

# Memòria: Where is my car?

**(A ón esta el cotxe? dispositius Android)**

**Dámaso López Pérez**

Enginyeria Informàtica

**Prof. Carlos Sanchez Rosa**

04-03-2013

# INDEX

0.- INDEX DE IL·LUSTRACIONS.....	4
1.- INDEX DE TAULES.....	5
10.- Pressupost.....	51
11.- Conclusions .....	53
11.1.- Sistema .....	53
11.2.- Software .....	53
11.3.- Aspectes generals .....	55
12.- Referències bibliogràfiques .....	56
2.- Descripció .....	6
2.- Introducció.....	7
2.1.- Justificació i objectius del PFC .....	7
2.2.- Funcionalitat de les diferents parts del PFC .....	7
2.3.- Mètode de desenvolupament.....	7
2.4.- Productes obtinguts .....	8
3.- Planificació del projecte .....	9
3.1.- Cicle de vida del projecte.....	9
3.2.- Planificació de les diferents tasques.....	10
3.3.- Descripció de les tasques .....	11
4.- Desviaments del pla.....	14
5.- Estudi de les tecnologies utilitzades.....	15
5.1.- Estudi tecnològic .....	15
5.2.- Arquitectura global .....	19
5.3.- Decisions tecnològiques .....	21
6.- Anàlisi de requisits .....	23
6.1.- Requisits de l'usuari .....	23
6.2.- Diagrama de casos d'ús .....	23
6.3.- Descripció dels casos d'ús .....	26
7.- Especificació de Requisits .....	31
7.1.- Requisits Funcionals .....	31
7.2.- Requisits NO Funcionals.....	32
8. Disseny. ....	33
8.1. Arquitectura i comunicació de l'app amb l'exterior. ....	33
8.2. Arquitectura bàsica d'aplicacions Android .....	34
8.3. Interfície gràfic.....	34
8.4. Diagrama de classes .....	36
8.5. Diagrames de seqüència.....	39
8.5.1. Acció "Search".....	39
8.5.2. Acció GPS On .....	40
8.5.3. Acció GPS Off .....	42
9.- Implementació WhereIsMyCar? .....	44
9.1.- Compliments en la fase de disseny amb la realitat de la fase 3.....	44

9.2.- Recursos .....	45
9.3.- Generació del projecte.....	46
9.4.- Implementació de la interface gràfic .....	48
9.5.- API's utilitzades.....	49
9.6.- Incidències en el desenvolupament .....	50

## 0.- INDEX DE IL·LUSTRACIONS

ILUSTRACIÓ 1 CICLE DE VIDA D'UNA APLICACIÓ ANDROID .....	9
ILUSTRACIÓ 2 FITES DESTACADES.....	10
ILUSTRACIÓ 3 GANTT .....	11
ILUSTRACIÓ 4 PROCÉS DE COMPILACIÓ DE JAVA.....	15
ILUSTRACIÓ 5 CICLE DE VIDA D'UN OBJECTE ACTIVITY .....	16
ILUSTRACIÓ 6 ARQUITECTURA D'UNA APP ANDROID .....	20
ILUSTRACIÓ 7 ARQUITECTURA APP WHEREISMYCAR .....	20
ILUSTRACIÓ 8 DIAGRAMA DE CASOS D'ÚS GENERIC .....	24
ILUSTRACIÓ 9 DIAGRAMA CAS D'ÚS APP.....	24
ILUSTRACIÓ 10 GOOGLE MAPS .....	31
ILUSTRACIÓ 11 COMUNICACIÓ DE L'APP AMB L'EXTERIOR .....	33
ILUSTRACIÓ 12 WIREFRAMESKETCHER .....	35
ILUSTRACIÓ 13 SCREENSHORT WIMC, COMENÇAMENT APP .....	35
ILUSTRACIÓ 14 SCREENSHORT WIMC, LOCALITZANT UBICACIÓ DEL COTXE .....	35
ILUSTRACIÓ 15 SCREENSHORT WIMC, UBICACIÓ VEHÍCLE AL MAPA .....	36
ILUSTRACIÓ 16 SCREENSHORT WIMC, RUTA FINS AL VEHÍCLE APARCAT .....	36
ILUSTRACIÓ 17 DIAGRAMA DE CLASSES.....	37
ILUSTRACIÓ 18 DIAGRAMA DE SEQÜÈNCIA "SEARCH" .....	39
ILUSTRACIÓ 19 DIAGRAMA DE SEQÜÈNCIA "GPS ON" .....	41
ILUSTRACIÓ 20 GPS OFF .....	41
ILUSTRACIÓ 21 DIAGRAMA DE SEQÜÈNCIA "GPS OFF" .....	43
ILUSTRACIÓ 22 GPS ON.....	43
ILUSTRACIÓ 23 GPS ON AMB ACTIVITAT.....	43
ILUSTRACIÓ 24 SCREENSHORT APP .....	44
ILUSTRACIÓ 25 SCREENSHORT APP 1 .....	45
ILUSTRACIÓ 26 LLIBRERIES PROPIES I EXTERNES.....	50

## 1.- INDEX DE TAULES

TAULA 1 CODIFICACIÓ DELS ROLS O ACTORS .....	25
TAULA 2 FITXERS DE CONFIGURACIÓ .....	46
TAULA 3 DIRECTORI DELS FITXERS DE CONFIGURACIÓ.....	48
TAULA 4 TAULA PRESSUPOST.....	52

## 2.- Descripció

Àrea Programació d'aplicacions mòbils.

Android és un sistema operatiu mòbil basat en Linux que ha estat desenvolupat per ser usat en dispositius mòbils com telèfons intel·ligents, tablettes entre d'altres. Està sent desenvolupat per una organització (Open Handset Alliance) liderada per Google.

Actualment compta amb una gran comunitat de desenvolupadors creant aplicacions per aquesta plataforma, i estenen així les funcionalitats de la mateixa. Donat que el llenguatge de programació és Java, fa que aquesta plataforma hagi esdevingut molt popular últimament, per la facilitat en que molts desenvolupadors poden aplicar els seus coneixements en aquest llenguatge a la nova plataforma, que bàsicament la plataforma que es dedica es JAVA amb l'ajuda de l'IDE Eclipse que fa que tot `plegat sigui més fàcil i ràpid de desenvolupar-ho.

El projecte consistirà en un localitzador del teu propi vehicle per tal d'averiguar a on esta ubicat quan desplaçem a una altra localitat que no coneixem i que ens preocupa després saber-ne la seva ubicació per tornar a casa, aquesta activitat estarà fer per la plataforma de Android (Google). Es desenvoluparà utilitzant la implementació nativa de OpenGL de la que disposa aquesta plataforma. Aquest localitzador despondrà de les funcionalitats, com pot ser geolocalització amb els graus de la ubicació, alguna dada de tipus text (carrers, número, etc.), localització amb el google maps, etc.

En resum el projecte sorgeix com a iniciació per introduir-me en el mon de la programació de dispositius mòbils, en una de les dues plataformes més grans, la de Android de Google. I que en alguna ocasió he tingut problemes per tal de localitzar el meu propi cotxe per no fer una foto al nom del carrer o el que sigui, però al final sempre el trobava.

Per tant, l'objectiu d'aquest PFC serà la de desenvolupar un localitzador de vehicles per Android per tal d'aprofundir els meus coneixements en aquestes dues àrees com es JAVA i Android. Per últim també m'agradaria remarcar, que estic treballant com a programador en JAVA, en que ara mateix no estic utilitzant Android i es una gran fita que tindre que superar, i el especial interès que tenia en el desenvolupament d'aquest PFC, per tal d'expandir els meus coneixements en aquest camp.

El projecte s'ha publicat com a programari lliure, les futures actualització i millores es poden descarregar a l'adreça:

- SVN amb el que trobarem totes les fites, codi font, etc:  
<https://xp-dev.com/svn/DSDinaMIX/PFC>

## 2.- Introducció

Aquest apartat servirà d'introducció de la memòria del Projecte Final de Carrera. Provaré de justificar la meua tria del mateix, el context en el que l'he desenvolupat, la planificació que faré servir, i la descripció general d'aquest document.

### 2.1.- Justificació i objectius del PFC

Aquest PFC es realitzarà en el context del àrea de programació de mòbils, amb el objectiu final tal com he comentat anteriorment serà la creació d'un localitzador per una de les plataformes tecnològiques, la de Android de Google, que passa per ser una de les dues més grans que es troben disponibles en el mercat actualment.

Primer de tot serà definir els requisits del mateix, per tal de donar-li forma a la idea inicial. Un cop definits els requisits inicials, el pla per portar-lo a terme, començarà amb la fase d'implementació amb un seguiment del projecte (en la forma de les diferents PACs), fins a dur a terme el producte final del mateix, el aplicació i la memòria.

En resum la meua tria d'àrea del PFC va ser principalment per introduir-me en el desenvolupament per aplicacions mòbils i les ganes d'aprendre aquesta tecnologia, i les opcions que em pot obrir en el futur. A continuació per triar la temàtica del mateix em vaig guiar per una afició personal.

### 2.2.- Funcionalitat de les diferents parts del PFC

L'objectiu del projecte és dissenyar i desenvolupar un petit APP o aplicatiu per tal de trobar la ubicació del vehicle en un destí que hipotèticament mai haves estat i no saps a on l'haves aparcats després d'un dia vèncer.

L'aplicatiu hauria de permetre aquestes operacions bàsiques:

- Permetre l'usuari ficar algunes dades de tipus text si vol.
- Rebre esdeveniments d'entrada i sortida com poden ser geolocalitzacions per GPS o GoogleMaps a l'usuari.
- Reproduir algun tipus de so per indicar si s'ha pogut guardar la ubicació abans d'anar-se'n.
- Fer vibrar el dispositiu mòbil, depenent dels esdeveniments.
- Mostrar el mapa amb una MENA de bruixola per tal d'enrutejar-ho fins al vehicle.

### 2.3.- Mètode de desenvolupament

La tècnica que s'utilitzarà serà la de l'orientada a objectes (OO), fent el desenvolupament en un llenguatge orientat a objectes. En aquest cas es farà servir Java per tot el desenvolupament, GoogleMaps per els mapes i GPS i Android com a plataforma hardware. Els objectius seran:

- Conèixer el funcionament del llenguatge Java per Android (i les llibreries que es poden usar en aquest tipus de dispositius).
- Aprofundir en la metodologia de desenvolupament d'aplicatius en plataformes mòbils.
- Aprofundir en el coneixement de la tecnologia GoogleMaps (i les característiques especials en dispositius mòbils).

- Recollida de requisits: especificació de la funcionalitat que se li vol donar a l'aplicació, tipus de dades que haurà de manipular, tipus usuaris que utilitzaran l'aplicació.
- Anàlisi i disseny: creació de les classes i diagrames que representaran l'esquelet de l'aplicació, amb el llenguatge UML Gantt i potser amb l'eina Microsoft Visio.
- Realització: implementació de l'aplicació.
- Proves i publicació: Un cop finalitzat el procés de desenvolupament, es portarà a terme un conjunt de proves per validar l'aplicatiu, i és publicarà en Google Play amb el compte que tinc.

## 2.4.- Productes obtinguts

Els productes obtinguts principalment han estat dos, l'aplicatiu i tota la documentació de la memòria. A més a més hi ha d'altres productes secundaris. A continuació veurem una descripció dels productes obtinguts.

- **Aplicatiu:** És el producte final del projecte. Construir una aplicació compatible amb mòbils Android (instal·lable apk). Hi haurà per una banda el codi font en l'entrega final, i al rebost públic de SVN. Per l'altre banda s'entregarà també el instal·lable de l'aplicatiu en format apk.
- **La Memòria / documentació:** Consisteix en per una banda aquest document com a memòria del projecte. Per altre banda als documents de les diferents PACs de les que ha consistit el projecte. A més a més hi hauran d'altres documents, com el origen del pla del projecte (Microsoft Project), diferents manuals que s'han fet servir en el desenvolupament d'aquest projecte, etc. La distribució de la documentació serà la següent:
  - **Links:** Aquí s'emmagatzemen els links que vaig visitant amb informació útil, per tal de tenir un historial.
  - **Manuals:** Els diferents manuals que han sigut d'utilitat per el desenvolupament del projecte.
  - **PACs:** Les 4 PACs de les que consta el PFC. Es podria dir que son com informes del estat.
  - **Presentacio:** Presentació final del projecte. Serà una explicació breu del projecte, a mode de petita introducció.
  - **Schedule:** Pla del projecte en format de Microsoft Project.
  - **Videos/Screenshort:** Vídeos o screenshots que s'han produït pel funcionament d'aquest, durant algunes de les practiques, i el que acompanyarà l'entrega final.



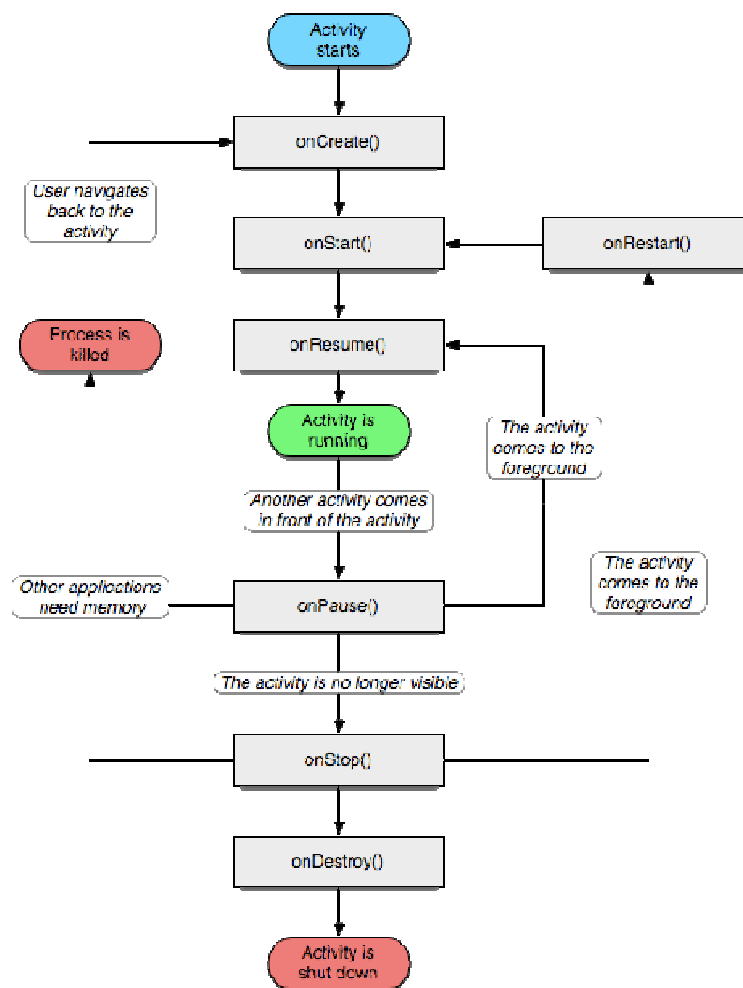
### 3.- Planificació del projecte

En aquest punt podem observar la planificació inicial, de la qual veurem una descripció exhaustiva, i com ha variat durant el temps, ja que per diferents motius he hagut de canviar una mica el planificació inicial.

#### 3.1.- Cicle de vida del projecte

El projecte de desenvolupament, com a qualsevol altre projecte de desenvolupament de programari, serà desenvolupat prenent com a base el cicle de vida clàssic de creació de programari o també conegut com a cicle en cascada. Ha estat adaptat a les necessitats de temps i requeriments del Projecte Final de Carrera, i al punt que sóc només una persona en el desenvolupament del mateix.

Com es pot observar en el gràfic següent, el projecte s'inicia en un anàlisi previ i la planificació del mateix (punt en el que estem ara mateix), a continuació analitzarem els requisits del aplicació, farem el disseny, continuarem amb la fase d'implementació del programari i les proves del mateix, per terminar amb la finalització del producte.



Il·lustración 1 Cicle de vida d'una aplicació Android

### 3.2.- Planificació de les diferents tasques

La planificació del projecte, com es pot veure quedarà dividida en 4 grans fases (Recollida de requisits / Documentació, Anàlisi i disseny de l'aplicació, Implementació del aplicació, Memòria i entrega).

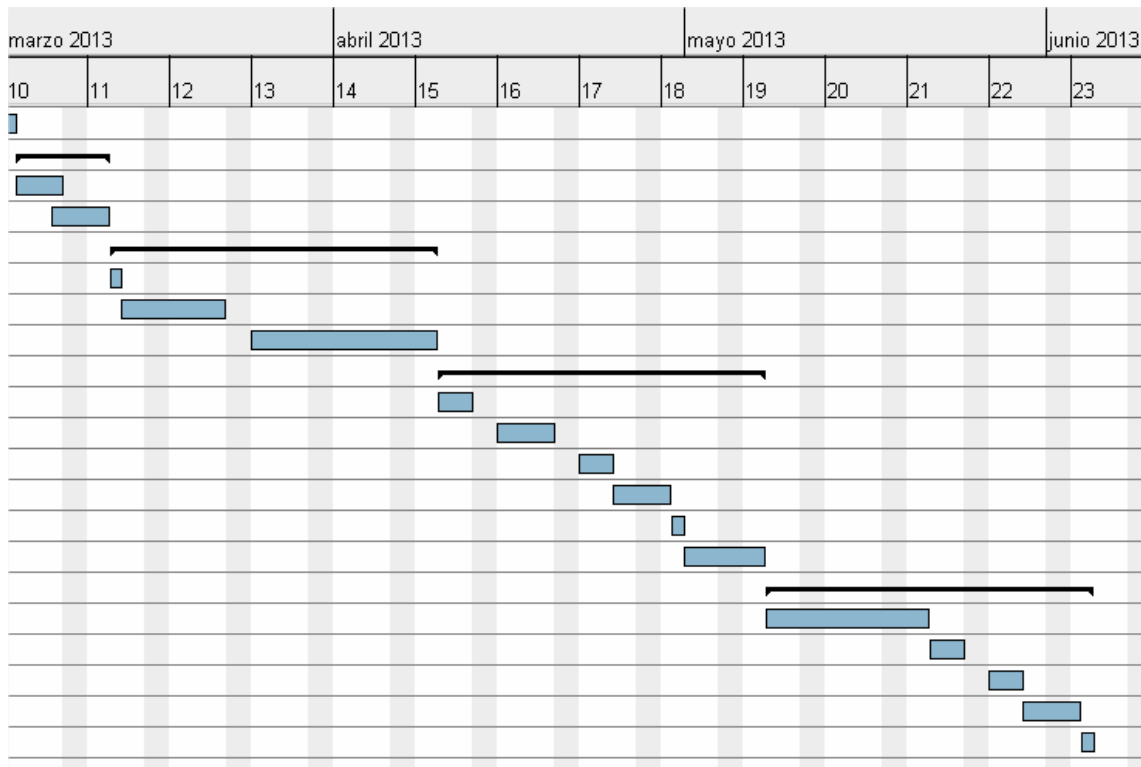
Aquestes contindran una sèrie de subtasques relacionades entre elles, i que per tal de dur a terme la tasca principal, s'hauran de completar totes les subtasques associades. El llistat de les mateixes i les dates associades a cadascuna d'elles es pot veure a continuació

Nombre	Fecha ...	Fecha ...
Realització pla de treball	4/03/13	5/03/13
[-] Recollida de requisits /Documentació	5/03/13	13/03/13
Requisits de l'aplicació	5/03/13	10/03/13
Documentació d'Android	8/03/13	13/03/13
[-] Anàlisi i disseny de l'aplicació	13/03/13	10/04/13
Definir requisits	13/03/13	14/03/13
Documents d'anàlisi	14/03/13	25/03/13
Documents de disseny	25/03/13	10/04/13
[-] Implementació de l'aplicació	10/04/13	8/05/13
Menús de l'aplicació	10/04/13	15/04/13
Diferents moduls	15/04/13	20/04/13
Acces a dades (txt,xml,etc)	22/04/13	25/04/13
Documentació del projecte	25/04/13	30/04/13
Joc de proves	30/04/13	1/05/13
Finalització de l'aplicació	1/05/13	8/05/13
[-] Memòria i entrega	8/05/13	5/06/13
Redactar memoria	8/05/13	22/05/13
Preparar presentació	22/05/13	27/05/13
Preparar video	27/05/13	30/05/13
Preparar producte final	30/05/13	4/06/13
Entrega final	4/06/13	5/06/13

#### Ilustración 2 Fites destacades

Com es pot veure en el gràfic següent al ésser un projecte portat a terme una sola persona, no hi hauran tasques que es solapin en el temps, ni que es portin a terme de manera paral·lela, ja que aquest punt no serà possible en un projecte unipersonal.

Algunes conclusions que es poden treure, es que la part dedicada a l'implementació és la més gran de totes les tasques, i potser la dedicada a la finalització de la memòria es una mica curta. Aquestes només son unes conclusions preliminars, observant aquest gràfic, més endavant tractarem el tema de la validesa de la planificació inicial, els desviaments que s'han dut a terme i els motius del mateix.



Il·lustració 3 Gantt

### 3.3.- Descripció de les tasques

En aquest apartat es descriurà més en detall en que consisteixen cadascuna de les tasques i planificades, el temps que es trigarà en dur-les a terme, els objectius i fites de la mateixa, tenint en compte que també tinc 4 assignatures més a part d'aquesta.

#### Tasca 1:

**Temporització:** 2 dies (del 4 al 5 de Març).

**Descripció:** Realització del document "Pla de treball":

- Estudi de la magnitud del projecte.
- Decisió de les tecnologies a utilitzar.
- Temporització del projecte.

**Objectius:**

- Aprofundir els coneixements sobre el desenvolupament Android.
- Tenir una visió clara sobre la funcionalitat que haurà de tenir l'aplicatiu.
- Tenir una visió de les funcionalitats que haurà el mateix.

**Fites:**

- Document del Pla de treball.

**Observacions:**

Comentar que el temps es curt perque ja aquesta idea la tenia fa temps,estic fent un curs d'android i he estat documentant-m'he.

## Tasca 2:

**Temporització:** 2 setmanes (del 5 al 13 de Març).

**Descripció:** Recollida i classificació de tota la informació que pugui ser rellevant per dur a terme el projecte:

- Temes relacionats amb el desenvolupament Android.
- Temes relacionats amb el desenvolupament GoogleMaps per Android.

**Objectius:**

- Aprofundir els coneixements sobre programació Java per la plataforma Android.
- Tenir una visió clara sobre la funcionalitat que haurà de tenir l'aplicatiu (fer un esborrany en paper).

**Fites:**

- Document descriptiu del desenvolupament Android.
- Document relatiu sobre API / framework per el desenvolupament .
- Document del pla de treball (entrega i si es pot entregar abans de la data).
- PAC 1 del PFC

## Tasca 3:

**Temporització:** 4 setmanes (13 de Març al 10 d'abril).

**Descripció:** Anàlisi i disseny de l'aplicatiu a implementar.

**Objectius:**

- Establir es requisits que haurà de tenir.
- Obtenir l'anàlisi del programari en funció dels requisits i de la funcionalitat que se li vol donar.
- Obtenir el disseny de l'aplicació a implementar basant-nos en l'anàlisi del punt anterior.

**Fites:**

- Obtenir l'anàlisi i el disseny de l'aplicació documentant les decisions preses.
- PAC 2 del PFC

## Tasca 4:

**Temporització:** 6 setmanes (10 d'abril al 8 de Maig).

**Descripció:** Implementació.

**Objectius:**

- Implementar entre el 70-85% de codi font.
- Generar els jocs de proves adients per garantir el correcte funcionament.
- Finalitzar la implementació del codi.
- Continuar amb la redacció de la memòria.

**Fites:**

- Creació de l'aplicatiu.
- Creació del document amb les especificacions de l'aplicatiu.
- Finalització de la fase de desenvolupament.
- Document sobre la implementació i el joc de proves.

## Tasca 5:

**Temporització:** 4 setmana (del 8 de Maig al 5 de Juny).

**Descripció:** Documentació del producte. Arranjaments finals de la memòria i preparar la presentació del aplicació que s'ha creat.

**Objectius:**

- Documentar la instal·lació de l'aplicatiu.
- Documentar l'ús de les diferents opcions de configuració i / o jugabilitat del mateix.
- Síntesi de la memòria presa durant el projecte.
- Crear una presentació del aplicació creat.

**Fites:**

- Documentació del producte creat.
- Memòria del projecte.
- Presentació de l'aplicatiu.
- PAC 4 - Entrega final.

## 4.- Desviaments del pla

Pel que fa al desviament de la planificació he pogut observar que entre l'entrega de la primera fita i aquesta segona hi ha hagut un petit desviament d'un parell de dies a conseqüència de les correccions i re-estructuració ja sigui de la part tècnica com pot ser la documentació, unificació i modulació dels diferents accions que tindrà l'aplicació com que no quedava massa clar les vistes o pantalles ja que al primer de tot tenia fet un esboç aproximat de les pantalla i amb l'ajuda del punt 6 d'aquesta fita i d'un software que he utilitzat he lograt aquest objectiu per portar-ho al final.

No obstant, el tema de la planificació de les posteriors fites com la implementació no queda afectada, ja que mentre que feia aquest anàlisi anava avançant la implementació com pot ser l'estructura del source, creació del projecte a l'entorn IDE d'Eclipse, creació d'un subversió per tindre-ho tot organitzat i així estar respaldat per algun esdeveniment ocasional com pot ser pèrdua de dades, etc.

Pel tema del diagrama GANTT com he dit a l'anterior punt queda igual i pel moment no es modificarà, com ja tinc el vist-i-plau d'aquesta fita "anàlisi del sistema" paso a realitzar la implementació que serà una tasca ardua però confortable a la vegada.

Pel que fa a la fita 3 d'aquesta entrega pot haver-hi un petit desviament a causes d'altres entregues ja que no s'ha pogut generar la documentació pertinent de la implementació, aquest desviament pot ser d'un parell de dies però que a la fi no hi genera cap problema pels darrers fites 4 i l'entrega de la memòria.

## 5.- Estudi de les tecnològies utilitzades

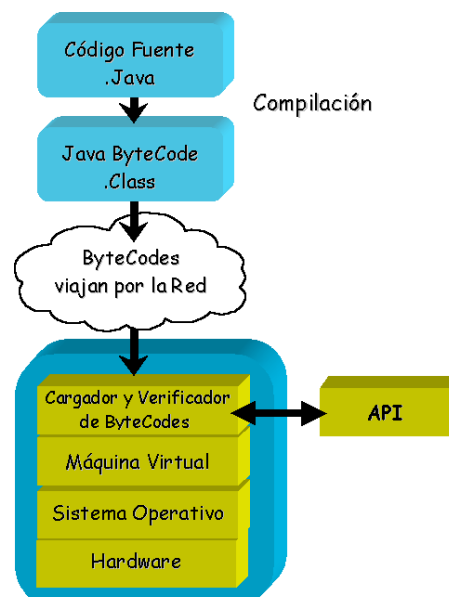
### 5.1.- Estudi tecnològic

#### Relació Java amb el desenvolupament mòbil

Java és un llenguatge interpretat i, per tant, pot semblar lent en comparació amb altres llenguatges, però ofereix un índex de re-utilització de codi molt elevat, sent possible trobar moltes llibreries lliures de Java. És un llenguatge flexible i potent tot i la facilitat amb la què es programa i dels resultats que ofereix. Un dels trets que el caracteritza i que el fa una eina molt valorada a l'hora de desenvolupar aplicacions distribuïdes, és el fet que és un llenguatge multi-plataforma.

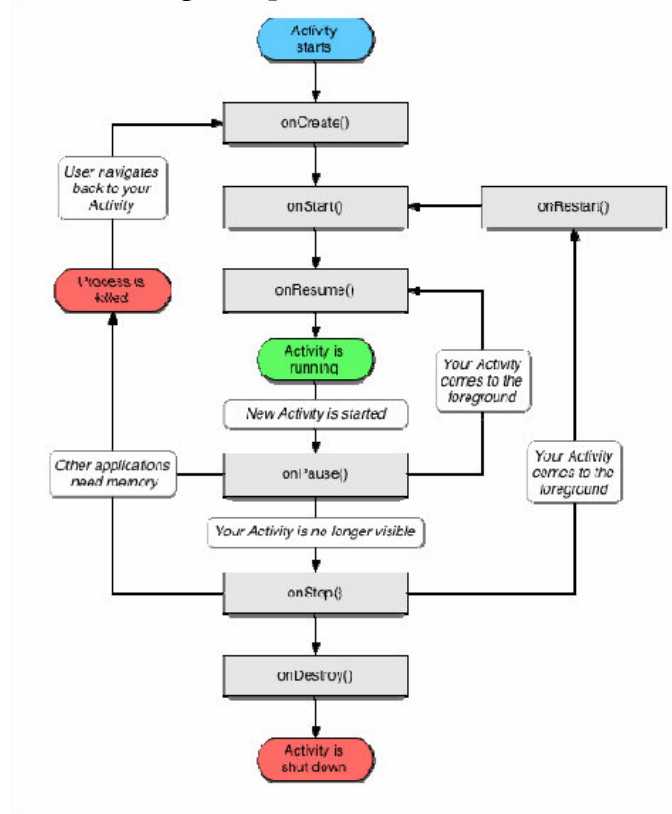
Generalment els programes de Java es compilen en un bytecode (fitxer .class) que pot córrer en una Màquina Virtual Java. Sun Microsystems disposa de tres implementacions diferents de Java: J2SE per a aplicacions d'escriptori; J2EE per a aplicacions distribuïdes i J2ME per a plataformes amb recursos més reduïts com ara mòbils o PDAs. Per a cada una de les tres implementacions és possible descarregar el JRE (entorn d'execució Java) per a executar aplicacions o el SDK (Eines per al desenvolupament d'aplicacions) per a programar aplicacions en Java, aquest últim també inclou el JRE.

El procés de compilació del llenguatge Java introdueix algunes novetats, com la generació d'un codi intermedi que serà interpretat per una maquina virtual, on s'executarà aquest codi intermedi o "bytecode", i no en el sistema operatiu pròpiament dit, com es pot observar en el següent gràfic:



Il·lustració 4 Procés de compilació de JAVA

Pel contrari Android té el següent procés o cicle de vida:



Il·lustración 5 Cicle de vida d'un objecte Activity

L'ADT (Android Development Tools) que és de Google proveeix com a "framework" de desenvolupament d'aplicacions per Android, juntament amb Eclipse que es el IDE suportat oficialment. ADT conté totes les funcionalitats requerides, per crear, compilar i desplegar aplicacions per Android, utilitzant tant Eclipse com la línia de comandes.

Les aplicacions Android estan escrites principalment en Java, les quals seran compilades i convertides en \*.class per el JDK. Un cop això, el ADT proveeix d'una eina anomenada "DX" que converteix aquestes classes Java en fitxers \*.dex. Durant aquest procés de conversió s'optimitza la classes i s'emmagatzemen totes dintre d'un únic fitxer, i per tant tindrà una mida molt més petita que el conjunt de classes original.

Un cop realitzat aquest pas, aquest fitxer \*.dex <sup>1</sup> i tots els recursos de l'aplicació (imatges, fitxers XML, música, etc), s'empaquetaran dintre d'un fitxer \*.apk <sup>2</sup> que serà el instal·lable o binari del nostre projecte. Aquest fitxer resultant doncs,

5656565616

<sup>1</sup> APK (Application Package File) És un format per a paquets d'aplicacions del sistema operatiu Android. El format es tracta d'una variant del format JAR, de la plataforma Java i s'utilitza per a distribuir o instal·lar components empaquetats tant per a versions destinades a smartphones com a tauletes. Un arxiu .apk normalment conté els fitxers següents:

- AndroidManifest.xml



es podrà instal·lar en dispositius Android via la eina "adb" (manualment) , a través de Google Play o en altres tipus com poden ser des del sistema de fitxers local, o d'un repositori alternatiu, com Amazon, GetJar, etc..

## Sistemes de posicionament geogràfic.

En l'actualitat hi ha sistemes de posicionament geogràfic com el GPS que cada vegada són més accessibles per a tots els públics.

El GPS és un sistema global de navegació per satèl·lit que permet determinar en tot el món la posició d'una persona, vehicle, objecte, etc amb gran precisió. Se li atribueix la seva invenció als governs francès i belga encara va ser desenvolupat i operat pel Departament de Defensa dels Estats Units.

El GPS funciona mitjançant una xarxa de 32 satèl·lits en òrbita sobre l'esfera terrestre que es mouen amb trajectòries sincronitzades per cobrir tota la superfície global. Al moment en què es pretén obtenir la posició, el receptor (P.ex: telèfon mòbil o el del cotxe) localitza automàticament com a mínim tres satèl·lits de la xarxa, dels quals es rep uns senyals indicant la identificació i l'hora del rellotge de cadascun d'aquests.

Tenint com a base aquests senyals, l'aparell sincronitza el rellotge del GPS i calcula el temps que triguen a arribar els senyals a l'equip i així mesura la distància al satèl·lit mitjançant el mètode de triangulació, basat en el mesurament de les distàncies de cada satèl·lit al punt desitjat. Amb això els satèl·lits obtenen una posició relativa a si mateixos. Per obtenir la posició absoluta del receptor GPS només cal calcular davant les posicions dels satèl·lits

Un dels usos més estesos dels GPS portàtils ha estat per al sector del automòbil. S'han desenvolupat GPSs que en realitat són mapes interactius que mostren

- 
- classes.dex
  - resources.arsc
  - res (carpeta)
  - META-INF (carpeta)

Talment com els JAR, els arxius apk no deixen de ser arxius ZIP amb una extensió diferent i, per tant, poden ser inspeccionats amb programari arxivador de fitxers com ara el 7-Zip, el Winzip, el WinRAR o l'Ark. El tipus MIME definit per a .apk és application/vnd.android.package-archive.

<sup>2</sup> DEX (Dalvik EXecutable) És la màquina virtual que utilitza la plataforma per a dispositius mòbils Android. DALVIK ha estat dissenyada per Dan Bornstein amb contribucions d'altres enginyers de Google. Està optimitzada per requerir poca memòria i està dissenyada per executar diverses instàncies de la màquina virtual, delegant en el sistema operatiu subjacent el suport d'aïllament de processos, gestió de la memòria, fils, etc.

No obstant, l'eina dx inclosa en el SDK d'Android permet transformar els arxius \*.class (fitxers compilats de JAVA) al format d'arxius DEX.

rutes i carreteres. Donat un punt d'inici i un punt final el GPS et calcula un recorregut i et va donant informació detallada de com arribar teu destí. Aquest dispositiu s'ha convertit en una gran eina per a professionals del transport, taxistes o per a qualsevol persona evitant la necessitat de consultar un mapa en paper durant la conducció o parant per a això.

Hi ha molts d'aquests dispositius que no només ofereixen informació sobre rutes, sinó que a més ofereixen localització de restaurants, cinemes, gasolineres properes. La situació d'aquests llocs es guarden en la memòria del receptor, juntament amb els mapes de carreteres i plànols dels carrers. Així pot avisar a l'usuari si està a prop d'un. Els mapes i aquests puntons d'interès s'actualitzen a través dels mitjans que el fabricant del receptor posi a disposició per a això, com en CD-ROM o al seu lloc web.

Avui dia ja es poden veure GPSs que a més de localitzacions d'interès també ofereixen l'estat del trànsit. Molts dels GPS actuals poden comunicar-se amb Internet per rebre informació actualitzada sobre el trànsit o incidències en la ruta. La comunicació es realitza en general a través del telèfon mòbil. El receptor GPS es connecta amb un servidor per rebre dades en temps real. Així pot avisar d'un carrer tallada, embussos en hora punta i poder buscar una ruta alternativa.

Existeixen fabricants de programes de cartografia per a GPS que disposen de serveis de informació de trànsit. Amb alguns d'ells, com l'empresa holandesa TomTom, és necessari realitzar subscripció i pagar una quota anual. Altres ofereixen el mateix servei sense necessitat de quotes, aquesta estaria ja inclosa en el preu de compra del programa de navegació

Android, pel que fa a sistema operatiu es refereix, han estat molts els fabricants de telèfons mòbils els que han utilitzat, com a sistema operatiu, una versió de Linux encastat. Empreses com Samsung, Motorola, HTC, Nokia, LG, Siemens han confiat en les prestacions i potència que aquest sistema oferia.

També cal recalcar que gràcies a l'enorme avenç tecnològic patit pel mercat dels dispositius mòbils, desembocaria en la necessitat de desenvolupar sistemes operatius més especialitzats en aquesta classe d'arquitectures i és així com avui en dia existeixen diversos sistemes operatius especials per a telèfons mòbils com el Symbian, Windows Mobile, IOS o l'Android.

El desembre de 2006 tant la BBC com el Wall Street Journal van publicar que Google estava apostant fort perquè la seva cercador i les seves aplicacions funcionessin en els telèfons mòbils

Google va ser fent passos fins que el novembre de 2007 la Open Handset Alliance es va marcar l'objectiu de definir els estàndards oberts per als

dispositius mòbils. Google va alliberar la majoria del codi d'Android sota llicència Apache, la qual és una llicència lliure i de codi obert. L'octubre de 2008 surt al mercat el primer telèfon que opera en Android. Era el G-1 de T-Mobile, només distribuït als EUA i Regne Unit.

Encara que han existit versions prèvies, és a partir de la 1.5 que es comença a tenir en compte a Android com un sistema operatiu realment potent. Anomenarem les versions més conegudes o últimes:

- Android 1.6, coneguda com Donut, es va llançar al setembre de 2009 i es la considera una actualització menor pel fet que es basava fonamentalment en la correcció d'errors presents en l'anterior. També és cert que les correccions propiciar l'aparició d'algunes novetats com una nova pantalla per controlar la bateria, que permetia comprovar quines aplicacions i serveis són els que més consumeixen, o un nou motor de text a veu.
- Android 2.2, coneguda com Froyo, va ser llançada el juny de 2010 i els canvis més nous són actualitzacions automàtiques per a aplicacions, funcionalitat de Ràdio FM, compatibilitat amb Wi-Fi IEEE 802.11n, compatibilitat amb Adobe Flash 10.1, compatibilitat amb l'API gràfica de OpenGL Embedded Systems 2.0, etc.
- Android 2.3, coneguda com Gingerbread, és l'última versió estable. Aquesta versió ha estat llançada recentment, el 6 de desembre de 2010. El primer mòbil comercialitzat que té aquesta versió de fàbrica és el Nexus S i els dispositius previstos per al seu llançament en els primers trimestres de 2011.

Algunes de les novetats que es mostren són millores en les funcionalitats de tallar, copiar i enganxar, millora la interconnexió amb les xarxes socials, reconeixement facial, gestor de descàrregues, etc.

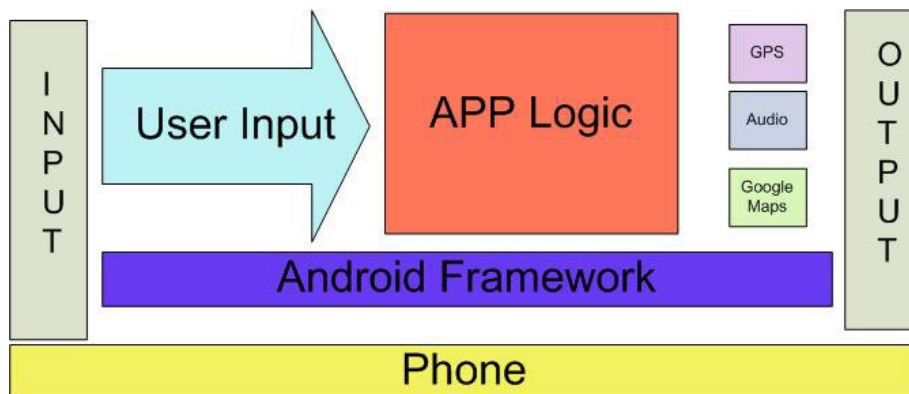
## 5.2.- Arquitectura global

Les aplicacions per a dispositius mòbils, Android en el cas d'aquest projecte tindran normalment una arquitectura com es pot veure en el gràfic següent:



Il·lustració 6 Arquitectura d'una app Android

En el esquema anterior podem veure una arquitectura típica d'Android però pel nostre cas o descriurem de la següent forma i trobarem les diferents capes del sistema operatiu del nostre telèfon, Android OS en el cas que ens ocupa.



Il·lustració 7 Arquitectura APP WhereIsMyCar

A continuació descriurem la resta de components del programa:

- ✿ **"Input":** En aquest nivell trobarem la pantalla tàctil, o el teclat del telèfon, el micròfon, la càmera, els acceleròmetres, etc.
- ✿ **"User input":** En el nostre producte, les entrades del usuari, serà quan pressioni la pantalla tàctil del dispositiu, a les àrees que definirem com els controls del app, o els diferents botons dels diferents menús.

- ✿ **“APP Logic”**: El mòdul de la lògica del nostre programa, contindrà tot lo referent al control dels diferents estats del posicionament del vehicle, les notes o dades que volguéssim guardar, la petita configuració si calgués, etc. Com moure’ns pel fons de pantalla, controlar el posicionament del vehicle pel GoogleMaps i qualsevol altre component “lògic” del mateix.
- ✿ **“GPS”**: Aquest és el mòdul responsable de mostrar pel GoogleMaps la posició exacta del vehicle, en aquest cas el mòdul es extern a la nostra aplicació ja que es pròpia de Google i que tindrem accés total a ella.
- ✿ **“Àudio”**: Aquest mòdul produirà els sons adients per cada estat dem l’app si fos necessari per tal d’avisar d’alguna forma a l’usuari dels diferents protocols o els incidents que pugui tindre, o per cada esdeveniment que així ho necessiti.
- ✿ **“GoogleMaps”**: Aquest és el mòdul responsable de mostrar un mapa de la zona amb els noms dels carrers i assenyalar la ubicació del cotxe.
- ✿ **“Output”**: La sortida o output seria el conjunt de la sortida dels mòduls d’àudio, GPS i del mapa (potser també la vibració del telèfon si així ho hem decidit).

### 5.3.- Decisions tecnològiques

A continuació detallaré les decisions tecnològiques que s’han dut a terme per el desenvolupament del app, i els motius que las van motiva. Algunes com es podrà veure venen obligades per el tipus de plataforma triada per el desenvolupament del projecte:

- **Llenguatge de programació**: El llenguatge triat serà Java en la seva versió 7 (l’ultima disponible). Aquesta tria be donada per Android, i el seu SDK que obliga al desenvolupament usant Java.
- **Framework de desenvolupament**: Si es vol desenvolupar per Android, i no fer-ho a molt baix nivell, s’ha de fer servir el ADT (Android Development Tools), i per tant no hi havia cap opció si es volia treballar amb aquesta plataforma.
- **Gràfics**: En principi de gràfics no hi haura, ja que tot com per exemple el mapa vindrà donat pel mateix framework de google que ja dona un codi que solsament s’haura d’injectar a la nostra programació.

Per una altra banda haurem d'especificar per a quins dispositius volem que el nostre producte funcioni, per fer proves, he decidit de que la API sigui la 4 perque tingui com a referencia els dispositius que tinguin la versió d'Android 1.6, com es la meva, no obstant més endavant es podria modificar aquesta decisió i fer que la mínima fos la versió 11.

## 6.- Anàlisis de requisits

En aquest apartat veurem tot lo relacionat amb el disseny de la nostra aplicació, a on està el vehicle aparcats en aquest cas. Parlarem de l'arquitectura, el flux del producte i les classes que formaran el mateix.

L'elecció d'Android com a plataforma sobre la qual anirà l'aplicació té dos raons fonamentals:

- És un sistema operatiu basat en GNU / Linux i es presenta com una bona alternativa basada en programari lliure enfront d'altres sistemes propietaris com el IOS d'Apple. De la mateixa manera el codi de l'aplicació, desenvolupat amb Java, un cop finalitzat serà llicenciat, en un principi, sota llicència LGPL per poder contribuir així a la comunitat del programari lliure.
- Android cada vegada està més implantat en dispositius mòbils. Al principi era una "insígnia" d'HTC, una de les empreses de telèfons mòbils que col·laboren en el desenvolupament i millora d'Android, però actualment altres marques com Motorola i Samsung també ho estan implantant en els seus dispositius com a mostra de confiança a l'estabilitat i seguretat d'aquest sistema Operatiu.

### 6.1.- Requisits de l'usuari

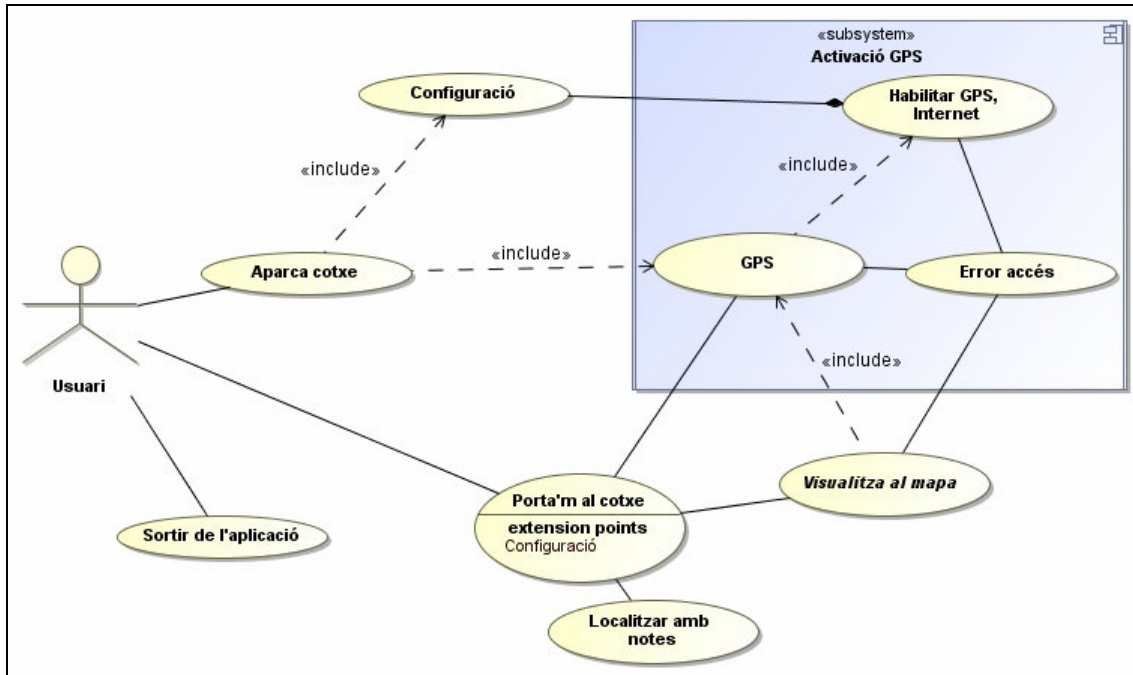
Cal que l'usuari disposi d'un dispositiu mòbil amb sistema operatiu Android. A més el mòbil, com a mínim ha de tenir connexió amb Internet, bé sigui per wifi o 3G, ja que serà a través d'Internet que l'aplicació obtindrà informació, tant de la seva pròpia localització, el mapa de Google Maps o la localització de les places d'aparcament.

Per obtenir la localització del dispositiu mòbil, cert és que n'hi hauria prou amb tenir connexió a internet, però seria convenient, per a una localització més precisa, tenir inclòs un sistema de GPS

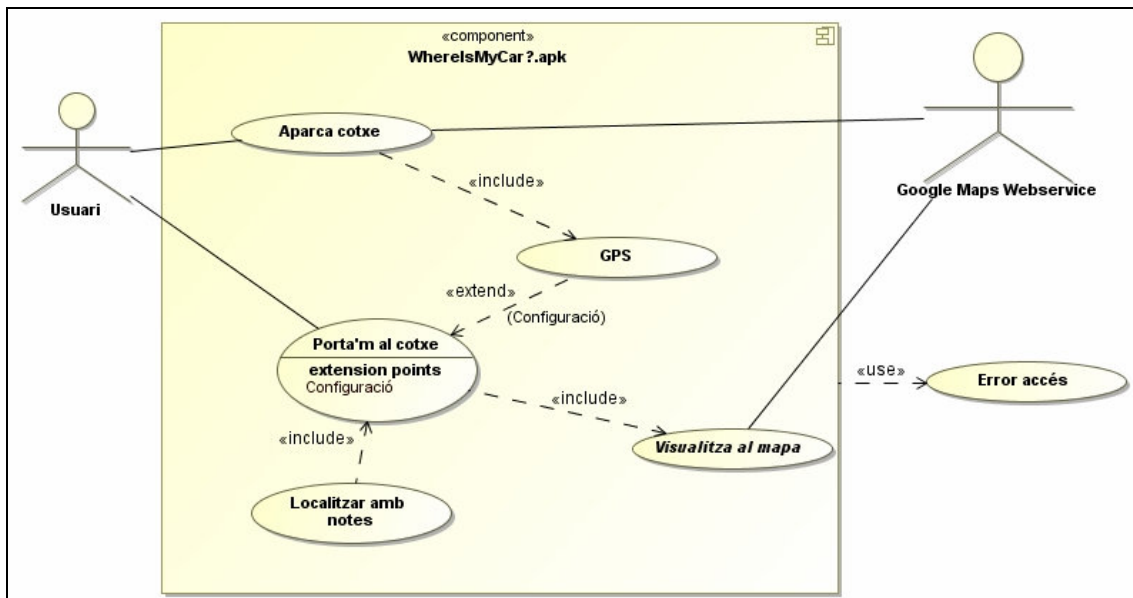
En aquesta secció analitzarem la fase de recollida de requisits del sistema. Per dur a terme aquesta tasca, partirem d'una vista general dels casos d'ús, i a continuació detallarem cadascun dels casos, es formalitzaran i concretaran els requisits no funcionals de cadascun dels requisits.

### 6.2.- Diagrama de casos d'ús

El següent diagrama recull una vista global del actor "usuari" i els casos d'ús que descriuen als requisits funcionals de l'aplicació:



Il·lustració 8 Diagrama de casos d'ús generic



Il·lustració 9 Diagrama cas d'ús app

Com es pot observar, el sistema distingirà un sol usuari pels casos d'ús anteriors, l'usuari que representarà, serà el que interactuarà amb el sistema. Aquest usuari, com es veu en el gràfic anterior, interactuarà accedint i sortint del sistema, modificant les diferents opcions del mateix.



Aquest sistema també disposarà d'una configuració previa que haurem de tindre en compte pel funcionament, haurem d'habilitar la senyal GPS e Internet per tal d'accedir al mapa i així mostrar la localització directa d'on esta situat el vehicle. A mode de resum, a la següent taula es poden observar una llista de tots els casos d'us que desenvoluparem amb més detall en el següent punt.

**Taula 1 Codificació dels rols o actors**

Codi	Descripció	Actor
Conf	Configuració	Usuari
HGpsInternet	Habilitar GPS & Internet	Usuari, GPS
SUser	Sortir de l'aplicació	Usuari
ACotxe	Aparca cotxe	Usuari
PCotxe	Porta'm al cotxe	Usuari
LNotes	Localitza amb notes	Usuari
GPS	Localitza amb GPS	GPS
VMapa	Visualitza al mapa	GPS
EAcces	Error d'accés	GPS,Usuari

### 6.3.- Descripció dels casos d'ús

En aquest punt descriurem en detall cadascun dels casos d'us que s'han detectat en el apartat anterior, per tal de fer un estudi més en profunditat de cadascun d'ells.

#### Conf Configuració

Identificador	Conf
Nom	Configuració de l'aplicació
Autor	Dámaso López
Resum	L'usuari inicia l'aplicació en el seu dispositiu Android, previament ha de tindre habilitat el GPS e Internet.
Actor	Usuari
Precondicions	Habilita el GPS e Internet
Postcondicions	Cap.
Flux normal	Pot accedir aparcar el vehicle
Flux alternatiu	En cas de que no estigui habilitat dona informació a l'usuari i dona permís per tal d'habilitar-ho.
Inclusions	HGpsInternet
Exclusions	EAcces

#### SUser Surtir

Identificador	SUser
Nom	Sortir aplicació
Autor	Dámaso López
Resum	L'usuari decideix sortir de l'app
Actor	Usuari
Precondicions	El usuari ha iniciat el programa i es troba en el menú principal
Postcondicions	Un cop tancada la aplicació el usuari es troba de tornada al escriptori o al menú, depèn d'on es trobés abans d'iniciar la aplicació.
Flux normal	<ol style="list-style-type: none"> <li>1. El usuari ha d'anar al menú principal</li> <li>2. El videojoc mostrarà el menú principal</li> <li>3. Seleccionar la opció de "EXIT"</li> <li>4. El programari es descarrega de memòria i surt</li> </ol>
Flux alternatiu	Es podria donar el cas que el usuari utilitzi el botó del que

	<p>disposen tots els dispositius Android, el de tornar endarrere. En aquest cas també es pot sortir del nostre programariç</p> <ol style="list-style-type: none"> <li>1. En qualsevol punt del videojoc, el usuari pot prémer aquest botó</li> <li>2. Depenen del punt on es trobi, l'haurà de prémer tantes vegades com sigui necessari fins sortir de l'aplicació</li> <li>3. El programari es descarrega de memòria i surt</li> </ol>
Inclusions	IUser
Exclusions	Cap

### HGPSInternet Habilita GPS, Internet

Identificador	HGpsInternet
Nom	Habilita GPS e Internet
Autor	Dámaso López
Resum	-
Actor	Usuari
Precondicions	En el cas de que l'usuari no hagi habilitat el GPS aquest cas d'ús li donarà el permís i després informará a l'usuari.
Postcondicions	<p>L'usuari triarà quina de les opcions vol, com aparcar el cotxe o porta'm al cotxe.</p> <p>En el cas que que no es pugui habilitar el GPS, veure flux alternatiu</p>
Flux normal	Habilitar GPS, Internet; enviar-li una notificació i procedir a l'aparcament del vehicle
Flux alternatiu	Si hagut un error, informar a l'usuari que no es pot continuar amb el procediment.
Inclusions	Conf, ACotxe, GPS
Exclusions	EAcces

### ACotxe Aparca cotxe

Identificador	ACotxe
Nom	Aparca cotxe
Autor	Dámaso López
Resum	L'usuari vol aparcar el cotxe i per aixó accedeix aquesta opció i es guarda automàticament la posició del GPS
Actor	Usuari, GPS

Precondicions	L'usuari ha de tindre habilitat el GPS i esperar un parell de segons fins que agafi la posició correcta. Aquesta precondició serà establerta una vegada quan l'usuari s'instal·li la nostra app.
Postcondicions	Cap
Flux normal	Agafa el dispositiu, habilitant el GPS i espera un segons fins que es troba la localització i el guarda per tal de mostrar-ho
Flux alternatiu	Veure cas d'ús Pcotxe i VMapa
Inclusions	Conf, HGpsInternet ,GPS, Pcotxe
Exclusions	EAcces

### PCotxe Porta'm al cotxe

Identificador	PCotxe
Nom	Porta'm al cotxe
Autor	Dámaso López
Resum	L'usuari vol tornar a casa i accedeix al producte per trobar-ne la localització i arribar al cotxe on l'havia aparcats
Actor	Usuari, GPS
Precondicions	L'usuari ha de tindre habilitat el GPS
Postcondicions	A tingut que guardar previamente la localització amb GPS o bé per notes
Flux normal	Agafa el dispositiu, habilitant el GPS i espera un segons fins que es troba la localització i li dona amb un mapa la ruta
Flux alternatiu	Donar-li la localització per dades o notes que havia guardat previamente, com carrer, foto,... per mapa de GoogleMaps e indicar-li la corresponent ruta.  Veure cas d'ús Lnotes i VMapa
Inclusions	Conf, HGpsInternet, ACotxe, GPS, VMapa, LNotes
Exclusions	EAcces

### LNotes Localitza amb notes

Identificador	LNotes
Nom	Localització amb notes
Autor	Dámaso López

Resum	Amb unes simples notes o algun tipus de foto que ha fet abans d'anar-se'n després d'aparcar el seu vehicle per tal de retornar al lloc on l'havia aparcat
Actor	Usuari
Precondicions	Ha de notar qualsevol dada que es pugui conèixer la ubicació
Postcondicions	Pot indicar amb googleMaps o un altre aplicatiu per tal de'arribar al lloc o indicar-li algú transeunt
Flux normal	Tecleja qualsevol ubicació com el carrer que está, el bloc,... o fer una petita foto del lloc a on l'aparcat
Flux alternatiu	Cap
Inclusions	Conf, HGpsInternet, ACotxe, GPS, VMapa, LNotes
Exclusions	EAcces

## GPS Sistema de posicionament global

Identificador	GPS
Nom	Sistema de posicionament global
Autor	Dámaso López
Resum	Amb l'ajuda del GPS podrem veure al mapa a on està ubicat el cotxe
Actor	Usuari, GPS
Precondicions	Haver guardat primerament la localització per tal de tornar a veure-la
Postcondicions	Cap
Flux normal	Una vegada obtingut la localització la podrem veure a traves de GoogleMaps
Flux alternatiu	La mateixa que el flux normal
Inclusions	Conf, HGpsInternet, ACotxe, GPS, VMapa, LNotes
Exclusions	EAcces

## VMapa Veure la localitza amb mapa

Identificador	VMapa
Nom	Localització amb mapa
Autor	Dámaso López
Resum	Localització del vehicle amb l'ajuda del GPS i el mostrarà al GoogleMaps
Actor	Usuari, GPS
Precondicions	Haver guardat primerament la localització per tal de tornar a veure-la en el mapa i donar una ruta exhaustiva

Postcondicions	Cap
Flux normal	Una vegada obtingut la localització la podrem veure a traves de GoogleMaps i veure-la per fer-li la ruta
Flux alternatiu	La mateixa que el flux normal
Inclusions	Conf, HGpsInternet, ACotxe, GPS
Exclusions	EAcces

Així doncs com a resum dels casos d'us, hem vist que hi ha un actor, serà l'usuari que executarà l'app amb les accions que s'han declarat.

## 7.- Especificació de Requisites

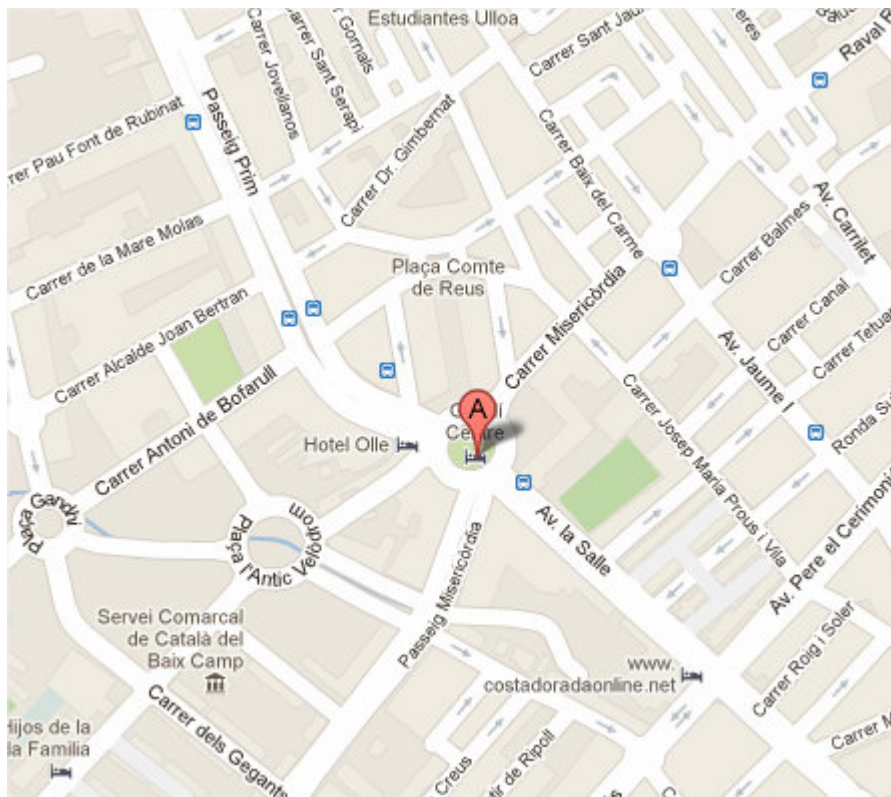
A continuació es defineixen l'especificació de requisits de sistema per l'aplicació a desenvolupar. Aquests requisits es divideixen en: Requisits Funcionals i Requisits no Funcionals.

A continuació es detalla cada un d'ells.

### 7.1.- Requisits Funcionals

Per a l'aplicació a desenvolupar l'usuari disposarà d'una interfície gràfica, el cùal està format per una senzilla barra d'eines a la part superior i a la resta de la interfície haurà un mapa, obtingut de Google Maps.

A la barra d'eines tindrà un camp de text per introduir una adreça o localització, un botó per efectuar la recerca de la localització i un botó GPS que se situarà al mapa la localització de l'usuari.



Il·lustració 10 Google Maps

Quan es prem el botó GPS, l'aplicació mostrarà la localització de l'usuari sobre el mapa. Per això recorrerà al millor proveïdor de localitzacions que disposi el

dispositiu mòbil. Cal fer èmfasi en que perquè aquesta funció tingui efecte el mòbil on correrà l'aplicació hauria de tenir almenys un proveïdor de localització, com Wifi, 3G, GPS, etc.

GPS interactuarà amb els serveis proporcionats per Google Maps per poder representar la informació al mapa.

## 7.2.- Requisites NO Funcionals

D'aquesta aplicació s'espera que:

- En estar visualitzant la localització del cotxe en el mapa es reflecteixi en temps real al mapa mostrat per l'aplicació.
- Si es realitza un zoom del mapa segons sigui un zoom d'allunyament o d'acostament.
- Mostreu com a mínim la localització del vehicle al mapa, sigui quin sigui el mètode per obtenir-la.
- Les cerques es limiten a resultats que es puguin obtenir a Google Maps, això implica que no es representaran elements que Google Maps no identifiqui o no localitzi.
- La funció de localització del dispositiu al mapa, per al cas que aquesta s'obtingui mitjançant GPS, només funcionarà en zones i àrees per als quals estigui habilitat el GPS, és a dir, on funcioni la llicència.

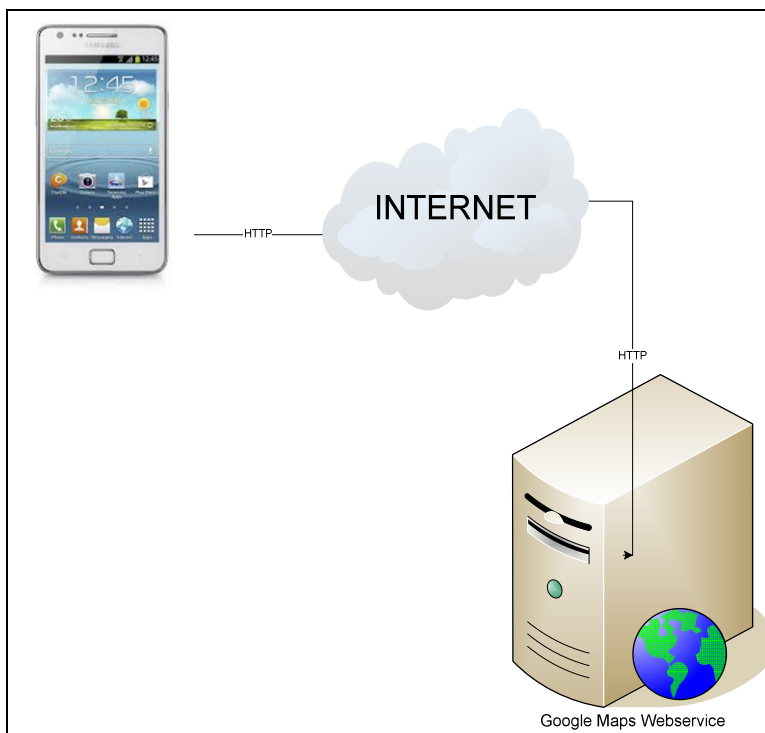


## 8. Disseny.

### 8.1. Arquitectura i comunicació de l'app amb l'exterior.

L'arquitectura de l'aplicació funcionaria de la següent manera:

Els WebServices o serveis web eren coneguts com un sistema software dissenyat per suportar operacions i interaccions màquina a màquina sobre una xarxa. Així mateix se'ls considera com un conjunt de protocols i estàndards que serveixen per intercanviar dades entre aplicacions.



**Il·lustració 11** Comunicació de l'app amb l'exterior

Aquestes aplicacions, que poden estar escrites en llenguatges de programació diferents i executades sobre qualsevol plataforma, poden fer ús dels serveis web per intercanviar dades en xarxes d'ordinadors tal com és Internet. Aquesta interoperabilitat s'aconsegueix gràcies a l'adopció d'estàndards oberts.

En recolzar-se en HTTP, els Webservices poden aprofitar-se dels sistemes de seguretat firewall sense necessitat de canviar les regles de filtrat. A més permeten que serveis i programari de diferents companyies ubicades en diferents llocs geogràfics puguin ser combinats fàcilment per proveir serveis integrats.

Encara que molts serveis web són de pagament, els webservices de Google Maps són gratuïts, disponibles per a tothom. Tan sols cal sol·licitar una clau per poder utilitzar l'aplicació.

Aquests serveis web envien peticions HTTP a URL específiques, transmetent als serveis paràmetres d'URL com a arguments. Normalment, aquests serveis retornen les dades de la sol·licitud HTTP en format JSON o XML perquè l'aplicació els analitzi i processi.

## 8.2. Arquitectura bàsica d'aplicacions Android.

Una aplicació en Android està definida pel contingut de la seva manifestació. Cada aplicació Android declara totes les seves activitats, punts d'entrada, interfícies de comunicació, permisos i intencions a través del `AndroidManifest.xml`.

A més es podrien esmentar 4 blocs principals en què es divideix l'aplicació:

- Activity : És el bloc bàsic de l'aplicació. Una activitat utilitza una sèrie de funcions per interactuar amb el seu entorn. Específicament, una activitat de sobrecarregar el mètode "onCreate". També és recomanable sobrecarregar els mètodes "onStop", "onPause", "onResume" i "onKeyDown". Amb aquestes funcions s'aconsegueix gestionar de manera més precisa l'aplicació, més endavant referenciarem per activitat o procés.

- Intent Receiver: Un objecte "reactiu" llançat per manejar una tasca específica. Es troba a l'espera d'intents d'accions per poder manejar quan són rebudes.

- Service: Un procés llançat en segon pla i per tant no tindrà cap interacció amb l'usuari.

- Content Provider: Framework per al maneig i emmagatzematge de dades.

## 8.3. Interfície gràfic.

Aquest serà l'aspecte de l'aplicació. S'ha realitzat amb l'ajuda de WireframeSketcher que ens facilita el treball de crear les diferents pantalles de la nostra app.

Aquesta eina és de pagament, i per a l'ús he utilitzat la prova que ens dona de 15 dies.



Il·lustració 12 WireframeSketcher



Il·lustració 13 Screenshot WIMC, començament app



Il·lustració 14 Screenshot WIMC, localitzant ubicació del cotxe



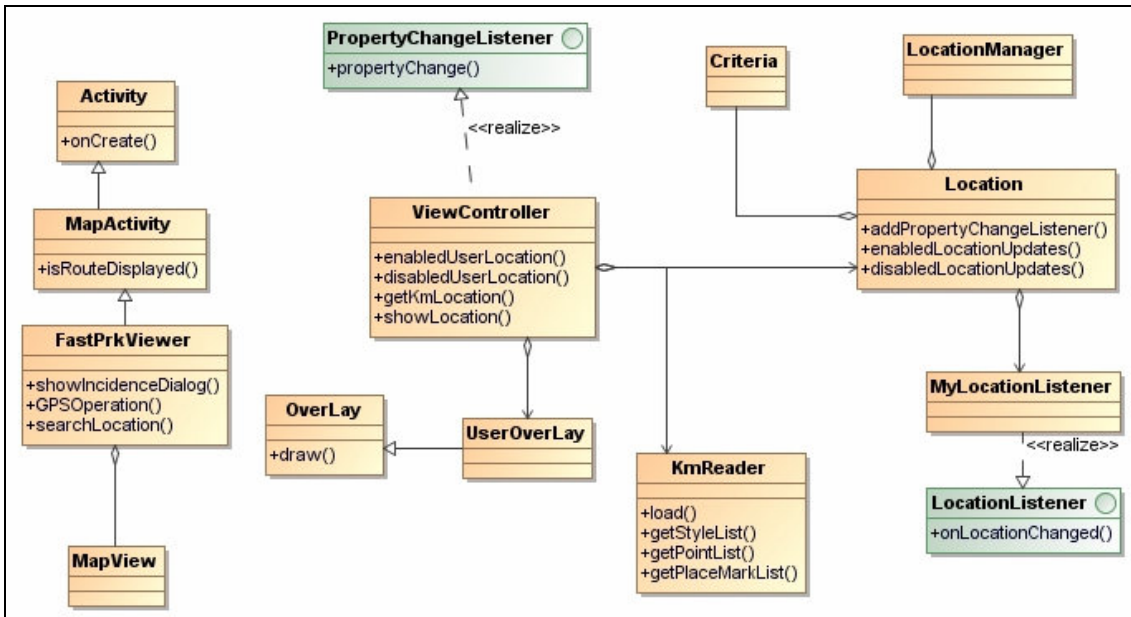
Il·lustració 15 Screenshot WIMC, ubicació vehicle al mapa



Il·lustració 16 Screenshot WIMC, ruta fins al vehicle aparcad

## 8.4. Diagrama de classes

El diagrama de relació entre classes que tindrà l'aplicació serà el següent:



Il·lustració 17 Diagrama de classes

A continuació s'explicarà la comesa de cadascuna d'aquestes classes:

- Activitat: com bé s'ha dit més amunt és el bloc bàsic de l'aplicació en Android. Una activitat utilitza una sèrie de funcions per interactuar amb el seu entorn. En general sobrecarregarà el mètode "onCreate" el qual s'utilitza comunament per construir tot la interfície gràfica. També és recomanable sobrecarregar els mètodes "onStop", "onPause", "onResumen" i "onKeyDown".

Amb aquestes funcions s'aconsegueix gestionar de manera més precisa l'aplicació.

- MapActivity: És una classe bàsica d'Android que s'estén d'Activity en la qual es pot incloure un mapa. Aquesta classe maneja el cicle de vida de l'aplicació i gestiona els serveis en background per poder mostrar el mapa per pantalla.

Cal que MapActivity contingui un atribut de la classe MapView

- FastPrkViewer: Classe bàsica de l'aplicació. Aquesta classe estén de MapActivity. S'encarrega de carregar la interfície gràfica definit en el fitxer main.xml i instanciar la resta de classes necessàries per a l'aplicació. Dos dels seus mètodes, GPSOperation i searchLocation són accions associades als botons de la interfície gràfica.

- MapView: Classe original d'Android. Aquesta classe serà la que permeti el maneig del mapa, és a dir, serà la classe utilitzada per mostrar per pantalla una ubicació, un punt, etc.

- ViewerController: Classe que conté la lògica per al maneig del mapa i les diferents fonts de localitzacions. Aquesta classe s'implementa de la interfície PropertyChangeListener. Exactament aquesta classe es troba a l'escolta de canvis en la localització del vehicle y0020usuario a través de la classe Locator. Cada vegada que rebi un canvi aquest serà representat al mapa amb la nova localització.

- Overlay: Classe original d'Android utilitzada per situar una imatge sobre el mapa.

- UserOverLay: Classe que s'estén de superposició. Es utilitza per situar sobre el mapa la icona que representi la localització de l'usuari del mòbil i la ubicació del cotxe.

- Locator: Classe que conté la informació per obtenir les coordenades de localització amb els mètodes disponibles al mòbil. A més de contenir un llistat de listeners de tipus PropertyChangeListener. Quan es detecti un canvi de

localització li ho notificarà a tots els listeners subscrits, en aquest cas serà la classe ViewerController.

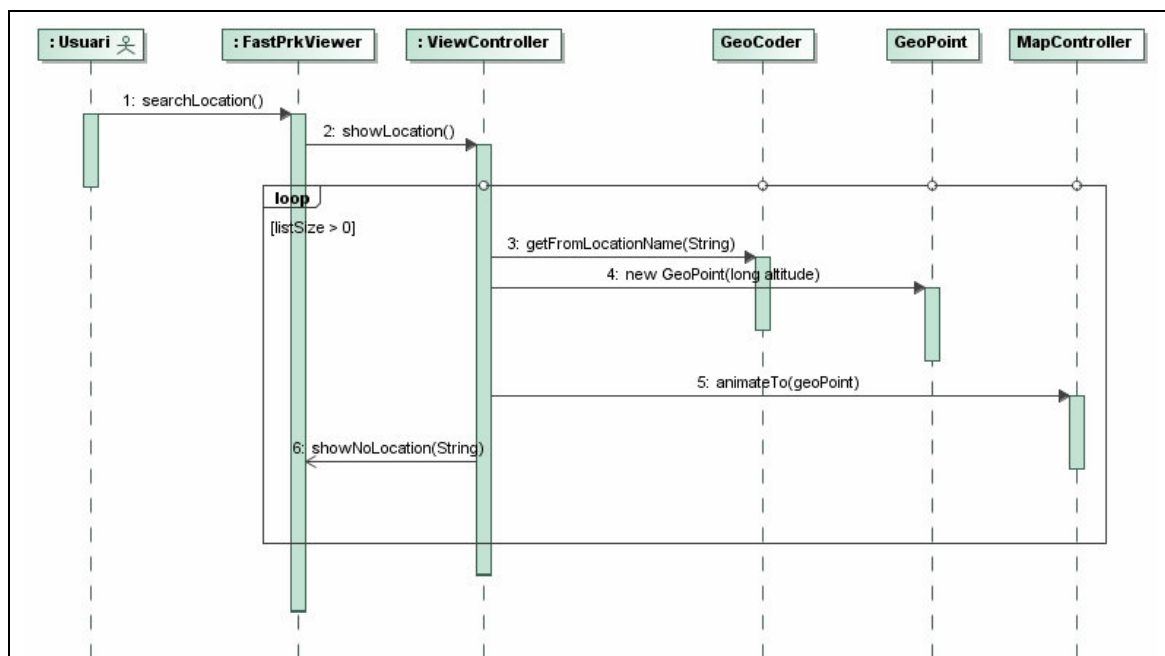
- Criterias: Classe original d'Android. S'utilitza per definir una sèrie de requisits per a l'elecció d'un proveïdor de localitzacions. Aquests requisits poden ser una major o menor precisió en la localització, consum de bateria, etc.

- LocationManager: Classe original d'Android utilitzada per manejar els serveis basats en localització. Aquí es pot definir quin proveïdor de localització es vol utilitzar com el GPS, xarxa, etc, o bé utilitzar la classe anterior esmentada, Criterias, per buscar el millor proveïdor de localització que s'adapti als requisits que se li han imposat.

- MyLocationListener: Aquesta classe estén de la interfície LocationListener d'Android, la qual, entre altres mètodes, obliga a implementar el mètode onLocationChanged (). Amb aquest mètode s'aconseguirà actualitzar la posició de l'usuari.

## 8.5. Diagrames de seqüència.

### 8.5.1. Acció "Search".



Il·lustració 18 Diagrama de seqüència "search"

Quan es defineix el botó Search en el fitxer xml de definició del layout, main.xml, s'estableix que el mètode associat a l'esdeveniment onClick<sup>3</sup> del botó serà searchLocation ().

Un cop dins del mètode searchLocation () es recull el contingut del quadre de text per poder realitzar la recerca. Per això es diu al mètode showLocation () i se li passa com a paràmetre la cadena de text obtinguda del quadre de text de la interfície gràfica.

Un cop dins de la classe ViewerController, utilitzant l'atribut Geocoder, es diu al seu mètode getFromLocationName (String, Integer), passant-li com a paràmetre la cadena de text del quadre de text i el nombre màxim de resultats que es desitja. Aquesta funció retorna una llista d'adreces.

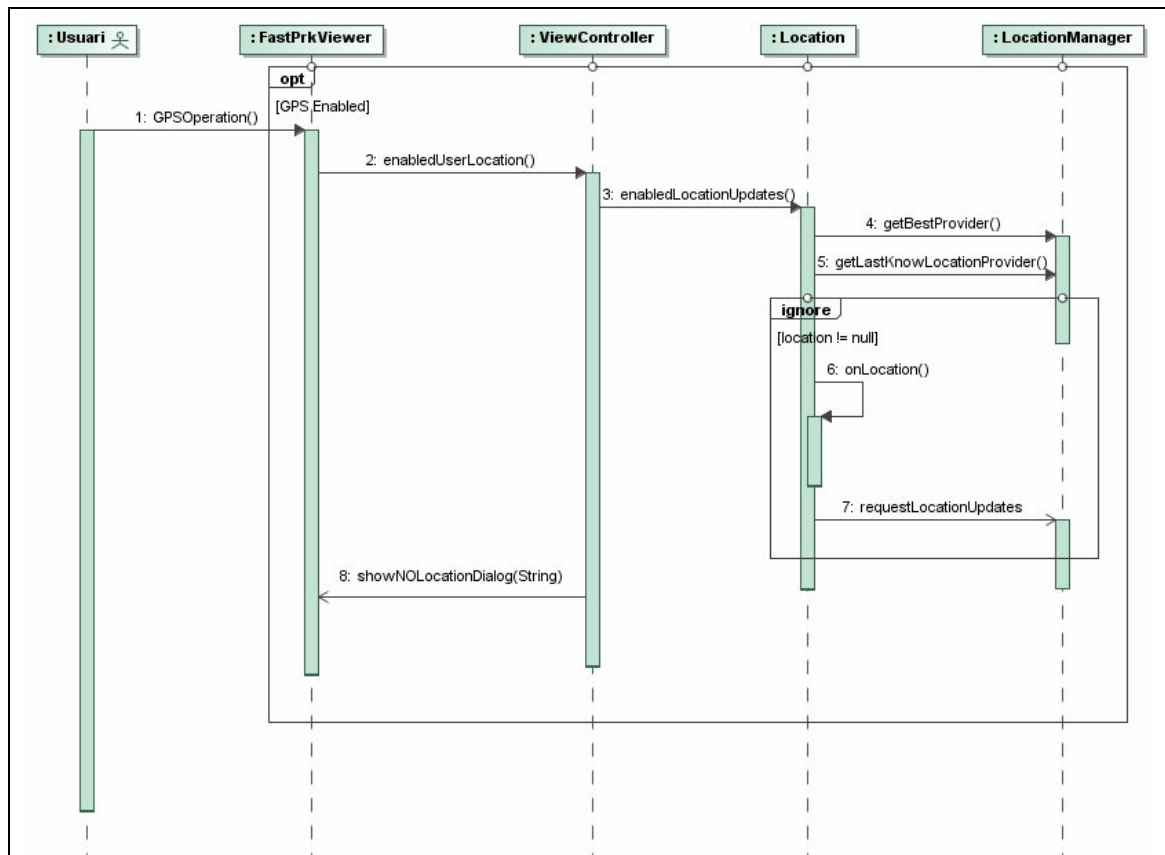
Si la mida de la llista d'adreces és superior a 0, es instancia un objecte de tipus Geopoint, passant-la longitud i latitud del primer element de la llista. Un cop obtingut el punt geogràfic se situa al mapa amb l'ajuda del mètode animateTo (GeoPoint) de l'MapController.

En cas que la mida de la llista que retorna el mètode getFromLocationName sigui 0, es procediria a cridar al mètode showNoLocationDialog de la classe FastPrkViewer, el qual mostrarà a l'usuari un missatge amb notificant la incidència ocorreguda.

### 8.5.2. Acció GPS On

<sup>3</sup> Esdeveniment generat quan es prem un botó, a aquest esdeveniment se li poden associar mètodes per executar diferents accions.





Il·lustració 19 Diagrama de seqüència "GPS ON"

L'acció de GPS On serà un subprocés en el cas que no estigui habilitat el GPS o l'usuari s'hagi oblidat, per tal d'obtenir la posició a on es troba el vehicle aparcats. S'iniciarà quan es prem el botó de GPS del mòbil quan aquest es troba desactivat, com que és una acció externa també l'he indicada per tindre-ho en compte, tal com es pot veure a la imatge:



Il·lustració 20 GPS OFF

El mètode associat a l'esdeveniment de premut del botó és `GPSOperation()`, definit en el fitxer xml de definició del layout, `main.xml`.

Per saber si quan s'ha premut el botó ha passat d'estat Off a On es comprova el mètode booleà `isChecked()`, l'`ToggleButton`. Si aquest mètode ens retorna `True` es fa una crida al mètode `enableUserLocation()` de `ViewerController`.

Dins `enableUserLocation`, fent ús d'un atribut de tipus `Locator`, es diu al mètode `enableLocationUpdates()`. Aquest és un mètode booleà, que en cas de tornar un valor `False` s'executaria el mètode `showNoLocationDialog()` de `FastPrkViewer`, el qual li mostrarà a l'usuari que no s'ha pogut obtenir la seva localització.

A l'interior del mètode `enableLocationUpdates ()` es fa ús d'un atribut de tipus `LocationManager`.

Amb aquest atribut es fa una crida al mètode `getBestProvider (Criteria)`. Aquest mètode retorna una cadena de text amb el nom del millor proveïdor de localitzacions trobat al mòbil. Perquè aquest mètode pugui saber quin proveïdor és millor que un altre cal passar-li com a paràmetre un objecte de la classe `Criteria`. Aquest objecte, definit com a atribut de la classe `Locator`, és parametrizat amb valors que defineixen l'ús de bateria de l'dipositivo, precisió desitjada del localitzador, etc.

Utilitzant la cadena de text amb el nom del proveïdor es diu al mètode `getLastKnownLocation ()` de la classe `LocationManager`. Es diu a aquest mètode per poder obtenir una localització més fiable. Aquest mètode retorna un objecte de tipus `Location`. Si aquest objecte resultés tenir un valor null, se surt del mètode donant `False` com a valor de retorn.

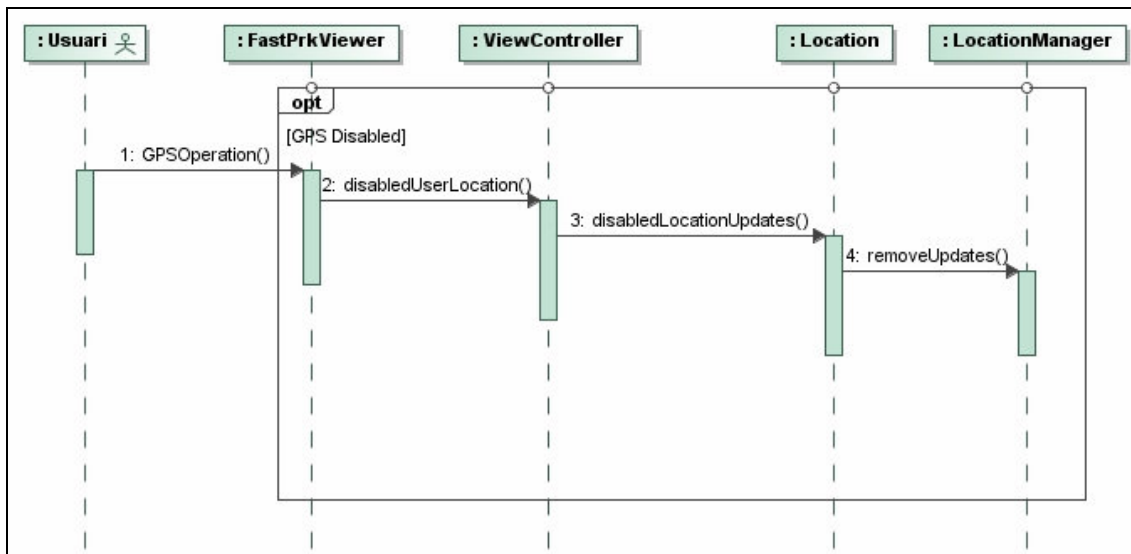
Si el valor d'`location` és diferent de null, es precedeix a convertir aquest valor d'un objecte `Location` a `Geopoint`, per tot seguit cridar al mètode privat `onLocationUpdated()`.

Aquest mètode fa una crida als listeners subscrits i executa el mètode `firePropertyChanged ()`, de la classe `PropertyChangeSupport`. Als listeners subscrits els arriba l'objecte de tipus `Geopoint`.

En aquest cas el listener subscrit és `ViewerController` el qual situarà el punt `Geopoint` sobre el mapa.

Un cop obtinguda la primera localització i enviada al mapa, es subscriu a `LocationManager` 1 listener perquè romangui a l'escolta dels canvis de localització detectats. La subscripció es realitza amb el mètode `requestLocationUpdates ()` de `LocationManager`. Amb cada actualització es llama al mètode privat esmentat anteriorment, `onLocationUpdated ()`.

### 8.5.3. Acció GPS Off



Il·lustració 21 Diagrama de seqüència "GPS OFF"

L'acció de GPS Off s'inicia quan es prem el botó de GPS del mòbil, com es veu a la imatge:



Il·lustració 22 GPS ON



Il·lustració 23 GPS ON amb activitat

El mètode associat a l'esdeveniment de premut del botó és `GPSOperation()`, definit en el fitxer xml de definició del layout, `main.xml`.

Per saber si quan s'ha premut el botó ha passat d'estat On a Off es comprova el mètode booleà `isChecked`, l'`ToggleButton`. Si aquest mètode ens retorna `False` es fa una crida al mètode `disableUserLocation()` de `ViewerController`.

Al interior d'aquest mètode es procedeix a cridar al mètode `disableLocationUpdates()` de la classe `Locator`.

Finalment s'executa el mètode `removeUpdates()` passant com a paràmetre l'atribut de la classe que fa de `Listener` de localització. Finalment amb aquest procés es fa que deixi d'actualitzar al mapa de l'aplicació dels canvis de localització que realitzi l'usuari.

## 9.- Implementació WhereIsMyCar?

### 9.1.- Compliments en la fase de disseny amb la realitat de la fase 3

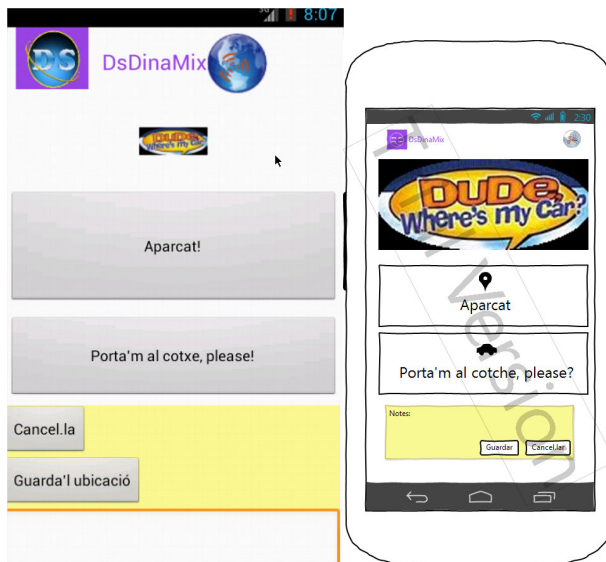
No tots els compliments de la fita 2 que comentava el disseny de com anava la implementació s'ha pogut confirmar ja que s'ha trobat amb diversos problemes com poden ser l'utilització de la classe `AsyncTask` per tal de fer tasques en segon pla o background, ja que pel moment es queda bloquejat i no controlo massa com recuperar les dades del GPS per tal de guardar-les de manera local en memòria.

A part s'ha tingut que compilar per l'API 17 Google, ja que hi ha instruccions que no son soportades per una API 14 Google pel simple motiu que han sortit posteriorment.

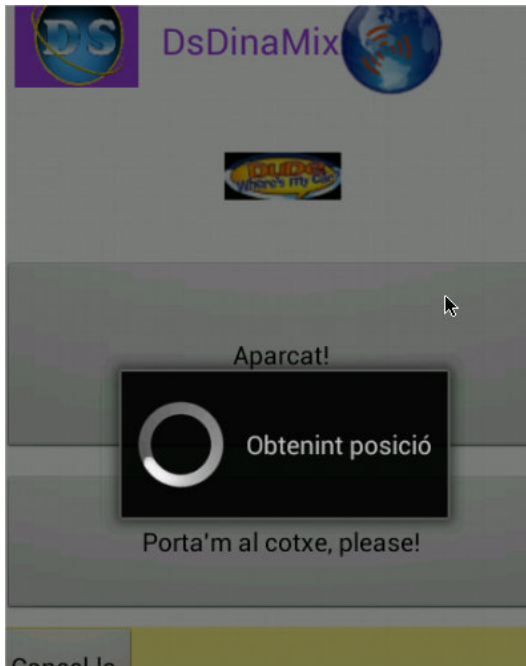
Per la següent línia dintre del manifest:

```
<activity android:name="com.google.ads.AdActivity" android:configChanges="keyboard | keyboardHidden | orientation | screenLayout | uiMode | screenSize | smallestScreenSize" />
```

Per una altra banda es té problemes amb els layouts per tal de mostrar la part:



Il·lustració 24 Screenshot app



Il·lustració 25 Screenshot app 1

## 9.2.- Recursos

A la web d'Android es pot obtenir el SDK per al desenvolupament d'aplicacions per a aquest Sistema Operatiu, el qual funcionarà amb l'IDE Eclipse. Amb aquest link

<http://developer.android.com/sdk/index.html>

La interfície de l'aplicació permetrà introduir dades, concretament de localitzacions geogràfiques, per realitzar-les. La forma en què s'obtindrà la informació serà a través del Google Maps. Per implementar aquestes accions serà necessari utilitzar l'API de Google Maps per Android on es poden trobar nombroses funcionalitats.

A la web de "Google Code" es pot veure tota la família de l'API de Google Maps. D'entre les opcions es farà ús dels serveis web per a Google Maps.

Aquests serveis web són un conjunt d'interfícies HTTP dels serveis de Google amb les que es poden obtenir dades geogràfiques per aplicacions amb mapes. A més s'haurà de mostrar per pantalla, al mapa, la localització del dispositiu mòbil on corre l'aplicació.

Per això s'utilitzarà el GPS del pròpi mòbil, si el té, en cas contrari haurà de realitzar l'acció mitjançant IP. Normalment per obtenir la ubicació geogràfica d'una adreça IP es consulta una base de dades d'informació de les adreces

assignades a un DNS passant-li la direcció que pretén buscar. D'aquesta base de dades s'obtindrà un DNS a resposta a la IP passada. Això derivarà en un o més adreces de llocs web provinents d'aquest DNS. Escanejant cadascuna de les adreces dels llocs web derivats es podria determinar la informació de l'adreça geogràfica.

D'altra banda l'API de Google Maps també disposa de mètodes que situen al mapa la localització en funció de la IP del dispositiu.

### 9.3.- Generació del projecte

A l'hora de crear el projecte s'utilitza l'IDE Eclipse amb l'SDK d'Android prèviament instal·lat. S'hauran de seguir els passos per crear un nou projecte amb l'Eclipse amb la diferència que s'haurà de seleccionar l'opció d'Android Project.

En el diàleg que es llança es defineix el nom del projecte, el seu domini, l'API que utilitzarà, l'activitat principal, etc. Com es va a fer ús de Google Maps s'ha d'incloure l'API de Google més actual, en aquest cas la versió 2.3.

Quan es crea un projecte Android, es genera automàticament una sèrie de fitxers, que els diferencien dels altres projectes Java. A continuació s'explica per a què serveixen aquests fitxers:


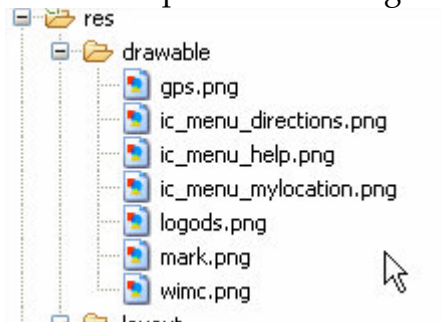
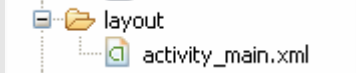


**Taula 2 Fitxers de configuració**

Fitxer	Utilitat
AndroiManifest.xml	<p>És un fitxer que conté la informació de l'aplicació que es mostrarà al sistema Android. S'edita per permetre que l'aplicació tingui accés a internet i que habilita les dues maneres de precisió a l'hora d'obtenir la localització per GPS o un altre mitjà.</p> <pre> &lt;uses-sdk     android:minSdkVersion="4"     android:targetSdkVersion="17"     android:maxSdkVersion="17" /&gt; &lt;uses-permission     android:name="android.permission.ACCESS_FINE_LOCATION" /&gt; &lt;uses-permission     android:name="android.permission.INTERNET" /&gt; &lt;uses-permission     android:name="android.permission.ACCESS_NETWORK_STATE" /&gt; &lt;uses-permission     android:name="android.permission.BLUETOOTH" /&gt; </pre>

	<pre>&lt;uses-permission android:name="android.permission.ACCESS_WIFI_STATE" /&gt;</pre> <p>A l'utilitzar el mínim del SDK 4 que equival a l'android 1.6 em de comentar la part ADMODS perquè el màxim API és la versió 17 d'android.</p>
activity_main.xml	<p>Fitxer que conté la definició de la interfície gràfica. A més de tots els elements que formen part de la interfície gràfica, cobra especial rellevància la declaració de la vista, pel banner del ADMods que en aquest cas l'he comentada pel funcionament de l'API 4.</p> <pre>&lt;!-- com.google.ads.AdView     android:id="@id/adView"     android:layout_width="wrap_content"     android:layout_height="wrap_content"     ads:adSize="BANNER"     ads:adUnitId="a14e5692f7ed5da"     ads:loadAdOnCreate="true"     ads:testDevices="TEST_EMULATOR, 859061051C3C461B16F7248C0DCDD8E1" /--&gt;</pre>
strings.xml	Fitxer que conté valors constants que seran utilitzats en l'aplicació.
colors.xml	Fitxer que conté els valors en RGB o HTML, per tal de facilitar el treball s'ha utilitzat un software free, "Just Color Picker"
dimens.xml	Les dimensions que tindrà els diferents components que es mostraran
ids.xml	Les identificacions del components
integers.xml	Constants que podem indicar, per exemple el total de línies que hi podran contenir el component EditText
main.xml	Fitxer que hi conté els diferents menú o possibilitats.
R.java	Fitxer que conté els identificadors dels elements continguts en activity_main.xml, main.xml i string.xml. És un fitxer que no ha de ser modificat per l'usuari, es modifica automàticament en funció dels canvis en els fitxers.

Un cop creat el projecte es pot observar la generació d'una sèrie de directoris, que seran explicats a continuació:

Taula 3 Directori dels fitxers de configuració

Fitxer	Utilitat
/gen	<p>Directori que conté el fitxer R.java.</p>  <p>Com es pot observar, al seu torn està contingut en el paquet <code>com.ds dinamix.whereismycar</code>. Coincideix amb el paquet on es troba l'activitat principal.</p>
/res/drawable	<p>És el directori on s'inclouen totes les imatges que són utilitzades per la interfície gràfica.</p> 
/res/layout	<p>És el directori que conté el fitxer <code>activity_main.xml</code>.</p> 
/res/values	<p>És el directori que conté el fitxer <code>strings.xml</code>, entre d'altres.</p> 
/res/menu	<p>És el directori que conté el fitxer dels menús, <code>main.xml</code>.</p> 

## 9.4.- Implementació de la interface gràfic

La interfície gràfica d'una aplicació en Android vindrà definit del tot en el fitxer `activity_main.xml` al directori `/res/layout` del projecte. Per tant, no es podrà mostrar res de l'aplicació que no s'inclouï en aquest fitxer.



El marc principal del interfície gràfica es realitza amb un `LinearLayout`. Amb això es van situant en posicions relatives dels altres del components de la interfície.

En primer lloc s'inclou els components de la part superior, això són:

- Unes imatges, logotip propi i el mateix logotip en forma de text.
- Un botó general: és el botó `Aparca` de la classe `Button`. El mètode associat a l'acció de prémer el botó es defineix en el fitxer `activity_main.xml`. És el botó encarregat de fer l'aparcament del vehicle una vegada que el cotxe ja està aparcat i per guardar la seva localització.
- Un altre botó per localitzar i portar-nos fins el vehicle, que en aquest cas quan tinguem la posició exacta la podem veure-la al google maps.
- Després tindrem un altre `LinearLayout` per tal de tenir una caixa de text (`EditText`) perquè l'usuari pugui apuntar qualsevol nota y guardar-la . En aquest cas no s'ha pogut finalitzar per falta de temps, però es deixa per altres versions.

Dintre d'aquest `LinearLayout` podem veure un parell de botons per guardar la informació que escriu l'usuari o cancel·lar-ho, es tenia la intenció de que aquestes notes es guardessin en una petita base de dades que en Android es faria en `SQLITE`.

La major part de valors constants, configuració, imatges i interfície gràfica d'usuari vindran predefinides en els fitxers XML que acompanyen l'aplicació.

Un d'aquests fitxers és el `R.java` que és el fitxer que conté l'identificador per a tots els valors de constants utilitzats en l'aplicació.

## 9.5.- API's utilitzades

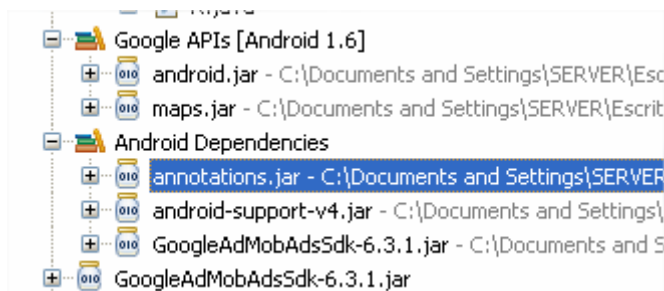
Per al desenvolupament d'aquesta aplicació s'han utilitzat les següents API s de programació:

- `Android.jar`: És l'API de Google per Android 1.6 que seria la mínima i Android 4.2.2 per a tablets i per tal d'utilitzar els `Admods`. Conté les llibreries bàsiques per al desenvolupament d'aplicacions Android. Coincideix pràcticament en la seva totalitat amb les llibreries Java habituals.
- `Maps.jar`: És un altre API de Google per Android 1.6 que seria la mínima i Android 4.2.2 per a tablets i per tal d'utilitzar els `Admods`. Conté les llibreries

necessàries per implementar les funcionalitats de mapes. Algunes de les classes que pertanyen a aquesta llibreria són la MapView o Overlay.

Després tindriem les dependències :

- Anotations.jar
- Android-suport-v4.jar
- GoogleAdMobAdsSdk-6.3.1.jar



Il·lustració 26 Llibreries propies i externes

## 9.6.- Incidències en el desenvolupament.

A la fase final del desenvolupament d'aquesta aplicació s'han trobat una sèrie d'incidències o problemes de com implementar la part de guardar la ubicació del cotxe que no ho tenia massa clar de com fer-la, i per tant ho vaig implementar amb ajuda d'altres webs com els que s'indiquen en l'apartat 12.

També s'ha tingut problemes i que no s'han pogut solucionar per posar els components a on es volia, tal com els layouts, imatges, botons, etc. No s'ha pogut acabar la implementació de guardar les notes.

## 10.- Pressupost

Aquest és un projecte de desenvolupament programari en el qual s'utilitzen eines de programari lliure i s'utilitzi codi llicenciat sota alguna llicència lliure pel que no tindrà cap cost econòmic lligat a llicències d'ús.

Malgrat això cal fer un càlcul del cost del desenvolupament, el qual inclou:

- Cost total de maquinari i programari.
- Manteniment.
- Suport Tècnic.

El maquinari a utilitzar serà un PC estàndard, per al desenvolupament de l'aplicació. Les primeres proves seran realitzades sobre el simulador que proporciona el Kit d' Desenvolupament Software d'Android. Posteriorment caldria provar aquesta aplicació amb un dispositiu mòbil el Sistema Operatiu sigui Android per provar les funcionalitats d'actualització en temps real de les localitzacions dels sensors a mesura que l'usuari es vagi desplaçant. El preu del PC es calcula al voltant de 800 € i el mòbil 250 €.

Prenent en consideració que el projecte el dugui a terme un enginyer especialista en desenvolupament programari per a aplicacions mòbils amb un sou mensual de 2.000 €, treballant 20 dies al mes, 40 hores per setmana, el preu de les fases serien:

- La fase de Planificació són de 15 dies pel que el cost seria de 750€.
- La fase d'Adaptació i configuració de l'entorn de desenvolupament són de 10 dies pel que el cost seria de 600 €.
- La fase de Disseny Funcional completa són 22 dies pel que el cost seria de 2.200 €.
- La fase Desenvolupament són 42 dies però la subtasca de la fase anterior Disseny Funcional, Revisió i correcció, i l'Estudi de Viabilitat, de la fase actual, coincideixen en el calendari. Com que són complementàries es realitzaran simultàniament de manera que la fase de Desenvolupament només se li consideraria els 35 dies per a l'escriptura del codi. Aquest cost seria de 2.500 €.
- Finalment la fase de Proves són 7 dies laborals pel que el cost seria de 700 €.

Resumint el pressupost seria de:

#### Taula 4 Taula pressupost

Descripció	Temps (dies)	Preu (€)
PC / Portàtil	--	500
Telèfon mòbil amb Android 1.6 / Tablet Android 4	--	450
Planificació	15	750
Adaptació i configuració del SDK en el PC / Portàtil	10	600
Disseny funcional	22	2.200
Desenvolupament	35	3.500
Proves	7	700
	<b>TOTAL</b>	<b>8.700</b>

## 11.- Conclusions

### 11.1.- Sistema

Amb l'elaboració i implantació del software he obtingut diversos coneixements de com cal crear una aplicació per a mòbils Android. No obstant, crec que s'ha donat un gran pas en l'avanç del benestar dels ciutadans per tal d'obtindre la posició GPS a on es troba el seu vehicle aparcat ja que potser que s'oblidi i no es sapigui a on s'havia deixat. També es pot observar el següent:

- Introduir encara més a les persones en la societat de la informació. Avui en dia, amb els avenços tecnològics, les persones poden obtenir informació sobre gairebé qualsevol cosa de manera ràpida i senzilla. Oferint informació sobre la localització, s'està aconseguint anar un pas més enllà.

Hi ha d'altres possibilitats pel producte que s'ha desenvolupat com per exemple els següents:

- Obtindre en cada moment la seva posició i posant algun tipus de dispositiu dintre del nostre vehicle per veure'l que esta en el mateix lloc que s'havia aparcat.
- Indicant qualsevol nota que l'usuari vulgues afegir com carrer, etc.

### 11.2.- Software

Amb el desenvolupament d'aquest programari s'han obtingut les següents observacions:

- L'arquitectura d'una aplicació per Android difereix de l'arquitectura d'una aplicació per a PC. Malgrat que el llenguatge de programació sigui un llenguatge tan estès com Java, el framework de l'aplicació és diferent. El framework més utilitzat per una aplicació d'ordinador es componen dels següents elements:

- **Control:** és l'element que controla l'accés a l'aplicació.
- **Model:** és un membre del controlador que maneja les operacions lògiques i d'ús de la informació.
- **Vista:** és l'element que s'encarrega de dibuixar la interfície gràfica d'usuari.

Una aplicació per Android es compon dels següents elements:

- **Activity**: les activitats representen el component principal de l'interfície gràfic. Es pot veure una activitat com el component anàleg a una finestra en qualsevol altre llenguatge visual.

- **Services**: els serveis són components sense interfície gràfica que es s'executen en segon pla. Conceptualment són similars als serveis en qualsevol altra arquitectura.

- **Content Provider**: és el mecanisme definit en Android per compartir dades entre aplicacions.

- **Broadcast Receiver**: és el component destinat a detectar i reaccionar enfront de determinats missatges o esdeveniments globals generats pel sistema.

Com es pot observar alguns elements de l'arquitectura per Android, conceptualment són molt semblants als que s'utilitzarien per a una aplicació per a PC encara que es veu clarament que l'arquitectura d'una aplicació per Android està adaptada a les limitacions maquinari que té un telèfon mòbil:

- Els elements, citats en el punt anterior, que componen l'aplicació han estar declarats en el AndroidManifest. El AndroidManifest és un fitxer de tipus xml que conté la informació de l'aplicació que es mostrarà al sistema Android. Si no es troba declarat en aquest fitxer no es pot crear de forma dinàmica en l'aplicació.

- Igual que els components han d'estar al AndroidManifest, els elements que componen la interfície gràfica han d'estar declarats en el fitxer /Layout/activity\_main.xml del projecte de l'aplicació. Les instàncies realitzades, d'elements visuals de l'aplicació, s'han de fer en referència al declarat en aquest fitxer

- L'emulador d'Android per a PC inclòs en el SDK d'Android per Eclipse és una bona eina de treball, útil per testear i depurar l'aplicació però hi ha determinats elements que formen part de l'aplicació que no poden ser provats en l'emulador. Un exemple és el NETWORK\_PROVIDER, és dir, el proveïdor de localització a través de la xarxa. Per provar les obtenció de localització del dispositiu mòbil a través de la xarxa cal fer les proves sobre un mòbil real amb Android.

En l'aspecte de les limitacions maquinari, desenvolupar una aplicació per Androi pot recordar al desenvolupament d'una aplicació per a microprocessadors encastats o microcontroladors.

La diferència està que els microcontroladors tenen aplicacions que generalment són processos i serveis que obeeixen a paradigmes de programació més habituals de els llenguatges estructurats. Una aplicació per Android malgrat que també pot utilitzar serveis està basat en programació orientada a objectes

anant més enllà i adaptant-se a l'arquitectura d'un telèfon mòbil, que encara que avanci a passos de gegant la seva tecnologia, poc té a comú amb l'arquitectura d'un PC.

### 11.3.- Aspectes generals

Tot i la dificultat inicial que un desenvolupador pot trobar en realitzar per primera vegada una aplicació per Android, sí que és cert que es trobarà amb un sistema dissenyat per facilitar la feina dels desenvolupadors.

També es pot observar que ofereix un alt nivell d'abstracció enfront del maquinari qual cosa facilita enormement el desenvolupament i integració entre aplicacions. Son moltes les facilitats que un desenvolupador es pot trobar a l'hora d'elaborar una aplicació ja que hi ha molt codi font, biblioteques , referències, tutorials i programari de simulació.

A través del web oficial d'Android, a més de trobar-se amb les últimes novedades del sistema operatiu, versions, etc, també es trobarà amb una guia de presentació i desenvolupament d'aplicacions que esmicola la major part dels secrets del desenvolupament d'aplicacions. D'altra banda també es pot tenir accés a pràcticament tota la biblioteca bàsica de classes de Java per Android.

Malgrat tanta facilitat és altament recomanable que el desenvolupador tingui al seu disposició un telèfon mòbil amb Android perquè les proves siguin el més fiables possibles i així poder provar tots els perifèrics de què disposa el mòbil, observar la eficiència de l'aplicació en el que a consum de recursos es refereix (CPU, bateria, etc) i com no fer les proves finals de qualsevol aplicació sobre el medi al qual està destinat.

## 12.- Referències bibliogràfiques

Bibliografía utilitzada:

- <http://www.google.com>
- <http://code.google.com/intl/es-ES/apis/maps/>
- <http://developer.android.com/index.html>
- “Manual Programación Android [sgoliver.net] v2.0.” de Salvador Gómez Oliver, Novembre de 2011.