

# Projecte Final de Carrera



## WheresApp

Aplicació per Android

**Joan Xatruch i Olivé**

Enginyeria Informàtica

Juny del 2012

**Consultor: Marc Domingo Prieto**



## Resum

Els dispositius mòbils han evolucionat d'una manera imprevisible des de la seva aparició fins a dia d'avui. Cada vegada es demanen més prestacions a un dispositiu tant petit que cap dins d'una butxaca i que per tant es té sempre a mà.

L'empresa responsable del cercador més usat del món, Google, va crear el sistema operatiu Android en el seu intent per introduir-se en el mercat dels dispositius mòbils, l'any 2005. L'Android està basat en Linux i enfocat a dispositius mòbils com telèfons intel·ligents, tauletes, o d'altres dispositius, amb limitats recursos de maquinari.

A dia d'avui, Android, ha aconseguit el 59% de quota de mercat de telèfons intel·ligents a nivell mundial.

Aquest projecte final carrera té com a objectiu el desenvolupament d'una aplicació per al sistema Android, aprofitant una de les prestacions que ofereixen els telèfons intel·ligents i que cada dia que passa és més important. Es tracta de la localització de la posició geogràfica del dispositiu.

L'aplicació desenvolupada al llarg d'aquest projecte utilitza el posicionament geogràfic juntament amb la connexió a Internet i permet a l'usuari conèixer, en temps real, la posició geogràfica de tots els seus contactes.

L'aplicació s'anomena WheresApp i incorpora una innovadora característica que permet als usuaris configurar la precisió amb la que volen ser trobats. Aquesta característica és la bombolla de protecció.

## Índex de continguts

<b>1. Introducció .....</b>	<b>6</b>
1.1. Justificació i context del projecte .....	6
1.2. Objectius .....	6
1.3. Descripció general .....	7
1.4. Planificació.....	8
1.4.1. Cicle de vida del projecte .....	8
1.4.2. Detall de les activitats .....	8
1.4.3. Temporització.....	10
1.5. Recursos i eines utilitzades .....	10
1.6. Productes obtinguts .....	11
1.7. Descripció dels capítols següents.....	12
<b>2. Anàlisi i Requeriments .....</b>	<b>13</b>
2.1. Requeriments funcionals.....	13
2.1.1. Gestió de l'aplicació.....	13
2.1.2. Alta al servei .....	13
2.1.3. Edició d'usuari .....	14
2.1.4. Baixa servei .....	14
2.1.5. Actualitzar llista contactes.....	14
2.1.6. Visualització dels contactes en llista .....	15
2.1.7. Visualització dels contactes sobre mapa .....	15
2.1.8. Mostra informació d'un contacte.....	16
2.2. Requeriments no funcionals .....	16
2.2.1. Maximitzar la usabilitat en la interfície d'usuari .....	16
2.2.2. Ús de funcionalitats pròpies dels dispositius mòbils .....	16
2.3. Diagrama de casos d'ús.....	17
<b>3. Disseny .....</b>	<b>19</b>
3.1. Visió general .....	19
3.2. Disseny del costat del client Android .....	20
3.2.1. Descripció de l'aplicació client.....	20
3.2.2. Disseny de classes .....	20
3.2.3. Interfície gràfica d'usuari .....	22

---

3.3.	Disseny del costat del servidor .....	26
3.3.1.	Infraestructura del Servidor .....	26
3.3.2.	Disseny de la base de dades.....	27
3.3.3.	Disseny de la interfície de comunicació (API) .....	28
<b>4.</b>	<b>Implementació .....</b>	<b>33</b>
4.1.	Costat del client Android.....	33
4.1.1.	Classe WheresAppActivity .....	33
4.1.2.	Classe LlistaContactesActivity .....	34
4.1.3.	Classe GestioDades .....	35
4.1.4.	Classe ContacteItemAdapter.....	35
4.1.5.	Classe ContacteInfoMapa .....	36
4.1.6.	Classe Contacte .....	37
4.1.7.	Classe GeneraContactes.....	37
4.1.8.	Classe ParserXML.....	38
4.1.9.	Classe ApiWEB.....	39
4.1.10.	Classe GeneradorXML.....	40
4.2.	Costat del servidor.....	40
4.2.1.	Implementació API web .....	40
4.2.2.	Implementació de l'accés a la base de dades .....	41
<b>5.</b>	<b>Possibles millores .....</b>	<b>42</b>
<b>6.</b>	<b>Conclusions .....</b>	<b>44</b>
<b>7.</b>	<b>Glossari .....</b>	<b>45</b>
<b>8.</b>	<b>Bibliografia.....</b>	<b>47</b>

## Índex de figures

Il·lustració 1: Cicle de vida del projecte .....	8
Il·lustració 2: Temporització de les tasques.....	10
Il·lustració 3: Temporització de les tasques.....	10
Il·lustració 4: Diagrama de casos d'ús.....	18
Il·lustració 5: Esquema general del sistema .....	20
Il·lustració 6: Pantalla alta a WheresApp .....	22
Il·lustració 7: Pantalla de llista de contactes.....	23
Il·lustració 8: Pantalla vista mapa.....	24
Il·lustració 9: Pantalla informació usuari.....	25
Il·lustració 10: Pantalla mapa vista satèl·lit.....	26
Il·lustració 11: Esquema infraestructura del servidor.....	27
Il·lustració 12: Base de dades .....	28
Il·lustració 13: Classe WheresAppActivity .....	33
Il·lustració 14: Classe LlistaContactesActivity .....	34
Il·lustració 15: Classe GestioDades .....	35
Il·lustració 16: Classe ContacteItemAdapter.....	35
Il·lustració 17: Classe ContacteInfoMapa .....	36
Il·lustració 18: Classe Contacte .....	37
Il·lustració 19: Classe GeneraContactes.....	37
Il·lustració 20: Classe ParserXML.....	38
Il·lustració 21: Classe ApiWEB.....	39
Il·lustració 22: Classe GeneradorXML .....	40

## 1. Introducció

### 1.1. Justificació i context del projecte

De totes les àrees disponibles per a la realització del projecte final de carrera vaig triar la de Xarxes de Computadors. Aquesta àrea m'interessava força donat que em permetia la possibilitat de desenvolupar una aplicació per a dispositius mòbils.

Vaig triar doncs la opció de desenvolupar una aplicació per Android, encara que no hi tenia cap experiència però em semblava força interessant, i motivant, el poder desenvolupar una aplicació per un dispositiu que, com molts altres usuaris, porto cada dia a sobre.

Després de varis dies de pensar sobre quina podria ser una aplicació interessant i a la vegada innovadora vaig decidir fer-ne una que permetés a un usuari conèixer la posició geogràfica de tots els seus contactes. Però el fet que els teus contactes puguin saber on ets en tot moment pot provocar, finalment, una certa reticència a usar l'aplicació. D'aquí esdevé la característica més innovadora que incorpora l'aplicació, que és la de permetre definir la precisió amb la que un usuari vol ser trobat pels seus contactes.

Així doncs, l'aplicació, anomenada WheresApp, permet saber, en temps real, on es troben tots els contactes de l'usuari, tot tenint en compte la Bombolla de protecció, la precisió definida per cada usuari.

### 1.2. Objectius

L'objectiu d'aquest projecte final de carrera és consolidar els coneixements adquirits durant els estudis d'Enginyeria Informàtica. El projecte, orientat en les xarxes de computadors, involucra varies tecnologies, algunes d'elles prou recents, per arribar a desenvolupar una aplicació per al sistema operatiu mòbil Android.

L'aplicació a desenvolupar s'anomena "WheresApp" i serveix per veure, en temps real, la distància a la que un usuari es troba de tots els seus contactes.

L'aplicació WheresApp incorpora un concepte innovador en el camp de la localització geogràfica que consisteix en que cada usuari



configura la distància mínima a la que vol ser localitzar pels seus contactes.

Aquesta distància mínima de protecció en el sistema WheresApp s'anomena bombolla de protecció i és una de les principals característiques d'aquest sistema de localització.

A més, també permet veure i localitzar la situació dels contactes sobre un mapa.

L'aplicació funciona permanentment en el dispositiu mòbil on s'hi instal·li així que permet en tot moment saber la posició dels nostres contactes tot tenint en compte la bombolla de protecció que cada usuari s'hagi configurat.

### **1.3. Descripció general**

L'aplicació WheresApp ha de permetre veure en tot moment i en temps real a quina distància es troba l'usuari dels seus contactes.

El correu de google de cada usuari serveix com a identificador al servei. Així doncs, només cal instal·lar-se l'aplicació per començar a utilitzar-la.

Una característica innovadora i interessant és la Bombolla de Protecció. La bombolla de protecció serveix per definir la distància mínima amb la que un usuari vol ser "trobat" pels seus contactes que també facin servir l'aplicació.

Aquesta Bombolla es pot definir especialment per cadascun dels contactes duals (els que un es té a l'altre) i una altra bombolla comuna pels contactes no duals.

Cada Bombolla pot prendre valors des de 0 (posicionament precís, sense mínims) fins al màxim. El valor màxim de bombolla simbolitza realment el no compartir dades de posicionament.

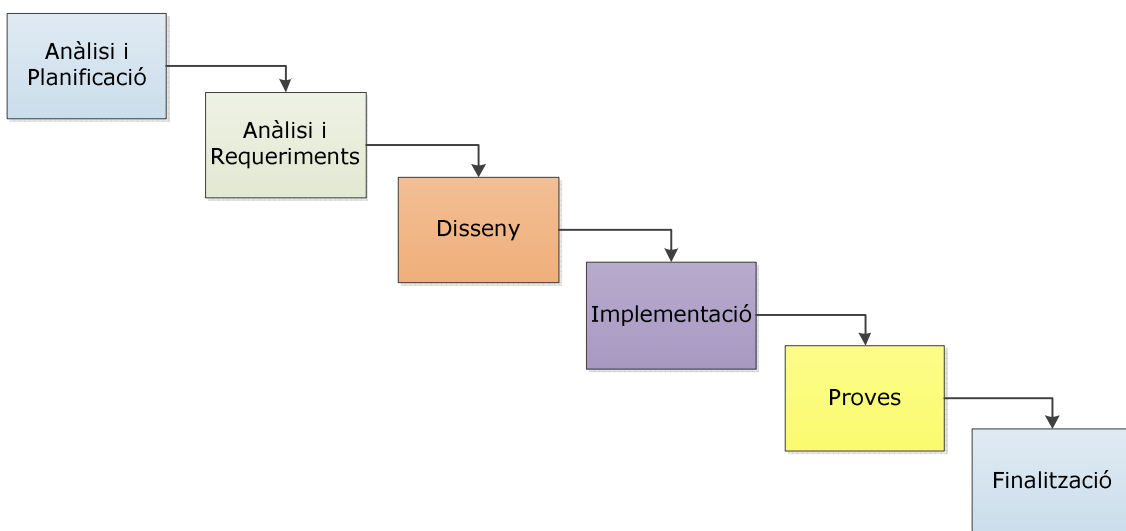
Es pot veure la posició de cada contacte sobre un mapa, tenint en compte la distància mínima que marca la bombolla de protecció. D'aquesta manera, un usuari amb una bombolla major de 0, podrà ser localitzat amb una precisió de la mida de la bombolla.

Pel funcionament de l'aplicació és necessària una part servidor amb una base de dades, que a través d'una API gestioni les peticions d'alta, i d'enviament i recepció de coordenades de cada contacte, tot basant-se en comunicacions segures.

## 1.4. Planificació

### 1.4.1. Cicle de vida del projecte

El projecte ha estat desenvolupat seguint el model de cicle de vida en cascada. Aquest model és útil especialment en aquests tipus de projectes donat que al admetre la possibilitat de ser iteratiu permet, per exemple, en un moment donat, fer modificacions al disseny i, una vegada revisades, poder passar a la següent etapa de nou.



Il·lustració 1: Cicle de vida del projecte

### 1.4.2. Detall de les activitats

Activitat	Descripció
<b>Anàlisi i Planificació</b>	
Definició del projecte	Definició de les característiques principals i abast del projecte.
Planificació del projecte	Planificació temporal de les tasques a dur a terme.
Elaboració del pla de treball	Elaborar el pla de treball enfocat per a ser entregat com la PAC 1.



Lliurament PAC 1	Lliurament del pla de treball elaborat.
<b>Anàlisi i Requisits</b>	
Definició dels requisits	Establir els requisits funcionals i no funcionals necessaris.
Preparació del software	Instal·lació i configuració del software necessari per al desenvolupament complet del projecte.
Estudi de la base de dades	Estudi del motor de bases de dades a utilitzar.
Estudi de la part servidor	Estudi del llenguatge a utilitzar per la part servidor. API i procediments.
Estudi de la interfície gràfica	Estudi i disseny esquemàtic de la interfície gràfica necessària per la funcionalitat definida.
<b>Disseny</b>	
Disseny de la base de dades	Disseny de les taules necessàries.
Disseny dels procediments del servidor	Dissenyar els procediments necessaris per la part servidor.
Disseny de la comunicació client – servidor	Disseny del protocol per comunicar els clients amb el servidor.
Disseny de classes aplicació Android	Disseny de les classes segons el funcionament de la plataforma Android.
<b>Implementació</b>	
Implementació de la part del servidor	Procediments. Càlcul distàncies i algorismes necessaris.
Implementació de l'aplicació client.	Implementació de l'accés al maquinari GPS, treball amb coordenades terrestres.
Implementació de l'aplicació client.	Implementació de la capa de xarxa de comunicació amb el servidor.
Implementació de l'aplicació client.	Implementació de la interfície gràfica d'usuari.
<b>Proves</b>	
Proves globals	Proves globals i correcció d'errors.
Proves reals	Proves en entorns reals utilitzant dispositius mòbils Android físics.
<b>Finalització</b>	
Memòria	Acabar els retocs finals a la memòria.
Vídeo de presentació	Elaborar vídeo de presentació i demostració real del projecte.
Lliurament final	Lliurament final de la memòria i

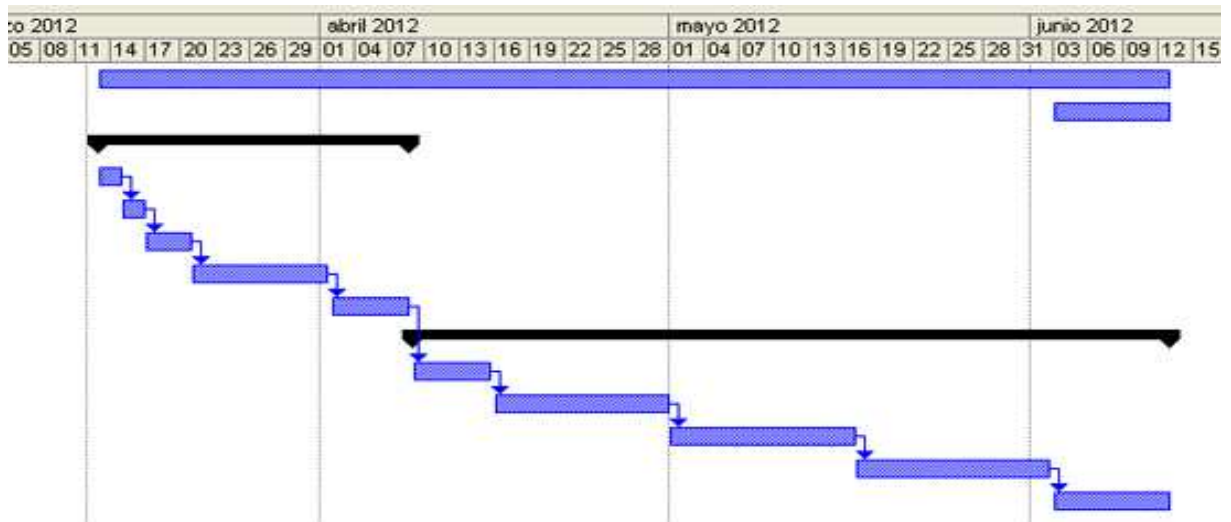


vídeo de presentació.

### 1.4.3. Temporitziació

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Elaboració memòria	92 dies	mar 13/03/12	mar 12/06/12	
2	Elaboració presentació	9,88 dies	dom 03/06/12	mar 12/06/12	
3	<b>Part Servidor</b>	<b>27 dies</b>	<b>mar 13/03/12</b>	<b>dom 08/04/12</b>	
4	Estudi i Decisió bbdd	2 dies	mar 13/03/12	mié 14/03/12	
5	Estudi i Decisió llenguatge api	2 dies	jue 15/03/12	vie 16/03/12	4
6	Disseny BBDD	4 dies	sáb 17/03/12	mar 20/03/12	5
7	Implementació API i algorismes	12 dies	mié 21/03/12	dom 01/04/12	6
8	Proves globals i correcció errors	7 dies	lun 02/04/12	dom 08/04/12	7
9	<b>Part Client Android</b>	<b>65 dies</b>	<b>lun 09/04/12</b>	<b>mar 12/06/12</b>	
10	Instal·lació, configuració Eclipse	7 dies	lun 09/04/12	dom 15/04/12	8
11	Implementació de la GUI	15 dies	lun 16/04/12	lun 30/04/12	10
12	Implementació accés hard (gps)	16 dies	mar 01/05/12	mié 16/05/12	11
13	Implementació comunicació amb servid	17 dies	jue 17/05/12	sáb 02/06/12	12
14	Proves globals i correcció errors	10 dies	dom 03/06/12	mar 12/06/12	13

II-lustració 2: Temporitziació de les tasques



II-lustració 3: Temporitziació de les tasques

### 1.5. Recursos i eines utilitzades

Per a la realització d'aquest projecte s'han utilitzat el següents recursos.

- Ordinador personal
- Servidor WEB Apache amb component PHP



- Base de dades MySQL
- Mòbils Android per a les proves de l'aplicació
  - HTC Desire
  - HTC Desire S
  - Samsung Galaxy 3
- Software lliure per al desenvolupament d'aplicacions per Android.
  - Programari Eclipse amb els components corresponents de Google per a desenvolupar per aplicacions per Android.
  - Editor per a la codificació del codi per a la part del servidor.
- API necessàries desenvolupades per Google per interactuar a través d'Android.
  - API de Google Maps
- Software privatiu:
  - Ms. Word per a la redacció de la memòria.
  - Ms. Visió per a l'edició dels elements gràfics.
  - Ms. PowerPoint per l'elaboració de la presentació.
  - Adobe Photoshop per l'elaboració d'elements gràfics de l'aplicació Android.
  - CamStasia Studio, com a editor de vídeo, per elaborar el vídeo de la presentació virtual.
- Càmera de vídeo per enregistrar la demostració real i adjuntar-lo al vídeo de la presentació virtual.

### **1.6. Productes obtinguts**

Els productes obtinguts per al correcte funcionament del projecte inclouen els de la part que s'executa en el servidor i els de la part que s'executa en el client, l'aplicació Android.



Per a la part del servidor s'obtenen:

- Arxius de codi font codificats en llenguatge PHP que executarà el servidor web Apache per a atendre les peticions de les aplicacions clients Android.
- Arxius necessaris per a construir i configurar la bases de dades MySQL.
- Arxius de codi font en java i binaris per poder instal·lar i executar l'aplicació en un mòbil Android.
- Memòria del projecte.
- Video de presentació virtual.

### **1.7. Descripció dels capítols següents**

Al llarg dels següents capítols es detallaran les etapes següents per al desenvolupament d'aquest projecte. Aquestes inclouen la d'estudiar els requeriments del projecte, el disseny, la implementació i les proves realitzades.

El capítol 2, que tracta l'anàlisi i els requeriments detalla tots els aspectes necessaris que s'han de poder fer des de l'aplicació Android. La instal·lació de l'aplicació, la gestió del propi usuari donat d'alta, i la visualització dels contactes, tant en llista, com sobre un mapa.

El capítol 3, que detalla el Disseny de tot el sistema, es divideix en dues parts prou diferenciades. Per una banda, el disseny de l'aplicació Android, de la que se'n detallen les classes necessàries pel seu funcionament, així com també la interfície gràfica d'usuari, pensada en tot moment per maximitzar la usabilitat.

Respecte el costat del servidor es detalla la infraestructura necessària per dur a terme les tasques requerides. El disseny de la base de dades. I per últim el disseny de les funcions necessàries per a que les aplicacions clients Android es puguin comunicar amb el servidor, a través de crides sobre el protocol HTTP.

El capítol 4 detalla tota la implementació de les parts destacades d'aquest projecte. Les classes necessàries per a l'aplicació Android i els procediments de la part del servidor que han de permetre a les aplicacions compartir dades entre elles i en temps real.



El capítol 5 recull un seguit de possibles millores a realitzar sobre aquest projecte. Millores que en alguns dels casos sortien fora de l'abast d'aquest projecte.

I per últim, sense tenir en compte els capítols de glossari i bibliografia, el capítol 6 relata les conclusions generals del projecte.

## 2. Anàlisi i Requeriments

### 2.1. Requeriments funcionals

#### 2.1.1. Gestió de l'aplicació

L'aplicació està compactada en un arxiu APK que el sistema Android entén i és capaç d'instal·lar l'aplicació que conté al sistema i també de desinstal·lar-la.

Una vegada l'aplicació s'instal·la es configura automàticament per a que s'executi cada vegada que es reinicia el sistema. A més a més l'aplicació és de tipus permanent, és a dir, el sistema sempre la mantindrà entre els processos que estant permanentment en execució.

És la pròpia aplicació qui gestiona eficientment els recursos del dispositiu segons si s'està executant o si està pausada, però sempre romandrà a la memòria ram del dispositiu mòbil.

#### 2.1.2. Alta al servei

La primera vegada que s'executa l'aplicació es demana a l'usuari procedir a donar-se d'alta al sistema.

Aquest procediment inclou que l'usuari escrigui el seu nom. Triï una de les adreces de correu vinculades al sistema Android i configuri les distàncies de les bombolles de protecció que desitja. S'han de configurar dues bombolles de protecció. Una per als contactes duals, els que un es té a l'altra, i una altra bombolla per tots els altres contactes.

Les adreces de correu entre les que pot escollir l'usuari a l'hora de donar-se d'alta són només les que estan vinculades com a compte google del dispositiu del que es tracta. Tant sigui com a compte primària o secundària però només se li permet triar



d'entre les vinculades. Com que cada dispositiu Android ha de tenir per força una compte vinculada sempre n'hi haurà almenys una de disponible.

### **2.1.3. Edició d'usuari**

L'usuari pot editar les dades introduïdes a l'hora de donar-se d'alta. Pot corregir el seu nom en cas de que hagi comés algun error.

El més important, però, és l'edició de les distàncies de les dues bombolles de protecció les quals poden ser modificades en qualsevol moment.

Qualsevol modificació de les dades de l'usuari s'envien directament al servidor que en farà ús en les properes peticions.

### **2.1.4. Baixa servei**

Es permet que l'usuari pugui donar-se de baixa del servei mitjançant una opció del menú de l'aplicació. Aquesta opció comunica al servidor que executi els procediments necessaris per donar de baixa a l'usuari. S'esborren totes les dades del servidor referents a l'usuari i després s'esborren tots els registres i configuracions emmagatzemats al dispositiu Android que fan referència a l'usuari actual.

A partir d'aquest moment si l'usuari tornar a executar l'aplicació se li tornarà a demanar l'acció de l'alta al sistema.

### **2.1.5. Actualitzar llista contactes**

Aquest procediment s'executa automàticament durant l'alta al sistema. Tot i això es disposa d'una opció específica en el menú de l'aplicació que permet reenviar els contactes de l'usuari cap al servidor.

L'aplicació llegeix tots els contactes de l'usuari emmagatzemats al dispositiu mòbil i envia al servidor només aquells que disposen d'una adreça de google tipus nom@gmail.com. Només s'envien aquests contactes per assegurar des d'aquest moment



que els contactes que també facin servir aquesta aplicació siguin usuaris d'Android i per conseqüència tinguin una compte de google vinculada als seus dispositius.

### **2.1.6. Visualització dels contactes en llista**

Una de les funcionalitats principals de l'aplicació WheresApp és la de mostrar la distància a la que un usuari es troba de tots els seus contactes. Dues vistes principals permeten visualitzar aquesta informació. En una llista veurem tots els nostres contactes que estiguin donats d'alta al sistema i facin servir l'aplicació.

En aquesta llista veurem tots els contactes amb la seva informació i la distància en metres o quilòmetres a la que ens trobem d'ells.

Aquesta llista proporciona certa interacció de manera que polsant sobre un contacte en concret es canviarà la vista i es mostrarà un mapa centrat a la posició on es troba el contacte en qüestió.

Durant el transcurs del temps aquesta llista s'anirà actualitzant automàticament en temps real mostrant l'evolució geogràfica dels contactes.

### **2.1.7. Visualització dels contactes sobre mapa**

La visualització dels contactes sobre mapa permet veure i localitzar la posició actual dels contactes de l'usuari.

Aquesta vista d'usuaris sobre el mapa permet fer ampliacions i reduccions per concretar la posició d'algun contacte. També permet canviar el fons del mapa, del típic vectorial mostrant carrers i carreteres al mapa amb fons d'imatges reals captades per satèl·lit i viceversa.

Si el contacte té una distància de bombolla superior a 0 veurem com la seva posició canvia aleatòriament sobre el mapa encara que sempre dins de la seva bombolla de protecció, sigui pels contactes duals o per tots els altres contactes.



### **2.1.8. Mostra informació d'un contacte**

En qualsevol moment podem veure tota la informació d'un contacte en concret polsant sobre la seva icona dibuixada en el mapa.

Podrem saber, a part del nom, l'adreça del contacte així com les distàncies de les seves bombolles de protecció.

## **2.2. Requeriments no funcionals**

### **2.2.1. Maximitzar la usabilitat en la interfície d'usuari**

La interfície gràfica d'usuari de l'aplicació Android ha de ser molt intuïtiva i fàcil d'utilitzar per a qualsevol usuari.

S'ha de tenir en compte, durant el disseny, que hi poden haver dispositius mòbils amb diferents dimensions de pantalla i l'aplicació ha de poder adaptar-se a tots els casos possibles. Només d'aquesta manera es garanteix un ús correcte de l'aplicació en qualsevol usuari d'un dispositiu Android.

### **2.2.2. Ús de funcionalitats pròpies dels dispositius mòbils**

L'aplicació ha de fer ús de les eines que ofereix la llibreria de desenvolupament per Android per aprofitar totes les característiques pròpies dels dispositius mòbils.

La interfície gràfica d'usuari és accessible gairebé en totes les funcionalitats de l'aplicació a través de la pantalla tàctil del dispositiu. Només el menú de configuració de l'aplicació és accessible a través del botó menú que precisament té aquesta funció en tots els dispositius mòbils Android.

Donat que WheresApp és un tipus d'aplicació que està permanentment en execució encara que l'usuari no la tingui visible en pantalla, s'ha de tenir en compte varis aspectes:

El consum de dades s'ha de minimitzar quan l'aplicació estigui en estat de pausa, tenint en compte de no perdre al funcionalitat característica d'una aplicació de temps real.



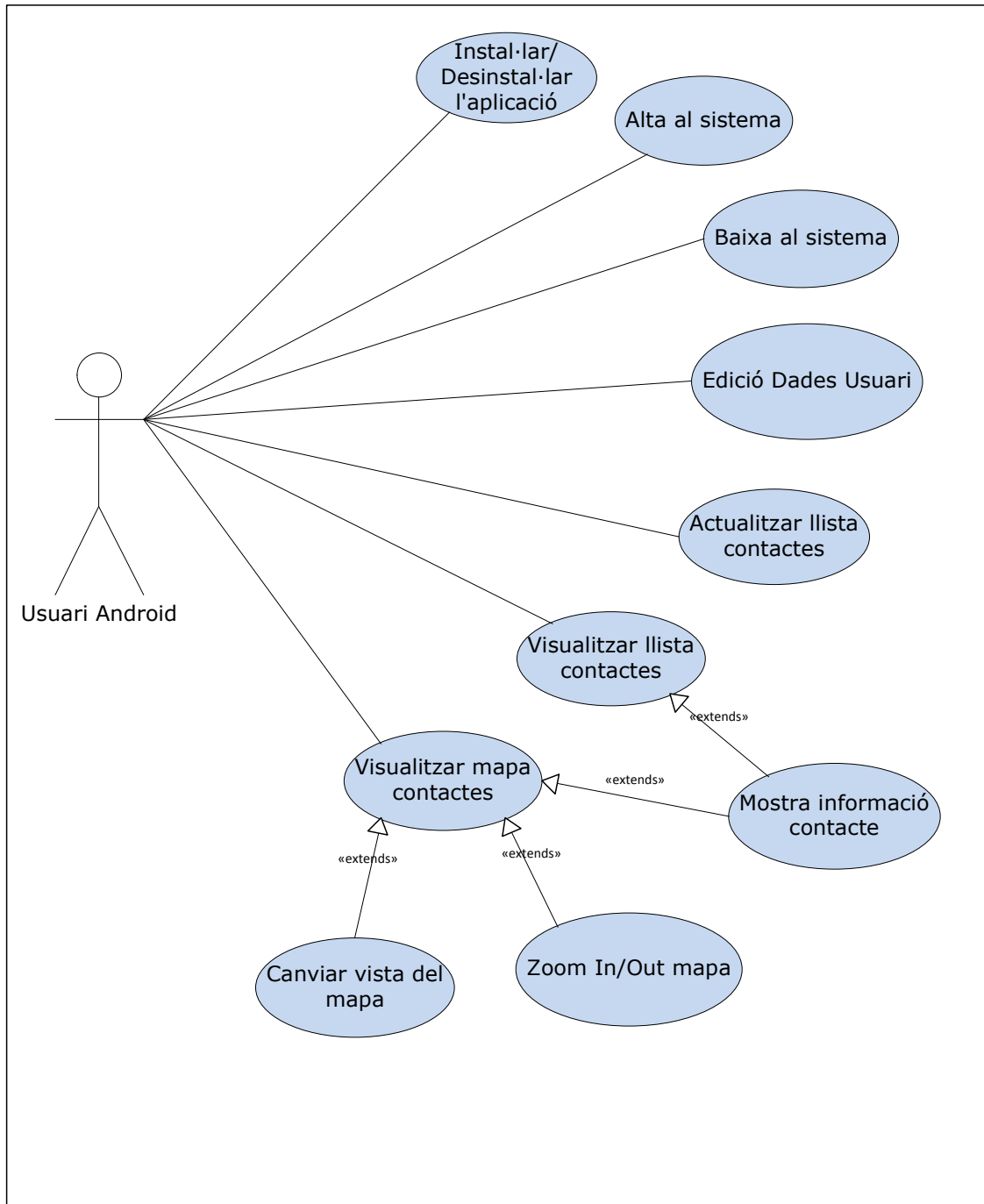


L'accés al maquinari GPS del dispositiu té un consum elèctric elevat. Això implica que la cadència d'actualització de la posició geogràfica del dispositiu depèn de l'estat de l'aplicació. La posició s'actualitza més sovint si l'aplicació es troba visible en primer pla, de igual manera s'actualitzen més sovint la posició dels contactes de l'usuari.

Els dispositius Android també ofereixen la possibilitat de posicionament geogràfic a partir de dades de posició de xarxes sense fils, consumint molt menys que el maquinari GPS. Per tant si el posicionament per xarxes sense fils és prou fiable no caldrà fer peticions al maquinari GPS, allargant la duració de la bateria.

### **2.3. Diagrama de casos d'ús**

El següent diagrama recull el resultat de l'etapa d'anàlisi del sistema. En aquest diagrama de casos d'ús hi observem un únic actor que interactua amb l'aplicació i n'utilitza totes les seves funcionalitats.



Il·lustració 4: Diagrama de casos d'ús



## 3. Disseny

### 3.1. Visió general

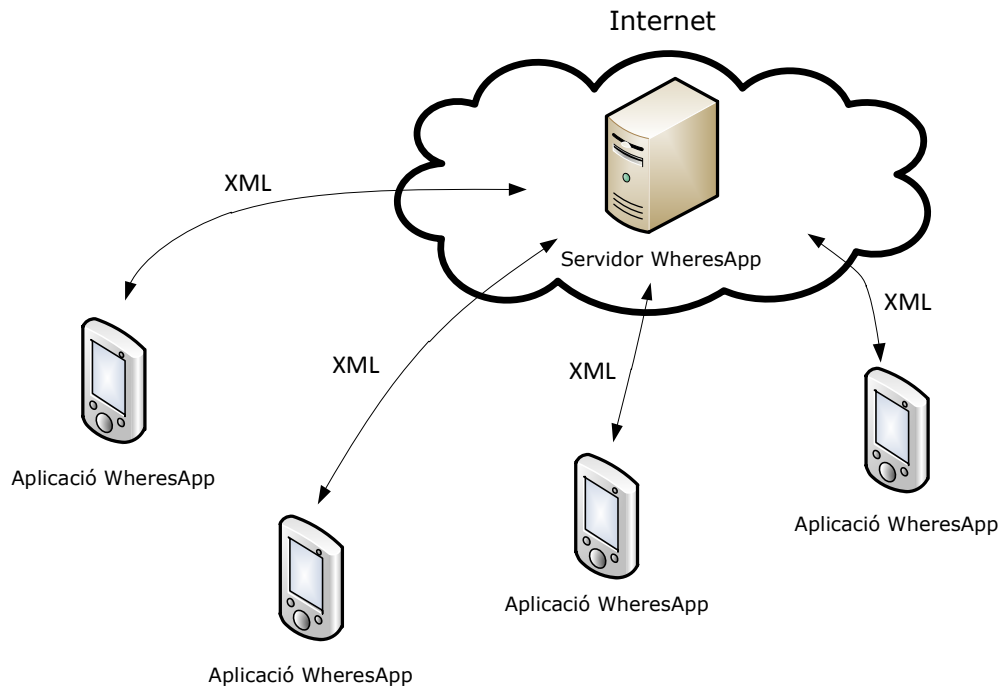
L'arquitectura general del sistema WheresApp es basa en el model client-servidor. Es distingeixen doncs dues parts ben diferenciades: El costat del client i el costat del servidor.

El servidor serà l'encarregat d'atendre totes les peticions de totes les aplicacions client. El servidor és l'únic que té accés a la base de dades del sistema. Tots els accessos a la base de dades només es faran després de comprovar que l'usuari que fa la petició és qui diu ser.

Les aplicacions clients s'executen sobre els dispositius mòbils Android i llencen peticions cap al servidor a través de la interfície de comunicació (API) que defineix el servidor i si tot és correcte envia la resposta de la petició cap a l'aplicació client.

El servidor defineix un conjunt de funcions a través d'una API. A través d'aquestes funcions, les aplicacions clients, podran donar d'alta al sistema a un usuari, donar-lo de baixa, editar les dades de l'usuari, demanar les dades de posicionament dels contactes d'un usuari, i comunicacions definides internament a l'aplicació. Totes les funcions comproven la contrasenya de l'usuari fet que dota al sistema de certa seguretat per evitar suplantacions d'identitat.

En el següent esquema es mostra l'estructura general del sistema.



**Il·lustració 5: Esquema general del sistema**

## 3.2. Disseny del costat del client Android

### 3.2.1. Descripció de l'aplicació client

L'aplicació Android WheresApp és la part visible per l'usuari.

Des d'ella s'interactua amb les diferents funcions que ofereix el sistema, posicionament dels contactes i funcions específiques donat un contacte concret.

### 3.2.2. Disseny de classes

En el següent diagrama es mostra el disseny de classes de l'aplicació WheresApp. S'han tingut en compte que els procediments d'accés a la xarxa o als dispositius de maquinari intern sempre s'han de fer des de fils d'execució diferents al fil que executa la interfície gràfica.



### 3.2.3. Interfície gràfica d'usuari

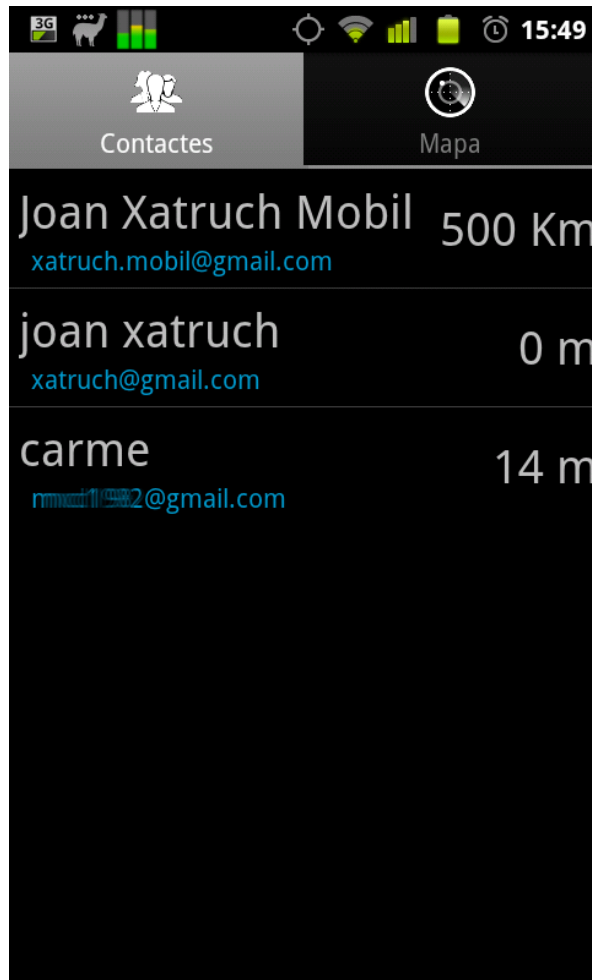


**Il·lustració 6: Pantalla alta a WheresApp**

Aquesta és la primera pantalla que veu l'usuari. Des d'ella es procedeix a donar-se d'alta al sistema. També serveix per editar les dades introduïdes prèviament en el moment de donar-se d'alta.

S'hi ha d'escriure el nom de l'usuari. S'ha d'escollir entre les adreces que es donen com a opció i configurar les bombolles de protecció que es desitgen.

L'adreça de correu no es permet escollir perquè l'aplicació llegeix al dispositiu mòbil quines són les adreces vinculades al compte de google. D'aquesta manera es dota al sistema d'una certa protecció.



**Il·lustració 7: Pantalla de llista de contactes**

Aquesta pantalla és una de les vistes disponibles amb WheresApp per veure la informació dels usuaris. La vista de llista de contactes mostra el nom del contacte, la seva adreça electrònica amb la que s'ha donat d'alta i la distància a la que es troba l'usuari de l'aplicació de cada contacte. Les distàncies es mostren sempre tenint en compte la bombolla de protecció. La llista presenta certa interacció tàctil, polsant sobre un contacte canvia a mode mapa mostrant la posició del contacte que hem polsat.



**Il·lustració 8: Pantalla vista mapa**

Aquesta pantalla és la vista de mapa dels contactes. Sobre el mapa s'hi veu la posició de tots els contactes. Si el contacte té un bombolla major de 0 es veurà com la posició va canviant aleatòriament sempre mantenint-se dintre de la bombolla que té configurada. El botó de dalt a l'esquerra permet canviar de vista mapa a vista satèl·lit i viceversa.





**Il·lustració 9: Pantalla informació usuari**

Polsant a sobre de qualsevol usuari s'obrirà una finestra mostrant la informació de l'usuari. Podem veure'n el nom, l'adreça i la bombolla que té configurada. Si l'usuari i el contacte son contactes duals veurem la bombolla dual i si no, veurem la bombolla genèrica.



**Il·lustració 10: Pantalla mapa vista satèl·lit**

Vista de satèl·lit del mateix mapa mostrant els contactes igualment i amb la mateixa funcionalitat que amb la vista de mapa.

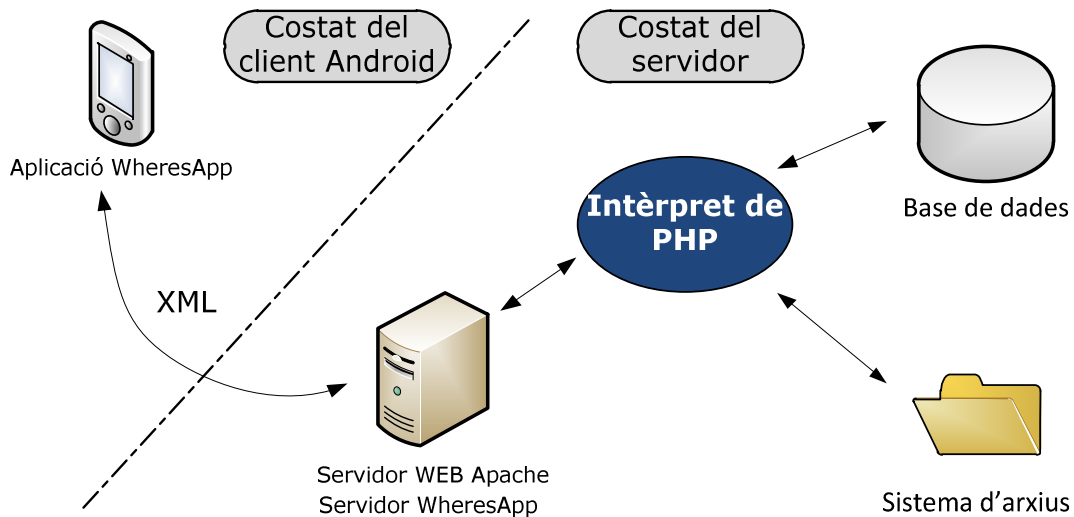
### **3.3. Disseny del costat del servidor**

#### **3.3.1. Infraestructura del Servidor**

El costat del servidor està basat en una sèrie de tecnologies que en conjunt formen un servidor LAMP. Un servidor LAMP inclou Linux com a sistema operatiu, Apache com a servidor web, MySQL com a motor de bases de dades i PHP com a llenguatge de programació interpretat.

Per tal de donar servei a l'aplicació WheresApp es necessita un servidor LAMP complet. L'aplicació client "parlarà" amb el servidor mitjançant una API programada en PHP que s'executarà sobre el servidor web Apache. Les funcions d'aquesta API són les encarregades de fer els accessos necessaris a la base de dades, actualitza-la convenientment segons demanin les aplicacions client.

El següent diagrama mostra la infraestructura física del servidor i com treballen en conjunt el servidor web Apache, l'interpret PHP, la base de dades i el sistema d'arxius del sistema.



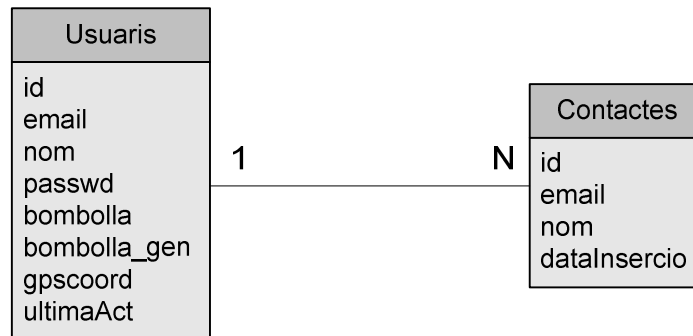
**Il·lustració 11: Esquema infraestructura del servidor**

### 3.3.2. Disseny de la base de dades

El motor de base de dades encarregat de gestionar tot el sistema és MySQL.

La base de dades conté per cada usuari, la seva posició GPS actual, els contactes relacionats amb l'usuari, mes algunes característiques especials de l'usuari.

La base de dades està formada per les següents taules:

**Il·lustració 12: Base de dades**

Amb aquestes taules se'n té prou per emmagatzemar tot el que el sistema WheresApp necessita per treballar.

La taula d'Usuaris emmagatzema tots els usuaris donats d'alta al sistema. D'ells se'n guardarà l'adreça electrònica, el nom real, les bombolles de protecció i les últimes coordenades GPS enviades pel dispositiu.

Per una altre costat la taula de Contactes contindrà tots els contactes de tots els usuaris. Se'n guardarà l'adreça electrònica, el nom i la data d'inserció.

### 3.3.3. Disseny de la interfície de comunicació (API)

El costat de servidor del sistema WheresApp, necessita una API que permeti als clients Android accedir a la base de dades per tal de fer-hi modificacions com ara, actualitzar la seva posició actual, o bé per donar-se d'alta com a nou usuari, o ja sigui per actualitzar els contactes, etc. Per fer aquesta feina es necessita una API. D'aquesta manera totes les aplicacions client accediran d'una manera segura i controlada a les dades del sistema.

El llenguatge escollit per a la implementació d'aquesta API és PHP. PHP és un llenguatge interpretat i encaixa perfectament en la funció que es vol per a aquest projecte.

La implementació de l'API és exclusiva per a aquest projecte i fonamentalment respondrà a un format XML establert prèviament.



Les crides a l'API es fan mitjançant operacions HTTP POST a una URL del tipus `http://www.wheresapp.com/api`. En l'operació POST se li ha de passar l'XML amb la informació de l'acció a realitzar.

Accions privades de l'API:

- **Donar-se d'alta al sistema**

Petició d'alta d'un nou usuari al sistema.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<WheresAppXML>  
  <function>ALTA/EDIT</function>  
  <values>  
    <name>NOM DE L'USUARI</name>  
    <email>EMAIL DE L'USUARI</email>  
    <passwd>CONTRASENYA GENERADA</passwd>  
    <bubble>ENTER</bubble>  
    <bubble_generica>ENTER</bubble_generica>  
  </values>  
</WheresAppXML>
```

Resposta del servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<WheresAppXML>  
  <function>ALTA</function>  
  <values>  
    <response>OK</response>  
  </values>  
</WheresAppXML>
```

- **Actualitzar posició GPS**

L'aplicació client ha d'actualitzar la seva posició gps cada cert temps. La cadència d'aquesta actualització la determinarà el client segons uns criteris establerts. Per exemple, l'actualització ha de ser més sovint si el client



s'està movent ràpidament que si s'està a un punt fix durant molta estona.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<WheresAppXML>
  <function>SET-GPS-COORD</function>
  <values>
    <email>EMAIL DE L'USUARI</email>
    <passwd>CONTRASENYA</passwd>
    <gpscoord>Lat,Long</gpscoord>
  </values>
</WheresAppXML>
```

Resposta del servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<WheresAppXML>
  <function>SET-GPS-COORD</function>
  <values>
    <response>OK</response>
  </values>
</WheresAppXML>
```

- **Actualitzar contactes**

Després de la petició d'alta i cada vegada que hi hagi algun canvi en els contactes locals del client es farà una petició d'actualització dels contactes. S'enviaran els contactes nous cap al servidor de manera que sempre siguin els mateixos que hi ha al dispositiu.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<WheresAppXML>
  <function>SET-CONTACTS</function>
  <values>
    <email>EMAIL DE L'USUARI</email>
```



```
<passwd>CONTRASENYA</passwd>
<contact>
  <email>EMAIL DEL CONTACTE 1</email>
  <name>NOM DEL CONTACTE 1</name>
</contact>
<contact>
  <email>EMAIL DEL CONTACTE 2</email>
  <name>NOM DEL CONTACTE 2</name>
</contact>
</values>
</WheresAppXML>
```

Resposta del servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WheresAppXML>
  <function>SET-CONTACTS</function>
  <values>
    <response>2</response>
  </values>
</WheresAppXML>
```

- **Demandar posició GPS dels contactes**

Amb aquesta petició, cridada pel client cada cert temps, cada usuari podrà conèixer la posició dels seus contactes tot tenint en compte la bombolla de protecció de cadascun d'ells.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<WheresAppXML>
  <function>GET-CONTACTS-GPS-COORD</function>
  <values>
    <email>EMAIL DE L'USUARI</email>
    <passwd>CONTRASENYA</passwd>
  </values>
</WheresAppXML>
```

Resposta del servidor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<WheresAppXML>
  <function>GET-CONTACTS-GPS-COORD</function>
  <values>
    <response>2</response>
    <contact>
      <email>EMAIL DEL CONTACTE 1</email>
      <name>NOM DEL CONTACTE 1</name>
      <gpscoord>Lat,Long</gpscoord>
      <bubble>ENTER</bubble>
    </contact>
    <contact>
      <email>EMAIL DEL CONTACTE 2</email>
      <name>NOM DEL CONTACTE 2</name>
      <gpscoord>Lat,Long</gpscoord>
      <bubble>ENTER</bubble>
    </contact>
  </values>
</WheresAppXML>
```

Les respostes del servidor són sempre en format XML i seguint l'estructura establerta. Els XML mal formats que rebí el servidor mai seran processats i es retornarà un error del tipus:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<WheresAppXML>
  <function>ERROR</function>
  <values>
  </values>
</WheresAppXML>
```

La primera vegada que s'instal·la l'aplicació client a l'Android es genera una contrasenya i es guarda dins del dispositiu.





Aquesta contrasenya serveix per validar que les operacions demanades realment son fetes per l'usuari que diu ser.

## 4. Implementació

### 4.1. Costat del client Android

#### 4.1.1. Classe WheresAppActivity

<b>WheresAppActivity</b>
com.xatruch.wheresapp
+WheresAppActivity():void
+carregarActivitatLlistaContactes():void
+generaContrasenyaSegura():String
+mostrarMissErrorCom():void
+onClickLL(view:View):void
+onCreate(savedInstanceState:Bundle):void
+pasSeguent(view:View):void

#### II-lustració 13: Classe WheresAppActivity

Aquesta classe extén la funcionalitat de la classe Activity d'Android. En Android, una activitat (en anglès, Activity) és el que permet la interacció entre l'usuari i el dispositiu. Així doncs, les classes Activity són les que s'encarreguen de crear les finestres en les que s'hi posen els diferents elements de la interfície gràfica d'usuari.

Tota classe que extén la classe Activity ha d'implementar com a mínim la funció onCreate() que s'executa quan es crea l'activitat. En el cas de la classe WheresAppActivity s'executa justament quan s'arranca l'aplicació WheresApp. La funció onCreate() llegeix l'estructura de la interfície gràfica des d'un arxiu XML vinculat i crea tots els components gràfics dins la finestra, que, en Android, sol ser en mode pantalla completa.

Una vegada carregada la interfície gràfica l'activitat resta a l'espera dels events que s'hi produeixin. La classe WheresAppActivity permet configurar el nom de l'usuari, l'adreça de correu i les bombolles de protecció. Seguidament polsant sobre el botó "Següent" es procedeix a actualitzar al

servidor les dades introduïdes i es llença la següent activitat LlistaContactesActivity.

Si es tracta d'una alta nova al sistema, al pulsar el boto "Següent" es genera una contrasenya aleatòria (basada en la codificació a caràcters alfanumèrics d'un número generat aleatòriament de 35 bits) i es procedeix a realitzar l'alta al sistema.

#### 4.1.2. Classe LlistaContactesActivity

<b>LlistaContactesActivity</b>
com.xatruch.wheresapp
+handlerActualitzarInfoMapa:Handler
+LlistaContactesActivity():void
+mostrarMissErrorCom():void
+onCreate(savedInstanceState:Bundle):void
+onCreateOptionsMenu(menu:Menu):boolean
+onOptionsItemSelected(item:MenuItem):boolean
+onPause():void
+onRestart():void
+onclickboto(view:View):void

**Il·lustració 14: Classe LlistaContactesActivity**

Aquesta classe també és una activitat. És l'activitat principal de l'aplicació ja que és l'encarregada de gestionar la llista de contactes y el mapa de l'aplicació. El canvi d'una pantalla a l'altra es fa mitjançant pestanyes situades a la part superior de l'aplicació.

Aquesta classe dóna la funcionalitat de mostrar la llista rebuda per la classe GestioDades i de gestionar les polsacions fetes sobre els contactes ja sigui sobre la llista o sobre el mapa.

A més d'implementar la funció onCreate() de la classe Activity és necessari implementar les funcions onPause() i onRestart() per tenir l'opció d'optimitzar l'ús dels recursos de posicionament geogràfic tant a través de xarxes wifi com del GPS del dispositiu.



### 4.1.3. Classe GestioDades

#### GestioDades

com.xatruch.wheresapp

+llistaContactes: List<Contacte>

+GestioDades(context: Context, listView: ListView, handlerActualitzarInfoMapa: Handler, locationManager: LocationManager, handleErrorMiss: Handler): void

+CridaAsincronaContinua(tempoEntreRepeticions: int): void

+CridaAsincronaContinuaPosGPS(email: String, passwd: String, gpsCoord: String): void

+getLlistaContactes(): ArrayList<Contacte>

#### II-lustració 15: Classe GestioDades

Aquesta classe és l'encarregada de gestionar totes les dades que van i vénen del servidor de WheresApp. Les comunicacions amb el servidor sempre són en un fil d'execució diferent al fil que executar la interfície gràfica, d'aquesta manera la interfície no es bloqueja mentre s'espera que la comunicació acabi. Una vegada acabada, les dades obtingudes s'han de passar al fil d'execució de la interfície per a que actualitzi les dades mostrades per pantalla.

Així doncs, és en aquesta classe on es gestionen tots els threads que s'encarreguen de totes les tasques que es realitzen en segon pla, sense interferir a la interfície gràfica.

Algunes d'aquestes tasques de segon pla envien o reben dades d'internet i per fer això es necessari un permís especial especificat dins del manifest del projecte.

### 4.1.4. Classe ContacteItemAdapter

#### ContacteItemAdapter

com.xatruch.wheresapp

+ContacteItemAdapter(context: Context, resource: int, textViewResourceId: int, llista: ArrayList<Contacte>): void

+getView(position: int, convertView: View, parent: ViewGroup): View

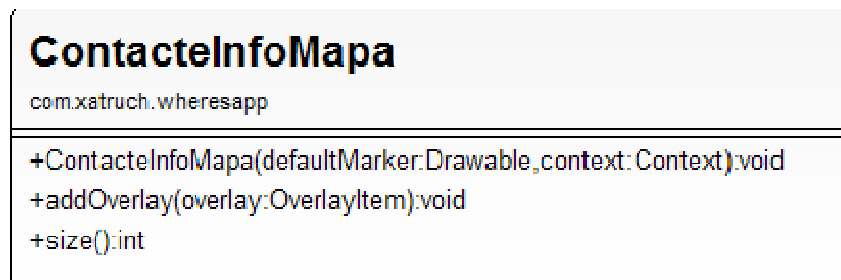
#### II-lustració 16: Classe ContacteItemAdapter

La classe ContacteItemAdapter facilita la feina de mostrar la llista de contactes. És qui, a partir d'un XML que descriu una estructura d'interfície gràfica, construeix les vistes de la llista de



contactes. S'anomena vista a cada contacte, és a dir, a cada ítem de la llista.

#### 4.1.5. Classe ContacteInfoMapa



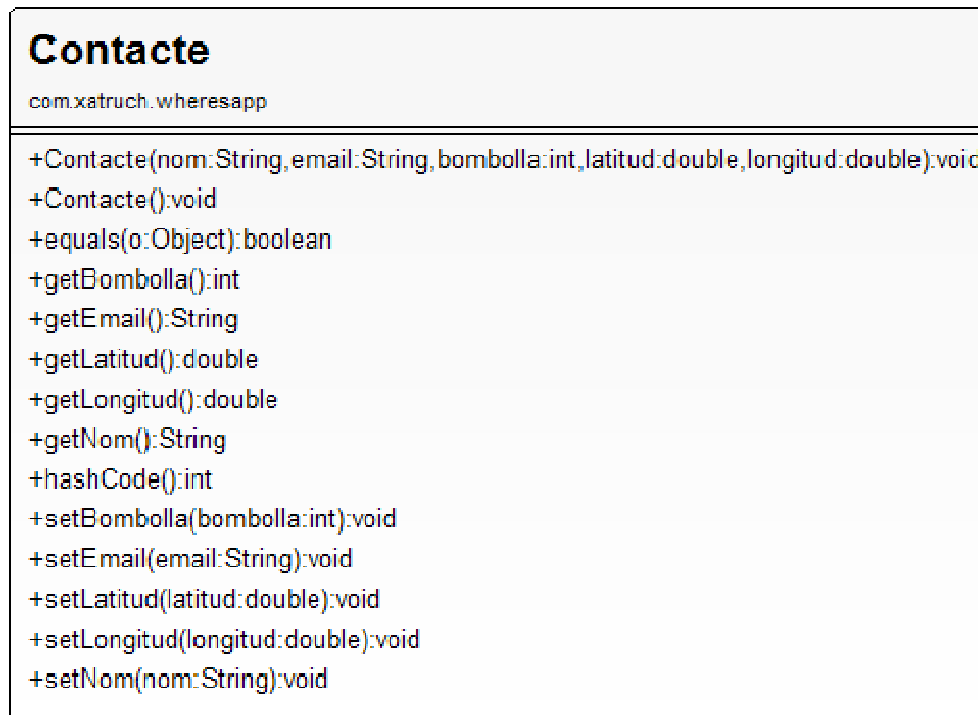
##### Il·lustració 17: Classe ContacteInfoMapa

Molt semblant a la classe anterior, però aquesta ens serveix d'ajuda per a mostrar els contactes sobre el mapa. L'API dels mapes de google permet afegir-hi a sobre, en qualsevol posició, qualsevol element gràfic.

Mitjançant aquesta classe es creen els vectors que contenen els elements a dibuixar sobre el mapa. Cada element és una imatge emmagatzemada als recursos del projecte. A cada imatge dibuixada sobre el mapa se li assigna un contacte i d'aquesta manera se'n pot veure la informació polsant sobre ella.



#### 4.1.6. Classe Contacte

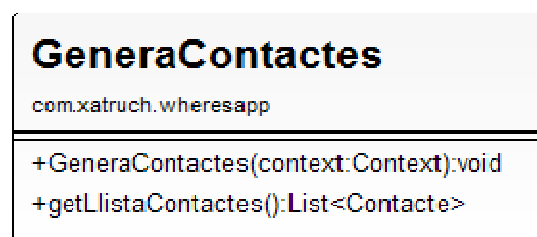


**Il·lustració 18: Classe Contacte**

Aquesta classe serveix per gestionar tota la informació relacionada amb un contacte concret.

Permet crear un contacte nou a partir de tota la informació sobre ell: nom, adreça de correu, bombolla de protecció i posició geogràfica. També és possible crear un contacte buit i anar-lo omplint amb informació segons arribin les dades que necessita.

#### 4.1.7. Classe GeneraContactes



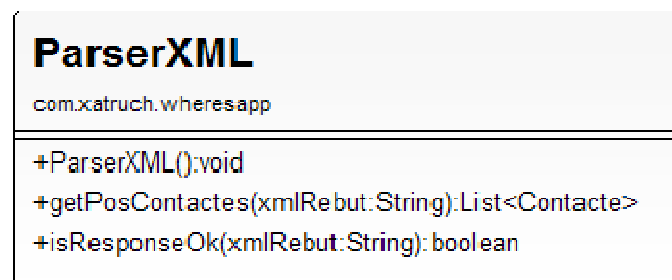
**Il·lustració 19: Classe GeneraContactes**

La classe `GeneraContactes` és l'encarregada de generar una llista amb tots els contactes que té l'usuari al dispositiu mòbil.

Utilitzant funcions específiques de l'API d'Android és possible accedir a la llista de contactes de l'usuari. Es busquen d'entre tots els contactes els que tenen adreces de correu tipus `nom@gmail` es munta una llista, extraient-ne els repetits. Els contactes es filtren per tipus `nom@gmail` per dotar al sistema de certa seguretat donat que les úniques adreces vàlides per donar-se d'alta al sistema són les vinculades al compte de google. Per tant quan un contacte d'aquests es doni d'alta com usuari ho farà sempre amb una adreça d'aquest tipus.

Els accessos a la llista de contactes del dispositiu requereixen un permís exprés dins del manifest del projecte.

#### 4.1.8. Classe `ParserXML`



**Il·lustració 20: Classe `ParserXML`**

La funció d'aquesta classe és la de transcriure l'XML rebut pel servidor i convertir-lo en informació que entén l'aplicació. El servidor ens envia un XML específic amb la llista dels nostres contactes i la seva informació actual juntament també amb la seva posició geogràfica. Aquesta classe, doncs, rep l'XML i genera una llista d'objectes `Contacte` per a que l'activitat encarregada de mostrar la llista de contactes pugui fer-ho.

A més, es comproven que les peticions llençades cap al servidor hagin acabat correctament comprovant que el servidor retorna un XML corresponent a un OK. De no ser així s'avisava a l'usuari mitjançant un missatge d'error per pantalla.



### 4.1.9. Classe ApiWEB

#### ApiWeb

com.xatruch.wheresapp

```
+ApiWeb(context: Context, handlerMissError: Handler): void  
+CridaServeiWeb(q: String): String  
+actualitzaContactes(email: String, passwd: String, llistaContactes: List<Contacte>): void  
+actualitzaCoord(email: String, passwd: String, gpsCoord: String): void  
+altaUsuari(nom: String, email: String, passwd: String, bombolla: int, bombolla_gen: int): void  
+getPosContactes(email: String, passwd: String): List<Contacte>
```

#### II-lustració 21: Classe ApiWEB

La classe ApiWeb implementa totes les funcions necessàries per accedir a l'api que defineix el servidor. Totes les peticions al servidor passen obligatòriament per aquesta classe, d'aquesta forma s'aconsegueix separar les diferents parts que componen l'aplicació. Amb aquesta abstracció, el manteniment de l'aplicació sempre serà més senzill. La classe ApiWeb treballa en combinació amb les classe GeneradorXML i ParserXML que creen i llegeixen els XML que s'envien o es reben del servidor.

Per fer les peticions cap al servidor s'utilitza el mètode POST. El mètode POST va molt bé quan el client té la necessitat d'enviar dades al servidor i a més permet incloure les pròpies dades com a part de la petició.

És d'aquesta manera com l'aplicació Android envia una petició HTTP POST al servidor incloent com a part de la petició l'XML corresponent. El servidor tractarà la petició i retornarà un altre XML com a resposta a la petició prèvia.



#### 4.1.10. Classe GeneradorXML

### GeneradorXML

com.xatruch.wheresapp

```
+GeneradorXML():void  
+actualitzaContactes(email:String,passwd:String,lListaContactes>List<Contacte>):String  
+actualitzaCoord(email:String,passwd:String,gpsCoord:String):String  
+altaUsuari(nom:String,email:String,passwd:String,bombolla:int,bombolla_gen:int):String  
+getPosContactes(email:String,passwd:String):String
```

#### Il·lustració 22: Classe GeneradorXML

Treballant juntament amb la classe anterior, aquesta és la que genera tots els XML necessaris per fer totes les peticions definides cap al servidor. A partir de funcions com ara `actualitzaContactes()` o `altaUsuari()` es genera l'XML amb totes les dades necessàries per dur a terme la petició.

## 4.2. Costat del servidor

### 4.2.1. Implementació API web

Les peticions fetes per les aplicacions client WheresApp les gestionen un conjunt d'scripts programats en PHP. Cada petició arranca un conjunt de procediments que s'encarreguen de:

- Rebre i emmagatzemar en variables temporals tota la informació continguda en la petició.
- Traduir l'XML rebut a les comandes necessàries per dur a terme la petició.
- Comprova la identitat de l'usuari, mitjançant la comparació de contrasenyes.
- Dur a terme l'operació demanada
- Retorna el resultat de l'operació a l'aplicació client.

L'aplicació client rebrà a través del mateix mecanisme pel qual ha fet la petició, la resposta a aquesta. En cas d'error pot mostrar el missatge adequat.





Els scripts PHP fan servir les classes proporcionades pel llenguatge PHP per a llegir i escriure els XML seguint la definició de les estructures definides per a aquest projecte.

#### **4.2.2. Implementació de l'accés a la base de dades**

La combinació del llenguatge PHP i del motor de base de dades MySQL és una de les millors alternatives ja que proporcionen una alta eficiència y robustesa als accessos a la base de dades.

A més, de fet, existeixen eines programades en PHP que faciliten la gestió de la base de dades i permeten administrar-la a través d'interfície gràfiques. D'altra manera s'hauria de fer des de l'entorn de comandes del servidor MySQL. En aquest projecte s'ha fer servir el PhpMyAdmin per fer les tasques de creació i administració de la base de dades.

El llenguatge PHP disposa de les funcions suficients per poder implementar totes les accions necessàries que demana el servidor de WheresApp. PHP ens permet fer d'una manera senzilla establir la connexió amb la base de dades, per exemple, amb la funció `mysql_connect()`, o bé fer consultes SQL utilitzant funció `mysql_query()`, que retorna directament un vector contenint les files que resulten d'executar la consulta SQL.

D'aquesta manera s'han implementat en PHP totes les funcions necessàries per a satisfer les peticions que les aplicacions Android fan al servidor de l'aplicació.

## 5. Possibles millores

A continuació s'enumeren un conjunt de millores de l'aplicació i del sistema en general. Algunes d'elles surten fora de l'abast d'aquest projecte com per exemple les que tenen com a objectiu únic millorar la seguretat de la comunicació entre les aplicacions clients i el servidor.

- Fer servir el número de telèfon com a identificador únic d'usuari per comptes de l'adreça de google vinculada al dispositiu.

D'aquesta manera s'amplien les possibilitats immensament i es podrien fer versions per a altres sistemes operatius mòbils sense que fes falta una adreça del servei de Google. Fer servir el número de mòbil com a identificador únic de l'usuari implica necessàriament validar aquest número a l'hora de donar-se d'alta donat que és possible que del dispositiu mòbil no es pugui llegir el número de telèfon de la targeta SIM i a més per evitar possibles altes falses. El fet de validar un número de telèfon implica una despesa econòmica ja que la manera més fàcil de fer-ho és enviant un SMS al número de telèfon que demana a l'alta i des de l'aplicació comprovar que realment s'ha rebut l'SMS correctament des del servidor de l'aplicació.

- Qualsevol canvi en els contactes de l'usuari ha de ser detectat automàticament per l'aplicació i enviar aquests canvis de nou cap al servidor. Sempre tenint present de minimitzar la transferència de dades entre l'aplicació i el servidor.
- Una altra millora força important seria la de donar més opcions a partir d'un contacte concret. Podrien sortir opcions com per exemple, trucar-li, o enviar-li un SMS, o enviar-li un correu electrònic, o compartir qualsevol arxiu multimèdia amb ell.
- La comunicació entre les aplicacions client i el servidor es realitzar mitjançant peticions sobre el protocol HTTP. D'aquesta manera, la contrasenya de l'usuari s'envia en text pla i pot ser visible per qualsevol que tingui accés a les dades que circulen per la xarxa. Així doncs és necessària una comunicació xifrada. Una manera fàcil de millorar la seguretat de les comunicacions seria fent servir una parella de claus pública i privada. Cada aplicació client tindria la seva parella de claus i igualment el servidor. Així, abans de



començar les comunicacions client-servidor només caldria intercanviar-se les claus públiques i xifrar/desxifrar cada missatge enviat/rebut per tal de validar la procedència del missatge.

Una manera encara més fàcil i segura d'aconseguir el funcionament explicat anteriorment seria l'ús del protocol HTTPS que precisament el que assegura és una comunicació xifrada entre les parts que es comuniquen. Aquesta seria l'opció escollida per la facilitat d'ús que comporta.



## 6. Conclusions

Els objectius definits a l'inici d'aquest projecte han estat assolits satisfactòriament. S'ha obtingut el producte que s'havia definit i complint els requeriments necessaris i dins de la temporització establerta durant la planificació.

Durant el desenvolupament d'aquest projecte he posat en pràctica molts dels coneixements assolits en els estudis d'Enginyeria Informàtica en els diferents àmbits, de gestió de projectes, de desenvolupament de programari, entre d'altres assignatures que m'han estat molt útils.

A banda de posar en pràctica els coneixements adquirits durant els estudis s'ha fet palès que el realment important dels estudis d'una Enginyeria és ensenyar a l'alumne a resoldre situacions imprevistes de la millor manera possible, és a dir, a resoldre amb astúcia els problemes que sorgeixen durant el desenvolupament d'un projecte. És d'aquesta manera que, per exemple, s'aconsegueix el desenvolupament satisfactori d'un projecte basat en una tecnologia força recent, i que no es veu durant els estudis, com és el sistema operatiu Android i tot el que implica en els dispositius mòbils, tauletes, i d'altres dispositius que cada dia més incorporen el sistema operatiu Android.

Tot i haver assolit els objectius del projecte final de carrera considero que hauria de seguir desenvolupant-se, aplicant les millores necessàries i corregint les fallades que apareguin. De fet, aquesta n'és la meva intenció.

## 7. Glossari

La següent taula conté els termes, abreviatures i acrònims utilitzats al llarg de la memòria.

<b>Android</b>	Android és un conjunt de programari per a telèfons mòbil que inclou un sistema operatiu, programari intermediari i aplicacions.
<b>Apache</b>	Apache HTTP Server és un servidor HTTP (de pàgines web) de codi obert multiplataforma desenvolupat per Apache Software Foundation.
<b>API</b>	Una Interfície de Programació 'Aplicacions (Application Programming Interface, API), és un conjunt de declaracions que defineix el contracte d'un component informàtic amb qui farà ús dels seus serveis.
<b>Eclipse</b>	Eclipse és un entorn integrat de desenvolupament de codi obert programada principalment en Java.
<b>Google</b>	Google Inc. és l'empresa propietària de la marca Google, el producte principal de la qual és el motor de recerca del mateix nom.
<b>GPS</b>	El Sistema de posicionament global, conegut vulgarment com a GPS (originàriament NAVSTAR Global Positioning System o NAVSTAR GPS), és un sistema de navegació per satèl·lit que permet saber amb molta precisió la pròpia situació geogràfica i l'hora de referència amb gran exactitud en gairebé qualsevol lloc de la Terra.
<b>HTC Desire</b>	El HTC desire és un smartphone desenvolupat per HTC.
<b>HTTP</b>	El protocol de transferència d'hipertext o HTTP (HyperText Transfer Protocol) estableix el protocol per a l'intercanvi de documents d'hipertext i multimèdia al web.
<b>HTTPS</b>	HTTPS (Hypertext Transfer Protocol sobre Secure Socket Layer) és la capçalera d'URI utilitzada per a indicar una connexió segura HTTP.
<b>Java</b>	El Java és un llenguatge de programació dissenyat el 1990 per James Gosling amb altres companys de Sun Microsystems a partir de C++.
<b>MySQL</b>	MySQL és un sistema de gestió de bases de dades relacional (anglès RDBMS - Relational DataBase Management System) multi-fil (multithread) i multiusuari, que usa el llenguatge SQL (Structured Query Language).
<b>Samsung</b>	El Samsung Galaxy és un smartphone desenvolupat



<b>Galaxy</b>	per Samsung.
<b>SIM</b>	La targeta SIM (acrònim de l'anglès Subscriber Identity Module) és una targeta intel·ligent desmuntable utilitzada en telefonia mòbil que emmagatzema de forma segura la identificació d'un client de telefonia mòbil.
<b>SMS</b>	El servei de missatges curts o SMS (Short Message Service) és un servei disponible en els telèfons mòbils que permet l'enviament de missatges curts (també coneguts com a missatges de text) entre telèfons mòbils.
<b>URL</b>	L'URL (de l'anglès, Uniform Resource Locator, «localitzador uniforme de recursos») és una cadena de caràcters que informa al navegador de la màquina on és el recurs a què fa referència.
<b>XML</b>	XML, de l'anglès eXtensible Markup Language («llenguatge de marques extensible»), és un metallenguatge extensible, d'etiquetes.



---

## 8. Bibliografia

- **Android developers**  
Pàgina oficial d'Android per desenvolupadors  
<http://developer.android.com/index.html>
- **Android Application Development**  
Videotutorials  
<http://thenewboston.org/watch.php?cat=6&number=154>
- **Customizing Android ListView Item Layout**  
<http://codehenge.net/blog/2011/05/customizing-android-listview-item-layout/>
- **Tratamiento de XML en Android (II): SAX Simplificado**  
<http://www.sgoliver.net/blog/?p=1580>
- **Clase SimpleXMLElement**  
Documentació oficial de PHP  
<http://www.php.net/manual/es/class.simplexmlelement.php>
- **How to execute Async task**  
Execució de tasques asíncrones  
<http://stackoverflow.com/questions/6531950/how-to-execute-async-task-repeatedly-after-fixed-time-intervals>
- **Add dynamically elements to a listView Android**  
Control de llistes d'Android  
<http://stackoverflow.com/questions/4540754/add-dynamically-elements-to-a-listview-android>
- **Interfaz de usuario en Android: Tab Layout**  
<http://www.sgoliver.net/blog/?p=2112>
- **Hello, Views**  
Estructures de pantalla d'Android  
<http://developer.android.com/resources/tutorials/views/index.html>