.com/embed/<code>"` .

```
putYouTubeVideo : function (code, _div){
    _div.prepend('<iframe class="vidioYoutube" '+
        'src="http://www.youtube.com/embed/'+code+' ">'+
        '</iframe>');
}
```

10. IMPLEMENTACIONS I RESULTATS

10.1 IMPLEMENTACIÓ DEL SERVIDOR JBOSS LOCAL

Per poder dur a terme la comunicació entre el servidor i la base de dades s'ha tingut que seguir el següents passos:

- Descarregar i posar el *driver* de MySQL de JBoss a la carpeta "modules/com" del servidor.
- Crear a la base de dades (SocialSongDB), amb tots els permisos, per el servidor JBoss. En el meu cas he creat l'usuari "tfg_projecte" amb la contrasenya "tfg".
- Posar el següent *datasource* la configuració del JBoss, l'arxiu *standalone.xml*.

```
<datasource jndi-name="java:jboss/datasources/MySQLDS" pool-name="MySQLDS"
enabled="true" use-java-context="true">

  <connection-url> jdbc:mysql://localhost:3306/SocialSongDB </connection-url>
  <driver>com.mysql</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <prefill>true</prefill>
  </pool>
  <security>
    <user-name>tfg_projecte</user-name>
    <password>tfg</password>
  </security>
  <statement>
    <prepared-statement-cache-size>32</prepared-statement-cache-size>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
```

- Afegir el driver en l'apartat de *drivers*.

```
<driver name="com.mysql" module="com.mysql">

  <xa-datasource-class>com.mysql.jdbc.Driver</xa-datasource-class>

</driver>
```

Una vegada configurat el JBoss amb la base de dades, només s'ha de configurar el JBoss en l'entorn de programació però poder-lo fer servir.

10.2 OPENSIFT

La implementació de la web en OpenShift és relativament senzilla, s'ha de tenir una compte en OpenShift i seguir les instruccions per crear una aplicació JBoss escalable. Una vegada es crea l'aplicació es té que configurar l'arxiu persistence.xml, i pujar el codi al repositori que t'especifiquen amb el sistema GIT.

```
<jta-data-source>java:jboss/datasources/MySQLDS</jta-data-source>
  <properties>
    <!-- Properties for Hibernate create-drop -->
    <property name="hibernate.hbm2ddl.auto" value="update" />
    <property name="hibernate.show_sql" value="false" />
    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect"/>
  </properties>
```

Si es crea una aplicació escalable pots posar més potencia (*gears*) a l'aplicació, en la conta gratuïta només es disposen de 3 *gears* dels quals he posat 1 en la base de dades MySQL i 2 en el servidor JBoss.

L'adreça de l'aplicació web és: <http://webapp-socialsong.rhcloud.com>

10.3 RESULTATS

10.3.1 REGISTRE

Registre de l'usuari Joan Pi (figura 51), una vegada que registrem l'usuari automàticament anem a la pantalla general i tenim una notificació del sistema, per si volem completar el nostre perfil (figura 52).



Figura 51, registre usuari

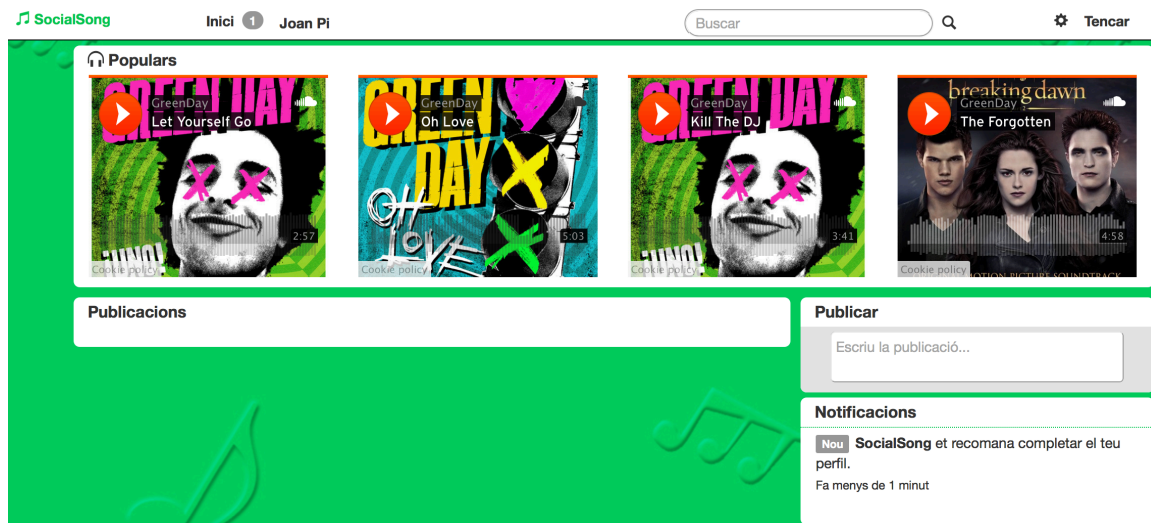


Figura 52, resultat del registre usuari

10.3.2 EDITAR

Modificarem les seues dades, si la modificació de les dades ha sigut correcte, ens surt el missatge de l'esquerra (figura 53), avisant que s'ha efectuat bé la modificació de dades, en cas que no s'hagués modificat correctament també mostraria un missatge d'error. En la figura 54 podem comprar que s'han registrat bé les dades.

The screenshot shows a web interface for editing user information. At the top left, a green box contains a checkmark and the text 'Informació Editada.'. The main form is titled 'Editar Informació' and is divided into three sections: 'Principal', 'Adreça', and 'Grup'. Each section has a 'modificar' link on the right. The 'Principal' section contains input fields for 'Nom' (Joan), 'Cognom' (Pi), 'Data de naixement' (18/04/1991), and 'Sexe' (Home), along with dropdown menus for 'Tipus de perfil' and 'Visibilitat de perfil', both set to 'Public'. A green 'Guardar' button is present. The 'Adreça' section contains input fields for 'Ciutat' (Girona) and 'País', and a dropdown menu for 'Tipus de perfil' set to 'Public'. A green 'Guardar' button is also present. The 'Grup' section is currently empty. At the bottom, there is a link for 'Condicions d'ús i galetes.'

Figura 53, Modificar dades

Información General	
Sexe	Home
Cumpleanys	18/04/1991
Ciutat	Girona

[Editar](#)

Figura 54, informació de l'usuari

10.3.3 PERFIL D'USUARI

Si no seguim a l'usuari no podem veure, ni escriure a l'usuari del perfil (Figura 55), per el contrari si seguim aquest usuari podem visualitzar tota la informació (Figura 56).

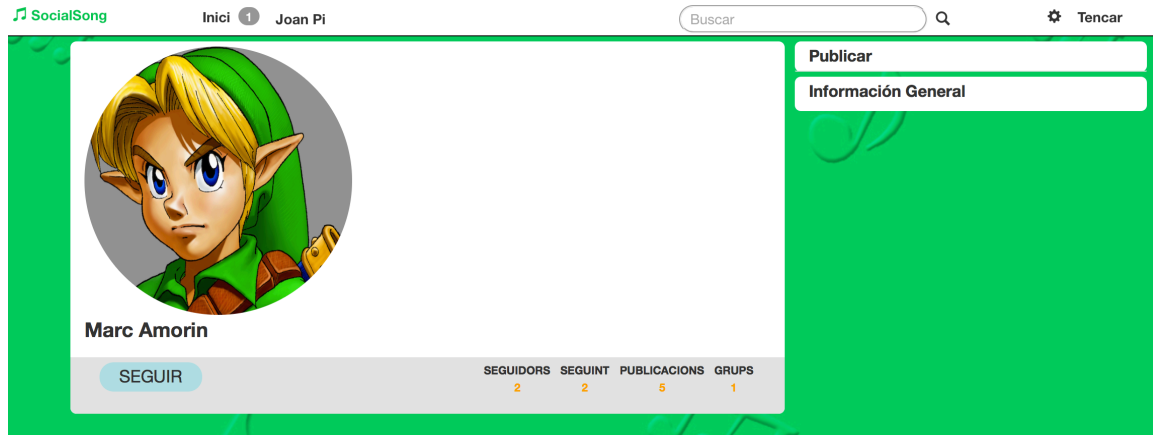


Figura 55, perfil d'usuari (sense seguir)

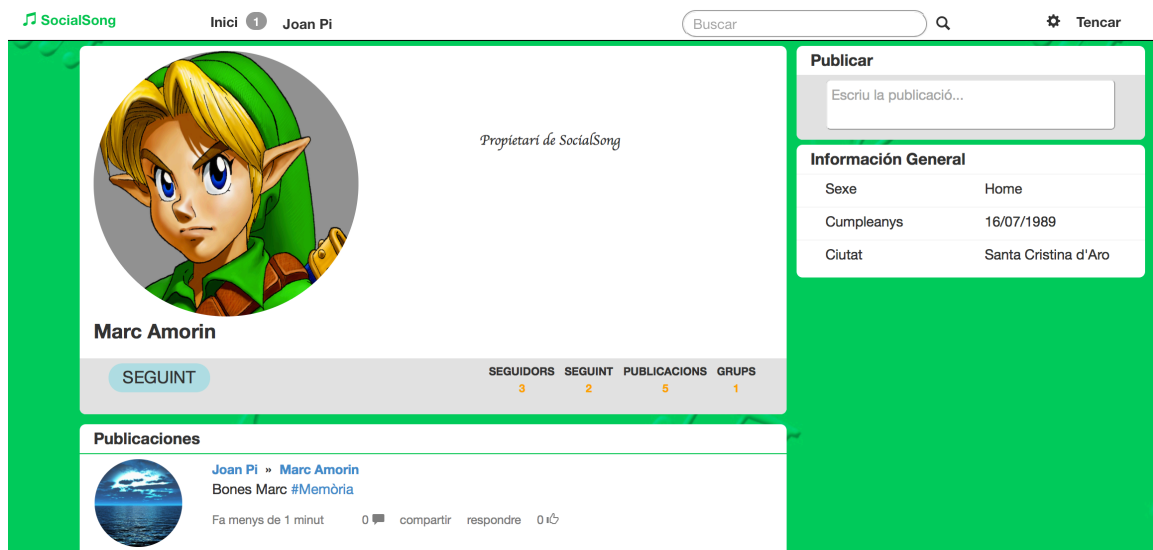


Figura 56, perfil d'usuari (seguint)

10.3.4 PUBLICACIÓ

Per fer una publicació s'ha de fer en la caixa de "Publicar" (figura 57), no es poden fer més de 140 caràcters, una vegada publicada es mostra automàticament amb les altres publicacions (figura 58).

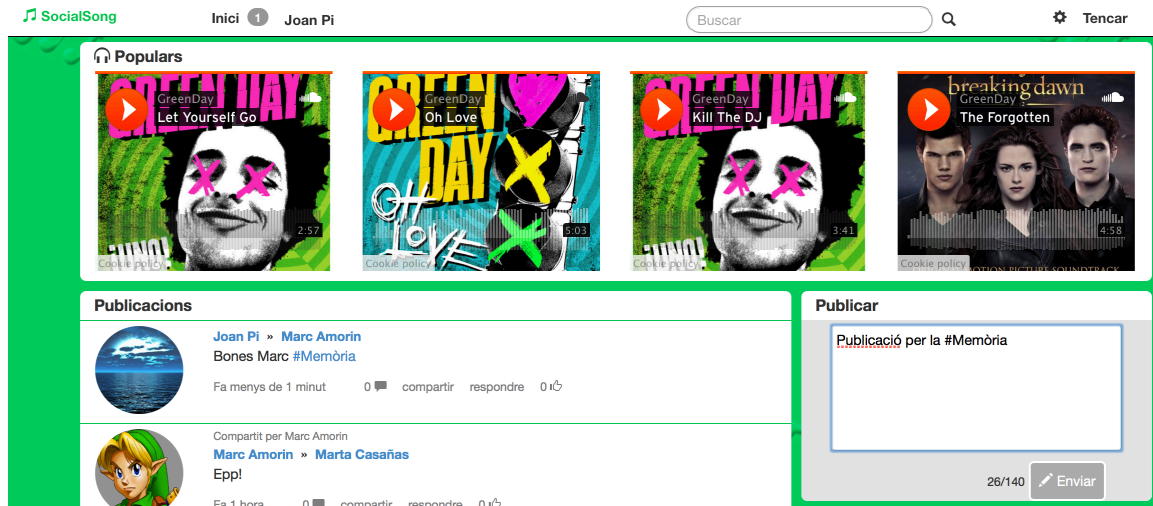


Figura 57, fer una publicació (I)

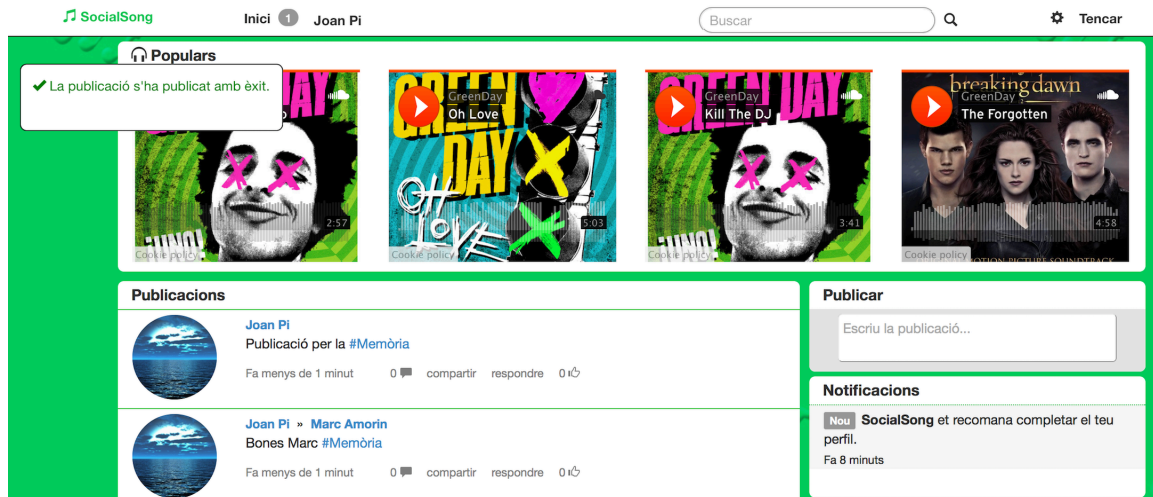


Figura 58, fer una publicació (II)

10.3.4 COMENTARI

Per fer una comentari, es té que clicar en la opció “respondre” de la publicació que volem comentar, seguidament es mostra un caixa de text, on podem escriure 140 caràcters (figura 59), per publicar el comentari s’ha de donar al la tecla enter del teclat. Finalment s’actualitza la publicació mostrant el comentari (figura 60).

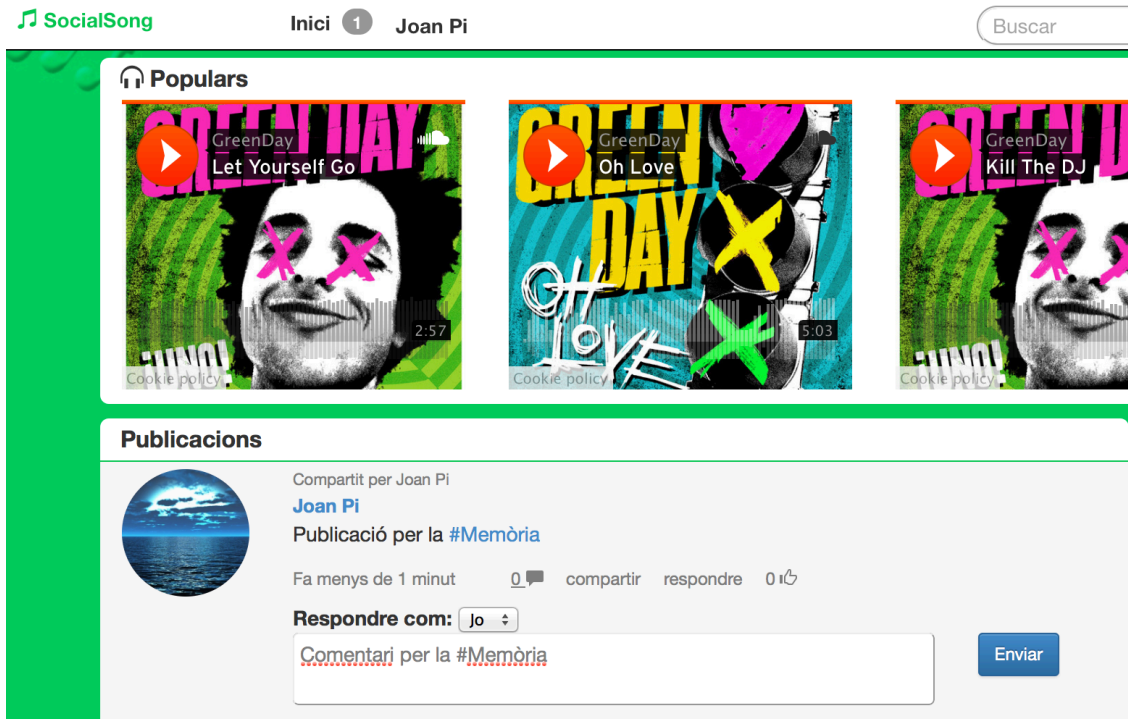


Figura 59, fer un comentari (I)

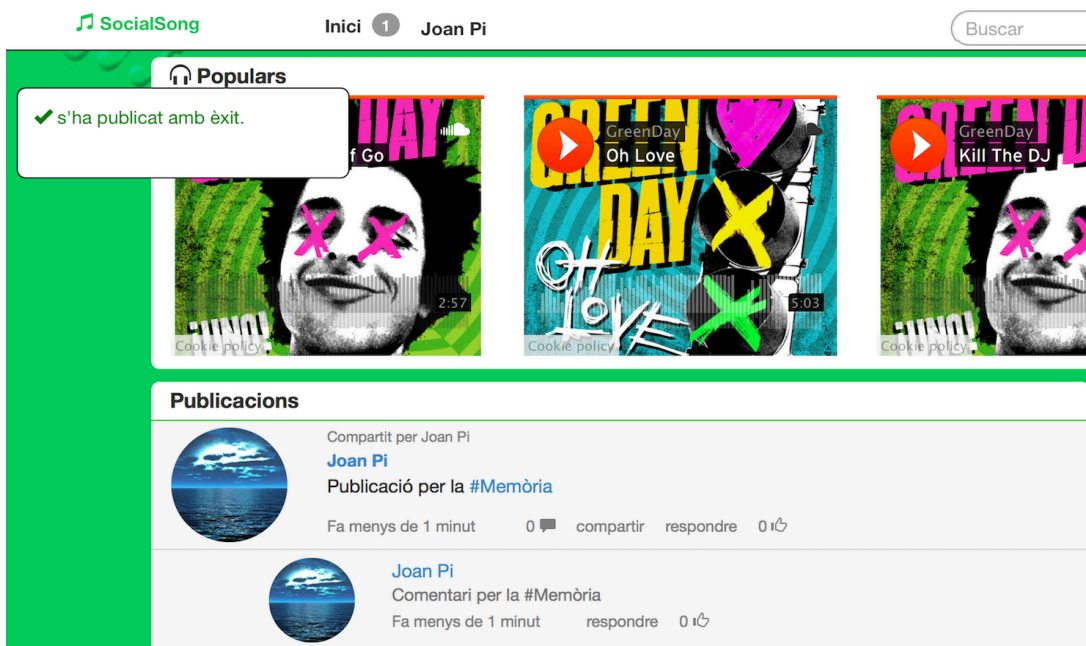
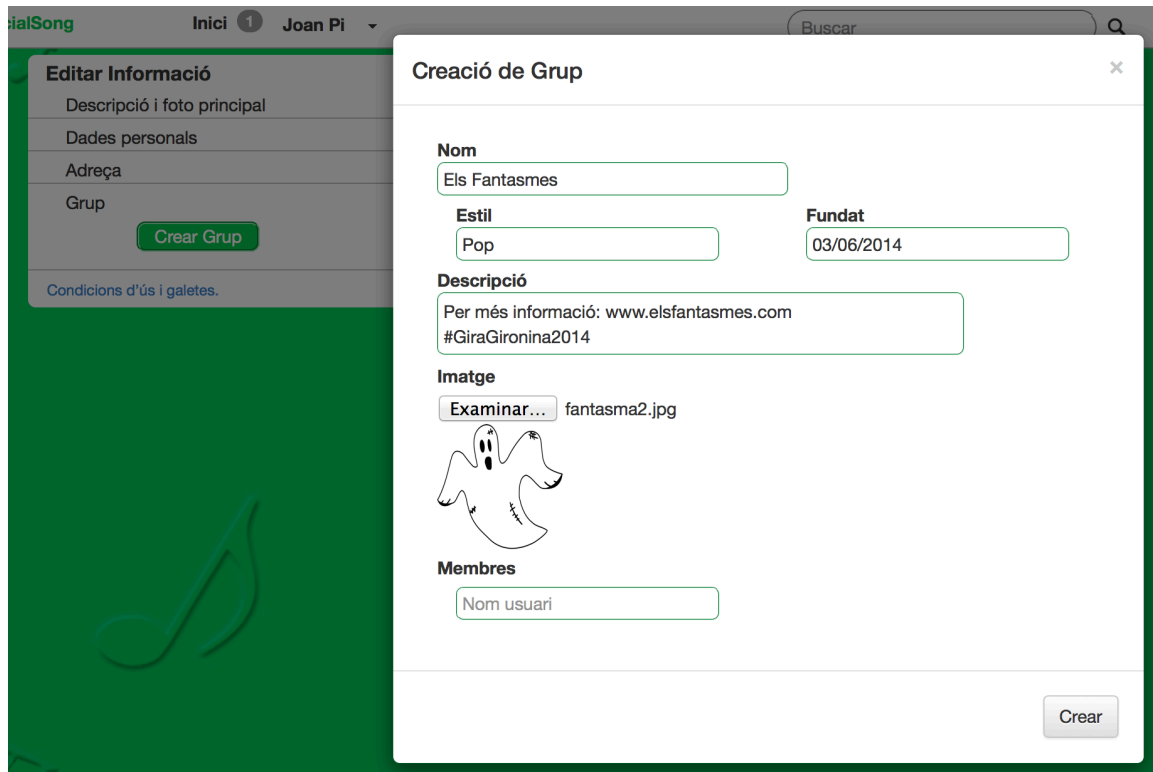


Figura 60, fer un comentari (II)

10.3.5 CREAR UN GRUP

En la configuració general d'un usuari hi ha l'apartat "Grup" que conté l'opció "Crear grup", si es clica es mostra una finestra amb un formulari, per crear el grup només s'ha de omplir aquest formulari i donar el botó de "Crear" (figura 61). Finalment es mostra el perfil del grup (figura 62).

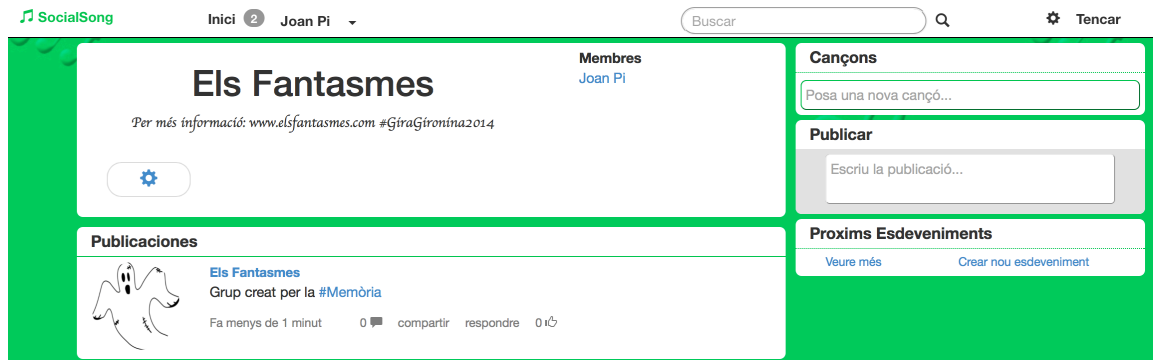


The screenshot shows a web browser window with the SocialSong application. A modal window titled "Creació de Grup" is open over the "Editar Informació" page. The form contains the following fields:

- Nom:** Els Fantasmes
- Estil:** Pop
- Fundat:** 03/06/2014
- Descripció:** Per més informació: www.elsfantasmes.com #GiraGironina2014
- Imatge:** fantasma2.jpg (with a preview of a ghost icon)
- Membres:** Nom usuari

A "Crear" button is located at the bottom right of the modal.

Figura 61, crear grup (I)



The screenshot shows the profile page for the group "Els Fantasmes". The page includes the following sections:

- Header:** SocialSong logo, "Inici 2 Joan Pi", search bar, and "Tancar" button.
- Group Name:** Els Fantasmes, with "Membres Joan Pi" next to it.
- Description:** Per més informació: www.elsfantasmes.com #GiraGironina2014
- Cançons:** A text input field with the placeholder "Posa una nova cançó...".
- Publicar:** A text input field with the placeholder "Escriu la publicació...".
- Proxims Esdeveniments:** "Veure més" and "Crear nou esdeveniment" buttons.
- Publicaciones:** A post from "Els Fantasmes" titled "Grup creat per la #Memòria", posted "Fa menys de 1 minut". It includes a ghost icon and interaction buttons for "compartir", "respondre", and "0 0".

Figura 62, crear grup (II)

10.3.6 SOUND CLOUD

Per posar una cançó en el grup, s'ha de posar l'adreça de la cançó de SoundCloud al input de l'apartat "Cançons" de la dreta del perfil de grup (figura 63), i prémer la tecla enter del teclat, seguidament la llista s'actualitza mostrant en gran la última cançó inserida (figura 64).



Figura 63, posar cançó SoundCloud (I)

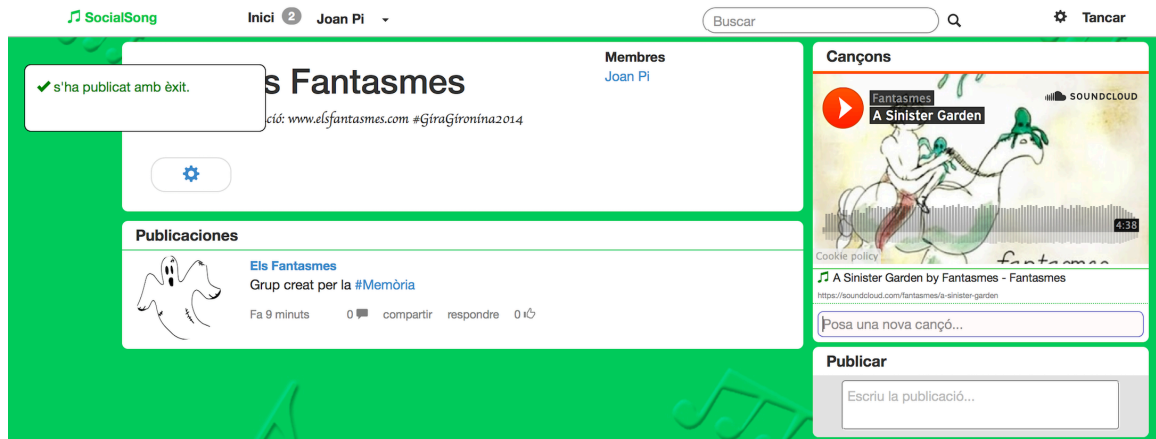


Figura 64 posar cançó SoundCloud (II)

10.3.7 INVITAR A MEMBRE

Per invitar a un membre s'ha d'anar a configuració del grup, a la pestanya de membres, i clicar el botó "+", buscar-lo (figura 65) i clicar sobre l'avió de paper (figura 66), automàticament s'actualitza la llista de membres (figura 67), finalment s'envia la invitació (figura 68).

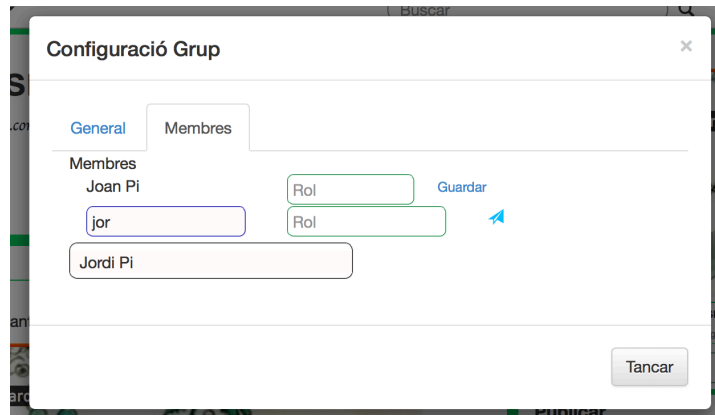


Figura 65, invitar membre (I)

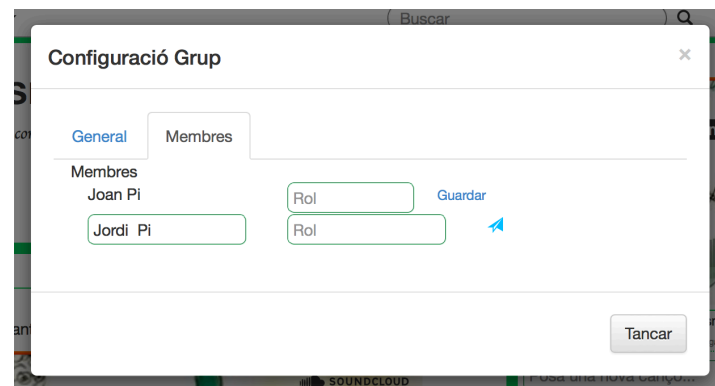


Figura 66, invitar membre (II)

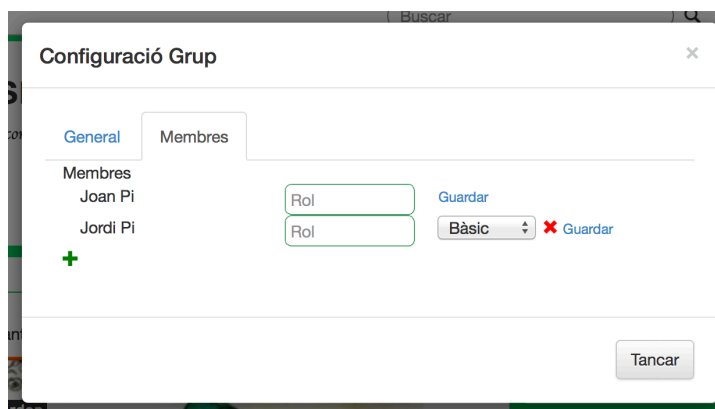


Figura 67, invitar membre (III)

Notificacions

Nou Tens una invitació per ser membre de **Els Fantasmés**

Acceptar **Denegar**

Fa menys de 1 minut

Nou **Joan** t'està seguint.

Fa 3 minuts

Figura 68, invitació de grup

10.3.8 YOUTUBE

Per publicar un vídeo de YouTube és el mateix sistema que una publicació normal, però a més a més li posem en el cos del missatge, entre els 140 caràcters, l'adreça del vídeo (figura 69), finalment s'actualitza la llista de publicacions (figura 70).

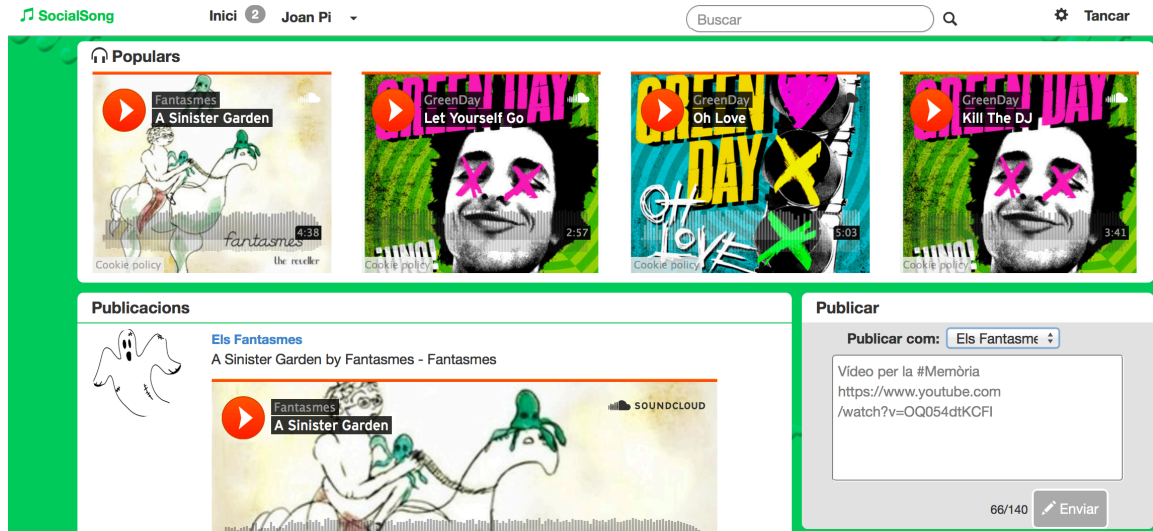


Figura 69, vídeo YouTube (I)

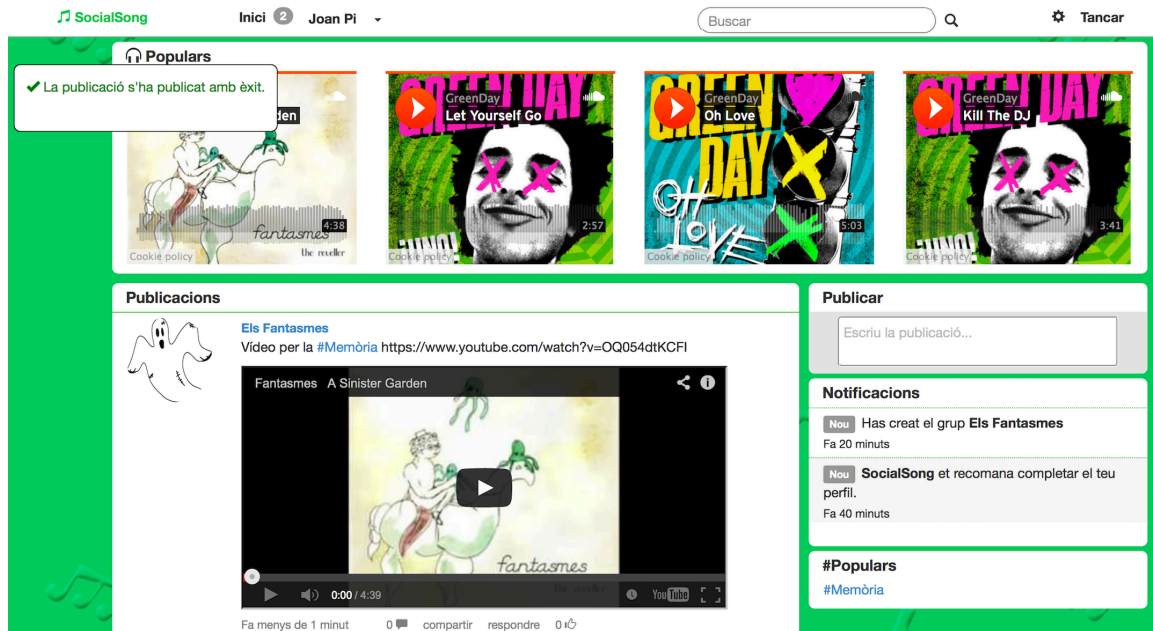


Figura 70 vídeo YouTube (II)

10.3.9 BUSCAR

Per cercar només s'ha de posar el nom del que volem cerca a la barra superior i prémer la tecla enter del teclat (figura 71). Seguidament mostra els resultats (figura 72).

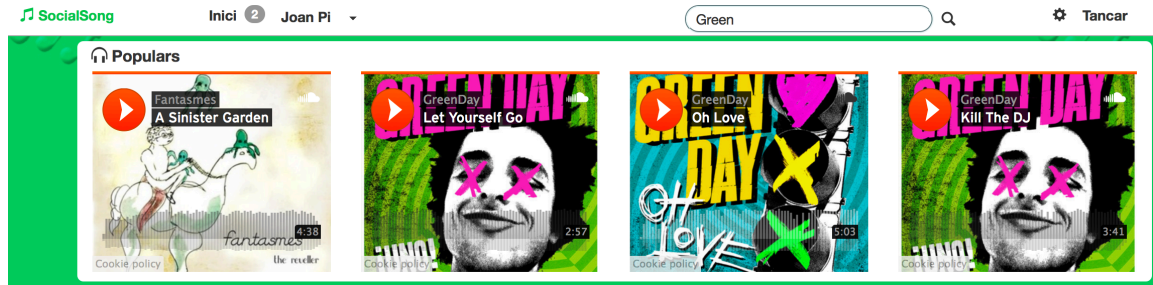


Figura 71, cercar (I)

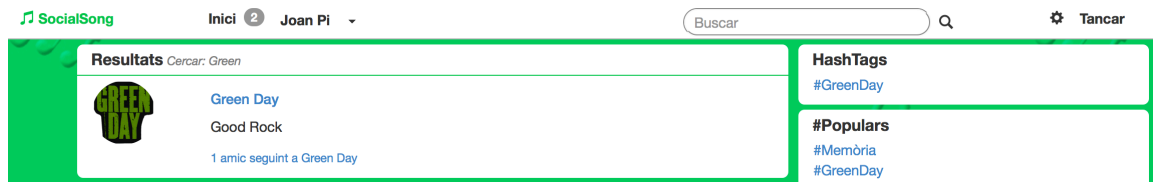


Figura 72, cercar (II)

10.3.10 SEGUIR A UN GRUP

Per seguir a un grup o podem fer de varies formes però la més tradicional es anar al seu perfil i clicar sobre el cor (negre) (figura 73), una vegada clicat es posa de color vermell, senyal que estem seguint al grup, una vegada que seguim el grup podrem fer publicacions i comentaris (figura 74).

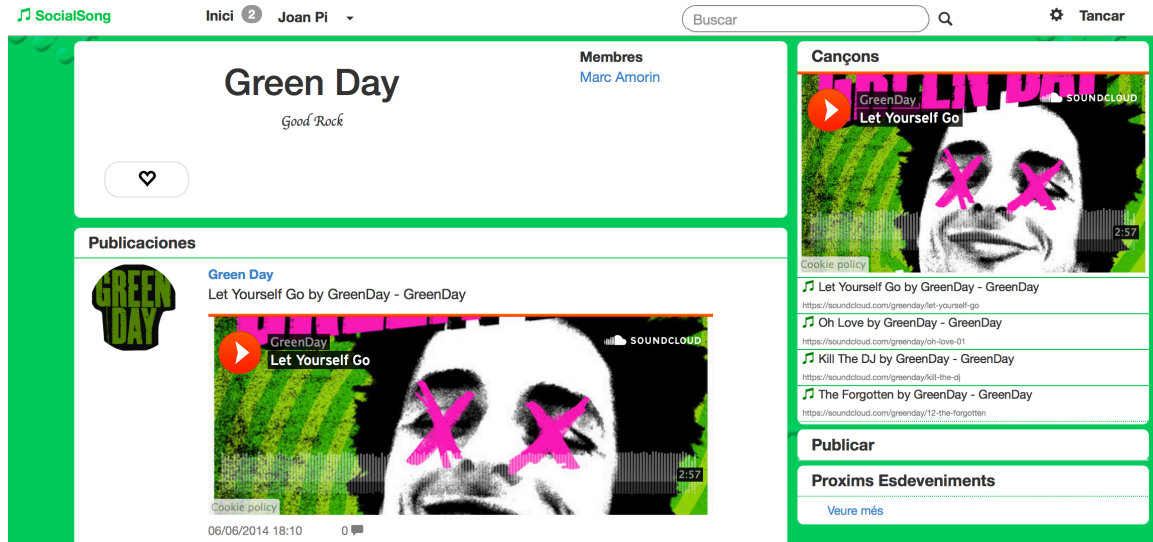


Figura 73, seguir grup (I)

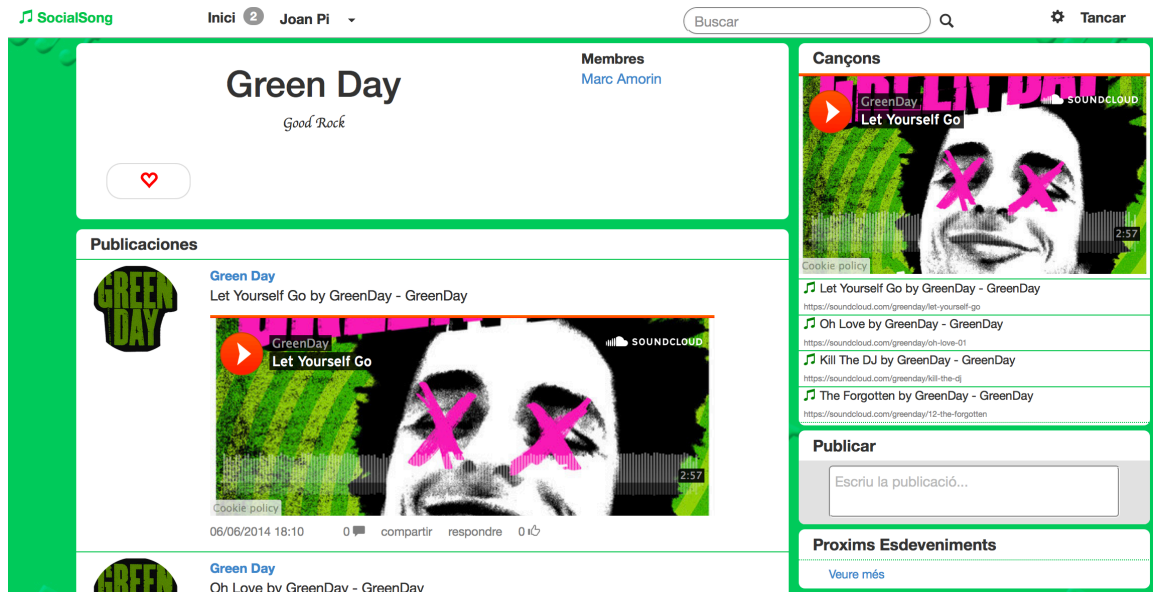


Figura 74 seguir grup (II)

10.3.11 PUBLICACIONS RELACIONADES AMB UN HASHTAG

Per veure les 50 últimes publicacions relacionades amb un hashtag només s'ha de clicar sobre de algun hashtag (figura 75).

The screenshot displays a social media interface for the hashtag #Memòria. At the top, the hashtag is shown with a close button. Below, four posts are visible:

- Post 1:** A video by 'Els Fantasmès' (profile picture: a ghost) titled 'Fantasmès A Sinister Garden'. The video thumbnail shows a person riding a green creature. The video player shows 0:00 / 4:39. Below the video are interaction options: 'Fa 7 minuts', '0' comments, 'compartir', 'respondre', and '0' likes.
- Post 2:** A group created by 'Els Fantasmès' (profile picture: a ghost) titled 'Grup creat per la #Memòria'. Below are interaction options: 'Fa 27 minuts', '0' comments, 'compartir', 'respondre', and '0' likes.
- Post 3:** A post by 'Joan Pi' (profile picture: a blue ocean) titled 'Publicació per la #Memòria'. Below are interaction options: 'Fa 39 minuts', '0' comments, 'compartir', 'respondre', and '0' likes.
- Post 4:** A post by 'Joan Pi' (profile picture: a blue ocean) mentioning 'Marc Amorin' (profile picture: a blue ocean) with the text 'Bones Marc #Memòria'.

Figura 75, publicacions relacionades amb un hashtag

11. CONCLUSIONS

En aquest apartat es parlarà dels objectius assolits en el projecte i de l'aprenentatge un cop acabat totes les fases del desenvolupament del projecte.

11.1 OBJECTIUS

S'han complert tot els requisits, apartat 6, que en el mes de març es van estipular. SocialSong té totes, o pràcticament totes, les funcionalitats d'una xarxa social, incorporant la temàtica de la música.

11.2 APRENTATGE

Durant el desenvolupament del projecte, m'he anat familiaritzant i aprenent cada vegada més sobre àmbits diversos, com són:

- Anàlisi i disseny d'aplicacions: les característiques de quest projecte han exigint una prèvia planificació i desenvolupament, cosa que m'ha ensenyat a idear requeriments, posar i complir terminis, etc.
- Java Enterprise Edition i Hibernate: he aprofundit molt en el llenguatge Java concretament en aquestes dues tecnologies.
- API REST: l'ús i la creació d'un servei web, mitjanament gran, m'ha servit per aprendre a estructurar bé les dades.
- Backbone i requireJS: estructurar l'aplicació web amb aquestes dos eines ha sigut relativament fàcil, he après a estructurar una aplicació, cosa que amb el JavaScript fa uns anys era inimaginable, i al correcte ús d'aquest llenguatge, poc usat en la meva curta trajectòria professional i d'estudiant.
- JQuery, Bootstrap, HTML i CSS: He aprofundit molt en els llenguatges HTML i CSS i en l'ús de les dos primeres eines que han fet més fàcil la programació en JavaScript i en HTML.
- APIs YouTube i SoundCloud: fer servir serveis web externs a suposat un gran aprenentatge a l'hora de estudiar totes les possibilitats que ofereixen i a ferles servir.

12.TREBALL FUTUR

Si es volgués treure a la llum SocialSong es necessitaria encara més funcionalitat i més interactivitat amb tecnologies relacionades amb la música i tot i així es necessitaria màrqueting i una font de d'ingressos com pot ser la publicitat en l'aplicació web.

L'aplicació tindria que tenir com a principals millores bàsiques:

- Interfície més polida: Fer una interfície millorada de cara a l'usuari amb millor rendiments dels espais, polir colors, etc. en definitiva més atractiva per l'usuari que està acostumat a utilitzar xarxes socials.
- Més funcionalitats per els usuaris: Inclourà millores en futures versions com poden ser missatges privats entre usuaris i grups, eliminar publicacions, alies per els usuaris, per facilitar la interactivitat i la cerca, etc.
- Augmentar la seguretat de l'usuari i de les seves dades: La seguretat de les dades és un tema molt sensible ara mateix per la societat, la pèrdua de dades, alteració de dades o difusió de dades es castiguen amb dures sancions econòmiques. La base de dades es tindria que registrar i tenir el nivell de seguretat que les dades necessiten.
- Implementar més APIs: La implementació de més APIs relacionades amb l'àmbit de la música és prioritari per atraure a més usuaris, la oferta de eines per escoltar música, comprar música fan que el públic no tingui una única plataforma per gaudir de la música. Es podrien implementar més APIs com: iTunes, Spotify, Groovevishark, etc.

13. BIBLIOGRAFIA

Oracle. (2014). *Java.net*. Recuperat 7 març 2014, des de: <https://java.net>

RedHat, Inc. (2014). *JBoss Application Server 7*. Recuperat 7 març 2014, des de: <http://jbossas.jboss.org>

Wikimedia Foundation. (2014). *Wikipedia*. Recuperat 7 març 2014, des de: <http://wikipedia.org>

StackExchange, Inc. (2014). *Stack Overflow*. Recuperat 7 març 2014, des de: <http://stackoverflow.com>

RedHat, Inc. (2014). *Hibernate*. Recuperat 10 març 2014, des de: <http://hibernate.org>

RedHat, Inc. (2014). *REStEasy*. Recuperat 10 març 2014, des de: <http://resteasy.jboss.org>

JQuery foundation. (2014). *JQuery*. Recuperat 10 març 2014, des de: <http://jquery.com>

DocumentCloud. (2014). *Backbonejs*. Recuperat 10 març 2014, des de: <http://backbonejs.org>

Thomas Davis. (2014). *Backbone Tutorials*. Recuperat 7 abril 2014, des de: <http://backbonetutorials.com>

Andy Chung. (2014). *RequireJS*. Recuperat 7 abril 2014, des de: <http://requirejs.org>

Twitter, Inc. (2014). *Bootstrap*. Recuperat 7 abril 2014, des de: <http://getbootstrap.com>

SoundCloud®. (2014). *SoundCloud for developers*. Recuperat 7 abril 2014, des de: <https://developers.soundcloud.com>

YouTube (2014). *YouTube for developers*. Recuperat 7 abril 2014, des de: <http://www.youtube.com/yt/dev/>

IndustriMusical. (2014). *Industria Musical*. Recuperat 5 maig 2014, des de: <http://industriamusical.es>

RedHat, Inc. (2014). *Openshift*. Recuperat 19 maig 2014, des de: <https://www.openshift.com>

Alexis Deveria. (2014). *Can I use*. Recuperat 1 juny 2014, des de: <http://caniuse.com>

Twitter, Inc. (2014). *REST API v1.1 Resources*. Recuperat 21 maig 2014, des de: <https://dev.twitter.com/docs/api/1.1>

Martín, I. (2014). *The JavaScript Programming Language*. Apunts procedents de UdGMoodle.

Martín, I. (2014). *REST APIs*. Apunts procedents de UdGMoodle.

Martín, I. (2014). *Browser MVC, Backbone.js*. Apunts procedents de UdGMoodle.

Martín, I. (2013). *Java EE*. Apunts procedents de UdGMoodle.

Boada, I. Soler, J. (2013). Tema 4. *Protecció de dades de caràcter personal*. Apunts procedents de UdGMoodle.

14. ANNEXOS

14.1 CONDICIONS DEL SERVEI

14.1.1 POLÍTICA D'US

Tenim la nostre pròpia política d'us, si vostè es registre considerem que accepta la política d'us.

Dades

Vostè és responsable del seu ús dels serveis, per qualsevol contingut que publiqui els serveis, i de les conseqüències derivades. El contingut que vostè envii, publiqui o exhibició podrà ser vist per altres usuaris dels Serveis, a través de serveis de tercers i llocs web. Només ha d'oferir un contingut que vol compartir amb els altres en virtut d'aquests termes.

Formularis d'inscripció

Formulari de registre del nostre lloc requereix que els usuaris ens donen informació de contacte (com el seu nom i cognom, correu electrònic).

Adreça de correu electrònic del client s'utilitza per enviar un missatge de correu electrònic de verificació, per posar-se en contacte amb el client, quan sigui necessari els recordatoris de contrasenyes a petició, recordatoris d'inactivitat, recordatoris de venciment de subscripció, confirmacions de comandes, informació sobre els canvis en els termes de utilitzar i informació molt rarament sobre les noves característiques per als usuaris amb comptes eliminades.

Informació personal

SocialSong considera que es dona el consentiment per mostrar les dades facilitades per l'usuari, tanmateix és mostraran conforma a la elecció de l'usuari.

Pujat Informació Pública

Si fas publicacions, puges cançons o vídeos, SocialSong considera que vostè dona el seu consentiment a mostrar la informació, tanmateix SocialSong no es fa càrrec dels danys que pot ocasionar a tercers, però si es fa càrrec de eliminar la informació de avís de perjudicis.

Si puges fotos de tu mateix, SocialSong tindrà en compte la seva càrrega com el seu consentiment exprés per a la utilització de la seva imatge. Si les imatges inclouran fotos

dels seus amics o altres tercers, assegureu-vos que aquestes persones han donat el seu consentiment a la publicació de la imatge abans de pujar-la.

Responsabilitats de SocialSong

SocialSong no es fa responsable del mal ús del sistema per part dels usuaris, tot i que si té constància del mal ús per part d'un usuari, té la opció, de adoptar les mesures corresponents.

Nosaltres no avalem, donem suport, representem ni garantim la veracitat, exactitud o fiabilitat de cap contingut o comunicació publicada a través dels serveis, ni recolza cap opinió expressada a través dels serveis. Vostè entén que en utilitzar els serveis, pot estar exposat a contingut que pot ser ofensiu, perjudicial, imprecís o inadequat, o en alguns casos, entrades mal etiquetades o enganyoses per qualsevol raó. Sota cap circumstància SocialSong serà responsable del contingut, incloent, però no limitat a, qualsevol error o omissió en qualsevol contingut, o qualsevol pèrdua o dany de qualsevol tipus incorreguts com a resultat de l'ús de qualsevol contingut publicat.

Els seus Drets

Vostè conserva els drets de qualsevol contingut que vostè envii, publiqui o mostri a través dels serveis. En enviar, publicar o mostrar contingut a través dels Serveis, vostè ens concedeix una llicència mundial, per utilitzar, copiar, reproduir, processar, adaptar, modificar, publicar, transmetre, mostrar i distribuir aquest contingut en qualsevol mitjà o mètode de distribució (actualment conegut o que desenvolupin posteriorment).

Vostè és responsable del seu ús dels serveis, per qualsevol contingut que vostè proporciona, i de les conseqüències dels mateixos, incloent l'ús del seu contingut per altres usuaris. Vostè entén que el seu contingut pot ser sindicat, transmès, distribuït o publicat, i pot estar subjecte a responsabilitat. SocialSong no serà responsable per qualsevol ús del seu contingut, d'acord amb aquests termes. Vostè manifesta i garanteix que té tots els drets, poders i autoritat necessaris per concedir els drets atorgats en aquest document a qualsevol contingut que vostè envii.

14.1.2 GALETES

Utilitzem galetes pròpies i de tercers per millorar els nostres. Si continua navegant o registre considerem que accepta el seu ús.

Què són les galetes?

Una galeta és un fitxer que es descarrega al seu ordinador en accedir a determinades pàgines web. Les galetes permeten a una pàgina web, entre altres coses, emmagatzemar i recuperar informació sobre els hàbits de navegació d'un usuari o del seu equip i, depenent de la informació que continguin i de la forma en què utilitzi el seu equip, es poden utilitzar per reconèixer l'usuari.

Quins tipus de galetes utilitza aquesta pàgina web?

- Galetes tècniques: Són aquelles que permeten a l'usuari la navegació a través d'una pàgina web, plataforma o aplicació i la utilització de les diferents opcions o serveis que en ella hi hagi com, per exemple, controlar el trànsit i la comunicació de dades, identificar la sessió, accedir a parts d'accés restringit, recordar els elements que integren una comanda, realitzar el procés de compra d'una comanda, realitzar la sol·licitud d'inscripció o participació en un esdeveniment, utilitzar elements de seguretat durant la navegació, emmagatzemar continguts per a la difusió de vídeos o so o compartir continguts a través de xarxes socials.

14.2 DOCUMENTACIÓ COMPLETA DE API REST

Autenticació

Consultes sobre l'autenticació de l'usuari actual.

Recurs	Paràmetres	Descripció
GET rest/auth/init Públic	Cap.	Per saber si hi ha un usuari amb la sessió iniciada. En cas de que hi hagi un usuari autenticat retorna "OK" i el seu ID, en cas oposat retorna un missatge de "ERROR".
POST rest/auth Públic	email (ob.) pwd (ob.)	Per iniciar secció amb un usuari. Retorna "OK" i el seu ID en cas que el tot hagi anat bé, en cas oposat retorna un missatge de "ERROR" i el motiu.
POST rest/auth/registre Públic	name (ob.) secondname (ob.) email (ob.) password (ob.) password2 (ob.)	Registre d'un usuari. En cas que el registre hagi sigut satisfactori retorna "OK" i el seu ID, en cas oposat retorna un missatge de "ERROR" amb el motiu de l'error.
POST rest/auth/logout Privat	Cap.	Tancar sessió. Retorna "OK" si s'ha fet satisfactòriament altrament retorna "ERROR".
Objecte JSON	Atributs	Descripció
Nom		
ModelJSON	state (String) message (String) currentId (Long)	Objecte bàsic que hereten la majoria de objecte que s'envien per l'API REST. <i>state</i> pot adquirir el valor "OK" o "ERROR", en cas de error <i>message</i> conté la raó de l'error. <i>currentId</i> és la id de l'usuari autenticat.

User

Informació bàsica d'un usuari.

Recurs	Paràmetres	Descripció
GET rest/user/my Privat	Cap.	Retorna la informació bàsica de l'usuari autenticat. En cas de fallada del servidor es retorna amb un missatge de ERROR.
POST rest/user/my/infoLite Privat	Cap.	Només retorna id de l'usuari, nom, cognom, entra altres. En cas que hagi anat malament es respon amb un missatge de ERROR.
Objecte JSON	Atributs	Descripció
Nom		
User	id (Long) is (String) name (String) secondname (String) email (String) notification (Long) description (String) relation (String) commonFriends (Long) img (Byte[]) nfollow (Long) nfollowers (Long) npost (Long) nFavGroup (Long) grups (Member[])	Hereta de l'objecte <i>ModelJSON</i> . <i>infoLite</i> només respon amb els camps <i>name</i> , <i>id</i> , <i>name</i> , <i>secondname</i> , <i>notifications</i> , <i>is</i> i <i>description</i> . L'atribut <i>is</i> pot prendre com a valor: USER, GRUP, NOTHING, POST, EVENT. <i>relation</i> pot prendre com a valor: OWN, FRIEND, NO_FRIEND, PENDING. <i>commonFriends</i> és el numero de usuaris seguits per l'usuari que es demana la informació i l'usuari actual. <i>nfollow</i> , <i>nfollowers</i> , <i>npost</i> , <i>nFavGroups</i> és el numero de usuaris seguits, seguidors, numero de publicacions i numero de grups seguits, respectivament. <i>grups</i> retorna una llista de l'objecte Member (es descriu més endavant).

Informació de l'usuari

Informació completa d'un usuari.

Recurs	Paràmetres	Descripció
GET rest/infouser/user/:id Privat	Id (ob.)	Retorna la informació completa de l'usuari amb id = :id.
GET rest/infouser/my Privat	Cap.	Retorna la informació completa de l'usuari autenticat.
PUT rest/infouser/my Privat	name (op.) secondname (op.) birthday (op.) birthdayPrivacy (op.) sex (op.) sexPrivacy (op.) city (op.) cityPrivacy (op.) country (op.) countryPrivacy (op.) photo (op.) description (op.)	Modificació de la informació d'un usuari, en cas que s'hagi produït correctament es retorna un objecte amb la informació de l'usuari amb la informació de l'usuari, en cas contrari es notifica amb error, i l'error produït. En cas de enviar cadenes buides la informació no s'actualitzarà com atribut buit.
Objecte JSON	Atributs	Descripció
Nom		
InfoUser	id (Long) sex (String) birthday (String) country (String) city (String) phone (String) user (User) sexPrivacy (String) birthdayPrivacy (String) countryPrivacy (String) cityPrivacy (String) phonePrivacy (String)	Hereta de l'objecte <i>ModelJSON</i> . Els atributs acabats amb "Privacy" notificant a l'usuari de quin caràcter són les dades, poden prendre el els valors: <i>PRIVATE</i> , <i>PUBLIC</i> o <i>FRIENDS</i> . Si un usuari no segueix a l'usuari el qual està demanant les dades no se li enviaren les dades de caràcter "FRIENDS" ni "PRIVATE". L'atribut <i>sex</i> té com a possibles valors: <i>NULL</i> , <i>MALE</i> o <i>FEMALE</i> . <i>user</i> és un objecte <i>User</i> (explicat anteriorment)

Notificacions

Informació sobre les notificacions d'un usuari.

Recurs	Paràmetres	Descripció
GET rest/notifications/my/:page Privat	page (op.)	Retorna una llista de 8 notificacions de l'usuari autenticat. En cas que <i>page</i> sigui null o 0, retornarà les 8 últimes en cas contrari retornarà les 8 que estarien a la pàgina desitjada. Expl: si <i>page</i> fos 3 enviaria la llista de notificacions des de la 16 a la 23 (entenen com l'última notificació la 0 en la llista).
PUT rest/notifications/:id Privat	id (ob.) option (op.) isNew (op.)	Retorna "OK" si ha anat bé el procés de actualització de la notificació, en cas contrari retorna un missatge de error.
Objecte JSON	Atributs	Descripció
Nom		
Notification	Id (Long) text (String) isNew (Booleà) status (String) date (String) typeNotifications (String) userTO (User) groupTO (Grup) userFrom (User) groupFrom (Grup)	Hereta de l'objecte <i>ModelJSON</i> . <i>text</i> conté la informació de la notificació. <i>isNew</i> és verdader si l'usuari mai a vist la notificació. <i>userTo</i> , <i>groupTO</i> : Només un dels dos no serà null, és per saber a qui va dirigit, a l'usuari o algun grup del que pertany. Si la notificació prové d'un grup l'atribut <i>groupFrom</i> vindrà la informació del grup, en cas que sigui d'un altre usuari <i>userFrom</i> tindrà la informació de l'usuari.

typeNotifications és el tipus de notificació, pot ser: FOLLOW, FOLLOW_GRUP, FRIEND_POST, GRUP, GRUP_POST, GRUP_MEMBER o SYSTEM_EDIT_PROFILE

Una notificació pot ser que comporti una resposta de l'usuari, per saber si necessita resposta es posa l'atribut *state* com a "PENDING", en cas que es respongui afirmativament la *state* s'actualitzarà a "OK", en cas negatiu *state* s'actualitzarà com "NO".

Seguidors

Informació sobre amistats dels usuaris.

Recurs	Paràmetres	Descripció
GET rest/friend/:id Privat	id (ob.)	Per saber si l'usuari autenticat segueix a l'usuari amb id = :id. Retorna un objecte <i>Friend</i> en cas que no hi hagi cap error en la crida, en cas contrari retorna un missatge de error.
POST rest/friend/:id Privat	id (ob.)	Per fer que l'usuari autenticat segueixi a l'usuari amb id = :id, en cas no hi hagi cap error en el procediment retornarà un objecte <i>Friend</i> , en cas contrari retorna un missatge de error.
PUT rest/friend/:id Privat	id (ob.) follow (ob.)	Per actualitzar la relació entre l'usuari autenticat i l'usuari amb id = :id, en cas no hi hagi cap error en el procediment retorna un objecte <i>Friend</i> , en cas contrari retorna un missatge de error.
GET rest/friend/followers/:id Privat	id (ob.)	Retorna una llista de usuaris (<i>User[]</i>) que són seguidors de l'usuari amb id = :id.

[GET rest/friend/follows/:id](#)
Privat

id (ob.)

Retorna una llista de usuaris (*User[]*) que segueixen al usuari amb id = :id.

Objecte JSON

Nom

Atributs

Descripció

Friend

user (USER)
relation (String)
userFriend (USER)

Hereta de l'objecte *ModelJSON*.

relation és la relació que té el *user*, amb el *userFriend*, pot tenir aquests valors: *OWN*, *FRIEND*, *NO_FRIEND* o *PENDING*.

Grup

Informació sobre els grups de música.

Recurs	Paràmetres	Descripció
GET rest/group/:id Privat	id (ob.)	Retorna la informació del grup amb id = :id.
POST rest/group Privat	name (ob.) description (op.) founded (op.) photo (op.) membres (op.) styleMusica (op.)	Per la creació d'un grup, només és obligatori el nom del grup. Retorna un objecte <i>Grup</i> si a sigut correcte la creació del grup, en cas contrari retorna un missatge de error.
PUT rest/group/:id Privat	id (ob.) name (op.) description (op.) founded (op.) photo (op.) membres (op.) styleMusica (op.)	Per la modificació d'un grup amb id = :id. Retorna un objecte <i>Grup</i> si a sigut correcte la modificació del grup, en cas contrari retorna un missatge de error.
POST rest/group/member/:id Privat	id (ob.) userId (ob.) type (op.) rol (op.)	Invitació a l'usuari amb id = <i>userId</i> al grup amb id = :id. Retorna un objecte <i>Grup</i> si la invitació s'ha enviat correctament, en cas contrari retorna un missatge de error.
PUT rest/group/member/:id Privat	id (ob.) memberId (ob.) type (op.) rol (op.)	Modificació de algun dels aspectes d'un membre amb identificació id = <i>memberId</i> del grup amb id = :id. Retorna un objecte <i>Grup</i> si la modificació a sigut correcte, en cas contrari retorna un missatge de error.
DELETE rest/group/member/:id/:memberId Privat	id (ob.) memberId (ob.)	Per eliminar el membre amb id = : <i>memberId</i> del grup amb id = :id. Retorna un objecte <i>Grup</i> si la eliminació ha sigut correcte, en cas contrari es notifica l'error.
GET rest/group/favourites/:id Privat	id (ob.)	Retorna una llista de grups favorits de l'usuari amb id = :id.

PUT rest/group/follow/:id
Privat

id (ob.)

Per que l'usuari autenticat segueixi (o deixi de seguir) al grup amb id = :id. Retorna la informació del grup si ha sigut correcte la operació, en cas contrari notifica l'error.

Objecte JSON

Nom

Atributs

Descripció

Group

Id (Long)
name (String)
founded (String)
styleMusic (String)
description (String)
typemember (String)
follow (Booleà)
members (Member[])
is (String)
img (Byte[])
events (Event[])
soundcloud (SCTrack[])
followers (Long)

Hereta de l'objecte *ModelJSON*.

typemember és la relació de "membre" que té el grup i l'usuari que està autenticat, pot assolir els valors: *FOUNDER*, *ADMIN*, *BASIC* o *NOT_MEMBER*.

L'atribut *follow* és verdader si l'usuari autenticat segueix al grup, en cas contrari false.

events és la llista de esdeveniments del grup.

soundcloud és la llista de cançons del grup.

followers és el numero de seguidors que té el grup.

L'atribut *is* pot prendre com a valor: *USER*, *GRUP*, *NOTHING*, *POST* o *EVENT*.

Esdeveniments

Informació sobre els esdeveniments d'un grup.

Recurs	Paràmetres	Descripció
POST rest/event Privat	groupId (ob.) place (ob.) description (ob.) date (ob.)	Creació d'un esdeveniment per el grup amb identificació id = <i>groupId</i> . Retorna la informació de l'esdeveniment creat, en cas contrari retorna un missatge de error.
Objecte JSON	Atributs	Descripció
Nom		
Event	description (String) date (String) place (String) groupName (String) groupId (Long) is (String)	Hereta de l'objecte <i>ModelJSON</i> . L'atribut <i>is</i> pot prendre com a valor: <i>USER</i> , <i>GRUP</i> , <i>NOTHING</i> , <i>POST</i> o <i>EVENT</i> . <i>description</i> és la descripció de l'esdeveniment. <i>date</i> és la data de l'esdeveniment, en cas que falti menys de 24h, es mostrarà com una "conta enrere".

SoundCloud

Cançons SoundCloud.

Recurs	Paràmetres	Descripció
POST rest/soundcloud Privat	groupId (ob.) name (ob.) iframe (ob.) url (ob.)	Per posar una cançó en el grup amb id = <i>groupId</i> . Retorna una llista de les 5 últimes cançons que el grup a posat.
GET rest/soundcloud/top4 Privat	Cap.	Retorna una llista de les 4 cançons més populars de les últimes 24 hores.
Objecte JSON	Atributs	Descripció
Nom		
SCTrack	id (Long) name (String) iframe (String) url (String)	L'atribut <i>name</i> és el nom de la cançó. L'atribut <i>iframe</i> és el codi HTML per obtenir la cançó. L'atribut <i>url</i> és la adreça original a SoundCloud.

Publicació

Informació sobre les publicacions de usuaris i grups.

Recurs	Paràmetres	Descripció
POST rest/post Privat	text (op.) userTold (op.) groupTold (op.) userFromId (op.) groupFromId (op.) postShareId (op.) scTrackId (op.) url (op.)	Creació d'una publicació. En cas que la creació es completi satisfactòriament retorna la publicació, en cas contrari retorna una missatge amb l'error. Tots els atributs són opcionals, però no es pot fer una publicació sense <i>text</i> o <i>scTrackId</i> , una de les dos no pot ser <i>null</i> . Tampoc es pot fer una publicació sense cap creador (<i>groupFrom</i> o <i>userFrom</i>).
DELETE rest/post/my/:id Privat	id (ob.)	Borra la publicació, de l'usuari autenticat, amb id = <i>:id</i> . Retorna un missatge de "OK" si s'ha fet satisfactòriament, en cas contrari s'envia una notificació amb error.
GET rest/post/my Privat	Cap.	Retorna una llista de totes les publicacions de l'usuari autenticat. Les publicacions poden ser fetes o dirigides a l'usuari autenticat.
GET rest/post/user/:id/:lastId Privat	id (ob.) lastId (op.)	Retorna, en el cas que <i>:lastId</i> fos <i>null</i> o <i>0</i> , una llista de les 15 últimes publicacions de l'usuari amb id = <i>:id</i> , en cas que <i>:lastId</i> no fos <i>null</i> es retornen les 15 publicacions amb id inferior a <i>:lastId</i> . Les publicacions poden ser fetes o dirigides a l'usuari amb id = <i>:id</i> .
GET rest/post/:id Privat	id (ob.)	Retorna la publicació amb id = <i>:id</i> .
GET rest/post/hashtag/:hashtag Privat	hashtag (ob.)	Retorna el llistat dels últimes 50 publicacions que tenen el hashtag <i>:hashtag</i> .

<p>GET rest/post/group/:id/:lastId Privat</p>	<p>id (ob.) iastId (op.)</p>	<p>Retorna, en el cas que <i>lastId</i> fos <i>null</i> o <i>0</i>, una llista de les 15 últimes publicacions del grup amb id = <i>:id</i>, en cas que <i>lastId</i> no fos <i>null</i> es retornen les 15 publicacions amb id inferior a <i>:lastId</i>.</p>
---	----------------------------------	---

Les publicacions poden ser fetes o dirigides al grup amb id = *:id*.

<p>GET rest/post/friends/:id/:lastId Privat</p>	<p>id (ob.) iastId (op.)</p>	<p>Retorna, en el cas que <i>:lastId</i> fos <i>null</i> o <i>0</i>, una llista de les 15 últimes publicacions dirigides o fetes, als usuaris que segueixi i/o als grups que segueixi l'usuari amb id = <i>:id</i> (també poden ser fetes o dirigides a l'usuari), en cas que <i>:lastId</i> no fos <i>null</i> es retornen les 15 publicacions amb id inferior a <i>:lastId</i>.</p>
---	----------------------------------	---

<p>POST rest/post/share Privat</p>	<p>id (ob.)</p>	<p>L'usuari autenticat comparteix la publicació amb id = <i>id</i>, si el procés a sigut satisfactori retorna la publicació, en cas contrari retorna una missatge de error.</p>
--	-----------------	---

Objecte JSON

Nom	Atributs	Descripció
Post	id (Long) text (String) date (String) userTo (User) groupTo (Group) userFrom (User) groupFrom (Group) postShare (Post) permit (String) commentaryTOList (Comentary[]) nLike (Long) like (Booleà) is (String) scTrackTO (SCTrack)	Hereta de l'objecte <i>ModelJSON</i> . <i>id</i> de la publicació. <i>text</i> cos de la publicació. <i>userTo</i> i <i>groupTo</i> per a qui va dirigit la publicació, <i>groupFrom</i> i <i>userFrom</i> qui ha fet la publicació. <i>postShare</i> si és una publicació compartida. <i>commentaryToList</i> la llista de comentaris. <i>nLike</i> el número de gent que li agrada, <i>like</i> si li agrada la usuari autenticat. <i>scTrackTO</i> si té una cançó.

permit els permisos que té l'usuari autenticat sobre la publicació. *permit* pot tenir els següents valor: *R_*, *_W_*, *__S*, *RW_*, *R_S*, *_WS*, *RWS*. Si conté una "R" té permís l'usuari actual de lectura de la publicació, si tenen una "W" vol dir que pot comentar la publicació i finalment si té una "S" pot compartir la publicació.

Comentari

Informació sobre els comentaris.

Recurs	Paràmetres	Descripció
POST rest/commentary Privat	text (ob.) postId (ob.) groupId (op.) userId (op.)	Creació d'una comentari a la publicació amb id = <i>postId</i> . En cas que el comentari es faci correctament retorna la informació del comentari, en cas contrari retorna una missatge de error. <i>groupId</i> i <i>userId</i> són els creadors del comentari, només un dels dos pot no ser null.

Objecte JSON	Atributs	Descripció
Nom Commentary	id (Long) text (String) date (String) postId (Long) img (Byte[]) name (String) idCommentarist (Long) nLike (Long) like (Booleà)	Hereta de l'objecte <i>ModelJSON</i> . <i>id</i> del comentari. <i>text</i> cos del comentari. <i>date</i> data del comentari. <i>postId</i> id de la publicació que va dirigit la publicació. <i>img</i> és la imatge del creador del comentari. <i>name</i> és el nom de l'usuari o grup que ha fet el comentari. <i>idCommentarist</i> la id del creador. <i>nLike</i> número de usuaris que li agraden el comentari. <i>like</i> és verdader si l'usuari actual li agrada la publicació, en cas contrari és fals.

Like

Informació sobre els “M’agrada” dels comentaris i publicacions.

Recurs	Paràmetres	Descripció
POST rest/like Privat	postId (op.) comentaryId (op.)	Creació del vincle “m’agrada” de l’usuari autenticat amb la publicació amb id = <i>postId</i> o amb el comentari amb id = <i>comentarid</i> . Només un dels dos pot ser no <i>null</i> . Retorna un objecte <i>Post</i> .
PUT rest/like Privat	postId (op.) comentaryId (op.) like (ob.)	Modificació del vincle “m’agrada” de l’usuari autenticat amb la publicació amb id = <i>postId</i> o amb el comentari amb id = <i>comentarid</i> . Només un dels dos pot ser no <i>null</i> . Retorna un objecte <i>Post</i> .

HashTag

Informació sobre els “M’agrada” dels comentaris i publicacions.

Recurs	Paràmetres	Descripció
GET rest/hashtag/tendency Privat	Cap.	Retorna una llista amb el nom dels 10 hashtags més populars de les últimes 24h.

Cerca

Informació sobre cerques de usuaris, grups, hashtags.

Recurs	Paràmetres	Descripció
GET rest/search/:text Privat	text (ob.)	Retorna una llista de usuaris, grups, hashtags que tenen relació amb el text <i>:text</i> .
GET rest/search/friends/:text Privat	text (ob.)	Retorna una llista de usuaris seguits que el nom de aquests contingui la paraula <i>:text</i> .

Objecte JSON

Nom	Atributs	Descripció
SearchItem	list (Object[]) hashtag (String[]);	Hereta de l'objecte <i>ModelJSON</i> . <i>list</i> és la llista que conte els usuaris i els grups trobats. <i>hashtag</i> és la llista de hashtags trobats.

Objecte JSON

Nom	Atributs	Descripció
Member	id (Long) name (String) typeMember (String) status (String) rol (String)	Hereta de l'objecte <i>ModelJSON</i> . <i>id</i> de l'objecte <i>Member</i> . <i>name</i> nom de l'usuari. <i>typeMember</i> tipus de membre (<i>BASIC</i> , <i>FOUNDER</i> , <i>ADMIN</i>). <i>status</i> estatus de la sol·licitud de membre (<i>OK</i> , <i>NO</i> , <i>PENDING</i>). <i>rol</i> és el rol que fa l'usuari en el grup (Vocalista, guitarrista....).