



**Universitat Oberta
de Catalunya**



SaveMyLocation

Memòria del projecte
d'Enginyeria Informàtica

realitzat per

Oriol Gabau Cortada

i dirigit per

Jordi Ceballos Villach

Universitat Oberta de Catalunya

Juny de 2014

En agraïment a la meva família i al meu entorn més proper
per estar en tot moment sempre al meu costat...

Índex de continguts

1	Introducció.....	8
1.1	Les virtuts de saber la posició.....	8
2	Estudi de viabilitat.....	9
2.1	Objectius.....	9
2.2	Història del projecte.....	9
2.3	Objecte.....	9
2.3.1	Descripció de la situació a tractar.....	9
2.3.2	Perfil del client/usuari.....	9
2.3.3	Objectius.....	10
2.4	Descripció del sistema a realitzar.....	10
2.4.1	Descripció.....	10
2.4.2	Recursos.....	10
2.4.3	Avaluació de riscos.....	11
2.4.4	Organització del projecte.....	12
2.5	Planificació del projecte.....	13
2.5.1	Estimació temporal.....	13
2.5.2	Anàlisi de cost-benefici.....	14
2.5.3	Diagrama de Gantt.....	15
2.6	Conclusions.....	16
3	Anàlisi funcional i disseny tècnic.....	17
3.1	Requeriments funcionals.....	17
3.1.1	Funcionalitats bàsiques.....	17
3.1.2	Monousuari.....	17
3.1.3	Mitjà d'informació de l'últim estat guardat.....	17
3.1.4	Accés directe per widget.....	17
3.1.5	Quantificació de la precisió.....	17
3.1.6	Ubicació mínima acceptada.....	19
3.2	Requeriments no funcionals.....	19
3.2.1	Tolerància d'error.....	19
3.2.2	Accés a les funcionalitats de posicionament d'Android.....	19
3.3	Pseudorequeriments.....	20
3.3.1	Plataforma.....	20
3.3.2	Llenguatge d'implementació.....	20
3.4	Casos d'ús.....	21
4	Prototip.....	25
4.1	Menú principal.....	25
4.2	Administrar posicions.....	25
4.3	Modificar opcions.....	26
4.4	Sobre el programa.....	26
4.5	Widget.....	26
4.6	Desar posicions.....	27
4.7	Icona del programa.....	27
5	Disseny tècnic.....	28
5.1	Arquitectura Android.....	28
5.2	Emmagatzematge de les dades.....	29
5.3	Diagrama de classes.....	30
5.4	Diagrames de seqüència.....	32
5.4.1	Acció desar posició.....	32
5.4.2	Acció llistar posicions.....	33
6	Implementació i funcionament de l'aplicació.....	34

6.1 Classes en Java.....	34
6.1.1 Menú principal (MainActivity).....	35
6.1.2 Desar posició (GetPositionActivity).....	36
6.1.3 Llistar posicions (PositionListActivity – PositionsListDBAdapter).....	38
6.1.3.1 Inserir/Editar una posició (PositionEdit).....	39
6.1.4 Opcions (OptionsActivity).....	40
6.1.5 Widget (WidgetIntentReceiver – WidgetProvider).....	41
6.1.6 Altres funcions.....	42
6.1.6.1 Canvi de tipus de mapa.....	42
6.1.6.2 Sobre l'aplicació.....	42
6.1.6.3 Crear enllaç directe a l'escriptori.....	42
6.1.6.4 Sortir de l'aplicació.....	42
6.2 Layouts en XML.....	43
6.3 Recursos gràfics.....	43
6.4 Arxiu de cadenes de text.....	43
6.5 Arxiu de configuració (AndroidManifest.xml).....	44
6.6 Funcions del SDK d'Android aplicades en la programació.....	46
7 Conclusions.....	47
7.1 Assoliment d'objectius.....	47
7.2 Diferències del resultat amb la planificació.....	48
7.2.1 Posicionament sobre Google Maps.....	48
7.2.2 Versió mínima d'Android 2.3.3 en comptes de 2.2.....	48
7.2.3 Activitat per desar manualment la posició sense timeouts.....	48
7.2.4 Replantejament dels timeouts.....	49
7.2.5 Canvis en el model de base de dades i la informació mostrada.....	49
7.2.6 Altres funcionalitats addicionals afegides.....	49
7.3 Dificultats destacables en la programació.....	50
7.3.1 Aplicació del timeout i la lluita contra el ANR.....	50
7.3.2 Base de dades SQL i mostrar el llistat de posicions.....	50
7.4 Futures millores.....	51
8 Bibliografia.....	53

Índex d'imatges

Imatge 1: Representació de les versions d'Android en ús (1 d'abril 2014).....	20
Imatge 2: Prototip del menú principal.....	25
Imatge 3: Prototip del llistat de posicions emmagatzemades.....	25
Imatge 4: Prototip del menú d'opcions del programa.....	26
Imatge 5: Prototip del widget.....	26
Imatge 6: Prototip de l'activitat Desar Posició.....	27
Imatge 7: Iconografia recomanada per Google.....	27
Imatge 8: Arquitectura del sistema operatiu Android.....	28
Imatge 9: Diagrama de classes previst de SaveMyLocation.....	30
Imatge 10: Diagrama de seqüència de l'acció desar posició.....	32
Imatge 11: Diagrama de seqüències de l'acció llistar posicions.....	33
Imatge 12: Estructura d'arbre de les classes .java.....	34
Imatge 13: Vista del menú principal.....	35
Imatge 14: Mostra de la barra d'acció en versions d'Android inferiors a la 4.0.....	35
Imatge 15: Vista inicial de la funció desar posició.....	36
Imatge 16: Botó que simula un interruptor per iniciar el procés de posicionament.....	36

Imatge 17: Toast detectant l'absència d'algun provider.....	36
Imatge 18: Posicionament inicial sense dades.....	36
Imatge 19: Mostra d'exemple de les diferents vistes segons la precisió obtinguda.....	37
Imatge 20: Llistat de totes les posicions emmagatzemades i la possibilitat d'esborrar-ne qualsevol mantenint pitjat una posició en concret.....	38
Imatge 21: Menú contextual dins del llistat de posicions que permet inserir un nou registre.....	39
Imatge 22: Edició d'una posició de forma manual tant amb la introducció de coordenades com amb el desplaçament de la marca sobre el mapa.....	39
Imatge 23: Menú d'opcions per canviar els timeouts del widget.....	40
Imatge 24: Vista dels dos estats del widget. Superior: inicial - Inferior: actualitzat.....	41
Imatge 25: Mostra de les cinc possibles notificacions rebudes a la barra de notificacions d'Android durant l'execució del widget.....	41
Imatge 26: Mostra dels diferents tipus de visualització del mapa.....	42
Imatge 27: Notificació toast informant sobre l'aplicació.....	42
Imatge 28: Estructura d'arxius layout de SaveMyLocation.....	43
Imatge 29: Recursos gràfics utilitzats a l'aplicació.....	43

1 Introducció

1.1 Les virtuts de saber la posició

La posició és la situació física en l'espai d'una cosa en referència a una altra. Aquesta informació sempre ha sigut de gran utilitat en la història de la humanitat. El transport, la cartografia, les estratègies de combat, una cosa tant simple com anar d'un punt A a un punt B és impossible sense saber la posició d'un respecte l'altre.

L'exèrcit americà ràpidament va veure els avantatges estratègics d'una tecnologia capaç d'esbrinar la posició d'un objecte en qualsevol punt del planeta i els esforços van donar fruit al conegut sistema GPS (Global Positioning System). El seu funcionament no és gaire complexe: un conjunt de satèl·lits (24 en concret pel sistema americà) distribuïts al llarg de l'òrbita terrestre emeten contínuament un senyal amb l'identificador, l'hora dels seus rellotges atòmics i la seva posició. Amb aquesta informació, el receptor terrestre pot esbrinar la distància al satèl·lit en base al temps de propagació i calcular la seva posició per mitjà de triangulació matemàtica. La precisió del sistema oficialment està xifrada en un radi de 15 metres però la tecnologia actual permet estar per sota dels 5 metres.

Al segle XXI la tecnologia de processament i la precisió dels rellotges digitals ha fet possible tenir al palmell de la mà un dispositiu amb tecnologia GPS integrada. Les possibilitats eren limitades però la incorporació d'aquestes característiques en un dispositiu mòbil, programable i obert a la comunitat posa a l'abast de la imaginació qualsevol aplicació possible. Tanmateix, un altre factor clau és la comunicació. Els anomenats smartphones inclouen en la seva totalitat la possibilitat de connexió a internet permanent i mòbil. Per tant, la unió fa la força i avui dia un gran nombre de persones tenen la possibilitat al palmell de la mà tant per saber on són com per comunicar-ho a qualsevol altra.

2 Estudi de viabilitat

2.1 Objectius

El present projecte té per objectiu la realització d'una aplicació per a la plataforma mòbil Android que permet emmagatzemar la posició actual de la forma més ràpida, senzilla i desatesa possible. Inclourà a més a més accions sobre aquesta posició i les emmagatzemades anteriorment.

2.2 Història del projecte

La motivació darrera d'un projecte d'aquestes característiques ve principalment per la inexistència d'aplicacions alternatives amb les característiques desitjades. Tot el ventall existent posen èmfasi en el posicionament amb una interfície sobre un mapa però cap dona prioritat a la rapidesa i la desatenció de la petició.

2.3 Objecte

2.3.1 Descripció de la situació a tractar

Localització del dispositiu mòbil amb la màxima precisió i amb els mínims recursos consumits, ja sigui per temps i per consum de bateria.

2.3.2 Perfil del client/usuari

L'aplicació va dirigida a qualsevol usuari amb o sense coneixements informàtics que tingui interès per memoritzar en el seu terminal mòbil de tipus Android la seva posició actual així com poder accedir i localitzar posicions prèviament emmagatzemades.

2.3.3 Objectius

Els principals propòsits de l'aplicació són:

- Emmagatzematge de la posició actual
- Rapidesa en l'obtenció de la posició
- Prioritat pel funcionament de forma desatesa
- Agilització en l'accés mitjançant un *widget*
- Permetre el tractament dels punts de localització realitzats
- Visualització dels punts en un mapa
- Accés a funcions de navegació cap a la posició desitjada

2.4 Descripció del sistema a realitzar

2.4.1 Descripció

Aplicació mòbil nativa¹ pel sistema operatiu Android

2.4.2 Recursos

La realització d'aquest projecte bàsicament té per requisits els especificats en l'eina de desenvolupament SDK d'Android:

- **Hardware**
 - Maquinari per a la programació i la redacció de la memòria de projecte:
 - Processador x86: >2Ghz
 - Memòria RAM: 2Gb
 - Disc-dur: 8Gb
 - VGA
 - Monitor
 - Connexió a Internet
 - Terminal mòbil per provar l'aplicació: (opcional)
HTC Desire[1]
- **Software**
 - Sistema operatiu: Linux
 - Navegador web: Firefox, Opera, Chrome, ...
 - Programa de planificació: Planner[2]
 - Programa de disseny UML: Violet[3] i ArgoUML[4]

¹ Dissenyada i programada per a una plataforma única fent ús de les seves capacitats exclusives.

- Suite ofimàtica: LibreOffice.org
- Software d'edició d'imatges: Gimp[5]
- Entorn de programació SDK: Android ADT Bundle[6]

- **Personal**

- Usuari experimentat en la redacció d'informes i amb coneixements de programació en llenguatge Java.

2.4.3 **Avaluació de riscos**

- **Inexperiència en la planificació**

- **Descripció**

Cal tenir molt en compte la pròpia inexperiència en la realització de projectes d'aquesta importància en un espai de temps tant gran i per tant, la planificació temporal i de costos podria patir greus diferències respecte la realitat.

- **Probabilitat**

Mitjana

- **Impacte**

Crític

- **Solució**

Una desviació en les previsions inicials poden ser rectificades amb un augment del temps de dedicació i una millora de la comunicació amb el tutor. Existeix un període de dues setmanes entre la data prevista i la data d'entrega que pot ser utilitzat per aquests imprevistos.

- **Inexperiència en programació Java**

- **Descripció**

Degut als meus coneixements limitats en Java per la inexperiència en aplicacions de certa magnitud existeix un risc de trobar obstacles que requereixen coneixements addicionals. No obstant, aquest projecte no sembla incloure funcions complexes que posin en perill l'execució d'aquest.

- **Probabilitat**

Mitjana

- **Impacte**

Mitjà

- **Solució**

En fase de planificació també es realitzarà una recopilació d'informació en forma de tutorials i llibres sobre programació Java que pot servir per solucionar les dificultats trobades en programació.

- **Inexperiència en programació per a mòbils**
 - **Descripció**

Es partirà de zero en la programació per a mòbils, concretament per a Android per la qual mai he programat per aquesta plataforma.
 - **Probabilitat**

Mitjana
 - **Impacte**

Mitjà
 - **Solució**

En fase de planificació també es realitzarà una recopilació d'informació en forma de tutorials i llibres sobre programació en Android que pot servir per solucionar les dificultats trobades en programació.

2.4.4 Organització del projecte

L'eina de treball bàsica per l'organització i planificació del projecte serà el programari Planner doncs té infinitat de possibilitats, així com l'optimització de recursos i el seguiment actiu de l'evolució del projecte. A més a més, és multi-plataforma i de codi obert sota llicència GNU.

2.5 Planificació del projecte

2.5.1 Estimació temporal

Caldrà tenir en compte els punts comentats en l'avaluació de riscos per evitar desajustos de temps inesperats.

Tanmateix, la planificació temporal del projecte és la següent:

Nº	Descripció de l'activitat	Duració (hores)
1	Estudi de viabilitat	15
2	Documentació prèvia	15
3	Redacció i entrega de la PAC1	5
4	Instal·lació de programari desenvolupament	2
5	Anàlisi funcional plataforma mòbil	10
6	Disseny funcional	10
7	Implementació de la plataforma mòbil	35
8	Proves i depuració	10
9	Preparació de la memòria	10
10	Preparació de la presentació	3
11	Revisió, tancament i entrega final	2
Nombre total d'hores		117

2.5.2 Anàlisi de cost-benefici

Tenint en compte que el redactor no rebrà cap compensació econòmica donat que es tracta del seu projecte de final de carrera, els costos derivats de la realització d'aquest estudi són:

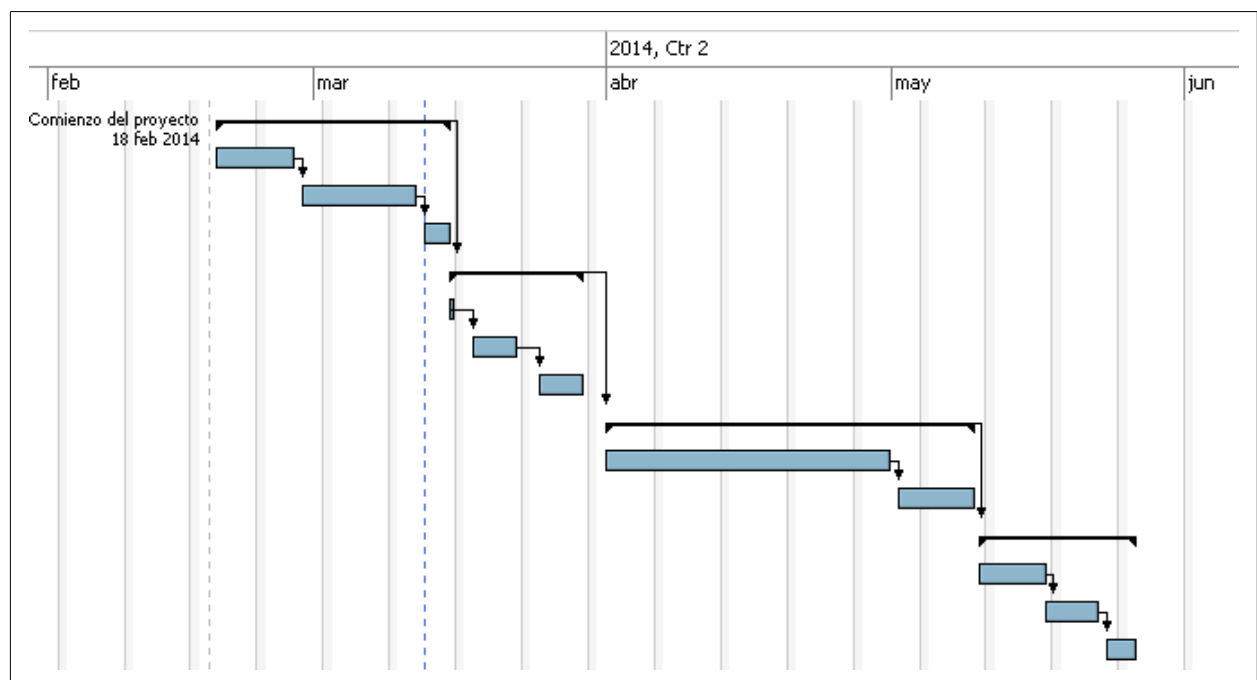
Taula de costos		Cost (Euros)
Hardware		
	Ordinador bàsic	250 €
	Connexió Internet	20 €/mes
	HTC Desire (Segona mà)	30 €
Software		
	Sistema operatiu: Linux	0 €
	Planificador: Planner	0 €
	Suite ofimàtica: LibreOffice.org	0 €
	Navegador Firefox	0 €
	Modelatge UML: ArgoUML i Violet	0 €
	SDK Android	0 €
	Editor d'imatges: GIMP	0 €
Personal		
	Redactor/Escriptor	0 €
	Cost total	280 € + 20 €/mes

S'ha procurat obtenir la major reducció de costos amb l'ús de programari lliure i gratuït. A més a més, l'ús de programari multi-plataforma garanteix una major mobilitat i independència del sistema operatiu escollit.

2.5.3 Diagrama de Gantt

Introduïdes les dades al Planner i en funció d'una jornada laboral ajustada als horaris del redactor/escriptor quedarà resumit en el següent esquema amb una finalització estimada pel 26 de maig del 2014.

WBS	Nom	Inici	Fi	Duració
1	PAC1 Pla de treball	Feb 18	Mar 15	3d 6h
1.1	Estudi de viabilitat	Feb 18	Feb 26	1d 2h
1.2	Documentació prèvia	Feb 27	Mar 11	1d 7h
1.3	Redacció i entrega PAC1	Mar 12	Mar 15	5h
2	PAC2 Anàlisi funcional i disseny tècnic	Mar 15	Mar 29	2d 1h
2.1	Instal·lació de programari desenvolupament	Mar 15	Mar 15	1h
2.2	Anàlisi funcional plataforma mòbil	Mar 17	Mar 22	1d
2.3	Disseny funcional	Mar 24	Mar 29	1d
3	PAC3 Implementació	Mar 31	Mai 9	5d 5h
3.1	Implementació plataforma mòbil	Mar 31	Abr 30	4d 3h
3.2	Proves i depuració	Mai 1	Mai 9	1d 2h
4	PAC4 Lliurament final	Mai 10	Mai 26	2d 4h
4.1	Preparació de la memòria	Mai 10	Mai 26	1d 2h
4.2	Preparació de la presentació	Mai 10	Mai 17	5h
4.3	Revisió, tancament i entrega final	Mai 23	Mai 26	5h



2.6 Conclusions

La realització d'aquest projecte no té per objectiu cap finalitat econòmica. Es tracta doncs d'una aportació d'interès comú pel bé de la comunitat i l'enriquiment de coneixements propis en una tecnologia d'actualitat.

Per tant, el projecte es considera **viable** tenint en compte les condicions de realització i els objectius establerts.

3 Anàlisi funcional i disseny tècnic

3.1 Requeriments funcionals

Aspectes específics de l'aplicació que tindrà lloc en aquest projecte.

3.1.1 Funcionalitats bàsiques

L'aplicació contindrà les següents funcionalitats

- a) Obtenir i desar posició actual
- b) Modificar les posicions desades
- c) Opcions d'usuari

3.1.2 Monousuari

Degut a les particularitats dels dispositius mòbils i del sistema Android, l'aplicació està dissenyada en l'ús per part d'un únic usuari. Queda en mans del sistema operatiu la funcionalitat per a múltiples usuaris com es pot donar en el cas d'Android per a tauletes amb una versió superior a la 4.2. L'aplicació per tant no conté cap mecanisme de perfils d'usuari.

3.1.3 Mitjà d'informació de l'últim estat guardat

L'usuari ha de rebre en tot moment la informació sobre el procés de localització i emmagatzematge realitzat per l'aplicació. En tot cas serà la barra de notificacions nativa d'Android l'encarregada de mostrar aquesta informació per tal que l'usuari tingui la llibertat de canviar d'aplicació mentre aquest està realitzant la seva funció.

3.1.4 Accés directe per widget

La principal utilitat de l'aplicació és l'emmagatzematge de la posició actual amb la màxima precisió i rapidesa. Per aquest motiu, l'aplicació disposarà d'un *widget* d'accés directe que l'usuari pot col·locar a fons d'escriptori del seu sistema Android. Un cop pitjat aquest botó, el sistema realitzarà el procés normal de localització però sense mostrar cap pantalla de l'aplicació, tan sols serà informat del procés en la barra de notificacions.

3.1.5 Quantificació de la precisió

Qualsevol dispositiu Android situat en un escenari real sol tenir al seu abast un conjunt d'indicadors que li serveixen per localitzar la posició del terminal. Cada punt d'informació representa de forma implícita una precisió que serà considerada en aquesta aplicació de major a menor de la següent forma:

- Precisió alta:

Estat només assolit per la funció GPS la qual és auto-conscient d'estar correctament posicionat i representa una precisió en un marge d'error inferior als 25 metres. Assolit

aquest punt, es considerarà el posicionament com a òptim i ha de donar per finalitzat el procés.

- Precisió mitja:

El sistema ha localitzat senyals sense fils tipus wifi sota el protocol 802.11. Aquestes senyals no inclouen el seu posicionament però Android automàticament transmet les adreces MAC d'aquests emissors als servidors de Google que responen amb la seva posició aproximada. Utilitzant triangulació de posicions i potències de senyal es pot obtenir una posició aproximada amb poca potència de càlcul. La precisió d'aquest estat es pot considerar per sota del centenar de metres donat que per norma general provenen dels encaminadors domèstics amb unes antenes poc potents. Google manté aquesta base de dades actualitzada mitjançant els seus vehicles especialitzats recorrent tots els carrers i també per part dels propis usuaris que utilitzen Google Maps i accepten la cessió d'aquesta informació.

- Precisió baixa:

Les antenes de telefonia també poden actuar com a localitzadors. Quan un telèfon mòbil sincronitza amb una antena del seu operador rep un nombre anomenat CellID que identifica inequívocament una antena en concret. Google disposa del posicionament d'aquestes antenes i amb un sistema semblant de triangulació als punts wifi, dóna resposta a les peticions de posicionament amb aquest identificador. La precisió d'aquest sistema és molt variable segons la localització, el terreny i la potència subministrada per l'antena però es pot considerar per sota del quilòmetre a la rodona.

- Precisió nul·la:

Existeixen casos on l'aplicació serà absolutament incapaç d'obtenir la posició.

- Opció de posicionament desactivada:

Android per qüestions de privacitat ofereix a l'usuari la possibilitat d'anul·lar qualsevol tipus de posicionament tant per les funcions de sistema com de les aplicacions de tercers.

- Absència de senyal de telefonia mòbil:

No tots els dispositius Android disposen de funcions de telefonia com pot ser l'execució en una tauleta o un emulador. Per tant, en aquests dispositius no serà possible obtenir aquest estat de precisió.

- Absència de senyals wifi:

Pot ser per limitacions de maquinari o per estar situat en un lloc sense cap senyal wifi propera.

- Absència de senyals GPS:

La funció de GPS és capaç de determinar en quin moment té el seu rellotge intern sincronitzat amb prou precisió com per ser considerat una localització correcte GPS. En cas de no trobar suficients satèl·lits o de no entrar en l'estat de localitzat, es considerarà com invàlida la posició GPS.

3.1.6 Ubicació mínima acceptada

Tenint en compte que el dispositiu mòbil requereix un cert temps en esbrinar la seva posició i que l'usuari és possible que no estigui en una posició estàtica, cal donar resposta a dues preguntes importants:

- Quant temps cal esperar per emmagatzemar una posició donant per suposat que no s'aconseguirà cap millor més endavant?
- Si no es té garanties de tenir cap posició òptima, quina de les possibles posicions anteriors cal donar per vàlida?

En el primer cas, cal considerar que el sistema tindrà coneixement de sobre quins mitjans té al seu abast per localitzar la seva posició donant lloc a una precisió baixa, mitja o alta. Per tant, s'estableix els següents períodes entre estats els quals, si no hi ha canvis, suposaran la posició actual obtinguda com la millor.

- Estat baix: 10 segons
- Estat mig: 10 segons
- Estat alt: 10 segons

Aquesta configuració per defecte estableix el temps màxim total en guardar la posició de 30 segons. No obstant, pot ser modificada en les opcions d'aplicació.

3.2 Requeriments no funcionals

Aspectes que tindrà el sistema però no són específics d'aquest projecte.

3.2.1 Tolerància d'error

El sistema realitza diverses funcions que van reduït el marge d'error donant cada cop una posició més pròxima a la realitat. Aquestes funcions augmenten la precisió com també requereixen més temps però malgrat tot existeix un marge d'error mínim que varia segons la quantitat de satèl·lits rebuts i les limitacions pròpies del xip encarregat d'interpretar els senyals provinents d'aquests.

Per tant, cal tenir en compte que el sistema oferirà com a màxim un posicionament amb un marge d'error d'entre un i deu metres del punt exacte.

3.2.2 Accés a les funcionalitats de posicionament d'Android

Aquesta aplicació requereix esbrinar la posició del dispositiu en un moment donat. Tal funcionalitat ve donada dins les pròpies llibreries que ofereix Android i per tant se'n farà ús. Ubicar-se implica un augment en el consum de bateria degut al processament continu dels senyals rebuts. No obstant, l'aplicació sols en farà ús en el moment que l'usuari desitgi obtenir la posició i serà sempre sobre el mínim temps possible.

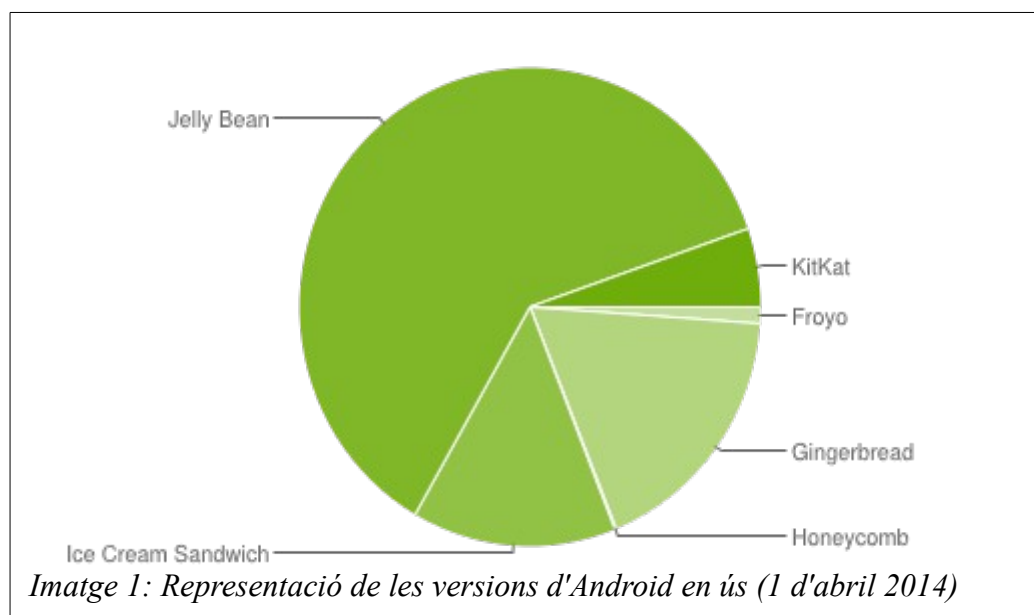
3.3 Pseudorequeriments

3.3.1 Plataforma

La plataforma sobre la qual s'executarà l'aplicació requerirà una versió mínima d'Android 2.2 (Froyo) la qual garanteix una compatibilitat amb totes les versions superiors. La distribució de versions actualment segons les estadístiques de Google: [7]

Versió	Nom	API	Distribució
2.2	Froyo	8	1.1%
2.3.3 – 2.3.7	Gingerbread	10	17.9%
3.2	Honeycomb	13	0.1%
4.0.3 – 4.0.4	Ice Cream Sandwich	15	14.3%
4.1.x	Jelly Bean	16	34.4%
4.2.x	Jelly Bean	17	18.1%
4.3	Jelly Bean	18	8.9%
4.4	KitKat	19	5.3%

Dades recollides durant un període de 7 dies per la Google Play Store a data 1 d'abril 2014



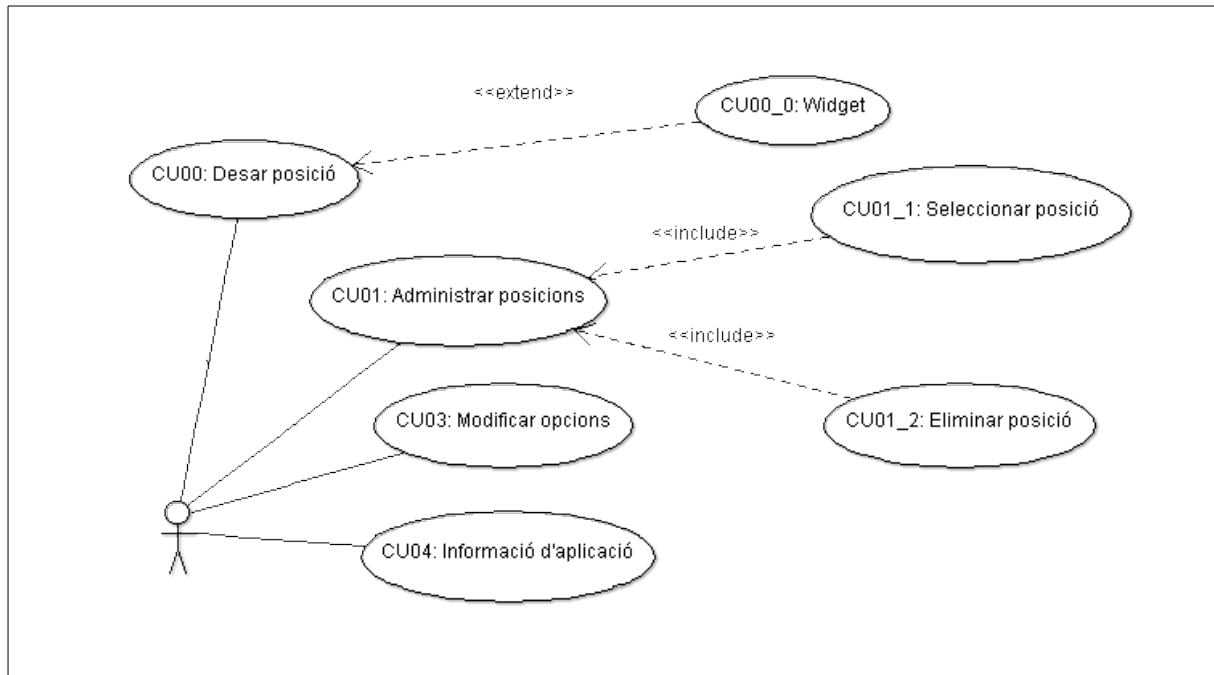
Aquest requeriment cal establir-ho en iniciar la programació i seguint les estadístiques pretén donar resposta a la pràctica totalitat de dispositius Android.

3.3.2 Llenguatge d'implementació

El llenguatge utilitzat en la implementació d'aquesta aplicació serà Java. El sistema Android disposa d'una màquina virtual anomenada Dalvik que serà l'encarregada d'executar el programa.

3.4 Casos d'ús

A continuació el diagrama de casos d'ús planificat per a l'aplicació:



Descripció dels casos d'ús:

Identificador:	CU00
Nom:	Desar posició
Resum:	Emmagatzema la posició actual
Actors:	Usuari
Precondicions:	Cap
Postcondicions:	Cap
Flux normal:	<ul style="list-style-type: none"> • Inici del cas d'ús quan l'usuari sol·licita aquesta acció • L'aplicació consulta la posició mitjançant una consulta al sistema. • Precisió inicial: nul·la i posicionament nul. Informa en barra de notificacions. • Consulta si disposa de cellID. Inicia comptador de temps. Si en 10 segons no localitza cap, guarda la posició anterior. Informa en barra de notificacions. • Consulta si detecta xarxes wifi. Inicia comptador temps. Si en 10 segons no localitza cap, guarda la posició anterior. Informa en barra de notificacions. • Consulta si la funció GPS dóna una posició vàlida. Si en 10 segons no es localitza, guarda la posició anterior. Informa en barra de notificacions. • En cas de localització satisfactòria, informa en barra de notificacions

	i guarda la posició trobada.
Flux alternatiu:	Cap
Inclusions:	Cap
Extensions:	L'accés mitjançant el <i>widget</i> procedirà pel flux normal

Identificador:	CU01
Nom:	Administrar posicions
Resum:	Llista les posicions emmagatzemades i permet accedir-hi / eliminar-les
Actors:	Usuari
Precondicions:	Haver emmagatzemat prèviament alguna posició
Postcondicions:	Cap
Flux normal:	<ul style="list-style-type: none"> • Inici del cas d'ús quan l'usuari selecciona l'opció “Administrar posicions” en el menú principal. • L'aplicació mostra en forma de llistat les posicions emmagatzemades. Els valors mostrats en la llista serà la data i hora en l'obtenció de cada posició i la seva precisió segons la quantificació de precisió
Flux alternatiu:	<ul style="list-style-type: none"> • L'aplicació no deixarà accedir al llistat quan no es compleixi la precondició, és a dir, quan no tingui cap posició emmagatzemada. • L'usuari visualitzarà un quadre informant del motiu
Inclusions:	<ul style="list-style-type: none"> • CU01_1: Seleccionar posició emmagatzemada • CU01_2: Eliminar posició emmagatzemada
Extensions:	Cap

Identificador:	CU01_1
Nom:	Seleccionar posició
Resum:	Accés a la posició escollida mostrant les coordenades desades
Actors:	Usuari
Precondicions:	Cap
Postcondicions:	Cap
Flux normal:	<ul style="list-style-type: none"> • Inici mitjançant la pulsació sobre la posició emmagatzemada dins la llista de posicions • Mostra la posició emmagatzemada amb les seves coordenades, precisió i temps de localització
Flux alternatiu:	Cap
Inclusions:	Cap
Extensions:	Cap

Identificador:	CU01_2
Nom:	Eliminar posició
Resum:	Acció d'esborrar una posició prèviament emmagatzemada
Actors:	Usuari
Precondicions:	Cap
Postcondicions:	Cap
Flux normal:	<ul style="list-style-type: none"> • Inici mitjançant la pulsació llarga sobre la posició emmagatzemada dins la llista de posicions • Informatiu demanant “Desitja eliminar aquesta posició” amb dues respostes: Sí i No
Flux alternatiu:	<ul style="list-style-type: none"> • Arrossegament cap al lateral de la posició per eliminar • Aparició en la part inferior de l'opció “desfer”
Inclusions:	Cap
Extensions:	Cap

Identificador:	CU03
Nom:	Modificar opcions
Resum:	Accés al menú de preferències de l'aplicació
Actors:	Usuari
Precondicions:	Cap
Postcondicions:	<ul style="list-style-type: none"> • El temps total serà la suma de la precisió baixa + mitja + alta • Els temps sempre seran números enters i positius
Flux normal:	<ul style="list-style-type: none"> • Inici del cas d'ús quan l'usuari selecciona l'opció “Opcions” en el menú principal. • Opcions modificables: <ul style="list-style-type: none"> ◦ Temps màxim per millorar la precisió baixa ◦ Temps màxim per millorar la precisió mitja ◦ Temps màxim per millorar la precisió alta • Opció no modificable: <ul style="list-style-type: none"> ◦ Temps màxim total en l'obtenció de localització. Càlcul amb la suma dels anteriors. • Part inferior: Botó per desar la nova configuració i Botó per cancel·lar els canvis
Flux alternatiu:	Cap
Inclusions:	Cap
Extensions:	Cap

Identificador:	CU04
Nom:	Informació d'aplicació
Resum:	Mostra informació de l'aplicació
Actors:	Usuari
Precondicions:	Cap
Postcondicions:	Cap
Flux normal:	<ul style="list-style-type: none"> • Inici del cas d'ús quan l'usuari selecciona l'opció “Sobre SaveMyLocation” en el menú principal • Mostra requadre amb el nom d'aplicació, l'autor i la versió
Flux alternatiu:	Cap
Inclusions:	Cap
Extensions:	Cap

4 Prototip

És difícil preveure l'aspecte visual final que tindrà l'aplicació per qüestions de desconeixement de la plataforma. Hi ha molts aspectes que en el moment de realitzar-los poden resultar poc pràctiques o patir diferències que esdevindran d'improvisar situacions no previstes.

Per tal d'acostar-se el màxim possible a l'objectiu desitjat s'han realitzat esquemes visuals amb la mateixa plataforma de programació que s'utilitzarà per desenvolupar l'aplicació.

4.1 Menú principal

Punt d'entrada a l'aplicació, senzill i funcional ha de permetre l'accés a totes les funcionalitats del programa a excepció del *widget*, que serà accessible únicament des de l'escriptori.

Els requadres de l'esquerra seran convertits en punts visualment més atractius o símbols que representin millor la funció d'aquella opció en concret.

Cal comentar que la icona de l'aplicació en aquesta captura és la predeterminada en la creació d'un projecte d'aplicació d'Android i serà plantejada en punts posteriors.

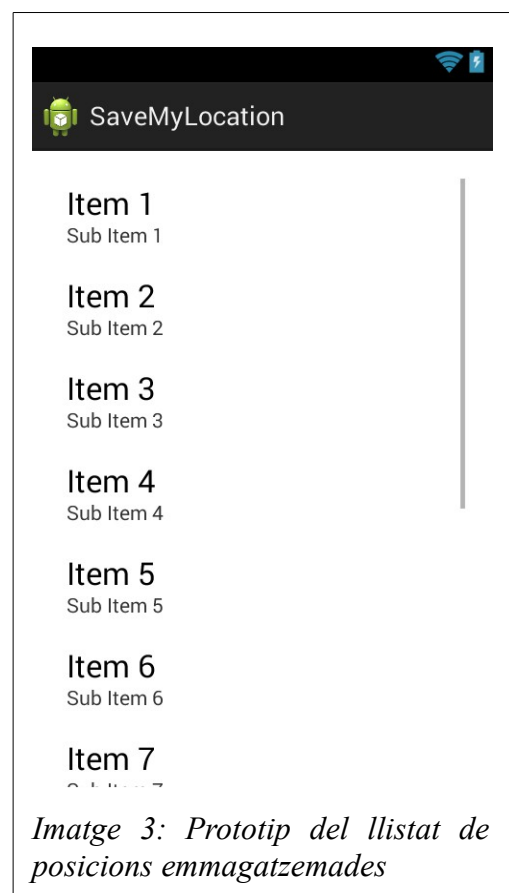
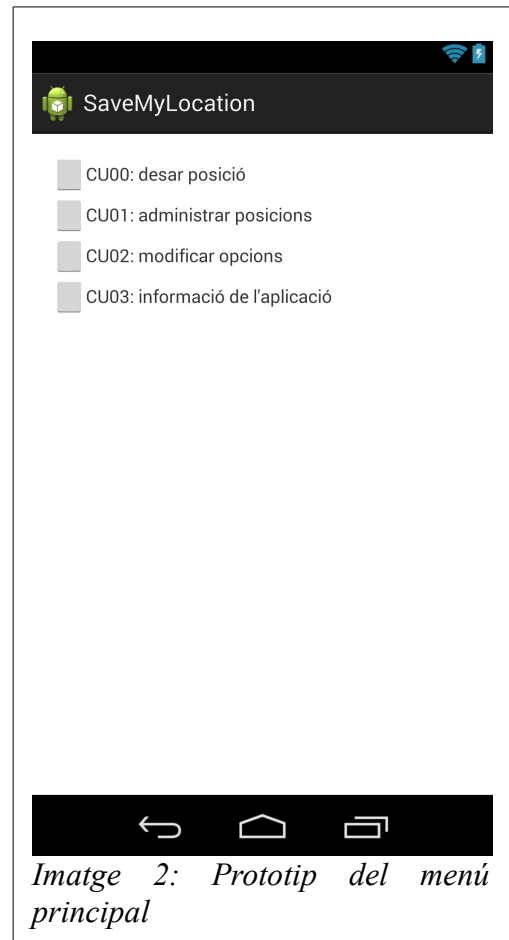
4.2 Administrar posicions

L'aspecte aproximat de la llista de posicions vindrà donat per la funció llista d'Android.

La principal informació que l'usuari ha de veure serà representada per *Item* de la imatge 3. D'una mida més gran que *Sub Item* i com a dada la data i l'hora de la presa de posició. S'estudiarà la possibilitat d'afegir una icona, símbol o punt per remarcar cada element de la llista.

En el camp *Sub Item* es poden representar les coordenades numèriques de cada posició.

És important que la llista tingui capacitat per a més elements dels que pot mostrar la pantalla. Per aquest motiu, ha de tenir desplaçament vertical.



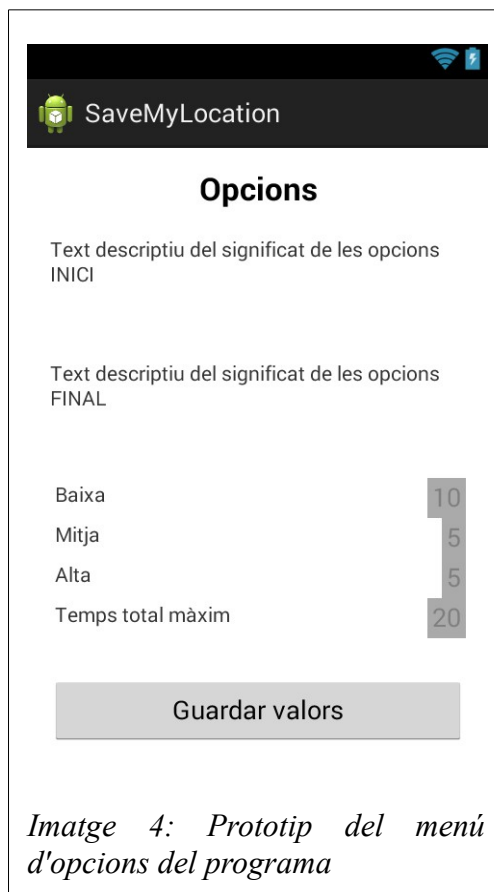
4.3 Modificar opcions

El menú d'opcions pot tenir un aspecte visual semblant a la imatge 4.

Degut a que hi ha pocs paràmetres modificables per l'usuari, realitzar un disseny amb únicament aquestes opcions quedaria molt simple. Inclourà una descripció del significat dels *timeouts* modificables i què representa cadascú d'ells.

El *timeout* total és la suma dels *timeouts* anteriors i seria recomanable que s'actualitzi tant bon punt l'usuari modifiqui un dels valors anteriors. No serà un requisit imprescindible però sí recomanable.

Finalment, un botó per salvar els canvis tornarà al menú principal. En cas de voler cancel·lar les modificacions l'usuari optarà per tornar enrere pel sistema natiu d'Android.



4.4 Sobre el programa

No cal posar molt èmfasi en aquesta opció. La subsegüent pantalla ha d'informar que es tracta d'una aplicació que forma part del projecte de final de carrera així com el nom del seu autor. Crear una activitat nova és excessiu i per tant s'utilitzarà les notificacions en pantalla d'Android, anomenades notificacions *toast*.

4.5 Widget

El disseny del *widget* ha de ser ben senzill doncs la seva única utilitat serà la de cridar la funció desatesa de localització i mostrar-ne el progrés mitjançant la barra de notificacions.

Visualment serà representat per un quadre de color negre amb una icona blanca i/o un text blanc. Per donar lloc a les dues possibilitats es poden crear els dos tipus de *widget* a gust de l'usuari.

La mida serà la mínima possible per contenir el text o la imatge. Android en versions posteriors a la 4.0 permet fer *widgets* de mida variable però si finalment s'assoleix un aspecte com aquest prototip, no té sentit permetre la variabilitat de mida més enllà de la mínima.



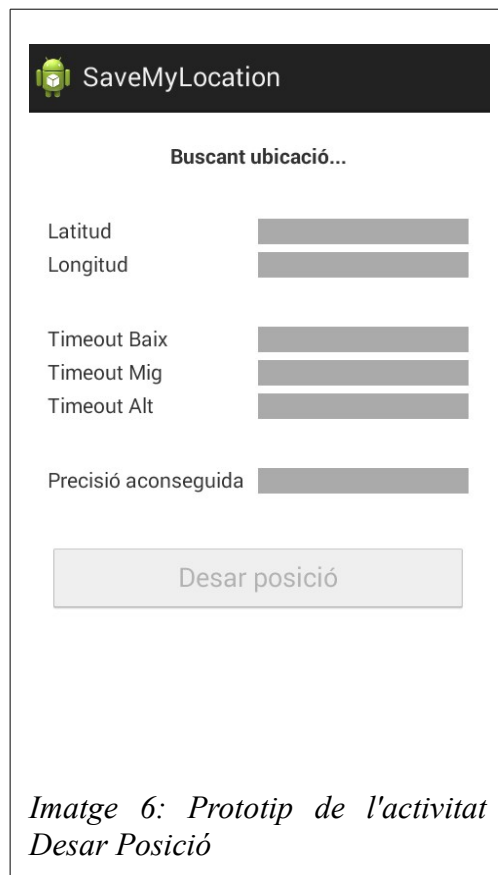
4.6 Desar posicions

Tant bon punt l'usuari entra en aquesta activitat s'inicia el procés d'ubicació per tal d'agilitzar-ho el màxim possible.

Els quadres de coordenades han de reflectir en temps real el posicionament. L'usuari percebrà una millora en la localització gràcies a l'indicador de precisió que mostrarà en forma de text si és baixa, mitja o alta. Una possible millora serà donar colors diferents a l'indicador de precisió en base a la seva millora.

Els quadres de *timeouts* partiran del temps establert per l'usuari en les opcions del programa i començaran a disminuir fins a zero quan s'arribi a cada nivell de precisió.

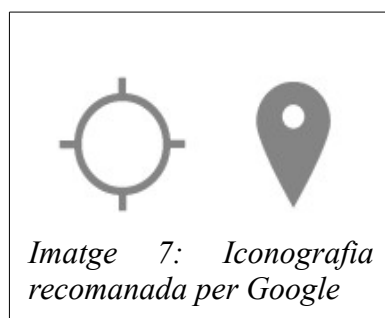
En qualsevol moment l'usuari ha de tenir la possibilitat de guardar la seva posició sempre i quan existeixi alguna dada recollida. També ha de quedar recollida a la base de dades la data i l'hora exacte en que l'usuari desi la posició.



Imatge 6: Prototip de l'activitat Desar Posició

4.7 Icona del programa

En aquests moments encara no s'ha dissenyat la icona que simbolitzarà el programa. Hi ha pautes de disseny o símbols que formen part dels conceptes de bones pràctiques pels quals recomanen utilitzar un tipus d'icones en concret segons la seva funció. Google ofereix tot un seguit d'icones[8] i marquen com a referència de cerca d'ubicació la icona mostrada a l'esquerra i de marca de posició a la dreta (requadre d'imatge 7). Es prendrà com a base de la icona del programa la unió dels conceptes relacionats amb els dos símbols.



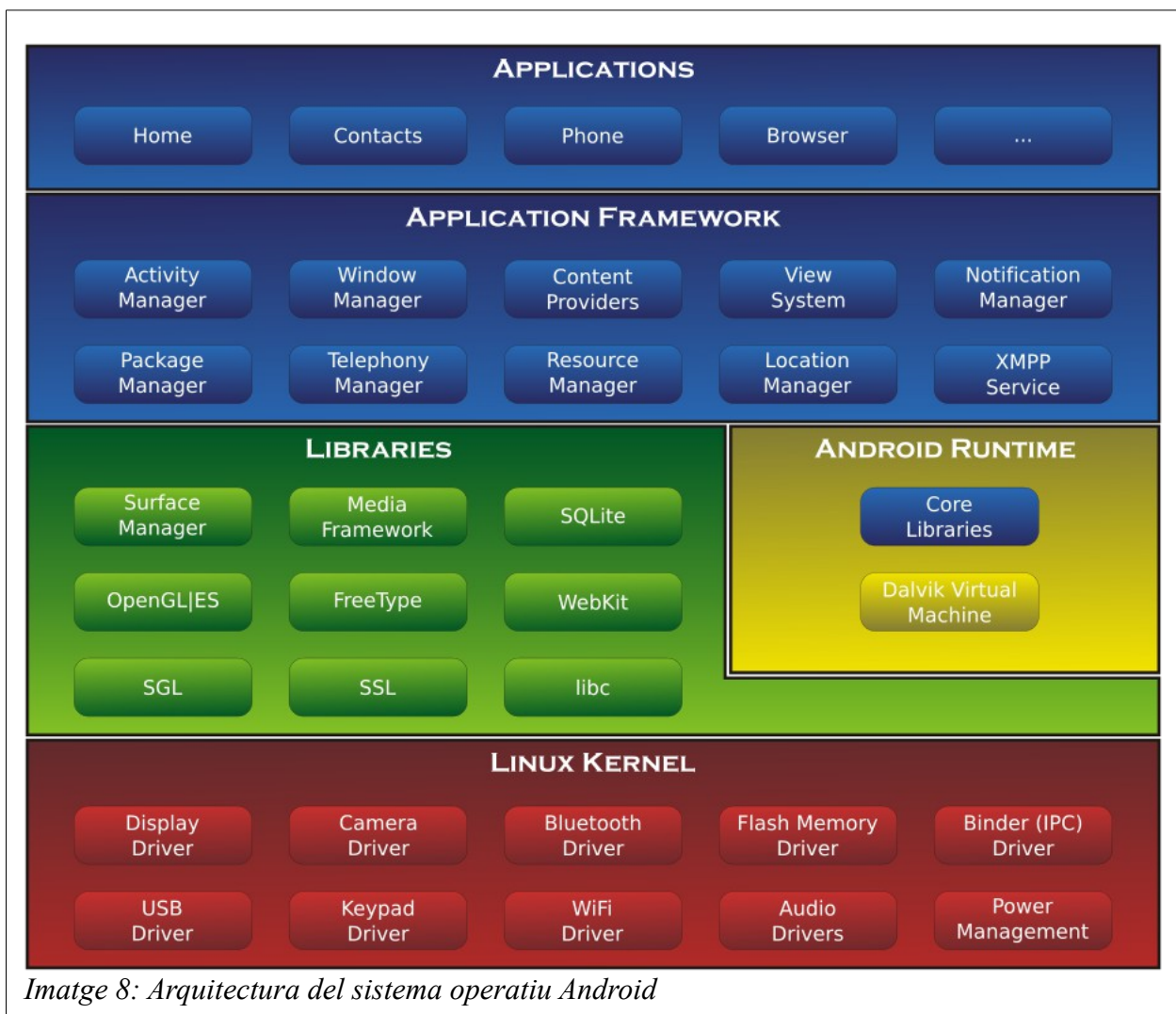
Imatge 7: Iconografia recomanada per Google

5 Disseny tècnic

L'arquitectura del projecte a realitzar estarà basada únicament en l'aplicació per a Android.

5.1 Arquitectura Android

Android[9] és un sistema operatiu creat inicialment per Android Inc. i posteriorment comprada per Google l'any 2005. Està basat en una versió modificada del nucli Linux i l'any 2007 va ser presentat al públic per la fundació Open Handset Alliance. Posteriorment Android va ser alliberat sota llicència lliure Apache 2.0 a excepció de la part del nucli Linux que manté la GPL v2.



Imatge 8: Arquitectura del sistema operatiu Android

De baix a dalt, l'arquitectura Android té les següents capes:

- Nucli Linux: Actua com a capa d'abstracció entre el *hardware* i el *software* i s'encarrega de la seguretat, gestió de memòria, gestió de processos, pila de xarxa, etc.
- Llibreries: escrites en llenguatge C/C++ són utilitzades per components del sistema i accessibles als desenvolupadors d'aplicacions mitjançant el SDK.

- Màquina d'execució *Android Runtime*: les aplicacions Android s'executen sobre una màquina virtual anomenada *Dalvik* a semblança de la màquina virtual de Java.
- *Framework* d'aplicacions: són tot un seguit de classes accessibles per la capa superior d'aplicacions que ofereixen multitud de serveis i funcions.
- Capa d'aplicacions: són totes les aplicacions de tercers escrites en Java i que tenen accés al *framework* per realitzar les seves funcions. L'aplicació creada en aquest projecte forma part d'aquesta capa.

5.2 Emmagatzematge de les dades

L'aplicació necessita emmagatzemar dos grups de dades:

- Els *timeouts* escollits per l'usuari:
Android posseeix unes llibreries de tractament XML que amb l'ús del *framework* corresponent anomenat *SharedPreferences* està perfectament indicat per ser utilitzat en aquest cas. Per tant, les preferències de l'usuari seran emmagatzemades amb aquest sistema.
- Les posicions desades per l'usuari:
Android també proveeix dins el seu *framework* llibreries pel tractament de base de dades, concretament SQLite. Les grans possibilitats que ofereix aquest sistema el fan ideal per emmagatzemar les posicions de l'aplicació.

La base de dades només tindrà una taula i encara que SQLite sigui una base de dades relacional, aquesta característica no serà necessària en cap moment. L'estructura de la taula serà:

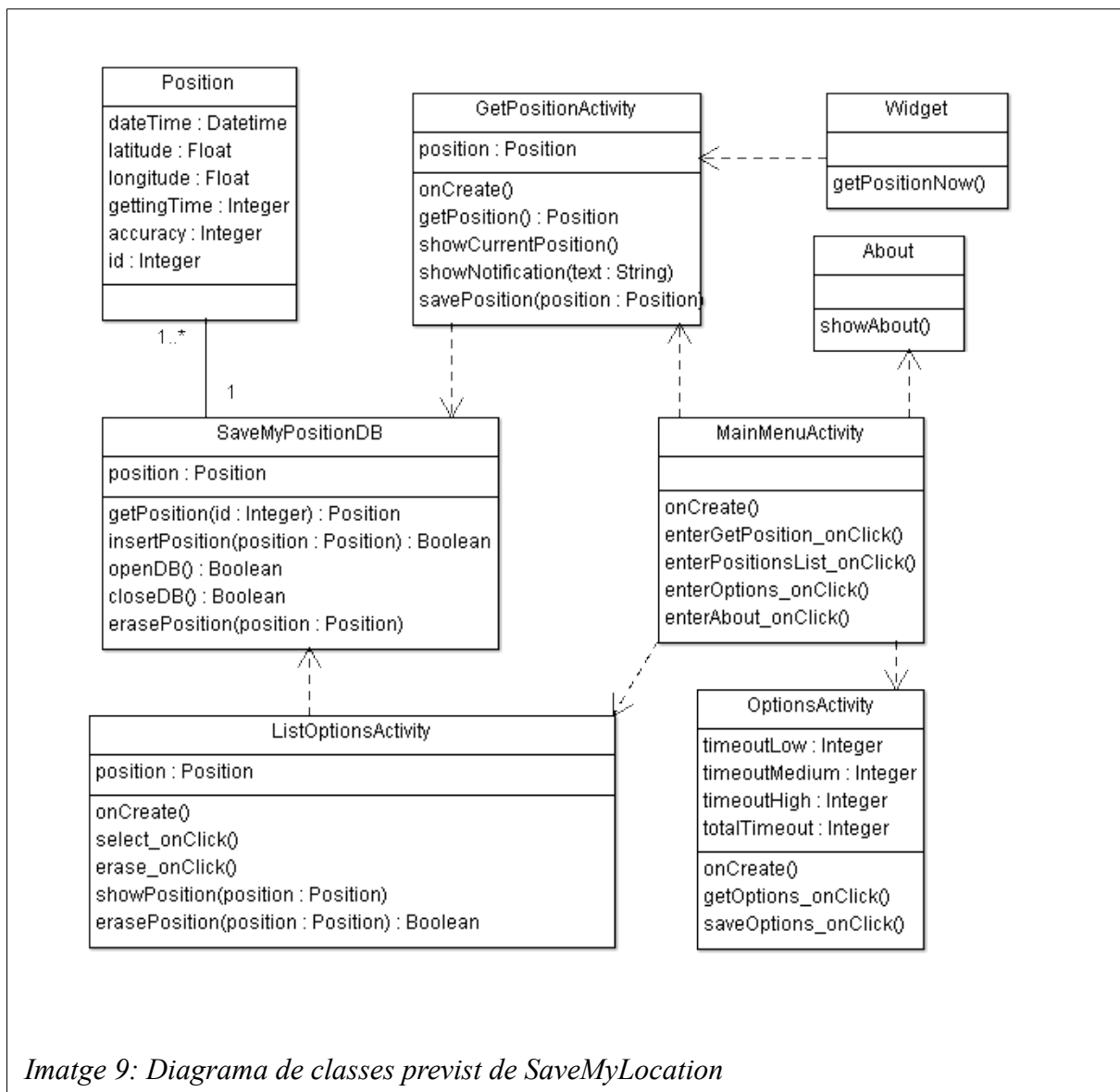
SaveMyLocationDB		Nom de la base de dades
positions		Nom de la taula
Nom	Tipus	
dateTime	DATETIME	Data i l'hora de la presa de posició. Format: YYYY-MM-DD HH:MM:SS
latitude	FLOAT	Coordenades de la latitud
longitude	FLOAT	Coordenades de la longitud
gettingTime	INTEGER	Temps transcorregut ubicant la posició
accuracy	INTEGER	Precisió en metres del posicionament

5.3 Diagrama de classes

El següent diagrama mostra l'estructura de classes previst que tindrà l'aplicació.

El mètode onCreate() està en totes aquelles classes considerades activitats i que Android necessita per poder mostrar la pantalla a l'usuari.

El mètode acabat amb onClick() forma part de les accions que pot realitzar l'usuari quan clica a la pantalla.



A continuació una descripció breu dels mètodes a utilitzar:

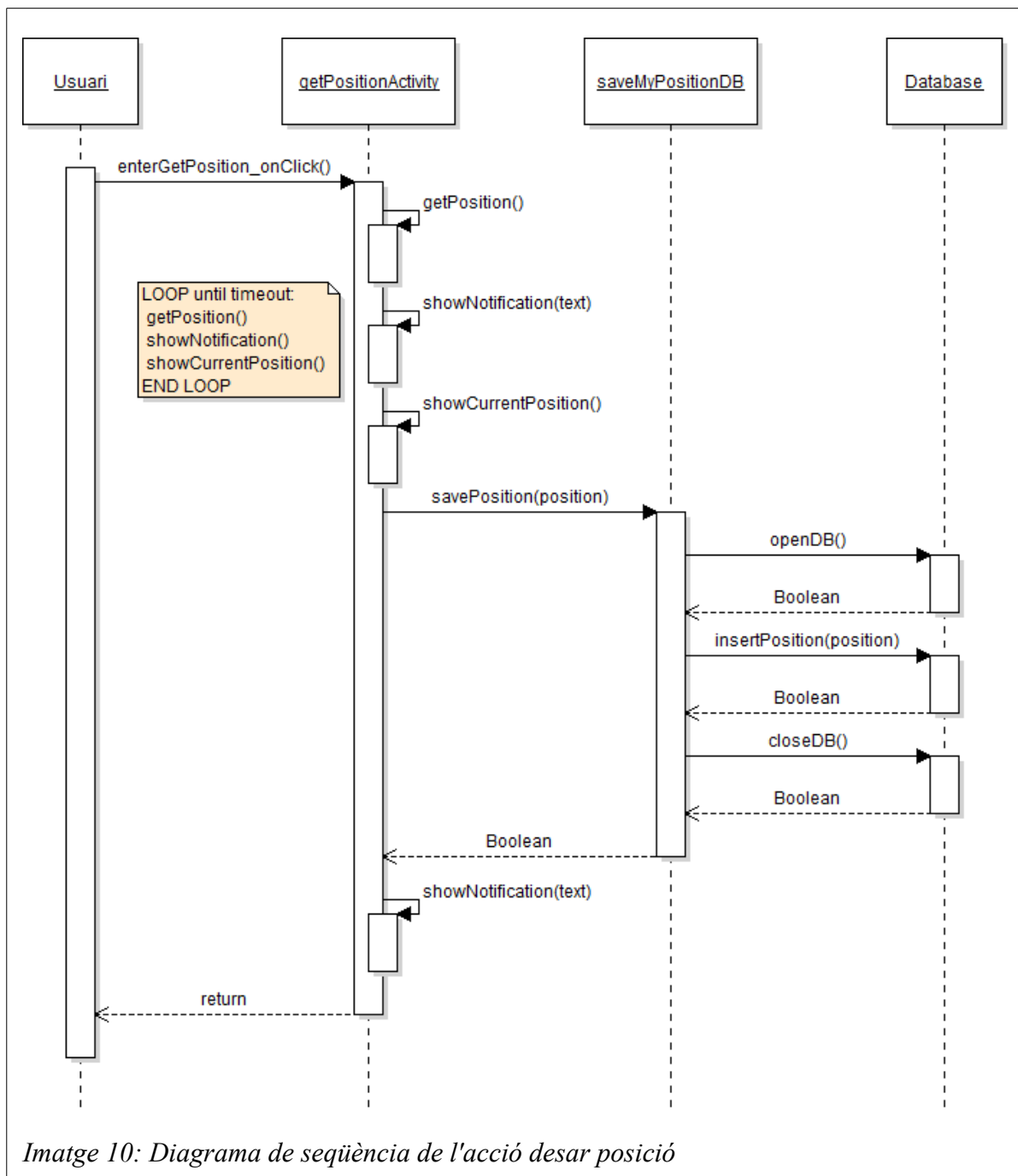
Classe	Mètode	Descripció
MainMenu	EnterGetPosition	Entra a l'activitat d'ubicar la posició
MainMenu	EnterPositionList	Entra a l'activitat on llista totes les posicions emmagatzemades dins la base de dades
MainMenu	EnterOptions	Entra a l'activitat per veure i modificar els <i>timeouts</i>
MainMenu	EnterAbout	Crida a la funció per mostrar informació del programa
GetPosition	GetPosition	Inicia la localització del dispositiu i cridarà a <code>savePosition()</code> si es localitza correctament
GetPosition	ShowCurrentPosition	Mostra les coordenades i la precisió del posicionament
GetPosition	ShowNotification	Informa del progrés a la barra de notifikacions
GetPosition	SavePosition	Desa les coordenades, temps i precisió a la base de dades
Options	GetOptions	Recull els <i>timeouts</i> de les <i>SharedPreferences</i>
Options	SaveOptions	Desa els <i>timeouts</i> de l'usuari a les <i>SharedPreferences</i>
ListPositions	ShowPosition	Mostra les coordenades i precisió d'una posició
ListPositions	ErasePosition	Elimina un registre de la base de dades. Retorna false si error
About	ShowAbout	Mostra quadre amb la versió i el creador del programa
Widget	GetPositionNow	Crida a la classe <code>getPosition</code> en segon plà per desar la posició sense mostrar l'activitat
SaveMyPositionDB	GetPosition	Obté una posició de la BD donat un ID
SaveMyPositionDB	InsertPosition	Desa una posició a la BD. Retorna false si hi ha error
SaveMyPositionDB	OpenDB	Obra la base de dades
SaveMyPositionDB	CloseDB	Tanca la base de dades
SaveMyPositionDB	ErasePosition	Elimina una posició de la base de dades

5.4 Diagrames de seqüència

A continuació la previsió dels diagrames de seqüència de les dues accions més importants que pot realitzar l'usuari. Les restants seran improvisades en temps d'implementació per la seva baixa complexitat.

5.4.1 Acció desar posició

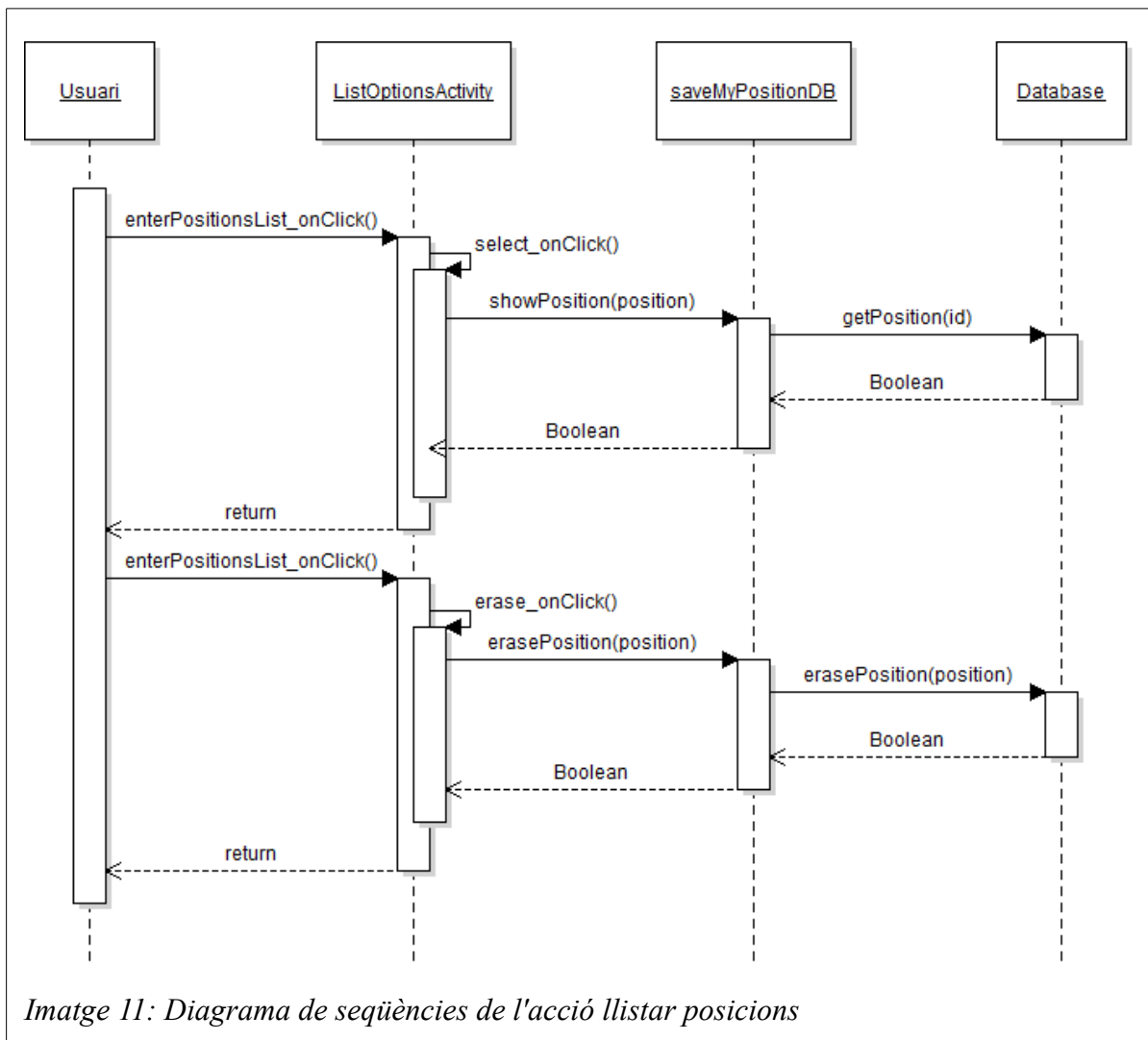
Tant bon punt entra l'usuari a `getPositionActivity` s'inicia el bucle de localització segons els *timeouts* corresponents. Si troba la posició s'inicia el procés de salvar-la fent ús de la classe `saveMyPositionDB` que obrirà, insertarà i tancarà la base de dades. Una notificació final informarà a l'usuari del resultat del procediment.



Imatge 10: Diagrama de seqüència de l'acció desar posició

5.4.2 Acció llistar posicions

L'usuari entrarà a l'activitat ListOptionsActivity on es mostrarà totes les posicions emmagatzemades i podrà marcar o esborrar una posició. Les dues accions cridaran a showPosition o erasePosition respectivament i serà la classe SaveMyPositionDB l'encarregada de treballar amb la base de dades per dur a terme les dues funcions. Les respostes seran de tipus booleà amb cert si s'ha realitzat correctament i fals si hi ha hagut algun tipus d'error.



6 Implementació i funcionament de l'aplicació

Tot seguit es detallarà el procés realitzat en la implementació així com l'estructura de l'aplicació, totes les seves funcionalitats i recursos utilitzats.

6.1 Classes en Java

L'aplicació consta de 10 classes programades en llenguatge Java:

- **GetPositionActivity**

Presenta la funcionalitat de buscar la posició del dispositiu i guardar-la dins la base de dades.

- **MainActivity**

Menú principal per accedir a totes les funcionalitats de l'aplicació (menys al *widget*).

- **OptionsActivity**

Permet modificar els valors de temps de les variables *timeout* per l'ús desatès del *widget*.

- **PositionEdit**

Presenta les coordenades tant en format numèric com sobre el mapa d'una posició guardada a la base de dades. Permet també les modificacions d'aquestes.

- **PositionsListActivity**

Mostra la llista de totes les posicions emmagatzemades.

- **PositionsListDBAdapter**

Conté els mètodes per realitzar funcions sobre la base de dades.

- **SaveMyLocationDefaultValues**

Recopilació de variables estàtiques per defecte a tot el programa.

- **SaveMyLocationSQLiteHelper**

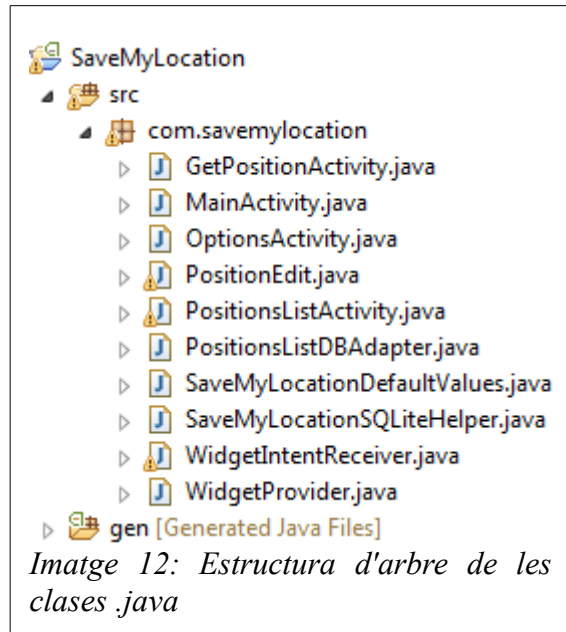
Defineix l'estructura de la base de dades.

- **WidgetIntentReceiver**

Defineix les funcionalitats del *widget*.

- **WidgetProvider**

S'encarregarà de cridar les funcionalitats corresponents del *widget*.

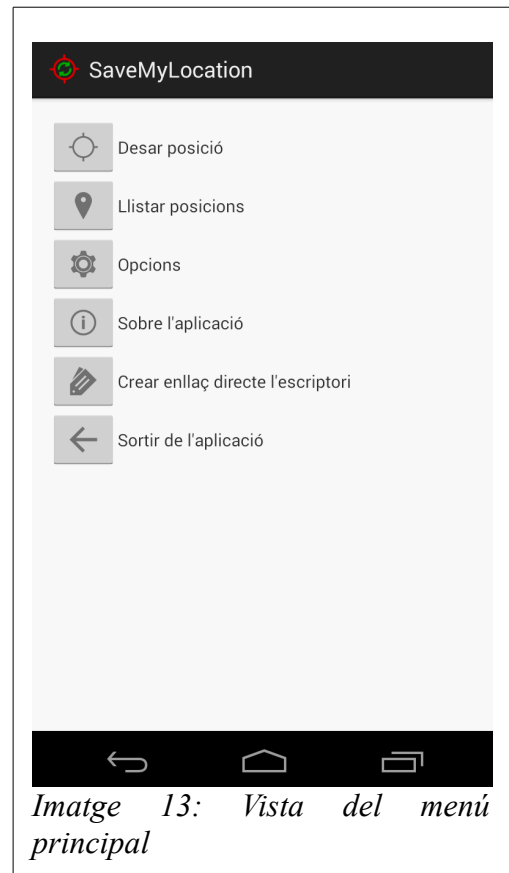
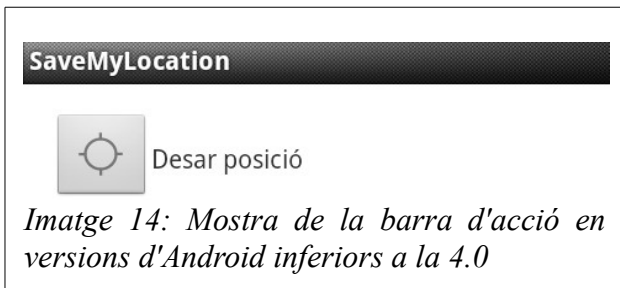


6.1.1 Menú principal (MainActivity)

El menú principal representa el punt d'entrada a l'aplicació i permet l'accés a totes les funcionalitats d'aquesta a excepció del *widget*, que serà accessible únicament des de l'escriptori.

L'esquema visual en forma de llista és senzill i funcional. Cal comentar certes diferències en quan a la barra d'acció *action bar*, és a dir, la barra superior que titula cada activitat. El motiu és fruit de l'evolució d'Android on per tal de permetre visualitzar la barra en versions anteriors a la 4.0 cal fer ús d'unes llibreries de compatibilitat.

L'aspecte en una versió anterior serà així:

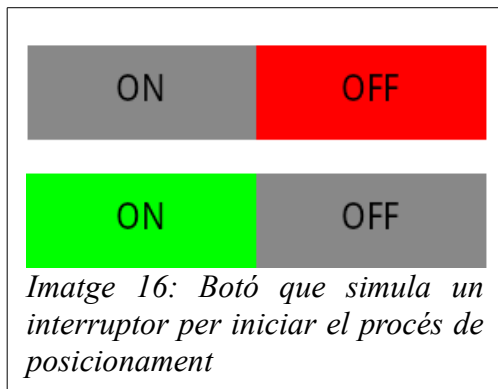


Estèticament menys agradable, aquesta i altres diferències existeixen en totes les activitats de l'aplicació. La majoria de les captures d'aquesta memòria han sigut realitzades amb el terminal de prova (HTC Desire amb Android 2.3.7) i es comentaran aquests petits detalls.

6.1.2 Desar posició (GetPositionActivity)

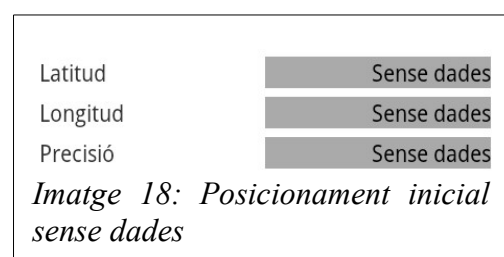
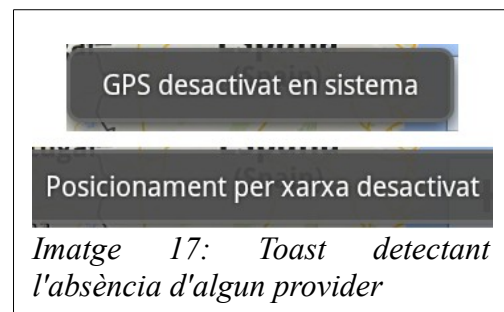
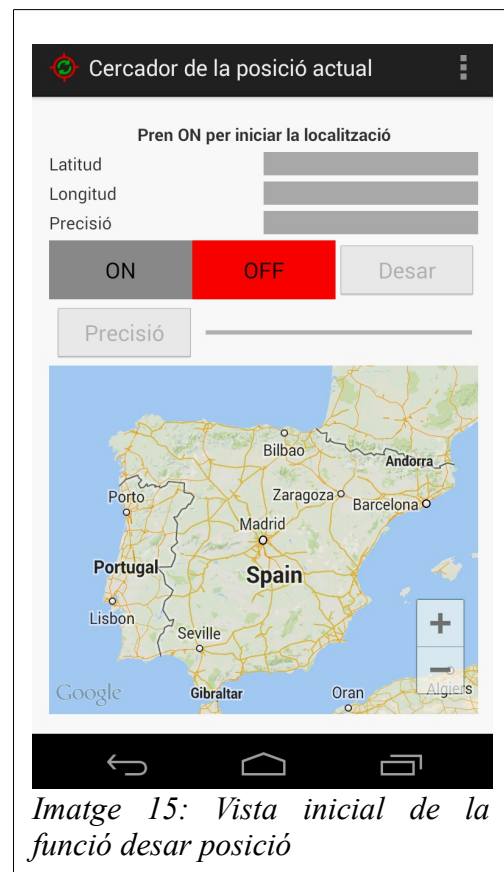
La primera i principal funció de l'aplicació és posicionar al dispositiu mòbil en l'espai. Aquesta activitat permet ubicar-lo sense tenir en compte les limitacions de temps doncs serà l'usuari qui tindrà en tot moment el control per engegar i parar el procés com també per desar la posició.

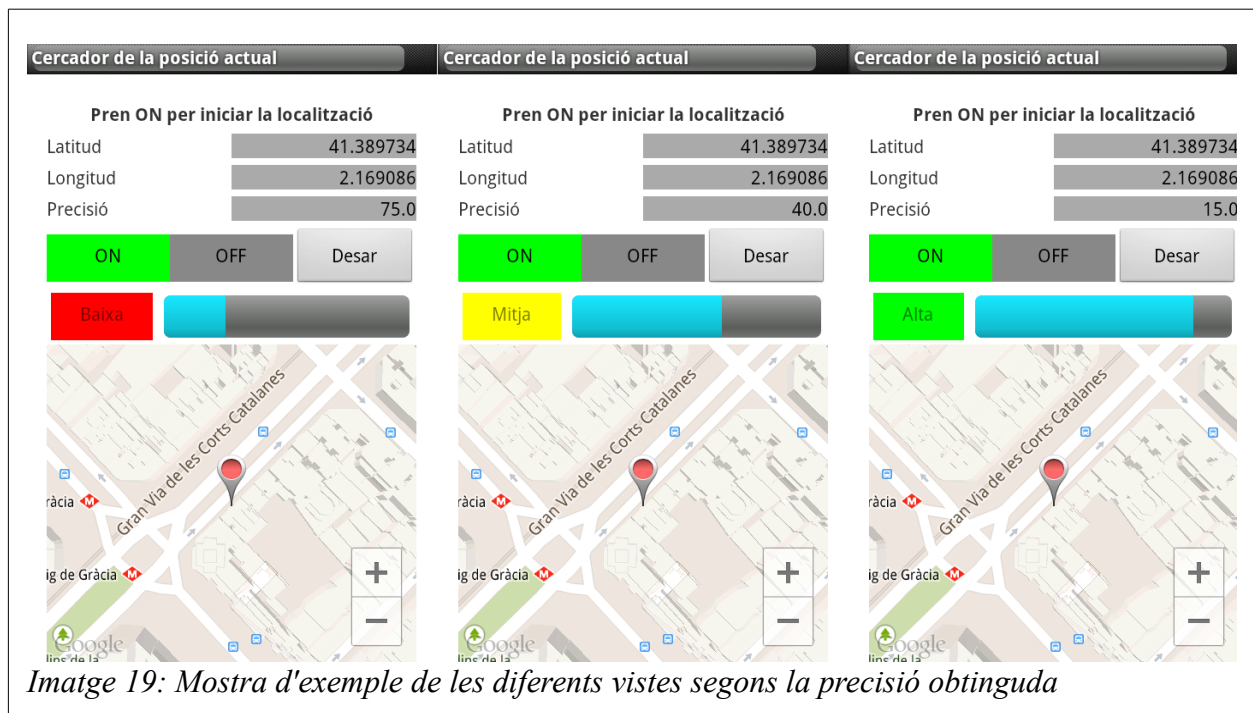
El primer pas és iniciar la localització amb l'ús del botó ON que visualment queda pintat de verd:



El *listener setOnClick* prèviament declarat detecta la pulsació i s'inicia el mètode *updatePosition*. Registrats els *listeners* corresponents que actuaran quan rebin una posició, comença el procés que encarrega al *LocationManager* l'engegada dels dos serveis de posicionament més comuns en telèfons intel·ligents: mitjançant xarxes de telefonia/wifi (*network_provider*) i amb l'ús del sensor GPS (*gps_provider*). Si qualsevol dels dos no està disponible, ja sigui per des-habilitació o que tal aparell no en disposa, l'usuari és informat en pantalla amb el tipus de notificació *toast*.

En un primer instant no existeix cap dada de posicionament i així ho mostra l'aplicació. El botó per desar la posició romandrà inhabilitat fins l'obtenció de les primeres dades.





Tan bon punt es detecti la primera posició serà mostrada en temps real omplint el formulari amb les coordenades. La precisió ve donada en els metres a la rodona de marge d'error que calcula el dispositiu. Per la seva importància, la precisió queda especialment reflectida en un quadre que canvia el color i el text a baixa, mitja o alta precisió. A més a més, just al costat una barra de progrés augmenta a partir dels cent metres fins a zero. El mapa i el cursor fa una representació visual de les coordenades desplaçant la càmera i el cursor fins a tal punt. El botó desar posició passa a estar habilitat a l'espera que l'usuari desitgi salvar les dades mostrades. Com el plantejament inicial de l'aplicació és obtenir la màxima precisió possible, qualsevol posició nova rebuda pel *listener* serà comprovada amb la més acurada obtinguda fins el moment i descartada si la precisió és menor (com més llunyà sigui del zero).

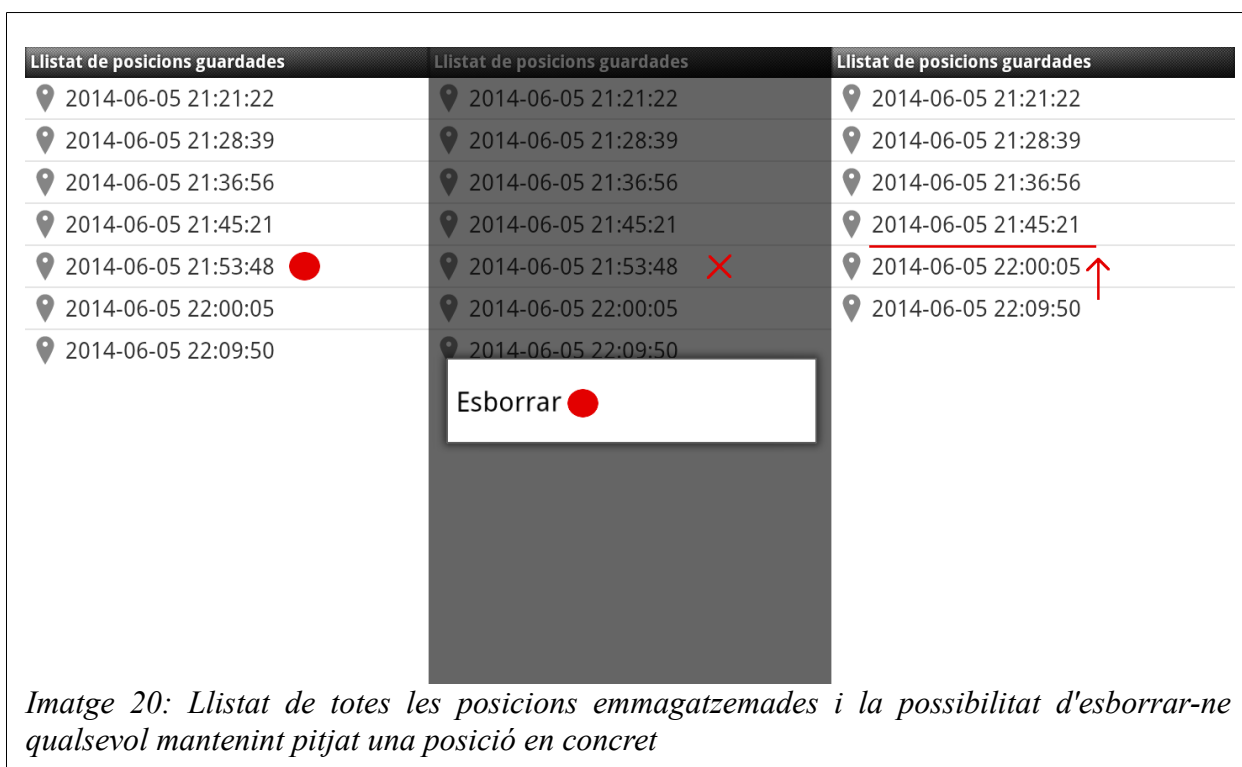
Arribat a aquest punt, l'usuari pot desar la posició tants cops com desitgi. Una notificació *toast* informa de l'acció i la posició queda enregistrada a la base de dades amb l'afegit de l'hora i dia exacte del moment realitzat. Tant si atura el procediment com si retrocedeix enrere cap al menú principal, l'aplicació aturarà els sensors per no malgastar bateria.

6.1.3 Llistar posicions (PositionListActivity – PositionsListDBAdapter)

La segona funció correspon al tractament de la base de dades amb les posicions emmagatzemades.

Aquesta vista les mostra totes en forma d'un llistat de files amb la data i l'hora de la presa com a informació resumida. L'usuari pot esborrar els registres individualment mantenint pitjat un registre on apareixerà un quadre amb l'opció d'esborrar.

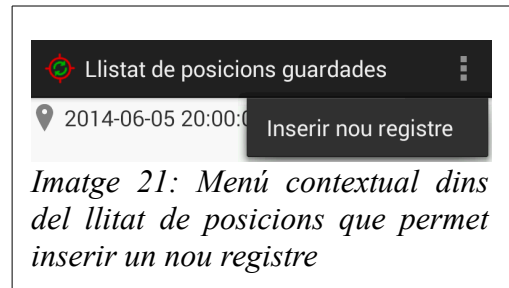
També pot afegir nous registres de forma manual pitjant sobre la tecla menú contextual i marcant inserir nou registre.



Imatge 20: Llistat de totes les posicions emmagatzemades i la possibilitat d'esborrar-ne qualsevol mantenint pitjat una posició en concret

6.1.3.1 Inserir/Editar una posició (PositionEdit)

A través del llistat anterior, amb una pulsació sobre la posició desitjada o mitjançant la tecla de menú contextual s'entra dins l'activitat d'edició.



Imatge 21: Menú contextual dins del llistat de posicions que permet inserir un nou registre

Existeixen dos possibilitats d'edició:

- Introducció manual de les coordenades. En pitjar sobre el formulari s'activarà el teclat numèric i l'usuari sols podrà entrar números positius i negatius. Cal fer ús del botó actualitzar mapa per visualitzar els canvis.
- Desplaçament del cursor sobre el mapa mantenint pitjat el dit sobre ell. Les coordenades s'actualitzaran en temps real.

Un cop finalitzat, l'usuari pot desfer els canvis o tornar enrere per cancel·lar-los. Si decideix guardar-los l'aplicació realitza una comprovació de la validesa de les coordenades entrades i notifica qualsevol error trobat.



Imatge 22: Edició d'una posició de forma manual tant amb la introducció de coordenades com amb el desplaçament de la marca sobre el mapa

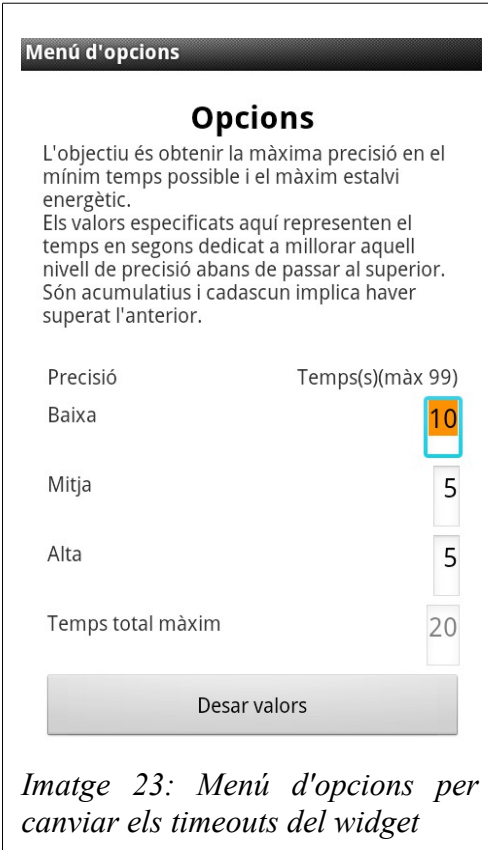
6.1.4 Opcions (OptionsActivity)

La vista del menú d'opcions permet a l'usuari fer canvis en els *timeouts* que s'utilitzen en l'execució del *widget*.

Tal com explica el text descriptiu, els temps especificats són els segons que dedicarà l'aplicació en intentar millorar aquell nivell de precisió per tal d'aconseguir passar al nivell superior. Els temps per defecte són fàcilment modificables en l'arxiu de variables globals *SaveMyPositionDefaultValues* i han quedat establerts de la següent manera:

- 10 segons per a la precisió baixa
- 5 segons per a la precisió mitja
- 5 segons per a la precisió alta

El temps total màxim vindrà donat per la suma dels anteriors. Per tal d'aconseguir actualitzar el comptador sumatori en temps real ha sigut necessari la creació d'un *Thread* paral·lel que cada segon fa la suma i actualitza la vista.



Menú d'opcions

Opcions

L'objectiu és obtenir la màxima precisió en el mínim temps possible i el màxim estalvi energètic.
Els valors especificats aquí representen el temps en segons dedicat a millorar aquell nivell de precisió abans de passar al superior. Són acumulatius i cadascun implica haver superat l'anterior.

Precisió	Temps(s)(màx 99)
Baixa	10
Mitja	5
Alta	5
Temps total màxim	20

Desar valors

Imatge 23: Menú d'opcions per canviar els timeouts del widget

Un aspecte millorable és la justificació completa del text descriptiu. Android no ofereix cap mecanisme natiu per ajustar el text als límits de la vista. Tots els exemples trobats[10] tracten de d'aplicar el mateix concepte mitjançant una inclusió d'un quadre codificat amb html[11] amb la conseqüent complexitat i consum de recursos. Finalment

6.1.5 Widget (WidgetIntentReceiver – WidgetProvider)

El *widget* proporciona l'accés a la funció principal de localització de l'aplicació de la forma més ràpida i desatesa possible. L'usuari ha de col·locar-lo en un lloc lliure de l'escriptori (anomenat *homescreen*) ocupant una quadrícula mínima de 3x2. Partint d'aquesta mida mínima, s'ha programat la possibilitat d'expandir la mida tant com l'usuari desitgi. No obstant, aquesta funció no existeix en versions d'Android anteriors a la 4.0 i per aquests casos la mida per defecte inalterable serà de 4x2.

La pantalla inicial descriu els efectes dels dos possibles punts d'acció que disposa:

- Actualitzar: el *widget* recupera la darrera posició de la base de dades, n'extreu les coordenades i calcula l'adreça.
- Posicionar: inicia el procés d'ubicació de forma desatesa.



Tal com mostra la imatge 25, en el procés de posicionament l'usuari veu a la barra l'inici de l'acció de posicionament. Els següents casos representen les possibles respostes obtingudes:

1. Inici del procediment
2. S'ha esgotat el temps de precisió baixa i el dispositiu no ha sigut capaç de trobar cap posició (ni tan sols l'última coneguda pel dispositiu).
3. S'ha esgotat el temps de precisió baixa i el dispositiu s'ha localitzat però no ha pogut millorar la precisió. Per tant, salva la posició a pesar de tenir una precisió baixa.
4. En la cerca s'ha superat la precisió baixa i ha sumat el temps per millorar la mitjana. No ha sigut possible obtenir l'alta i salva la posició.
5. En cada etapa de temps s'ha localitzat amb una precisió del nivell superior i ha arribat a la màxima exactitud en el màxim de temps permès. Arribat a aquest punt òptim, emmagatzema la millor posició.



6.1.6 Altres funcions

6.1.6.1 Canvi de tipus de mapa

Durant la programació de l'aplicació s'ha inclòs la possibilitat de canviar el tipus de mapa tant en el moment d'obtenir la posició manualment (*GetPositionActivity*) com en el moment d'editar una manualment (*PositionEdit*). Aquesta funció no estava prevista en la planificació però la implementació no ha representat gaires dificultats.



6.1.6.2 Sobre l'aplicació

Aquesta funció del menú principal és una petita mostra en forma de notificació *toast* sobre el creador de l'aplicació així com el motiu de la creació d'aquesta. Aquest apartat, molt comú en tots els programes, és una traducció literal de la coneguda opció *about*.

6.1.6.3 Crear enllaç directe a l'escriptori

Android ofereix la possibilitat de crear accessos directes des de l'escriptori en comptes d'accedir únicament des del menú d'aplicacions. Aquesta opció facilita a l'usuari la creació de l'enllaç. Una notificació *toast* informarà de la correcta creació.

6.1.6.4 Sortir de l'aplicació

El sistema operatiu inclou un sistema propi d'alliberament de memòria en cas que l'usuari deixi d'utilitzar l'aplicació. Tanmateix, l'opció tancarà directament l'aplicació i la base de dades alleugerant l'experiència d'usuari tant bon punt decideixi finalitzar el seu ús.

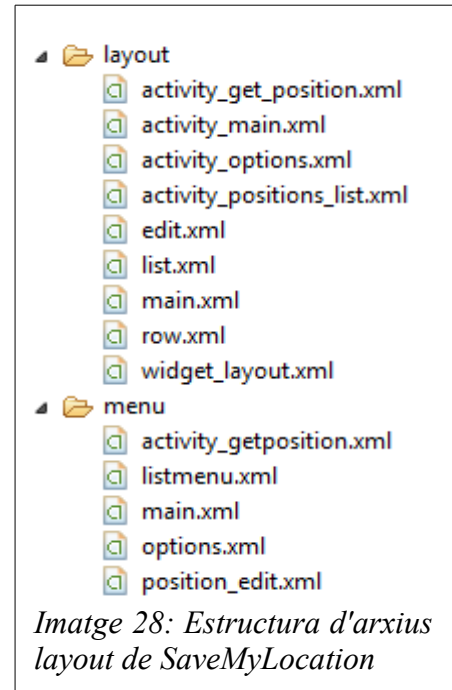


6.2 Layouts en XML

La representació visual de les pantalles d'activitat s'anomenen *layouts* i estan codificades en un llenguatge d'etiquetes XML[12].

Els arxius que pegen de *menu* corresponen als menús contextuais com és el cas d'inserir un registre nou, canviar la visualització de mapa, ...

L'esquema visual de totes les activitats té una certa semblança per no perdre la sensació d'estar dins la mateixa aplicació i facilitar l'aprenentatge.



6.3 Recursos gràfics

Els recursos gràfics utilitzats en l'aplicació han sigut extrets dels exemples oferts a la comunitat[13] per part de Google sota llicència *Creative Commons*.

La icona principal de l'aplicació és una creació pròpia amb l'ús del programa Gimp que consta de dues de les icones anteriors superposades i pintades de verd i vermell.

Android utilitza un sistema de carpetes on les icones es classifiquen per la seva resolució/mida. Així, de forma automàtica, l'aplicació prendrà una o altra segons les especificacions de pantalla del dispositiu on s'executi l'aplicació.



6.4 Arxiu de cadenes de text

Seguint les recomanacions de bones pràctiques, les cadenes de text corresponents a els *layouts* estan especificades en un arxiu anomenat *strings.xml* facilitant la traducció a múltiples idiomes en el futur.

6.5 Arxiu de configuració (AndroidManifest.xml)

Tota aplicació Android conté un arxiu de configuració *AndroidManifest.xml* que detalla certs aspectes de l'aplicació. Pel cas de *SaveMyLocation* les modificacions han sigut:

- **Permisos**

Per tal que una aplicació pugui accedir a les funcionalitats del dispositiu on s'executa, Android requereix una declaració de cadascú d'ells. Aquest aspecte requereix una optimització especial atès que l'usuari ha d'acceptar específicament aquests permisos per autoritzar la instal·lació de l'aplicació en el seu sistema. La proliferació d'aplicacions malicioses genera certa desconfiança i un dels motius principals en el desistiment d'instal·lació és el dubte de per què tal aplicació requereix accedir a un determinat recurs.

S'ha minimitzat al màxim els permisos requerits per tal de garantir un correcte funcionament de l'aplicació:

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
```

- *Access_Coarse_Location* i *Access_Fine_Location* són necessaris per accedir al sistema de localització de xarxa/wifi i de GPS respectivament.
- *Internet* dona accés a la connectivitat de l'aplicació que serà utilitzada per descarregar la visualització dels mapes.
- *Access_Network_State* necessari per tal que la API de Google Maps pugui comprovar l'estat de connexió per descarregar dades.
- *Write_External_Storage* és certament polèmic doncs dona accés al contingut de la targeta externa de memòria quan l'aplicació no ho necessitaria. No ha sigut possible retirar aquest permís per ser requisit imprescindible per utilitzar les funcions de Google Maps[14].
- *Install_Shortcut* permetrà a l'aplicació crear l'accés directe a l'escriptori.

- **Declaració d'activitats**

Qualsevol activitat ha d'estar declarada prèviament aquí. En total, l'aplicació consta de 5 activitats. *MainActivity* és la principal i la que serà llançada en iniciar.

```
<activity
    android:name="com.savemylocation.MainActivity"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

```

        </intent-filter>
    </activity>
    <activity
        android:name="com.savemyLocation.OptionsActivity"
        android:label="@string/title_activity_options"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="com.savemyLocation.PositionsListActivity"
        android:label="@string/title_activity_positions_list"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="com.savemyLocation.PositionEdit"
        android:label="@string/title_activity_details"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="com.savemyLocation.GetPositionActivity"
        android:label="@string/title_activity_get_position"
        android:screenOrientation="portrait" >
    </activity>

```

- **Intents del *widget***

Defineix la interacció que podrà realitzar l'usuari amb el *widget* per executar les seves dues funcions.

```

<receiver android:name=".WidgetProvider" >
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_provider" />
</receiver>
<receiver
    android:name=".WidgetIntentReceiver"
    android:label="@string/app_name" >
    <intent-filter>
        <action
            android:name="com.savemyLocation.intent.action.UPDATE_WIDGET" />
        <action
            android:name="com.savemyLocation.intent.action.UPDATE_LOCATION" />
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_provider" />
</receiver>

```

- **Requisits de l'API Google Maps versió 2**

La interacció de les aplicacions de Google amb els seus servidors funciona mitjançant la seva API corresponent. Per permetre l'accés i tenir un control de les peticions, Google API requereix d'una clau que s'obté mitjançant un certificat del creador de l'aplicació.

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyAxrDbFFQ90a5RjV2WLRwjyAoYqSt1bl6Q" />
```

També és necessari incloure les extensions OpenGL que seran utilitzades pels mapes.

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />
```

6.6 Funcions del SDK d'Android aplicades en la programació

A mida que s'avança en la programació de l'aplicació i cal satisfer la necessitat d'ampliar coneixements en noves funcions hom va descobrint que s'han anat aplicant els conceptes dels diversos tutorials adaptats d'una forma o altra. Un cop finalitzat, es pot esmentar amb certesa que l'aplicació fa ús dels següents recursos:

- Estructures visuals *Layouts*
 - Formularis: *TextView*, *EditText*, *Buttons*, ...
 - Llistes
 - *Fragments*
- Emmagatzematge i tractament de preferències amb l'ús de *SharedPreferences*
- Emmagatzematge i tractament d'informació en base de dades SQL
- *Widgets*
- Menús contextuais
- Localització geogràfica
- Notificacions
 - *Toast*
 - Barra d'estat
- *Threads*
 - *AsyncTask*
- Llibreries de tercers
 - Google Maps API v2

7 Conclusions

El desenvolupament de SaveMyLocation ha sigut una tasca força més complicada del que tenia previst. La manca de coneixements en el camp de la programació Android ha suposat un repte rere l'altre. En canvi, la programació en Java la qual tenia força oblidada no ha sigut especialment problemàtica (a excepció d'un cert malbaratament de temps en localitzar errors de codi que amb experiència s'aprecien molt abans).

A mida que s'avança en la programació van sorgint idees, improvisacions, conceptes nous de com fer millor una cosa encara que no estava previst. Cada pas d'aquests, per ben petit que fos, si era un concepte nou requeria d'una nova cerca a internet, documentar-se i posar-ho en pràctica. Això és lent, molt lent, i principal motiu del meu retard sobre la planificació inicial.

La primera conclusió es alhora un agraïment a la gran comunitat de persones que utilitzen aquesta plataforma i comparteixen els seus coneixements per Internet. M'ha sorprès comprovar que qualsevol dubte algú ja s'hi havia trobat anteriorment i és gràcies a aquesta gran comunitat que Internet és ben ple de respostes.

Tot seguit, més conclusions sorgides un cop finalitzat aquest projecte.

7.1 Assoliment d'objectius

La conclusió que he obtingut en les proves realitzades tant a interior com a exterior es que la meva aplicació **funciona i fa el que havia de fer**.

L'objectiu principal des del primer moment sempre ha sigut la localització més precisa amb els mínims recursos necessaris. Tenint en compte que cada segon dedicat a millorar la precisió és un segon perdut si no s'aconsegueix, l'aplicació realitza correctament la seva funció en funció d'uns timeouts que l'usuari pot ajustar a les seves necessitats.

No menys important és la possibilitat d'executar l'acció de la forma més accessible possible. L'accés directe mitjançant el widget amb una sola pulsació fa realitat aquest objectiu.

Com a tercer objectiu està la capacitat de funcionar de forma desatesa. Un usuari pot treure el mòbil de la butxaca, engegar la pantalla, pitjar el widget i tornar-lo a guardar.

A nivell personal també he assolit objectius. SaveMyLocation ha sigut el meu primer contacte amb la programació per a mòbils i el fet enriquidor d'acabar-la amb èxit ha despertat en mi l'interès per dur a terme altres projectes personals.

7.2 Diferències del resultat amb la planificació

Arribats a aquest punt, són moltes les diferències amb la planificació inicial. Algunes per desconeixement de la plataforma de desenvolupament en general i d'Android, altres per ampliacions a petició del tutor en fases més avançades del procés o fins i tot per ampliacions improvisades a iniciativa pròpia.

7.2.1 Posicionament sobre Google Maps

La planificació inicial treballava únicament amb coordenades, sense cap mena de representació visual més enllà d'un llistat amb totes les posicions emmagatzemades. El tutor va incloure per primera vegada el concepte de “visió de mapa” i això m'ha dut a incloure la funcionalitat amb l'ús de l'API de Google Maps.

Aquesta funcionalitat ha sigut inclosa en l'activitat de desar posició manualment i l'edició d'una posició en concret. Per iniciativa pròpia he volgut anar més enllà i sota una improvisació no planificada no ha sigut gaire difícil fer movable la marca de posició sobre el mapa a més de reflectir-ho en coordenades en el formulari.

7.2.2 Versió mínima d'Android 2.3.3 en comptes de 2.2

Tal com explica l'apartat anterior, incloure Google Maps no estava previst. Durant el procés cercant informació[15] de la seva implementació descobreixo que forma part dels *Google Play Services* i que incloure-ho requereix de dos processos diferents si es desitja la compatibilitat per sota de la 2.3.3. Per no fer més complexe la implementació de funcionalitats addicionals vaig decidir descartar la compatibilitat prevista per la 2.2. La clau de la decisió va ser no tenir problemes amb el terminal de proves HTC Desire, el qual ja tenia modificat amb una versió no oficial 2.3.7.

7.2.3 Activitat per desar manualment la posició sense timeouts

La principal funció de l'aplicació des del primer dia de planificació era el *widget* que oferia el posicionament desatès en funció dels *timeouts*. La previsió passava per crear la funció de desar posició accessible des del menú principal mentre que el *widget* tan sols l'havia de cridar en segon pla. La falta d'experiència en planificació no em va deixar veure que la programació de localització i la del *widget* juntes eren massa grans per dur-les alhora. A més a més, el *widget* és més complexe que una simple crida a la funció. Així doncs, per provar exclusivament la part de localització vaig crear una activitat de prova manual que poc a poc vaig anar adaptant com a la funció de desar posició.

En un replantejament posterior de les funcionalitats del programa no tenia gaire sentit repetir la funció del *widget* però accessible dins el programa. Així va nàixer l'activitat de desar la posició amb engegada i parada manual i deixant de banda les limitacions dels *timeouts*. El resultat ha sigut positiu i ha quedat com una part que afegeix força valor al conjunt del programa.

7.2.4 Replantejament dels timeouts

La idea inicial en el plantejament era dedicar uns segons a buscar el cellID i obtenir la precisió baixa, uns segons al wifi per la precisió mitja i uns segons al GPS. El concepte en si mateix no era del tot correcte, un cop més per falta de coneixements:

- El posicionament en Android no funciona com si fos “vull una posició cellID ara mateix o diga'm que no t'has trobat”. Funciona activant uns *listeners* que descansen durant un temps especificat però que quan desperten tant et poden tornar cap com moltes posicions alhora i el més important, no quan la demanes sinó quan en tenen alguna per donar.
- El gestor de localització engloba el posicionament per cel·les de telefonia i el de xarxes wifi, especificat amb l'indicador *network_provider* així que sense poder-ne fer distinció el meu plantejament no era possible realitzar-lo.
- Els *timeouts* dedicats a cada precisió no els plantejava alhora. És a dir, els 10 segons de GPS eren després d'haver acabat i aturat el posicionament wifi. Finalment en la implementació s'engeguen els *listeners* en paral·lel de bon principi i cada temps es suma als dedicats al nivell de precisió anterior.

7.2.5 Canvis en el model de base de dades i la informació mostrada

Amb l'entrada de la nova funcionalitat de Google Maps calia guanyar el màxim d'espai possible en les activitats de desar i edició de posició per tal de veure el mapa tant gran com sigui possible. Per aquest motiu vaig decidir no incloure a la base de dades la informació de temps d'ubicació ni la precisió alhora que tampoc es mostra en cap formulari.

7.2.6 Altres funcionalitats addicionals afegides

El *widget* estava previst que fos el més semblant a un simple polsador que cridés al codi corresponent però la realitat ha sigut molt més complexe. De fet, vaig intentar evitar-ne la programació buscant la manera de realitzar la mateixa funció amb un accés directe. Al final crec que aquests obligatòriament han de cridar a activitats i aquestes són totes en primer pla (*foreground*) incompatible totalment amb l'objectiu d'executar-ho en segon pla (*background*). A pesar de no aconseguir-ho d'aquí va sorgir l'opció d'afegir l'accés directe del menú principal.

Gràcies a un exemple trobat per internet d'un *widget* amb unes funcionalitats que bé podrien adaptar-se a les meves necessitats, vaig descobrir la manera de recuperar informació i mostrar-la dins el propi *widget*. Ja fora de planificació havia d'aprofitar aquesta possibilitat i d'aquí va sortir la funció de veure les coordenades de l'últim posicionament. Semblava poca cosa i molt poc informadora/útil. Cercant entre les possibilitats de les llibreries d'Android vaig descobrir la classe *Geocoder* que m'ha servit per mostrar l'adreça de la posició. Ara sí sembla un *widget* de gran utilitat!

7.3 Dificultats destacables en la programació

Programar en una plataforma totalment desconeguda es ja de per si una dificultat constant. Hi ha certs punts on m'he trobat dificultats que mereixen especial atenció.

7.3.1 Aplicació del timeout i la lluita contra el ANR

En la programació de la funció que realitza el *widget* arriba un punt en que tens els *listeners* de xarxa i GPS engegats i segons els *timeouts* escollits per l'usuari s'ha d'aturar el procés i salvar la posició en un nivell de precisió o altra.

La primera dificultat és trobar la manera d'aplicar l'aturada en un moment precís. L'solució fàcil és controlar el temps dins la funció *onUpdate*, funció cridada cada cop que un *listener* rep una nova posició. Fer-ho així és senzill però molt poc professional. No tan sols es podria allargar molt l'aturada per sobre el temps establert sinó que podria, encara que improbable, no aturar-se mai.

Buscant una millor solució tan sols quedava l'alternativa del *thread*: un fil paral·lel que dormit fins esgotar-se el temps aturés el procés. Tot semblava funcionar correctament però sense cap mena d'explicació ni sortida d'error el *thread* tancava l'execució en un moment indeterminat.

En la cerca d'informació vaig sospitar del sistema ANR[16] (Application Not Responding). Android inclou aquesta funcionalitat que detecta i tanca els fils de treball bloquejats o sense activitat per evitar el bloqueig d'aplicacions o el consum de bateria per malfuncionament d'aquestes.

Els esforços per evitar l'ANR no funcionaven. M'aturava el procés si el dormia 10 segons, si el feia dormir 10 vegades durant un segon, si entre dormides realitzava alguna acció sense utilitat, etc.

Finalment, després de molt temps invertit, la solució va venir de la funció *AsyncTask* que inclou Android i que té una utilitat ampliada dels *thread* de Java.

7.3.2 Base de dades SQL i mostrar el llistat de posicions

Aquesta part va ser molt extensa per dos motius. En primer lloc per ser una de les primeres coses que vaig realitzar quan encara no havia agafat gaire pràctica i en segon lloc per la quantitat de conceptes nous que cal saber. Com crear la base de dades, esbrinar les classes i mètodes d'Android que treballen amb les dades, generar tots els mètodes que s'adaptin a l'aplicació, etc. Tanmateix no n'hi ha prou, també cal esbrinar com traslladar les dades al *layout*, crear menús contextuais, ... El bloc en si mateix són molts conceptes nous que gràcies a exemples trobats a internet va ser possible acabar-los realitzant.

7.4 Futures millores

Tot és millorable. A mida que s'avança van sorgint noves idees i funcionalitats que queden en una llista de coses per fer (si queda temps). A continuació, algunes millores pendents:

- **Refer-ho tot:**

Sí, pot semblar excessiu però és una sensació que creix com més aprop del final estàs.

He començat aquest projecte sense cap coneixement d'Android i ara en canvi he guanyat molta agilitat i qualitat de codi. Amb l'experiència obtinguda hi ha coses que de tornar-les a fer les hagués fet diferent.

N'he canviat moltes. En contínues revisions de codi he anat fent petites millores però les grans requereixen un recurs limitat, el temps.

- **Possibilitat de nombrar les posicions:**

En el disseny es va prendre la base de tota posició pel seu *DateTime*, és a dir, la data i l'hora de la seva presa. És correcte, però com a funcionalitat afegida estaria bé que l'usuari pogués afegir més informació sobre cada posició.

- **Accedir directament al detall de la posició des del *widget*:**

Finalment ha quedat com a informatiu de l'última posició de la base de dades sense possibilitat d'usar-ho com accés directe a aquesta.

- **El text justificat en la descripció de les opcions:**

El fet de que Android no tingui la funció de justificació completa en un *TextView* m'ha sorprès. Potser ha sigut inclòs en versions recents de la plataforma i tota la informació a internet fa referència al passat però la realitat es que visualment no queda bé i simular aquesta propietat és costós.

- **Llistar totes les posicions guardades en un mapa:**

És una alternativa visualment més atractiva que el format de llista implementat. No estava previst en fase de disseny i al final per falta de temps no s'ha pogut improvisar.

- **Esborrar totes les posicions de la base de dades alhora:**

Les úniques limitacions en el nombre de posicions guardades venen per la capacitat del dispositiu i/o la comoditat de treballar amb llistes molt llargues. Encara que guardar moltes posicions no crec que sigui un cas d'ús gaire habitual, una opció per esborrar-les totes en comptes d'una per una és una bona millora.

- **Disposició de les icones en el menú principal en forma de quadrícula:**

La idea de canviar la visualització del menú principal a icones grans la vaig tenir tant bon punt vaig començar la implementació. Van haver-hi dos motius pels quals no ho vaig fet. El primer era la inexperiència. Els *layouts* de l'aplicació va ser el primer que

vaig fer i no m'acabava de sortir com jo volia. El segon motiu és que l'esquema anomenat *grid layout* va ser implementada en versions 4.0 i superiors. Així que per qüestions de compatibilitat vaig pensar que no es podria fer i vaig implementar-ho en forma de llista com estava previst. Ara sí sé que es pot i sé com fer-ho. Un cop més, la falta de temps no m'ha permès canviar-ho.

- **Importació/exportació del llistat de posicions en XML**

Permetria intercanviar posicions, fer còpies de seguretat, ... No obstant no entra dins de l'objectiu del programa i per aquest ús existeixen molts altres programes més especialitzats.

8 Bibliografia

El format utilitzat per mostrar la bibliografia segueix l'esquema separat per comes:

Pàgines web:

[Índex]: Autor/Creador de la pàgina, Títol, Data d'accés, Enllaç URL

Bibliografia:

- 1: Wikipedia, HTC Desire, Setembre 2013, http://es.wikipedia.org/wiki/HTC_Desire
- 2: Richard Hult and Mikael Hallendal, Planner, 2014, <https://wiki.gnome.org/action/show/Apps/Planner?action=show&redirect=Planner>
- 3: Cay S. Horstmann & Alexandre de Pellegrin, Violet UML Editor, 2014, <http://violet.sourceforge.net/>
- 4: Tigris, ArgoUML, 2014, <http://argouml.tigris.org/>
- 5: The Gimp Team, Gimp, 2014, <http://www.gimp.org/>
- 6: Google Inc., Get the Android SDK, Setembre 2013, <https://developer.android.com/sdk/index.html>
- 7: Google Inc., Distribució de versions Android (1 d'abril), 2014, <http://developer.android.com/about/dashboards/index.html>
- 8: Google Inc., Iconography Android, 2014, <http://developer.android.com/design/style/iconography.html>
- 9: Wikipedia, Android, 2014, <http://ca.wikipedia.org/wiki/Android>
- 10: Piotr, Justifying block of text inside TextView in Android, 2014, <http://blog.wittchen.biz.pl/justifying-block-of-text-inside-textview-in-android/>
- 11: Wikipedia, HTML, 2014, http://ca.wikipedia.org/wiki/Hyper_Text_Markup_Language
- 12: Wikipedia, Extensible Markup Language, 2014, http://ca.wikipedia.org/wiki/Extensible_Markup_Language
- 13: Google Inc., Android Developer Design Downloads, 2014, <https://developer.android.com/design/downloads/index.html>
- 14: Google Inc., Getting Started (Google Maps Android API v2), 2014, <https://developers.google.com/maps/documentation/android/start>
- 15: Google Inc., Google Play Services SDK, 2014, <http://developer.android.com/google/play-services/setup.html>
- 16: Google Inc., ANR Application Not Responding, 2014, <http://developer.android.com/training/articles/perf-anr.html>



Tots els logotips utilitzats i marques registrades formen part dels seus respectius propietaris i han sigut utilitzats amb caràcter purament informatiu.

La resta del projecte està autoritzada sota llicència Creative Commons (by-nc-nd).