



**OPEN SOURCE FRAMEWORK FOR TRUCKS FLEET
CONTROL (CLIENT)**

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per Jordi Agramunt Luzón
i dirigit per Jordi Carrabina Bordoll

Bellaterra, 16 de Setembre de 2009

El sotasignat, Dr. Jordi Carrabina Bordoll

Professor de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Jordi Agramunt Luzón

I per tal que consti firma la present.

Signat:

Bellaterra, 16 de Setembre de 2009.

*A tota la gent que, tenint jo una certa edat i després
d'haver cursat una carrera de la qual no n'havia tret cap profit,
va creure en mi i em va recolzar per tirar endavant
en aquest camí que tots sabíem que seria llarg.*

AGRAÏMENTS

A l'Anna, la meva companya, pel seu suport, la seva paciència, les seves correccions ortogràfiques i gramaticals i el seu inestimable ajut en la confecció i preparació de la presentació.

Al Víctor, el projectista que elaborava la part servidor, pels seus consells, la documentació facilitada i l'enorme facilitat que ha suposat treballar coordinament amb ell.

Al Jordi Carrabina, el director del projecte, per les bones orientacions i directrius facilitades, per la bona disposició a atendre i resoldre els dubtes que han anat sorgint i per facilitar les eines necessàries on poder provar l'aplicació desenvolupada.

Índex

Capítol 1. Introducció	6
1.1. Motivació	6
1.2. Objectius	7
1.3. Organització de la memòria	7
Capítol 2. Fonaments, Entorn i Metodologia de Treball	9
2.1. Estat de la Tecnologia	9
2.1.1. Geolocalització i SIG	9
2.1.2. Terminals mòbils per a geolocalització	9
2.1.3. Android	10
2.2. Estudi de mercat	11
2.3. Entorn i Metodologia de treball	12
2.4. Planificació	15
Capítol 3. Disseny i Implementació	16
3.1. Arquitectura del sistema	16
3.2. Disseny de components	17
Capítol 4. Test i Resultats	20
4.1. Entorn de Verificació	20
4.2. Característiques del codi	20
4.3. Validació a nivell de sistema/plataforma	22
Capítol 5. Conclusions	24
5.1. Conclusions	24
5.2. Experiència personal i professional	25
5.2.1. Formació	25
5.2.2. Experiència professional	26
5.3. Evolució futura	26
6. Referències i Bibliografia	28
ANNEXOS	30
ANNEX I. ARBRE DE L' APLICACIÓ	31
ANNEX II. BONES PRÀCTIQUES	32
ANNEX III. PANTALLES DE L' APLICACIÓ	34
ANNEX IV. DDMS	38

Capítol 1. Introducció

1.1. Motivació

Hi ha diferents al·licients que han estat claus a l'hora de decidir la realització d'aquest projecte.

El primer d'ells ha estat l'oportunitat de poder realitzar una aplicació que funcionés sobre un dispositiu mòbil. L'aparició recent d'entorns de desenvolupament oberts per a mòbils, amb llibreries i eines de desenvolupament ben documentades, motiven alhora que faciliten la realització d'aquest tipus de projectes. A més, en el cas d' Android que és la plataforma sobre la qual s'ha desenvolupat l'aplicació, al ser una plataforma de programació de codi obert¹, basada en Java i fàcilment extensible ha propiciat que s'estigui utilitzant en altres dispositius a part de telèfons mòbils. Alguns dels dispositius on ja s'està usant aquest sistema operatiu i/o on es preveu que s'utilitzarà en breu són:

- Discs durs portàtils multimèdia.
- Marcs de fotos digitals.
- Televisors.
- Aparells de veu sobre IP.
- Mini ordinadors portàtils.

Un altre aspecte que s'ha tingut en compte a l'hora de triar aquest projecte ha estat el poder experimentar amb un sistema d'informació geogràfica (GIS en endavant). L'increment d'ús d'aquests sistemes juntament amb la seva cada vegada més àmplia aplicació en diversos àmbits com poden ser investigacions científiques, sociologia, màrqueting, geografia, logística... fan que sigui molt atractiu investigar-hi i fer-ne ús.

Un altre motivació per a la realització del projecte és el fet d'estar inclòs en un projecte de més abast com és el control d'una flota de camions a nivell global. El PFC que ens ocupa

n'implementa la part client i es complementa amb un altre PFC que en desenvolupa la part servidor. Les dades dels diferents camions recollides per la part client són enviades al servidor a través d'un servei web per tal de tenir informació a temps real de la flota i poder prendre les decisions adients per tal d'optimitzar-ne el rendiment. La coordinació entre els dos projectes era un altre dels reptes ja que la part client l'ha estat desenvolupant un projectista des d'Alemanya.

1.2. Objectius

L'objectiu principal és implementar un sistema de localització geogràfica per a dispositius mòbils i enviar les dades de posicionament a un servei web de l'aplicació servidor.

Els requisits d'aquesta aplicació són:

- Enviar dades de posicionament a un servei web.
- Estar desenvolupada amb tecnologia de codi obert.
- Mostrar una interfície d'usuari integrada amb Google Maps on es vegi la posició actual del dispositiu mòbil.
- Interactuar amb el mapa per poder buscar llocs i definir rutes.
- Implementar un sistema de recuperació després de pèrdues de cobertura tant GPS com 3G o GPRS .

1.3. Organització de la memòria

La memòria està organitzada en quatre grans blocs.

En un primer bloc s'explica l'abast del projecte i el motius que n'han motivat el seu desenvolupament.

En el segon bloc, es posa èmfasi en la vessant més teòrica del projecte incloent els diferents aspectes que representen l'entorn del projecte. Aquests aspectes van des de les tecnologies que s'aborden, a les aplicacions que actualment les usen. També s'hi inclou un estudi sobre com estan representades actualment i quines possibilitats i innovacions pot aportar el nostre projecte.

El tercer bloc afronta la part més pràctica, on s'explica l'entorn i la manera de treballar, la planificació i, finalment, s'aborden les qüestions més tècniques del projecte. En aquest últim punt hi ha la descripció de la tecnologia utilitzada, les funcionalitats de l'aplicació i el resultat de la seva implantació i de les proves realitzades en un dispositiu real.

Finalment, s'exposen les conclusions, es resumeixen l'experiència i formació del projectista i es proposen possibles evolucions del projecte desenvolupat. En aquest últim punt s'hi posa especial atenció ja que hi ha la intenció d'estendre'l i també pot ser aprofitat per futurs projectistes.

Capítol 2. Fonaments, Entorn i Metodologia de Treball

2.1. Estat de la Tecnologia

2.1.1. Geolocalització i SIG

Els Sistemes d'Informació Geogràfica donen les eines per poder oferir serveis d'informació d'ubicació geogràfica o geolocalització. Els diferents software GIS conjuntament amb la integració de GPS en els dispositiu mòbils fan que cada vegada hi hagi més aplicacions tant en l'àmbit particular com professional orientades a explotar les possibilitats que aquests ofereixen.

Aquests serveis, que com hem comentat anteriorment estan cada vegada més estesos, van des d'usos socials com poden ser informacions d'ubicació de posició en cotxes particulars, bicicletes, excursionisme, etc. a usos comercials com control de posicionament dels vehicles d'una empresa que és el cas pràctic sobre el que versa aquest projecte.

Tot i això, cal aclarir que el sistema de posicionament GPS en petits dispositius mòbils com telèfons encara no és tot lo acurat que es desitjaria. En molts casos hi ha errors significatius en el posicionament que, tot i que no és cas d'aquest projecte, poden ser un impediment a l'hora d'usar-los en depèn de quines aplicacions.

2.1.2. Terminals mòbils per a geolocalització

Cada vegada més, els terminals mòbils incorporen noves funcionalitats hardware i software. Una de les més recents i que s'està estenent més àmpliament és la de dotar els dispositius de receptors GPS per a la geolocalització.

La majoria de fabricants ja compten entre els seus productes de gamma mitja o alta amb telèfons dotats d'aquesta tecnologia. Com a mostra, podem constatar que diversos productes de Nokia, els Iphone, Blackberry, Sony-Ericsson, Motorola, LG, HTC... tenen al mercat models amb aquesta funcionalitat.

2.1.3. Android

Android és una plataforma de software de codi lliure per a telèfons i dispositius mòbils. Aquesta plataforma engloba un sistema operatiu, una màquina virtual java anomenada *Dalvik*, un conjunt de llibreries de baix nivell i conjunt inicial d'aplicacions destinades a l'usuari final.

Al voltant d'aquesta plataforma, es va crear la *Open Handset Alliance*, un grup de 47 companyies de tecnologia i telefonia mòbil unides per accelerar la innovació en la telefonia mòbil recolzant-se en la primera plataforma complerta, oberta i gratuïta.

Android va ser creada de zero i desenvolupada per ser vertaderament oberta de manera que permetés als desenvolupador fer aplicacions que aprofitin tot el potencial que pot oferir avui dia un mòbil.

Aquest sistema va ser presentat el novembre de 2007 i des de llavors ja ha ofert als desenvolupadors 5 revisions del seu paquet de desenvolupament de software. Actualment, hi ha disponible la versió 1.5 revisió 3 d'aquest paquet. També, i a l'entorn d'aquesta plataforma, aquest any s'ha organitzat la segona edició del *Android Developer Challenge*, un concurs creat per Google i on s'ofereixen importants premis econòmics als millors projectes. Aquest concurs està obert a tothom que vulgui desenvolupar una aplicació que funcioni sobre el sistema operatiu Android.

Els primers mòbils amb aquest sistema operatiu van sortir al mercat el 2008 de la mà de Motorola i, actualment, HTC també ha tret al mercat diferents models amb aquest sistema operatiu.

2.2. Estudi de mercat

Hi ha una gran varietat de fabricants de mòbils i, a més, cada fabricant té multitud de models funcionant amb diferents sistemes operatius. Entre els sistemes operatius recents més populars pensats pel tipus de mòbil (*smartphone*) que es necessita per poder desenvolupar la nostra aplicació, podem trobar:

- Windows Mobile és el sistema operatiu per a dispositius mòbils desenvolupat per Microsoft. En un principi pensat per aparells de tipus *Pocket PC*, actualment i gràcies a l'increment de funcionalitats dels telèfons mòbils denominats *smartphones* també s'està usant en aquests dispositius. No es va contemplar desenvolupar la nostra aplicació sobre aquest sistema ja que Microsoft mai s'ha caracteritzat per recolzar desenvolupadors independents ni per obrir el codi font dels seus sistemes.
- Symbian OS, actualment és el sistema operatiu per *smartphones* líder al món. Aquest sistema operatiu va ser pensat des d'un principi per ser destinat a dispositius mòbils i com a competència de Windows Mobile. Va néixer com a producte de l'aliança entre diferents companyies privades i, des del 2008 en què Nokia en va comprar la majoria de les accions, s'ha creat una fundació² l'objectiu de la qual és crear una plataforma oberta de software per telèfons mòbils. El principal problema que vam trobar per desenvolupar l'aplicació per aquesta plataforma és que l'API de desenvolupament més usada i amb més documentació i eines per desenvolupar-hi està feta en C++. També hi ha una API Java però per problemes de llicències encara no és la més estesa tot i que està en camí.

- Iphone OS, usat en el mòbils d'Apple, té una sèrie de característiques que el fan molt atractiu a l'usuari. Facilitat d'ús, interfícies molt elegants i elaborades, pantalla tàctil multipunt, gran qualitat de so i moltes aplicacions de tot tipus disponibles. Es va pensar en desenvolupar la nostra aplicació per aquest sistema però es va descartar principalment perquè la filosofia de la plataforma no està basada en codi obert i les eines de desenvolupament que es proporcionen només funcionen sobre sistemes operatius MAC.
- Android és el sistema operatiu proposat per Google i explicat en l'apartat anterior. Vam utilitzar aquesta plataforma com a destí per l'aplicació desenvolupada perquè complia tots els requisits necessaris. Plataforma innovadora, de codi obert, amb molt potencial de futur, amb moltes eines i documentació per desenvolupar, amb una API basada en Java, amb eines de desenvolupament disponibles per als sistemes operatius més populars, amb mòbils en el mercat que utilitzen aquest sistema operatiu, etc. A més, i gràcies a la aliança muntada al voltant d'aquesta plataforma (*Open Handset Alliance*) tot fa preveure que en un futur proper estarà molt estès.

2.3. Entorn i Metodologia de treball

Per realitzar el projecte s'han utilitzat diferents eines i entorns de desenvolupament que ara procedirem a enumerar. S'ha usat l'entorn de desenvolupament eclipse 3.4 (*Ganymede*). Per poder compilar i desplegar l'aplicació en l'emulador s'ha hagut d'instal·lar el *plugin* d'Android per Eclipse. A més del *plugin*, també ha calgut descarregar-se el paquet d'eines de desenvolupament que proporciona Google per a la plataforma Android. Entre aquestes eines hi ha les llibreries Java, una eina que es connecta a l'emulador i permet simular diferents comportaments del mòbil (recepció de SMS, recepció de coordenades GPS, etc.), eines de línia de comandes per connectar-nos a l'emulador i navegar pel sistema de fitxers o fer connexions a la base de dades interna.

A part de les eines pròpies per desenvolupar el projecte, per poder fer les proves d'enviament de dades al servei web, s'ha hagut de muntar un servidor web (*apache-tomcat-5.5.27*) amb el servei desplegat i una base de dades (*Mysql Server 5.0*) per emmagatzemar les dades enviades pel client. Tant l'aplicació servidora com els scripts de creació de base de dades han estat facilitats pel

projectista que desenvolupa la part servidor. Només s'han hagut d'adaptar algunes rutes de fitxers de configuració i tornar a crear el paquet, un arxiu *.war* en aquest cas.

A tall de recopilació, detallem les diferents eines usades i les seves principals funcions:

- Eclipse 3.4 per codificar i compilar el projecte.
- Plugin ADT (*Android Development Tools*) per eclipse versió 1.1 per poder desplegar l'aplicació en l'emulador i depurar-la.
- Plugin *Subversive* per eclipse. S'ha usat un repositori *Subversion* per a control de versions i per aquesta raó hem utilitzat aquest *plugin*.
- Llibreries Java d'Android versió 1.1 per poder usar totes les classes ja desenvolupades per google.
- DDMS (Dalvik Debug Monitor Service), eina amb múltiples usos proporcionada per google. Dóna informació de memòria, informació dels processos, permet simular recepció de trucades, sms, de dades de posicionament, etc. En el nostre cas ha estat molt útil per la simulació de l'enviament de dades de posicionament. Per més informació sobre aquesta eina, veure <http://developer.android.com/guide/developing/tools/ddms.html>
- ADB (Android Debug Bridge), eina de línia de comandes que permet administrar l'estat d'una instància de l'emulador. S'ha usat per comprovar dades del sistema de fitxer i per connectar-nos a la base de dades interna d'Android i comprovar que les dades s'estaven emmagatzemant correctament.
- TCPmon per veure l'intercanvi de dades entre el servei web i l'aplicació client.

Cal remarcar també, que per poder treballar amb les llibreries de Google Maps i utilitzar els seus mapes, s'ha d'obtenir una clau i registrar-la en l'aplicació. Si es treballa en mode Depuració es pot utilitzar un certificat que ja s'instal·la amb el paquet Android de Google. Quan es genera l'aplicació definitiva per distribuir entre els usuaris cal generar un certificat. Els passos a l'hora de fer-ho són:

- Agafar l'empremta digital del certificat tant si és el proporcionat per Google com si és generat. Ex: `keytool -list -alias uab -keystore projecte.keystore`
- Genera la clau a partir de l'empremta digital. Per fer-ho cal anar a la següent URL i introduir l'empremta: <http://code.google.com/intl/ca-ES/android/maps-api-signup.html>
- En tots els *layouts* on s'utilitzen mapes de Google afegir-hi la clau generada en el lloc corresponent.

```
<com.google.android.maps.MapView
    android:id="@+id/mapview1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true"
    android:apiKey="0xA581MArObW0crSv1Ft459c64frLVZwDa-ZmiA"
/>
```

1. Part del fitxer de layout on s'inserta la clau

La metodologia que s'ha seguit a l'hora de realitzar el projecte ha estat:

- Buscar possibles plataformes de sistemes operatius per mòbils on es pogués usar l'aplicació que es volia desenvolupar. Un cop trobades, se'n van anar descartant per diferents motius com manca de documentació o poc actualitzada, entorns de desenvolupament que només funcionen per sistemes operatius en concret (l'entorn de desenvolupament d'Iphone només funciona sobre Mac), no contemplar codi obert, etc. Al final es va optar per Android per diferents motius com ser innovadora, apostar clarament pel codi obert, tenir eines de desenvolupament potents i documentació completa i actualitzada, tenir el recolzament de una gran quantitat d'empreses importants en el sector.
- Identificar les parts que composarien l'aplicació i buscar informació sobre si era possible implementar-les en la plataforma escollida.
- Implementar els diferents mòduls que composaven l'aplicació i anar-los validant a mesura que s'anaven desenvolupant.
- Fer proves unitàries sobre l'emulador i també sobre un terminal real.

2.4. Planificació

S'ha intentat respectar la planificació inicial presentada en l'informe previ.

- Desembre 2008: Plataforma de desenvolupament decidida i operativa (Eclipse 3.4 + Android).
- Gener 2009: Primer prototip d'aplicació amb crida a servei web per plataforma Android.
- Febrer 2009: Afegir la recollecció de dades reals de localització interactuant amb el hardware GPS.
- Març 2009: Integració amb googlemaps per mostrar la localització.
- Abril 2009: Interacció amb googlemaps per definir rutes i buscar localitzacions.
- Maig 2009: Gestió d'errors de cobertura.
- Juny 2009: Portabilitat a altres sistemes operatius mòbils respectant el precepte d'usar codi obert.
- Juliol 2009: Portabilitat a altres sistemes operatius mòbils respectant el precepte d'usar codi obert.
- Agost 2009: Proves, millores i/o optimitzacions, correcció d'errors i memòria.
- Setembre 2009: Preparació de la presentació del projecte i presentació del mateix.

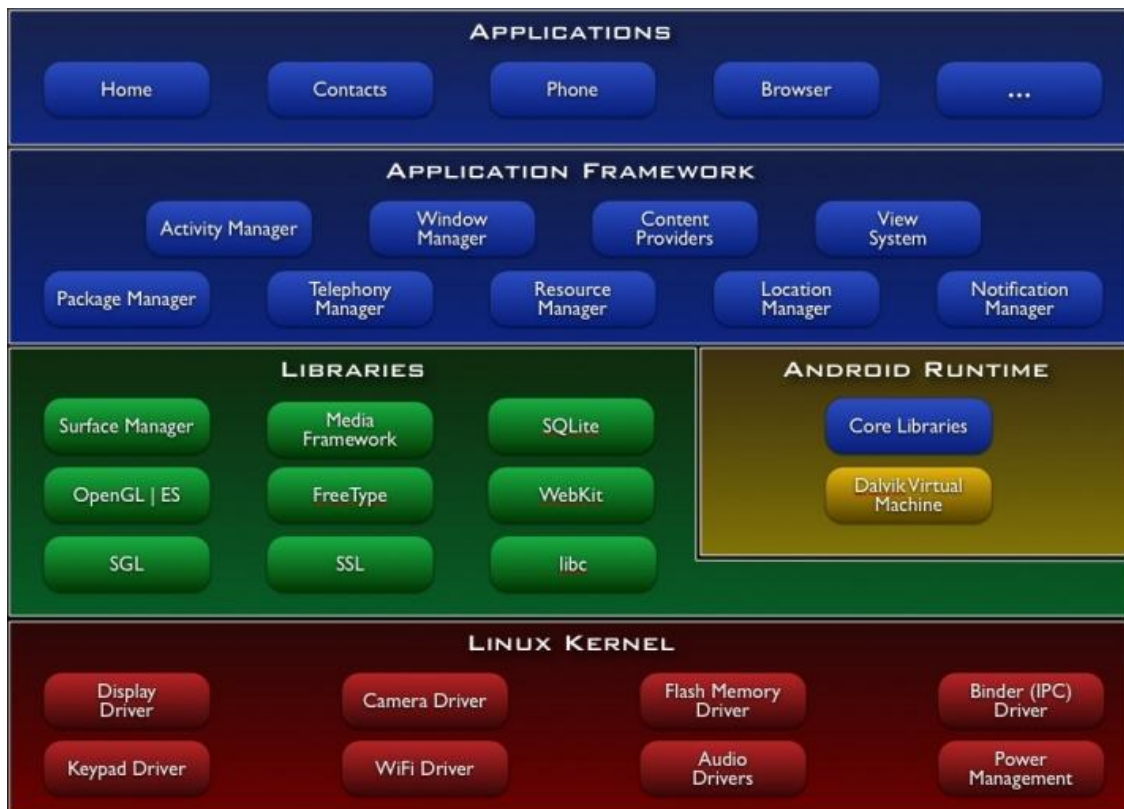
Capítol 3.

Disseny i Implementació

3.1. Arquitectura del sistema

L'aplicació està feta per tal que s'executi sobre un dispositiu amb sistema operatiu Android. Això fa que l'arquitectura del sistema estigui condicionada a l'arquitectura d'aquesta plataforma. Les parts que la componen són:

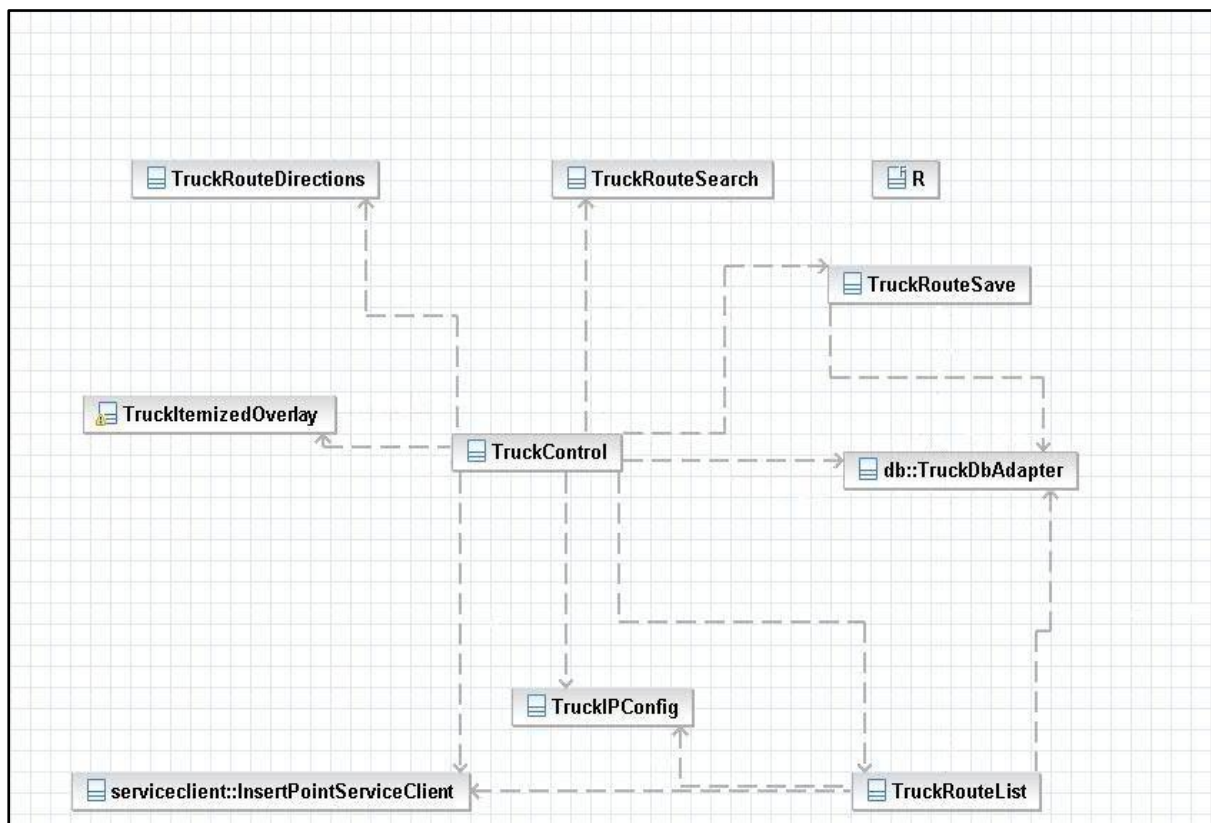
- Nucli linux.
- Una capa de llibreries que ofereixen diferents funcionalitats.
- Una màquina virtual java (Dalvik).
- Una carcassa d'aplicacions de sistema.
- Les aplicacions d'usuari entre les quals s'hi troba la desenvolupada en aquest projecte.



3.2. Disseny de components

Els components que formen l'aplicació són classes Java, fitxers de configuració i fitxers de recursos per a les imatges i per a les interfícies de les activitats que componen l'aplicació.

De les classes que componen l'aplicació, la majoria estenen de la classe *android.app.Activity* que és la classe base de totes les classes que en una aplicació tindran interacció amb l'usuari. A continuació, mostrem el diagrama de classes i explicarem les funcions principals de cadascuna d'elles:



2. Diagrama de classes

- **TruckControl:** classe principal i punt d'entrada de l'aplicació. En ella es dibuixa el mapa i conté la classe que escolta els esdeveniments del GPS per poder mostrar la localització del camió en el mapa.

- `TruckItemizedOverlay`: ens permet dibuixar objectes sobre el mapa tals com el camió o la línia que mostra la ruta per la qual va passant el camió.
- `InsertPointServiceClient`: construeix l'objecte SOAP i crida al servei web enviant-li l'identificador del camió i la localització del mateix.
- `TruckRouteDirections`: s'utilitza per buscar una ruta entre dues localitzacions. Per fer-ho, crida a una URL de google passant-li un paràmetre que indica que volem que ens retorni un fitxer KML amb l'itinerari. Aquest fitxer es parseja i es mostra la ruta sobre el mapa.
- `TruckRouteSearch`: busca una localització. Si només troba una coincidència la mostra sobre el mapa, si en troba més d'una mostra la llista per tal que es seleccioni quina d'elles es vol veure sobre el mapa.
- `TruckRouteList`: mostra la llista de rutes guardades per tal de poder seleccionar la que ens interessi i mostrar-la al mapa o enviar-la al servei web o esborrar-la.
- `TruckRouteSave`: guarda la ruta actual a la base de dades per tal de poder usar-la posteriorment.
- `TruckDbAdapter`: adaptador que ens permet accedir i treballar amb la base de dades interna que porta Android.
- `TruckIPConfig`: ens possibilita configurar la IP del servei web. Desa la configuració escollida en un fitxer.

Totes les classes de l'aplicació que codifiquen una interacció amb l'usuari (estenen de la classe *android.app.Activity* o derivades d'aquesta) tenen un fitxer XML associat que modela la presentació per pantalla. Explicarem breument l'objectiu i composició de cadascun d'aquests fitxers (Annex III, captures de pantalla):

- *main.xml*: s'utilitza en la classe `TruckControl` per mostrar la vista del mapa.
- *route_directions.xml*: utilitzat en la classe `TruckRouteDirections`. Conté dues caixes de text per poder indicar el lloc origen i el destí de l'itinerari a buscar i un botó per iniciar la cerca.
- *route_ip_config.xml*: usat en la classe `TruckIPConfig`, conté una caixa de text per indicar la IP que volem configurar i un botó de confirmació.

- *route_list.xml*: es fa servir en la classe *TruckRouteList* i conté una llista que estarà composta per les rutes que tinguem guardades en base de dades.
- *route_save.xml*: s'usa en la classe *TruckRouteSave*. Conté dues caixes de text, una per introduir l'identificador de la ruta a desar i una altra per introduir la descripció de la mateixa. També conté un botó de confirmació.
- *route_search.xml*: utilitzat en la classe *TruckRouteSearch*, està compost per una caixa de text on introduïrem el lloc a buscar, un botó per inicialitzar la cerca i una llista seleccionable on es mostraran els resultats en cas que la cerca retorni més d'un resultat.

Com a fitxers de recursos tenim dos fitxers de imatge que s'utilitzen en diferents parts de l'aplicació -*camioverd2.png* i *fi2.png*- per mostrar la posició del camió en el mapa i com a senyal d'inici i fi de ruta en el mapa respectivament. També tenim un fitxer anomenat *strings.xml* que conté literals usats en diferents punts del programa.

Finalment, tenim el fitxer de configuració principal, s'anomena *AndroidManifest.xml* i conté diferents indicacions: activitats que formen l'aplicació, permisos que necessita per tal que es pugui executar correctament, llibreries utilitzades, paquet i versió (Annex V, descripció fitxer).

Capítol 4.

Test i Resultats

4.1. Entorn de Verificació

Per verificar el bon funcionament de l'aplicació ha calgut verificar que s'executa correctament i també que les dades que s'envien al servei web són rebudes i tractades correctament.

Per verificar que s'executa correctament s'ha utilitzat l'emulador que proporciona google en instal·lar les eines de desenvolupament de la plataforma Android. Des de l'Eclipse, s'ha instal·lat l'aplicació en l'emulador, s'han comprovat les diferents pantalles de l'aplicació i que el comportament ha estat coherent en tots els casos i que no hi ha hagut traces d'error generades per la nostra aplicació.

Per comprovar que les dades enviades al servei web eren rebudes i tractades correctament, ha calgut instal·lar l'entorn servidor i, amb l'ajut del programa TCPmon s'ha comprovat que l'intercanvi de missatges era correcte. Per acabar-ho de verificar també s'han comprovat la coherència de les dades introduïdes en la base de dades per l'aplicació servidora.

4.2. Característiques del codi

La memòria utilitzada pel programa en temps d'execució s'acostuma a moure al voltant dels 2MB (Annex IV, DDMS).

La grandària del paquet instal·lat en l'emulador és de 508KB (284KB aplicació i 224KB dades).

Pel què fa al temps d'execució, hi ha diversos aspectes de l'aplicació a tenir en compte i valorar en aquest apartat.

La pantalla inicial de l'aplicació mostra un mapa de Google Maps que es descarrega d'Internet. La càrrega de l'aplicació en si és gairebé instantània però hi pot haver retards en funció de la connexió a Internet utilitzada. Si l'estem executant en un emulador i la connexió a Internet és bona no hi ha retard, si ho fem des del terminal en ocasions hi pot haver un petit retard en carregar la pàgina.

A part d'això, totes les funcionalitats són molt ràpides a excepció de dues d'elles: enviament de dades al servei web i recuperació d'itineraris.

L'enviament de dades al servei web depèn de la connexió a Internet i del què tarda l'aplicació servidor a respondre. Per evitar retards o parades en l'aplicació quan s'està en el mode de navegació, s'ha optat per crear un *thread* paral·lel en la classe que envia dades al servei web per tal de no esperar la resposta del mateix i continuar amb l'execució de l'aplicació principal. En canvi, quan s'està enviant una ruta prèviament seleccionada si que s'espera al retorn per saber si s'ha pogut enviar correctament.

En la recuperació d'itineraris entre dos llocs, pot haver-hi retard si la ruta demanada està localitzada entre dos punts molt allunyats. Com que el mètode per recuperar una ruta és parsejar un fitxer *kml* recuperat d'una crida a una adreça web i mostrar-la en el mapa, com més gran és el fitxer retornat més lenta és la visualització en la pantalla del terminal.

Tot i aquestes consideracions, cal remarcar que s'han intentat seguir, en la mesura del possible, les bones practiques recomanades per Google a l'hora de programar per a la plataforma Android i, per extensió, per a sistemes encastats (veure Annex II, Bones Pràctiques) .

4.3. Validació a nivell de sistema/plataforma

Per fer la validació general, s'ha instal·lat l'aplicació en un terminal real, un mòbil HTC Dream, i s'han provat les diferents funcionalitats de la mateixa. Una vegada configurat el terminal per tenir accés a la xarxa on hi havia instal·lada l'aplicació servidor i des d'on es connectava a Internet, els resultats han estat:

- La recepció de dades GPS del terminal no és del tot acurada. La codificació estava implementada de forma que només mostrava canvis de posició en el mapa quan hi havia actualització del senyal GPS per canvi de lloc. Es va veure però que, tot i no moure'ns de lloc, el GPS enviava actualitzacions de posició a l'aplicació de diversos metres de variació. Per evitar aquest comportament, es va canviar la codificació de forma que només s'agafin actualitzacions quan la posició varia més de 10 metres. Amb aquest canvi, es continuen rebent algunes actualitzacions de posició encara que no ens haguem mogut però ja va bé per provar diferents aspectes de l'aplicació.
- La configuració de la IP del servidor a la que es connecta el terminal per enviar les dades de posicionament estava mal aconseguida. Cada vegada que el terminal i el portàtil que fa de servidor es connectaven a una xarxa diferent (a casa o a la universitat), s'havia de canviar el codi font, compilar i baixar el codi al terminal. Tot i que en un entorn de funcionament real aquest problema segurament no hi seria ja que el client sempre es connectaria contra la mateixa IP, per millorar el sistema, s'ha afegit una nova funcionalitat a l'aplicació per poder configurar la IP a la que es connecta. Aquesta configuració es guarda en un fitxer del propi terminal mòbil.
- Després de comprovar la coherència i el correcte funcionament dels diferents menús i les finestres emergents, se n'ha intentat unificar el patró de comportament per tal que l'ús de la mateixa fos més fàcil i intuïtiu.
- L'enviament de dades de posicionament al servidor durant la navegació és correcte.

- La cerca de llocs i itineraris també es realitza correctament tot i que en el cas de la cerca d'itineraris pot ser lenta en funció de la ruta retornada.
- Les diferents opcions disponibles amb les rutes desades, mostrar-les en el mapa, enviar-les al servei web i esborrar-les també funcionen correctament.

Capítol 5. Conclusions

5.1. Conclusions

Dels objectius plantejats a l'inici del projecte es pot dir que s'han aconseguit els més importants.

L'objectiu principal, que era implementar un sistema de localització geogràfica per a dispositius mòbils i enviar les dades de posicionament a un servei web, s'ha realitzat amb èxit. De la resta de requeriments que s'havien plantejat que complís l'aplicació també se n'han aconseguit la majoria:

- Enviar dades de posicionament a un servei web.
- Estar desenvolupada amb tecnologia de codi obert.
- Mostrar una interfície d'usuari integrada amb Google Maps on es vegi la posició actual del dispositiu mòbil.
- Interactuar amb el mapa per poder buscar llocs i definir rutes.
- Implementar un sistema de recuperació després de pèrdues de cobertura tant GPS com 3G o GPRS. En aquest cas, s'ha implementat un sistema per desar les rutes automàticament i poder-les enviar en qualsevol moment. La part de tenir un sistema de recuperar pèrdues de cobertura GPS no que està implementada

També s'han implementat algunes funcionalitats addicionals que no estaven previstes en un primer moment com:

- Poder desar les rutes de forma personalitzada.
- Enviar rutes prèviament desades al servei web.
- Obrir rutes prèviament desades per visualitzar-les en el mapa.

- Permetre configurar la IP del servidor web al que volem enviar les dades.

Un dels objectius que només s'ha aconseguit parcialment era intentar que l'aplicació fos multiplataforma. L'estar subjecte a les llibreries i eines de desenvolupament de les diferents plataformes fa que sigui molt complicat fer un producte que funcioni correctament en més d'una d'elles. En aquest cas però, com que Android forma part de tota una aliança d'empreses importants moltes d'elles fabricants de telèfons mòbils i tot sembla indicar que aquest sistema operatiu s'anirà estenent cada vegada més entre els diversos fabricants, es pot dir que l'aplicació acabarà sent multiplataforma ja que no depèn del mòbil en el qual s'instal·li sinó que només cal que utilitzi el sistema operatiu d' Android.

Finalment dir que també s'han aconseguit objectius a nivell personal com la introducció en la programació per sistemes encastats, el coneixement de molts aspectes de la plataforma Android, l'aprenentatge de diferents recursos que ofereix com, per exemple, la possibilitat d'emmagatzemar dades en la base de dades interna o en fitxers, la utilització de diferents *layouts* de presentació, el traspàs de dades entre diferents activitats, etc.

5.2. Experiència personal i professional

5.2.1. Formació

- Cursant el projecte d'Enginyeria Informàtica Superior a l'Escola Tècnica Superior d'Enginyeria de Bellaterra, UAB (Universitat Autònoma de Barcelona), 2000-09.
- Certificat per Sun Microsystems com a programador en la plataforma "Java 2 Standart Edition 5.0", Desembre del 2007.
- Certificat per Sun Microsystems com a desenvolupador de componentsweb per la plataforma "Java 2 Enterprise Edition 1.4", febrer del 2008.
- Certificat per Sun Microsystems com a desenvolupador de components de negoci per la plataforma "Java 2 Enterprise Edition 1.3", febrer del 2008.
- Diplomant en Magisteri d'Educació Física per la Universitat de Barcelona (UB), 1995-98.

5.2.2. Experiència professional

Treballant en el sector de la informàtica des del febrer del 2001.

Des del 2001 fins l'any passat treballant en l'empresa francesa de serveis informàtics Steria i ocupant diferents càrrecs en la mateixa. Programador junior als inicis, analista programador més tard i analista Java en l'última etapa en aquesta empresa.

Des del setembre de 2008 treballant en l'empresa BSIS, empresa fundada recentment per fer-se càrrec de la part d'informàtica del Grup Banc Sabadell. Actualment, ocupant el càrrec de responsable de l'equip de manteniment incidental i petit evolutiu de les aplicacions Java del Banc Sabadell, entre ells tots els portals de les diferents empreses de Grup.

5.3. Evolució futura

L'aplicació està penjada al "Sourceforge" i és pública a tothom que vulgui contribuir-hi. L'adreça de connexió és: <http://sourceforge.net/projects/truckfleetctr/>. En aquesta adreça s'hi pot trobar el codi font, la documentació del codi (javadoc) i hi ha configurat el software de control de versions "Subversion". En breu també s'hi afegiran documents de com funciona l'aplicació i de les solucions tècniques usades per la mateixa.

En el desenvolupament de l'aplicació s'han detectat possibles millores o evolucions que podrien ser objecte d'estudi per tal d'elaborar un altre projecte en un futur o per oferir funcionalitats més enfocades al sector professional per empreses que hi poguessin estar interessades.

La primera d'aquestes funcionalitats seria la possibilitat de rebre rutes enviades des del servidor al terminal. Amb aquesta funcionalitat, l'empresa, des del seu centre de control, podria enviar rutes al transportistes en funció de les necessitats de cada moment i d'aquesta manera optimitzar el

procés de negoci. Aquesta possibilitat, s'ha estudiat i s'ha vist que una implementació factible i relativament fàcil seria l'enviament de fitxers de tipus kml adjuntats en un missatge MMS. Un cop rebuts aquests fitxers, es desarien en l'estructura de fitxers del mòbil i es podrien obrir les rutes des de l'aplicació (recordem que l'aplicació ja implementa la funcionalitat de parsejar fitxers kml i mostrar les rutes en el mapa).

Una funció que es podia haver implementat era la d'oferir la navegació directament sobre un ruta que s'hagués cercat prèviament amb el cercador de rutes o que s'hagués obert d'entre les rutes desades. Al final, per tal de millorar la usabilitat, es va optar per oferir pantalles diferenciades ja que mentre s'està navegant ja s'està marcant la ruta per la que es va passant. Tot i això, potser seria un canvi a tenir en compte i, marcant les rutes prèviament cercades amb un color diferent de les ruta que s'està seguint en podria ser una bona solució.

Un altre funcionalitat desitjable, seria la d'oferir la cerca de rutes i emplaçaments a partir de la veu. Actualment, en la última versió del sistema operatiu Android, Google Maps ja ofereix aquesta funcionalitat, per tant, s'hauria de mirar com afegir-la a aquesta aplicació.

L'encriptació de les dades que s'envien al servei web també seria una funcionalitat desitjable d'implementar.

Per últim, un altre millora però que aniria condicionada a que també és canviés l'aplicació servidora, seria l'ús del protocol "Rest Services" en comptes de "Web Services" a l'hora d'enviar les dades del client cap al servidor. Els "Rest Services", al ser més lleugers optimitzarien el temps d'enviament i el de l'aplicació en general ja que es fa un ús gairebé constant de l'enviament de dades cap al servidor per tal de notificar la posició del camió.

6. Referències i Bibliografia

La totalitat de referències i bibliografia fan referència a adreces web. Per buscar informació a l'hora de desenvolupar el projecte i de redactar la memòria no s'han fet servir mitjans de cerca d'informació tradicionals com llibres, revistes o d'altres publicacions. També cal dir que per poder solucionar dubtes tècnics, s'ha fet la subscripció a diversos fòrums i grups de discussió relacionats amb Android. Dividirem temàticament les diferents referències usades:

Android:

- <http://developer.android.com/index.html>
- <http://groups.google.com/group/android-developers/>
- <http://www.android-spa.com/viewforum.php?f=5>
- <http://www.anddev.org/>
- <http://www.android.com/>

Altres sistemes operatius per telèfons mòbils:

- <http://developer.apple.com/iphone/>
- <http://www.symbian.org/>
- http://en.wikipedia.org/wiki/Symbian_OS
- <http://www.microsoft.com/windowsmobile/es-es/default.mspx>
- <http://www.moblin.org/>
- <https://developer.sonyericsson.com/site/global>
- <http://na.blackberry.com/eng/developers/>

Aparició d'entorns oberts per a mòbils i sistemes encastats que usen Android:

- <http://www.hdmp4.com/>
- <http://www.gadgetvenue.com/android-embedded-digital-photo-frame-05144124/>
- http://www.eetasia.com/ART_8800569927_499495_NT_b5ffd5e1.HTM
- <http://www.fayerwayer.com/2009/05/futurologia-netbooks-dell-corriendo-android/>

Sistemes d'informació geogràfica:

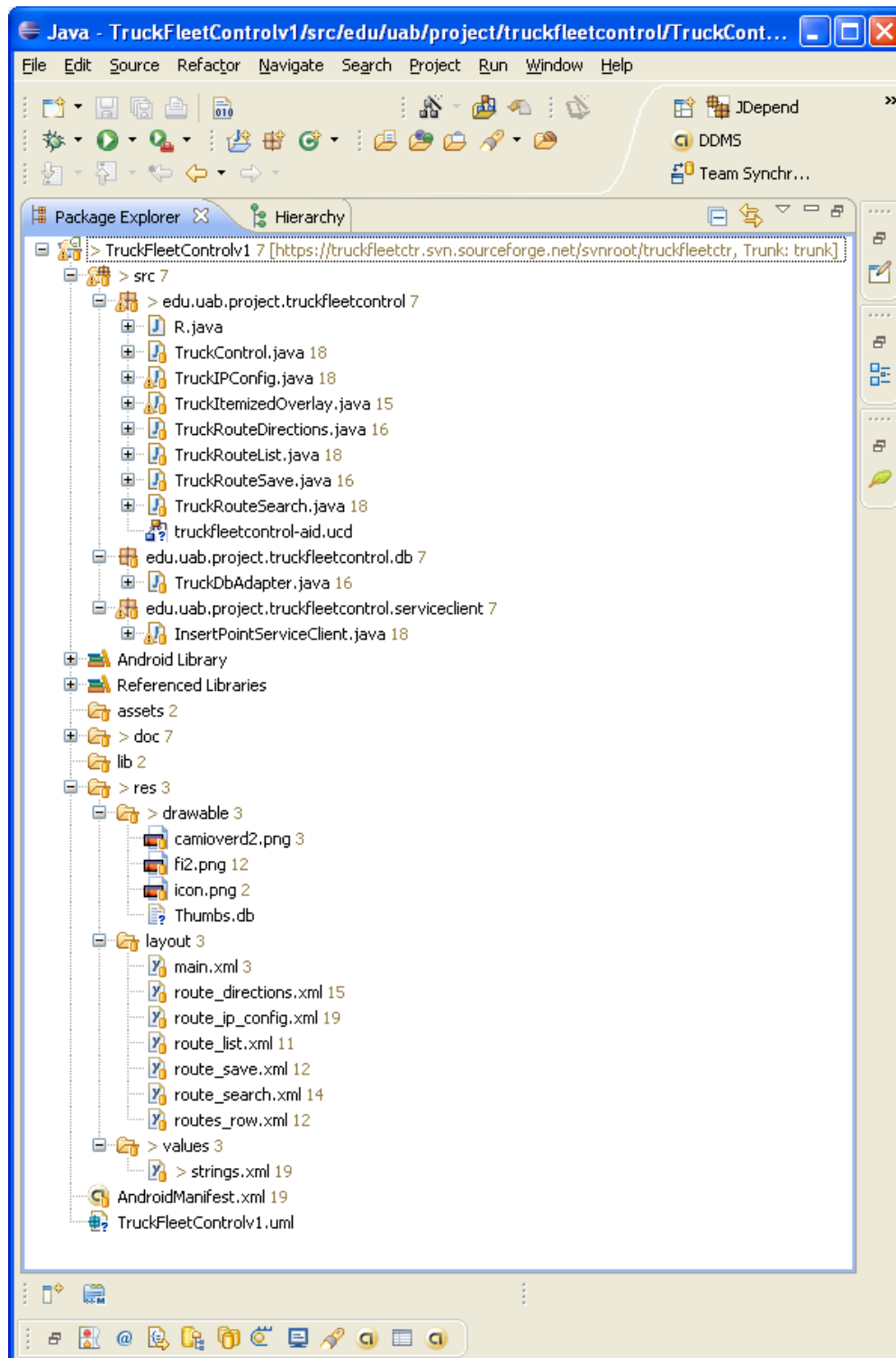
- http://es.wikipedia.org/wiki/Sistema_de_Informaci3n_Geogr3fica
- http://es.wikipedia.org/wiki/Servicio_basado_en_localizaci3n

Altres:

- <http://ksoap2.sourceforge.net/>

ANNEXOS

ANNEX I. ARBRE DE L'APLICACIÓ



ANNEX II. BONES PRÀCTIQUES

A continuació descriurem una part de les bones pràctiques recomanades per Google a l'hora de desenvolupar aplicacions per Android i n'il·lustrarem algunes d'elles amb exemples de codi de la nostra aplicació. La majoria de bones pràctiques seguides tenen per objectiu optimitzar el rendiment de l'aplicació.

1. Accedir a variables locals és sempre molt més ràpid que fer crides repetides a un mètode.

Normal:

```
if (pairs.getSize() < 1000) {  
...  
    for (int i = 1; i < pairs.getSize(); i++) {  
...  
...  
}
```

Optimitzat:

```
int pairsLength = pairs.getSize();  
if (pairsLength < 1000) {  
...  
    for (int i = 1; i < pairsLength; i++) {  
...  
...  
}
```

2. Evitar l'ús de "getters i setters" interns a la classe. En crides internes a la classe és millor accedir directament al camp de la variable en comptes d'usar crides a mètodes per recuperar el valor.

Normal:

```
if (!this.getViewMode()) {...
```

Optimitzat:

```
if (!viewMode) {...
```

3. No fer servir col·leccions multidimensionals encara que això impliqui haver de fer servir diferents col·leccions d'una sola dimensió.
4. Usar mètodes nadius en comptes de fer iteracions en Java.
5. Usar referències concretes en comptes d'interfícies. Tot i que la programació convencional diu que és millor utilitzar interfícies ja que d'aquesta forma es pot instanciar després amb qualsevol objecte que implementi la interfície, quan es

programa per plataformes encastades és millor utilitzar la classe concreta ja que una crida a través d'una referència a una interfície pot trigar fins a dues vegades més que una crida a un mètode virtual fet a través d'una referència concreta.

Normal:

```
private List<Address> addressList=new ArrayList<Address>();
```

Optimitzat:

```
private ArrayList<Address> addressList=new ArrayList<Address>();
```

6. Usar mètodes estàtics. Si no s'ha d'accedir a camps de l'objecte és millor utilitzar mètodes estàtics ja que la crida és més ràpida que en els mètodes no estàtics.

Normal:

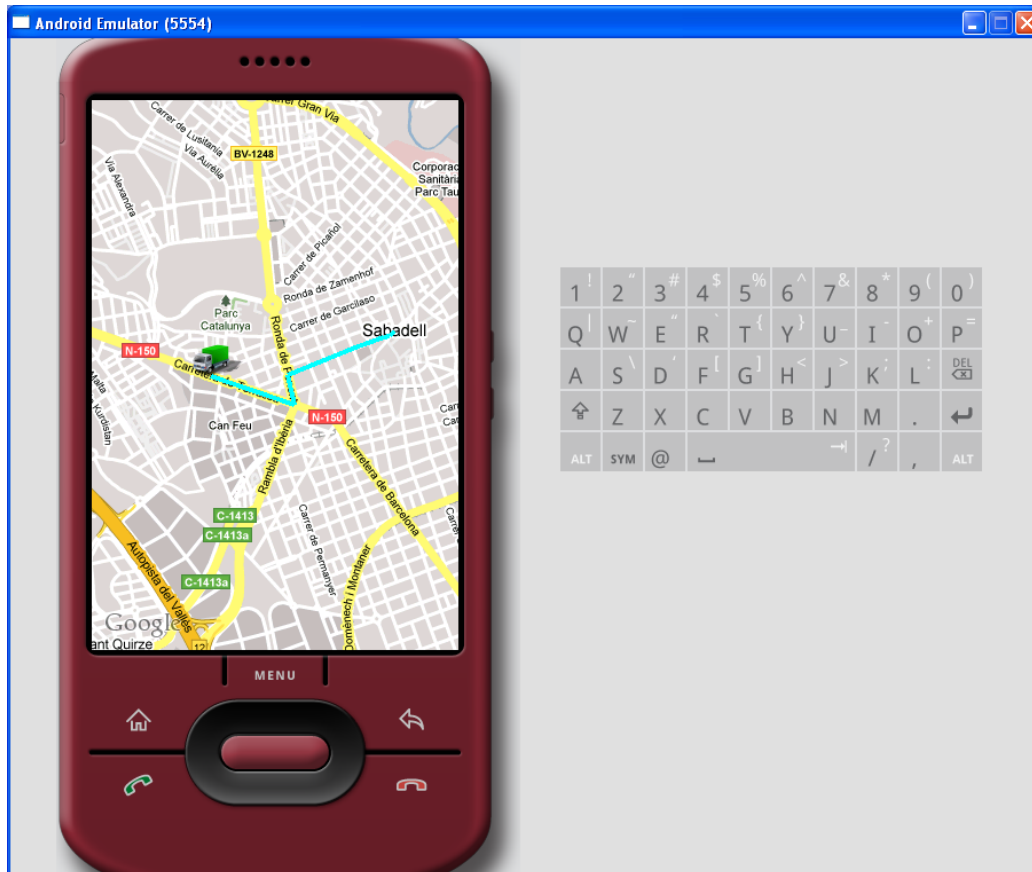
```
public String webServiceCallStatic(double latitude,  
                                double longitude,  
                                String id,  
                                String url){...
```

Optimitzat:

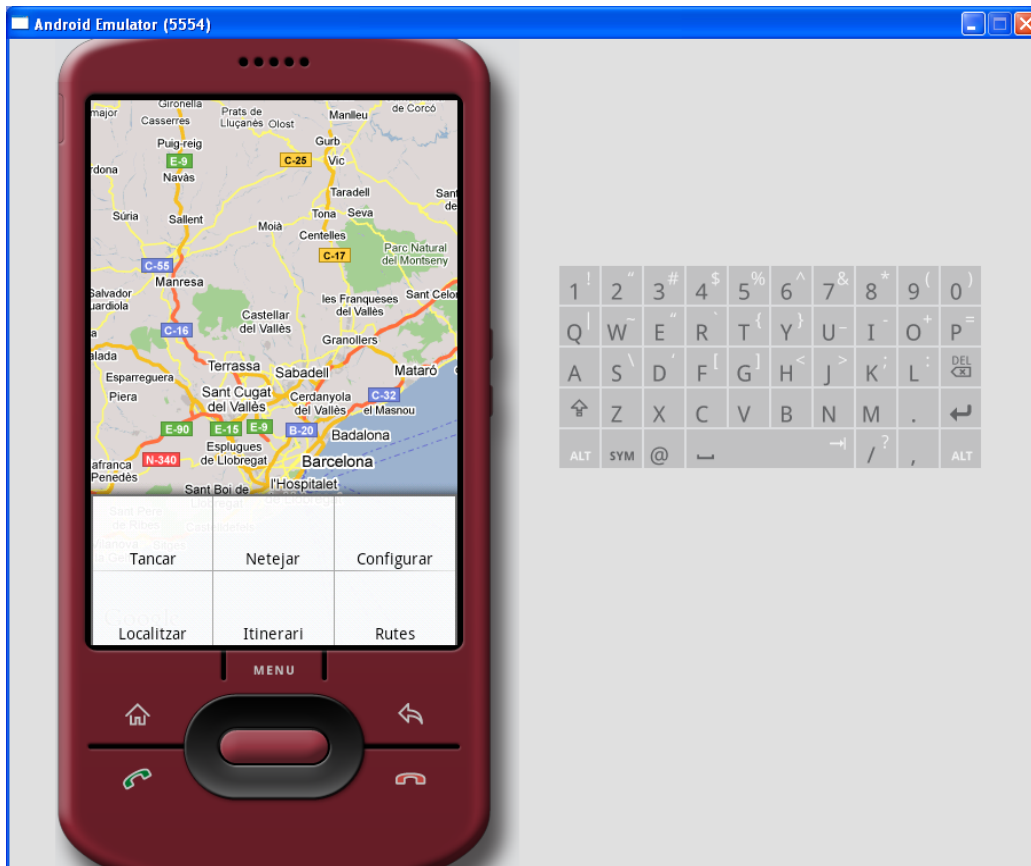
```
public static String webServiceCallStatic(double latitude,  
                                          double longitude,  
                                          String id,  
String url){...
```

7. Tot i les recomanacions en contra de les regles de la programació orientada a objecte, fer servir abast de paquet en comptes de privat en les variables membre i mètodes que han de ser usats des de classes internes és una bona optimització. En cas que la classe formi part d'una API pública potser cal plantejar-s'ho però si només forma part d'una aplicació caldria usar aquesta optimització.

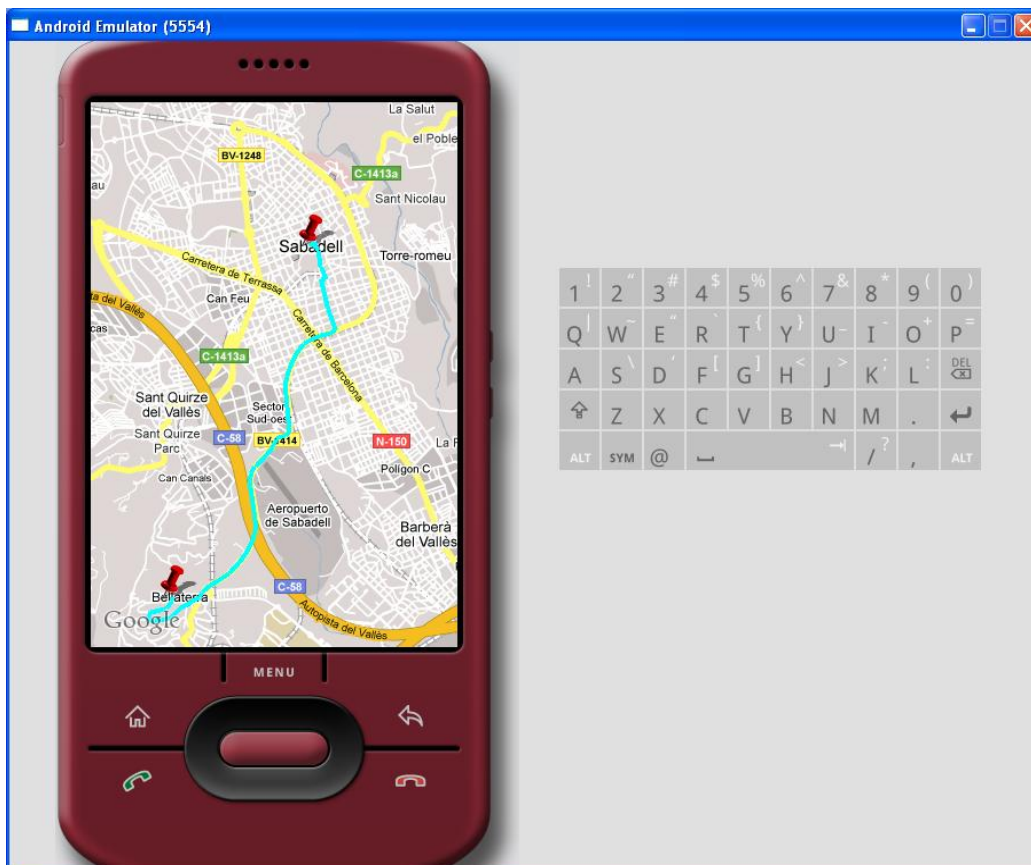
ANNEX III. PANTALLES DE L'APLICACIÓ



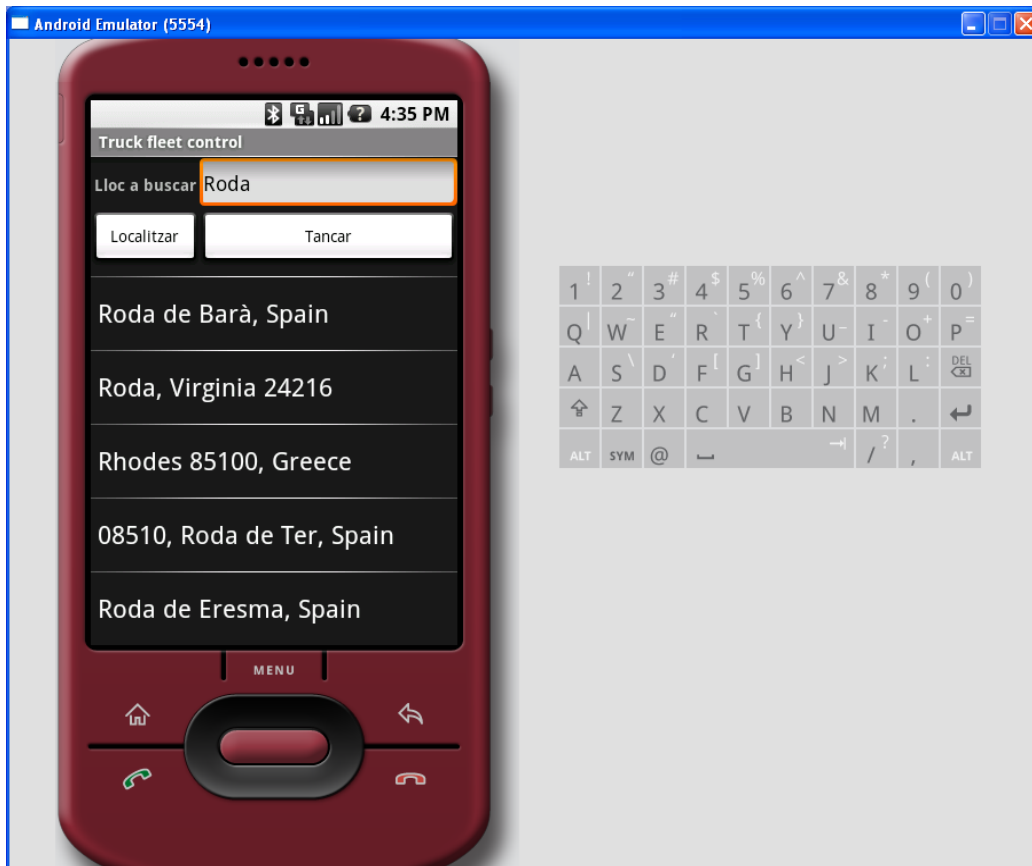
3. Pantalla de navegació



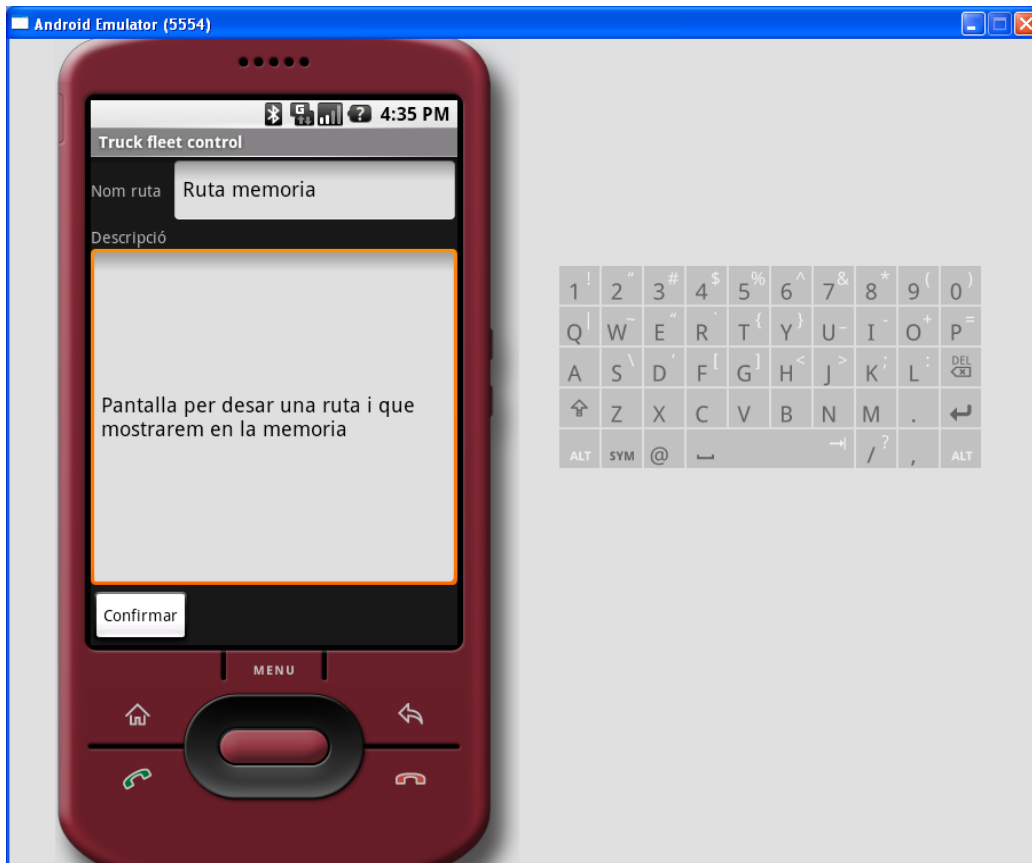
4. Menú principal



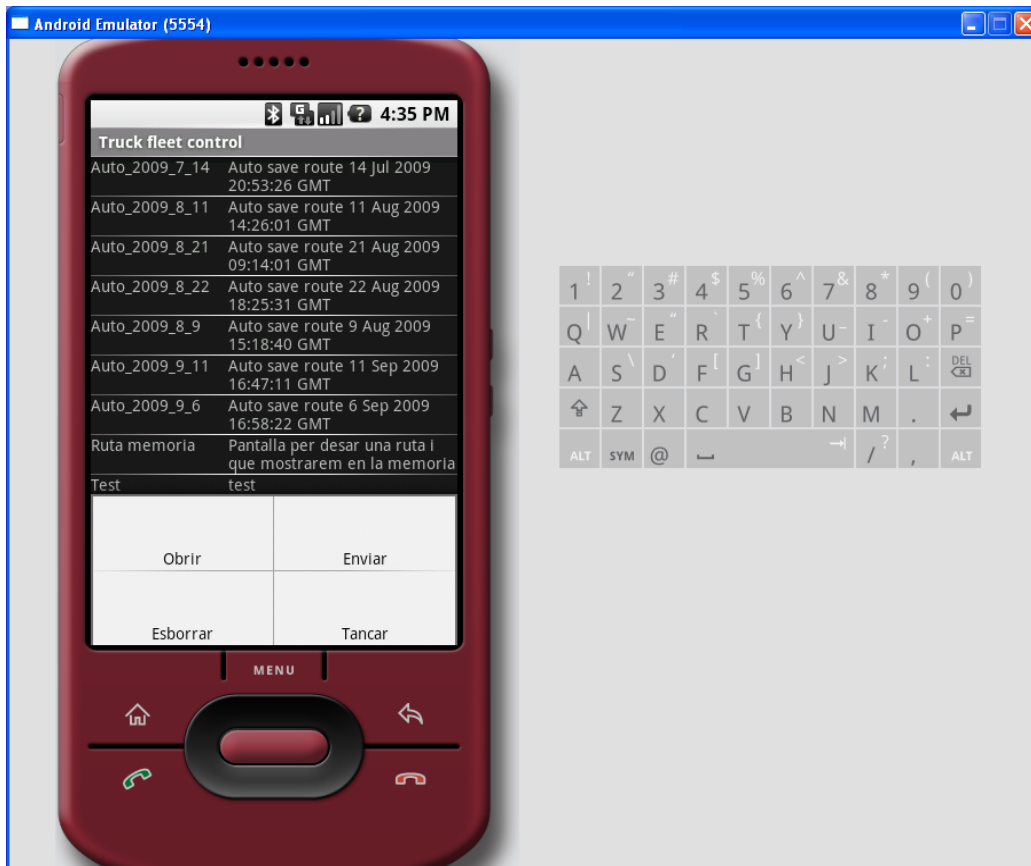
5. Resultat de cerca d'itinerari



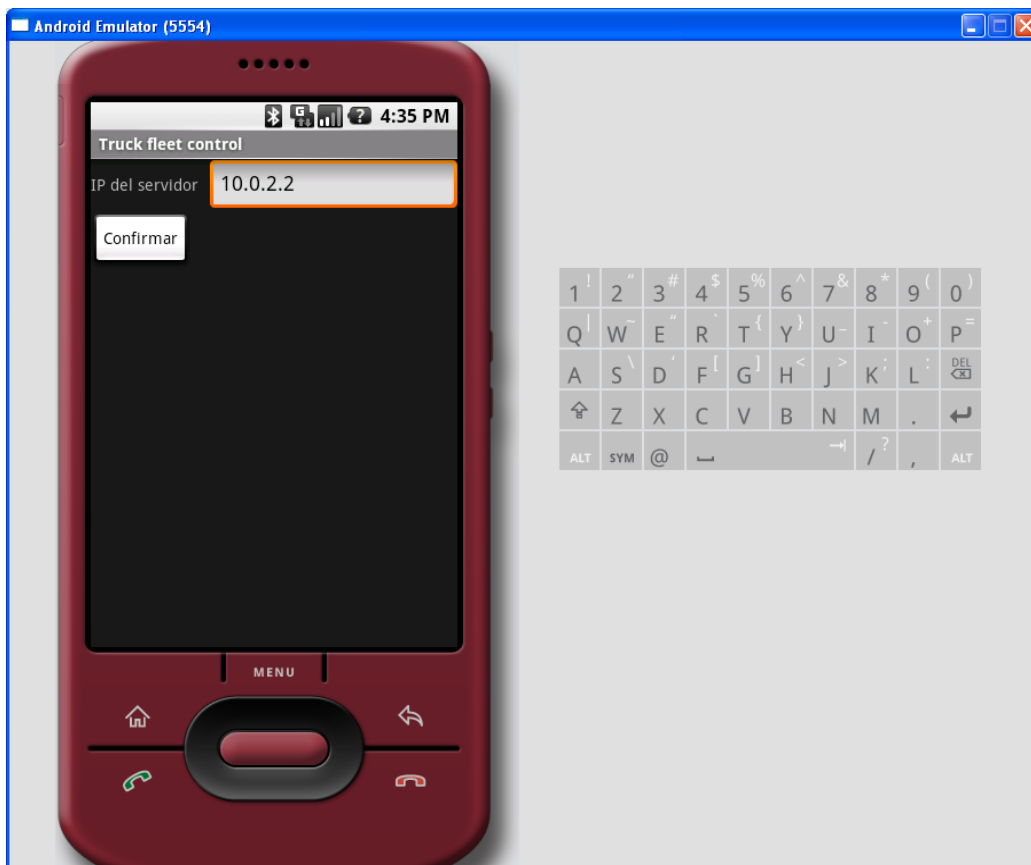
6. Resultat de cerca d'emplaçament



7. Pantalla per desar un ruta

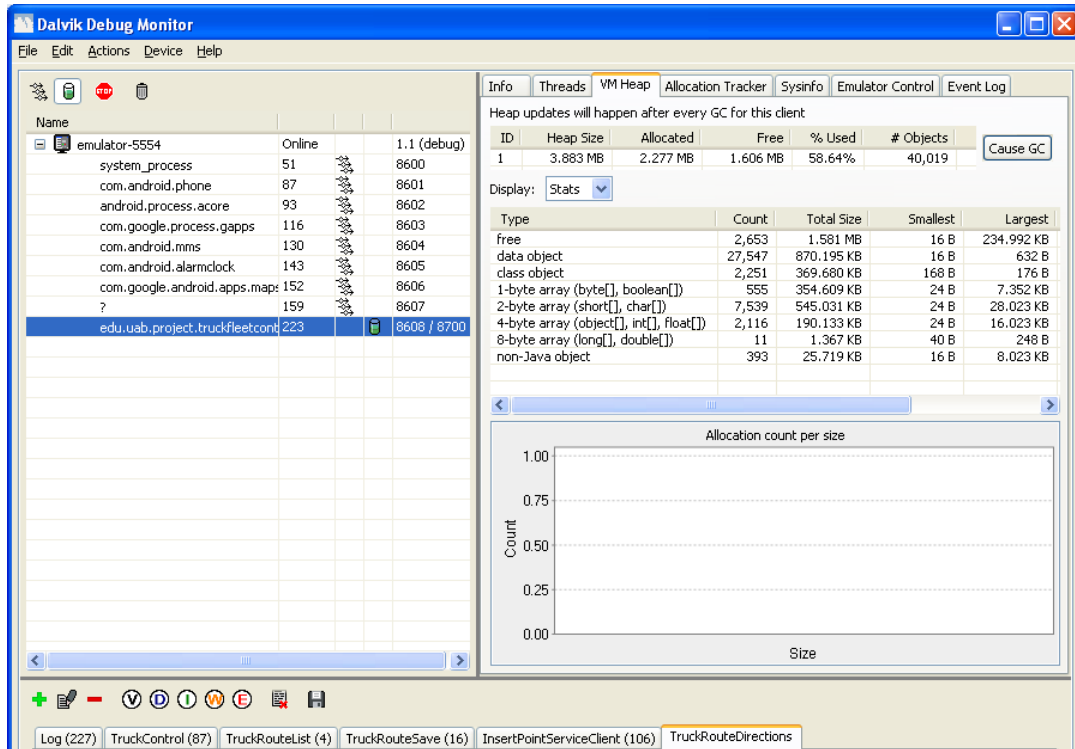


8. Llistat de rutes desades i menú associat

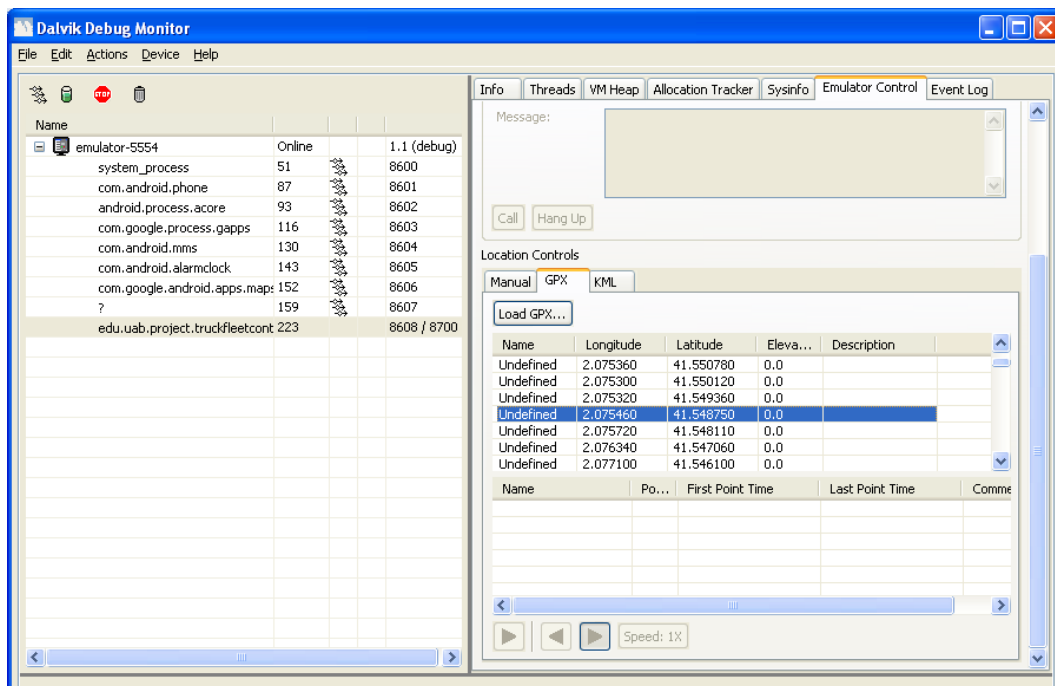


9. Configuració de l'adreça del servidor web

ANNEX IV. DDMS



10. Ús de memòria



11. Enviament simulat de coordenades de posició

ANNEX V. AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>

<!-- Paquet i versió de l'aplicació -->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.uab.project.truckfleetcontrol"
    android:versionCode="1"
    android:versionName="1.0.0">

    <!-- Permís per accedir a internet -->
    <uses-permission android:name="android.permission.INTERNET" />

    <!-- Grup de permisos per accedir als serveis de localització -->
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <user-permission android:name="android.ACCESS_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission
android:name="android.permission.ACCESS_MOCK_LOCATION" />
    <uses-permission
android:name="android.permission.CONTROL_LOCATION_UPDATES" />

    <!-- Permís per accedir a estats interns del telèfon -->
    <uses-permission
android:name="android.permission.READ_PHONE_STATE" />

    <application android:icon="@drawable/icon"
android:label="@string/app_name">

        <!-- Inclusió de la llibreria de mapes de Google -->
        <uses-library android:name="com.google.android.maps"/>

        <!-- Activitat principal (main) -->
        <activity android:name=".TruckControl"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!-- Activitats que componen l'aplicació -->
        <activity android:name=".TruckRouteSave"/>
        <activity android:name=".TruckRouteList"/>
        <activity android:name=".TruckRouteSearch"/>
        <activity android:name=".TruckRouteDirections"/>
        <activity android:name=".TruckIPConfig"/>

    </application>
</manifest>
```

2. Codi del fitxer AndroidManifest.xml comentat

Jordi Agramunt Luzón

RESUM

Aquest projecte desenvolupa la part client d'una aplicació per controlar una flota de camions. Està implementada per funcionar sobre un dispositiu mòbil que funcioni amb el sistema operatiu Android.

RESUMEN

Este proyecto desarrolla la parte cliente de una aplicación para controlar una flota de camiones. Está implementada para funcionar sobre un dispositivo móvil que funcione con el sistema operativo Android.

SUMMARY

This project develops the client system of an application that is used for control a truck fleet. It is implemented in order to work on a mobile device that uses the Android operating system.