

Tell-App Messaging

TFC – Aplicacions Mòbils

Memòria

Fco. Javier Sagra Sánchez

Coordinat per Joan Vicent Orença



ÍNDEX

1. Introducció a la missatgeria mòbil.....	4
1.1. WhatsApp.....	4
1.2. Line.....	5
1.3. Spotbros.....	5
2. Objectius	6
3. Funcionalitats principals	7
4. Planificació del projecte.....	8
5. Anàlisi Funcional.....	10
5.1. Actors.....	10
5.2. Model de casos d'ús.....	10
5.3. Fitxes de casos d'ús	11
6. Anàlisi Funcional – Prototipus de les Pantalles.....	17
6.1. Pantalla de Registre.....	17
6.2. Demanar nova contrasenya	18
6.3. Llista de converses.....	19
6.4. Contactes.....	20
6.4.1. Llistat de contactes.....	20
6.4.2. Afegir un nou contacte	21
6.5. Crear un nou grup	21
6.6. Pantalla de conversa.....	23
6.6.1. Editar la conversa.....	24
6.7. Configuració del compte d'usuari.....	25
7. Disseny Tècnic.....	27
7.1. Diagrama de components de l'arquitectura	27
7.1.1. Servidor.....	28
7.1.2. Dispositiu mòbil.....	29
7.1.3. Google Cloud Messaging.....	29
7.2. Interfície amb el servidor	29
7.3. Disseny relacional de la Base de Dades.....	32
7.3.1. Disseny de la Base de Dades del Servidor	32
7.3.1.1 Taula Users	33
7.3.1.2. Taula Chats	34
7.3.1.3. Taula Chats Users	34
7.3.1.4. Taula Messages.....	35
7.3.2. Disseny de la Base de Dades del dispositiu	36
7.4. Diagrama de classes.....	37
7.4.1 Capa de visualització	37
7.4.2. Capa d'accés a dades i al servidor	38
7.5. Diagrames de Seqüència	39
7.5.1 Registre del dispositiu.....	39
7.5.2. Recepció d'un missatge	40
8. Implementació.....	42
8.1. Google Cloud Messaging.....	42
8.1.1. Registre a Google Cloud Messaging.....	42
8.1.2. GCM al servidor	43

8.1.3. GCM al dispositiu	44
8.2. Localització	44
8.3. Llistes d'objectes. ListView i Adapters	45
8.4. Diàlegs modals	48
8.4.1. Diàlegs d'avís.....	48
8.4.2. Diàlegs de confirmació	49
8.4.3. Diàlegs de progrés.....	49
8.5. RESTful asíncron.....	50
9. Manual d'Usuari	51
9.1. Registre de l'usuari	51
9.1.1. Iniciar sessió.....	51
9.1.2. Registre d'un nou usuari	54
9.2. Llista de Converses	54
9.3. Gestió de Contactes.....	56
9.4. Creació de converses.....	59
9.4.1. Converses directes entre dos usuaris.....	59
9.4.2. Converses de grup	59
9.5. Conversa i Missatges	60
9.5.1. Edició de la Conversa	62
9.6. Ajusts de compte	63
10. Conclusions.....	65
10.1. Futures millores fora de l'àmbit del projecte	65
10.2. Fonts d'informació.....	65

1. Introducció a la missatgeria mòbil

La missatgeria mòbil va nèixer amb els dispositius del mateix nom, i va suposar una revolució a les comunicacions, al permetre enviar missatges SMS (*Short Message Service*) entre els terminals sense haver d'establir una trucada telefònica. Aquests missatges quedaven emmagatzemats als proveïdors de servei fins que el destinatari els rebia, i això feia que es poguessin enviar missatges amb la garantia de que si el destinatari no hi era disponible, tan bon punt ho fos rebria el missatge.

Evidentment, aquest servei de missatgeria mòbil tenia un cost, que afegit a l'àmplia adopció per part dels usuaris, suposava uns ingressos molt apreciats per les operadores de telefonia mòbil.

Però amb l'arribada dels *smartphones*, dispositius mòbils amb connexió a Internet mitjançant una tarifa de dades, les regles del joc van canviar. De seguida es va pensar en fer un servei de missatgeria aprofitant la connexió a Internet i les notificacions Push, que és el què ara estem vivint.

D'aquesta manera, un usuari pot enviar un missatge a un altre usuari mitjançant un servei d'aquest tipus. Però no l'envia directament a aquest, sinó a un servidor intermig que s'encarrega de notificar al destinatari que té missatges nous. Així doncs, el destinatari se'ls descarrega i es completa el cercle. Aquest procés elimina el cost de la missatgeria SMS, ja què al fer servir la tarifa de dades contractada, i al ser majoritàriament missatges de text, no suposa un cost ni un consum de dades addicional al client.

Veiem a continuació alguns exemples de serveis d'aquest tipus, que actualment competeixen entre si:

1.1. WhatsApp¹

Va ser dels primers, i és el més conegut. Actualment té una base d'usuaris de l'ordre de més de 200 milions (més que Twitter)², i darrerament ha estat notícia ja què ha anunciat que començarà a cobrar una quota anual pel servei, cosa lògica per un costat degut al cost de manteniment dels servidors. En el següent enllaç es troba [una entrevista que van fer recentment als seus creadors](#)³, defensant el seu model de negoci.



¹ Whats App (en línia) <http://www.whatsapp.com/?l=es> [data consulta: 01/06/2013]

² 20 Minutos (en línia) <http://www.20minutos.es/noticia/1788588/0/whatsapp/twitter/tecnologia/> [data consulta: 01/06/2013]

³ El País (en línia) http://tecnologia.elpais.com/tecnologia/2013/05/17/actualidad/1368788101_263082.html [data consulta: 01/06/2013]

1.2. Line⁴

Alternativa a WhatsApp, sense cap cost per a l'usuari, però que es monetitza amb una botiga a dins de l'aplicació, on l'usuari pot comprar *stickers* (emoticones en format d'imatges) per a insertar als missatges, jocs i d'altres.

El principal problema de Line en aquests moments és que té un consum bastant elevat de la bateria del dispositiu, fet que fa que molts dispositius baixin del dia d'autonomia.



1.3. Spotbros⁵

Alternativa feta a Barcelona que a més de les funcionalitats de missatgeria també funciona com a xarxa social de proximitat; podem trobar grups de conversa propers a la nostra localització.

També posa a disposició dels desenvolupadors un SDK amb el què es poden realitzar aplicacions fent servir la plataforma Spotbros.



⁴ Line (en línia) <http://line.naver.jp/es/> [data consulta: 01/06/2013]

⁵ Spotbros (en línia) <http://www.spotbros.com/> [data consulta: 01/06/2013]

2. Objectius

L'objectiu principal d'aquest document és descriure l'aplicació Android **TellApp Messaging**, tant tècnica com funcionalment, i publicar el seu manual d'ús. Es tracta d'un servei de missatgeria basat en la coneguda aplicació **WhatsApp**, que permet a dos o més usuaris comunicar-se entre si gràcies a la connexió a Internet dels *smartphones*.

S'ha escollit fer una aplicació d'aquest tipus ja que proporciona un bon exercici de síntesi del que s'ha après al llarg dels estudis d'Enginyeria Tècnica d'Informàtica de Sistemes, i també per tal de proporcionar a tot aquell que ho desitgi una alternativa a les grans aplicacions de missatgeria, ja que encara que es tracti d'una implementació bastant simple, es poden estendre a posteriori les seves funcionalitats per tal d'assolir una alternativa factible.

Per la realització d'aquest projecte s'han fet servir diferents tecnologies com Java per al desenvolupament Android, Java i Groovy (Grails) per a la part de servidor, un contenidor servlet com és Apache, una base de dades MySQL i un servidor Linux (Ubuntu).

Personalment, ha estat una primera presa de contacte amb Android, i m'ha resultat una experiència molt profitosa de cara no només a conèixer la tecnologia sinó també per a posar en pràctica tot el que he après al llarg dels estudis.

3. Funcionalitats principals

L'objectiu d'aquest apartat és descriure les funcionalitats bàsiques que ha de contemplar la nostra aplicació, establint així un marc bàsic d'objectius a assolir.

A grans trets, aquestes són les funcionalitats més importants que necessitem implementar a la nostra aplicació:

Pantalla d'Inici - Registre de l'usuari. El primer cop que s'arrenqui l'aplicació, aquesta ha de demanar a l'usuari que es registri als seus servidors, o que iniciï sessió amb el seu compte. En el cas que l'usuari no s'enrecordi de la seva contrasenya, podrà demanar canviar-la introduint la seva adreça de correu electrònic.

Interfície de converses actives: Aquesta serà la pantalla principal de l'aplicació en el cas que l'usuari ja s'hagi enregistrat, o la que veurà després de l'enregistrament, i mostrarà totes les converses actives a les què l'usuari prengui part. Podrà accedir a qualsevol d'aquestes prement a sobre d'elles. També, des d'aquesta pantalla, l'usuari podrà crear noves converses, sortir de les converses actives i accedir a les opcions generals de l'aplicació.

Crear noves converses. Accedint a la pantalla corresponent, l'usuari ha de poder crear noves converses amb un o varis contactes, podent també posar un títol a aquesta. Un cop creada, s'afegirà a la llista de converses actives i es notificarà a tots els usuaris involucrats.

Interfície de conversa: A la interfície de conversa es mostraran els missatges de la conversa seleccionada, i es podran afegir de nous. També podrem veure qui participa a la conversa, i canviar el títol d'aquesta.

Configuració de l'aplicació: Aquesta interfície mostrarà quina versió de l'aplicació estem fent servir, i també podrem establir configuracions bàsiques, com activar o desactivar els avisos sonors, entre d'altres.

4. Planificació del projecte

S'ha de realitzar la planificació del projecte tenint en compte les dates crítiques o fites de l'assignatura, basades en entregues, que són les següents, per tal d'assolir amb èxit la consecució del projecte.

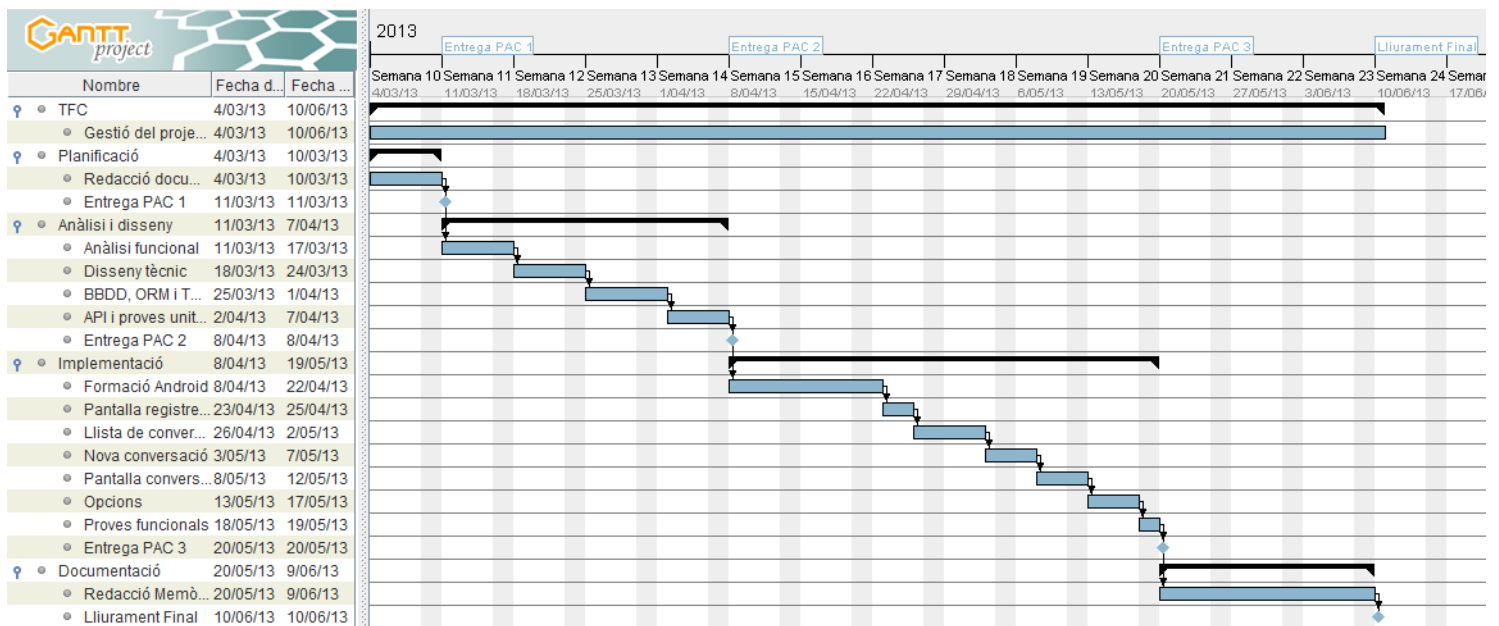
Data de lliurament	Fita	Lliurar
11 de març	PAC 1	
8 d'abril	PAC 2	
20 de maig	PAC 3	
10 de juny	Lliurament final	

A continuació es desglossa la planificació temporal del projecte, tenint en compte les fites establertes al Pla Docent de l'assignatura, que es divideixen en l'entrega de tres PACs i una entrega final:

Tasca	Inici	Final
PAC 1 - Planificació		
Redacció document planificació	4/03/13	10/03/13
Entrega PAC 1	11/03/13	11/03/13
PAC 2 - Anàlisi i disseny		
Anàlisi funcional	11/03/13	17/03/13
Disseny tècnic	18/03/13	24/03/13
BBDD, ORM i TDD	25/03/13	1/04/13
API i proves unitàries	2/04/13	7/04/13
Entrega PAC 2	8/04/13	8/04/13

PAC 3 - Implementació		
Formació Android	8/04/13	22/04/13
Pantalla registre dispositiu	23/04/13	25/04/13
Llista de conversacions	26/04/13	2/05/13
Nova conversació	3/05/13	7/05/13
Pantalla conversació	8/05/13	12/05/13
Opcions	13/05/13	17/05/13
Proves funcionals	18/05/13	19/05/13
Entrega PAC 3	20/05/13	20/05/13
Entrega Final		
Redacció Memòria i presentació virtual	20/05/13	9/06/13
Lliurament Final	10/06/13	10/06/13

Afegim també el diagrama de Gantt corresponent, amb l'evolució temporal del projecte i totes les seves fases:



5. Anàlisi Funcional

El primer pas a l'hora de realitzar l'anàlisi funcional d'una nova aplicació és estudiar els requeriments per tal de definir els casos d'ús d'aquesta. Amb els casos d'ús es pretén indicar el funcionament bàsic de l'aplicació, des del punt de vista de les possibles accions que es poden realitzar amb aquesta, així com els actors implicats en tots els escenaris possibles d'ús.

A continuació es mostren els actors que prenen part, per després veure l'esquema del model de casos d'ús seguit de les fitxes per cadascú d'aquests.

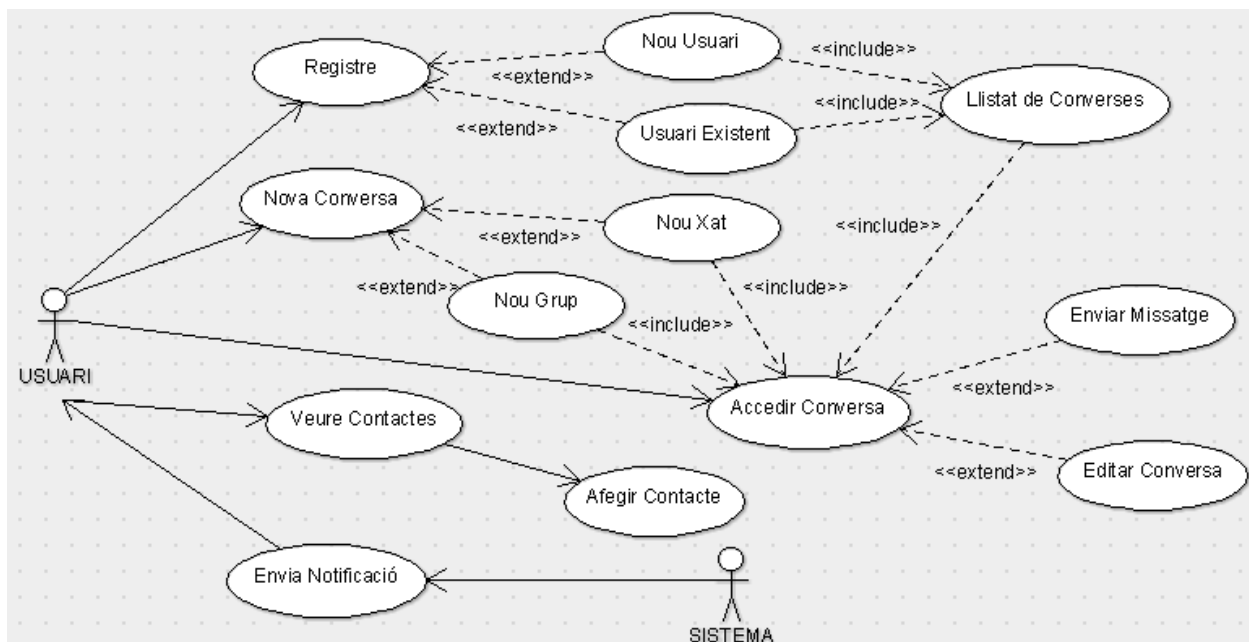
5.1. Actors

A l'aplicació que ens ocupa trobem dos actors principals, que són els següents:

- **Usuari.** Representa al dispositiu (mòbil) client de l'aplicació, i per tant a l'usuari que la fa servir.
- **Sistema.** Es tracta de la part servidora, amb la què l'usuari interactuarà per tal d'obtenir noves dades i també enviar-ne de noves.

5.2. Model de casos d'ús

Seguidament es mostra el diagrama del model de casos d'ús



Aquest esquema mostra les accions bàsiques que poden iniciar tant l'usuari com el sistema, que a grans trets són les que es descriuen a continuació.

Per **part del sistema**, aquest només pot iniciar un enviament de missatge cap a l'usuari (o dispositiu), que es tracta d'una notificació de tipus *Push* per indicar que hi han noves dades disponibles per a aquest, amb l'objectiu que aquest faci un *request* d'aquestes dades.

Per **part de l'usuari** (o dispositiu), aquest pot iniciar les següents accions:

- **Registrar el dispositiu**, al instal·lar per primer cop l'aplicació; mitjançant això es vincularà el dispositiu amb Google Cloud i s'iniciarà sessió per tal de fer servir aquesta. L'usuari pot iniciar sessió, si ja s'ha creat un compte abans, o enregistrar-se al servei, creant un compte nou. L'identificador del compte serà el correu electrònic. En el cas que iniciï sessió, haurà d'introduir l'email (ID del seu compte) i la seva contrasenya. Si es registra com a nou usuari, haurà d'introduir, a més, un nom per a mostrar a la resta d'usuaris. A més, si no se'n recorda de la seva contrasenya, ha de poder tornar-la a demanar.

- **Veure la llista de converses actives**, des d'on podrà accedir a qualsevol conversa. Un cop enregistat l'usuari/dispositiu, accedirà a aquesta pantalla, i passarà a ser la pantalla principal de l'aplicació. Ha de mostrar les converses actives, i fent click sobre qualsevol conversa accedirem a la pantalla de la conversa. També podem, des d'aquesta pantalla, crear noves converses, tant si són amb un sol contacte o amb varis (el què s'anomena **grup**), veure i editar contactes i accedir a la configuració de l'aplicació.

- **Crear una nova conversa**, que pot ser amb un usuari (*conversa*) o amb un grup d'usuaris (*grup*). En cas de ser amb un usuari, seleccionarem aquest de la llista de contactes de l'aplicació; en cas de ser un grup, anirem seleccionant els usuaris als quals volem convidar, i donarem un nom al grup. En qualsevol dels dos casos, al crear la conversa s'obrirà aquesta

- **Pantalla Conversa**: Un cop dintre d'una conversa, ja sigui accedint des de la llista de converses o al crear-ne una de nova, l'usuari podrà editar la informació de la conversa (si es tracta d'un grup), escriure un missatge, afegir més contactes a la conversa (si és un grup i l'ha creat ell) i abandonar la conversa.

- **Contactes**: Des de la pantalla de converses actives es podrà accedir a la pantalla de contactes. Aquí es mostraran els contactes de l'usuari, i podrà afegir d'altres al servidor cercant-los pel seu correu electrònic.

- **Canviar la seva configuració d'usuari**. Des de la vista de converses actives, l'usuari podrà canviar els següents valors de la configuració del seu compte d'usuari: nom a mostrar, contrasenya al servidor.

5.3. Fitxes de casos d'ús

Ara que coneixem els casos d'ús de l'aplicació, es descriuran els més complexos o que necessiten més explicació mitjançant les seves fitxes de casos d'ús.

Cas d'Us	Usuari inicia sessió a l'aplicació
Descripció	L'usuari obre per primer cop l'aplicació. L'usuari ja s'ha enregistat anteriorment a l'aplicació. Se li demana la seva adreça d'email i la seva contrasenya. Si el compte és incorrecte, se li avisarà.
Precondicions	Cap

Postcondicions	L'usuari inicia sessió en l'aplicació.
Flux principal	Primera arrencada de l'aplicació. L'usuari obté el registre del seu compte i dispositiu.
Dades d'entrada	Contrasenya i compte de correu
Dades de sortida	L'usuari recupera el seu nom (emmagatzemat al servidor) i accedeix a la pantalla de llista de converses.

Cas d'Us	Usuari es registra a l'aplicació
Descripció	L'usuari obre per primer cop l'aplicació i s'ha d'enregistrar en el servei. Haurà d'introduir la seva adreça de correu, una contrasenya i el seu nom per a mostrar. S'enregistrarà al servidor i s'obrirà l'activity amb la llista de missatges actius.
Precondicions	Cap
Postcondicions	L'usuari inicia sessió en l'aplicació.
Flux principal	Primera arrencada de l'aplicació. L'usuari obté el registre del seu compte i dispositiu.
Dades d'entrada	Nom d'usuari, contrasenya i compte de correu
Dades de sortida	Un cop enregistrat l'usuari, es passa a la pantalla de llista de converses. Si l'email ja s'ha enregistrat anteriorment, es mostrarà un error indicant-ho.

Cas d'Us	Afegir un contacte
Descripció	L'usuari selecciona l'opció <i>Contactes</i> de la pantalla de Llista de Missatges. Apareixerà l'interfície dels contactes, que contindrà dos pestanyes: una amb la llista de contactes de l'usuari, i l'altra amb un formulari per tal d'afegir un nou contacte.

	<p>La llista de contactes mostrarà tots els contactes de l'usuari, i una icona a cadascú d'ells per tal de crear una conversa amb l'usuari o obrir-la si ja existeix. També podrà eliminar el contacte prement a sobre del contacte i seleccionant <i>Eliminar</i>.</p> <p>El formulari per tal d'afegir un nou contacte constarà d'un camp per a posar el email del nou contacte i un botó de confirmació. Si el contacte existeix al servidor, s'afegirà a la llista de contactes i s'informarà. En el cas que no existeixi, s'informarà de què el contacte no existeix.</p>
Precondicions	L'usuari està registrat al servidor.
Postcondicions	L'usuari pot haver afegit / esborrat contactes.
Flux principal	S'accedeix des de la llista de contactes.
Dades d'entrada	Cap / Email del contacte
Dades de sortida	Cap / Contacte afegit / Contacte esborrat.

Cas d'Us	Afegir una conversa amb un Usuari
Descripció	<p>Des de la pantalla de converses actives, i mitjançant una icona, s'obre la pantalla per crear una nova conversa amb un usuari. Aquesta pantalla mostra una llista amb tots els contactes, i al fer click a sobre de qualsevol d'ells, es genera una conversa amb l'usuari seleccionat, si aquesta no existia prèviament, o s'obre la conversa amb l'usuari seleccionat.</p> <p>Es pot fer el mateix a la llista de converses, fent click a la icona amb el símbol "més", que portarà a la mateixa pantalla de contactes.</p>
Precondicions	Usuari està registrat
Postcondicions	S'obre la pantalla de conversa amb la conversa generada.
Flux principal	Des de la llista de converses actives
Dades d'entrada	Cap
Dades de sortida	Usuari seleccionat; s'obre conversa acvita amb ell.

Cas d'Us	Afegir una conversa amb un Grup d'Usuaris
Descripció	Des de la pantalla de converses actives podrem generar un nou grup. A la pantalla per crear un nou grup, li posarem un nom i seleccionarem quins contactes prendran part en ell. Per seleccionar els contactes, s'obrirà un nou Activity amb una llista de contactes que podrem seleccionar mitjançant un CheckBox. Un cop seleccionats els contactes, es generarà el grup i s'obrirà la pantalla de la conversa.
Precondicions	Usuari està registrat
Postcondicions	S'obre la pantalla de conversa de Grup.
Flux principal	Des de la llista de converses actives.
Dades d'entrada	Cap.
Dades de sortida	S'obre la conversa generada.

Cas d'Us	Pantalla de Conversa
Descripció	<p>A la pantalla de conversa es mostren tots els missatges que s'han enviat a la conversa actual, ordenats per data i hora de recepció (el darrer és l'últim. L'usuari pot escriure nous missatges i enviar-los a la conversa amb el botó d'enviament.</p> <p>S'ha de poder tornar enrera a la llista de converses mitjançant un botó destinat a tal finalitat.</p> <p>Si la conversa és un grup, s'ha de poder editar les dades del grup, amb una icona d'edició que portarà a l'usuari fins a la pantalla d'edició de la conversa.</p>
Precondicions	Usuari clica a la conversa a la llista de converses
Postcondicions	S'obre la pantalla de conversa. L'usuari pot enviar nous missatges.
Flux principal	Des de la llista de converses actives.
Dades d'entrada	Usuari tecleja nou missatge.
Dades de sortida	Missatge s'envia a la resta de participants i apareix a la conversa.

Cas d'Us	Pantalla d'edició de la conversa
Descripció	<p>Des de la pantalla de conversa accedim a la pantalla d'edició de la conversa, si aquesta és un grup. En aquesta pantalla apareix el nom de la conversa, que es pot editar, i la llista de contactes de l'usuari amb un checkbox al costat, per a seleccionar els contactes que es vol afegir a aquesta.</p> <p>En el cas dels contactes que ja en prenen part, el checkbox apareix marcat i deshabilitat, indicant que ja està afegit i que no es pot treure de la conversa.</p> <p>Finalment, per a emmagatzemar els canvis, es premerà la icona destinada a aquest fi, i s'informarà del resultat de l'operació al servidor.</p>
Precondicions	Usuari clica a la icona d'edició de la conversa i accedeix a aquesta pantalla.
Postcondicions	Usuari pot canviar o no el nom de la conversa i afegir nous usuaris.
Flux principal	Conversa.
Dades d'entrada	Usuari tecleja nou nom, afegeix nous usuaris.
Dades de sortida	Dades s'emmagatzemen al servidor i es reflexen a la conversa. Si servidor falla, es comunica a l'usuari.

Cas d'Us	Pantalla d'edició del compte d'usuari
Descripció	<p>Des d'aquesta pantalla l'usuari pot canviar les següents dades del seu compte d'usuari:</p> <ul style="list-style-type: none"> - el seu nom per a mostrar (nom amb el que apareixeran els seus missatges) - la seva contrasenya (amb confirmació) <p>L'usuari desa els canvis al servidor mitjançant un botó destinat a tal fi, situat a la cantonada superior dreta. Disposarà d'un altre botó per a tornar enrera, a la cantonada superior esquerra.</p>

	<p>L'usuari disposarà en aquesta pantalla d'un menú contextual, des d'on podrà:</p> <ul style="list-style-type: none"> - Provar el servei, demanant al servidor que li enviï una notificació de prova - Esborrar el seu compte del dispositiu, incloent les dades de l'aplicació: converses, missatges, contactes, registre i dades personals.
Precondicions	Cap
Postcondicions	Usuari edita les seves dades del compte i s'envien al servidor, que confirma els canvis o avisa d'errors.
Flux principal	Llista de converses
Dades d'entrada	Nou nom per a mostrar, nova contrasenya i confirmació de contrasenya.
Dades de sortida	Avís del canvi de les dades d'usuari, o avís d'error.

La resta de casos d'usuari no tenen més complexitat i no cal que fem fitxes de casos d'ús per a ells.

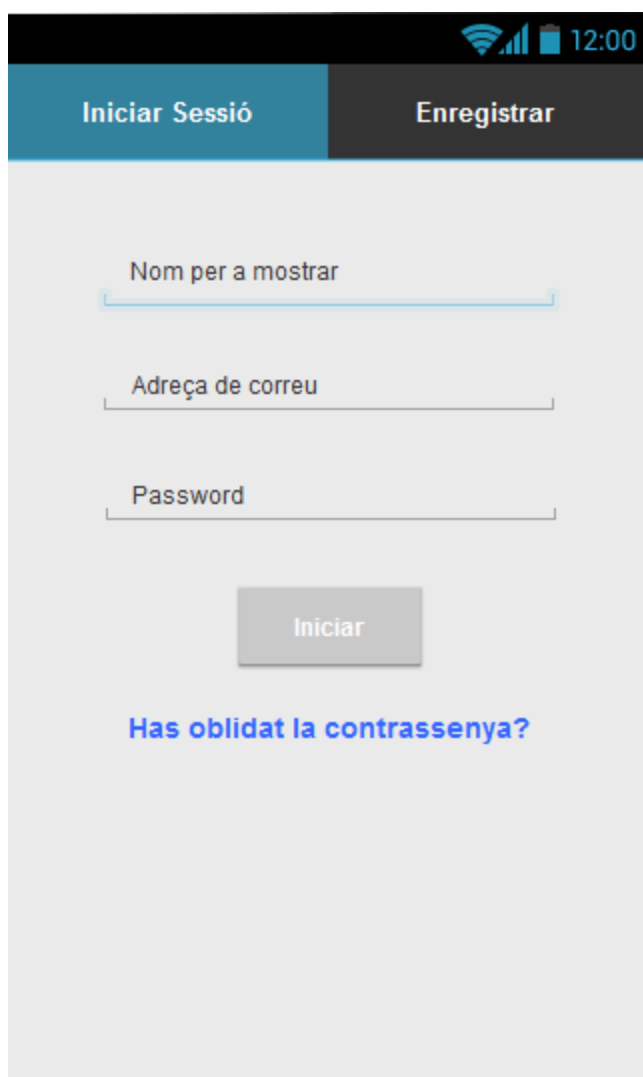
6. Anàlisi Funcional – Prototipus de les Pantalles

En aquest apartat es farà un disseny bàsic o *mock-up* de com han de ser les pantalles de l'aplicació: quina és la seva interfície, quins botons han d'incloure, què fa cada botó, etc.

Per a fer aquests mock-ups s'ha emprat l'eina Pencil (<http://pencil.evolus.vn/>), que ens permet dissenyar interfícies d'usuari tant per aplicacions d'escriptori, com per web i aplicacions mòbils (iOS i Android).

6.1. Pantalla de Registre

El següent mock-up mostra com hauria de ser la pantalla de registre:



El mock-up mostra una pantalla de registre amb un disseny modern i funcional. A la part superior, hi ha una barra de navegació amb dos botons: "Iniciar Sessió" (en blau) i "Enregistrar" (en gris). A sota, hi ha tres camps de text per a "Nom per a mostrar", "Adreça de correu" i "Password". A continuació, hi ha un botó "Iniciar" i un enllaç "Has oblidat la contrassenya?". A la part superior dreta, hi ha un indicador de bateria i l'hora "12:00".

Serà la primera pantalla que l'usuari veurà el primer cop que obri l'aplicació. Des d'aquí l'usuari es podrà enregistrar al servidor, o iniciar sessió en aquest amb el seu compte, si ja s'havia enregidrat prèviament.

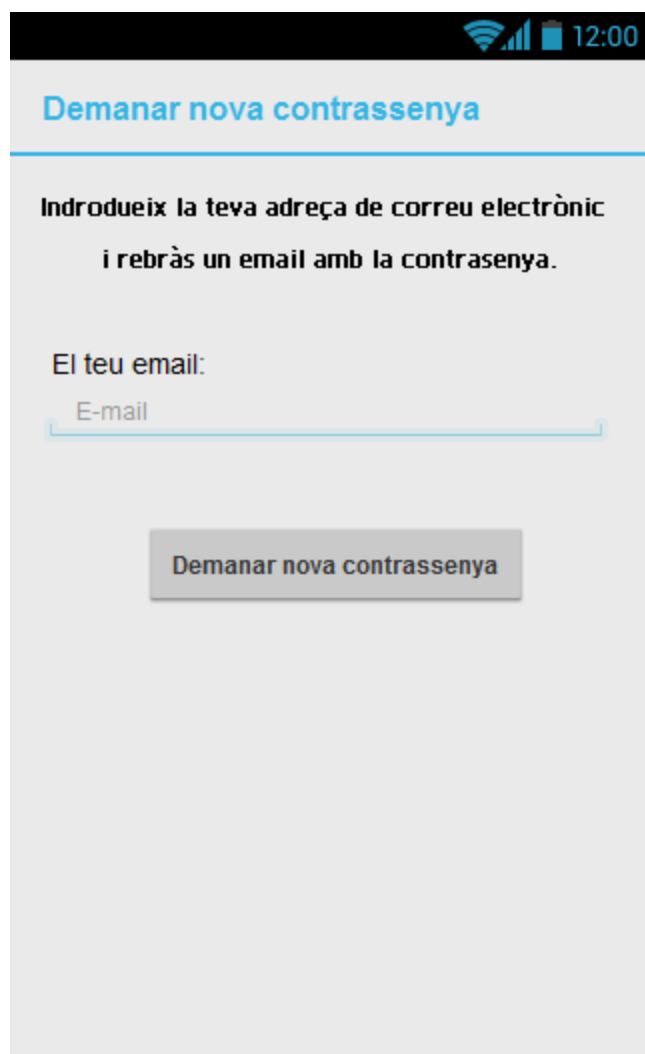
Hi hauràn dos *tabs* o pestanyes per diferenciar entre iniciar sessió i enregistrar un nou compte. El en cas del *tab* d'iniciar sessió, l'usuari haurà d'introduir una adreça de correu electrònic i la seva contrasenya, mentre que en el cas de l'enregistrament del compte serà necessari introduir, a més, un nom per a mostrar a la resta d'usuaris.

En el cas que un usuari no estigui enregidrat, s'enregistrarà; si ja s'havia enregidrat (es comprova per això el compte de correu), es mirarà al servidor que el password coincideixi per al email proporcionat.

Si tot ha estat correcte, s'obre la pantalla de llista de converses; si no, s'informa de l'error a l'usuari.

6.2. Demanar nova contrasenya

En el cas que l'usuari hagi oblidat la seva contrasenya, podrà demanar que el servidor li reenvii per tal d'accedir a l'aplicació prement a sobre del text "Has oblidat la contrasenya?" de la pantalla anterior. S'obrirà llavors la següent pantalla:

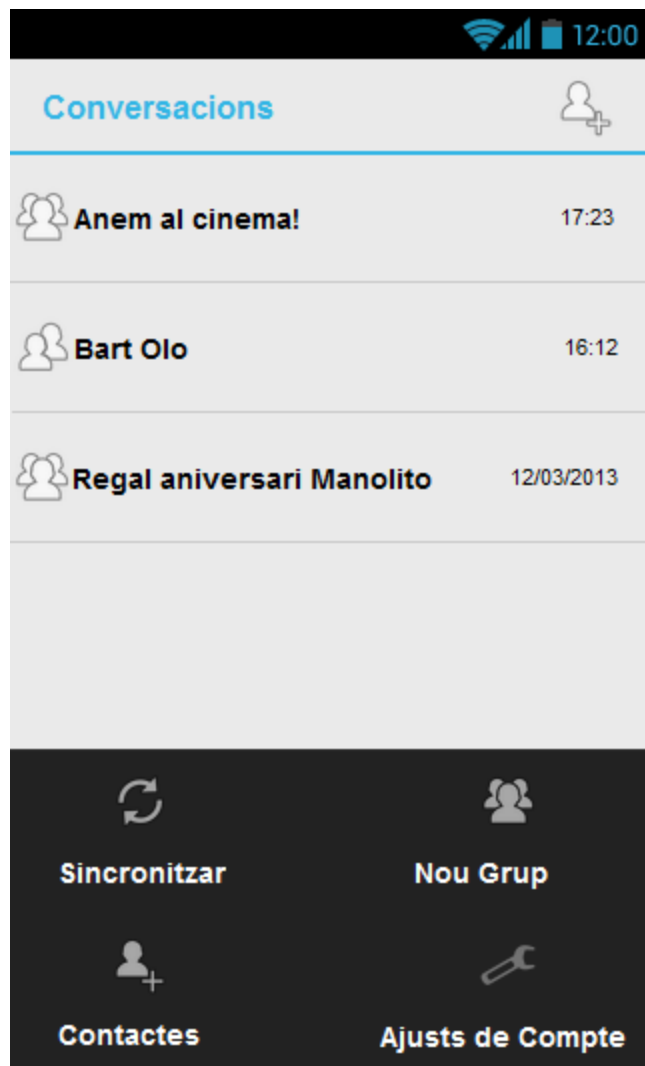


The screenshot shows a mobile application screen with a status bar at the top displaying signal strength, Wi-Fi, and the time 12:00. The main content area has a light gray background. At the top, there is a blue header with the text "Demanar nova contrassenya". Below this, there is a section with the text "Introdueix la teva adreça de correu electrònic i rebràs un email amb la contrasenya." followed by "El teu email:". Below the text is a text input field with a light blue border and a placeholder "E-mail". At the bottom of the screen, there is a gray button with the text "Demanar nova contrassenya".

L'usuari haurà d'introduir el compte de correu amb el qual es va enregistrar a l'aplicació i fer click al botó; llavors se li enviarà al compte d'email que hagi introduït (si és un compte d'email vàlid al servidor) la seva contrasenya per a accedir a l'aplicació.

6.3. Llista de converses

Un cop l'usuari s'ha enregirat a l'aplicació, la pantalla principal d'aquesta passarà a ser la llista de converses actives, de la qual es mostra el disseny tot seguit:



Aquesta pantalla mostrarà totes les converses actives, ordenades per activitat: les més recents es mostren més amunt. Cada conversa mostra una icona representant si es tracta d'una conversa entre l'usuari i un contacte (icona amb dos persones) o si es tracta d'un grup (icona amb tres persones). També es mostra el nom de la conversa i l'hora de l'últim missatge rebut o enviat.

Prement i mantenint el dit sobre una conversa podrem sortir d'aquesta i esborrar-la. Si es tracta d'una conversa de grup, es sortirà del grup i s'informarà a la resta de participants. Independentment de quin tipus de conversa sigui, s'esborraran els missatges emmagatzemats d'aquesta.

Al costat del nom de la pantalla trobarem una icona per accedir a la pantalla de gestió de contactes, on podrem afegir ràpidament contactes o converses amb un contacte.

A la part inferior, prement el botó d'opcions del terminal, apareixeran les opcions de l'aplicació. Hi hauran quatre opcions:

Sincronitzar: Fa una crida al servidor per a recuperar els missatges pendents de rebre per part del dispositiu.

Crear nou grup: accedirem a la pantalla per a crear un nou grup.

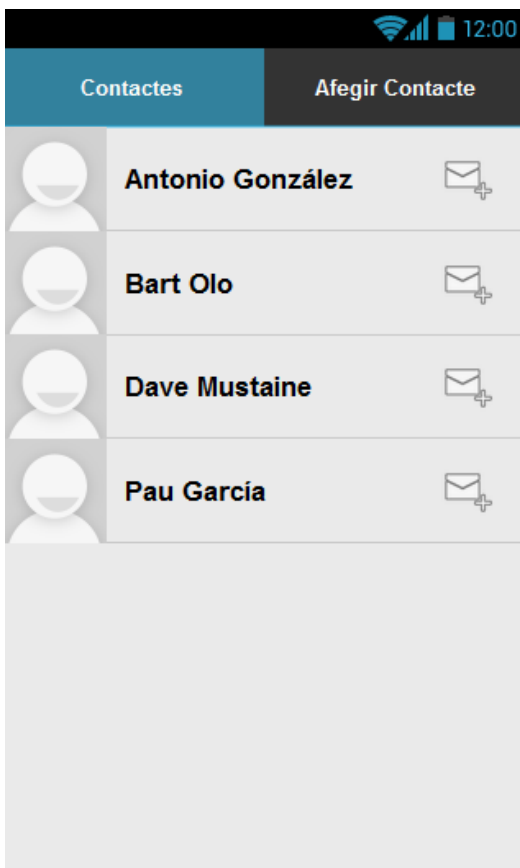
Contactes: accedirem a la pantalla dels contactes, on podrem afegir-ne de nous o iniciar una conversa individual amb qualsevol d'ells. Té la mateixa funció que el botó que es troba al costat del títol de la pantalla.

Ajusts del compte, on accedirem a la pantalla d'ajust del compte de registre a l'aplicació.

6.4. Contactes

La gestió dels contactes permet veure els contactes que tenim a l'aplicació, podent crear una nova conversa individual amb qualsevol d'ells, i també serveix per a afegir nous contactes. La interfície és la següent:

6.4.1. Llistat de contactes

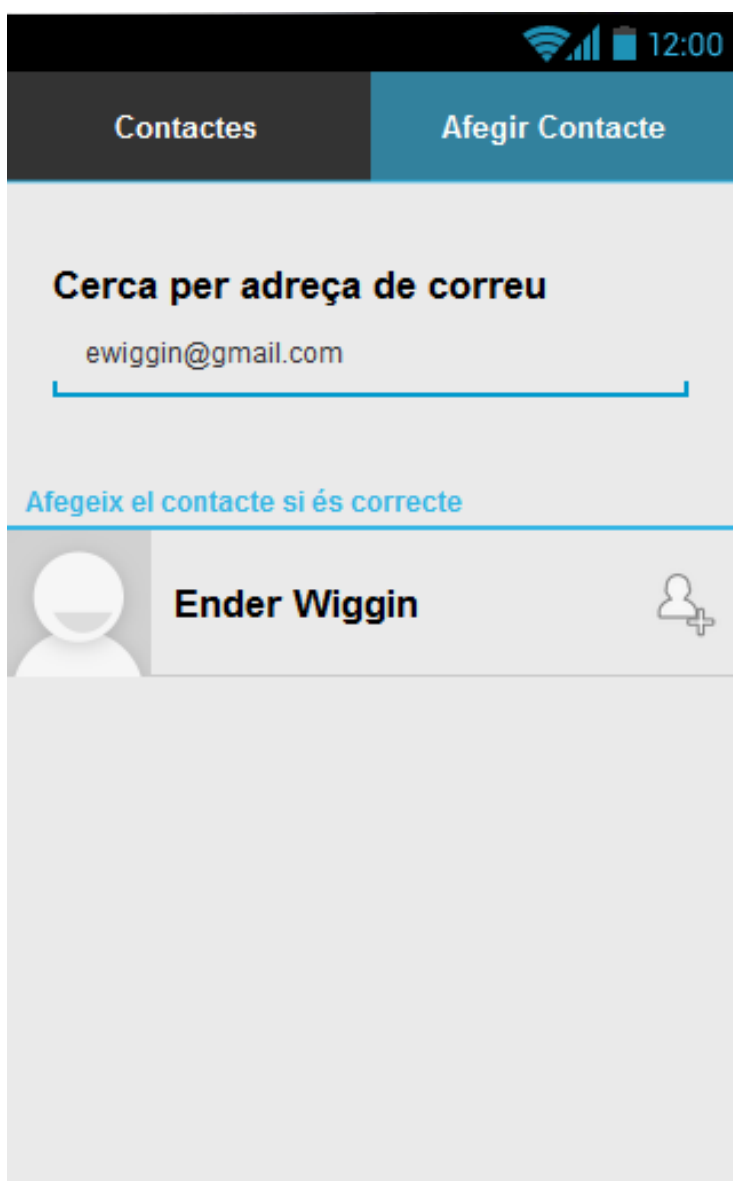


Ens trobem amb una pantalla amb dos pestanyes: una amb la llista de contactes i l'altra per a afegir nous contactes a la llista de contactes.

La interfície de llista de contactes ens mostrarà els contactes ordenats alfabèticament pel seu "nom per a mostrar", que hauràn introduït al enregistrar-se ells també a l'aplicació. En una segona versió de l'aplicació, els contactes podran afegir una foto seva, per això es reserva l'espai per aquesta mitjançant una icona. A la dreta del nom del contacte hi ha una icona d'un sobre per tal de crear una nova conversa amb el contacte seleccionat.

6.4.2. Afegir un nou contacte

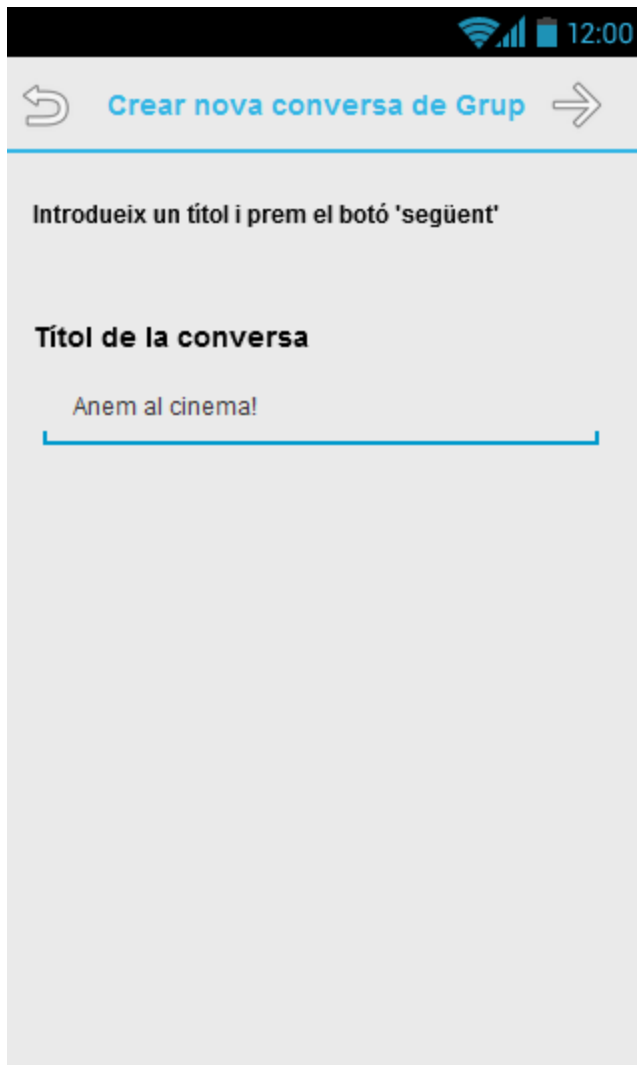
La interfície per afegir un nou contacte ens permet cercar un contacte al servidor mitjançant el seu nom; si el contacte existeix, s'afegirà a la llista de contactes; si no existeix, s'informarà de tal fet a l'usuari.



6.5. Crear un nou grup

Per tal de crear un nou grup, l'usuari haurà de seguir dos passos: introduir el nom del grup i després afegir als contactes que prendran part a la conversa.

Mirem detingudament el primer pas, que és introduir el nom del grup:



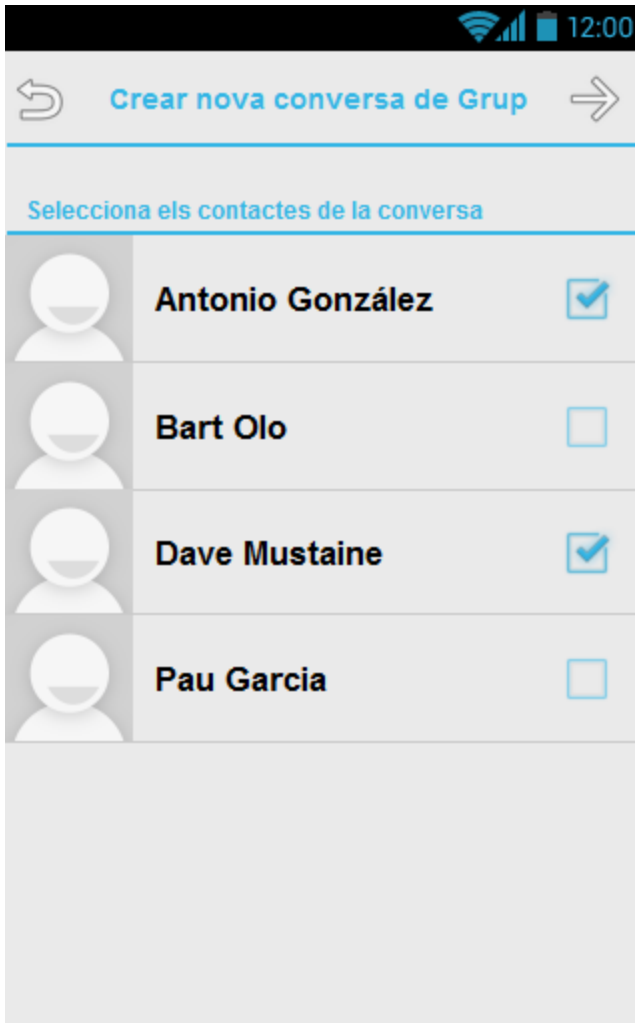
Només es tracta d'escriure el nom que vulguem i prémer la fletxa "següent" per anar al següent i darrer pas. També podem prémer el botó "enrera" i tornar a la llista de conversacions.

En aquest segon pas haurem d'afegir els contactes que volem que prenguin part a la conversa. Per tal de fer això, cada contacte disposarà a la seva dreta d'un element CheckBox per tal que puguem seleccionar aquells contactes que ens interessin.

Al títol de l'aplicació veurem els botons de navegació que ens permetran tornar enrere per a canviar el títol del grup o finalitzar la seva creació, establint-se el nou grup i convidant a tots els seus usuaris.

Cada usuari rebrà una notificació i el nou grup s'afegirà a la seva llista de converses.

A continuació es pot veure com és el disseny de la pantalla:



6.6. Pantalla de conversa

Aquesta pantalla és la que mostrarà tots els missatges que els usuaris hagin escrit a la conversa, ordenats per hora de publicació: els més nous apareixeran a sota.

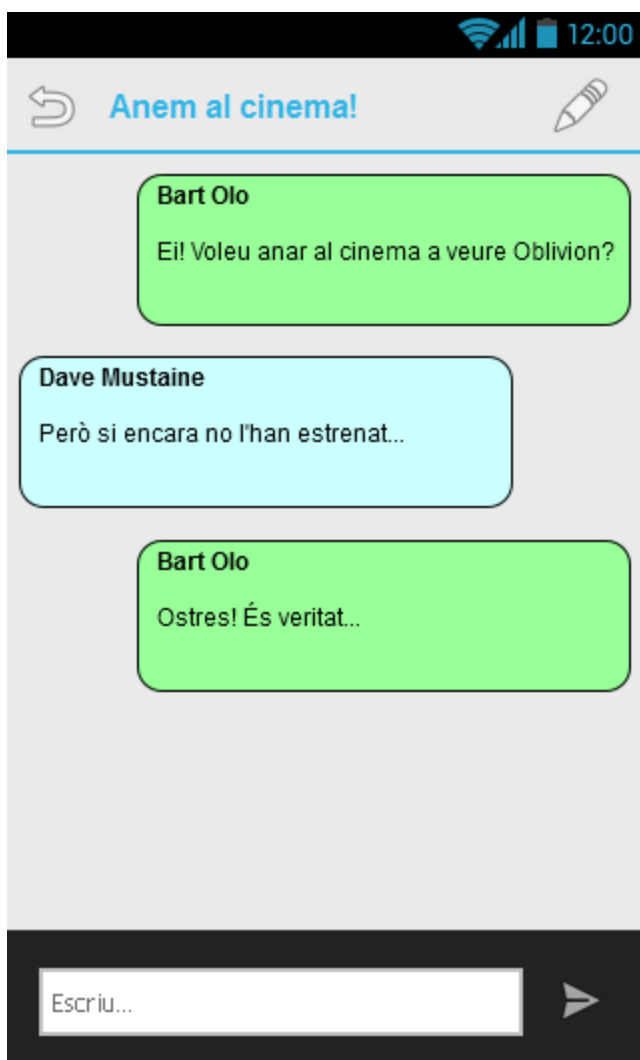
Cada missatge es mostrarà en una capsa: els missatges del propi usuari es veuran en una capsa desplaçada cap a la dreta, i els de la resta d'usuaris apareixeran desplaçats cap a l'esquerra.

Les capsas de missatges mostraran, a més del missatge, el nom de l'usuari que l'ha enviat i l'hora o data d'enviament.

De cara a una propera versió, seria interessant mostrar també un *hint* informant de si s'ha enviat i si ha estat llegit.

A la part inferior de la pantalla trobem la capsa de text on escriurem els missatges; al rebre el focus s'obrirà el teclat del dispositiu, i al prémer el botó d'enviament (fletxa a la dreta de la capsa de text) s'enviarà el missatge a tots els contactes de la conversa.

També podem tornar a la llista de converses mitjançant el botó “tornar” a l’esquerra del títol de la conversa, o editar les propietats de la conversa amb la icona del llapis, però només si es tracta d’una conversa de grup. Veiem com és el disseny de la interfície:



6.6.1. Editar la conversa

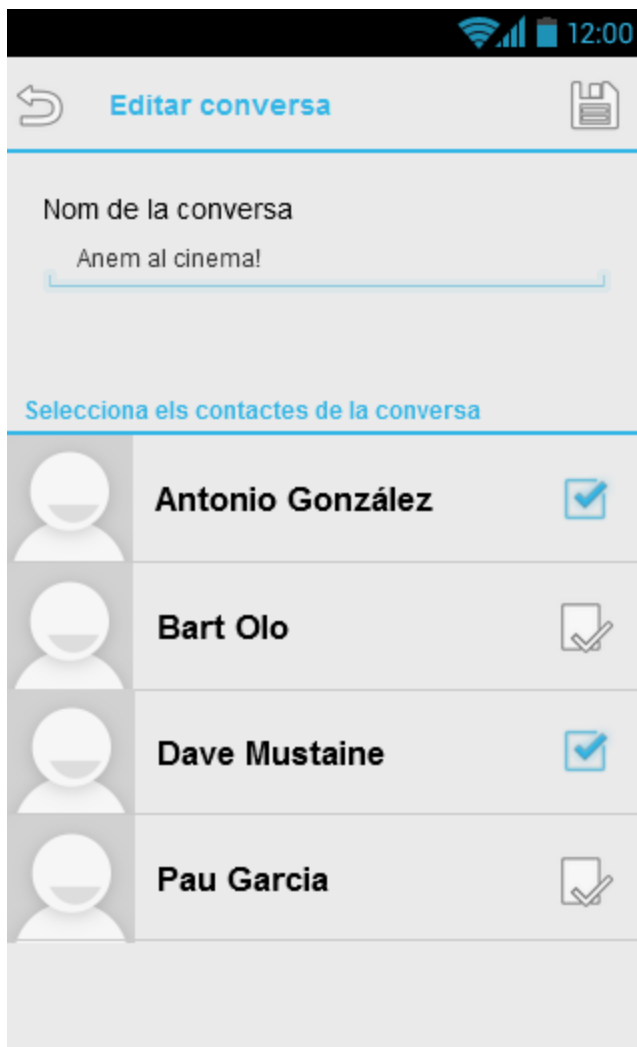
Aquesta pantalla serveix per a editar els paràmetres de la conversa. Només hi han dos paràmetres que es poden canviar d’una conversa:

- El títol d’aquesta
- La llista de participants

La interfície d’edició de la conversa permetrà editar aquests dos paràmetres en una mateixa pantalla. Per un costat, trobem un camp de text que conté el nom de la conversa; per l’altre, la llista de contactes de l’usuari amb un *checkbox* per a marcar els que volem afegir a aquesta. Els contactes que ja prenen part a la conversa apareixen amb el *checkbox* marcat, però deshabilitat. Això indica que no és possible fer-los fora. L’única manera d’esborrar un contacte d’una conversa és que aquest surti d’aquesta.

Per tal d'afegir usuaris a la conversa, només s'han de marcar els què volem afegir i clicar la icona de desar els canvis.

Tot seguit podem veure la interfície d'edició de la conversa:



A la barra del títol de la pantalla trobem els botons de navegació. El de l'esquerra ens permet tornar enrera, a la conversa, descartant els canvis que haguem efectuat. El de la dreta ens permet desar els canvis realitzats, actualitzant la conversa.

Al afegir un o més usuaris, es notificarà a tots els usuaris participants a la conversa.

6.7. Configuració del compte d'usuari

Aquesta pantalla conté un formulari mostrant el nom per a mostrar de l'usuari, la seva adreça de correu i dos camps per a introduir i confirmar una nova contrasenya.

L'usuari podrà canviar el seu nom per a mostrar (si es deixa en blanc no es canviarà) o canviar la contrasenya del seu compte al servidor i per tant a l'aplicació. No cal dir que ambdós camps de contrasenya han de coincidir.

En canvi, el camp de l'adreça de correu és només informatiu i està deshabilitat i no es podrà editar, ja que aquest és l'índex de la taula d'usuaris a la base de dades del servidor.

Per últim, al menú contextual d'aquesta pantalla trobem les següents opcions:

- Provar el servei: Demana al servidor que enviï al dispositiu una notificació, per tal de saber que el servei està aixecat.
- Esborrar el compte d'usuari del dispositiu: Si es prem i s'accepta la confirmació, s'esborrarà tota la base de dades de l'aplicació del dispositiu, es treurà el registre del dispositiu a Google Cloud Messaging i al servidor i es retornarà a la pantalla inicial de login/registre de l'aplicació.

The screenshot displays the 'Editar dades del compte' (Edit account data) screen. At the top, there is a back arrow icon and the title 'Editar dades del compte'. Below the title, there are four input fields: 'Nom per a mostrar' (Name to display) with the value 'Javier S.', 'Adreça de correu' (Email address) with the value 'javiasan@gmail.com', 'Contrassenya (en blanc per no canviar)' (Password (blank to not change)), and 'Confirmar contrassenya' (Confirm password). A button labeled 'Aplicar canvis' (Apply changes) is positioned below the form. At the bottom of the screen, there is a dark navigation bar with two buttons: 'Provar servei' (Test service) with an information icon, and 'Esborrar compte' (Delete account) with a delete icon. The status bar at the top shows signal strength, Wi-Fi, and the time 12:00.

A l'esquerra del títol de la pantalla trobarem també un botó per tal de tornar enrere, cap a la pantalla de la llista de missatges.

7. Disseny Tècnic

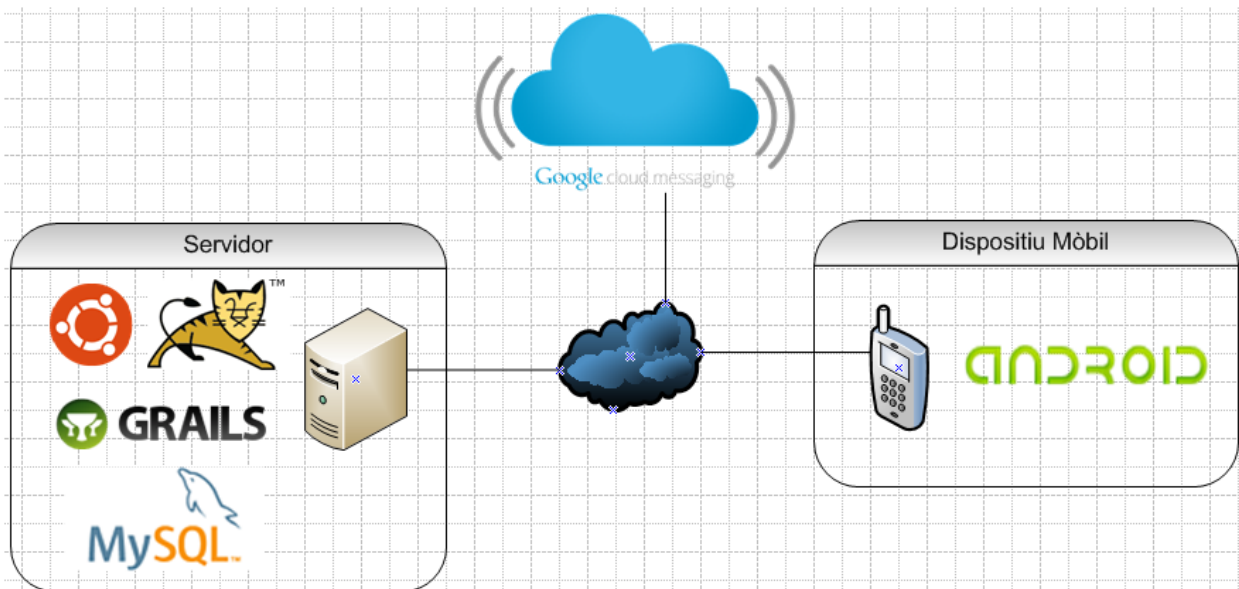
En aquest apartat s'exposarà tot el que fa referència a la part tècnica del disseny de l'aplicació, com l'arquitectura del servei, el disseny relacional de la base de dades i altra documentació que sigui necessària per a entendre correctament com funciona internament l'aplicació.

Des d'aquest paràgraf introductor es pot avançar que es tracta d'un projecte basat en un model Client-Servidor⁶, on els clients seran els diferents dispositius Android donats d'alta al sistema, i on hi haurà un servidor HTTP que processarà les comandes dels clients.

7.1. Diagrama de components de l'arquitectura

L'arquitectura del sistema la conformen diverses tecnologies que tot seguit veurem, i que totes plegades donen el servei que l'aplicació proposa.

Tot seguit veiem un diagrama que mostra aquests components:



Mirant l'esquema podem veure que els actors que participen a l'aplicació són:

- Servidor de l'aplicació
- Dispositiu mòbil Android
- Servidor de Google Cloud Messaging⁷

Tot seguit passem a descriure cadascú d'aquests actors, les tecnologies que empren i quin és el rol que té cadascú d'ells.

⁶ Wikipedia ("Client-Server model"). [en línia]. http://en.wikipedia.org/wiki/Client%E2%80%93server_model [data de consulta: 02/06/2013]

⁷ Google Cloud Messaging [en línia]. <http://developer.android.com/google/gcm/index.html> [data de consulta: 03/06/13]

7.1.1. Servidor

Es tracta d'una màquina Linux, amb sistema operatiu Ubuntu 10.04 a on s'ha instal·lat un servidor Tomcat, que es tracta d'un contenidor de *servlets* o aplicacions fetes amb Java.

El servidor *Tomcat* té desplegada una aplicació desenvolupada amb *Grails*⁸, que és un framework de desenvolupament MVC⁹ mitjançant *Groovy*¹⁰, *Spring*¹¹ i *Hibernate*¹². Per què s'ha emprat Grails? Perquè permet treballar no només amb Groovy sinó també amb Java, podent desenvolupar aplicacions MVC d'una manera molt ràpida, i també automatitza el treball amb bases de dades mitjançant *Hibernate*, ja què en comptes d'haver d'escriure fitxers de mapeig XML podem definir classes que es mapejaran com a objectes relacionals a la base de dades.

La base de dades emprada és *MySQL*¹³ degut a la seva versatilitat i senzillesa, i a la seva perfecta integració amb Linux. En aquesta base de dades s'emmagatzemaran les dades relatives als usuaris, a les converses i als missatges pendents de rebre per part dels dispositius.

La funció del servidor serà rebre les peticions dels dispositius mòbils, que poden ser missatges per a enviar a d'altres contactes, com peticions de recollida de dades. En el cas de les peticions de recollida de dades, es tracta de peticions HTTP, que es respondran mitjançant **JSON**¹⁴ (*JavaScript Object Notation*) directament al dispositiu per la xarxa d'Internet. Pel que fa als missatges que envien els dispositius i que s'han de reenviar a d'altres dispositius, el servidor ho comunicarà al servidor de Google Cloud Messaging per tal què aquest s'encarregui de notificar a la resta de dispositius que hi han missatges pendents de recollir.

En el moment en què un dispositiu reculli al servidor un missatge i ho notifiqui a en aquest, el missatge s'esborrarà del servidor. Es generarà, això sí, un registre d'activitat de cara a generar estadístiques d'ús de la missatgeria, per tal de generar una gràfica:

Estadísticas de Uso



A més, al servidor disposem també d'un servei de correu **Postfix** que s'encarrega d'enviar emails. Això només passarà quan un usuari s'oblidi de la seva contrasenya i així ho demani a la

⁸ Grails [en línia] <http://www.grails.org/> [data de consulta: 03/06/13]

⁹ Wikipedia ("MVC") [en línia] <http://en.wikipedia.org/wiki/Model%E2%80%93View%E2%80%93Controller> [data de consulta: 03/06/2013]

¹⁰ Groovy [en línia] <http://groovy.codehaus.org/> [data de consulta: 03/06/2013]

¹¹ Spring [en línia] <http://www.springsource.org/> [data de consulta: 03/06/2013]

¹² Hibernate [en línia] <http://www.hibernate.org/> [data de consulta: 03/06/2013]

¹³ MySQL [en línia] <http://www.mysql.com/> [data de consulta: 03/06/2013]

¹⁴ JSON [en línia] <http://www.json.org/> [data de consulta: 03/06/2013]

pantalla corresponent; en aquest moment, el servidor executarà un script **Python** que generarà un email que serà processat per l'abans comentat servei de correu.

Així doncs, resumint, les característiques de la part servidor seràn les següents:

- Mantenir un rol passiu, estant constantment en execució a l'espera de les sol·licituds dels clients (dispositius mòbils).
- Processar aquestes sol·licituds un cop arribin, i un cop finalitzades, tornar a l'espera.

7.1.2. Dispositiu mòbil

Es tractarà dels dispositius mòbils que consumiran el servei mitjançant l'aplicació instal·lada en ells.

Aquests dispositius enviaran nous missatges pertanyent a les converses al servidor, per tal que aquest els difongui a la resta de dispositius que prenen part a la conversa. També recolliran les dades del servidor quan els hi arribi una notificació PUSH des del servidor de Google Cloud Messaging.

Totes les comunicacions amb el servidor es faran mitjançant HTTP, enviant els missatges encapsulats amb JSON.

Tanmateix, els dispositius emmagatzemaran les dades a una base de dades SQLite local a l'aparell, i a la que només tindrà accés l'aplicació.

7.1.3. Google Cloud Messaging

Es tracta d'un servei de Google que permet enviar notificacions PUSH als dispositius enregistrats en ell.

Per a registrar un dispositiu, aquest necessita d'un codi d'aplicació, que li proporciona la mateixa App instal·lada. El dispositiu mirarà d'enregistrar-se a GCM mitjançant aquest codi d'aplicació, i si té èxit rebrà un Register ID que l'identificarà únicament com a dispositiu enregistrat a l'aplicació donada. Quan el dispositiu informa d'aquest Register ID al servidor de l'aplicació, es formalitza el registre del dispositiu en aquesta.

Mitjançant aquest Register ID, el servidor de l'aplicació podrà demanar a Google Cloud Messaging que enviï una notificació al dispositiu, per tal que aquest darrer sàpiga que disposa de noves dades per rebre.

Així doncs, la funció de Google Cloud Messaging és la d'enregistrar dispositius a l'aplicació, i també la de permetre al servidor d'aquesta demanar l'enviament de notificacions als diferents dispositius enregistrats amb ell.

7.2. Interfície amb el servidor

El servidor només podrà enviar al mòbil notificacions PUSH per tal que aquest comenci una petició HTTP contra el server. Així doncs, totes les comunicacions entre servidor i dispositiu les començarà aquest darrer.

Només hi ha un cas en què el servidor inicia les comunicacions: si hi han missatges pendents de rebre per cap usuari, al servidor hi ha un procés **Cron** que cada 10 minuts ho notifica a aquets.

Totes les respostes del servidor seran en format JSON, i sempre incluiran un camp “status”, que serà 0 si la resposta és correcta, o 1 si ha hagut cap error (dades no trobades, usuari invàlid...)

Veiem quins tipus de requests pot realitzar el dispositiu sobre el servidor. A tots ells, **el dispositiu enviarà l’email de l’usuari i la seva contrasenya**, per tal que el servidor verifiqui que es tracta d’un usuari del servei.

Les accions dels controladors encarregats d’escoltar per HTTP són:

Controlador	Mètode	Descripció
UserController	userRegister	El dispositiu envia al servidor el nom de l’usuari, l’email, la contrassenya i el RegisterID amb GCM per tal que el servidor l’emmagatzemi i el registri com a nou usuari.
UserController	userUnregister	El dispositiu envia al servidor un avís per tal que esborri el RegisterID de l’usuari, conforme s’ha desregistrat al servei.
UserController	searchUser	El dispositiu cerca un contacte. Envia un email, i el servidor li respon amb el nom del nou contacte i status=0, o amb status=1 si no l’ha trobat.
ChatController	createChat	S’ha creat una nova conversa al dispositiu i aquest li envia al servidor totes les dades referents a aquesta: nom de la conversa, un array amb els usuaris implicats, si és un grup o no... Si el servidor respon amb status=0 i l’ID de la nova conversa; s’ha creat. En el cas de converses amb un contacte, abans de crear la conversa es mira si ja existeix una conversa entre tots dos contactes i si es així, es retorna el id de la conversa. Així si algú ja parla amb algú altre i esborra la conversa, i la torna a crear, no es crearà una de nova, i l’altre usuari continuarà a la mateixa conversa a la què ja hi era.
ChatController	updateChat	Es canvia el nom de la conversa o s’afegeixen nous usuaris. S’envia en nom de la conversa i un array amb els nous contactes. Si el nom és diferent, es notifica a

		tots els usuaris de la conversa, els nous inclosos. També es notifica a tothom que s'han afegit nous usuaris amb aquesta.
ChatController	sendMessage	L'usuari envia un nou missatge a una conversa. S'emmagatzema al servidor i es notifica a la resta d'usuaris de la conversa. Es genera un registre de missatge pendent per cada usuari de la conversa, que s'esborrarà quan l'usuari se'l descarregui.
UserController	updateAccount	L'usuari envia el seu nou nom per a mostrar i/o la seva nova contrasenya. També s'envia l'antiga, per tal que el servidor pugui saber que el missatge és realment seu.
MessageController	getUserMessages	L'usuari demana al servidor els seus missatges pendents de rebre. Aquesta acció pot ser executada al rebre una notificació, o al sincronitzar manualment les converses per part de l'usuari. El servidor munta un array amb els missatges pendents de l'usuari i se-l's envia. Cada missatge conté un camp type que s'interpreta de la següent manera: <ul style="list-style-type: none"> - 1: Missatge normal escrit per qualsevol usuari - 2: S'ha agregat un usuari a una conversa. - 3: S'ha marxat un usuari d'una conversa. - 4: Un usuari canvia el títol de la conversa. Com que cada missatge conté el nom i l'email de qui l'envia, és senzill saber qui entra o se'n va, qui canvia el títol, si un usuari es canvia el nom, etc. Després serà funció de l'Adapter canviar la vista o el resource de cada missatge a la llista de converses.
MessageController	msgReaded	Un cop rebuts tots els missatges per part del servidor, al processar-los, el dispositiu munta un array amb les IDs dels missatges i l'envia al servidor per tal que aquest els dongui per llegits i els esborri de la taula de missatges pendents de la base de dades.
ChatController	userLeavesGroup	Quan un usuari deixa una conversa, fa una crida a aquesta acció per a notificar-ho al sistema. Quan el darrer usuari d'una conversa marxa, aquesta s'esborra del

		servidor.
UserController	userPing	Per tal de provar el servei, s'envia l'email d'usuari (amb la contrassenya, com sempre) per tal que el servidor generi una notificació de prova. El tipus de la notificació és 0, així que el dispositiu només genera la notificació però no la tracta.
UserController	userLogin	A l'hora d'autenticar-se, un usuari envia el seu compte de correu i el seu password. Si tot va bé, el servidor respon amb status=0 i li envia també el seu nom d'usuari.
UserController	getUserPass	Si un usuari no s'enrecorda de la seva contrassenya, fa una crida a aquest web service passant-li el seu email, i el servidor li envia un correu electrònic amb el seu password.

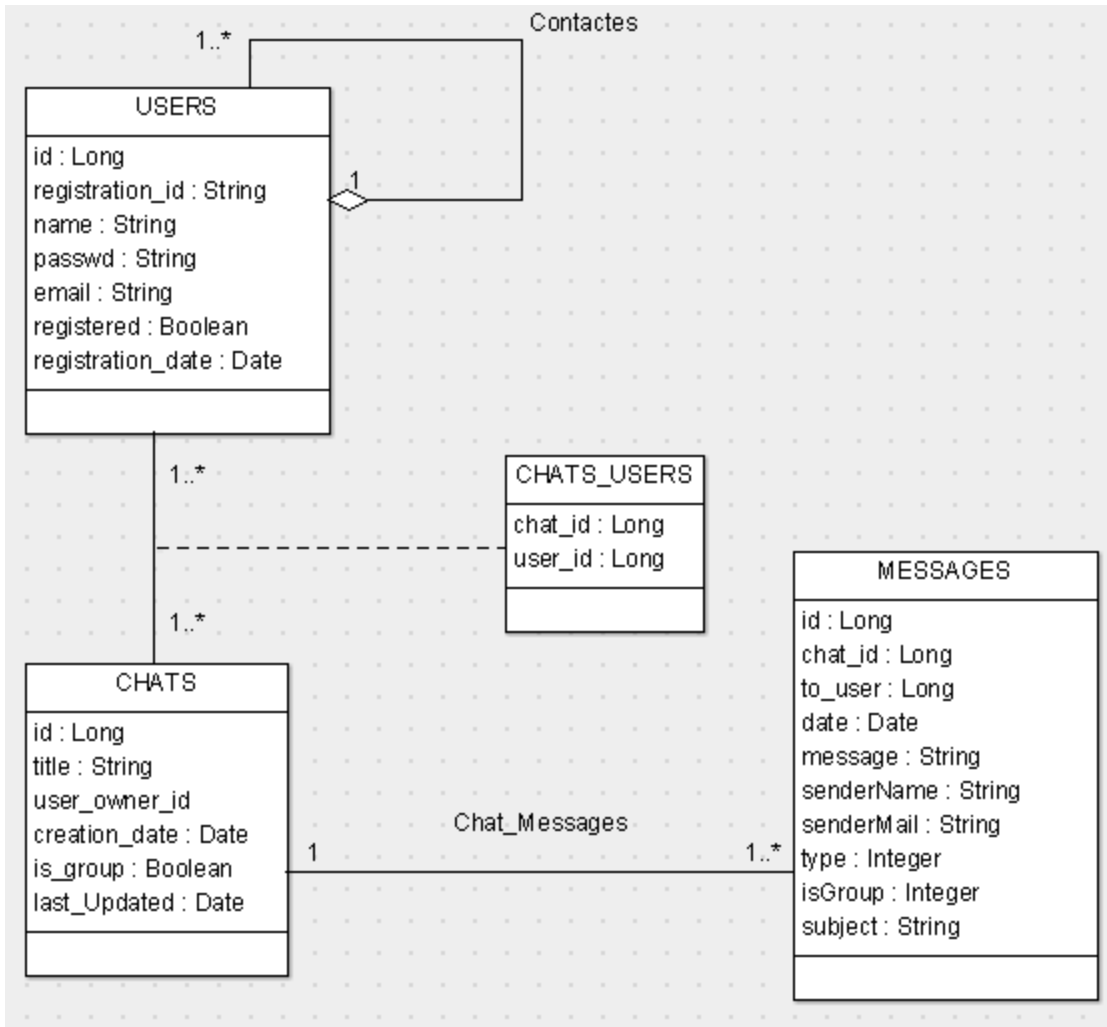
Per tal d'enviar el correu electrònic, el servidor té instal·lat el servidor de correu **Postfix**, així que des de Java només fem una crida a un **script Python** que s'encarrega de compondre el correu i entregar-lo al servidor de correu.

7.3. Disseny relacional de la Base de Dades

El disseny relacional de l'aplicació es divideix en dos parts: el disseny de la base de dades del servidor i el disseny de la base de dades del dispositiu.

7.3.1. Disseny de la Base de Dades del Servidor

A la part del servidor disposem d'una base de dades que emmagatzema tota la informació referent als usuaris, converses, contactes, missatges... Veiem com és l'esquema d'aquesta base de dades, creat amb ArgoUML (<http://argouml.tigris.org/>)



Aquest esquema relacional, ja normalitzat, ens mostra les taules que intervendran a l'aplicació des del servidor.

7.3.1.1 Taula Users

Camp	Descripció	Tipus	Longitud	Clau
Id	ID del usuari	BigInt	20	Primary
registration_id	Codi de registre de GCM	Text		No
Name	Nom de l'usuari	VarChar	75	No
Passwd	Password de l'usuari	VarChar	16	No
Email	Adreça de correu de l'usuari	VarChar	75	Unique
registered	Flag: està enregistrat?	Int	1	No

La taula **USERS** té emmagatzemats tots els usuaris que s'han enregistrat a l'aplicació. Dels usuaris s'emmagatzema el seu nom a mostrar, el seu email, la contrassenya, si està enregistrat o no (per exemple, s'ha tret el registre del dispositiu), i també s'emmagatzema el seu Register ID proporcionat per Google per a la nostra aplicació; s'empra per a identificar únicament els dispositius, per tal de poder enviar al dispositiu una notificació.

En el cas que un usuari esborri el seu compte del dispositiu, no s'esborra al servidor però sí que es modifica el flag registrered a zero, per a indicar que actualment no té cap dispositiu vinculat.

7.3.1.2. Taula Chats

Camp	Descripció	Tipus	Longitud	Clau
Id	ID del chat	BigInt	20	Primary
Date_create	Data de creació del chat	Varchar	14	No
Date_update	Data de la darrera actualització	Varchar	14	No
Is_group	És un grup (1) o una conversa (0)	Int	1	No
Title	Títol del chat	Varchar	75	No
User_owner_id	ID de l'usuari que l'ha creat	BigInt	20	No

També disposem d'una taula **CHATS** on s'emmagatzemaran totes les converses actives. Per conversa activa s'entén aquella a la que hi pren part almenys un usuari. Si els usuaris van sortint de les converses, quan surt el darrer usuari la conversa s'esborra. Ens interessa emmagatzemar el ID, la data de creació, la darrera data d'activitat a la conversa, si és un grup o una conversa entre dos contactes, el títol de la conversa i l'usuari que l'ha creat.

7.3.1.3. Taula Chats_Users

Camp	Descripció	Tipus	Longitud	Clau
chatId	ID del chat	BigInt	20	Primary
userId	ID de l'usuari	BigInt	20	Primary

A la taula **CHATS_USERS** emmagatzemarem un registre per cada parell de dades usuari – conversació. Així és senzill consultar a quines converses intervé un usuari. Si un usuari es

compra un nou dispositiu i es registra, en una propera versió se li podria enviar totes les converses a on prenia part.

7.3.1.4. Taula Messages

Camp	Descripció	Tipus	Longitud	Clau
Id	ID del missatge	BigInt	20	Primary
chatId	ID del chat	BigInt	20	No
contactMail	Email del remitent	Varchar	75	No
contactName	Nom del remitent	Varchar	75	No
dateTime	Data i hora d'enviament	Varchar	14	No
isGroup	Flag si es tracta d'un grup el chat	int	1	No
message	Text del missatge	text		No
toUser	Email de l'usuari destinatari	Varchar	75	No
type	Tipus de missatge: 1 – Missatge normal 2 – Usuari afegit al grup 3 – Usuari marxa del grup 4 – Usuari canvia nom del grup	int	1	No
Subject	Títol del chat (grup)	Text		No

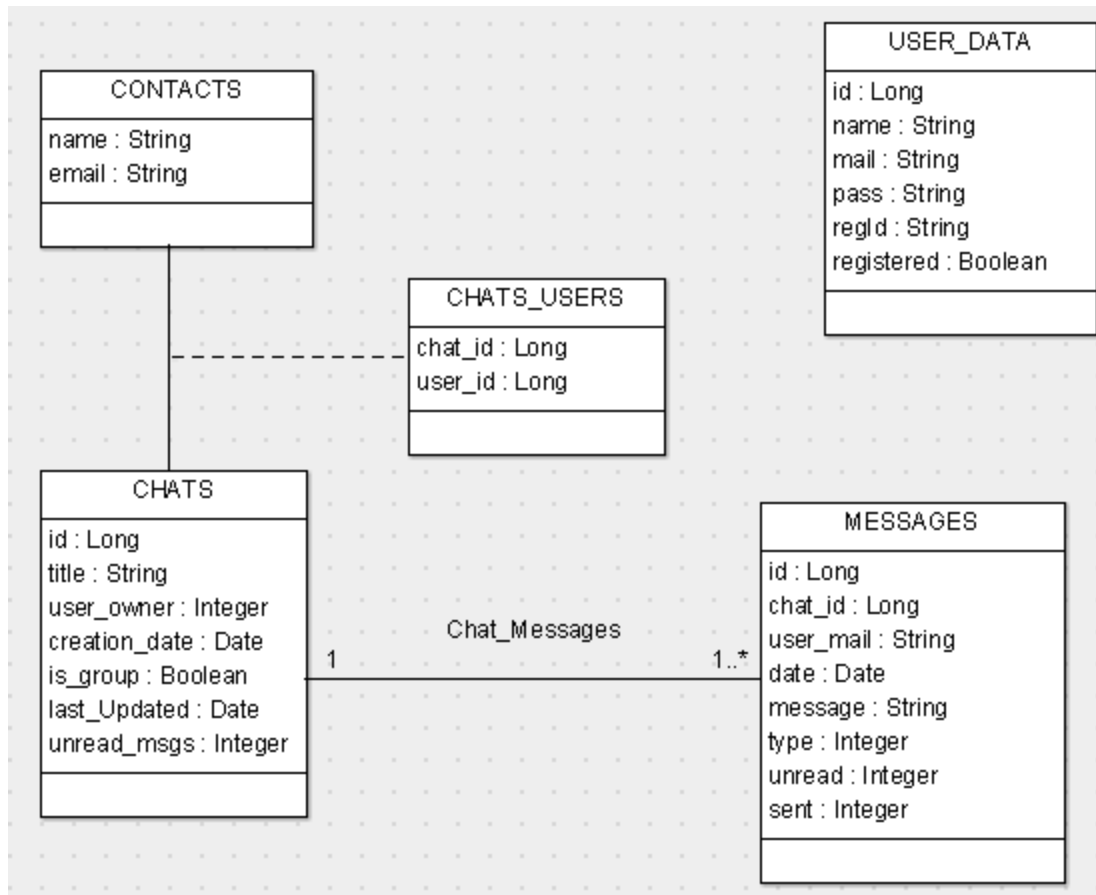
La taula **MESSAGES** emmagatzema cadascú dels missatges que s'envien a les converses. Quan un dispositiu rep una notificació *PUSH* mitjançant *GCM*, aquest es connecta i es descarrega tots els missatges que tingui pendents de rebre. Un cop descarregats, s'esborren del servidor. Es fa d'aquesta manera i no a dins de les mateixes notificacions *GCM* perquè aquestes tenen un límit de 4 KB d'informació adjunta, i podria no cabre tot el missatge.

Cada missatge, a més, disposa de les dades del remitent (nom i email de qui envia el missatge), perquè d'aquesta manera podem afegir nous contactes fàcilment a la nostra llista de contactes. Passa el mateix amb el títol de la conversa (si es tracta d'un grup) i el flag *isGroup*: d'aquesta manera, en el moment que un dispositiu consulti els seus missatges pendents (mitjançant el seu email, camp *toUser*) no haurem de realitzar joins a les consultes al servidor,

agilitzant la velocitat d'execució. Tanmateix, com que els missatges un cop confirmats s'esborren, no suposa un creixement d'espai a la base de dades.

7.3.2. Disseny de la Base de Dades del dispositiu

Un cop hem presentat l'esquema relacional de la base de dades al servidor, toca veure com s'emmagatzemaran les dades al dispositiu. Fem un cop d'ull al seu model relacional:



Per un costat tenim la taula **USER_DATA**, on s'emmagatzemen les dades de configuració o registre del compte d'usuari: nom a mostrar, email, contrasenya, Register ID a Google Cloud Messaging i si està enregistrat o no el dispositiu. També s'emmagatzema el ID de l'usuari, encara que sempre es treballarà amb el ID 1.

A la taula **CONTACTS** emmagatzemarem les dades dels contactes, dels quals només ens interessa conèixer el nom per a mostrar i el seu email.

La taula **CHATS** emmagatzema la informació de les converses, i el seu contingut és el mateix que vam veure per al servidor.

Hi haurà una taula **MESSAGES** on s'emmagatzemaran tots els missatges de les converses: ID de missatge, ID de conversa, email de l'usuari que ha enviat el missatge (si està buit és que és del mateix usuari del dispositiu), data i hora del missatge, el missatge en si mateix i el tipus de missatge.

Aquest tipus de missatge serà un sencer per a distingir a nivell d'aplicació els possibles missatges. Els tipus possibles són els següents:

- 1: Missatge normal
- 2: Usuari afegit a la conversa
- 3: Usuari ha deixat la conversa
- 4: Usuari canvia títol de la conversa

Es fa aquesta distinció per tal de mostrar un literal (cadena de text) o altre en funció de quin sigui el tipus de missatge, o per mostrar el missatge rebut si el tipus és 1. No ha de sortir igual si un contacte escriu quelcom que si un usuari es marxa de la conversa.

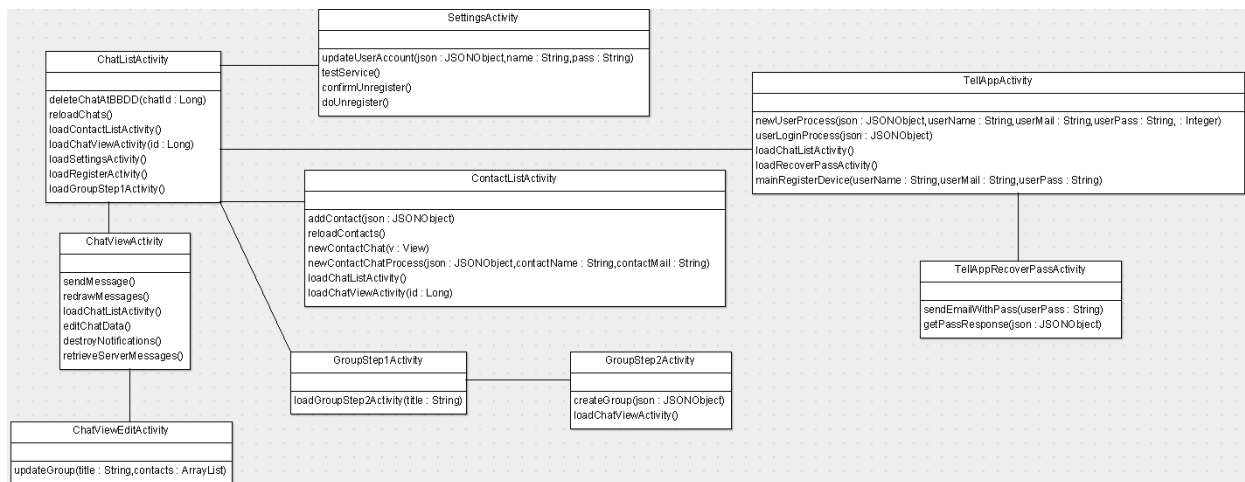
Per últim, la taula **CHATS_USERS** relaciona les converses amb els contactes, i podem saber quants contactes hi prenen part en una conversa fent una cerca per ID de conversa.

7.4. Diagrama de classes

Podem englobar les classes de l'aplicació segons la funció que tenen dins d'aquesta. Els dos grups principals de funcionalitat són la capa de visualització i la capa d'accés a dades i al servidor. A continuació es descriuran les classes que pertanyen a cadascú d'aquests tipus.

7.4.1 Capa de visualització

Les classes que fomen part de la capa de visualització són els *Activities*. En el següent esquema es mostra quines classes d'aquests tipus hi han, i els que estan relacionats entre sí.



Les relacions mostrades representen la navegació entre els diferents Activities o pantalles. A continuació es descriu quina és la utilitat de cadascuna d'aquestes classes que conformen la capa de visualització:

Classe	Funció
TellAppActivity	Pantalla d'inici de l'app. Aquí l'usuari haurà d'autenticar-se o registrar-se al servei.

TellAppRecoverPassActivity	Es fa servir quan un usuari no recorda la seva contrasenya, per introduir la seva adreça de correu i rebre-la.
ChatListActivity	Aquesta classe construeix la llista de converses a les que l'usuari pren part.
ChatViewActivity	Construeix el fluxe de missatges de la conversa passada per paràmetre al cridar al Intent.
ChatViewEditActivity	Edita els paràmetres de la conversa.
ContactListActivity	Gestió dels contactes. Construeix dos <i>tabs</i> o pestanyes: un amb la llista de contactes i l'altre per a cercar més contactes per la seva adreça de correu electrònic.
GroupStep1Activity	Primer pas per a crear un nou grup de conversa.
GroupStep2Activity	Segon pas per a crear un nou grup de conversa.
SettingsActivity	Edició de les dades del compte d'usuari.

7.4.2. Capa d'accés a dades i al servidor

La capa d'accés a dades i al servidor inclou aquelles classes que implementen serveis o eines comunes per a accedir a la base de dades i/o a les comunicacions amb el servidor. Aquestes classes són les següents, representades en forma de llista perquè no tenen relació directa l'una amb les altres:

Classe	Funció
GCMIntentService	Aquesta classe s'encarrega d'enregistrar i treure el registre a l'aplicació, i rebre i tractar les notificacions de Google Play.
CommonUtilities	En aquesta classe es defineixen constants que s'empraran a les altres classes de l'aplicació.
HttpServerAsync	Classe de suport creada per a tractar amb les connexions HTTP asíncrones amb el servidor. Simplifica les connexions amb aquest.
ServerUtilities	Aquesta classe conté constants amb les adreces del servidor i mètodes per a interactuar amb ell.

TellAppDB	Classe que genera, esborra i modifica la base de dades. Conté mètodes per a tractar totes les consultes que es realitzin amb la base de dades SQLite.
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------

Aquestes altres classes representen objectes emmagatzemats a la base de dades:



Com que tot està a la base de dades SQLite, s'anirà consultant segons faci falta, per no carregar la memòria amb objectes. En el cas de les tres classes que es mostren, ni tan sols estàn relacionades entre elles, ja que no fa falta.

Per què no fa falta? Perquè per exemple, a la llista de converses veurem les converses que hi han, però no pas els missatges; llavors, què sentit té tenir un ArrayList amb tots els missatges d'una conversa? O tots els contactes que premen part en una conversa?

S'ha pensat que és millor que aquestes classes simples representin els ListItems: si estem veient la llista de converses o de contactes, o els missatges d'una conversa, és millor tenir les dades a una col·lecció dels tipus Conversation, Contact o Message.

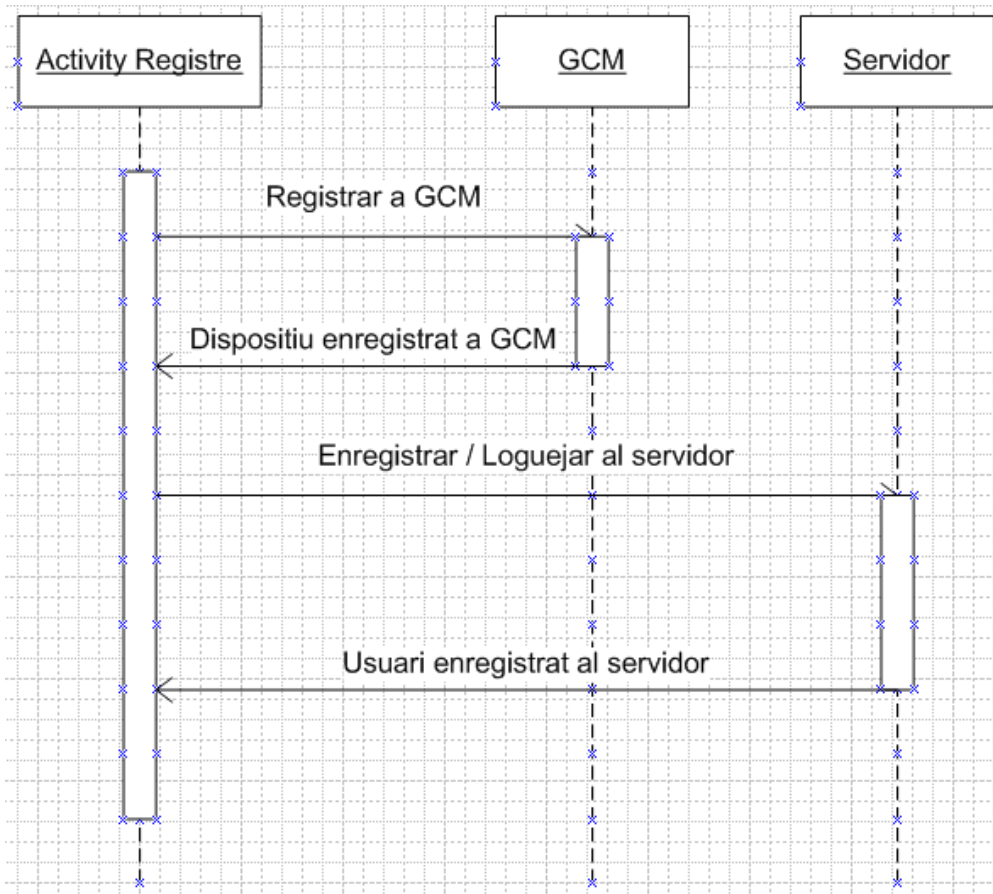
7.5. Diagrames de Seqüència

El funcionament de l'aplicació és bastant senzill e intuïtiu i per la majoria d'accions no calen més explicacions, però en aquest apartat s'ampliarà més la informació de què fa l'aplicació mitjançant uns diagrames de seqüència en aquells casos en què es pugui aportar quelcom més.

7.5.1 Registre del dispositiu

Veiem a continuació el diagrama de seqüència de l'enregistrament del dispositiu, tant de si es tracta del registre com de l'autenticació.

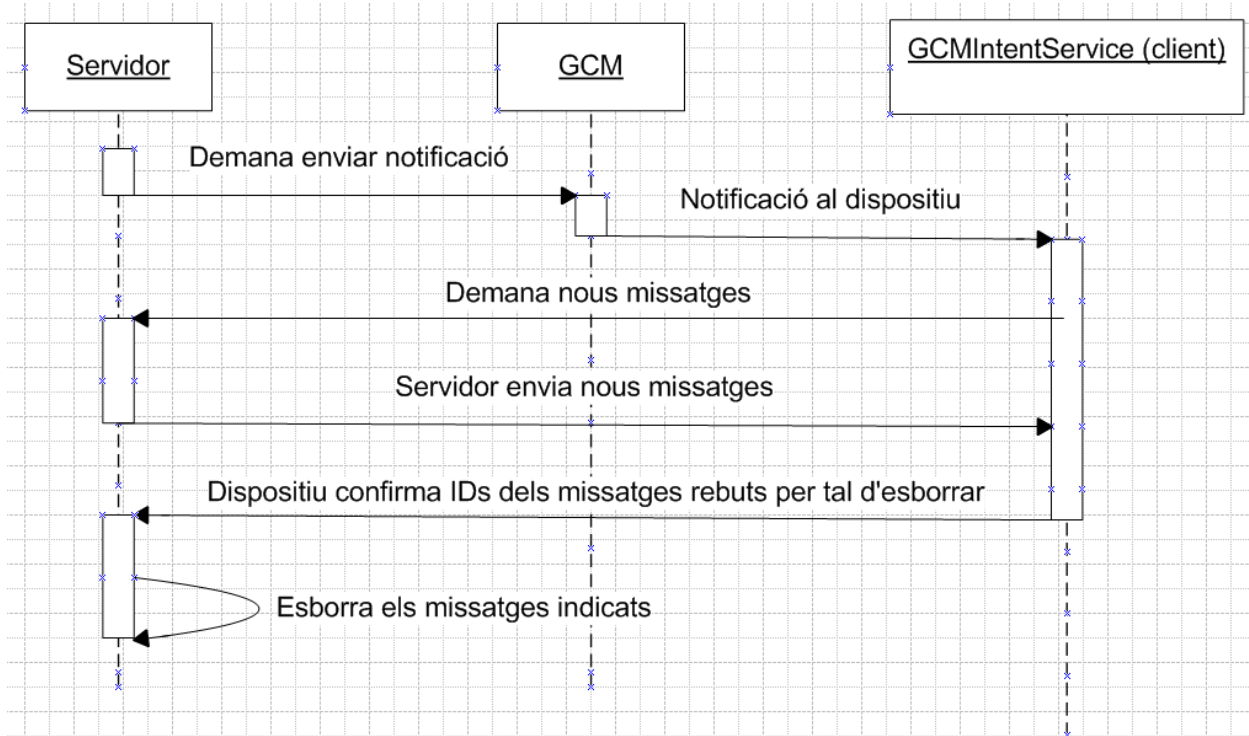
A l'esquema que es veu més a baix podem veure que l'aplicació, mitjançant el formulari de la pantalla inicial, recull les dades de l'usuari; llavors es registra l'aplicació a Google Cloud Messaging i s'obté el Register ID del dispositiu amb l'aplicació. El següent pas és enviar les dades de l'usuari juntament amb el Register ID al servidor de l'aplicació, on es finalitza el registre.



Si ha anat tot correctament, el dispositiu està enregistrat; en cas contrari, es treu el registre del dispositiu, ja que no s'ha pogut registrar al servidor.

7.5.2. Recepció d'un missatge

La recepció de missatges per part dels dispositius mòbils clients consta de diverses passes, que tot seguit es veuran mitjançant un diagrama de seqüència:



Pot semblar que és el servidor qui inicia aquest procés, però no és així: es tracta d'un altre dispositiu que ha enviat un missatge a, entre d'altres, el nostre client d'exemple.

El servidor té un missatge per al mòbil client, llavors crida al servei de Google Cloud Messaging, per tal que notifiqui al destinatari de què disposa de nous missatges per a descarregar. Un cop fa això, el servidor passa a l'estat espera, fins que rebí cap altra petició per part d'un client.

El client destinatari, al rebre el missatge, demana al servidor que li enviï tots els seus missatges pendents de rebre.

El servidor rep la petició del client, es torna a activar i li envia tots els seus missatges. Torna a entrar en mode d'espera.

El client processa els missatges i va muntant una llista amb les IDs dels missatges que ha processat correctament. Quan ha enllestit amb tots els missatges, torna a realitzar una petició al servidor, aquest cop comunicant-li quins missatges ha processat amb èxit.

El servidor rep aquesta llista de missatges i els esborra de la base de dades. Així doncs, l'escenari ideal és aquell en què no hi ha cap missatge emmagatzemat al servidor, perquè tots els usuaris s'han descarregat els seus missatges.

8. Implementació

En aquest capítol es pretén explicar les decisions preses a l'hora de realitzar la implementació de l'aplicació en aquells punts més crítics, i exposar com s'han resolt.

8.1. Google Cloud Messaging

Com que el projecte es tracta d'una app de missatgeria, és necessari realitzar connexions amb el servidor per tal de rebre i enviar els missatges. El procés d'enviament de missatges no té més misteri, ja que quan l'usuari envia un missatge aquest s'envia al servidor; però, quan hem de rebre els missatges del servidor?

Una primera aproximació possible seria realitzar peticions de missatges al servidor espaiades al temps, per exemple cada dos o tres minuts; però això comporta els següents problemes:

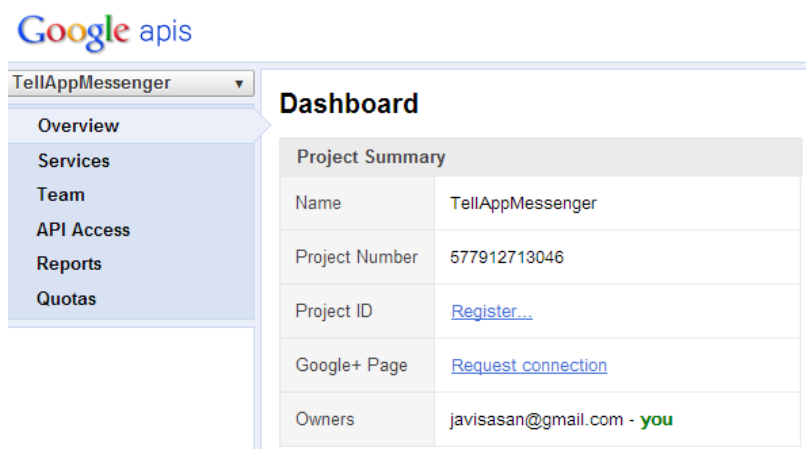
- Per un costat, els missatges que rebrem no seràn *real-time* (en temps real).
- D'altra banda, i això és encara més crític, si durant varies hores no rebem cap missatge però l'aplicació es connecta cada interval de temps prefixat, estarem consumint tant bateria com servei de dades.

Per tal de resoldre aquests problemes, la plataforma Android proporciona l'eina *Google Cloud Messaging*¹⁵, que integra les notificacions *PUSH*, que fan que tan aviat com el dispositiu disposi de notificacions se li enviïn.

Per tal de fer servir Google Cloud Messaging (GCM), s'han hagut de donar les següents passes que veiem a continuació:

8.1.1. Registre a Google Cloud Messaging

El primer que hem de fer és anar a *Google APIS Console*¹⁶ i enregistrar la nostra aplicació, per a obtenir el codi de projecte que la identifica:



The screenshot shows the Google APIs Console interface. On the left, there is a navigation menu with options: Overview, Services, Team, API Access, Reports, and Quotas. The main content area is titled 'Dashboard' and contains a 'Project Summary' table with the following information:

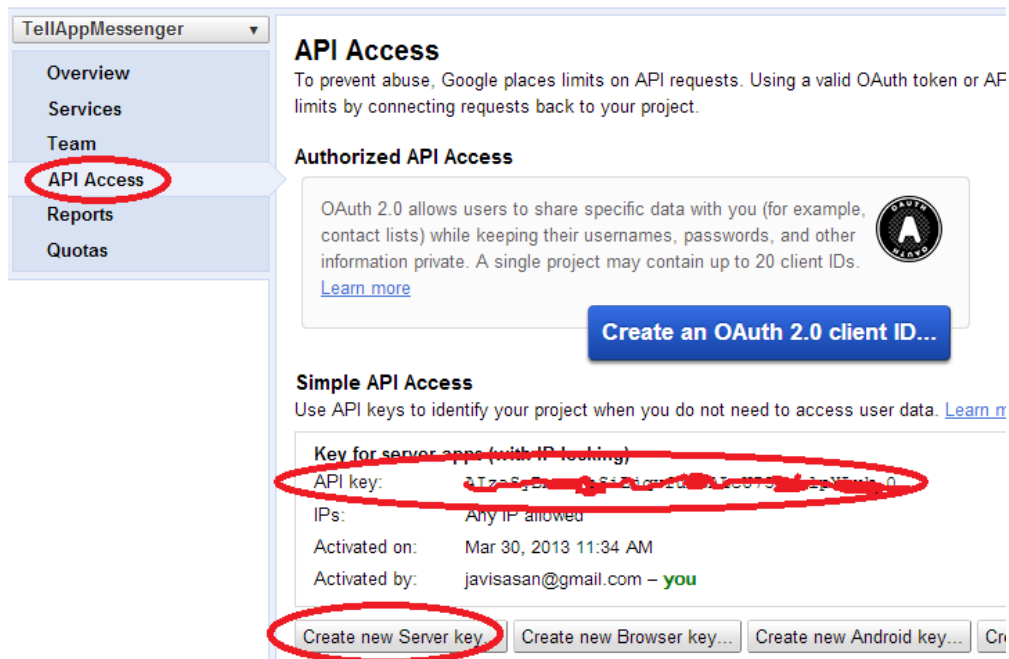
Project Summary	
Name	TellAppMessenger
Project Number	577912713046
Project ID	Register...
Google+ Page	Request connection
Owners	javisasan@gmail.com - you

¹⁵ Google Cloud Messaging (en línia) <http://developer.android.com/google/gcm/index.html> [data consulta: 04/06/2013]

¹⁶ Google Apis Console (en línia) <https://code.google.com/apis/console> [data consulta: 04/06/2013]

Aquest número de projecte és el que haurem de posar a la nostra aplicació Android per tal d'enregistrar el dispositiu amb el projecte.

Un cop tinguem el número de projecte, haurem de demanar una clau (*API Key*) per a fer servir GCM des del nostre servidor. Anirem a la pestanya API Access i generarem una nova clau de servidor:



The screenshot shows the Google API Access console. On the left, a navigation menu for 'TellAppMessenger' has 'API Access' circled in red. The main content area is titled 'API Access' and includes a 'Create an OAuth 2.0 client ID...' button. Below that, the 'Simple API Access' section is visible, with a sub-section 'Key for server apps (with IP locking)' circled in red. This section displays an API key, IP settings (Any IP allowed), and activation information. At the bottom, the 'Create new Server key...' button is also circled in red.

Aquesta API Key és la que haurem de fer servir al servidor per a comunicar-nos amb el dispositiu.

8.1.2. GCM al servidor

Per tal de fer servir GCM al servidor, farem servir el plugin *Grails Android GCM plugin*. Per tal d'instal·lar-ho, només haurem d'anar a la carpeta del projecte des de la consola i teclejar la comanda:

```
grails install-plugin android-gcm
```

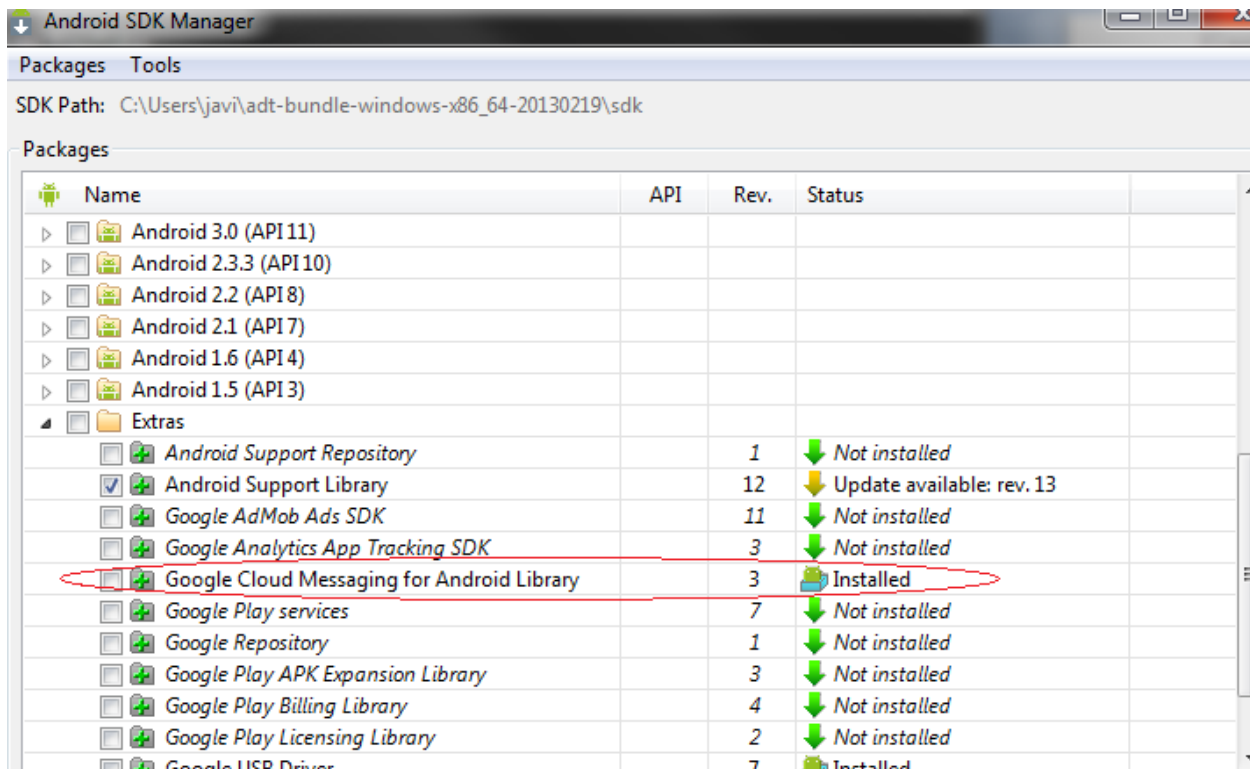
Ara, per enviar notifikacions des del servidor als dispositius, haurem de fer servir el següent codi:

```
def key = grailsApplication.config.api.key.config.property.name
flash.message = androidGcmService.sendMessage (
    [
        'subject'      : "PRUEBA",
        'type'         : "0"
    ],
    [user.registrationId],
    'prueba', key
)
```

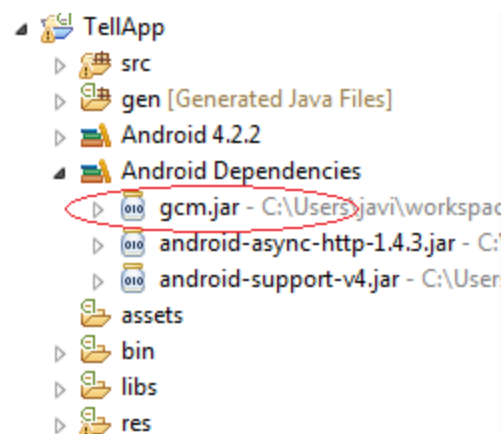
Com es pot apreciar, l'API Key proporcionada per Google (variable *key*) es recupera des de la configuració del projecte, així si aquesta canvia només l'hem de canviar a un lloc.

8.1.3. GCM al dispositiu

Per tal de fer servir GCM des del dispositiu, primer hem d'instal·lar, des de l'Android SDK Manager, el component extra *Google Cloud Messaging for Android Library*:



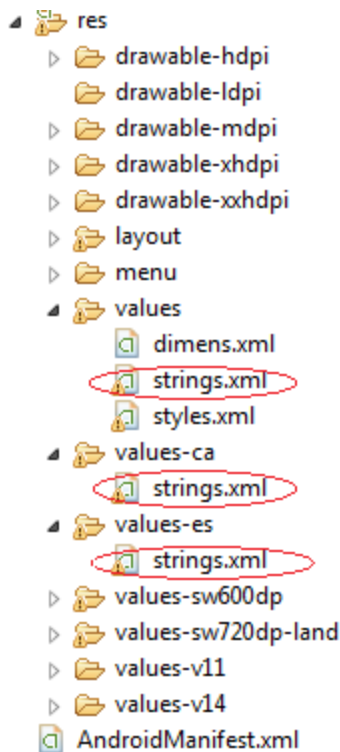
I després haurem d'afegir la llibreria *.jar* corresponent al nostre projecte:



Des de la classe **GCMIntentService** es gestiona tot el referent a les notifikacions GCM a nivell de dispositiu, com rebre notifikacions, o realitzar l'enregistrament o desenregistrament. Per tal d'això, es fa servir la variable **SENDER_ID** que no és més que el número de projecte proporcionat per *Google APIS Console*.

8.2. Localització

L'aplicació es troba localitzada als idiomes **anglès** (éssent l'idioma per defecte), **català** i **castellà** . Les frases o *literals* en els diferents idiomes es troben en el fitxer *strings.xml* de la carpeta de recursos de l'aplicació, dintre d'una altra carpeta on s'emmagatzemen els valors:



Hi ha una carpeta per defecte i una carpeta per a cadascuna de les localitzacions:

- values : Carpeta per defecte. Es deixa en anglès.
- values-ca : Carpeta per localització en català.
- values-es : Carpeta per localització en castellà.

Si el nostre idioma és català o castellà, s'agafarà el literal de la carpeta corresponent; si el idioma del dispositiu no és cap d'aquests, s'agafarà el literal de la carpeta per defecte. En el cas que un literal no s'hagi traduït a un idioma que no sigui el per defecte, es mostrarà el literal amb l'idioma per defecte. Si un literal no té traducció a l'idioma per defecte, pot produir un error en temps d'execució; per tant s'ha d'anar amb compte de que TOTS els literals es trobin a l'idioma i carpeta per defecte.

Per a recuperar un literal emprem el següent codi:

```
Resources res = getResources();
String miStr = res.getString(R.string.dialog_connecting);
```

8.3. Llistes d'objectes. ListViews i Adapters

A l'aplicació s'empren molt activament les llistes: llista de missatges, de contactes, de converses... Però a l'hora de representar aquestes llistes i personalitzar-les s'han trobat algunes dificultats. En aquest apartat es veurà com s'ha resolt aquest punt.

Per a representar una llista de dades customitzada, hem de tenir els següents components:

- Una classe que representi un element (o Item), i que extengui a LinearLayout
- Un adaptador que generi tants items o elements de la classe anterior com objectes li passen
- Un layout per a representar gràficament a cada element

La idea és que l'adaptador rebi una llista d'objectes, que convertirà a elements o Items. Llavors cada item representa l'objecte rebut segons el layout donat.

D'aquesta manera s'automatitza la creació de llistes. Per exemple, aquest seria un **Item**:

```
public class ItemChat extends LinearLayout {
    private TextView lblTitle, lblDate, lblUnread, lblShortDesc;
    private Chat chat;
    public ItemChat(Context context, Chat chat) {
        super(context);
        this.chat = chat;
        inicializar();
    }
    private void inicializar() {
        String infService = Context.LAYOUT_INFLATER_SERVICE;
        LayoutInflater li = (LayoutInflater)getContext().getSystemService(infService);
        li.inflate(R.layout.item_chatlist, this, true);
        // Calculamos fecha a mostrar...
        String date = chat.getDate().toString();
        SimpleDateFormat df = new SimpleDateFormat("yyyyMMdd");
        String today = df.format(new Date());
        if(date.substring(0, 8).equals(today.substring(0, 8))){
            // Solo hora
            date = date.substring(8,10)+"."+date.substring(10,12);
        }else{
            // Dia
            date = date.substring(6,8)+"/"+date.substring(4, 6)+"/"+date.substring(0, 4);
        }

        lblTitle = (TextView) findViewById(R.id.lblChatTitle);
        lblDate = (TextView) findViewById(R.id.lblChatDate);
        lblUnread = (TextView) findViewById(R.id.lblChatUnread);
        lblShortDesc = (TextView) findViewById(R.id.lblChatShortDesc);

        String title = chat.getTitle();
        if(title.length() > 24){
            title = title.substring(0, 23) + "...";
        }
        lblTitle.setText(title);
        lblDate.setText(date);
        //lblUnread.setText(unread);
        lblShortDesc.setText(chat.getLastMessage());

        // Mensajes sin leer
        String unread = chat.getUnread();
        if(unread.equals("0")) {
            unread = "";
            lblUnread.setText(unread);
        }else{
            lblUnread.setBackgroundResource(R.drawable.item_chat_unread_count);
            lblUnread.setText(unread);
        }
    }
}
```

Aquest seria l'adaptador:

```
public class ItemChatAdapter extends BaseAdapter {

    private Activity activity;
    private List<Chat> listChats;
```

```

// Constructor
public ItemChatAdapter(Activity activity, List<Chat> listChats){
    this.activity = activity;
    this.listChats = listChats;
}

public int getCount() {
    return listChats.size();
}

public Object getItem(int position) {
    return listChats.get(position);
}

public long getItemId(int position) {
    Chat chat = listChats.get(position);
    return chat.getId();
}

public View getView(int position, View convertView, ViewGroup parent) {
    ItemChat lstItem = new ItemChat(activity, listChats.get(position) );
    return lstItem;
}
}

```

I a l'activity hauríem de posar aquest codi:

```

private void reloadChats() {
    // Obtenemos los Chats de la base de datos
    TellAppDB db = new TellAppDB(context);
    chatList = db.getChats();
    db.cleanup();
    db = null;

    // Obtenemos el ListView del layout
    chatsView = (ListView) findViewById(R.id.chatListView);

    // Rellenamos el ListView con los chats
    adapter = new ItemChatAdapter(this, chatList);
    chatsView.setAdapter(adapter);
    adapter.notifyDataSetChanged();
}

```

Pel que fa al layout, seria:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    style="@style/ChatListItem"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/lblChatDate"
        style="@style/ChatListItemShadow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:text="@string/chat_item_date"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <TextView
        android:id="@+id/lblChatTitle"
        style="@style/ChatListItem"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:text="@string/chat_item_title"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView

```

```

android:id="@+id/lblChatShortDesc"
style="@style/ChatListItemShadow"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignTop="@+id/lblChatUnread"
android:text="@string/chat_item_last"
android:textAppearance="?android:attr/textAppearanceSmall" />
</RelativeLayout>

```

I això muntaria una llista amb les converses, com la que es veu a sota.



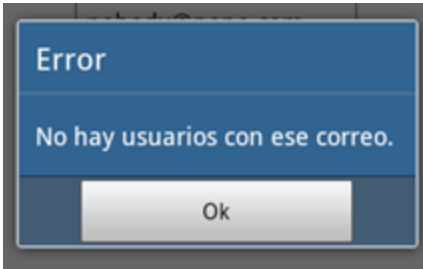
Un cas especial de llista és la llista dels missatges d'una conversa. Per tal de fer que al obrir una conversa es mostrés sempre el darrer missatge situat a sota de tot, s'ha hagut de fer una nova classe, *CustomListView*, que hereta de *ListView*, per tal de sobrecarregar el seu mètode *onSizeChanged()* i fer que es situï al final, seleccionant el darrer element.

8.4. Diàlegs modals

Hi han tres possibles diàlegs modals: els avisos que es descarten amb un botó d'ok, els confirmadors, amb dos botons acceptar o cancel·lar, i els diàlegs de progrés, que s'han resolt de la manera següent:

8.4.1. Diàlegs d'avís

Els diàlegs d'avís són els més simples; es mostra un diàleg amb un títol i un text, i un botó per a tancar el diàleg.



Com que es fan servir bastant a l'aplicació, s'ha realitzat una classe de suport que els construeix, **CustomDialog**. D'aquesta manera, allà on vulguem mostrar un diàleg d'avís, només haurem de posar el següent codi:

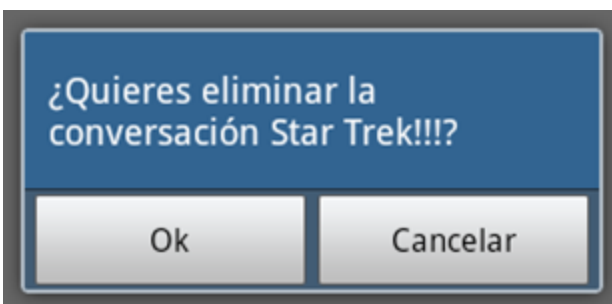
```
CustomDialog dialog = new CustomDialog(context, "títol", "missatge");
```

8.4.2. Diàlegs de confirmació

Són els diàlegs més elaborats, ja que s'han de definir les accions a prendre en el cas de prémer el botó de confirmació o el de cancel·lació. Com que no es fan servir gaire a l'aplicació, s'ha optat per no fer una classe de suport pera fer-los servir i posar directament el codi allà on faci falta. Per mostrar un diàleg de confirmació farà falta el següent codi:

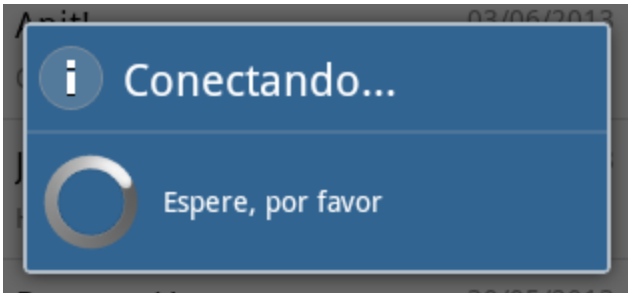
```
AlertDialog.Builder builder = new AlertDialog.Builder(context);
builder.setMessage("missatge");
builder.setPositiveButton("botó ok", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int id) {
        // Acció confirmació
    }
});
builder.setNegativeButton("botó cancel", new
DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int id) {
        // Acció cancelació
    }
});
AlertDialog dialog = builder.create();
dialog.show();
```

La representació del diàleg és asíncrona, així que si volem tractar les pulsacions dels botons, ho haurem de fer dintre de la definició del botó, ja sigui allà mateix o cridant a una funció. El diàleg resultant és el següent:



8.4.3. Diàlegs de progrés

Els diàlegs de progrés es mostren quan l'aplicació està fent alguna acció, per indicar a l'usuari que efectivament l'està realitzant, i es tanquen quan l'acció finalitza.



Per tal de mostrar un diàleg de progrés, primer definim a la classe un atribut de tipus **ProgressDialog**:

```
private ProgressDialog prgDialog = null;
```

Llavors, allà on vulguem mostrar el diàleg posarem el següent codi:

```
prgDialog = ProgressDialog.show(context, "titol", "missatge");
```

I quan l'operació que s'estava realitzant finalitzi, tancarem el diàleg amb el següent codi:

```
prgDialog.dismiss();
```

8.5. RESTful asíncron

Donat que durant el desenvolupament es feia servir un mòbil amb la versió d'Android 2.3, es va passar per alt que a partir de la versió d'Android 3 cap endavant el comportament de la classe `HttpPost` passava a ser asíncron; així doncs, les consultes al servidor donaven error, perquè es passava a aquella condició abans de rebre el resultat d'aquest. Es va resoldre aquesta situació fent servir una llibreria externa, *Android Asynchronous HTTP Client*¹⁷, emprada per apps importants com ara **Instagram**, **Pinterest** i d'altres. Aquesta llibreria permet fer les consultes al servidor des del mateix *UI thread* (fil d'execució de la interfície d'usuari) d'una manera molt senzilla. Es va realitzar una classe de suport, **HttpServerAsync**, per a assistir a les crides HTTP al servidor, de manera que als *Activities* l'execució es va simplificar de la següent manera:

```
// Request al servidor REMOTO de manera ASINCRONA
HttpServerAsync.post("adreça_del_servidor", params, new JsonHttpResponseHandler() {
    @Override
    public void onSuccess(JSONObject json) {
        // Codi o crida en cas que la crida al servidor tingui èxit.
    }
    // En caso de error...
    public void onFailure(Throwable e){
        // Codi o crida en cas que la crida al servidor falli.
    }
});
```

D'aquesta manera tan elegant es va aconseguir solucionar el problema de les crides al servidor asíncrones.

¹⁷ Android Asynchronous HTTP Client (en línia) <http://loopj.com/android-async-http/> [data de consulta: 05/06/2013]

9. Manual d'Usuari

Aquest manual d'usuari pretén explicar com funciona l'aplicació per tal que qualsevol usuari aprengui totes les seves característiques.

Encara que les Apps han de tenir una experiència d'usuari senzilla i auto-explicativa, no està de més disposar, i fer una ullada, al seu manual d'usuari; sempre hi poden haver *features* desconegudes, sobretot per a certs grups d'usuaris, com la gent gran.

Bàsicament, les funcionalitats que l'aplicació permet són les següents:

- Enregistrar-nos al sistema, iniciar sessió o recuperar la contrasenya del nostre compte.
- Cercar altres usuaris i afegir-los a la llista de contactes de l'App
- Crear converses amb un altre contacte o amb un grup de contactes
- Enviar missatges dins de les converses i editar aquestes
- Canviar la configuració del compte: nom de l'usuari, contrasenya
- Enviar una notificació de prova al dispositiu
- Treure el registre del dispositiu i esborrar les dades

9.1. Registre de l'usuari

El primer pas que ha de donar l'usuari un cop descarregada l'app és iniciar sessió o enregistrar-se com a nou usuari del servei. És per això que la pantalla inicial de l'app disposa de dos pestanyes amb aquest fi.

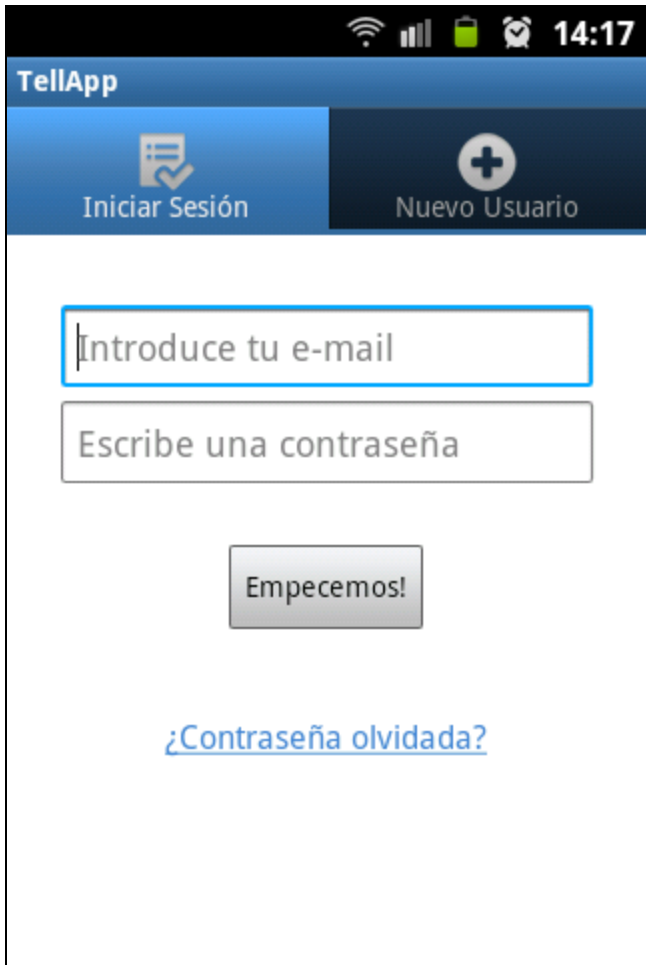
9.1.1. Iniciar sessió

Un usuari que s'ha registrat anteriorment al sistema ja disposa d'un compte, i el pot tornar a fer servir per a connectar-se amb un altre dispositiu (per exemple, es canvia el mòbil). Per tal d'autenticar-se a l'app fent servir aquest mateix compte, l'usuari ha d'emprar la pestanya *Iniciar Sessió*, on podrà omplir les seves dades d'usuari, que són:

Compte de correu electrònic

Contrassenya

El compte de correu electrònic és l'identificador únic de cada usuari al sistema, i aquest té associada una contrassenya que l'usuari ha assignat.



Pantalla de inici de sessió

Un cop introduïdes aquestes dades, i si que siguin correctes, l'usuari accedirà a l'entorn de missatgeria de l'app. En el cas que no siguin correctes o que no es pugui connectar amb el servidor, l'app ho notificarà amb una finestra modal.

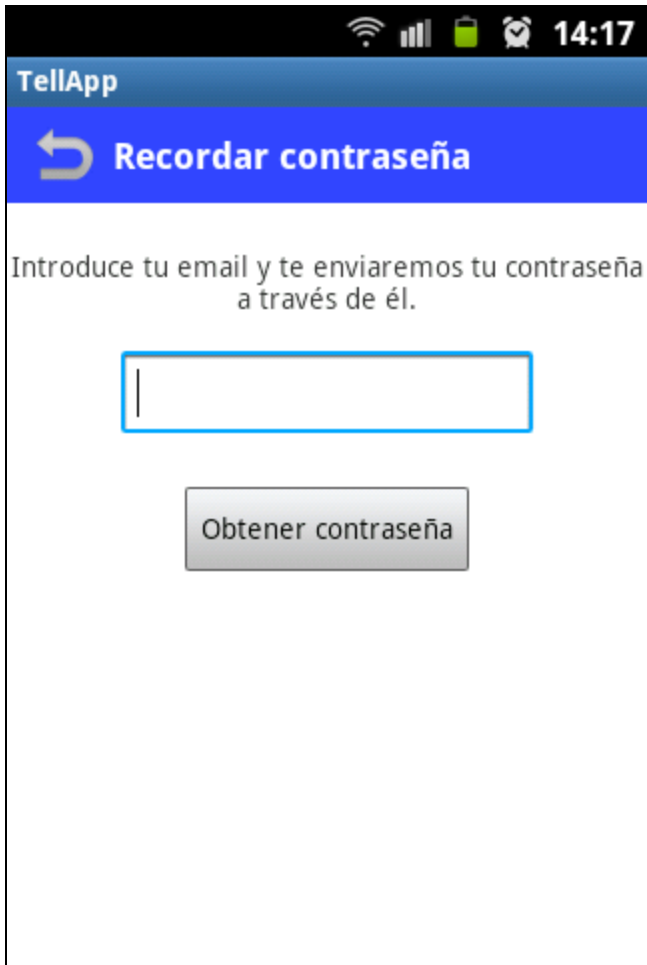
Pot succeïr que l'usuari no s'enrecordi de la seva contrassenya; en aquest cas pot fer servir l'enllaç situat sota el formulari, "Contrassenya oblidada?", per tal d'accedir a una nova pantalla on podrà escriure el seu email i, si és un usuari que té un compte creat al sistema amb la adreça de correu introduïda, se li enviarà un email a aquesta contenint la seva contrassenya.

En el cas que un usuari faci servir aquesta pantalla, hi ha tres resultats possibles:

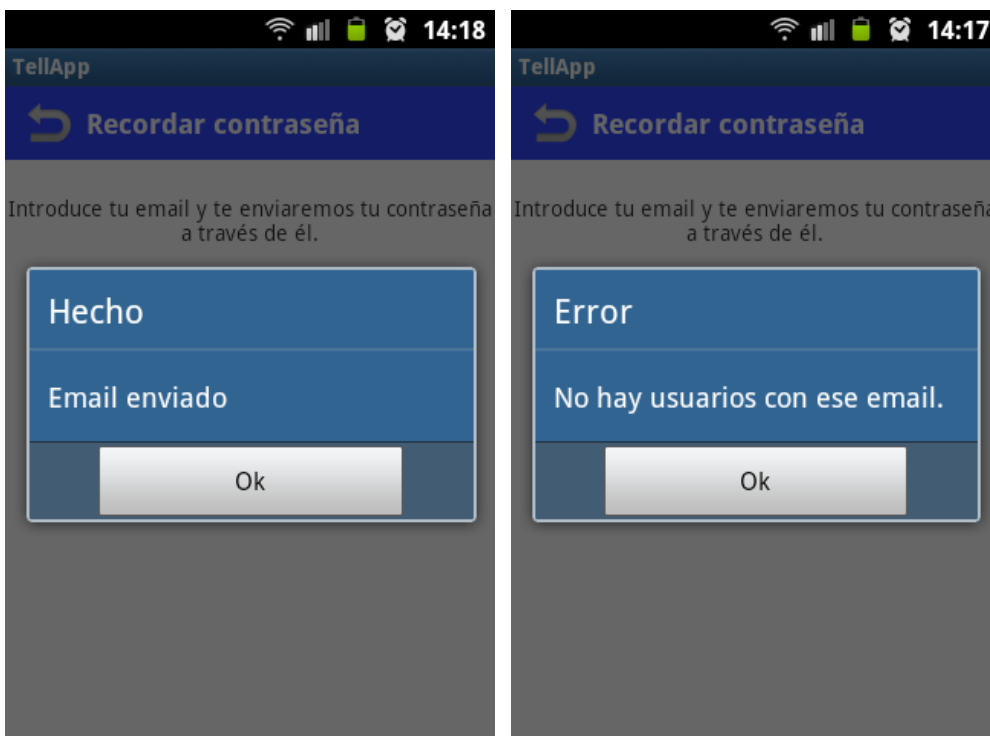
L'adreça de correu existeix al servei i s'envia un mail amb la contrassenya

L'adreça de correu no existeix al servei, i s'informa a l'usuari d'això

No hi ha connexió amb el servidor. S'informa a l'usuari d'aquest fet.



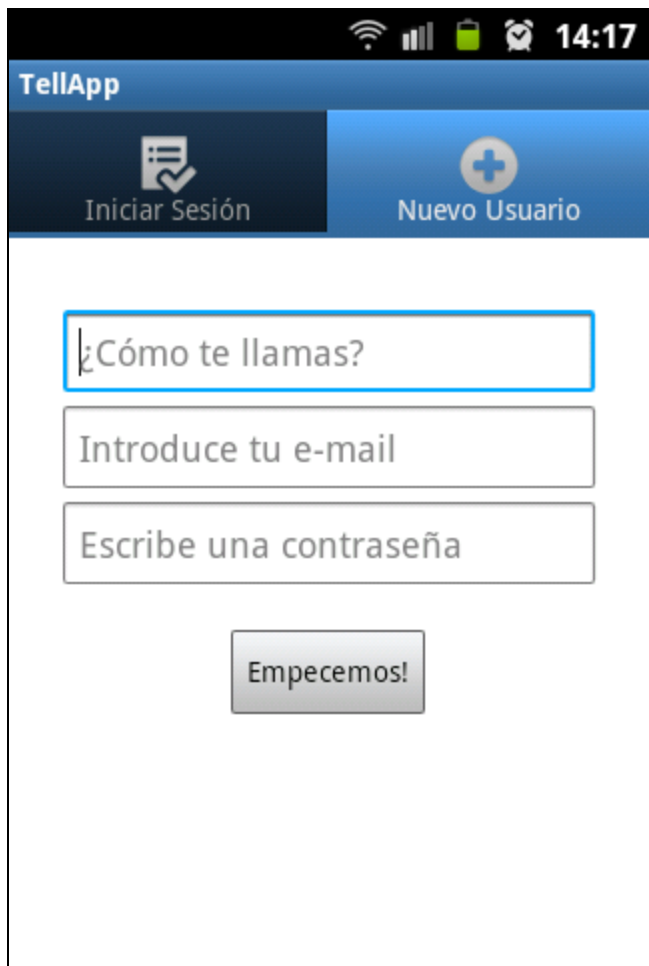
Recuperar contraseña.



9.1.2. Registre d'un nou usuari

Si l'usuari no ha fet servir mai l'aplicació en cap altre dispositiu, no tindrà compte al servei, i per això s'haurà d'enregistrar com a nou usuari d'aquest.

Amb aquest fi, haurà de seleccionar la segona pestanya, *Nou Usuari*, on haurà d'introduir les dades com a nou usuari en el formulari.



The screenshot shows the registration screen of the TellApp. At the top, there is a status bar with icons for Wi-Fi, signal strength, battery, and alarm, along with the time 14:17. Below this is the app header 'TellApp' and two navigation tabs: 'Iniciar Sesión' (log in) and 'Nuevo Usuario' (new user). The 'Nuevo Usuario' tab is selected. The main content area contains three text input fields: '¿Cómo te llamas?' (How do you call yourself?), 'Introduce tu e-mail' (Enter your e-mail), and 'Escribe una contraseña' (Write a password). Below these fields is a button labeled 'Empecemos!' (Let's start!).

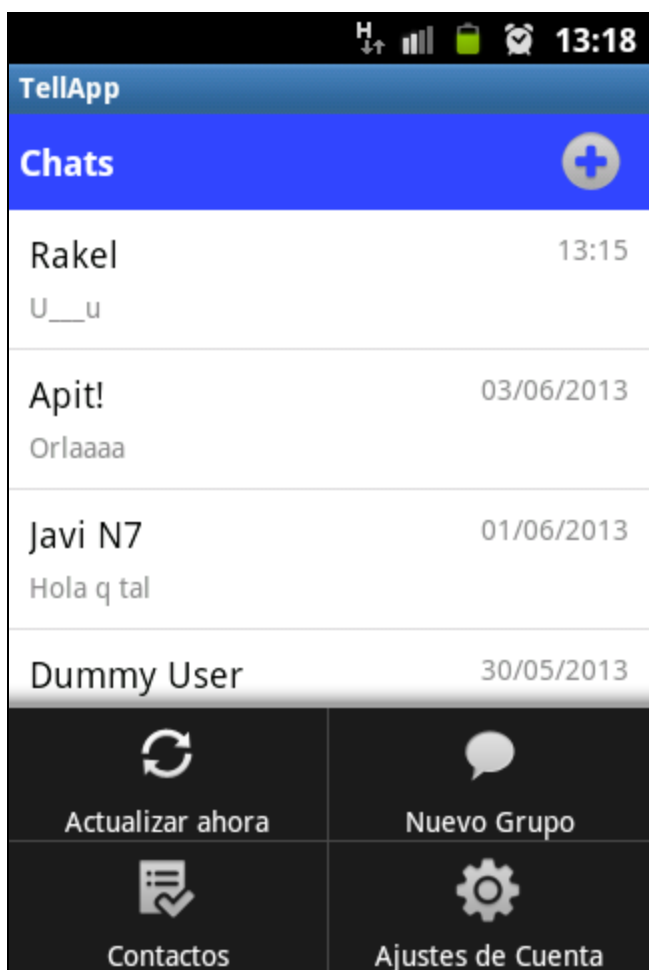
Bàsicament, hi han tres dades a omplir:

- El nom amb el què volem que ens vegin la resta d'usuaris
- L'adreça de correu, que serà la base del nostre registre, i a on es podran enviar les dades si l'usuari perd el seu compte
- La contrasenya, que donarà seguretat al compte de l'usuari; només podrà fer servir el compte si s'enrecorda de la contrasenya.

Un cop creem el compte d'usuari, podem passar a la vista principal de converses.

9.2. Llista de Converses

Un cop l'usuari s'ha autenticat o loguejat a l'aplicació, accedirà a la pantalla de llista de missatges, que al seu cop serà la nova pantalla inicial de l'aplicació. En aquesta pantalla es mostraran totes les converses actives, ordenades per activitat, de més recent a més antiga.



Aquesta llista de converses mostra el títol de la conversa, un extracte del darrer missatge i l'hora o data de recepció d'aquest darrer missatge.

A la barra del títol de la pantalla hi ha una icona que permet crear un nou chat amb un contacte concret. És una mena d'accés ràpid a la mateixa opció de gestió de contactes que apareix al menú contextual.

Al menú contextual disposem de les següents opcions:

Actualitzar ara, que demanarà al servidor tots els missatges que estiguin pendents de rebre. Encara que hi ha un procés en aquest que reenvia els missatges cada deu minuts, per si s'ha perdut cap, si un usuari vol demanar una actualització immediata ho pot fer amb aquesta opció.

Nou Grup. Aquesta opció permet crear una nova conversa de grup amb múltiples contactes. Més endavant ho tractarem, quan parlem de la creació de converses.

Contactes. Aquesta opció porta a la pantalla de gestió de contactes, on podem veure els contactes i afegir-ne de nous. També podem crear converses individuals amb ells.

Ajusts del compte. La pantalla a la que porta aquesta opció ens permet canviar les dades del nostre compte.

A la llista de converses es mostren totes les converses actives, com ja s'ha dit. Fent tap a qualsevol d'aquestes, accedirem a la pantalla de conversa, on veurem els missatges d'aquesta conversa.

També podem esborrar una conversa, prement a sobre fins que ens aparegui el següent missatge:



Esborrant la conversa s'esborren tots els missatges d'aquesta i s'allibera l'espai ocupat a la base de dades.

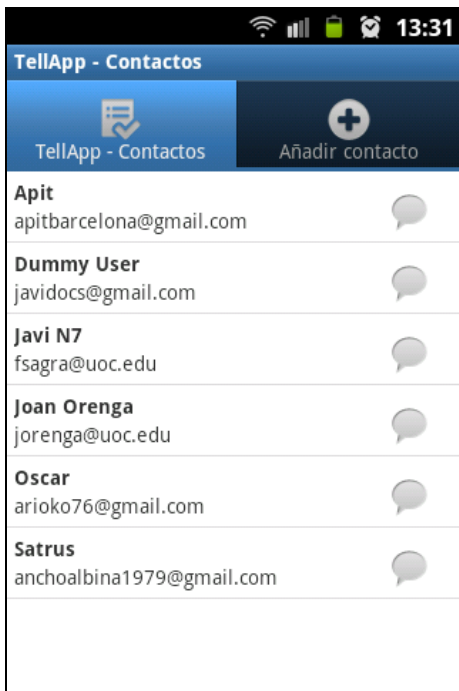
9.3. Gestió de Contactes

Prement a l'opció contextual "Contactes" o a la icona d'accés ràpid de la llista de converses, accedirem a la gestió dels contactes.

Un usuari es comunica amb els seus contactes mitjançant les converses, ja siguin amb un usuari concret o un grup compost per l'usuari i més contactes. Sempre que es rep un missatge d'un altre usuari, aquest s'afegeix com a contacte.

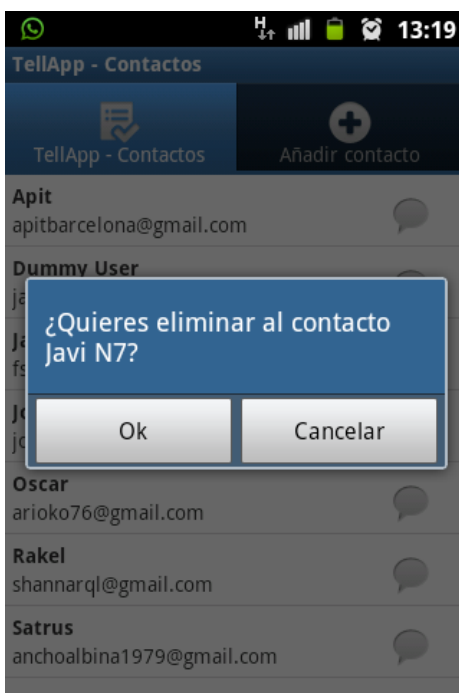
Però un usuari també pot cercar d'altres usuaris, si coneix la seva adreça de correu. En aquest apartat es descriu com cercar usuaris i afegir-los com a contactes.

A continuació es mostra la pantalla principal dels contactes, on es llisten tots aquests ordenats alfabèticament:



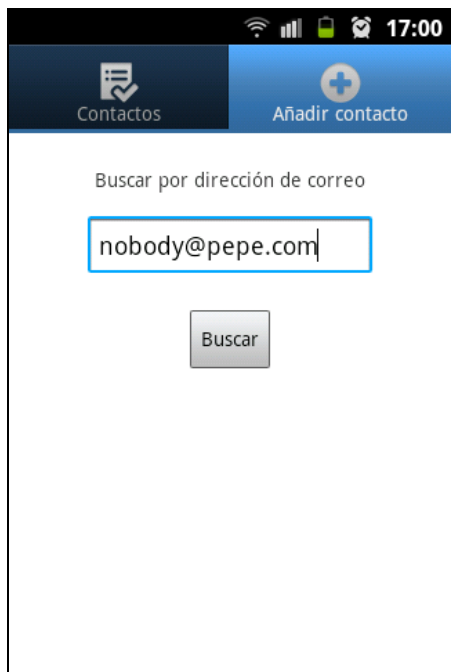
Com es veu a la imatge, la pantalla de gestió de contactes consta de dues pestanyes. La primera mostra tots els contactes que tenim al dispositiu, ordenats alfabèticament, i dels que podem veure tant el nom com l'adreça de correu. A la banda dreta hi ha un globus de text que serveix per a crear una conversa individual amb aquest contacte.

Podem esborrar contactes prement a sobre de qualsevol d'ells fins que surti el següent missatge:

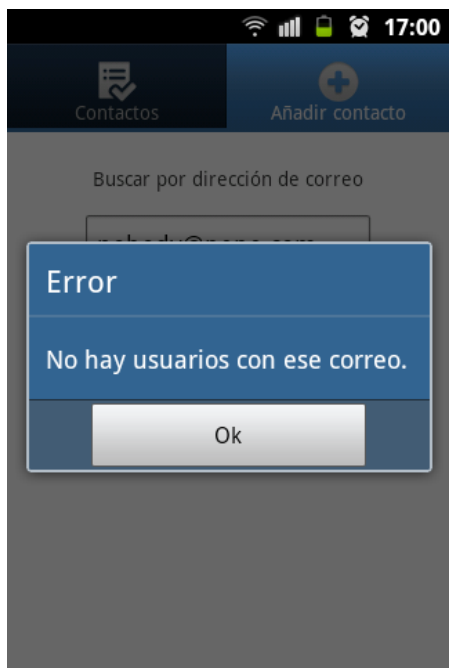


Però només podrem esborrar contactes que no tenen cap missatge emmagatzemat a la base de dades de l'aplicació. És a dir, que per a esborrar un contacte abans haurem d'esborrar les converses a les quals ha participat. Això és necessari de cara a garantir la integritat referencial de les dades.

La segona pestanya, "Afegeix Contacte", s'empra per a cercar nous contactes i afegir-los a la nostra llista de contactes.



El seu funcionament és molt senzill, ja que només s'ha d'escriure l'adreça de correu del contacte que es vulgui afegir, i prémer el botó "Cercar". Si l'usuari existeix, s'afegirà automàticament a la llista de contactes. En cas contrari, s'informarà a l'usuari:



També s'informarà a l'usuari si hi ha cap problema de connexió amb el servidor.

9.4. Creació de converses

Hi han dos tipus de converses possibles: les converses entre dos contactes, i les converses de grup. Ara veiem en detall com podem crear converses d'aquests tipus.

9.4.1. Converses directes entre dos usuaris

Com s'ha vist anteriorment a la gestió dels contactes, la primera pestanya d'aquesta conté la llista dels contactes de l'usuari, i a la dreta de cadascú d'aquests trobem un globus de diàleg. Prenent a sobre d'aquest globus, crearem una conversa individual amb aquest usuari.

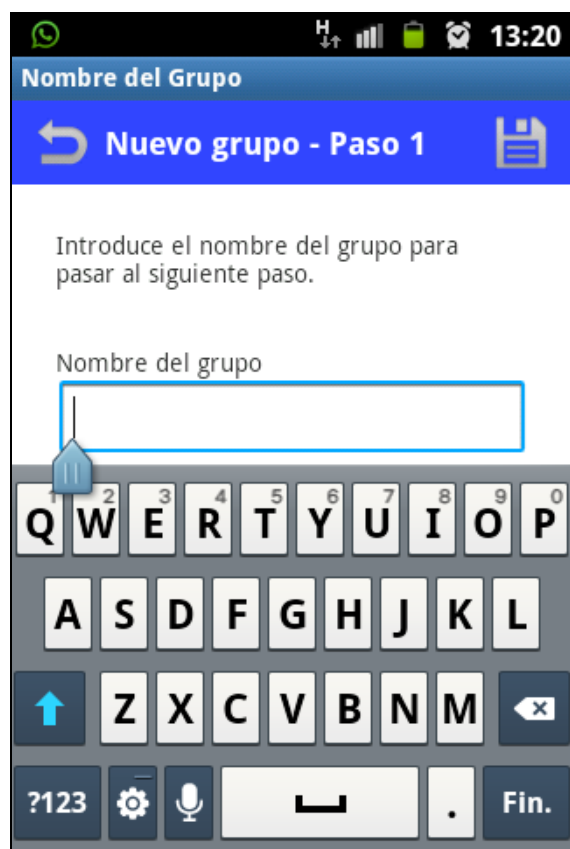
Al crear una conversa amb un usuari accedirem directament a la pantalla d'aquesta.

9.4.2. Converses de grup

Una altra opció de cara a crear converses és emprar les converses de grup, que permeten crear una conversa amb varis contactes a l'hora.

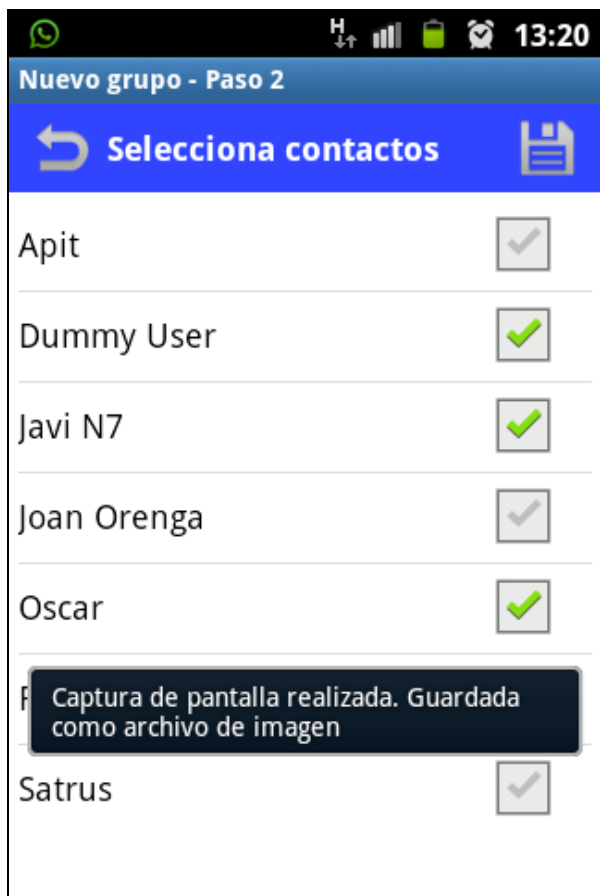
Crearem una conversa de grup des de la pantalla principal de llista de missatges, al menú contextual, opció "Nou Grup".

S'han de seguir dos passos per a crear aquest tipus de converses: el primer pas ens demanarà el nom del grup:



Introduïrem aquest i premerem el botó de guardar (imatge del diskette) per a passar al segon pas. També podem tornar enrera amb la icona de la fletxa situat a la cantonada superior esquerra.

El segon pas serà el següent:



En aquest segon pas, un cop establert el nom del grup, s'hauran de seleccionar els contactes que prendran part en aquest.

Es veu la llista de tots els contactes, amb un checkbox a la dreta. Prement a sobre de cada línia que representa un contacte, activarem/desactivarem aquest a la conversa.

Finalment, prement a sobre la imatge del diskette es crearà el grup i accedirem a la pantalla de la conversa de grup.

Si la creació del grup provoca error, o el servidor no respon, s'informarà a l'usuari i el grup no es crearà.

Un cop creat el grup i enregistrat al servidor, aquest enviarà una notificació a tots els usuaris implicats en aquest. Si cap dels usuaris no tenia algun dels contactes, li seràn afegits en aquest moment.

9.5. Conversa i Missatges

Ja sigui des de la llista de converses prement una qualsevol, o al crear una de nova (sigui amb un altre contacte o un grup d'aquests), accedirem a la pantalla de conversa, on veurem tots els missatges i events que tenen lloc en aquesta.



En aquesta pantalla veurem els missatges d'una conversa concreta. Els missatges del usuari apareixeran centrats a la banda dreta, mentre que els missatges dels altres contactes apareixeran centrats a la banda esquerra amb un to més clar.

De cada missatge veurem qui l'envia, el missatge en sí mateix i l'hora a la que l'ha enviat el servidor.

També es mostraran notificacions, alineades al mig i en un to fosc.



Aquestes notificacions són de tipus:

- **Missatges sense llegir:** Al accedir a una conversa amb missatges que encara no hem llegit.
- **Un usuari canvia el nom del grup:** Quan un usuari edita el nom de la conversa
- **Un usuari s'ha incorporat al grup:** Quan algú afegeix a un usuari en una conversa
- **Un usuari deixa el grup:** Quan un usuari marxa d'un grup

A la part superior esquerra hi ha una icona per tornar enrera, a la llista de converses, mentre que a la part superior dreta apareix una icona d'un llapis per tal d'editar una conversa, però **només si es tracta d'un grup**.

A més, existeix un menú contextual amb una opció per a refrescar els missatges, igual que la que apareixia a la llista de converses, i que demana al servidor els missatges pendents de rebre.

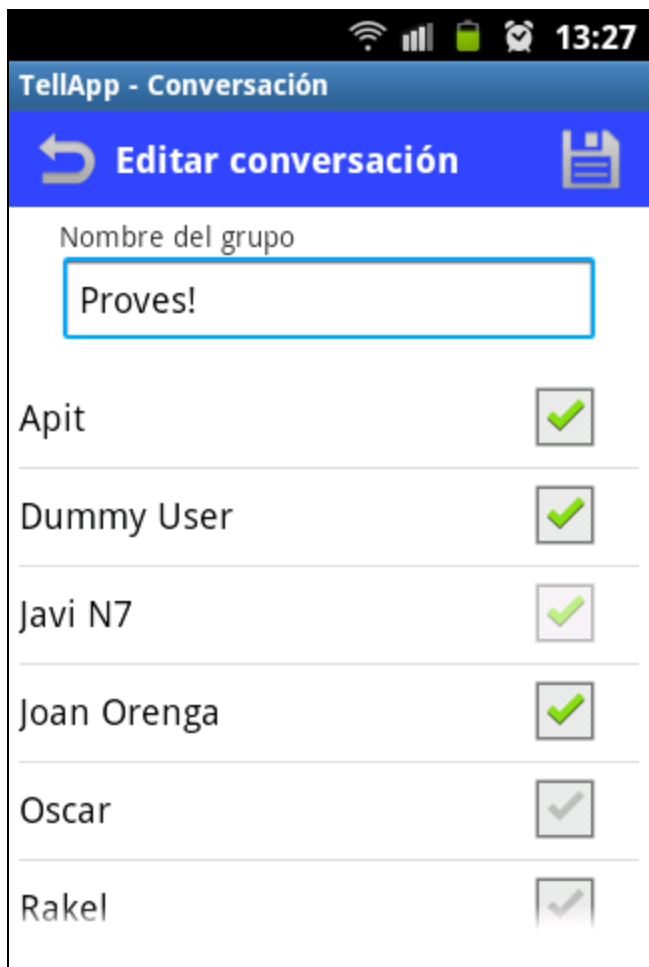


9.5.1. Edició de la Conversa

A la pantalla de conversa, si es tracta d'una conversa de grup, es podran editar les dades d'aquesta. Bàsicament es poden editar les següents dades:

- **Nom de la conversa:** Es pot canviar el nom de la conversa, cosa que farà que es notifiqui a la resta d'usuaris que participen en aquesta

- **Usuaris de la conversa:** Qualsevol usuari que prengui part a la conversa podrà afegir nous contactes per tal que prenguin part en aquesta.



Si un usuari edita una conversa (de grup), podrà afegir més usuaris per tal que prenguin part en aquesta, però **mai podrà treure** cap usuari que ja en prengui part. Per això, els contactes que ja en prenguin part apareixen amb el checkbox marcat, però deshabilitat.

9.6. Ajusts de compte

Finalment, existeix una pantalla d'ajust del compte d'usuari, accessible des del menú contextual de la llista de missatges.

En aquesta pantalla l'usuari pot canviar alguns paràmetres del seu compte d'usuari:

Canviar el seu nom per a mostrar (s'actualitzarà a la resta d'usuaris)

Canviar la seva contrasenya

L'usuari pot canviar (o no) la seva contrasenya, introduint una de nova a tots dos camps destinats a tal finalitat.

Els canvis es desen al clicar sobre la icona del diskette. Si ambdós camps de contrasenya no coincideixen, se li informarà. Si hi ha errors de connexió, també se li informarà.

TellApp - Cuenta

Ajustes de cuenta

Nombre para mostrar

Javi

Dirección de correo

javisasan@gmail.com

Contraseña (en blanco para no cambiar)

Confirmar contraseña

Probar servicio

Borrar cuenta

Aquesta pantalla també té un menú contextual amb un parell d'opcions:

Provar el servei: Un usuari pot provar l'estat del servei amb aquesta icona; si aquest està funcionant, rebrà una notificació.

Esborrar compte: Amb aquesta opció s'esborren totes les dades de l'usuari al dispositiu (dades personals, contactes, conversacions i missatges) i s'allibera el registre amb Google Cloud Messaging.

10. Conclusions

Desenvolupar aquest projecte ha estat una feina llarga i dura, degut a que la meva experiència i coneixement de la plataforma és nul·la. Però un cop enllestit, i veient el què he aconseguit en dos o tres mesos, fa que valgui la pena la inversió d'hores que he realitzat.

Val a dir que la meva experiència web en *back-end* m'ha resultat molt profitosa de cara a plantar cara ben aviat a la tecnologia mòbil, i m'ha permès estalviar hores per a dedicar-les a Android.

Actualment, la programació de dispositius mòbils és una de les feines amb més sortida, i amb aquest projecte he pres una primera toma de contacte d'allò més satisfactòria, que m'obre les portes a seguir profunditzant i encarar la meva carrera en aquest àmbit; el fet que al meu cercle social ja es faci servir l'App per a les necessitats de missatgeria (bàsiques, de moment) no fa més que motivar-me a prendre aquest camí.

10.1. Futures millores fora de l'àmbit del projecte

Com és normal, les fites temporals del projecte han deixat fora d'aquesta multitud de *features* que haurien donat a l'aplicació molta més funcionalitat i, per tant, més valor i professionalitat. Però l'experiència m'ha agradat tant que les implementaré fora de l'entrega del projecte.

Algunes de les possibilitats de cara a ampliar la funcionalitat de l'aplicació són:

- *Disseny*. El disseny de l'aplicació és molt espartà; no hi han imatges, ni icones pròpies. Es podria començar per donar un bon disseny a l'aplicació, que fes gaudir més de l'experiència d'emprar-la. Gran part de l'èxit d'una bona aplicació és degut a una bona experiència d'usuari (*UX*).
- *Encriptació*. En una aplicació com la que estem veient la seguretat és molt necessària; encriptar les dades que s'intercanvia el dispositiu amb el servidor, sobre tot la contrasenya, és crucial. Malauradament, la limitació temporal no ha permès assolir aquest punt. Seria el primer punt a tenir en compte de cara a una ampliació del projecte.
- *Imatges de perfil i de grup*. Seria interessant que l'usuari pogués incloure al seu perfil una imatge, com la resta d'aplicacions de missatgeria, per tal que la interfície no fos tan estèril. La imatge d'una conversa seria el contacte amb el què estem parlant, o si és un grup, la imatge que designem.
- *Imatges, links, vídeos... a les converses*. També seria molt interessant poder pujar imatges a les converses, o vídeos, o enllaços que obrissin el navegador... per tal de poder substituir completament qualsevol altre aplicació de missatgeria. També emoticones!

10.2. Fonts d'informació

Per tal de realitzar aquest projecte, s'han consultat les següents pàgines:

- Android Developer (<http://developer.android.com/develop/index.html>)

- StackOverflow (<http://stackoverflow.com/>)

I els següents articles:

- Creando tareas asíncronas en Android con AsyncTask (<http://miguelangelv.wordpress.com/2011/01/31/creando-tareas-asincronas-en-android-con-asynctask/>)
- Making the most of Google Cloud Messaging (<http://developer.android.com/training/cloudsync/gcm.html>)
- Grails Android GCM Plugin (<http://mindfood.osoco.es/index.php/grails-android-gcm-plugin/>)
- Add a border with rounded corner to a Layout (<http://androidcookbook.com/Recipe.seam?recipeld=2318>)
- Dialogs (<http://developer.android.com/guide/topics/ui/dialogs.html>)
- Nexus 7 no es detectado como dispositivo ADB en Windows (<http://holacodigo.blogspot.com.es/2013/01/nexus-7-no-es-detectado-como.html#.Ua96bUBU9pe>)