



# StatusDroid

Aplicació d'accés a la xarxa StatusNet per Android

**Alumne** Daniel González Santos  
**Consultor** Jordi Ceballos Villach  
**Data** 30/05/2011



## **Agraïments**

Voldria agrair el recolzament que ha suposat per mi tenir a la meva família i als meus amics al costat durant la realització d'aquest projecte.

Segurament hi ha hagut moments que no els he prestat tota l'atenció que es mereixen però m'han demostrat que sempre es pot contar amb ells. Una part d'aquest treball no s'hauria pogut fer sense la seva comprensió.



## Resum

Actualment en el món de les xarxes de computadors i d'Internet en particular es troben en clar ascens un seguit de tecnologies que estan revolucionant la manera de la qual els usuaris interactuen amb la xarxa i entre ells. Aquestes dues tecnologies són les xarxes socials, amb empreses punteres que ofereixen els seus serveis en aquest àmbit com poden ser Facebook o Twitter, o si mirem més la vessant professional LinkedIn, i les plataformes mòbils, entenent per plataforma no només el maquinari sinó tot l'ecosistema de programari i maquinari que conforma una experiència d'usuari consistent en el que s'ha anomenat *smartphones*.

En aquest projecte ens centrarem en la integració d'ambdues tecnologies, com es poden relacionar entre sí i quines són les implicacions d'aquesta integració pel que fa a la utilització i el boom dels dispositius mòbils per part dels usuaris de les xarxes mòbils, boom que està arribant fins i tot a plantejar-se a les operadores la opció d'acabar amb les facilitats d'accés a la xarxa mitjançant la tarifa plana per problemes d'augment desmesurat de tràfic.

Així l'objectiu principal serà realitzar una aplicació per una plataforma mòbil, en el nostre cas aquesta plataforma serà la impulsada per Google i la Open Handset Alliance, Android, que ofereix un entorn de desenvolupament d'aplicacions adient pels objectius que ens hem marcat per aquest projecte.

Pel que fa a la xarxa social utilitzarem StatusNet, que si bé no es de les grans i les més conegudes ens oferirà una sèrie d'avantatges com la disponibilitat del codi al tractar-se d'una aplicació web amb llicència lliure GPL i que ens donarà la possibilitat de muntar un entorn de desenvolupament complet sense dependre d'aplicacions exteriors. A més d'altres avantatges que detallarem a la memòria.

L'abast d'aquest projecte es centra en dos elements d'actualitat molt ambiciosos i la seva integració però sense perdre de vista que el producte principal, tot i els avantatges que hem comentat de disposar d'un entorn de desenvolupament per StatusNet, serà sempre l'aplicació mòbil per a Android.

# Índex

1	Introducció.....	7
1.1	Justificació i context.....	7
1.2	Descripció.....	8
1.3	Objectius.....	9
1.3.1	Instal·lació entorn de desenvolupament.....	9
1.3.2	Desenvolupament API client StatusNet.....	11
1.3.3	Aplicació client.....	12
1.4	Enfocament.....	13
1.5	Planificació.....	14
1.5.1	Cronologia del PFC.....	14
1.5.2	Fites Rellevants.....	16
1.5.3	Diagrames de Gantt.....	18
1.6	Productes.....	20
1.6.1	Servidor StatusNet.....	20
1.6.2	Servei d'implementació de l'API StatusNet.....	21
1.6.3	Aplicació client StatusDroid.....	21
1.7	Entorn de desenvolupament.....	21
2	Tecnologies.....	23
2.1	StatusNet.....	23
2.1.1	Què es el microblogging?.....	23
2.1.2	Característiques principals d'StatusNet.....	24
2.2	Android.....	28
2.2.1	Desenvolupament Android.....	28
2.2.2	Nivells d'APIs.....	29
2.2.3	Estructura d'aplicació Android.....	29
3	Arquitectura.....	32
3.1	Arquitectura del sistema.....	32
3.2	Arquitectura de l'aplicació.....	34
4	Disseny aplicació StatusDroid.....	36
4.1	Casos d'ús.....	36
4.2	Implementació de l'API Client.....	38
4.2.1	Llenguatge de la comunicació.....	40
4.2.2	Funcionament dels Serveis i Fils d'execució.....	40
4.2.3	Tipus de Servei.....	40
4.3	Entorn d'usuari.....	41
4.3.1	Pantalla de configuració principalment.....	41
4.3.2	Pantalla de configuració.....	45
4.3.3	Pantalla principal.....	47
4.3.4	Pantalla d'enviament de missatges.....	51
4.3.5	Pantalla d'informació d'usuari.....	53
4.3.6	Opcions contextuais.....	55
4.4	Persistència de dades.....	57
4.5	Activitats.....	58
4.5.1	Configuració.....	58
4.5.2	Llistat de missatges.....	58



---

4.5.3 Informació d'usuari.....	58
5 Continguts i estructures de l'entrega.....	60
5.1 Instal·lació.....	60
5.1.1 Instal·lació a l'emulador Android.....	61
6 Conclusions.....	66
6.1 Futurs desenvolupaments.....	67
7 Bibliografia.....	69
10 Llicència.....	70



## Índex d'il·lustracions

Imatge 1: Diagrama de Gantt 1 - Tasques.....	19
Imatge 2: Diagrama de Gantt 2 - Tasques.....	19
Imatge 3: Diagrama de Gantt 3 - Planificació.....	20
Imatge 4: Diagrama arquitectura.....	33
Imatge 5: Arquitectura aplicació.....	34
Imatge 6: Diagrama Casos d'Ús.....	37
Imatge 7: Diagrama classes API StatusNet.....	39
Imatge 8: Pantalla de Configuració Principal.....	42
Imatge 9: Pantalla de configuració - Menú contextual.....	44
Imatge 10: Pantalla de configuració.....	46
Imatge 11: Pantalla Principal.....	48
Imatge 12: Pantalla de missatges - Menú d'accions.....	50
Imatge 13: Pantalla d'enviament de missatges.....	52
Imatge 14: Pantalla d'informació d'usuari.....	54
Imatge 15: Menú contextual general.....	56
Imatge 16: Creació del dispositiu virtual.....	62
Imatge 17: Configuració dispositiu virtual.....	63
Imatge 18: Aplicació instal·lada al dispositiu virtual.....	65



# 1 Introducció

## 1.1 Justificació i context

Actualment les xarxes de computadors, i en especial Internet i les diferents formes d'accedir-hi, son un element fonamental per a la transmissió de tota mena d'informació entre els diferents estaments de la societat.

En els últims anys en aquest concepte de xarxes de computadors s'ha estès de forma contundent al món de la mobilitat amb els anomenats *smartphones* que permeten als seus usuaris disposar de diverses i variades aplicacions que fins fa uns anys només estaven disponibles en entorns que requerien uns equipaments més voluminosos que els actuals telèfons mòbils.

Dintre del món de les comunicacions mòbils i les plataformes d'aplicacions per les mateixes existeixen avui en dia diferents alternatives per als usuaris i desenvolupadors. Per una banda tenim el primer dominador del mercat, Symbian, que ja pràcticament està en fase de desaparèixer després de que la companyia Nokia anunciés el seu acord amb Microsoft per desenvolupar els seus telèfons amb Windows Phone.

També tenim aquesta darrera opció que hem mencionat, Windows Phone, que representa l'aposta d'un gran de la informàtica com es Microsoft per reintroduir el seu sistema operatiu en el món dels dispositius mòbils deixant enrera la seva etapa amb Windows Mobile.

Pel que fa a l'entorn professional trobem la principal alternativa a BlackBerry, una empresa dedicada a la fabricació de dispositius mòbils destinats al seu ús en entorns empresarials que ofereix facilitat d'integració amb els sistemes de les empreses com poden ser els servidors de correu.

I finalment ens queden els dominadors actuals del mercat, Apple, amb la seva gama d'iphones àmpliament coneguda i de gran èxit, i que ens ofereix el mercat d'aplicacions més gran que existeix per a aquest tipus de dispositius. I Android, l'aposta de Google per fer-se amb



un tros del mercat de dispositius mòbils i estendre la seva tecnologia dominant de serveis a Internet també en aquests, i que, al igual que Apple, ofereix una botiga d'aplicacions per als seus dispositius.

A més també s'ha produït durant aquest període de temps un augment de l'ús de les xarxes socials, avui en dia existeixen tot tipus de xarxes que són utilitzades a diari per milions de persones com poden ser Facebook, per a la interacció social, LinkedIn, dirigida més als contactes professionals, o Twitter, una plataforma de missatges curts i microblogging, que ofereixen als usuaris la possibilitat d'interaccionar entre ells i compartir la informació que considerin oportú amb els seus coneguts o fins i tot amb la resta del món.

Tenint present els dos conceptes anteriors actualment la societat de la informació s'està movent cada cop més al món de la mobilitat i les xarxes socials, donant especial importància a totes les oportunitats que donen aquests dos elements combinats permetent als usuaris d'ambdues tecnologies estar permanent connectats i comunicant tots aquells aspectes que considerin rellevants a la resta de gent.

## **1.2 Descripció**

El projecte que hem realitzat i que desenvoluparem al següent document és un client pels serveis que ofereix la plataforma de *microblogging* StatusNet que funcioni sobre la plataforma Android. Inicialment s'havia planejat la possibilitat d'incloure l'aspecte de desenvolupament de la part servidor, però es va descartar per limitacions temporals en el desenvolupament del projecte, centrant-nos en el desenvolupament de l'aplicació per a Android,

Per a la part servidor es va pensar en utilitzar una plataforma àmpliament estesa com es Twitter que disposa d'una API per a comunicar-se amb aplicacions externes, però justament es va produir pocs dies després de començar el projecte un canvi en la política de creació de noves aplicacions de Twitter que no recomana crear nous clients cosa que fa menys viable aquesta alternativa ja que requereix registrar un compte de desenvolupador per poder accedir a totes les eines i funcionalitats de Twitter.





Com a alternativa vàlida hem trobat la plataforma StatusNet, la seva finalitat és la creació d'una plataforma de microblogging que funcioni de forma similar a Twitter, de fet per la recerca inicial que hem realitzat hem pogut comprovar que l'API de comunicació que utilitza es compatible amb la de Twitter per facilitar la creació d'aplicacions per la mateixa. A això s'ha d'afegir que es una plataforma de codi obert que utilitzen serveis com [identi.ca](http://www.identi.ca) (<http://www.identi.ca>), per tant ens permetrà instal·lar un entorn de desenvolupament local on fer proves sense cap problema. Partint d'aquestes bases ens va semblar que era una millor opció utilitzar StatusNet i oferir la possibilitat d'accedir a diversos serveis com [identi.ca](http://www.identi.ca), que utilitzar directament Twitter.

En resum, la proposta de projecte final és el desenvolupament d'un sistema client – servidor de comunicació mitjançant missatges curts estil Twitter utilitzant la plataforma StatusNet com a base ja que aquesta ens permetrà instal·lar un servidor per poder realitzar la comunicació i desenvolupament de manera local sense perjudici perquè després la nostra aplicació es pugui connectar i comunicar amb d'altres sistemes que utilitzin aquesta plataforma.

## **1.3 Objectius**

Els objectius del projecte els donarem en dos grans blocs, començarem per definir els objectius generals que ens hem marcat alhora de realitzar aquest projecte per després aprofundir en cadascun d'ells i a quins elements donarem més importància dintre dels mateixos i que són els que prioritzarem durant la realització del projecte.

### ***1.3.1 Instal·lació entorn de desenvolupament***

Com veurem quan expliquem l'arquitectura general del sistema StatusNet és una plataforma de microblogging basada de codi lliure, aquesta característica a diferència d'altres plataformes tancades com pot ser Twitter permet la instal·lació d'un entorn de desenvolupament totalment funcional de manera local per a poder realitzar proves.



Així en una primera fase utilitzarem aquesta avantatge per a instal·lar un servidor d>StatusNet local que ens permetrà realitzar totes les proves de configuració i comunicació que necessitem sense haver de dependre de les funcionalitats ofertades per un servidor de tercers.

Si bé aquesta tasca es durà a terme durant l'evolució del projecte com a eina de desenvolupament s'ha de tenir present que no és un objectiu final del mateix, es a dir, només es un pas intermedi per facilitar la obtenció de l'aplicació final i aprofundir en el coneixement complet del sistema. De la mateixa podrem extreure una visió general del sistema i fer una reflexió sobre el seu funcionament, però no serà un element final en el lliurable del projecte que es centrarà principalment a l'aplicació client.

En aquest objectiu general inclourem els següents objectius particulars que volem assolir amb la seva realització:

- Obtenció i instal·lació del servidor d>StatusNet, en tractar-se d'una plataforma de codi lliure sota llicència AGPLv3 es pot obtenir directament de la seva web el codi de la mateixa i realitzar la seva instal·lació en un entorn local.
- Configuració de l'entorn de desenvolupament. Es configurarà l'entorn de desenvolupament del nostre servidor StatusNet per adaptar-lo a les necessitats del mateix i del nostre desenvolupament.
- Proves de funcionament del servidor. Les proves de funcionalitat es realitzaran de dues maneres, per una banda comprovarem el funcionament de l'entorn web per defecte que ve amb el servidor i per l'altra la comunicació directa amb l'API del servei que és el que utilitzarem a la nostra aplicació.
- Comprovació de la funcionalitat a implementar i el seu correcte funcionament per detectar abans de l'inici de la implementació de l'aplicació client possibles riscos que puguin aparèixer durant la mateixa i intentar evitar-los en la mesura del possible.

Finalitzat aquest punt s'haurà de disposar d'un entorn servidor StatusNet completament funcional i operatiu per a les necessitats de la nostra aplicació client.



### *1.3.2 Desenvolupament API client StatusNet*

Un cop tenim funcionant l'apartat servidor d'StatusNet haurem de realitzar el desenvolupament d'una API client que encapsuli totes les funcionalitats necessàries que volem incloure a la nostra aplicació per tal de facilitar la seva extensibilitat i modularitat de manera independent a la mateixa i no barrejar el propi codi de l'aplicació amb el de l'API el que facilitarà el seu desenvolupament.

Aquest mètode d'implementació modular ens permetrà anar provant amb un cost de prova reduït, ja que no caldrà realitzar una implementació complerta abans de fer les proves, tots els mètodes de l'API que implementem i assegurar el seu correcte funcionament abans de fer ús dels mateixos en etapes més avançades de desenvolupament del projecte.

Les fites que volem assolir durant el desenvolupament d'aquest part del projecte es reflecteixen en el següent llistat:

- Definició de la interfície de l'API per a aquells elements que s'implementaran a l'aplicació client i que requereixen una crida als serveis d'StatusNet.
- Implementació dels mètodes de l'API d'StatusNet que hem definit a la interfície i que es cridaran des de la nostra aplicació.
- Execució de proves contra l'API per comprovar el correcte funcionament d'aquelles crides al servei d'StatusNet implementades i poder assegurar el seu funcionament des de l'API del client de forma satisfactòria.

L'API d'StatusNet està basada en l'API de Twitter per oferir la màxima compatibilitat amb la mateixa, en els mètodes i comunicació bàsics són molt similars i per tant es pot aconseguir cert nivell d'interoperabilitat entre ambdues plataformes.

Això, per altra banda, implica que la complexitat de l'API no fa possible implementar totes les funcionalitats que aquesta ofereix en el marc temporal que tenim disponible per al desenvolupament del nostre projecte i per tant haurem de limitar el número de funcionalitats implementades al client, el llistat complet de les funcionalitats implementades es donarà en



més detall a l'apartat corresponent d'aquest disseny quan hagem aprofundit en l'estructura i les possibilitats de l'API d>StatusNet.

### ***1.3.3 Aplicació client***

L'últim objectiu que es vol assolir amb el desenvolupament d'aquest projecte és el desenvolupament d'una aplicació client que faci ús de les APIs d>StatusNet que hem creat en l'objectiu anterior per connectar-se amb els servidors d>StatusNet.

Es precis comentar en aquest moment que quan parlem de connexió amb els serveis d>StatusNet ens referim a qualsevol dels serveis que utilitzen aquesta eina de microblogging no només el servidor local de desenvolupament que hem creat en el primer objectiu que hem detallat. Això vol dir que la nostra aplicació client, i per extensió la API, haurà de tenir la opció de configurar quin servei es vol utilitzar, per mencionar un dels mes coneguts i amb el qual realitzarem proves d'integració identi.ca, que es un servei de microblogging semblant a Twitter.

El requisit anterior farà necessari disposar de funcionalitats de configuració dins de l'aplicació que permetin als usuaris configurar el servei al qual volen accedir, a més d'una infraestructura per emmagatzemar aquestes dades de configuració de manera que l'usuari no hagi d'introduir-les cada cop que vulgui realitzar una connexió al servei. Com veurem la plataforma Android ja proporciona funcionalitats que cobreixen aquests aspectes de la nostra aplicació i que detallarem en els següents apartats d'aquest mateix document.

Així els objectius específics que volem assolir en aquest objectiu els detallem en el següent llistat:

- El servei StatusNet al qual es connecti l'aplicació ha de ser configurable per l'usuari, això requerirà unes pantalles de configuració i una infraestructura per emmagatzemar les dades que introdueixi l'usuari com la URI del servei o les dades d'usuari.
- L'aplicació ha de donar accés a totes aquelles característiques que definim com a essencials per al servei, aquestes venen definides pel que implementem a l'API client i per tant les detallarem en els següents apartats.



- Òbviament tractant-se d'una aplicació client hi haurà d'existir una interfície gràfica que hem dividit en dos grans apartats:
  - Configuració: Tenim una part de la GUI dedicada a la configuració dels paràmetres de l'aplicació, essencialment serà la configuració dels serveis d>StatusNet.
  - Interacció: Serà la part principal de l'aplicació on es mostrin les línies temporals a les qual estigui subscrit l'usuari així com la possibilitat d'afegir-hi noves subscripcions.

Amb la finalització d'aquest objectiu tindrem disponible ja les tres parts essencials de la nostra aplicació, la part servidor, definida com una instal·lació local del servei d>StatusNet que si bé no es imprescindible ens servirà per a fer proves; l'API client, que definirà tots els elements de l'API StatusNet als quals podrem accedir des de les aplicacions que l'utilitzin; i finalment, la pròpia aplicació client desenvolupada sobre Android que donarà serveis als usuaris que vulguin accedir a servidors d>StatusNet.

## 1.4 Enfocament

La finalitat principal que ens em imposat en aquest projecte és el desenvolupament d'una aplicació per accedir a la plataforma StatusNet a través de dispositius mòbils Android. Inicialment l'abast que volíem assolir era més ambiciós permetent també el desenvolupament de la part servidor però per temes de planificació es va decidir eliminar aquesta part i centrar-nos a la part client.

Tot i aquesta limitació la part servidor es troba representada gràcies a que StatusNet sent una plataforma de codi lliure ens ha permès construir un entorn de desenvolupament local on realitzar proves sense haver-hi d'accedir a entorns exteriors. A més aquesta plataforma té àmplies possibilitats de personalització i creació de modificacions que es podrien utilitzar en el futur per oferir nous serveis per exemple en un entorn com la UOC que volgués desenvolupar serveis addicionals als oferts de base per StatusNet.



Així l'enfocament que hem adoptat ha estat per una banda la implantació d'un entorn de desenvolupament d>StatusNet que ens permeti ser independents de la disponibilitat d'un servei extern i que a més ens permetria l'expansió del servei amb noves funcionalitats si ho consideréssim oportú.

I per l'altra, el desenvolupament de l'aplicació per a dispositius Android dividida en dues parts clarament diferenciades, la primera el servei que implementarà aquelles funcionalitats de l'API d>StatusNet que fem servir i la lògica per utilitzar-les, i la segona la pròpia aplicació client que es el que veurà l'usuari al seu dispositiu quan es vulgui connectar a la plataforma.

## 1.5 Planificació

### 1.5.1 Cronologia del PFC

Hem dividit el desenvolupament del projecte en diferents fases per intentar ajustar les mateixes a les diferents PACs que hem de realitzar, així la primera PAC comprèn aquest document de planificació, la segona estarà centrada en definir el disseny de tots els elements del projecte, la tercera serà la que comprendrà la PAC 3 i serà en la qual es realitzarà la implementació i les proves de l'aplicació, i finalment tenim una fase on es realitzarà la memòria i la presentació del PFC.

- **Especificació**

Aquesta fase conclou amb la presentació d'aquest document, a la mateixa hem realitzat la proposta de projecte i l'especificació inicial del seu abast i objectius que ens plantegem aconseguir amb el seu desenvolupament. En finalitzar l'especificació es considera acabada i entregada la PAC 1.

- **Formació**

Per a la realització del projecte farem ús de plataformes i tecnologies que no hem utilitzat al llarg de la carrera, com pot ser el desenvolupament d'aplicacions per a Android o l'estudi de les possibilitats i formes de desenvolupar per a StatusNet.



- **Anàlisi i Disseny**

La fase de anàlisi i disseny compren tant l'estudi de com es realitzarà el desenvolupament i quines parts de les tecnologies emprarem com el disseny tècnic d'aquells elements que formaran part del nostre projecte per poder utilitzar aquest disseny com a base per a les següents fases de la implementació. El resultat d'aquesta fase serà el document de disseny que s'entregarà al finalitzar la PAC 2.

- **Implementació**

Finalitzat el disseny la següent fase inclou la implementació gradual dels diferents elements del disseny per a realitzar el nostre projecte, començarem per la instal·lació del servidor en local per a utilitzar-lo com a base pel desenvolupament dels altres elements, seguirem amb la implementació de la comunicació entre el client Android i el servidor, per afegir després la interfície gràfica que finalitzi l'execució del projecte.

- **Proves**

Aquesta fase es realitza de forma paral·lela a l'anterior, es realitzaran proves unitàries dels diferents elements que conformen el nostre projecte per un cop superades totes integrar tots els elements i realitzar les proves de verificació finals que incloguin tot el conjunt i que seran el pas previ a l'entrega final de la PAC 3 que inclourà tota la implementació finalitzada.

- **Tasques finals**

La última fase serà la fase final de documentació i presentació del projecte es realitzarà un cop tancada i entregada la implementació a la fase anterior i consistirà en ajuntar i completar la documentació que hem anat generant durant les altres fases de desenvolupament del projecte per arribar a tenir un producte final. A més aquesta fase també inclourà el vídeo de presentació del projecte.

El cicle de vida aquí descrit es un cicle iteratiu e incremental per la qual cosa hi ha diverses fases que es realitzen de forma conjunta durant tot el procés i fases posteriors poden afectar a fases anteriors, per exemples modificacions en la implementació per errors a la fase de proves.



## 1.5.2 Fites Rellevants

Les fites més importants que hem definit al PFC es poden dividir en dos grups, aquelles relacionades amb el procés de creació del programari esmentat en el punt anterior, com pot ser la creació del servei que implementa l'API StatusNet, i aquelles que tenen a veure amb la temporalització d'entregues previstes al llarg del semestre formant les PACs i la memòria i presentació final del TFC. Resumint tot això en una llista les dates més rellevants son les enumerades a continuació:

- **Fita 1 – Planificació Inicial:** La primera fita important es la que conclou amb la presentació de la planificació complerta pel PFC. Es la presa de contacte inicial amb el PFC, els conceptes sobre els quals aquest tractarà i la realització d'una planificació de tot el treball que es durà a terme durant el curs. El resultat d'aquesta primera fita serà el l'entrega de la PAC1, on s'inclourà una primera informació sobre les tecnologies del projecte en les quals s'aprofundirà al llarg de tot el curs, i el document de planificació. (14/03/2011)
- **Fita 2 – Definició arquitectura:** La segona fita s'assolirà amb la definició de l'arquitectura bàsica sobre la qual funcionarà la nostra aplicació, això inclourà l'elecció de les tecnologies específiques, servidor, base de dades, entorn de desenvolupament, ..., així com una primera prova del funcionament d'aquestes. Al final d'aquesta fase es tindrà un document que explicarà les raons de l'elecció d'aquesta arquitectura i com ens ajudarà a implementar i solucionar els diversos aspectes que hem de tractar al PFC. (24/03/2011)
- **Fita 3 – Anàlisi i Disseny:** La següent fita rellevant torna a tenir relació amb el cicle de vida del programari, si en el punt anterior havien definit una arquitectura sobre la qual funcionaria aquest, i les tecnologies que farem servir en aquest apartat definirem la part funcional de l'aplicació. En un procés de creació normal





aquest apartat hauria de ser previ a l'elecció de tecnologies, per poder fer una selecció més adient, però atenent les limitacions temporals del projecte s'ha preferit començar també a muntar una arquitectura sobre la que realitzar el desenvolupament tot i no tenir l'anàlisi complert. Per això es bo comentar que aquesta fita i l'anterior no son excloents i es realitza en certa manera de forma paral·lela. Aquesta fita conclourà amb el lliurament del document de disseny que es correspon amb la PAC 2 de l'assignatura. (04/04/2011)

- **Fita 4 – Implementació servidor:** Aquesta fita es la primera que té a veure amb la implementació pròpia del projecte de programari que es vol realitzar com a producte a entregar amb el PFC. Consta de la instal·lació de l'entorn servidor local d>StatusNet que farem servir com a base de desenvolupament per a la nostra aplicació Android. Com a resultat d'aquesta fase es disposarà de l'entorn StatusNet operatiu en un sistema local. (07/04/2009)
- **Fita 5 – Implementació API StatusNet:** La següent fita que ens hem proposat a la realització del projecte és la implementació del servei que implementarà les funcionalitats de l'API StatusNet que farà servir la nostra aplicació. Aquesta implementació ja formarà part del lliurament del producte final i per tant haurà de passar un joc de proves exhaustiu per assegurar el seu correcte funcionament, finalitzada aquesta fita disposarem d'un servei amb possibilitat d'accés al nostre servidor StatusNet. (18/04/2011)
- **Fita 6 – Implementació aplicació client:** En aquesta fita continuarem amb la part d'implementació del programari, en aquest cas, i un cop ja tenim enllestida la part de comunicació amb el servidor, procedim a la implementació de la part client de l'aplicació que ens permetrà accedir als serveis de la plataforma StatusNet a través d'una aplicació Android que faci ús del servei que hem



desenvolupat a la fita anterior. Al final d'aquesta fita es disposarà d'una primera versió de l'aplicació StatusDroid integrada amb la resta del sistema i que s'anirà millorant en futures versions, aquesta juntament amb el treball de la fita anterior es el que definirà l'entrega de la PAC3. (09/05/2011)

- **Fita 7 – Proves i verificació:** Un cop tancada la implementació passem a la fase de prova i verificació del correcte funcionament de l'aplicació, aquesta fase, al igual que l'anterior, ja que es desenvolupa en paral·lel, conclourà amb el lliurament de la PAC 3. (09/05/2011)
- **Fita 8 – Tasques Finals:** Aquesta es la última fita del projecte, on s'ajunten tots els documents elaborats fins a aquest moment per realitzar la memòria del PFC i es complementa amb els aspectes finals de la seva realització. També es prepara una presentació del funcionament i funcionalitat de l'aplicació. Els documents finals d'aquesta fase son la memòria i la presentació de l'entrega del PFC. (30/05/2011)
- **Fita complementaria – Formació:** Una última fita que es durà a terme paral·lelament a totes les altres es la de la formació en les diverses tecnologies que s'aplicaran al projecte. Aquesta fita està programada sobretot per realitzar-se al principi del projecte, però serà continuada durant tot aquest.

### 1.5.3 Diagrames de Gantt

Per acabar amb la planificació detallarem els diagrames de Gantt utilitzats i que detallen la planificació complerta que s'ajusta a les fites anteriorment esmentades i que inclou el detall de les tasques que es realitzaran en cadascuna de les fites i fases del desenvolupament del projecte.

#### Tasques



	Nombre	Duració	Inicio	Terminado	Predecessores	Nombres del Recurso
1	<b>E</b> ESPECIFICACIÓ	11 days	3/03/11 8:00	13/03/11 17:00		
2	Lectura i Valoració TFC	2 days	3/03/11 8:00	4/03/11 17:00		
3	Recerca d'informació	7 days	4/03/11 8:00	10/03/11 17:00		
4	Planificació	3 days	11/03/11 8:00	13/03/11 17:00		
5	Lliurament PAC1	1 day	14/03/11 8:00	14/03/11 17:00	1	
6	<b>E</b> FORMACIÓ	18 days	15/03/11 8:00	1/04/11 17:00		
7	Formació arquitectura	6 days	15/03/11 8:00	20/03/11 17:00		
8	Formació Android	7 days	21/03/11 8:00	27/03/11 17:00		
9	Formació StatusNet	5 days	28/03/11 8:00	1/04/11 17:00		
10	<b>E</b> ANÀLISI I DISSENY	21 days	14/03/11 8:00	3/04/11 17:00		
11	Especificació casos d'ús	7 days	14/03/11 8:00	20/03/11 17:00		
12	Disseny arquitectura programari	7 days	18/03/11 8:00	24/03/11 17:00		
13	Disseny GUI de l'aplicació	3 days	25/03/11 8:00	27/03/11 17:00		
14	Definició missatgeria	5 days	28/03/11 8:00	1/04/11 17:00		
15	Documentació Anàlisi i Disseny	2 days	2/04/11 8:00	3/04/11 17:00		
16	Lliurament PAC 2	1 day	4/04/11 8:00	4/04/11 17:00	6;10	
17	<b>E</b> IMPLEMENTACIÓ	34 days?	5/04/11 7:00	8/05/11 17:00		
18	Instal·lació servidor	3 days?	5/04/11 7:00	7/04/11 17:00		
19	Implementació API StatusNet	12 days?	7/04/11 7:00	18/04/11 17:00		
20	Comunicació client-servidor	6 days?	19/04/11 7:00	24/04/11 17:00		
21	<b>E</b> Implementació GUI	8 days?	25/04/11 7:00	2/05/11 17:00		
22	Configuració	3 days?	25/04/11 7:00	27/04/11 17:00		
23	Client	5 days?	28/04/11 7:00	2/05/11 17:00		
24	Integració de components	4 days?	3/05/11 7:00	6/05/11 17:00		
25	Documentació implementació	2 days?	7/05/11 7:00	8/05/11 17:00		
26	<b>E</b> PROVES	33 days?	6/04/11 7:00	8/05/11 17:00		
27	<b>E</b> Proves unitàries	26 days?	6/04/11 7:00	1/05/11 17:00		
28	Proves funcionament servidor	3 days?	6/04/11 7:00	8/04/11 17:00		
29	Proves API StatusNet	4 days?	15/04/11 7:00	18/04/11 17:00		
30	Proves comunicació	3 days?	22/04/11 7:00	24/04/11 17:00		
31	Proves funcionalitat GUI	3 days?	29/04/11 7:00	1/05/11 17:00		
32	Proves integrades	3 days?	6/05/11 7:00	8/05/11 17:00		
33	Documentació proves	2 days	7/05/11 7:00	8/05/11 17:00		
34	Lliurament PAC3	1 day?	9/05/11 8:00	9/05/11 17:00	17;26	

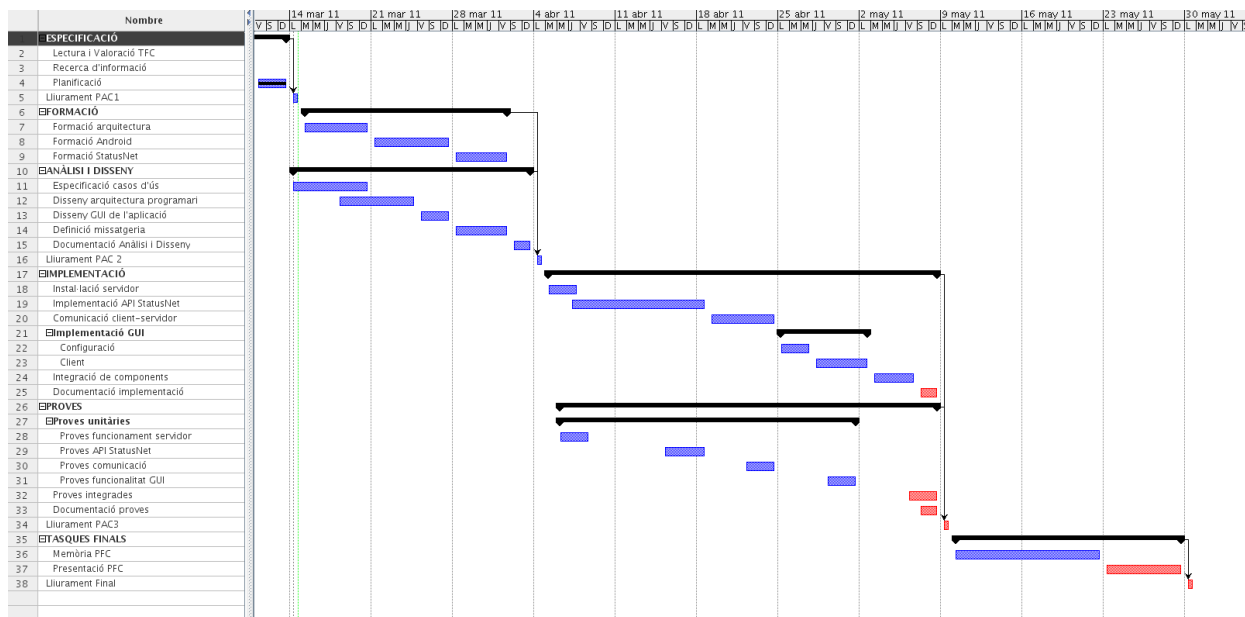
Planificació PFC - pàgina1

Imatge 1: Diagrama de Gantt 1 - Tasques

	Nombre	Duració	Inicio	Terminado	Predecessores	Nombres del Recurso
35	<b>E</b> TASQUES FINALS	20 days?	10/05/11 8:00	29/05/11 17:00		
36	Memòria PFC	13 days?	10/05/11 8:00	22/05/11 17:00		
37	Presentació PFC	7 days?	23/05/11 8:00	29/05/11 17:00		
38	Lliurament Final	1 day?	30/05/11 8:00	30/05/11 17:00	35	

Imatge 2: Diagrama de Gantt 2 - Tasques

## Diagrama



Imatge 3: Diagrama de Gantt 3 - Planificació

## 1.6 Productes

Tal i com s'ha esmentat en apartats anteriors el projecte pretén aproximar-se al desenvolupament d'una aplicació Android on intervé un client, una API que ofereix les funcionalitats d'accés al servei StatusNet, i una part de servidor. A partir d'aquests elements aquests són els productes que seran resultat del desenvolupament del projecte

### 1.6.1 Servidor StatusNet

El primer producte que obtindrem serà un servidor StatusNet completament funcional, els avantatges d'aquesta implementació seran no haver de disposar d'un servei extern per realitzar les proves d'integració el que sempre pot facilitar tot el procés de prova i verificació de les diferents funcionalitats sense necessitat de dependre d'un tercer.

Aquest producte no s'inclourà com a lliurable ja que la plataforma StatusNet es una plataforma de codi lliure disponible a la mateixa web del producte i per tant no té sentit incloure aquest en el projecte, però si que detallarem com s'ha realitzat la implementació i la seva arquitectura perquè és una part rellevant del desenvolupament del nostre projecte.



### *1.6.2 Servei d'implementació de l'API StatusNet*

La primera aplicació que s'inclou al lliurament definitiu d'aquest projecte és el servei Android que implementa l'API de comunicació amb els servidors d>StatusNet. En aquest servei s'inclouran totes aquelles funcionalitats de les quals faci ús la nostra aplicació.

El fet de separar aquest producte de l'altre, encara que en el lliurament final es troben empaquetats a la mateixa aplicació per facilitar el seu ús, és la possibilitat que dona Android d'implementar serveis que puguin ser utilitzats per més d'una aplicació de manera simultània, més endavant en aquesta memòria detallarem el perquè d'aquesta decisió i les seves implicacions en el desenvolupament i futures ampliacions.

### *1.6.3 Aplicació client StatusDroid*

L'últim producte que es desenvoluparà durant aquest projecte serà la part visible del mateix i que consistirà en l'aplicació client d>StatusNet que hem anomenat StatusDroid. Aquesta aplicació és la que oferirà les opcions d'accés a les funcionalitats d>StatusNet a través del servei que hem comentat a l'apartat anterior.

Als següents apartats de la memòria detallarem la seva estructura i les decisions de disseny que hem pres per a la seva implementació.

## **1.7 Entorn de desenvolupament**

---

L'entorn de desenvolupament que s'ha utilitzat alhora de realitzar aquest projecte consta dels següents elements més rellevants, a més d'altres de suport genèrics que no detallarem perquè no són part fonamental de l'entorn.

- **Eclipse IDE:** L'entorn de desenvolupament integrat recomanat per a desenvolupar aplicacions sobre Google està basat en Eclipse, un dels IDE més utilitzats en entorns Java.



A aquest s'han afegit els plugins necessaris subministrats per Google per a poder treballar directament amb l'estructura d'aplicacions adient per a Android i amb l'emulador de la plataforma.

- **Android SDK:** El kit de desenvolupament d'Android que aporta les eines necessàries per a treballar amb els paquets d'Android, des de la creació del projecte fins a l'execució del mateix en un emulador de la plataforma.
- **LAMP:** LAMP són les sigles de Linux, Apache, MySQL i PHP, i formen el servidor web que farem servir com a base per a la instal·lació del servidor local de StatusNet que emprarem al desenvolupament de la pràctica.
- **Git:** Sistema de control de versions distribuït que utilitzarem com a repositori de la informació del projecte durant el desenvolupament del mateix. Aquesta eina ens permetrà accedir sempre a la informació més actualitzada des de qualsevol lloc on tinguem accés al repositori i per tant poder treballar en diverses localitzacions.



## 2 Tecnologies

Abans d'arribar a aquest punt hem comentat un seguit de tecnologies que utilitzaríem durant el projecte, com StatusNet o Android, sense donar detalls específics sobre el que són i quines funcionalitats proporcionen, en aquest apartat comentarem una mica en que consisteixen aquestes tecnologies i de quina manera seran utilitzades pel nostre projecte

Com ja hem comentat les dues tecnologies rellevants pel projecte seran StatusNet, un servei de microblogging de codi lliure i que s'utilitza en aplicacions web con identi.ca, i Android, el sistema operatiu per smartphones desenvolupat per Google y que està auspiciat per la Open Handset Alliance formada per diversos fabricants de hardware, software i proveïdors de serveis.

### 2.1 StatusNet

StatusNet és una plataforma de microblogging de codi lliure sota llicència AGPLv3, el cas comercial més rellevant semblant a aquest plataforma, salvant les distàncies que podem trobar, seria com ja hem comentat en anteriors punts d'aquest document Twitter. L'objectiu d'ambdues plataformes es oferir serveis semblants als seus usuaris que componen eines de microblogging i agregació d'aquests per facilitar l'experiència d'usuari.

#### 2.1.1 Què es el microblogging?

Amb el boom de Twitter tothom sap més o menys que és el microblogging, StatusNet és una altra plataforma que ofereix aquests serveis. Bàsicament podríem definir microblogging com un servei de missatges curts per afavorir la comunicació d'informació de manera concisa i directa. En el cas dels serveis més coneguts aquest límit es troba en 140 caràcters, tal i com podem trobar en serveis com identi.ca, bàsicament per mantenir compatibilitat amb Twitter ja que no hi ha cap limitació pel que fa a la tecnologia.

Al voltant d'aquesta senzilla idea ha evolucionat el concepte de microblogging, passant d'un simple servei d'intercanvi de missatges curts a oferir sobre aquesta funcionalitat d'altres serveis



com geolocalització, transmissió d'imatges o fins i tot audio o video, a més de serveis perifèrics com l'acortament d'urls.

Aquesta diversitat ha fet que les APIs associades a aquests serveis siguin cada cop més complexes i ofereixin més funcionalitats, el que dona grans possibilitats d'interacció a l'usuari.

### *2.1.2 Característiques principals d'StatusNet*

StatusNet té moltes semblances amb serveis com Twitter la seva base es la mateixa, ser una plataforma de microblogging. A continuació explicarem les característiques que considerem més rellevants d'aquesta plataforma tot i que no totes elles es veuran reflectides a la nostra aplicació per temes de limitació de temps.

#### *Codi lliure*

StatusNet es una aplicació servidor desenvolupat sota la llicència AGPLv3 (Aferro GNU Public License) dissenyada especialment per a la redistribució d'aplicacions que donen els seus serveis a través de xarxes com és el cas que ens ocupa amb StatusNet.

Aquest fet suposa molts avantatges de cara a l'extensibilitat i personalització de la plataforma i també pel que fa al desenvolupament d'aplicacions ja que permet disposar d'un entorn de desenvolupament local que ens permetrà realitzar proves exhaustives sense dependre de les polítiques i restriccions de tercers.

A més, encara que no s'utilitzi concretament en aquest projecte, permet la realització de modificacions de la plataforma per adaptar-la a les característiques que es puguin necessitar en determinats desplegaments de la mateixa i que podrien no estar disponibles a l'aplicació base proporcionada per StatusNet.

#### *Sistema federat*

Una de les característiques més rellevants des del punt de vista del servei StatusNet i que la diferencia d'altres serveis similars tancats i centralitzats és que es tracta d'un sistema federat, això implica que les diferents implantacions que existeixen d'StatusNet es poden comunicar





entre elles per intercanviar informació i dades sempre que els administradors i els usuaris així o acordin.

Ser un sistema federat suposa que existirà una sincronització entre les diferents implantacions que existeixin del mateix sempre i quan els administradors així ho considerin necessari, això implica que per una banda l'usuari tindrà un punt d'entrada comú encara que utilitzi més d'un servei basat en la mateixa plataforma, [identi.ca](http://identi.ca) està configurada de manera que permet la federació. I per altra banda si un dels serveis al quals es connecta l'usuari no estigues disponible per qualsevol causa aquest encara podria continuar utilitzant els serveis de la plataforma conjunta i en el moment que el servei caigut es reconnecti al sistema actualitzarà totes les dades com si mai hagués estat caigut.

Tot i aquestes avantatges el sistema federat també representa inconvenients que compliquen l'estructura del sistema, aquesta estructura fa que es requereixi un protocol de sincronització entre els diferents nodes que implementin el servei d>StatusNet per intercanviar la informació, i per la part que ens afecta a nosaltres un servei federat implica que poden existir il·limitats proveïdors del mateix i per tant requerirà que l'usuari configuri a quins d'ells vol accedir, a més aquests proveïdors no tenen perquè oferir els mateixos nivells de servei i per tant necessitem un sistema de descobriment de funcionalitats, sistema que proporciona StatusNet.

### *Interacció amb StatusNet*

El servei d>StatusNet dona als usuaris diverses formes de comunicació per tal d'accedir a les funcionalitats que ofereix, entre elles trobem les següents:

- Aplicació web: És la forma d'accés bàsic que es proporciona per defecte a les instal·lacions d>StatusNet, una aplicació en entorn web a la qual es pot accedir des de qualsevol navegador.
- Correu electrònic: StatusNet permet que els usuaris enviïn els canvis d'estat a través de missatges de correu electrònic.



- SMS: Semblant al cas anterior però fent ús dels missatges SMS que podem trobar a qualsevol mòbil.
- XMPP/Jabber: Protocol estàndar de comunicació per sistemes de missatgeria instantània, utilitzat en serveis com Google Talk, permet actualitzar les dades a StatusNet a través d'un client de missatgeria instantània que suporti aquest protocol.
- Aplicacions de tercers: Altres aplicacions de tercers que utilitzin l'API pública d>StatusNet per interactuar amb el servei.

Tot i que el servei d>StatusNet està capacitat per oferir totes les formes de comunicació que hem comentat en els punts anteriors en el nostre projecte farem servir la última opció ja que la nostra aplicació implementarà una API client en un entorn Android per comunicar-se amb el servei d>StatusNet.

### *Extensibilitat*

La plataforma StatusNet proporciona grans possibilitats d'extensibilitat, disposa d'un sistema definit de temes i plugins que permeten als desenvolupadors afegir funcionalitats que no siguin presents a la instal·lació base d>StatusNet sense gaire complicació ampliant les possibilitats de la plataforma.

Adicionalment i en tractar-se d'una plataforma de codi lliure qualsevol persona pot modificar el codi per afegir-hi modificacions que vagin més enllà que el que es pot realitzar amb el framework de plugins ofert pel servei. En el nostre cas aquesta funcionalitat no serà d'utilitat ja que només utilitzarem l'entorn bàsic com a base de desenvolupament.

### *Arquitectura*

Sense entrar en detalls específics i concrets de l'arquitectura d>StatusNet ja que aquests no formen part d'aquest projecte i el seu disseny si que mencionarem la base sobre la qual està construïda la plataforma i de quina manera la implantarem.

Bàsicament StatusNet es tracta d'una aplicació web construïda en PHP i que utilitza com a sistema d'emmagatzamament per defecte una base de dades MySQL com fan multitud



d'aplicacions web, i per tant els seus requisits es poden trobar en multitud de sistemes i serveis d'allotjament online sense cap mena de problema.

En el nostre cas el servidor de proves estarà desplegat en un entorn LAMP què és l'idoni i el recomanat per la guia d'instal·lació d>StatusNet, això implica un servidor Apache corrent en un entorn Linux amb MySQL com a base de dades y PHP com a llenguatge, qualsevol distribució Linux servirà en aquest aspecte.

### *StatusNet API*

Una part fonamental del servei StatusNet és l'API publica que permet la comunicació des d'aplicacions externes, aquesta API està basada en l'API de Twitter per permetre compatibilitat amb aquesta altra xarxa de microblogging i facilitar la migració d'aplicacions que utilitzessin ja aquest sistema per a comunicar-se per tant és un sistema ja provat.

De totes les funcionalitats d'aquesta API els elements que implementarem seran els següents:

- **Configuració:** Aquesta part de l'API permet recuperar les dades de configuració del servidor, la utilitzarem per recuperar la informació de funcionament del mateix com el màxim número de caràcters que admet el missatge.
- **Autenticació:** Per fer ús de l'API de StatusNet es poden fer servir diversos mètodes d'autenticació, bàsica per HTTP, protegida amb SSL sobre HTTPS i OAuth, d'aquests tres mètodes implementarem els dos primers.
- **Línia temporal:** L'API proporciona diversos mètodes per recuperar un conjunt de línies temporals, des de la pública visible per tots els usuaris fins a aquelles associades a les identitats que segueix l'usuari.
- **Comptes:** Secció de l'API d>StatusNet que permet canviar les dades de la compte d'usuari associada a la sessió en curs, implementarem la funció bàsica de consulta de comptes.



---

## 2.2 Android

---

L'altra tecnologia rellevant que serà utilitzada com a base de la implementació de la nostra aplicació serà Android. Android és un sistema operatiu basat en Linux i desenvolupat per Google, ara sota el control de la Open Handset Alliance formada a més de per Google per altres companyies de software, venedors de hardware i operadores de telecomunicacions que inverteixen en el desenvolupament del projecte.

El sistema operatiu Android és, com StatusNet un projecte de codi lliure, en aquest cas amb doble llicència GPL pel que fa al kernel Linux i la llicència Apache per a la resta de complements públics del sistema. A més sobre aquest sistema s'han desenvolupat un seguit d'aplicacions que no necessàriament són de codi lliure com pot ser el propi Android Marketplace.

### 2.2.1 *Desenvolupament Android*

El desenvolupament d'aplicacions sobre Android es totalment lliure utilitzant l'Android SDK que es pot descarregar de la pròpia web d'Android. Aquest kit de desenvolupament permet generar l'estructura de projectes i els paquets finals per a desplegar les aplicacions als sistemes finals, a més proporciona opcions de virtualització de diferents dispositius permetent provar el funcionament de les aplicacions sense haver de realitzar el desplegament en cadascun d'ells.

L'entorn de desenvolupament a més d'utilitzar l'SDK es complementa amb l'utilització d'Eclipse integrant-se el primer amb el segon a través d'un plugin que permet controlar les diferents funcionalitats que ofereix l'SDK des de l'Eclipse permetent treballar de manera integrada de forma senzilla i eficient.

El llenguatge sobre el qual es desenvolupen les aplicacions Android és el llenguatge Java tot i que es compilen a un bytecode diferent del que utilitza la màquina virtual estàndard d'Oracle i que s'executa sobre una màquina virtual diferent anomenada Dalvik. Aquesta màquina virtual està basada en una altra màquina virtual, Harmony, desenvolupada per Apache i no es



compatible amb el codi Java estàndar, d'aquí la necessitat de comptar amb un SDK que pugui generar el codi adient i no el compilador estàndard de Java.

### **2.2.2 Nivells d'APIs**

Android al llarg de la seva curta vida a vist un conjunt de versions algunes de les quals eren meres actualitzacions i altres que definien noves formes d'interactuar amb els dispositius modificant així l'API subjacent per a programar sobre ella, en el moment de desenvolupar una aplicació s'ha d'escollir quin nivell de l'API es prendrà com a objectiu de la mateixa per poder assegurar que tots els elements que s'utilitzin estiguin suportats dintre d'aquell nivell de l'API.

La nostra aplicació no utilitzarà accés a característiques avançades dels nous dispositius i les noves APIs, per exemple la última versió de l'API, la 11, ofereix funcionalitats específiques per a sistemes tablet que no són el nostre sistema objectiu i per tant no tindria sentit posar com a requisit aquest nivell ja que deixaríem fora un gran conjunt de dispositius.

El nivell d'API que hem trobat adient escollir per a la nostra aplicació ha estat el nivell 7, aquest es correspon amb la versió d'Android 2.1. La raó per escollir aquest nivell ha estat per una banda perquè va suposar l'estabilització de la branca 2.x d'Android i per tant ja disposava d'un seguit de característiques comunes i provades, i per l'altra perquè segon les últimes estadístiques la majoria de dispositius Android actuals funcionen sobre versió 2.1 o superior i per tant com a solució de compromís ens ha semblat que era la millor opció.

### **2.2.3 Estructura d'aplicació Android**

L'estructura d'una aplicació Android ha de seguir un seguit de recomanacions definides per la guia de desenvolupadors d'Android que permetin al sistema entendre i interactuar amb la mateixa. Així Android defineix quatre tipus de components bàsics per a estructurar una aplicació que es poden o no utilitzar per a donar forma a la mateixa.

Aquests components els detallarem a continuació



## **Activitats (Activity)**

Les activitats són l'element visible d'una aplicació, cada activitat representa un de les pantalles visibles de l'aplicació i ofereix un entorn gràfic sobre el qual l'usuari interactuarà amb l'aplicació. Aquestes activitats són les encarregades de tractar els events que produeixi l'usuari i actuar en conseqüència amb les accions d'aquest per mostrar els resultats desitjats.

Una aplicació contarà com a mínim amb una activitat que serà el punt d'entrada de la mateixa i que es carregarà quan l'usuari iniciï la seva execució.

## **Serveis (Service)**

Els serveis a diferència de les activitats no ofereixen interacció directa amb l'usuari ja que no suposen una representació gràfica com les activitats. Per contra els serveis estan destinats a realitzar un seguit de tasques no associades amb la presentació de pantalla i oferir aquests serveis a una o diverses aplicacions de manera que un mateix servei pugui donar suport a diferents aplicacions d'un dispositiu sense que cadascuna d'aquestes hagi de reimplementar les funcionalitats del servei.

Per exemple existeixen serveis que permeten realitzar trucades o enviar SMS que venen definits per defecte al sistema.

## **Receptors de Broadcast (Broadcast Receivers)**

Un altre component que podem utilitzar alhora de realitzar aplicacions Android són els receptors de broadcasts aquests són els encarregats de respondre a determinats events genèrics del sistema, per exemple que s'hagi rebut un missatge, i avisar a les aplicacions interessades d'aquest fet.

Existeixen receptors per a events coneguts pel sistema com per exemple bateria baixa o una captura fotogràfica i d'altres que poden ser definits per les aplicacions per dur a terme diverses tasques.

## **Proveïdors de Continguts (Content Provider)**

Els proveïdors de continguts són els encarregats de carregar les dades persistents que formin part del sistema. Existeixen diversos proveïdors que es poden trobar definits per defecte



al sistema com pot ser la llista de contactes telefònics i que poden ser consultar per les aplicacions que tinguin permís per accedir als mateixos.

Si l'aplicació necessita emmagatzemar informació persistent haurà de fer ús d'un proveïdor de continguts, o varis dependent de la complexitat, que ofereixin aquest servei a l'aplicació.

La nostra aplicació no farà servir els quatre tipus de components per a la seva implementació, només utilitzarà Activitats, Serveis i Proveïdors de Continguts que són els que hem considerat necessaris per a implementar la funcionalitat desitjada i definida en aquest document.

Amb això podem donar per tancada la definició dels elements bàsics que utilitzarem al nostre sistema, en el següent punt veurem com a partir d'aquests elements definim l'arquitectura del sistema complet.



## **3 Arquitectura**

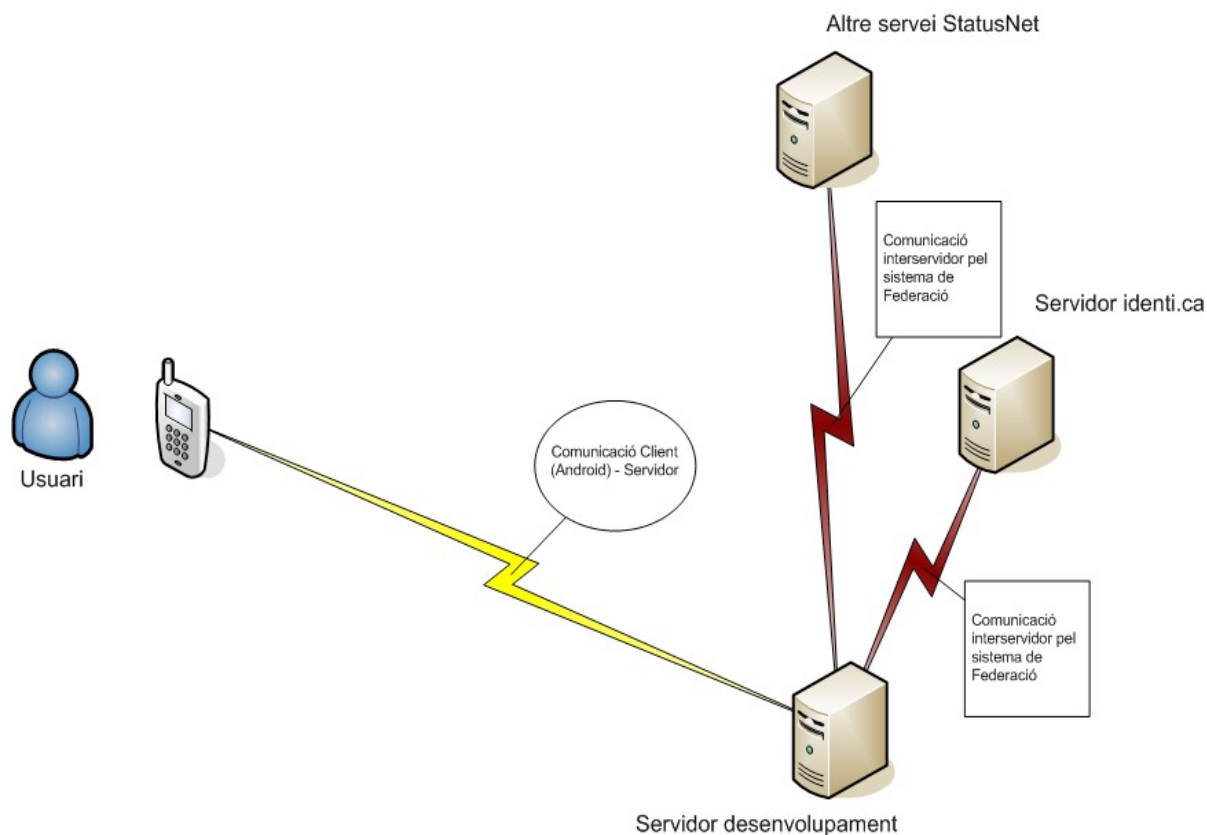
El tipus de projecte que desenvolupem no està basat en una arquitectura complicada i per tant el gràfic resultant com es pot veure a la figura adjunta és molt simple i està format per components que ja hem definit en els anteriors punts.

### **3.1 Arquitectura del sistema**

Com que la base principal que realitzarem serà la implementació de l'aplicació sobre Android la resta d'elements estan més presents com a prova de concepte que com a entitats essencials per al desenvolupament del projecte, tot i això estan inclosos al diagrama com a representació del sistema complet sobre el qual treballarem.

Els components que formen part del sistema del nostre projecte són els següents:





Imatge 4: Diagrama arquitectura

- **Servidor StatusNet**

Al diagrama hem representat tres servidors, un d'ells el nostre servidor de proves i l'altre el servidor d'identica que és el que utilitzarem per a realitzar proves d'integració amb un entorn real.

Recordem també que encara que no formi part de l'abast d'aquest projecte els servidors poden estar federats, es a dir que poden intercanviar informació entre ells i per tant no caldria que l'usuari interaccionés amb tots de manera simultània sinó que podria comunicar-se amb tots ells a partir d'un.



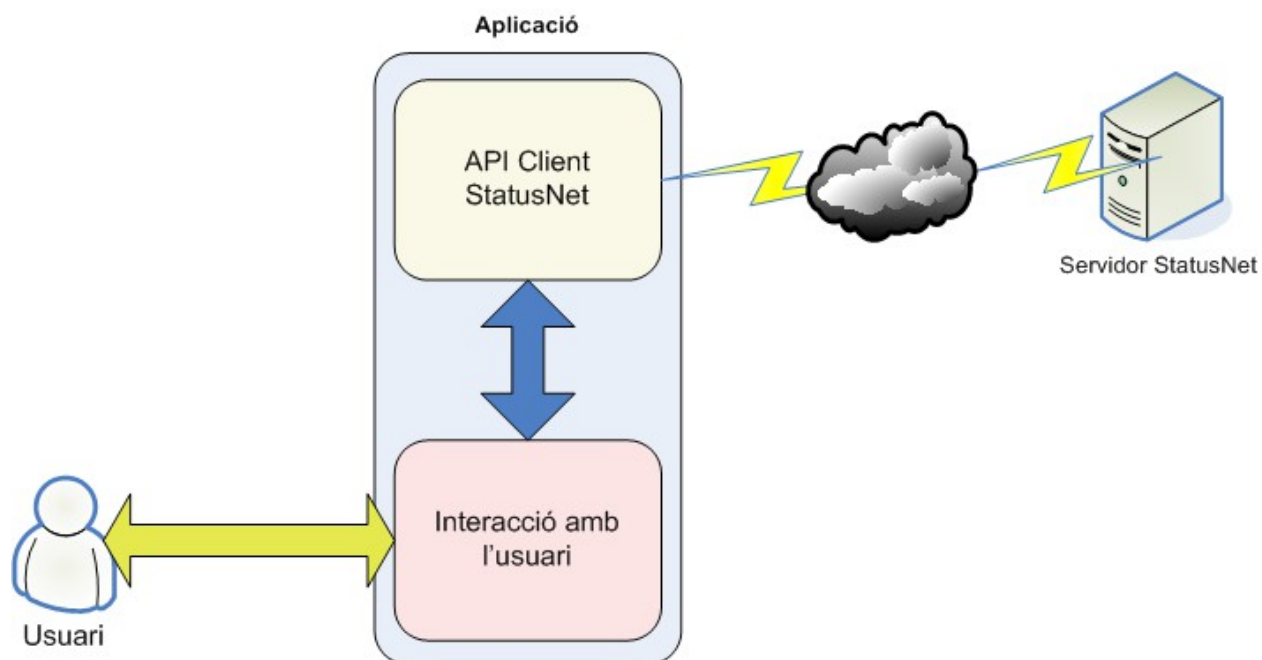
- **Aplicació mòbil Android**

L'accés als serveis oferts per un servidor StatusNet es pot realitzar a través de diverses plataformes de comunicació com hem comentat en un apartat anterior (web, correu electrònic, SMS, XMPP, ...), en el nostre cas aquest accés es realitzarà a través d'una aplicació Android desplegada en un dispositiu mòbil tal i com hem representat al diagrama.

Com es pot veure l'arquitectura del sistema es molt simple i es tracta d'una aplicació normal client - servidor on només implementem la part de l'aplicació ja que la del servidor ens ve donada pel sistema StatusNet que hem escollit.

## 3.2 Arquitectura de l'aplicació

A més de la visió general del sistema exterior també disposem d'una aplicació desenvolupada en Android que té la seva pròpia arquitectura interna. Pel seu desenvolupament hem decidit dividir el sistema en dos blocs principals tal i com s'indica a la figura.



Imatge 5: Arquitectura aplicació



Les funcionalitats que ofereixen els blocs és la següent:

- **API Client**

L'element base de la nostra aplicació serà la implementació d'una API que realitzi les tasques de client StatusNet i implementi les funcionalitats de comunicació per interactuar amb l'API del servidor per a totes aquelles actuacions que hem definit anteriorment com a dins de l'abast del nostre projecte.

Aquesta API estarà implementada com a servei Android i exposarà un seguit de mètodes a través de la seva interfície que seran explotats per la nostra aplicació des d'on l'usuari podrà interactuar amb StatusNet. El tipus de servei escollit serà privat ja que no volem que terceres aplicacions tinguin accés al servei.

- **Aplicació**

L'altra element que forma part de l'arquitectura és la pròpia aplicació que consta de l'entorn gràfic per a interactuar amb l'usuari i la lògica necessària per a comunicar amb el servei que ofereix l'API Client i tractar les respostes per presentar-les a l'usuari.

L'aplicació estarà formada per un seguit d'activitats que donaran lloc a les diferents pantalles amb les qual podrà interactuar l'usuari i un proveïdor de continguts per emmagatzemar les dades de configuració que l'usuari introdueix i que es volen mantenir de forma persistent entre execucions.

En els següents apartats entrarem en el detall de l'estructura de l'API i de l'aplicació per a definir concretament quins seran aquells elements que la componen i amb els quals acabarem treballant.



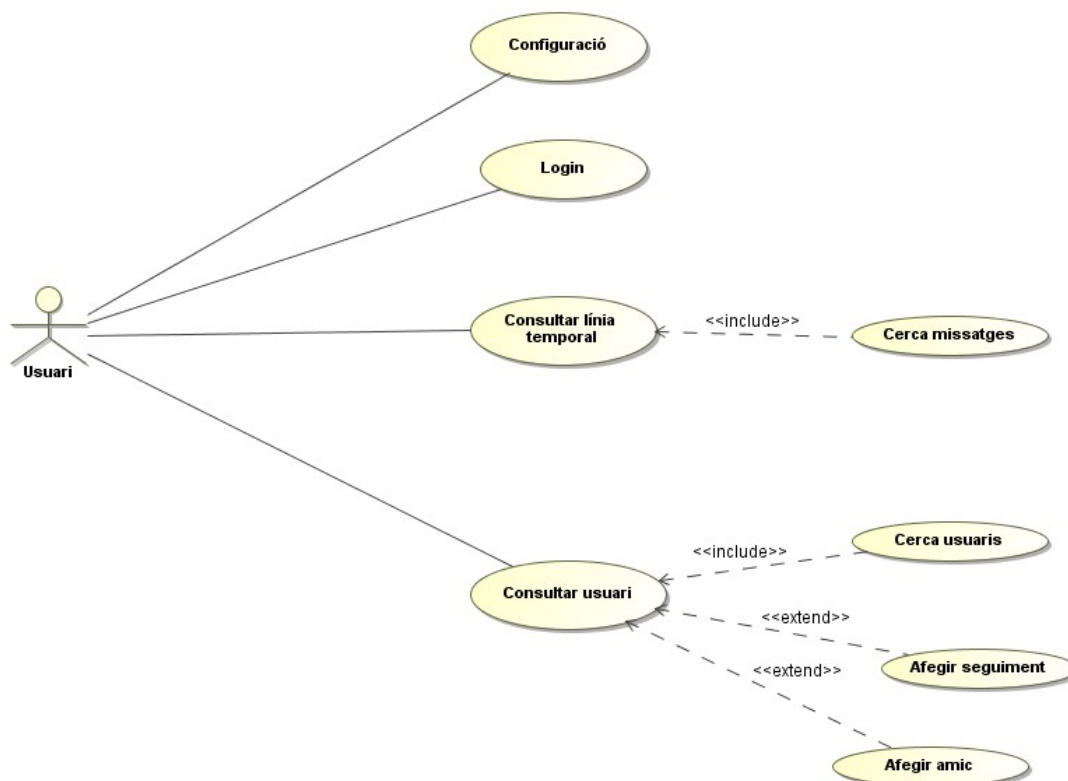
## 4 Disseny aplicació StatusDroid

La nostra aplicació client s'anomenarà StatusDroid i la seva funcionalitat comprendrà els elements que hem detallat en els apartats anterior. Per assolir aquesta funcionalitat caldrà realitzar una aplicació Android que implementi tots els elements que hem comentat dividits en els dos components principals, una API Client que serà la que realitzarà la comunicació amb els servidors StatusNet i una aplicació que controlarà els elements gràfics i la navegació de l'usuari.

### 4.1 Casos d'ús

Els casos d'ús que tenim a la nostra aplicació es poden veure representats en el següent diagrama

Els casos d'ús representats són els següents:



Imatge 6: Diagrama Casos d'Ús

- **Configuració:** Cas d'ús que representa l'activitat de configuració d'un servei StatusNet a l'aplicació per part de l'usuari.
- **Login:** El procés de consulta de diversos elements del sistema, com el llistat d'amics o la dades d'usuari requereixen la identificació de l'usuari, això s'aconsegueix en el cas d'ús del login on l'usuari introdueix les seves dades.
- **Consultar línia temporal:** Permet realitzar consultes sobre línies temporals a partir de diversos paràmetres, com llistat públic, d'amics, un tag concret, ...



- **Cerca:** El procés de cerca permet que els usuaris cerquin per diversos paràmetres dintre de l'aplicació, en el nostre cas implementarem la cerca de misstges associats a la línia temporal.
- **Consulta d'usuari:** Permet consultar la informació disponible per a un determinat usuari.
  - **Afegir seguiment:** Afegeix l'usuari que s'estigui consultant a la llista de seguiment de l'usuari que està realitzant la petició.
  - **Afegir amic:** Si l'usuari consultat ja està seguint a l'usuari de l'aplicació es permetrà afegir a aquest com a amic.

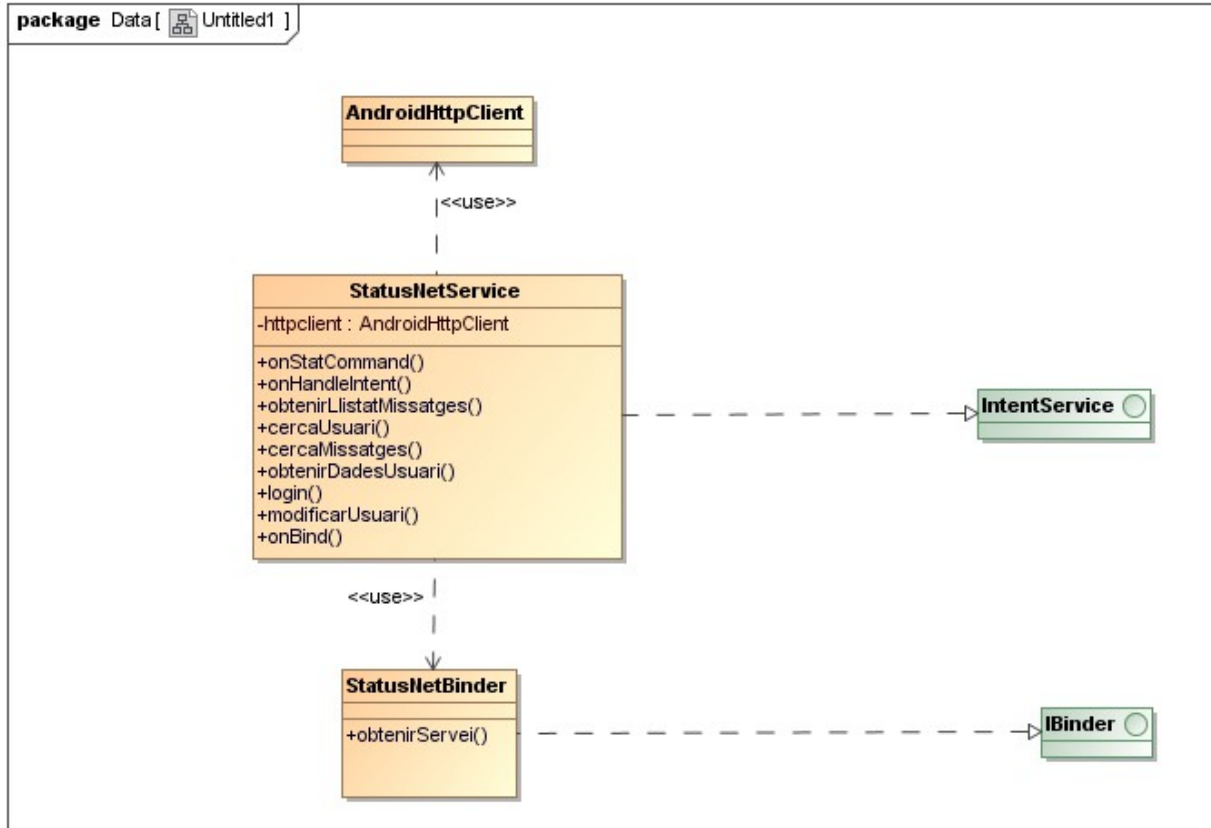
## 4.2 Implementació de l'API Client

La implementació de l'API Client es farà fent servir les funcionalitats que ofereix l'entorn Android que ja hem explicat anteriorment quan hem parlat de les característiques més rellevants d'aquesta tecnologia.

En concret utilitzarem la funcionalitat que ofereixen els Serveis d'Android ja que s'ajusta perfectament al que volem implementar amb l'API de Client. Així construirem un servei al qual les activitats que formin part de la nostra aplicació principal es puguin connectar per demanar que aquest interaccioni amb els diversos serveis a nivell de servidor que ofereix StatusNet.

Segons la documentació d'Android això requerirà la implementació del Servei i a més d'una interfície de comunicació per a realitzar el lligam entre el servei i aquelles aplicacions que vulguin utilitzar les actuacions que aquest ofereix.

El diagrama simplificat de les classes que utilitzarem mostrant aquelles més rellevants per a oferir la funcionalitat desitjada és el que es pot veure en la següent imatge



Imatge 7: Diagrama classes API StatusNet

On tenim com ja hem indicat una classe que serà la classe principal del servei i que contindrà el conjunt de mètodes que per una banda s'han d'implementar per definir el cicle de vida del servei, i per altra el conjunt de mètodes públics que formen part de l'API de Client que utilitzarà l'aplicació.

Adicionalment s'ha de definir una interfície que es la que es subministrarà a la implementació quan realitzi el lligam amb el servei per a poder realitzar la interacció amb el mateix, aquesta interfície pot o bé implementar ella directament els mètodes d'accés a l'API o com hem fet en aquest cas ens ha de permetre d'obtenir la referència al nostre servei.



També hem representat l'ús del client HTTP d'Android sobre el qual realitzarem totes les comunicacions amb el servidor per això hem considerat rellevant incloure'l en el diagrama per representar-lo com a classe fonamental pel funcionament.

#### ***4.2.1 Llenguatge de la comunicació***

Comentar que l'API d>StatusNet permet la comunicació amb HTTP mitjançant diversos llenguatges, XML o JSON, per motius d'eficiència implementarem la nostra aplicació per fer ús d'aquest últim mètode ja que a igualtat de dades ofereix un rendiment per bit enviat superior ja que no és un llenguatge tan explícit com l'XML. Hem considerat que tractant-se de dispositiu mòbils on no sempre la tarifa de dades és una tarifa plana i la transmissió de les mateixes pot encarrir molt el cost JSON era millor alternativa que XML.

Per aquesta raó farem servir les APIs de parseig i tractament de missatges de JSON que ja venen incloses a l'API d'Android i que facilitaran la tasca de llegir les dades i presentar els resultats als usuaris.

#### ***4.2.2 Funcionament dels Serveis i Fils d'execució***

La implementació del servei requereix que les crides es realitzin des d'un fil d'execució independent ja que per convencions de la plataforma si les activitats fan las crida la fan des del mateix fil on s'executa l'entorn gràfic de l'usuari bloquejant aquest.

Aquesta característica d'Android ja es remarca a la documentació del mateix i s'haurà de tenir present alhora de realitzar la implementació per aconseguir que l'aplicació sigui funcional i no presenti una mala experiència d'usuari.

#### ***4.2.3 Tipus de Servei***

Finalment acabarem fent una referència al tipus de servei que farem servir ja que considerem que es un factor rellevant a tenir present durant el desenvolupament.





Android per defecte publica els serveis de forma genèrica perquè qualsevol aplicació, que compleixi els permisos establerts, pugui accedir als mateixos, en el nostre cas no volem utilitzar aquest comportament per defecte ja que requereix utilitzar els serveis realitzant crides entre processos (IPC) i complicant la implementació. Per tant farem servir crides locals i treballarem amb una implementació privada del servei, el que ens permetrà un desenvolupament més ràpid i senzill.

## 4.3 Entorn d'usuari

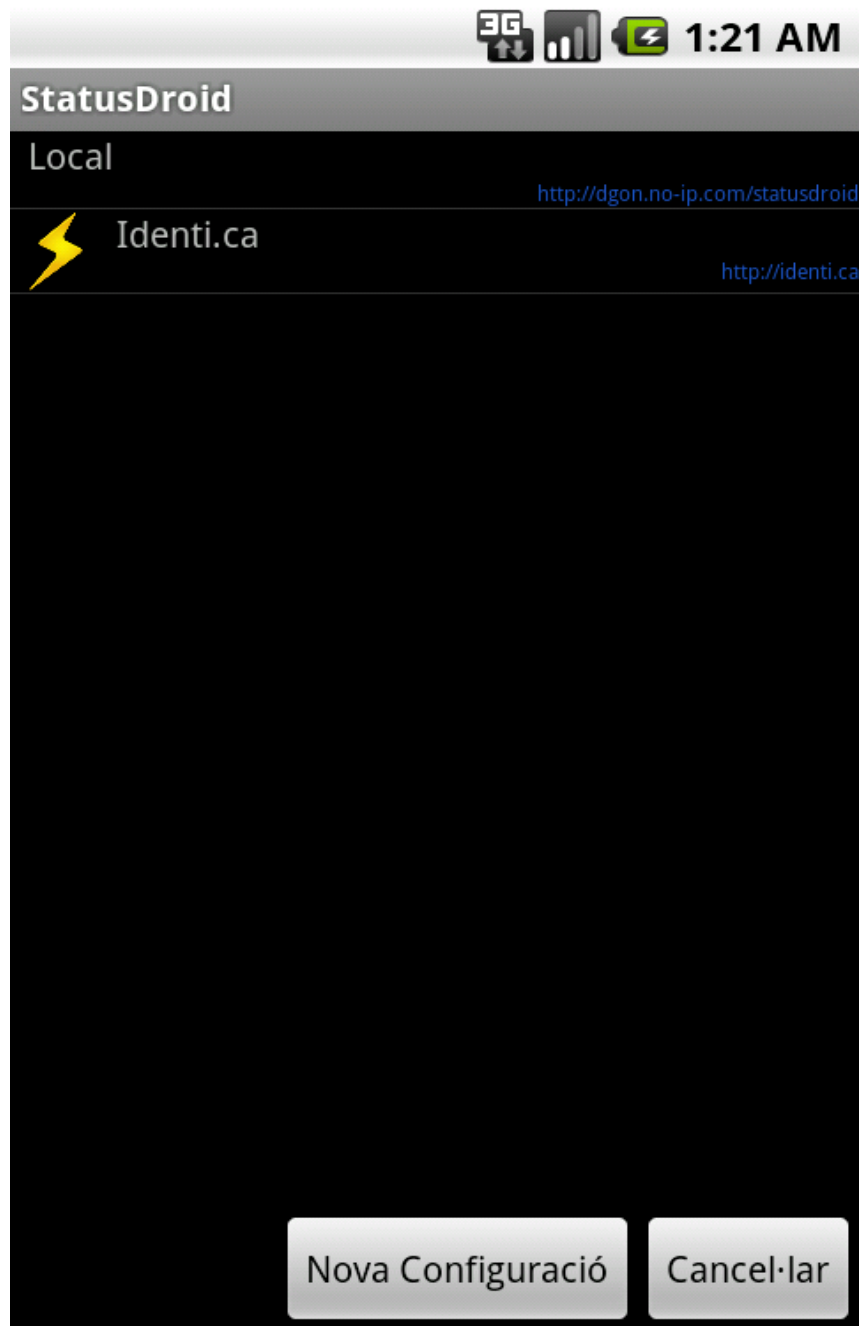
El segon mòdul que forma part de la nostra aplicació és aquell encarregat de l'entorn de l'usuari, en aquest cas hem definit l'aplicació així per fer-la el més modular possible separant els elements propis de la lògica de comunicació dels elements que realitzen la capa d'usuari, podríem dir que es tracta d'una implementació en 3 capes a l'estil MVC, on l'API es la capa de control, l'entorn d'usuari que comentarem a continuació s'encarrega de les dades de la vista, i les dades es troben al servidor d>StatusNet, a més d'un conjunt de dades que tindrem en un entorn local.

En aquest apartat definirem el disseny de l'entorn d'usuari entrant en detall en cadascuna de les funcionalitats que oferirà aquest a l'usuari final de la nostra aplicació.

### 4.3.1 Pantalla de configuració principalment

Aquesta serà la pantalla de configuració principal estarà formada per un llistat de tots aquells serveis StatusNet que tinguem configurats al nostre dispositiu, recordem que StatusNet a diferència de Twitter no es un servei únic sinó una plataforma de la qual existeixen múltiples implantacions a tot el món, per exemple en el nostre cas farem servir dues, una serà el nostre entorn local de desenvolupament i l'altra la plataforma identi.ca per a realitzar proves en un entorn de producció real.

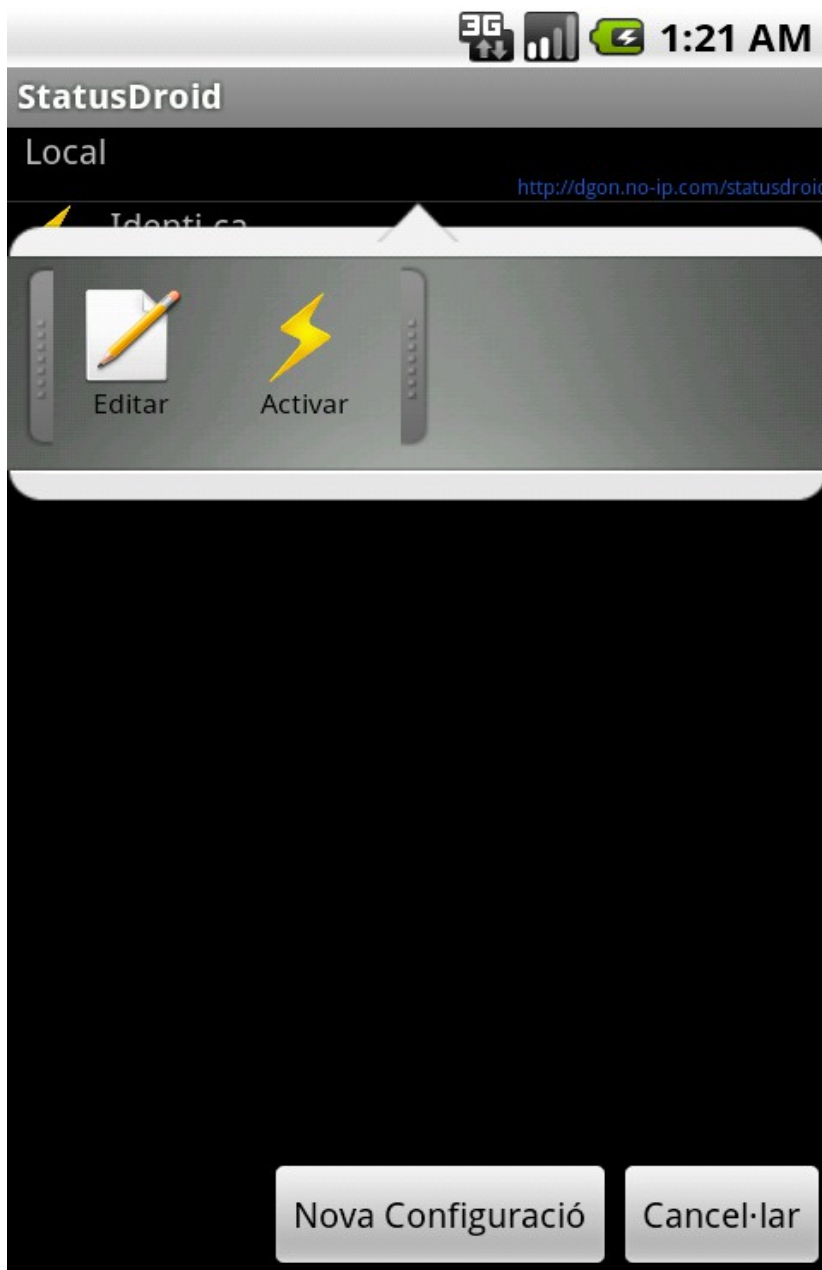
Les opcions que ens oferirà aquesta pantalla seran la de gestió d'aquestes configuracions, crear noves configuracions o editar les existents. A més també es podrà des d'aquesta pantalla seleccionar aquella configuració a la qual volem estar connectats en un moment determinat.



*Imatge 8: Pantalla de Configuració Principal*



La interacció amb les diferents configuracions es farà mitjançant un menú d'accions ràpides (*Quick actions*), que ens mostrarà les opcions contextuais disponibles per cadascuna de les opcions de configuració guardades.



*Imatge 9: Pantalla de configuració - Menú contextual*



### ***4.3.2 Pantalla de configuració***

La pantalla de configuració és on l'usuari pot configurar l'accés als diferents serveis StatusNet que podem trobar a la xarxa i la pantalla que apareixerà el primer cop que s'executi l'aplicació i no hi hagi una configuració inicial per a la connexió.

En aquesta pantalla l'usuari podrà introduir les dades de configuració de l'entorn així com el comportament de la seguretat pel que fa a les seves credencials.



3G 1:01 AM

StatusDroid

## Servidor

Nom

URL

Autoconnectar

## Usuari

Nom

Contrassenya

Guardar Cancel·lar

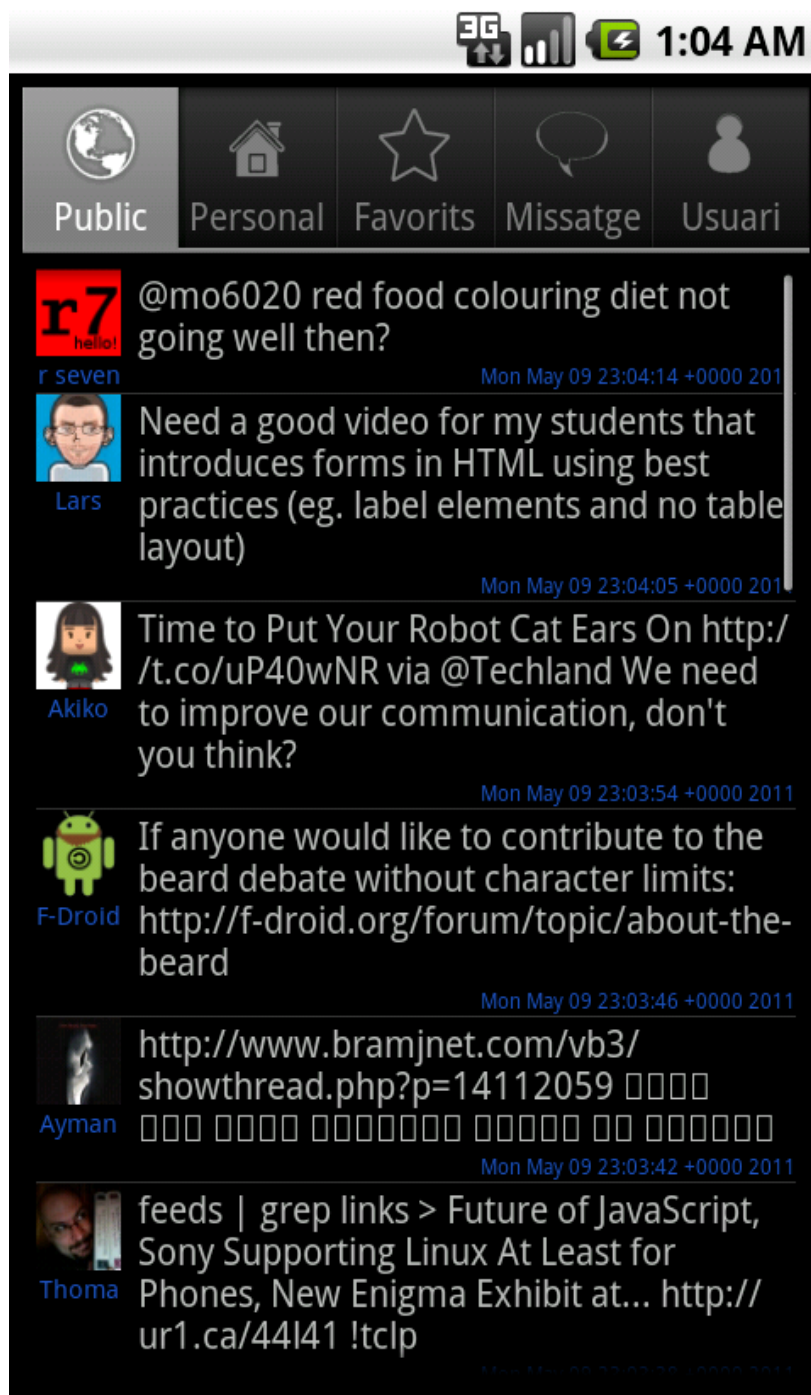
Imatge 10: Pantalla de configuració



### ***4.3.3 Pantalla principal***

La pantalla principal de l'aplicació serà aquella que mostra el llistat dels missatges de la línia temporal pública, des d'aquesta es podrà accedir a les diferents opcions que ofereix l'aplicació per mitjà d'un sistema de navegació per pestanyes.

Els missatges es mostraran ordenats cronològicament tal i com els retorna StatusNet i les dades rellevants del missatge com a usuari creació, tipus de missatge i hora a la qual es va enviar es mostraran clarament en un element annex al mateix.



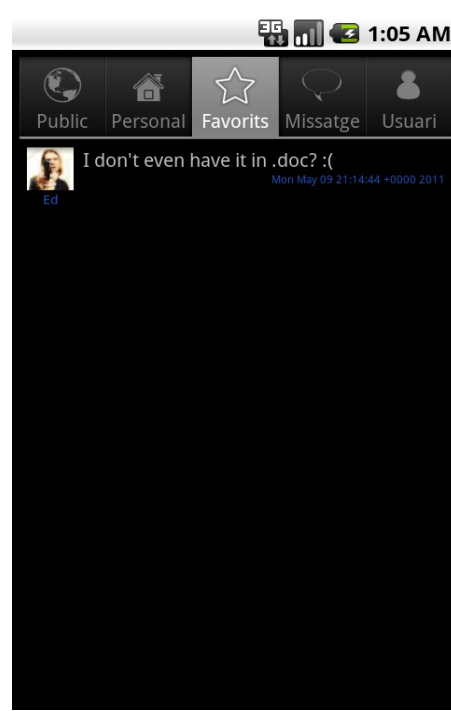
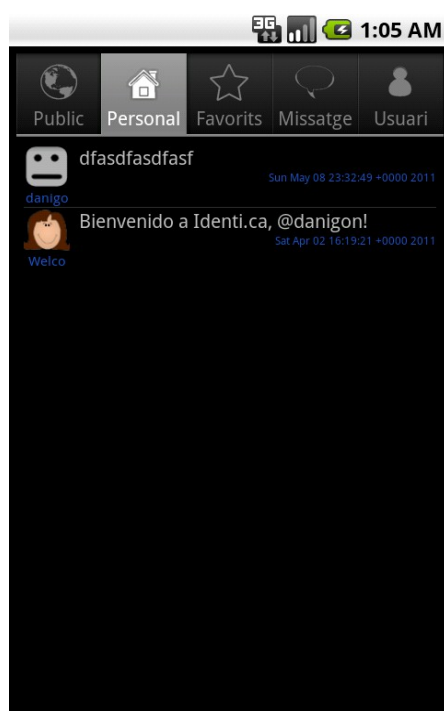
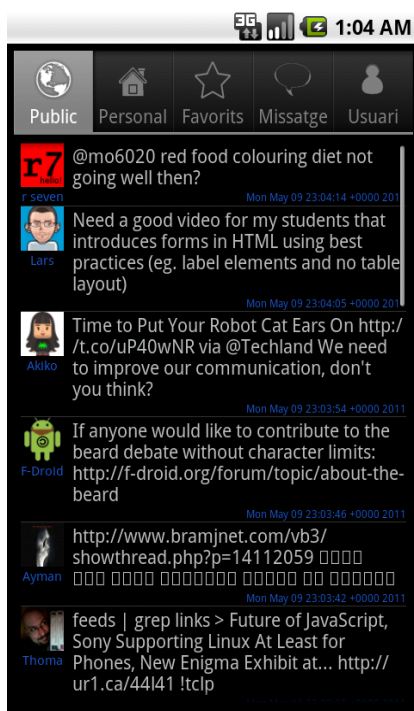
Imatge 11: Pantalla Principal





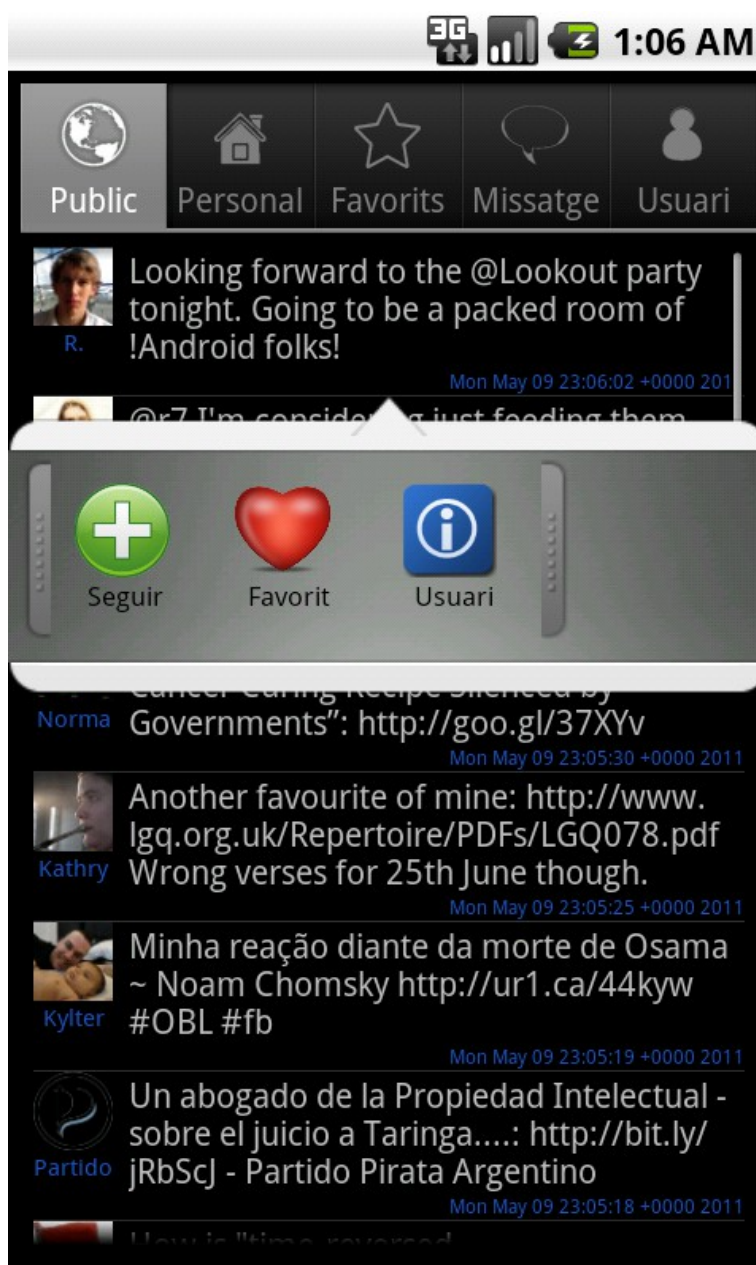
Les opcions de navegació relacionades amb les línies temporals que tenim disponibles són les següents:

- **Públic:** És la línia temporal pública que està associada a tots aquells missatges d>StatusNet emesos sense cap restricció i que poden ser visualitzats per tots els usuaris.
- **Personal:** La línia temporal personal és la que ens mostra els missatges que enviem nosaltres i aquells dels usuaris als quals estem seguint, ja siguin públics o privats i estiguin destinats al nostre usuari.
- **Favorits:** La última línia temporal a la qual tenim accés i que també forma part d>StatusNet és aquella associada als missatges que hem marcat com a favorits.





Des de qualsevol de les pantalles de missatges tenim diferents opcions associades als missatges que són accessibles com en el cas del llistat de configuracions des d'un menú d'accions ràpides.



Imatge 12: Pantalla de missatges - Menú d'accions



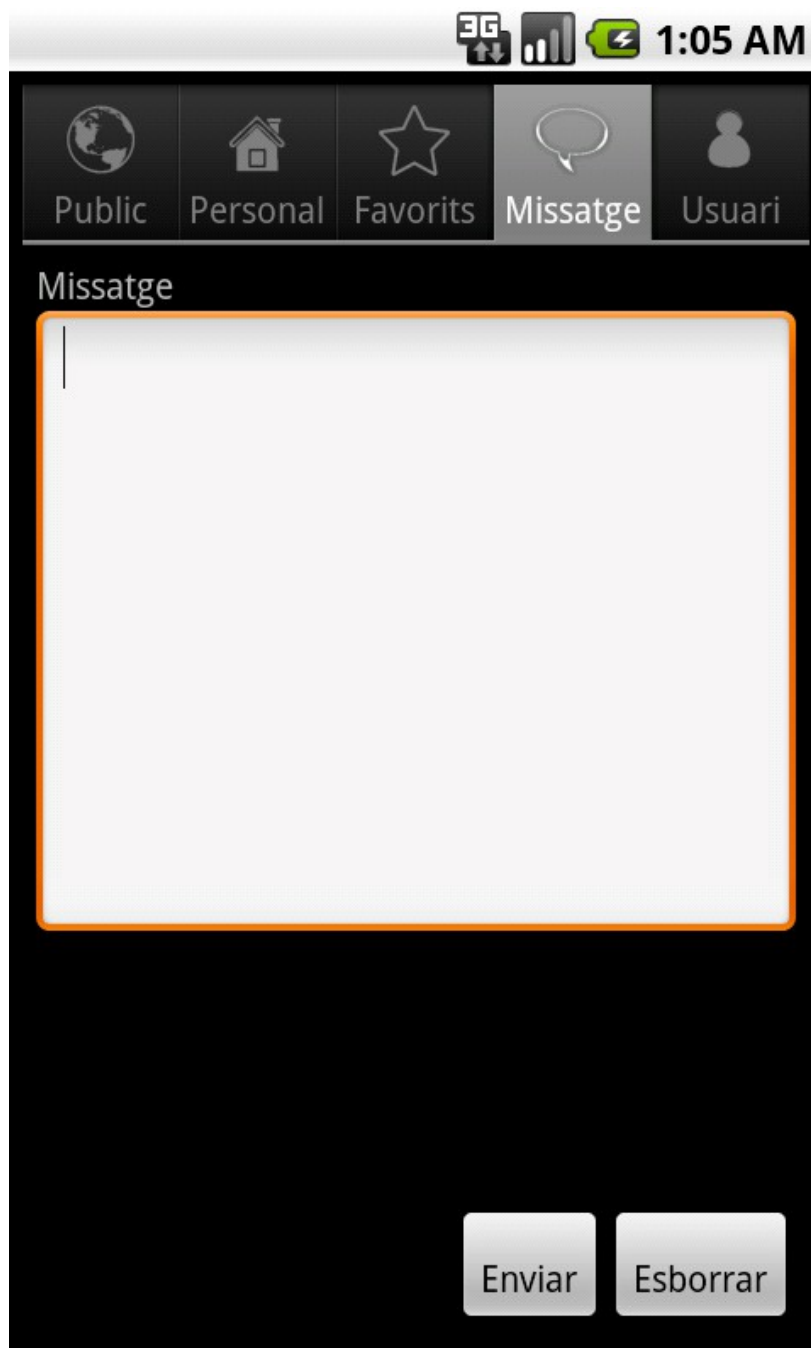
Les accions disponibles en aquest menú són les que detallem a continuació:

- **Seguir:** Ens permet afegir l'usuari que hagi enviat al missatge al nostre llistat d'usuaris a seguir, amb la qual cosa aconseguirem que els seus missatges apareixin també al nostre llistat personal.
- **Favorit:** Afegeix el missatge al nostre llistat de missatges favorits per poder-lo consultar des de la pestanya associada al mateix.
- **Usuari:** Permet accedir a la pantalla d'informació de l'usuari que ha creat el missatge.

#### *4.3.4 Pantalla d'enviament de missatges*

La opció d'enviament de missatges és la opció principal d'interacció amb el sistema, aquesta ens permet enviar en nom del nostre usuari nous missatges a StatusNet que apareixeran al sistema, per fer-ho simplement s'ha d'introduir un missatge al quadre de text i premer el botó enviar.

La limitació existent a StatusNet indica que la grandària màxima del missatge serà de 140 caràcters i per tant aquesta pantalla realitzarà la limitació del missatge a aquesta longitud per evitar problemes d'enviament amb missatges de més llargada.



*Imatge 13: Pantalla d'enviament de missatges*



#### ***4.3.5 Pantalla d'informació d'usuari***

La pantalla d'informació d'usuari mostra les dades d'un usuari en concret, aquestes inclouen la imatge de perfil, el seu alias dins de l'aplicació i el seu nom si aquest està informat i es públic, i les dades de relació amb altres usuaris del sistema.

També ens permetrà veure si estem seguint a l'usuari que estem consultant o encara no el tenim al llistat d'usuaris que estan sota el nostre seguiment.



Imatge 14: Pantalla d'informació d'usuari

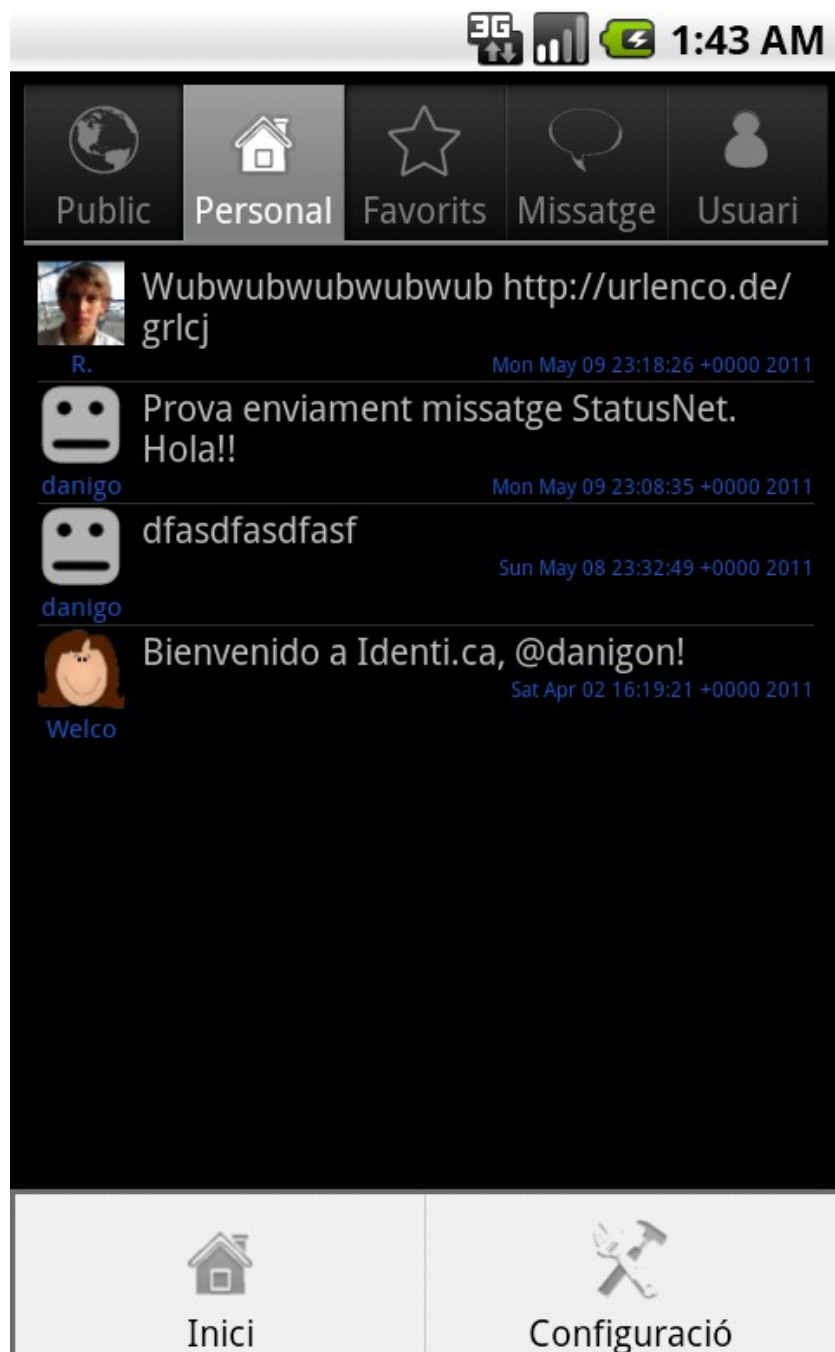


#### **4.3.6 Opcions contextuais**

Finalment disposem d'un parell d'opcions contextual que ens facilitaran la navegació entre els elements principals de la navegació. Aquestes opcions estan associades al menú principal d'Android i són accessibles mitjançant el botó de menú dels dispositius Android.

Aquestes opcions son les següents:

- **Inici:** Permet accedir directament a la pantalla inicial de l'aplicació on es mostra la línia temporal pública amb els missatges associades a la mateixa.
- **Configuració:** Accés directe a la pantalla de configuració principal on es mostra el llistat dels servidors configurats.



Imatge 15: Menú contextual general





## 4.4 Persistència de dades

---

La persistència de les dades de configuració de la nostra aplicació serà un element fonamental per a dur a terme la implementació de la funcionalitat desitjada ja que dades com la configuració de la connexió s'han d'emmagatzemar perquè l'usuari no hagi de reintroduir les mateixes cada cop que executi l'aplicació.

La implementació de la persistència la farem utilitzant les funcionalitats que ja em comentat que ofereix Android per a aquesta tasca, en concret implementarem un Proveïdor de continguts que utilitzarà com a motor de persistència la base de dades SQL Lite interna d'Android.

Aquest proveïdor de continguts ens permetrà emmagatzemar les dades de configuració i totes aquelles que considerem necessàries per al correcte funcionament del sistema.

L'estructura de la base de dades on es guarda la informació és molt simple i està formada per una única taula que guarda les dades de configuració dels servidors StatusNet configurats. El model de dades és el següent:

### Taula serverConfig

- **id**: Identificador únic de l'element de configuració.
- **serverName**: Nom de la configuració del servidor StatusNet.
- **serverUri**: URI d'accés al servidor StatusNet.
- **autoconn**: Indicador d'autoconnexió a aquest servidor.
- **user**: Nom d'usuari per a accedir al servei.
- **password**: Password de l'usuari per a accedir al servei.

Comentar que pel que fa als passwords s'ha tingut present que s'han d'emmagatzemar oferint un mínim nivell de seguretat per això aquests són xifrats utilitzant AES256 abans de guardar-se a la base de dades per complicar la pèrdua dels mateixos. La clau d'enciptació s'obté de manera aleatòria el primer cop que s'executa l'aplicació i per recuperar les dades



emmagatzemem la llavor del generador al gestor de preferències d'aplicació que ofereix Android.

## **4.5 Activitats**

---

L'últim element que ens queda per definir per a donar per acabat el disseny de l'entorn d'usuari són les activitats que el componen. Recordem que cadascuna de les activitats s'encarrega del control de cadascuna de les pantalles i la navegació entre pantalles de l'usuari equival a la navegació entre activitats. A partir d'aquestes dades definim les següents activitats:

### **4.5.1 Configuració**

És l'activitat que s'encarrega de la pantalla de configuració i controla la presentació i els events que es presenten a la mateixa, s'haurà d'encarregar de recuperar i persistir les dades per a realitzar la presentació de les mateixes a la pantalla i deixar-les disponibles per a la resta de l'aplicació.

### **4.5.2 Llistat de missatges**

El llistat de missatges és la pantalla principal de l'aplicació excepte en la primera execució que es carrega per defecte la pantalla de configuració inicial on l'usuari ha d'introduir les dades d'una connexió per poder continuar.

Aquesta pantalla mostra un llistat de missatges associat a algun filtre sobre les línies temporals tal i com hem comentat anteriorment existeixen diverses opcions de filtratge que configuraran el comportament d'aquesta activitat, ja sigui aquells missatges associats a la línia pública, a la personal o als favorits.

### **4.5.3 Informació d'usuari**

Aquesta activitat gestiona la obtenció i presentació de les dades d'usuari. Alhora de realitzar la obtenció delegarà la tasca a l'API StatusNet associada a la nostra aplicació mitjançant el servei Android detallat anteriorment.



Les dades que es mostren en aquesta pantalla seran aquelles més rellevants que retorna el mètode d'informació d'usuari que té l'API d>StatusNet, això inclou la imatge del perfil, l'alias, el nom i les seves dades d'activitat amb altres usuaris (amics, seguidors, ...).



## 5 Continguts i estructures de l'entrega

A l'entrega final completa s'adjunten els següents elements que formen part del projecte, el document bàsic de l'entrega es aquesta mateixa memòria però també es poden trobar els següents lliurables:

### 1. Aplicació StatusDroid

Aquesta és l'aplicació principal del lliurament, està formada per un paquet *apk*, format estàndard de paquet per a aplicacions Android, que pot ser desplegat tant a l'emulador d'Android disponible al SDK o a qualsevol dispositiu compatible que compleixi amb les especificacions marcades per poder executar la mateixa.

### 2. Codi font

El codi font de l'aplicació generada que forma part del producte entregar anterior també es troba com a element lliurable dintre del projecte realitzat. En aquest codi es pot trobar la implementació de l'aplicació que està empaqueta al paquet *apk* per Android.

### 3. Manual d'usuari

Un altre document que s'adjunta amb l'entrega és el manual d'usuari de l'aplicació on es detalla el funcionament i les opcions de les diferents pantalles presents a l'aplicació StatusDroid indicant la seva funcionalitat i com navegar per la mateixa.

Pel que fa a altres elements no s'adjunten amb aquesta memòria ja que es poden obtenir directament de la web, tals i com pot ser el servidor StatusNet. Per més informació consultar la bibliografia associada.

## 5.1 Instal·lació

L'aplicació es pot executar de dues maneres, per una banda es pot executar directament en qualsevol dispositiu Android compatible, es a dir, aquells que tinguin una versió de l'API que



sigui igual o superior a la utilitzada en el desenvolupament del projecte, en el nostre cas tots aquells dispositius que tinguin la versió 2.1 o superior del sistema operatiu.

Per altra banda es possible executar l'aplicació fent ús de l'emulador d'Android present al SDK, per fer-ho partint del paquet *apk* d>StatusDroid s'han de seguir les dades detallades al següent punt.

### 5.1.1 Instal·lació a l'emulador Android

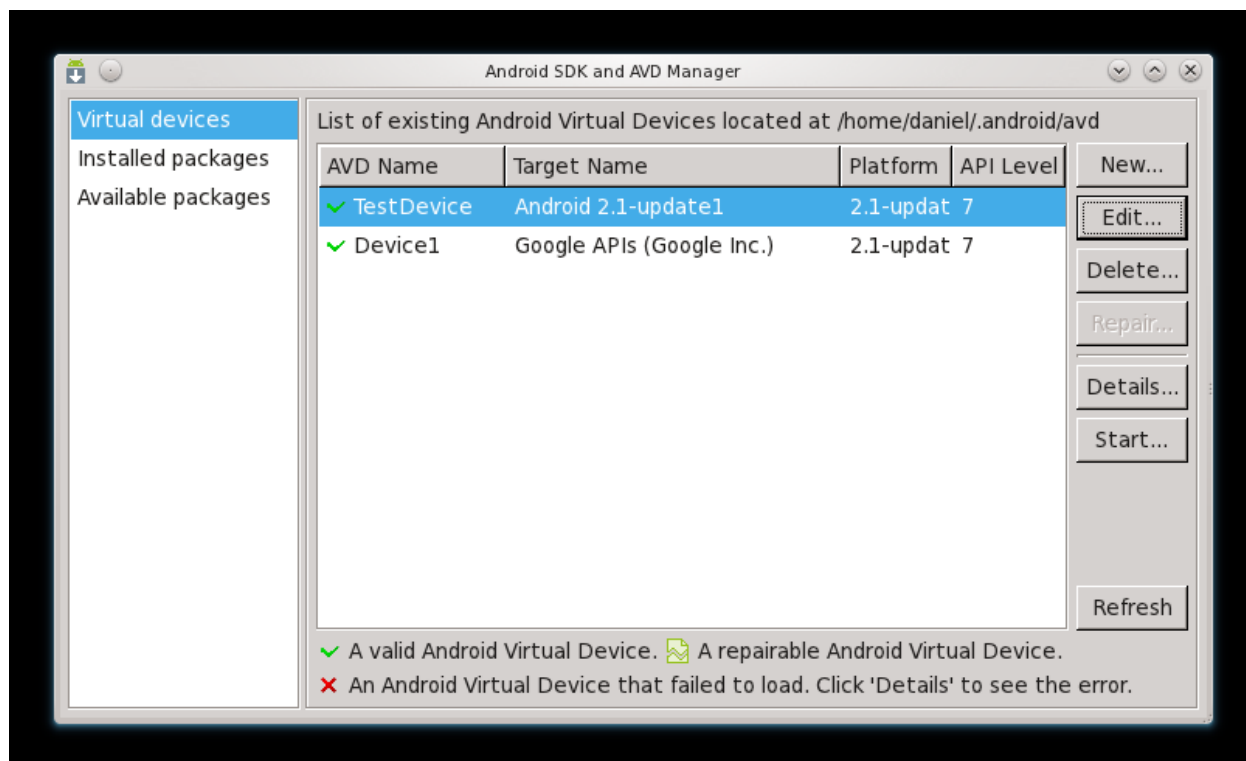
L'execució de la nostra aplicació es pot realitzar sobre l'emulador d'Android, aquest emulador forma part de la plataforma de desenvolupament (SDK) d'Android i es pot aconseguir de forma totalment gratuïta visitant la web <http://developer.android.com/sdk/index.html> .

En aquesta web trobarem disponibles les versions del SDK per tres plataformes diferents Windows, Linux i Mac, hem d'escollir aquella versió que s'adapti al nostre sistema i seguir els passos per a la seva instal·lació que s'indiquen a la mateixa pàgina. En aquesta guia no indicarem aquest procediment ja que ja està ben explicat a la web i a més a més es depenent de la plataforma on es vulgui instal·lar l'SDK, en el nostre cas tant el desenvolupament com les proves es van realitzar utilitzant l'SDK sobre un sistema Linux, però l'execució de l'emulador i l'aplicació generada hauria de ser independent d'aquest.

Un cop instal·lat i si hem seguit les indicacions de la pàgina indicada hauríem de tenir ja disponible l'emulador i configurat el PATH del sistema per poder executar l'eines necessàries per a la instal·lació de l'aplicació.

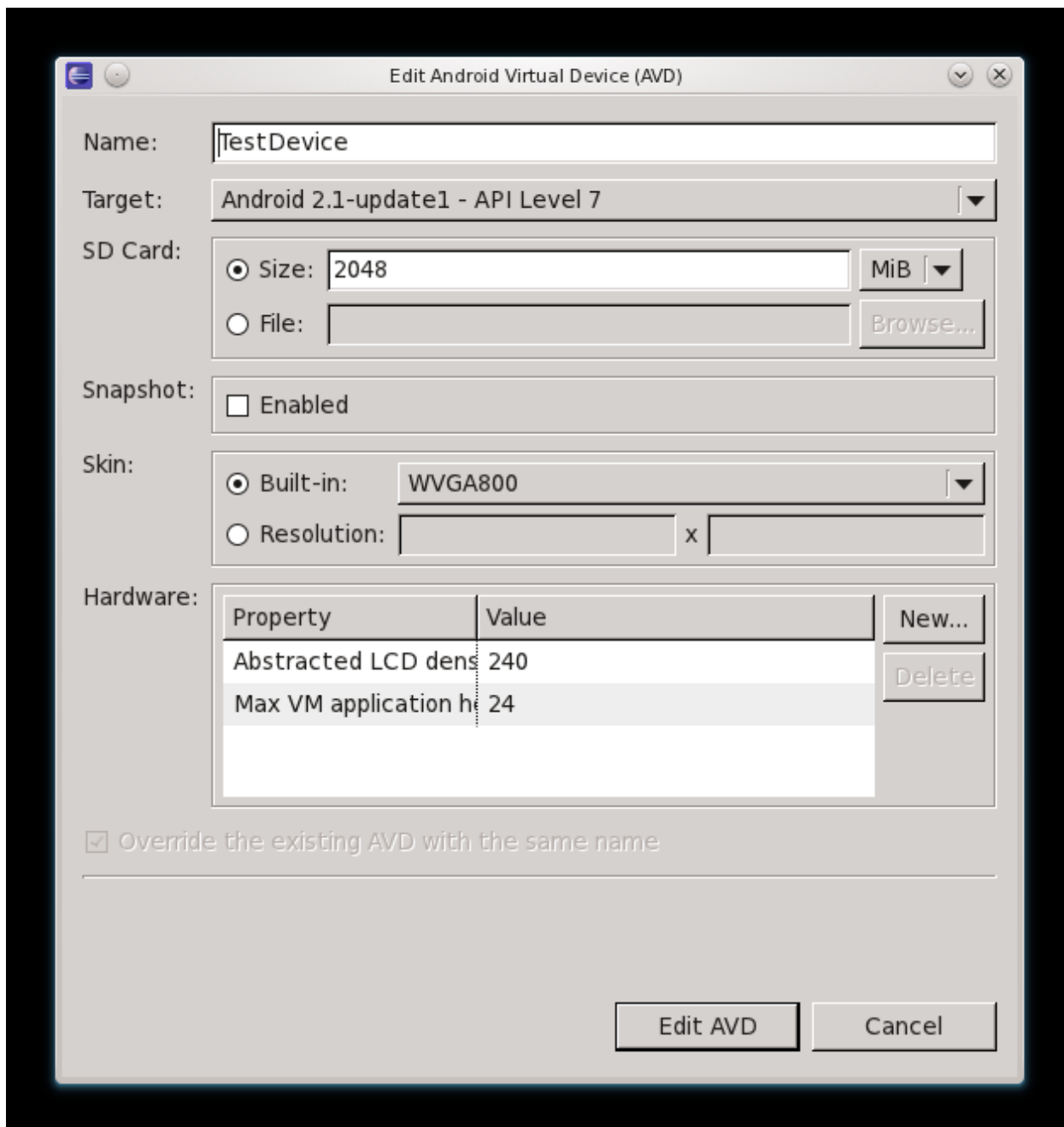
### **Creació dispositiu virtual**

El primer pas per a poder executar una aplicació Android es crear un emulador d'un dispositiu, per fer-ho obrir l'administrador de dispositius virtuals del SDK (AVD). En aquesta pantalla crearem un nou dispositiu amb les característiques necessàries per a l'execució de la nostra aplicació:



*Imatge 16: Creació del dispositiu virtual*

En aquest cas tenim dos dispositius creats, el primer conté la configuració recomanada per a l'execució de l'aplicació StatusDroid, que com ja es va detallar al disseny tècnic estarà basat en una versió base 2.1 d'Android el que es correspon amb l'API versió 7 tal i com es veu a la figura. Les opcions que s'hauran de seleccionar es poden veure a la següent captura:



Imatge 17: Configuració dispositiu virtual



Un cop que tenim el dispositiu virtual només l'hem d'executar per accedir a l'emulador d'Android.

### **Instal·lació de l'aplicació**

La instal·lació de l'aplicació s'ha de fer utilitzant la línia de comandaments del sistema on ens trobem. En el nostre cas l'aplicació es troba en un paquet d'Android anomenat StatusDroid.apk, per tant un cop hem assegurat que l'emulador està funcionant haurem d'executar la següent comanda per tal d'instal·lar l'aplicació:

#### **adb install StatusDroid.apk**

Si el procediment va bé veurem aparèixer la nostra aplicació al menú de l'emulador tal i com es pot veure a la següent captura de pantalla:

A partir d'aquest punt només ens queda executar l'aplicació instal·lada, per a més informació sobre la navegació i les funcionalitats de la mateixa consultar el document adjunt sobre el lliurament d'aquesta PAC.





Imatge 18: Aplicació instal·lada al dispositiu virtual



## 6 Conclusions

L'abast del projecte es centra en dos grans camps de gran expansió actualment com són les xarxes socials i les plataformes mòbils, per aquest fet s'ha hagut de limitar el desenvolupament a un seguit de característiques determinades ja que la limitació temporal feia impossible abastar la totalitat de possibilitats que ofereixen ambdós entorns.

Aquestes possibilitats són precisament el que ens va impulsar en primer moment a realitzar aquest projecte, fins i tot arribant a plantejar la possibilitat d'implementar un entorn de desenvolupament propi independentment d>StatusNet que implementes una xarxa social bàsica sobre la qual treballar.

A partir de les premisses inicials s'ha aconseguit no només una simple aplicació d'accés a les xarxes socials com es poden trobar desenes a les botigues d'aplicacions com la AppStore d'Apple o el mateix Marketplace d'Android, sino disposar d'un entorn complet de desenvolupament des de la part servidor fins al client a la plataforma Android.

Aquest darrer fet és el que dóna molta més rellevància al projecte ja que al tenir disponible tot el sistema aquest pot ser reutilitzat i ampliat com es vegi necessari, afegint noves funcionalitats tant a la part client, com a la part servidor, per tant hem creat una plataforma de desenvolupament sobre la qual es poden crear d'altres projectes.

Pel que fa al producte final que s'ha entregat m'ha servit per veure les possibilitats que ofereixen aquestes dues grans tecnologies actuals, per una banda StatusNet ens demostra que un cop solucionats els temes de comunicació bàsica estil Twitter que ofereix la plataforma, gràcies a la seva disponibilitat es poden afegir noves funcionalitats sense haver de desenvolupar costoses aplicacions sinó simples mòduls addicionals per a la plataforma.

Per altra banda, el desenvolupament d'aplicacions en un entorn com a Android m'ha fet veure un entorn al qual no havia accedit durant la carrera i considero que es molt interessant la visió que dóna del desenvolupament d'aplicacions tot el que fa referència a les noves plataformes mòbils i les seves funcionalitats. Val a dir, però, que també m'ha fet veure al mateix



temps les limitacions i problemàtiques amb les quals ens podem trobar alhora de desenvolupar per a aquests entorns.

## **6.1 Futurs desenvolupaments**

Si bé la plataforma i l'aplicació són perfectament funcionals per aquells elements que ofereixen degut a l'amplitud de la plataforma que hem utilitzat es poden afegir moltes més funcionalitats de les que finalment hem desenvolupat.

Per exemple seria interessant desenvolupar el fet de la geolocalització que ofereix StatusNet conjuntament amb la disponibilitat de dispositius GPS que trobem a les plataformes mòbils per enviar les dades de l'usuari i poder situar els missatges un un mapa, el que facilitaria per exemple enviar informació sobre llocs d'interés, restaurants, teatres, ..., o actes que siguin rellevants per determinats grups de persones, com poden ser localitzacions de concerts d'un determinat grup.

Aquesta característica podria donar molt joc en el futur, el que sumat al desenvolupament de l'API com a servei ens permetria segregat diferents característiques en aplicacions específiques que ofereixin funcionalitats destinades a diverses activitats.

Un altre element que podríem millorar és la integració amb el dispositiu, actualment l'aplicació tal i com està dissenyada es independent de la resta del sistema, seria força interessant afegir-hi un integració amb el llistat de contactes que ens permetés seguir a determinats usuaris a través d'aquest llistat de forma transparent, el que milloraria l'experiència d'usuari.

Un altre element en el qual podríem centrar-nos seria en el desenvolupament de la part servidor, afegint noves funcionalitats gràcies a les possibilitats d'ampliació que ens ofereix la plataforma StatusNet que hem muntat.

Podríem per exemple muntar un servidor StatusNet per a la UOC per facilitar la comunicació entre els estudiants, millorar la interacció i comunicació entre ells i al qual es pogués accedir des de diferent entorns com la web, els dispositius mòbils o fins i tot clients de missatgeria instantània ja que StatusNet suporta la comunicació mitjançant XMPP.



Sobre aquesta idea base es podrien afegir moltes funcionalitats que serien útils per als estudiants, avisos de dates d'entrega, comunicacions de notes o publicacions per part dels consultors, .... Qualsevol cosa que es pogués comunicar a través d'aquest sistema seria immediatament enviat a tots els usuaris.

Per tant creiem que donat el temps oportú tant la part client com la part servidor podrien oferir moltes possibilitats de desenvolupaments futurs, algunes de les quals creiem que es podrien aprofitar per millorar l'experiència d'usuaria de la UOC.



## 7 Bibliografia

- 1 StatusNet – Pàgina oficial: <http://status.net/>
- 2 Guia de desenvolupadors StatusNet: <http://status.net/wiki/Development>
- 3 API per desenvolupadors de Twitter: <http://developer.twitter.com/doc>
- 4 Android – Pàgina oficial: <http://www.android.com/>
- 5 Guia de desenvolupadors d'Android: <http://developer.android.com/index.html>
- 6 Quick Actions Dialog (Lorenz): <http://www.londatiga.net/it/how-to-create-quickaction-dialog-in-android/>
- 7 Android Icon Generator: <http://www.androidicongenerator.net/>
- 8 “Android User Interface Development”, Jason Morris. Packt Publishing 2010.
- 9 Jackson – JSON Processor: <http://jackson.codehaus.org/> (LGPL)



## 10 Llicència

(Llicència GPL )

Aquest treball està subjecte - excepte que s'indiqui el contrari- a una llicència GPL. Es pot copiar i distribuir el Programa (o un treball basat en ell, segons s'especifica en l'apartat 2, com a codi objecte o en format executable segons els termes dels apartats 1 i 2, suposat que a més compleixi una de les següents condicions:

1. Acompanyar-lo amb el codi font complet corresponent, en format electrònic, que ha de ser distribuït segons s'especifica en els apartats 1 i 2 d'aquesta Llicència en un medi habitualment utilitzat per a l'intercanvi de programes, o
2. Acompanyar-lo amb una oferta per escrit, vàlida durant almenys tres anys, de proporcionar a qualsevol tercera part una còpia completa en format electrònic del codi font corresponent, a un cost no major que el de realitzar físicament la distribució del font, que serà distribuït sota les condicions descrites en els apartats 1 i 2 anteriors, en un medi habitualment utilitzat per a l'intercanvi de programes, o
3. Acompanyar-lo amb la informació que vas rebre oferint distribuir el codi font corresponent. (Aquesta opció es permet només per a distribució no comercial i només si vostè va rebre el programa com a codi objecte o en format executable amb tal oferta, d'acord amb l'apartat 2 anterior).