

# PFC — NARINAND — MEMÒRIA

*Aplicació Android per carregar tracks en format GPX i poder-ne seguir la pista*

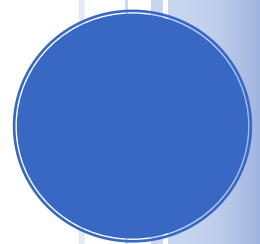
## Memòria

Memòria explicativa del projecte de desenvolupament d'una aplicació per al sistema operatiu Android.

Autor: Xavier Bach-Esteve Burch

Consultor: Jordi Ceballos VillachSiX

30/05/2011



# ÍNDIX DE CONTINGUTS

1	Introducció .....	6
1.1	Justificació i context del projecte .....	6
1.1.1	Estat actual del mercat.....	6
1.1.2	Desenvolupament per a Android .....	7
1.1.3	Android Market.....	7
1.1.4	Decisió del projecte.....	8
1.2	Descripció del projecte .....	9
1.3	Objectius.....	10
1.4	Característiques de l'aplicació .....	10
1.5	Planificació.....	12
1.5.1	Detall de les activitats .....	12
1.5.1.1	Anàlisi tècnic i planificació .....	12
1.5.1.2	Disseny tècnic.....	12
1.5.1.3	Desenvolupament aplicació.....	12
1.5.1.4	Documentació .....	13
1.5.2	Planificació temporal .....	13
1.6	Eines utilitzades .....	15
1.7	Productes obtinguts.....	16
2	Requeriments inicials .....	17
2.1	Escenari inicial .....	17
2.2	Usuaris finals .....	17
2.3	Requeriments funcionals.....	17
2.4	Requeriments no funcionals.....	18
3	Disseny .....	19
3.1	Aspectes a tenir en compte.....	19
3.2	Arquitectura .....	19
3.2.1	Arquitectura del sistema Android.....	19
3.2.2	Arquitectura de l'aplicació .....	20
3.2.2.1	El fitxer de manifest.....	20
3.2.2.2	Elements de l'aplicació <i>Narinand per Android</i> .....	21
3.2.2.3	Resources o recursos.....	21
3.2.2.3.1	Drawables .....	21
3.2.2.3.2	Layout .....	21
3.2.2.3.3	Menu.....	22
3.2.2.3.4	Raw.....	22
3.2.2.3.5	Values.....	22

3.2.2.3.6	Xml.....	22
3.2.2.4	Classe R.java.....	22
3.2.2.5	Resultat final .....	22
3.3	Disseny de classes .....	22
3.3.1	Paquet edu.uoc.narinad.gpx.....	23
3.3.1.1	Classe GpxReader.....	23
3.3.1.2	Classe GpxWriter .....	23
3.3.1.3	Classes de definició dels tipus utilitzats pels fitxers gpx .....	24
3.3.2	Paquet edu.uoc.narinand .....	28
3.3.2.1	Classe Narinand.....	28
3.3.2.2	Classe NarinandItemizedOverlay.....	29
3.3.2.3	Classe MyRouteOverlay.....	29
3.3.2.4	Classe LoadRoute.....	30
3.3.2.5	Classe ViewRoutes .....	30
3.3.2.6	Classe NewRoute.....	31
3.3.2.7	Classe Import.....	32
3.3.2.8	Classe FileBrowser .....	32
3.3.2.9	Classe DictionaryOpenHelper.....	32
3.3.2.10	Classe Utils .....	33
3.3.2.11	Classe GlobalData .....	34
3.3.2.12	Classe Settings .....	34
3.3.2.13	Classe MyLogger .....	35
3.4	Disseny de la persistència.....	35
3.4.1	Disseny de la base de dades .....	35
3.4.2	Disseny relacional.....	36
3.4.3	Diagrama de la base de dades.....	36
3.5	Llibreries a utilitzar .....	37
4	Implementació.....	39
4.1	Decisions d'implementació .....	39
4.2	Els fitxers gpx.....	39
4.3	Paquet edu.uoc.narinand.gpx.....	40
4.3.1	Definicions del tipus de dades GPX.....	40
4.3.2	Component GPXReader .....	42
4.3.3	Component GpxWriter.....	43
4.4	Paquet edu.uoc.narinad .....	44
4.4.1	AndroidManifest.xml.....	44
4.4.2	Dissenys de pantalla o <i>Layouts</i> .....	44
4.4.3	Menú de l'activitat <i>NewRoute</i> .....	46

4.4.4	Pantalla de preferències.....	46
4.4.5	Fitxers de recursos.....	47
4.4.5.1	Fitxer de recursos Strings.xml .....	48
4.4.5.2	Fitxer de recursos Arrays.xml .....	48
4.4.5.3	Fitxer de recursos Styles.xml.....	49
4.4.6	Activitat <i>Narinand</i> .....	50
4.4.6.1	Layout.....	50
4.4.6.2	Lògica de negoci.....	50
4.4.6.3	Captures de pantalla .....	51
4.4.7	Activitat <i>NewRoute</i> .....	52
4.4.7.1	Layout.....	52
4.4.7.2	Lògica de negoci.....	52
4.4.7.3	Captures de pantalla .....	54
4.4.8	Activitat <i>LoadRoute</i> .....	55
4.4.8.1	Layout.....	55
4.4.8.2	Lògica de negoci.....	55
4.4.8.3	Captures de pantalla .....	56
4.4.9	Activitat <i>FileBrowser</i> .....	57
4.4.9.1	Layout.....	57
4.4.9.2	Lògica de negoci.....	57
4.4.9.3	Captures de pantalla .....	57
4.4.10	Activitat <i>ViewRoutes</i> .....	58
4.4.10.1	Layout.....	58
4.4.10.2	Lògica de negoci.....	58
4.4.10.3	Captures de pantalla .....	59
4.4.11	Activitat <i>Preferences</i> .....	60
4.4.11.1	Layout.....	60
4.4.11.2	Lògica de negoci.....	60
4.4.11.3	Captures de pantalla .....	60
4.4.12	Altres elements .....	61
4.4.12.1	Classe <i>DictionaryOpenHelper</i> .....	61
4.4.12.2	Classe <i>MyRouteOverlay</i> .....	61
4.4.12.3	Classe <i>NarinandItemizedOverlay</i> .....	62
4.4.13	Altres captures de pantalla .....	63
5	Conclusions .....	65
6	Referències.....	66

# ÍNDIX D'IL·LUSTRACIONS

Il·lustració 1. Comparativa de vendes de sistemes operatius mòbils .....	6
Il·lustració 2. Lloc web oficial d'Android .....	7
Il·lustració 3. Logotip Android Market.....	8
Il·lustració 4. Exemple de ruta publicada a la web Wikiloc.....	10
Il·lustració 5. Distribució temporal de les tasques .....	13
Il·lustració 6. Diagrama de Gantt de la distribució de tasques .....	14
Il·lustració 7. Arquitectura del sistema Android .....	19
Il·lustració 8. Diagrama de classes del paquet edu.uoc.narinand.gpx.....	23
Il·lustració 9. Classe GpxReader .....	23
Il·lustració 10. Classe GpxWriter .....	23
Il·lustració 11. Classe PtsegType .....	24
Il·lustració 12. Classe ExtensionsType .....	24
Il·lustració 13. Classe RteType.....	24
Il·lustració 14. Classe GpxType.....	25
Il·lustració 16. Classe WptType .....	25
Il·lustració 15. Classe TrkType .....	25
Il·lustració 17. Classe EmailType.....	26
Il·lustració 18. Classe PtType .....	26
Il·lustració 19. Classe CopyrightType .....	26
Il·lustració 20. Classe TrksegType.....	26
Il·lustració 21. Classe MetadataType.....	26
Il·lustració 22. Classe PersonType .....	27
Il·lustració 23. Classe BoundsType .....	27
Il·lustració 24. Classe LinkType.....	27
Il·lustració 25. Diagrama de classes del paquet edu.uoc.narinand .....	28
Il·lustració 26. Classe Narinand.....	28
Il·lustració 27. Classe NarinandItemizedOverlay.....	29
Il·lustració 28. Classe MyRouteOverlay .....	29
Il·lustració 29. Classe LoadRoute .....	30
Il·lustració 30. Classe ViewRoutes .....	30
Il·lustració 31. Classe NewRoute.....	31
Il·lustració 32. Classe Import .....	32
Il·lustració 33. Classe FileBrowser.....	32
Il·lustració 34. Classe DictionaryOpenHelper.....	32
Il·lustració 35. Classe Utils .....	33
Il·lustració 36. Classe GlobalData.....	34
Il·lustració 37. Classe Settings .....	34
Il·lustració 38. Classe MyLogger.....	35
Il·lustració 39. Disseny relacional de la base de dades .....	36
Il·lustració 40. Diagrama de la base de dades .....	36
Il·lustració 41. Percentatges d'ús de les diverses versions d'Android .....	37
Il·lustració 42. Evolució històrica de les diverses versions d'Android .....	38
Il·lustració 43. Esquema principal d'un fitxer gpx .....	39
Il·lustració 44. Exemple de codi dels tipus de dades que formen un objecte gpx.....	41
Il·lustració 45. Mètode per llegir el tipus TrksegType d'un fitxer gpx.....	42
Il·lustració 46. Mètode per serialitzar el tipus LinkType a un fitxer gpx .....	43
Il·lustració 47. Fragment del fitxer AndroidManifest.xml .....	44
Il·lustració 48. Fragment del fitxer de layout de la pantalla principal .....	44
Il·lustració 49. Fragment del fitxer de layout de la pantalla de nova ruta.....	45

II·lustració 50. Fitxer routemenu.xml.....	46
II·lustració 51. Fragment del fitxer Settings.xml.....	47
II·lustració 52. Fragment del fitxer Strings.xml .....	48
II·lustració 53. Exemple de lectura d'un recurs.....	48
II·lustració 54. Fitxer Arrays.xml.....	49
II·lustració 55. Fitxer Styles.xml .....	49
II·lustració 56. Fragment de TableLayout del fitxer main.xml.....	50
II·lustració 57. Codi per mostrar un botó amb imatge a la vista principal .....	51
II·lustració 58. Pantalla principal .....	51
II·lustració 59. Fragment del fitxer newroute.xml .....	52
II·lustració 60. Registre de locationManager .....	52
II·lustració 61. Gestió de les capes del mapa.....	53
II·lustració 62. Pantalla NewRoute amb vista aèria .....	54
II·lustració 63. Pantalla NewRoute amb menú .....	54
II·lustració 64. Pantalla NewRoute. Ruta finalitzada.....	54
II·lustració 65. Pantalla NewRoute. Diàleg de guardar fitxer.....	54
II·lustració 66. Fragment del fitxer loadroute.xml.....	55
II·lustració 67. Pas de paràmetres entre activitats.....	56
II·lustració 68. Pantalla LoadRoute .....	56
II·lustració 69. Fitxer file_row.xml.....	57
II·lustració 70. Fragment de codi per carregar un ListAdapter .....	57
II·lustració 71. Pantalla del navegador de fitxers.....	57
II·lustració 72. Fitxer viewroutes.xml.....	58
II·lustració 73. Fitxer viewrouteselements.xml.....	58
II·lustració 74. Creació del menú contextual.....	59
II·lustració 75. Gestió d'events del menú contextual.....	59
II·lustració 76. Menú contextual .....	59
II·lustració 77. Fragment del fitxer Settings.xml.....	60
II·lustració 78. Codi font per carregar pantalla de preferències.....	60
II·lustració 79. Pantalla de preferències .....	60
II·lustració 80. Sentència SQL de creació de taula TRACKS.....	61
II·lustració 81. Mètode de consulta a la base de dades SQLite .....	61
II·lustració 82. Mètode per dibuixar una línia a un mapa .....	62
II·lustració 83. Mètode per afegir un ítem al mapa .....	62
II·lustració 84. Constructor de la classe NarinandItemizedOverlay .....	63
II·lustració 85. Avís d'allunyament de ruta.....	63
II·lustració 86. Definició ubicació de fitxers .....	63
II·lustració 87. Definició temps de captura .....	63
II·lustració 88. Informació de l'aplicació.....	64
II·lustració 89. Entrada de dades de la ruta per guardar.....	64

# 1 INTRODUCCIÓ

## 1.1 Justificació i context del projecte

### 1.1.1 Estat actual del mercat

Actualment, a causa de la generalització de l'ús de la banda ampla mòbil entre els usuaris finals, estan a l'ordre del dia els dispositius mòbils intel·ligents tàctils.

Existeixen principalment dos tipus de dispositius:

- *Smartphones*. Es tracta de telèfons mòbils intel·ligents que incorporen tot tipus de sensors –acceleròmetre, gps, càmera, pantalla tàctil...- i multitud d'utilitats i aplicacions –suite ofimàtica, agenda, contactes, navegació gps, jocs...-. Amb aquests telèfons intel·ligents s'ha trencat la barrera del que és un telèfon per passar a ser veritables ordinadors de butxaca.
- *Tablets*. Són dispositius una mica més grans que els *Smartphones*, però tot i això molt portables que deixen una mica de banda les tasques de la telefonia per centrar-se completament en la navegació web, agenda digital, navegació gps, centre multimèdia...

Aquests dos tipus de dispositius es comercialitzen amb diferents sistemes operatius. Els principals sistemes per a *Smartphones* són:

- ✓ Microsoft Windows Phone
- ✓ iOS d'Apple
- ✓ Android de Google
- ✓ Blackberry OS de RIM (Research In Motion)

Symbian Pel que fa a els dispositiu tipus *Tablet*, els més estesos són:

- ✓ iOS d'Apple
- ✓ Android de Google

Tot i que per *tablets* també existeix l'homòleg de Microsoft, avui en dia té una quota de mercat ínfima respecte als dos sistemes enumerats anteriorment. Blackberry i Symbian no disposen de sistema operatiu per a *Tablets*.

Així doncs, es pot dir que avui en dia, els gegants dels sistemes per a dispositius mòbils avançats –que inclouen suport per a *tablets* i *smartphones*- són iOS d'Apple i Android de Google.

<b>Company</b>	<b>2010 Units</b>	<b>2010 Market Share (%)</b>	<b>2009 Units</b>	<b>2009 Market Share (%)</b>
Symbian	111,576.7	37.6	80,878.3	46.9
Android	67,224.5	22.7	6,798.4	3.9
Research In Motion	47,451.6	16.0	34,346.6	19.9
iOS	46,598.3	15.7	24,889.7	14.4
Microsoft	12,378.2	4.2	15,031.0	8.7
Other Oss	11417.4	3.8	10432.1	6.1
<b>Total</b>	<b>296,646.6</b>	<b>100.0</b>	<b>172,376.1</b>	<b>100.0</b>

Source: Gartner (February 2011)

Il·lustració 1. Comparativa de vendes de sistemes operatius mòbils

La diferència entre aquests dos fabricants que dominen el mercat és que Android és un sistema operatiu lliure, mentre que iOS és propietari. Això fa que els experts prediguin un gran creixement del sistema operatiu Android passant a liderar el mercat mundial.

De fet, segons un estudi realitzat per la consultora *Gartner* el passat mes de febrer de 2011, durant l'any 2010, les vendes mundials de dispositius amb sistema operatiu Android van créixer un 888,8% respecte el 2009, passant a ocupar la segona posició en quota de mercat, darrere de *Symbian*. En tercera posició es troba *Blackberry* i en quarta *iOS*.

Així doncs, el sistema operatiu *Android* ha passat en un any d'un 3,9% de quota de mercat a un 22,7%, i les expectatives són que continuï aquest creixement durant els propers anys.

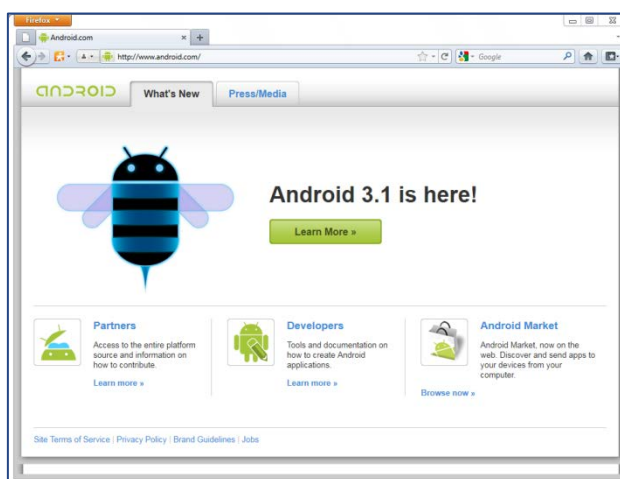
Les dades concretes d'aquest estudi es poden veure en la taula reflectida a la Il·lustració 1.

Aquest gran creixement de quota de mercat que ha tingut *Android* ha comportat que hagi crescut també de forma espectacular el nombre d'aplicacions –gratuïtes o de pagament– que es poden trobar al repositori principal d'aquest sistema, l'anomenat *Android Market*.

### 1.1.2 Desenvolupament per a Android

Mentre que a dia d'avui els desenvolupadors que volen dissenyar aplicacions pel sistema operatiu *iOS* han de pagar una quota, els que volen dedicar-se a desenvolupar aplicacions per a *Android* poden aconseguir el *SDK* i l'emulador de forma gratuïta. Aquest fet, juntament amb les xifres comentades en el punt anterior fan preveure que un gran nombre de professionals i amateurs del desenvolupament es llançaran cap a el sistema *Android*.

A més a la pàgina web oficial<sup>1</sup> d'aquest sistema operatiu hi ha molta i variada informació, tutorials, descàrregues... per ajudar a iniciar-se al desenvolupament d'aplicacions per a *Android*.



Il·lustració 2. Lloc web oficial d'Android

Aquestes facilitats fan que sigui molt fàcil iniciar-se en el desenvolupament i que sigui una opció atractiva per a molta gent que té inquietuds informàtiques o que vol fer negoci amb desenvolupaments de software.

### 1.1.3 Android Market

Des de la mateixa pàgina oficial d'*Android* es pot accedir al lloc web anomenat *Android Market*. Aquest lloc no és res més que un repositori d'aplicacions que es poden instal·lar en el sistema operatiu en qüestió.

Qualsevol persona que hagi desenvolupat una aplicació per a aquest sistema pot publicar-la en aquest repositori –seguint les normes exposades en la web de desenvolupament–.

Les aplicacions estan classificades en aquest lloc web per categories –aplicacions i jocs–, per subcategories –aplicacions de negoci, socials... o bé jocs d'arcade, puzzle...-. També es poden filtrar separant les gratuïtes de les de pagament, i per últim hi ha un cercador per fer un filtre de text lliure.

<sup>1</sup> <http://www.android.com>





Il·lustració 3. Logotip Android Market

D'aquesta manera es poden veure i escollir les aplicacions desitjades d'entre la multitud que hi ha publicades.

D'aquesta manera és molt senzill per als desenvolupadors poder posar al mercat –

ja sigui per fer negoci o per donar-se a conèixer gratuïtament- les seves pròpies aplicacions, que tenen visibilitat a nivell mundial.

#### 1.1.4 Decisió del projecte

En vistes de tots els punts exposats anteriorment i com a posseïdor d'un dispositiu amb sistema operatiu *Android*, es va optar per orientar el projecte final de carrera cap al desenvolupament d'una aplicació per a aquests dispositius.

Per tal de desenvolupar alguna aplicació que no fos una repetició d'alguna altra ja existent es va fer un sondeig a diversos tipus d'aplicacions que es poden trobar al *Market*. Mentre es realitzava aquest sondeig es va prendre consciència de fins a quin punt havia crescut ràpidament el nombre d'usuaris d'aquest sistema, ja que existien aplicacions –gratuïtes i/o de pagament- per a qualsevol tipus de tasca que es pot realitzar amb dispositius mòbils.

Un cop analitzades les diverses tipologies d'aplicacions existents en el repositori citat en el punt anterior, es va detectar que existien un conjunt d'eines que es podien utilitzar per registrar rutes mitjançant els receptors gps que incorporen els *Smartphones*. Aquestes aplicacions registren les coordenades mentre dura l'activitat i al final presenten el recorregut que s'ha fet sobre un mapa, permetent –algunes d'elles- exportar-lo a format GPX.

El format GPX<sup>2</sup> és una especificació de fitxers XML d'intercanvi entre aplicacions i webs, i que serveix per a representar captures de punts de gps. Es tracta d'un format obert i estàndard de facto utilitzat per moltes pàgines web i aplicacions. Aquest estàndard ha estat desenvolupat per a la companyia *Topografix*.

Una de les mancances que es va detectar a les aplicacions per a Android que treballen amb aquest format, i que estan disponibles a l'*Android Market* va ser que no n'hi ha cap que permetin carregar la ruta abans de fer-la per poder-ne tenir una idea abans de sortir a fer l'activitat.

És per això, que es va apostar per a fer una aplicació que a més de les funcionalitats que ja permeten moltes altres aplicacions, permetés tenir aquest valor afegit que és la guia de cap on tirar en moments de dubte o de mala senyalització –comuna en rutes d'alta muntanya-.

Per a aquesta aplicació es va decidir utilitzar el nom de *Narinand*, fent un joc de paraules entre l'expressió catalana "Anar-hi anant" i la paraula *Android* que fa referència a la plataforma sobre la qual corre l'aplicació.

---

<sup>2</sup> <http://www.topografix.com/gpx.asp>

## 1.2 Descripció del projecte

Així doncs, amb aquest projecte s'ha fet l'anàlisi, el disseny i el desenvolupament d'una aplicació per al sistema operatiu *Android* que permet aprofitar les diverses funcionalitats que ofereixen els dispositius de tipus *Smartphone* o *Tablet*, principalment la característica de geo-posicionament.

Aquesta aplicació fa ús dels receptors GPS incorporats als dispositius que hi ha al mercat per a ajudar als excursionistes –sobretot als d'alta muntanya- a seguir les rutes de senderisme, passejades...

Aquest desenvolupament s'ha realitzat utilitzant les eines disponibles a tal efecte. Es tracta d'un SDK que es pot descarregar gratuïtament<sup>3</sup> i també les API de les diverses versions d'aquest sistema i un accés a les referències per als desenvolupadors.

Les aplicacions per a *Android* es dissenyen utilitzant el llenguatge de programació *Java* per a la lògica de programa, XML per a la definició de les interfícies d'usuari o capa de presentació i les API descrites en el paràgraf anterior per interactuar amb el sistema.

L'aplicació desenvolupada fa ús dels diversos components que formen part d'un sistema *Android*.

- **Activitats (*Activities*)**. Cada *Activity* –o Activitat- és una pantalla única amb una interfície d'usuari. Tot i que una aplicació engloba una o diverses activitats cadascuna d'elles és independent. En el manifest de l'aplicació es declara quina és l'activitat principal, i que per tant es veu en engegar l'aplicació. Si els permisos són els adequats, una aplicació qualsevol pot obrir una activitat d'una altra.
- **Serveis (*Services*)**. Un *Service* –o Servei- es caracteritza per no tenir interfície d'usuari. Els serveis corren en *background* i estan pensats per dur a terme tasques amb períodes d'execució llargs. Els serveis poden ser iniciats des d'altres components com ara les Activitats. Per a aquest projecte no ha estat necessari implementar cap component d'aquest tipus.
- **Proveïdors de contingut (*Content Providers*)**. Els *Content Providers* –o Proveïdors de Continguts- s'utilitzen per accedir a dades persistides des de la mateixa o d'altres aplicacions –sempre i quan es tinguin els permisos adequats-. En aquest projecte s'utilitzen per a llegir i escriure fitxers gpx, fitxers de log i accés a la base de dades de l'aplicació.
- ***Intents***. Són missatges que corren dins del dispositiu per notificar a les aplicacions de diversos events, canvis d'estat del maquinari, dades entrants, events d'aplicació. Aquest tipus d'elements s'han utilitzat per enviar informació entre les diverses activitats de l'aplicació i per generar alertes de dispositiu.
- ***Broadcast Receivers***. Mitjançant aquest component es poden captar missatges de *Broadcast* del sistema, com pot ser un avís de bateria baixa. Aquests components no fan cap acció per si mateixos i no tenen interfície d'usuari, tot i que poden iniciar una Activitat en rebre un dels missatges. Una aplicació també pot generar missatges d'aquest tipus.

En el sistema Android les activitats no es tanquen mai per part de l'usuari. És el mateix sistema el que s'encarrega d'alliberar recursos quan li fan falta, mentrestant poden continuar obertes en segon pla. Aquest comportament ha fet que moltes vegades el rendiment de les bateries del dispositiu es vegi compromès, ja que es continuen consumint recursos fins i tot quan no s'utilitzen les activitats.

---

<sup>3</sup> <http://developer.android.com/index.html>

Un projecte Eclipse d'*Android* conté els següents elements:

- *AndroidManifest.xml*. On es descriu l'aplicació i els components que conté (activitats, serveis...)
- *Build.xml*. Es tracta de l'script *Ant* per compilar l'aplicació, generant el fitxer .apk que permet instal·lar-la al dispositiu.
- *Bin/*. Directori que conté l'aplicació quan és compilada.
- *Src/*. Directori que conté el codi font en *Java* de l'aplicació.
- *Res/*. Directori que conté els recursos (icones, interfícies d'usuari, localitzacions...)
- *Assets/*. Conté altres fitxers estàtics que es volen empaquetar amb l'aplicació per desplegar al dispositiu.

### 1.3 Objectius

Amb la realització d'aquest projecte es persegueixen una sèrie d'objectius:

- Posar en pràctica tots els coneixements adquirits durant la carrera d'Enginyeria Informàtica, tant pel que fa a la planificació, gestió i execució de projectes, com per al desenvolupament en *Java* i en l'ús de dispositius en xarxa i les seves interaccions.
- Iniciar-se en el desenvolupament per a sistemes operatius *Android*.
- Aprofundir en l'ús de diverses eines de desenvolupament: entorn de desenvolupament *Eclipse*, SDK d'*Android*.
- Iniciació a la programació utilitzant les llibreries API d'*Android* i de *Google Maps*.
- Aprendre el funcionament de l'emulador d'*Android* per a debugar i testejar aplicacions per a aquest sistema operatiu.
- Desenvolupar una aplicació que fins ara no és present al mercat i que pot ser vàlida i útil per a molta gent.

### 1.4 Característiques de l'aplicació

L'aplicació que es s'ha desenvolupat permet a un usuari poder registrar, mitjançant el receptor gps, les diverses coordenades de la posició on es troba. Això fa que mentre s'està fent la ruta i quan es finalitza es disposi d'un conjunt de coordenades que es poden representar sobre el mapa per tenir una idea del recorregut que s'ha fet.

Tal com ja s'ha comentat anteriorment, d'aplicacions que fan això ja n'hi ha diverses disponibles de forma gratuïta, i és per això que amb aquesta aplicació s'ha volgut donar un valor afegit anant una mica més enllà pel que fa a ajuda en ruta.

El valor afegit ve donat per la possibilitat que l'usuari pugui carregar un fitxer gpx creat anteriorment per algú altre amb qualsevol dels dispositius que permeten crear fitxers amb aquest format estàndard.

Tractant-se d'un format obert i estàndard es poden llegir fitxers publicats a qualsevol lloc i amb qualsevol aplicació. Així doncs, durant tot el trajecte es pot visualitzar el

The screenshot shows the Wikiloc website interface. At the top, there are navigation links: 'wikiloc', 'Pujar rutes', 'Mapa del Món', and 'Cercar'. The main content area features a trail titled 'Aneto'. It includes a satellite map with an orange trail line and a green location pin. Below the map is an elevation profile graph. The trail statistics are as follows:

- Distància recorreguda:** 16,49 quilòmetres
- Altitud mín:** 1.803 metres, **màx:** 3.402 metres
- Desnivell acum. pujant:** 1.557 metres, **baixant:** 1.573 metres
- Grau de dificultat:** Dificil (indicated by a red dot icon)

Additional information includes the location 'aprop de es Bòrdes, Catalunya (Espanya) per highlander\_2498' and a link to 'Veure a Google Earth'. At the bottom, there are five small thumbnail images showing different views of the trail.

Il·lustració 4. Exemple de ruta publicada a la web Wikiloc

mapa, i a la vegada la ruta feta per la persona que ha compartit el fitxer gpx. D'aquesta manera, en cas que es desviï de la ruta marcada el dispositiu emet un avís, visual i configurable com a sonor i/o de vibració, evitant així anar per camins equivocats en casos de mala senyalització.

Al mateix temps, mentre es va fent la ruta es van guardant els punts per on es passen, i així al final es té la representació del camí fet sobreposant-la a la ruta carregada originalment, i la possibilitat d'exportar-la a gpx per a poder-la compartir amb altres excursionistes, tant si utilitzen aquesta aplicació com si no.

Una característica que es volia afegir com a tret diferencial era la opció de descarregar el tros de cartografia que fa referència al lloc per on ha de transcórrer la ruta per a tenir-lo emmagatzemat al dispositiu i no haver d'estar pendent de si hi ha cobertura 3G o no en els diversos punts de la ruta. Finalment, però, s'ha hagut de descartar aquesta funcionalitat ja que les API de Google Maps actualment no ho permeten.

S'ha valorat la opció d'utilitzar unes llibreries de tercers que sí que ho permetin, però s'ha descartat la idea pel fet que utilitzen mapes del projecte *OpenStreetMap* que, en tractar-se de un projecte col·laboratiu on la gent ha de dibuixar els mapes de manera desinteressada, no té cobertura de mapes complets a gran part del territori, sobretot a alta muntanya.

Com a resum es pot dir que amb aquest projecte s'ha implementat una aplicació d'ajuda a la navegació per senders per a tots aquells excursionistes que alguna vegada han pres el camí equivocat per manca de senyalització, que simplement volen tenir la seguretat de no perdre's en cap moment o bé volen fer la ruta exacta que ja ha fet alguna altra persona.

## 1.5 Planificació

En aquest apartat es presenta el detall de les activitats que s'han realitzat al llarg d'aquest projecte amb la seva temporització.

### 1.5.1 Detall de les activitats

#### 1.5.1.1 Anàlisi tècnic i planificació

En la primera etapa del projecte calia definir quines activitats calien fer i en quant de temps s'havien de dur a terme.

Activitat	Descripció
Selecció del projecte	Decisió del projecte que es volia dur a terme i comunicació al consultor.
Definició del pla de projecte	Definició de la temporització de les tasques que s'havien de dur a terme per a realitzar el projecte.
Instal·lació i configuració entorn de desenvolupament	Descàrrega, instal·lació i configuració dels diversos components necessaris per a desenvolupar el projecte.
Entorn de desenvolupament a punt	Es disposa d'un ordinador amb totes les eines necessàries per a posar-se a treballar.
Entrega PAC1	Entrega del document explicatiu d'aquesta fase d'anàlisi tècnic i planificació.

#### 1.5.1.2 Disseny tècnic

En aquesta segona fase es tractava de fer el disseny de tots els aspectes que tenen a veure amb l'aplicació, i la resolució de possibles problemes o riscos que s'haguessin produït.

Activitat	Descripció
Disseny de l'arquitectura	Definició de l'arquitectura de l'aplicació que es desenvolupa.
Disseny de classes	Definició de les classes que donen les funcionalitats necessàries per al bon funcionament de l'aplicació.
Disseny de la persistència	Definició de l'estructura de la base de dades que emmagatzema la informació generada per l'aplicació.
Resolució de problemes tecnològics	Estudi i resolució de problemes sorgits un cop analitzats tots els punts de la implementació.
Entrega PAC2	Entrega del document explicatiu d'aquesta fase de disseny tècnic.

#### 1.5.1.3 Desenvolupament aplicació

Aquesta tercera fase és la que conté el gruix més important de feina, ja que és on s'ha realitzat el desenvolupament i test de l'aplicació que ha resultat del projecte.

Activitat	Descripció
Anàlisi i prototip	Anàlisi i prototipatge del conjunt de l'aplicació segons els diversos dissenys realitzats.
Desenvolupament	Fase de desenvolupament de l'aplicació. Generació de codi per a dotar de funcionalitat l'aplicació.
Proves funcionals	Proves realitzades amb emulador i amb dispositiu real per tal de validar el funcionament de l'aplicació.
Entrega PAC3	Entrega del document explicatiu d'aquesta fase de desenvolupament.

#### 1.5.1.4 Documentació

Darrera fase del projecte on es generen els documents explicatius de la feina realitzada i de les tecnologies utilitzades.

Activitat	Descripció
Redacció memòria	Període en què es redacta aquest document explicatiu de tots els aspectes del projecte.
Gravació presentació virtual	Gravació d'un vídeo de presentació del projecte per tal de poder-lo mostrar al tribunal qualificador.
Entrega final	Entrega de la memòria i la presentació virtual.

#### 1.5.2 Planificació temporal

En la Il·lustració 5 es pot veure la distribució temporal de totes les tasques esmentades en l'apartat anterior.

	Nombre de tarea	Comienzo	Duración	Fin
1	<input type="checkbox"/> <b>Anàlisi tècnic i planificació</b>	<b>mar 01/03/11</b>	<b>10 días?</b>	<b>lun 14/03/11</b>
2	Selecció del projecte	mar 01/03/11	1 día?	mar 01/03/11
3	Definició del pla de projecte	jue 03/03/11	8 días	lun 14/03/11
4	Instal·lació i configuració entorn de desenvolupament	lun 07/03/11	2 días	mar 08/03/11
5	Entorn de desenvolupament a punt	mar 08/03/11	0 días	mar 08/03/11
6	Entrega PAC1	lun 14/03/11	0 días	lun 14/03/11
7	<input type="checkbox"/> <b>Disseny tècnic</b>	<b>mar 15/03/11</b>	<b>15 días?</b>	<b>lun 04/04/11</b>
8	Disseny de l'arquitectura	mar 15/03/11	5 días	lun 21/03/11
9	Disseny de classes	mar 22/03/11	3 días	jue 24/03/11
10	Disseny de la persistència	vie 25/03/11	2 días	lun 28/03/11
11	Resolució de problemes tecnològics	mar 15/03/11	6 días	mar 22/03/11
12	Entrega PAC2	lun 04/04/11	1 día?	lun 04/04/11
13	<input type="checkbox"/> <b>Desenvolupament aplicació</b>	<b>mar 05/04/11</b>	<b>25 días?</b>	<b>lun 09/05/11</b>
14	Anàlisi i prototip	mar 05/04/11	3 días?	jue 07/04/11
15	Desenvolupament	vie 08/04/11	18 días	mar 03/05/11
16	Proves funcionals	mié 04/05/11	4 días	lun 09/05/11
17	Entrega PAC3	lun 09/05/11	0 días	lun 09/05/11
18	<input type="checkbox"/> <b>Documentació</b>	<b>mar 10/05/11</b>	<b>15 días?</b>	<b>lun 30/05/11</b>
19	Redacció memòria	mar 10/05/11	9 días	vie 20/05/11
20	Gravació presentació virtual	lun 23/05/11	5 días	vie 27/05/11
21	Entrega final	lun 30/05/11	1 día?	lun 30/05/11

Il·lustració 5. Distribució temporal de les tasques





## 1.6 Eines utilitzades

Per a la realització d'un projecte de desenvolupament d'un producte informàtic sempre són necessàries diverses eines.

En aquest cas, en tractar-se d'un producte per a un sistema operatiu obert s'han pogut trobar totes les eines de programari necessàries per al desenvolupament de forma gratuïta, sense haver d'utilitzar cap llicència comercial. En el cas de la documentació i presentació sí que s'han utilitzat programes comercials llicenciats.

Pel que fa al maquinari, s'han utilitzat els dispositius que es posseïen en el moment determinat del projecte.

Tot seguit es detallen tots els productes, tant de programari com de maquinari, que han intervingut en aquest desenvolupament:

- **Telèfon intel·ligent amb sistema operatiu *Android*** (en aquest cas les proves s'han realitzat amb un HTC Desire HD). Aquest terminal utilitza la versió 2.2 (*FroYo*) d'*Android*. El que s'ha aconseguit amb aquest dispositiu és poder realitzar proves en un entorn real, ja que tot i que es disposa de l'emulador, amb aquest no es poden reproduir totes les condicions.
- **Ordinador personal amb Windows 7** per a realitzar el desenvolupament. Ordinador estàndard d'usuari amb sistema operatiu de Microsoft –és de pagament, però ja es disposava d'aquest sistema abans de posar-se a treballar amb el projecte-.
- **IDE de desenvolupament Eclipse** amb les integracions corresponents amb l'SDK d'*Android*. Aquest entorn de desenvolupament és molt conegut per a diversos programadors, i destaca per ser gratuït i per adaptar-se molt bé als desenvolupaments amb *Java*.
- **SDK d'*Android*** per a la versió 2.2 (nivell d'API 8) i amb les API 8 de *Google*. El conjunt de llibreries per a desenvolupar aplicacions *Android* és indispensable per a dur a terme el projecte. S'ha decidit utilitzar la versió 8 d'API ja que és la que correspon a la versió 2.2 (*FroYo*) d'*Android* i per tant equival a la versió del dispositiu real. Com que el projecte treballa amb les llibreries de *Google Maps*, també ha estat necessari utilitzar les API del mateix nivell que fan referència a les interaccions amb els mapes de *Google*.
- **Emulador d'*Android*** inclòs en l'SDK. Eina indispensable per a poder realitzar proves mentre s'està desenvolupant l'aplicació, i així poder debugar el codi a la recerca d'errors. L'emulador permet reproduir diverses situacions que es poden trobar en dispositius reals –pèrdua de connexió, entrada de trucades, gir de pantalla...- i s'integra completament amb l'entorn de desenvolupament *Eclipse*.
- **Java JDK versió 1.6.0\_22**. Com que el llenguatge de programació per a dispositius *Android* és el *Java*, és imprescindible tenir instal·lat el kit de desenvolupament de *Java*. Per a aquest projecte s'ha utilitzat la versió 1.6.0\_22, la més nova en el moment d'iniciar el desenvolupament.
- **Editor d'imatges *GIMP***, utilitzat per a la creació i edició de les diverses icones que utilitza l'aplicació.
- **Dissenyador de diagrames de classes *Astah Community***, utilitzat en les tasques de disseny funcional de l'aplicació.
- **Microsoft Project**, per a dissenyar el diagrama de Gantt de la temporització del projecte. Aquest programa és amb llicència comercial, però s'ha utilitzat la que es va obtenir per a realitzar l'assignatura MGPI d'aquesta carrera.



- **Microsoft Word i PowerPoint**, per a la redacció dels informes i memòria, i la construcció de la presentació. També és de llicència comercial, tot i que si no s'hagués disposat d'aquesta es podria haver utilitzat *OpenOffice*.
- **Camtasia**, en la seva versió de prova per a la realització del vídeo de la presentació del projecte.

## 1.7 Productes obtinguts

Al llarg de tot el projecte s'han generat els següents productes per a ser lliurats:

- Document descriptiu del projecte, on es feia la proposta de projecte al consultor.
- Pla de treball, que descrivia les accions que es volien dur a terme i la temporització per a fer-les.
- Disseny tècnic, on es descrivien els aspectes tècnics del producte: arquitectura, persistència, plataforma...
- Aplicació *Narinand*, amb:
  - Instal·lable de l'aplicació.
  - Codi font degudament comentat.
  - Informe del desenvolupament.
  - Manual d'instal·lació, tant de l'emulador, com de l'aplicació.
- Memòria del projecte i vídeo amb la presentació d'aquest.

## 2 REQUERIMENTS INICIALS

En el moment de decidir la temàtica sobre la que tractaria aquest projecte i els resultats que es volien aconseguir es van tenir en compte tots els aspectes que s'han explicat al primer apartat d'aquesta memòria. Un cop presa la decisió d'introduir-se a la tecnologia de desenvolupament per a sistemes *Android*, i explorat el mercat d'aplicacions d'aquest sistema es va prendre la determinació de dissenyar una aplicació que fes ús de les característiques de mapes i gps.

### 2.1 Escenari inicial

Partint de la realitat que gairebé tots els tipus d'aplicacions ja existeixen al mercat de l'*Android*, es va decidir fer alguna cosa que encara que ja existís, tingués algun valor afegit que li donés validesa.

Així doncs, partint del fet que al mercat hi ha diverses aplicacions que permeten gravar rutes fent ús de les característiques gps dels dispositius actuals i els mapes de *Google*, i que a més permeten compartir-les de diverses maneres (kml, gpx...), es va pensar en donar suport al sentit contrari, que és carregar fitxers.

D'aquesta manera, a més de poder gravar la ruta que s'està duent a terme, l'aplicació *Narinand* permet carregar rutes fetes anteriorment, o compartides per a altres persones.

Això dóna la opció de veure gràficament al dispositiu el camí que cal seguir abans i durant l'execució de la ruta. Aquest fet permet obtenir informació molt valuosa per a l'usuari, com ara veure si queda molt per acabar i assegurar-se que s'està seguint el camí correcte, i per tant evitar equivocar-se en la direcció a seguir.

### 2.2 Usuaris finals

L'aplicació està especialment pensada per a excursionistes que realitzen caminades per senders. Ajuda sobretot en senders d'alta muntanya que a vegades no estan tan ben indicats com seria desitjable, sobretot en època hivernal i primavera, quan la neu tapa o esborra els recorreguts, i no sempre hi ha senyalitzacions quan es necessiten.

Això no treu que es pugui utilitzar per tot tipus de caminades per tot arreu, fins i tot urbanes. De la mateixa manera es pot utilitzar per a altres finalitats com rutes amb bicicleta..., configurant els paràmetres de captura de manera correcta.

Sigui quina sigui la tipologia d'usuari, les opcions del programa són les mateixes per a tots ells, amb accés des de la configuració dels paràmetres, fins a la gravació de rutes, passant per la càrrega de fitxers gpx.

### 2.3 Requeriments funcionals

L'aplicació *Narinand* ha de complir els següents requeriments funcionals:

- **Nova ruta.** Ha de permetre iniciar una ruta, mostrant en tot moment el mapa de la zona on es troba –vista topogràfica o de satèl·lit-, indicant en tot moment la posició exacta i el camí recorregut.
- **Gestió de rutes.** S'han de poder tractar les diverses rutes que es poden haver emmagatzemat al dispositiu, permetent carregar-les al mapa, exportar-les a format gpx o esborrar-les.
- **Càrrega de fitxers.** S'ha de poder navegar pel sistema de fitxers per escollir i carregar fitxers en format gpx, per posteriorment, visualitzar-los al mapa.
- **Informació d'allunyament.** Ha de ser possible rebre notificacions, visuals, de vibració i sonores que indiquin que la posició actual està massa allunyada de la posició més propera de la ruta carregada.

- **Configuració dels paràmetres.** L'usuari ha de poder configurar diversos paràmetres necessaris pel funcionament de l'aplicació: temps de captura, distància de captura, directoris de treball, distància d'allunyament màxima...

## 2.4 Requeriments no funcionals

En aquest cas es tracten els requeriments d'interfície d'usuari que han de complir els següents aspectes:

- Ha de funcionar per a qualsevol dispositiu que tingui sistema operatiu *Android*, a partir de la versió 2.2.
- Ha de poder-se visualitzar correctament el contingut independentment de la mida de la pantalla del dispositiu.
- Ha de poder-se visualitzar correctament, tant per pantalla vertical, com apaïxada, adaptant la posició dels objectes.
- Ha de permetre introduir dades mitjançant l'ús del teclat físic –si el dispositiu en disposa- o del teclat virtual, que ha d'aparèixer en el moment en què s'han d'introduir dades.

## 3 DISSENY

### 3.1 Aspectes a tenir en compte

En aquesta part del document dedicada al disseny, es pretén donar una idea del resultat final d'aquest projecte un cop analitzats els requeriments i feta l'anàlisi funcional de l'aplicació.

En els següents punts es detalla l'arquitectura utilitzada, tant pel sistema operatiu *Android*, com per l'aplicació *Narinand* que s'ha desenvolupat. Així mateix es presenten els diagrames de classes que s'han definit per a dur a terme les funcionalitats demanades en els requisits. També es fa un disseny de la petita base de dades que utilitza l'aplicació per persistir les dades que genera, i finalment es descriuen les llibreries utilitzades, i les decisions d'implementació que s'han pres.

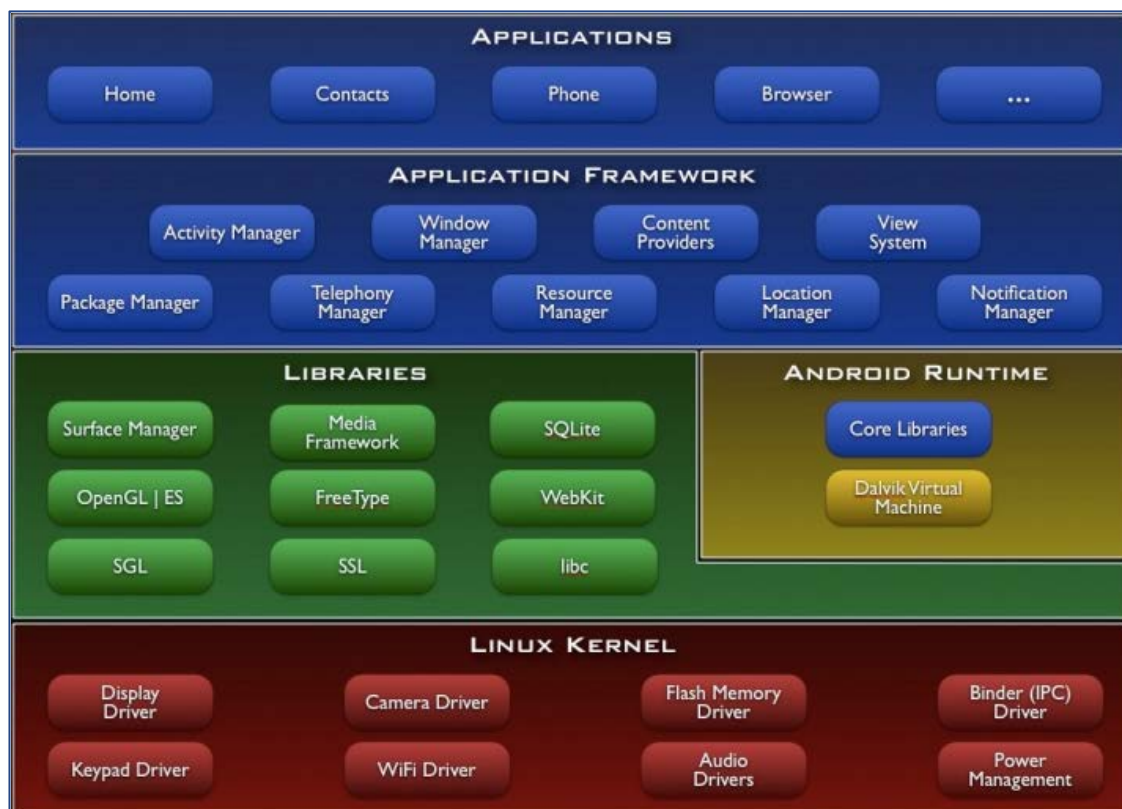
Amb tot això es pretén mostrar una imatge acurada del disseny funcional i de l'arquitectura d'una aplicació d'*Android*.

### 3.2 Arquitectura

El sistema operatiu Android està especialment dissenyat per a dispositius mòbils tàctils. Les aplicacions per a aquest sistema corren sobre un *kernel*<sup>4</sup> de Linux amb una màquina virtual que les executa, i que s'explica tot seguit amb més detall.

#### 3.2.1 Arquitectura del sistema Android

El sistema operatiu *Android* està basat en un kernel versió 2.6 de Linux per a dur a terme tasques com la seguretat, gestió de processos, gestió de memòria...



Il·lustració 7. Arquitectura del sistema Android

<sup>4</sup> Paraula amb la qual s'anomena el nucli del sistema operatiu.

Sobre aquest *kernel* hi ha diverses eines, serveis i llibreries per a donar facilitats als desenvolupadors, els quals accedeixen a aquests serveis mitjançant el *framework* d'aplicacions que incorpora.

Al mateix nivell es troba el *Runtime*, el qual conté un conjunt de llibreries que donen gairebé la mateixa funcionalitat que les llibreries de *Java*. Aquí també es troba la màquina virtual *Dalvik*. Aquesta màquina virtual és la que s'encarrega d'executar les aplicacions que estan empaquetades amb fitxers “.apk”.

Tota aplicació Android corre sobre el seu propi procés, que utilitza la seva pròpia instància de la màquina virtual. *Dalvik* ha estat dissenyat per a poder tenir diverses instàncies obertes de manera eficient.

Per sobre de les llibreries i del *runtime* s'hi troba el *framework* d'aplicació. Gràcies a aquest *framework*, els desenvolupadors poden treure profit del hardware de l'equip, accedir a informació de localització, afegir notificacions a la barra d'estat...

Això és perquè els desenvolupadors tenen accés a la mateixa API utilitzada per les aplicacions del nucli. D'aquesta manera s'aconsegueix simplificar la reutilització de components, ja que qualsevol aplicació pot publicar les seves capacitats i qualsevol altra en pot fer ús.

Per últim, a la capa superior s'hi troben les aplicacions que estan desenvolupades amb el llenguatge de programació Java. Algunes són comunes a tots els equips que tenen *Android* com el telèfon, els contactes el calendari..., i d'altres que s'instal·len per l'usuari com ara la que s'ha obtingut de la realització d'aquest projecte.

## 3.2.2 Arquitectura de l'aplicació

### 3.2.2.1 El fitxer de manifest

Tota aplicació ha de contenir un fitxer xml anomenat *AndroidManifest.xml*.

Aquest fitxer conté:

- Definició de quins permisos necessitarà l'aplicació per a funcionar correctament. Per a aquest projecte són necessaris el següents permisos:
  - Accés al dispositiu GPS per a poder saber la posició actual.
  - Accés a internet per a poder descarregar els mapes per mostrar.
  - Accés a la vibració del dispositiu per a poder fer-lo vibrar quan es mostra l'alerta d'allunyament de ruta.
  - Permís d'escriptura a l'emmagatzemament extern per a poder escriure i llegir els fitxers gpx, els de log i la base de dades a la memòria externa del dispositiu.
- Definició dels permisos que les activitats o serveis de l'aplicació hauran de demanar a altres aplicacions que els vulguin utilitzar. En aquest cas no é necessària la interacció amb altres activitats o serveis.
- Versió del SDK que s'utilitzarà. En aquest cas la versió 8.
- Element per descriure l'aplicació. Aquest element contindrà els diversos elements que estaran disponibles –serveis, activitats...-, cadascun descrivint el seu nom i comportament. En aquest cas conté la llista d'activitats que formen part de l'aplicació i que es descriuen el següent apartat.

### 3.2.2.2 Elements de l'aplicació *Narinand per Android*

L'aplicació desenvolupada en aquest projecte fa ús de la majoria de components característics d'*Android*.

- **Activitats (*Activities*)**. L'aplicació *Narinand* està composta per 6 activitats, declarades totes elles al fitxer de manifest de l'aplicació<sup>5</sup>:
  - *Narinand*. Pantalla que ofereix el menú principal de l'aplicació i que s'obre tot just quan es llança el programa.
  - *NewRoute*. Pantalla des d'on es visualitza el mapa i les rutes carregades i que s'estan fent actualment.
  - *LoadRoute*. Pantalla des d'on es pot cercar i carregar un fitxer gpx amb una ruta.
  - *ViewRoutes*. Pantalla des d'on es pot interactuar amb les rutes que s'han guardat al dispositiu, permetent exportar-les a gpx, carregar-les o esborrar-les.
  - *FileBrowser*. Pantalla des d'on es pot navegar pel sistema de fitxers per tal d'anar a cercar un fitxer gpx per carregar.
  - *Settings*. Pantalla des d'on es poden canviar els ajustos del programa.
- **Proveïdors de contingut (*Content Providers*)**. Aquesta aplicació n'utilitza per:
  - Llegir i escriure fitxers gpx.
  - Escriure fitxers de log.
  - Accedir a la base de dades de l'aplicació.
- ***Intents***. En aquesta aplicació s'utilitzen per a diverses tasques com ara:
  - Generar avisos al dispositiu indicant que s'ha allunyat de la ruta marcada.
  - Passar dades entre diverses activitats, com la ruta del fitxer a obrir, informació del diàleg a mostrar...

### 3.2.2.3 Resources o recursos

Un altre punt important de les aplicacions Android són els recursos. Un projecte d'*Android* conté una carpeta anomenada "res" d'on en pengen un conjunt de subcarpetes on es defineix tot el que fa referència a la capa de presentació, des de les imatges fins a les pantalles i textos, passant pels menús i efectes de so.

#### 3.2.2.3.1 *Drawables*

Es tracta d'un conjunt de tres carpetes:

- "drawable-hdpi" → Resolució de 72x72
- "drawable-mdpi" → Resolució de 48x48
- "drawable-ldpi" → Resolució de 36x36

Aquí s'hi col·loquen totes les imatges i icones que utilitza l'aplicació. En cadascuna d'aquestes carpetes s'hi guarden en una resolució diferent, per tal que es visualitzin correctament a diversos tipus de pantalla independentment de la seva mida, adaptant-se perfectament als diversos tipus de dispositius mòbils.

#### 3.2.2.3.2 *Layout*

En aquesta carpeta s'hi defineixen mitjançant fitxers XML, les pantalles que s'han de mostrar. Tot i que les pantalles també es poden definir per codi en temps d'execució, és més recomanable utilitzar el sistema XML, ja que dóna independència a la capa de presentació i permet moure elements de manera molt ràpida i senzilla.

---

<sup>5</sup> Veure apartat 3.2.2.1 referent a aquest fitxer

### 3.2.2.3.3 *Menu*

En aquesta carpeta s'hi poden definir, també mitjançant fitxers XML, els elements que formen part dels menús que poden aparèixer a l'aplicació quan es prem la tecla "Menu" del dispositiu. De la mateixa manera que les pantalles, també es poden definir els menús dinàmicament mitjançant el codi font.

### 3.2.2.3.4 *Raw*

Aquesta carpeta conté fitxers binaris que formen part de l'aplicació. En el cas del projecte actual s'ha utilitzat per emmagatzemar el fitxer que conté el so que es reproduïx quan es genera un avís.

### 3.2.2.3.5 *Values*

En aquesta carpeta s'hi emmagatzemen diversos fitxers XML, tots ells de recursos. Per aquesta aplicació s'hi ha posat tres fitxers:

- Definició de les diverses opcions dels desplegable del menú de configuració dels paràmetres de l'aplicació.
- Definició dels textos a mostrar per pantalla. Molt útil per a poder fàcilment crear localitzacions per a diversos idiomes sense necessitat de tocar el codi font.
- Definició dels estils que s'apliquen a diversos elements de l'aplicació.

### 3.2.2.3.6 *Xml*

En aquesta carpeta s'hi defineixen, mitjançant fitxers XML les opcions que ha de tenir el menú de configuracions. D'aquesta manera, mitjançant aquest fitxer, es crea automàticament la pantalla amb la llista d'ajustos.

### 3.2.2.4 *Classe R.java*

Una particularitat dels projectes per a *Android* és la classe *R.java*.

Aquesta classe es genera automàticament –si s'utilitza Eclipse per a desenvolupar- i conté una referència a cadascun dels recursos que s'han definit. És a dir, crea una manera fàcil i ràpida d'accedir als recursos, classificant-los per tipus, mitjançant el nom amb el que s'han definit.

### 3.2.2.5 *Resultat final*

Un cop desenvolupada l'aplicació el resultat és un fitxer amb extensió *.apk* que conté tot el que s'ha definit, tant de components, com de recursos. Aquest és el tipus de fitxer que s'utilitza per instal·lar aplicacions al sistema *Android* i que s'executen a la màquina virtual Dalvik.

## 3.3 Disseny de classes

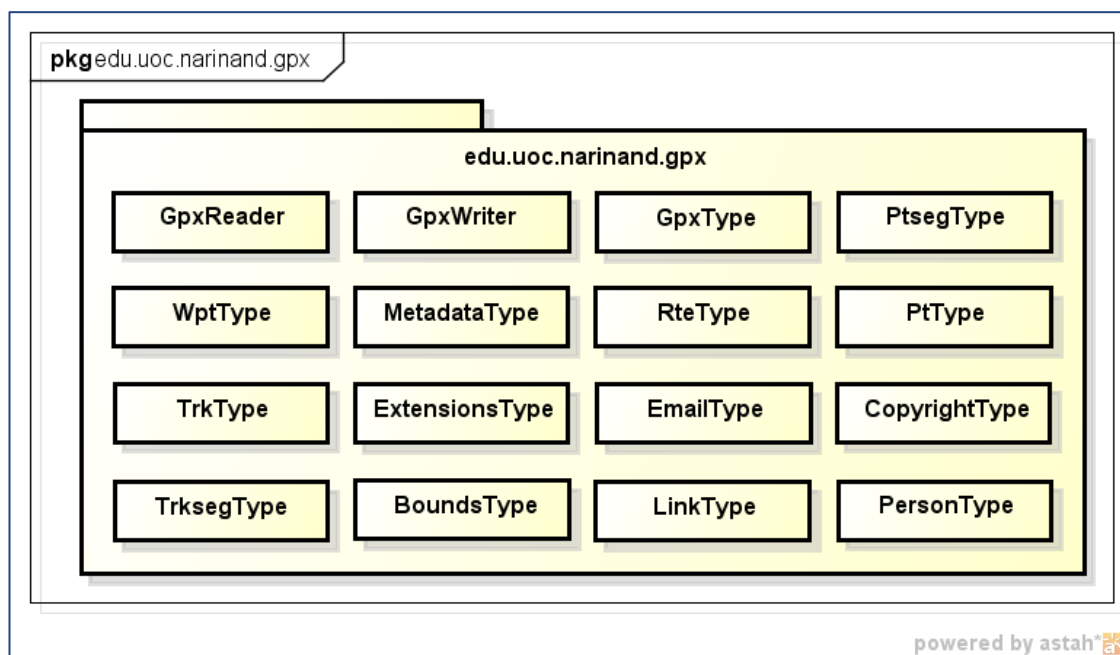
El projecte s'ha dividit en dos *packages* o paquets ben diferenciats. Un d'ells és el que conté totes les classes que fan referència a la gestió dels fitxers *gpx*. Aquest paquet conté totes les classes que defineixen els diversos tipus de dades que té un fitxer d'aquest tipus, i les que s'encarreguen de llegir i gravar aquests fitxers.

Per altra banda, es disposa del paquet on hi ha totes les classes que gestionen l'aplicació *Android* i els seus events. Algunes classes d'aquest paquet utilitzen classes de l'altre per a poder realitzar les operacions de lectura i escriptura de fitxers *gpx*.



### 3.3.1 Paquet edu.uoc.narinand.gpx

Aquest paquet consta d'un total de 16 classes, 14 de les quals amb els diversos tipus de dades que componen els fitxers gpx i les dues restants per a gestionar aquests fitxers – opcions de carregar i gravar-.



Il·lustració 8. Diagrama de classes del paquet edu.uoc.narinand.gpx

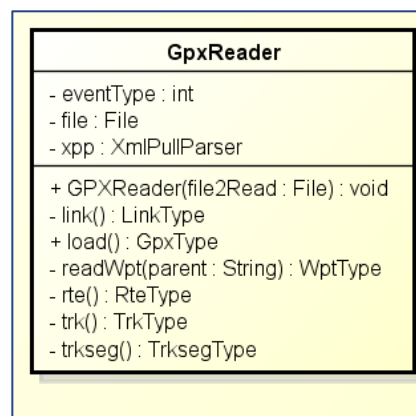
#### 3.3.1.1 Classe GpxReader

Classe encarregada de llegir un fitxer d'intercanvi en format gpx.

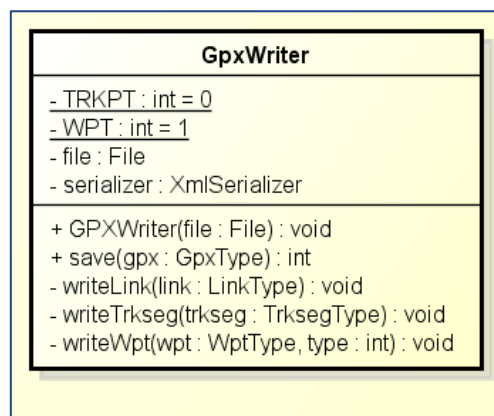
En el moment d'instanciar-la se li passa un objecte de tipus *File* que conté el fitxer gpx que es vol llegir.

En el moment en què es crida el mètode públic *load*, es llegeix el fitxer mitjançant un parsejador de fitxers xml i es retorna un objecte de tipus *GpxType* amb les dades contingudes en el fitxer.

#### 3.3.1.2 Classe GpxWriter



Il·lustració 9. Classe GpxReader



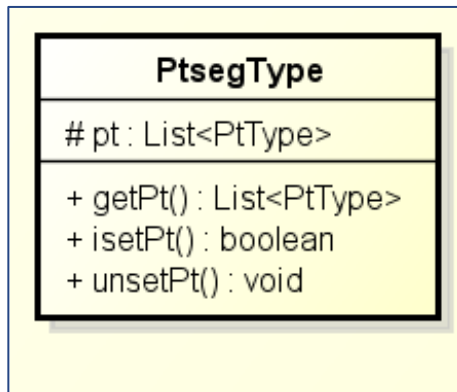
Il·lustració 10. Classe GpxWriter

Aquesta classe té la funció de generar un fitxer en format gpx i escriure'l a la memòria externa. En el moment d'instanciar-la se li passa un objecte de tipus *File* que conté el fitxer que es generarà. Quan es crida al mètode públic *save*, passant-li per paràmetre un objecte de tipus *GpxType*, aquest s'encarrega de serialitzar l'objecte convertint-lo en fitxer xml.

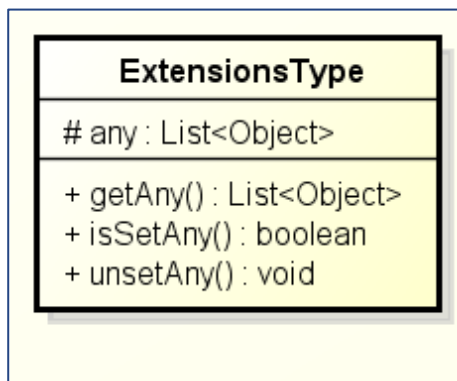


### 3.3.1.3 Classes de definició dels tipus utilitzats pels fitxers gpx

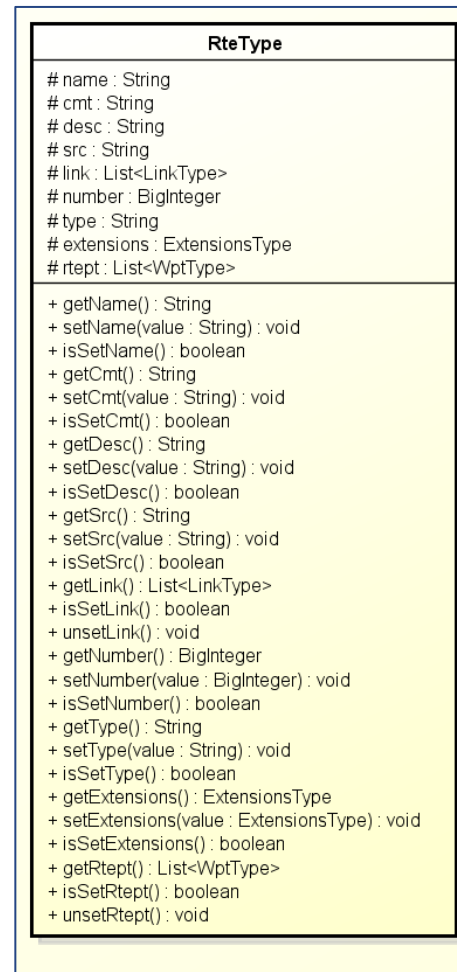
Un fitxer gpx conté diversos elements, alguns de simples, i d'altres de complexos. Per tal de poder crear un objecte que representi en memòria un fitxer gpx ha calgut definir tot un conjunt de tipus que es mostren tot seguit.



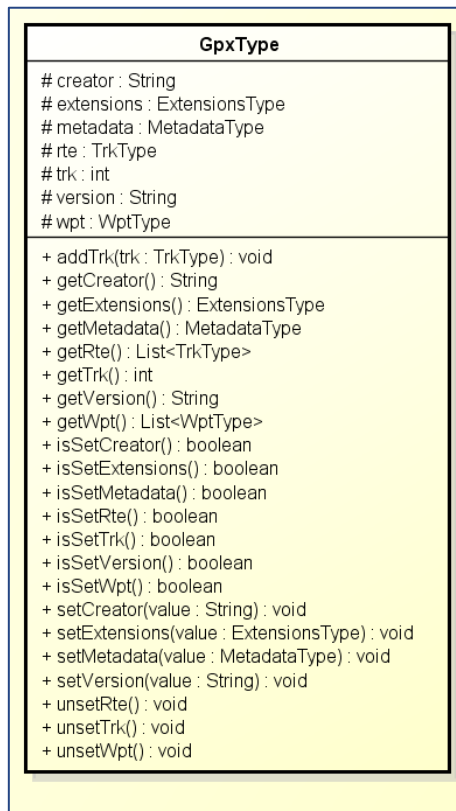
II·lustració 11. Classe PtsegType



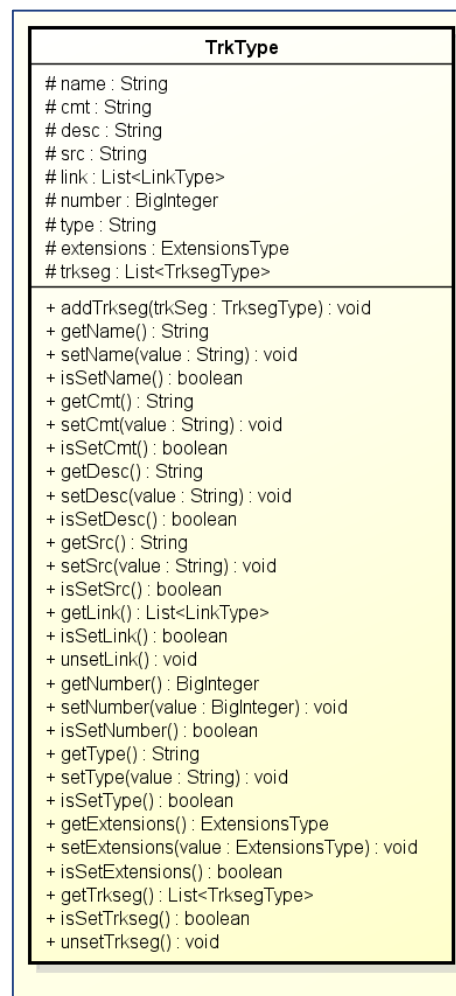
II·lustració 12. Classe ExtensionsType



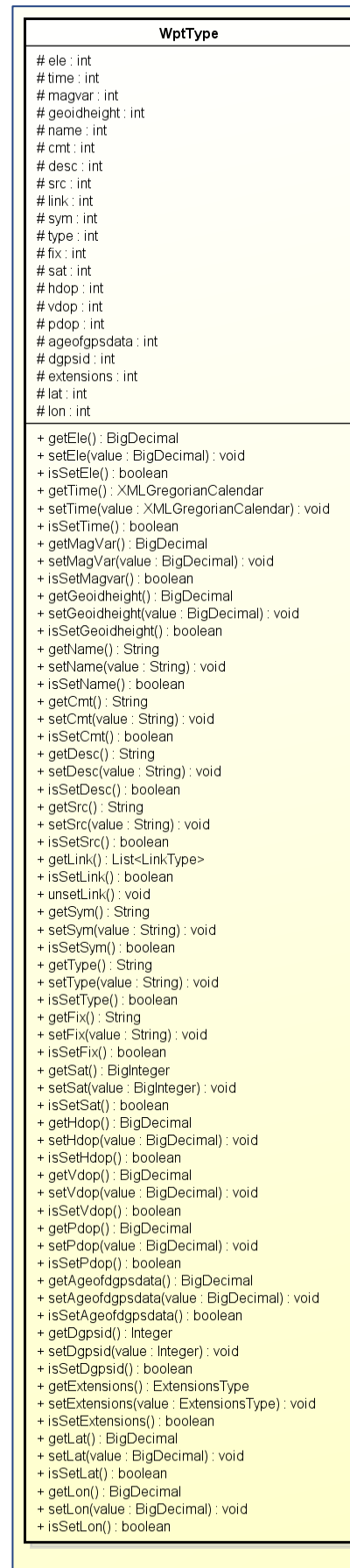
II·lustració 13. Classe RteType



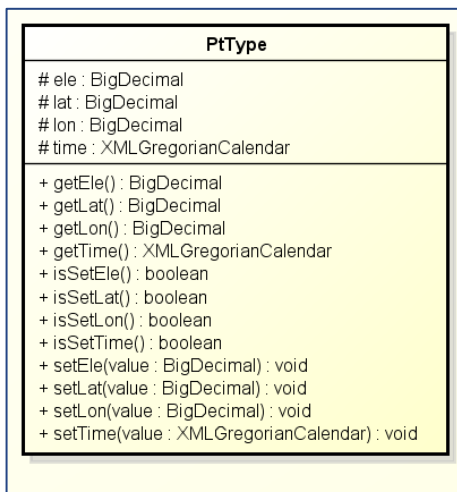
II -il·lustració 14. Classe GpxType



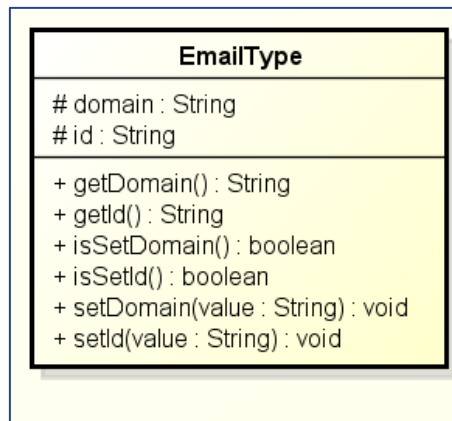
II -il·lustració 16. Classe TrkType



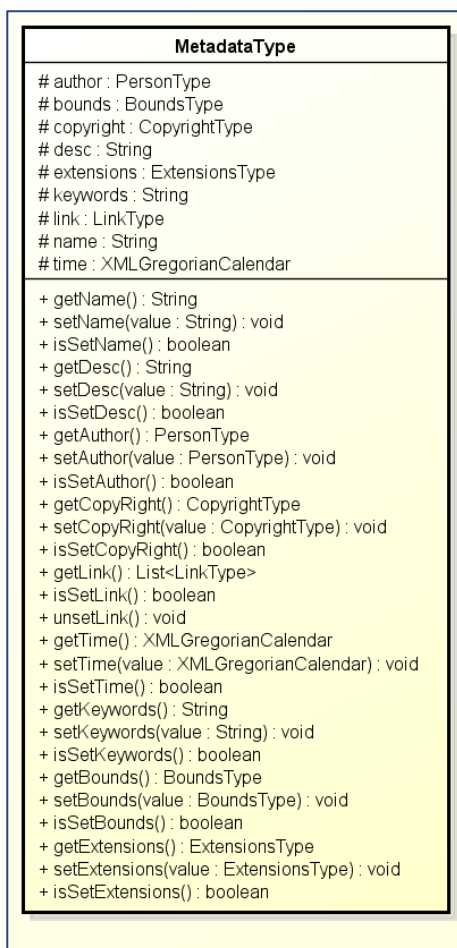
II -il·lustració 15. Classe WptType



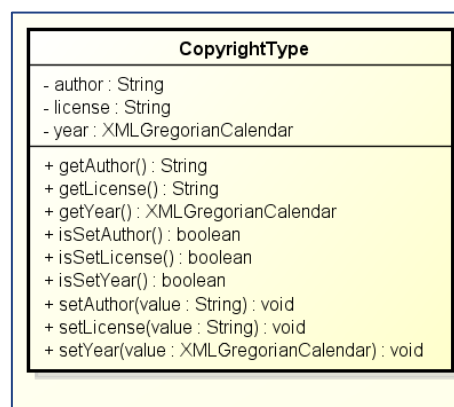
II -il·lustració 18. Classe PtType



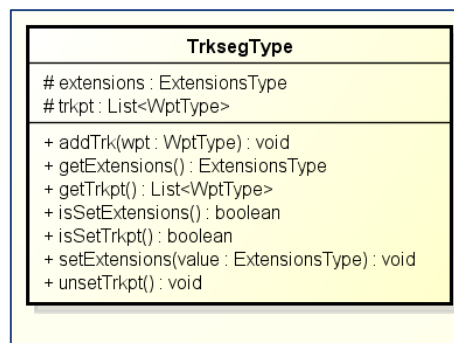
II -il·lustració 17. Classe EmailType



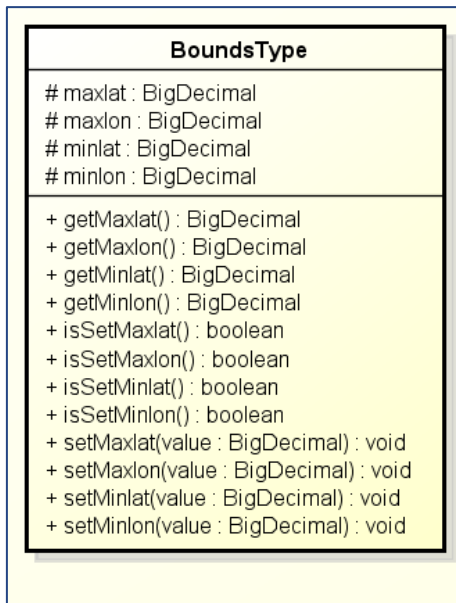
II -il·lustració 21. Classe MetadataType



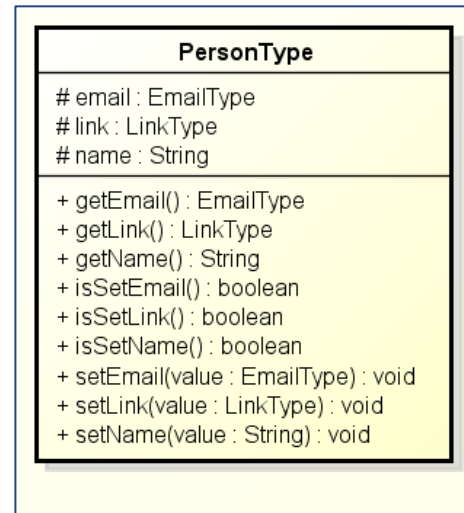
II -il·lustració 19. Classe CopyrightType



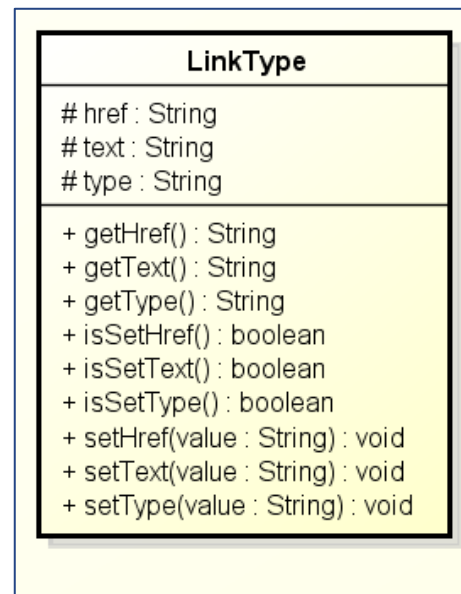
II -il·lustració 20. Classe TrkseType



Il·lustració 23. Classe BoundsType



Il·lustració 22. Classe PersonType



Il·lustració 24. Classe LinkType

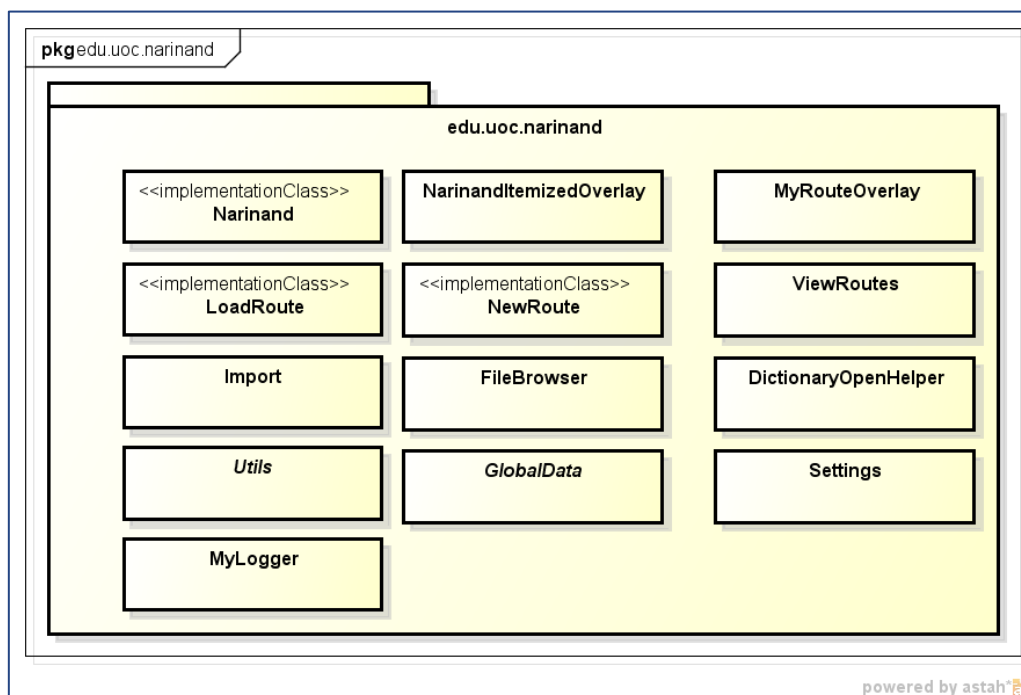
Com que es tracta de classes creades per representar l'objecte GxpType i els seus components, només cal definir-ne els camps i els mètodes *getters* i *setters* per poder escriure i llegir-ne els valors.

### 3.3.2 Paquet edu.uoc.narinand

En aquest paquet es desenvolupen totes les classes que donen la funcionalitat al programa, tant per mostrar pantalles, com per dur a terme les tasques i controls d'events, és a dir, la lògica de negoci.

Aquest paquet té la classe principal, anomenada Narinand, que és la que carrega el menú principal de l'aplicació i fa les crides a les altres per dur a terme les accions corresponents.

Moltes d'aquestes classes estenen o implementen d'altres classes bàsiques d'Android – Activity, ItemizedOverlay, ListItem..., i d'aquesta manera gestionen els diversos events que es donen en un dispositiu amb aquest sistema.



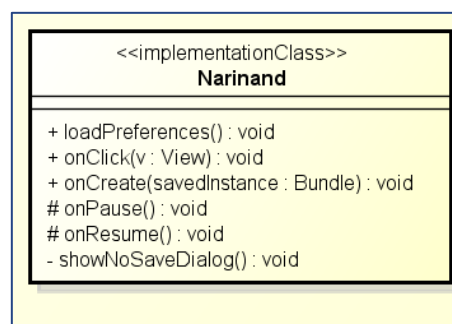
Il·lustració 25. Diagrama de classes del paquet edu.uoc.narinand

Aquest paquet consta de 13 classes de les quals es mostra el detall tot seguit.

#### 3.3.2.1 Classe Narinand

Aquesta classe estén la classe *Activity* d'Android, i d'aquesta manera s'utilitza per mostrar per pantalla el menú principal de l'aplicació.

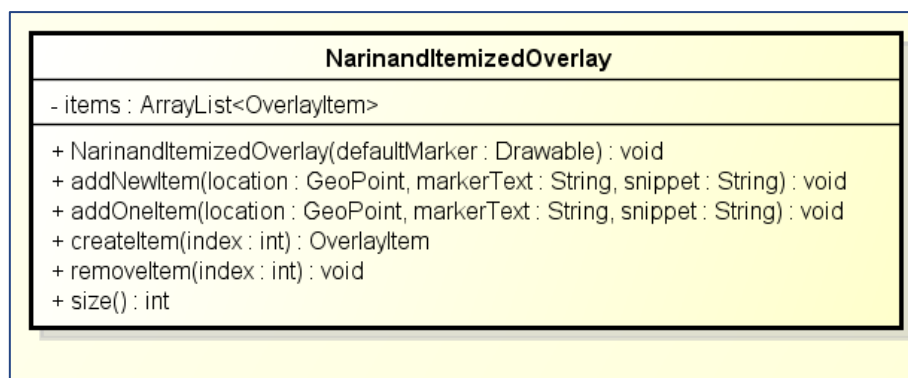
A més implementa la classe *View.OnClickListener* per escoltar events de clic a la pantalla. Això fa que es puguin posar elements que reaccionin quan l'usuari prem la pantalla.



Il·lustració 26. Classe Narinand

### 3.3.2.2 Classe NarinandItemizedOverlay

Aquesta classe estén la classe *ItemizedOverlay*, i té a veure amb les API de *Google Maps*. Cada instància que es crea d'aquesta classe permet dibuixar capes sobre el mapa que contenen símbols d'ajuda a la navegació.



Il·lustració 27. Classe NarinandItemizedOverlay

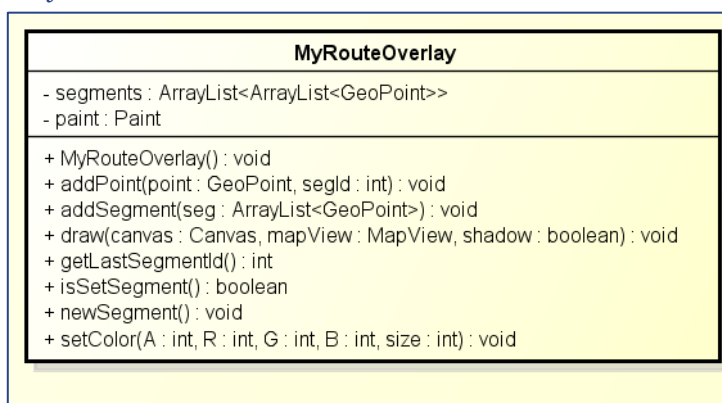
En aquest cas els símbols que es poden visualitzar són les marques de posició actual, d'inici i final de ruta actual, i d'inici i final de ruta carregada.

La classe conté els mètodes necessaris per afegir i treure elements de la capa de visualització.

### 3.3.2.3 Classe MyRouteOverlay

Aquesta classe estén la classe *Overlay*, que també forma part de les API de *Google Maps*. En aquest cas s'encarrega de dibuixar les línies de ruta, tant de la que s'està gravant, com de la que es carrega des d'un fitxer.

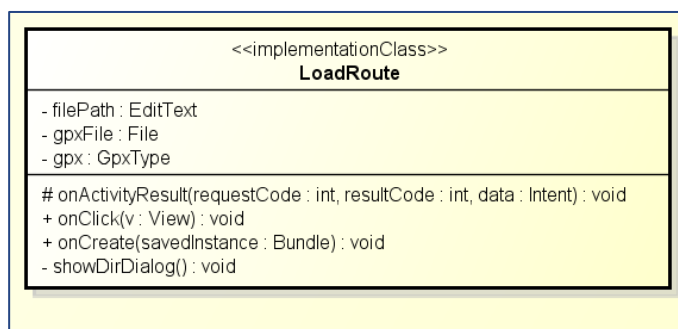
Igualment com passa amb la classe que dibuixa ítems, cada instanciació d'aquesta crea una nova capa que es superposa al mapa.



Il·lustració 28. Classe MyRouteOverlay

Aquesta classe conté els mètodes per afegir punts, segments, per definir el color i la mida de la línia, i per dibuixar.

### 3.3.2.4 Classe LoadRoute



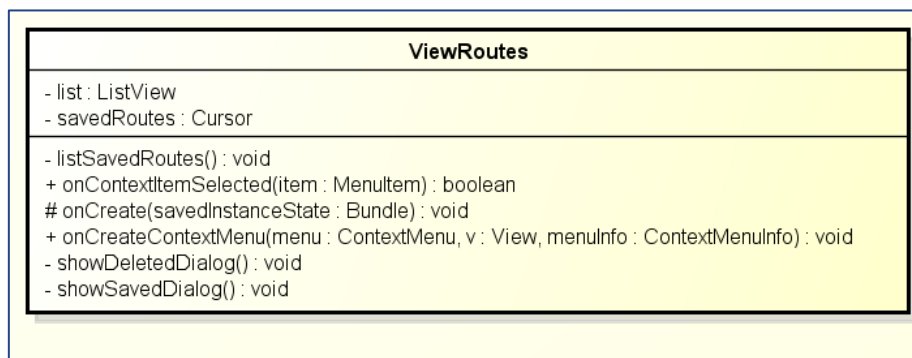
Il·lustració 29. Classe LoadRoute

Aquesta classe s'encarrega de mostrar per pantalla l'activitat de càrrega de ruta, i per tant, estén la classe *Activity*. Quan es crea, dibuixa a la pantalla els controls per cercar i carregar un fitxer gpx.

En aquest cas s'han de controlar els events de clic sobre els botons de control.

### 3.3.2.5 Classe ViewRoutes

Mitjançant aquesta classe es mostra a l'usuari una llista amb les rutes que es tenen guardades a la base de dades local del dispositiu, i a la vegada se li dóna la opció de tractar-les, ja sigui esborrant-les, exportant-les a fitxer gpx o bé carregant-les a la vista de mapa.



Il·lustració 30. Classe ViewRoutes

Com que es tracta d'una visualització, estén la classe *Activity*. En aquest cas, a més, es fa ús del menú contextual per a mostrar les opcions.

Fent una ullada als mètodes es veu clarament aquest ús del menú contextual, mitjançant *onContextItemSelected*, que registra l'event de selecció d'un ítem del menú contextual, i *onCreateContextMenu*, que registra l'event de creació del menú contextual. Aquest menú es crea després d'una pressió continuada de la pantalla.



### 3.3.2.6 Classe NewRoute

Aquesta és la classe on s'hi desenvolupen la major part de les tasques, ja que controla tots els events i accions que es duen a terme mentre es visualitza o es realitza una ruta.

És per això que de ben segur que és l'activitat que resta en primer pla més temps de totes les que formen part d'aquesta aplicació.

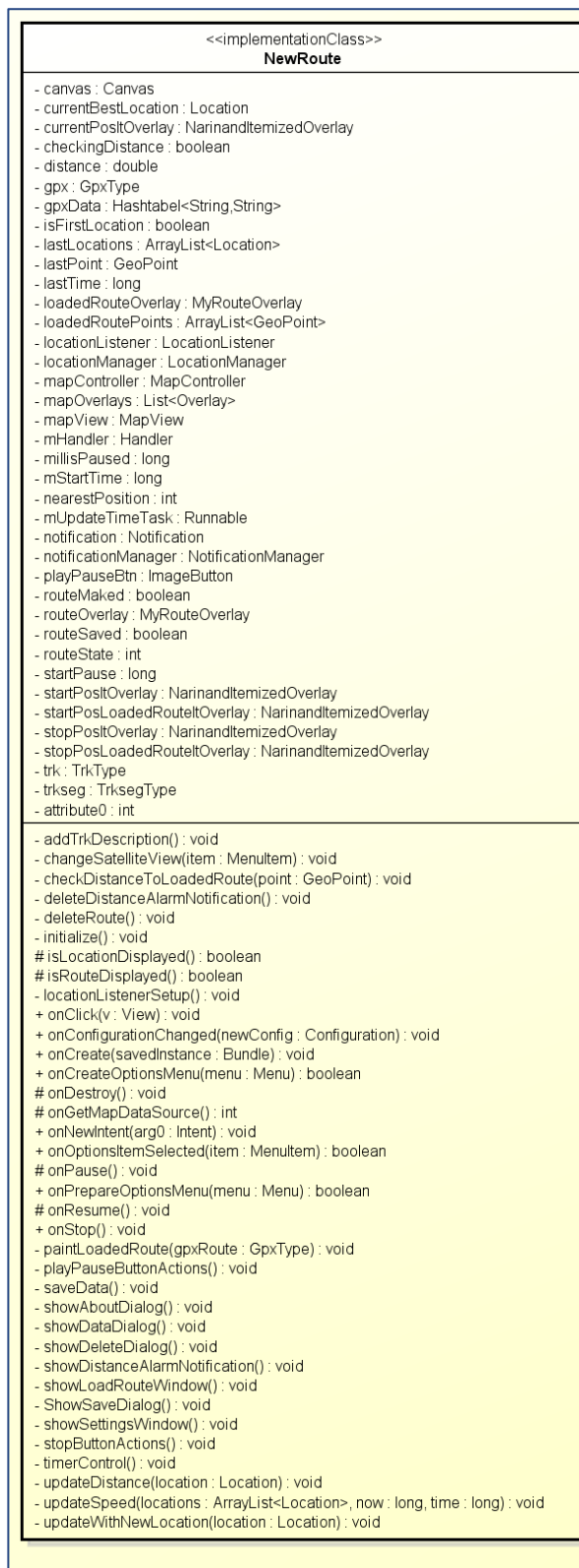
Es tracta d'una extensió de la classe *Activity* i per això es carrega com a pantalla amb la qual l'usuari pot interactuar.

En aquesta activitat s'ha de mostrar el mapa que es descarrega d'internet mitjançant l'ajuda de les API de *Google Maps*.

Hi ha una gran quantitat de mètodes que donen funcionalitat a aquesta classe, com ara el càlcul de la distància del punt actual amb la ruta carregada, la configuració de l'element que escolta les entrades des del dispositiu gps, el control del menú de pantalla, dels botons d'engegada, pausa i aturada...

A més, també hi ha mètodes que s'encarreguen de presentar la informació útil per a l'usuari per pantalla, com poden ser els que mostren la velocitat, el temps, la distància recorreguda, els que pinten la posició actual...

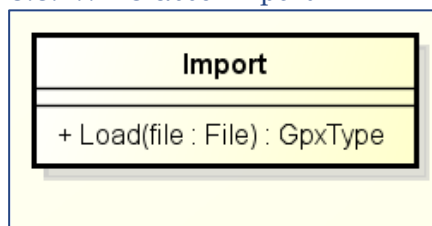
Amb tot això s'aconsegueix que l'aplicació tingui tota la utilitat desitjada.



Il·lustració 31. Classe NewRoute



### 3.3.2.7 Classe Import



Il·lustració 32. Classe Import

Es tracta d'una classe abstracta i molt senzilla on només hi ha un mètode.

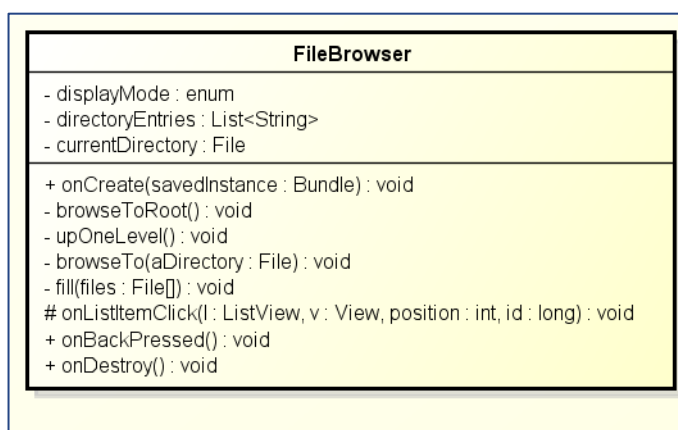
Aquest mètode rep un objecte de tipus *File* que conté un fitxer gpx, i retorna un objecte de tipus *GpxType* omplert amb les dades d'aquest fitxer.

### 3.3.2.8 Classe FileBrowser

Aquesta classe estén un subtipus d'activitat, anomenada *ListActivity*. Això és així perquè és necessari mostrar una llista d'elements per pantalla.

En realitat es mostra una llista de directoris i fitxers per la qual es pot navegar per escollir el fitxer que es vol carregar.

Destaquen mètodes per navegar a un directori concret, per anar al directori arrel, per pujar un nivell...

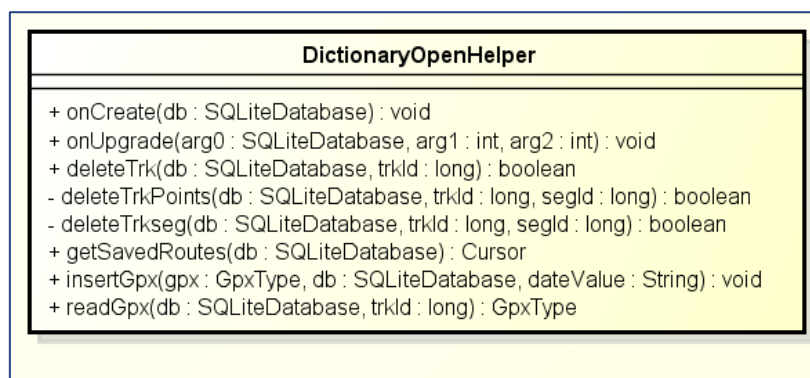


Il·lustració 33. Classe FileBrowser

D'aquesta manera es permet tenir un explorador de fitxers bàsic però efectiu per ajudar a l'usuari a trobar el fitxer desitjat.

### 3.3.2.9 Classe DictionaryOpenHelper

Mitjançant aquesta classe es controla tot el que fa referència a la gestió de la base de dades SQLite.

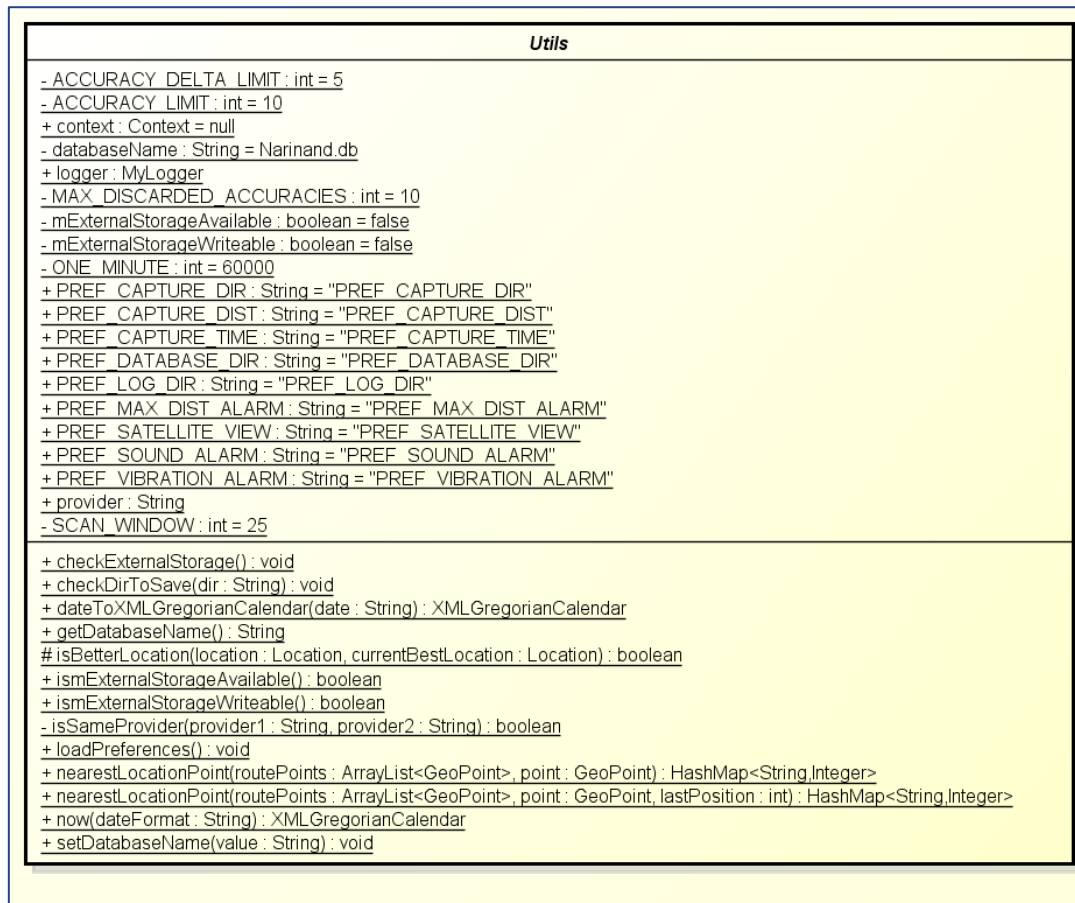


Il·lustració 34. Classe DictionaryOpenHelper

Amb els mètodes d'aquesta classe és possible crear la base de dades, i afegir, llegir i esborrar elements de les diverses taules.

### 3.3.2.10 Classe Utils

Es tracta d'una classe abstracta on hi ha un conjunt de mètodes que ajuden a dur a terme diverses operacions i càlculs que poden ser utilitzats des de qualsevol activitat de l'aplicació.



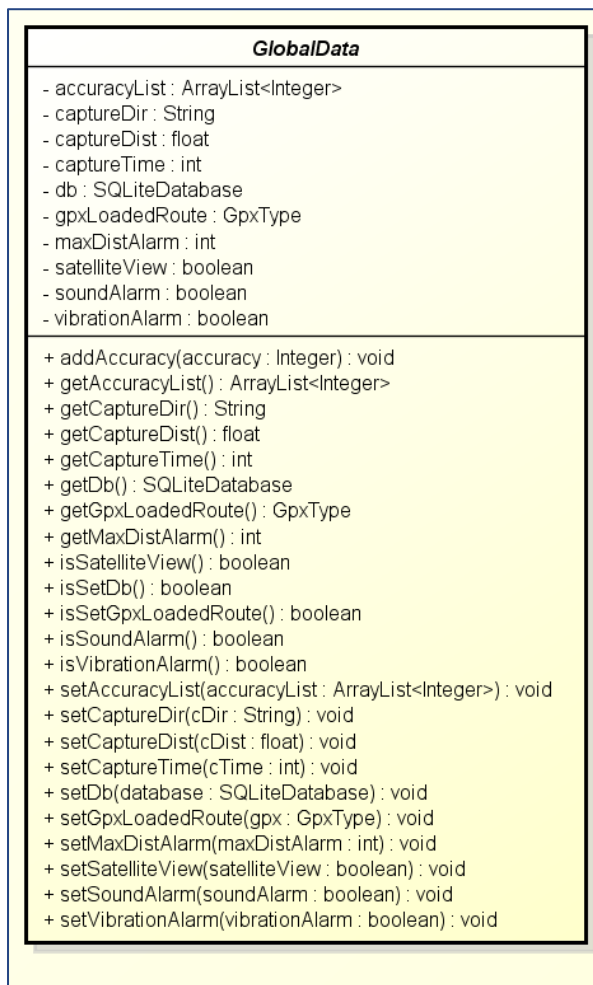
Il·lustració 35. Classe Utils

En aquesta classe s'hi defineixen un conjunt de constants que fa referència a paràmetres que no varien al llarg del cicle de vida del programa i que s'han de consultar una o diverses vegades.

També hi ha diversos mètodes amb funcionalitats molt diverses: comprovar l'emmagatzematge extern, comprovar si existeix un directori concret, crear una data amb format, calcular el punt amb distància menor, obtenir la data actual...

Amb tot aquest conjunt d'utilitats es dona un valor afegit a les activitats que formen el programa i s'evita duplicar mètodes en diferents classes.

### 3.3.2.11 Classe GlobalData



II·lustració 36. Classe GlobalData

Es tracta d'una classe abstracta on s'emmagatzemen els valors de les variables globals que comparteixen diverses classes i/o activitats de l'aplicació.

Mitjançant l'ús d'aquesta classe s'assegura que els paràmetres que han de ser consultats i/o actualitzats des de diversos elements del programa tinguin sempre un valor real en l'instant en què es necessiten.

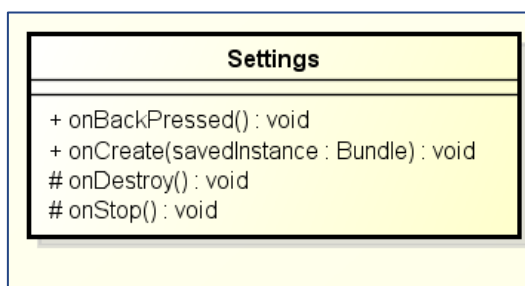
Tot el conjunt de mètodes incorporats en aquesta classe no són res més que els corresponents *getters* i *setters* per a les variables globals definides com a privades.

### 3.3.2.12 Classe Settings

Aquesta classe estén la que en *Android* s'anomena *PreferenceActivity*.

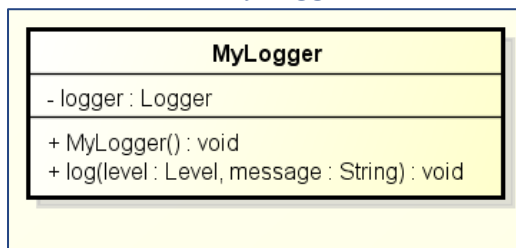
La funció d'aquest tipus d'activitats és la de carregar una pantalla amb totes les opcions de configuració de l'aplicació.

Per fer-ho es serveix del fitxer xml que s'ha definit en el seu moment, i on hi consten tots els elements que han d'aparèixer en els ajustos del programa.



II·lustració 37. Classe Settings

### 3.3.2.13 Classe MyLogger



Il·lustració 38. Classe MyLogger

Mitjançant aquesta senzilla classe s'aconsegueix crear un objecte que s'utilitza per registrar en un fitxer de log qualsevol excepció que hi pugui haver en l'execució del programa.

D'aquesta manera, en cas d'error, sempre es pot obtenir informació del que ha fallat.

## 3.4 Disseny de la persistència

Les úniques dades necessàries que cal persistir són les que fan referència a les rutes que s'han generat per a l'usuari, per a poder-les veure, exportar o compartir més endavant, no tan sols quan s'ha acabat de fer la ruta.

El sistema gestor de bases de dades que utilitza el sistema operatiu *Android* és l'anomenat *SQLite*<sup>6</sup>, i ve incorporat amb el *runtime* d'aquest sistema operatiu. Es tracta d'un SGBD molt lleuger que consumeix poca memòria –aspecte molt important en dispositius mòbils-.

Aquest sistema utilitza una API pròpia –en comptes de JDBC-, justament per evitar l'ús de massa memòria del dispositiu mòbil.

En tot cas, es tracta d'un SGBD relacional, i en conseqüència es pot definir l'estructura de la base de dades utilitzant aquest model.

### 3.4.1 Disseny de la base de dades

En l'aplicació que s'ha desenvolupant, per cada ruta es genera –seguint l'estructura de l'esquema gpx- un *track*. Aquest *track* podrà tenir un o diversos segments –un en el cas idoni, o diversos en cas que es talli la comunicació del receptor de gps amb els satèl·lits i es perdin alguns punts, o bé es pausi l'aplicació per part de l'usuari-. Així doncs hi podrà haver un sol segment si no s'ha perdut la comunicació en cap moment, o diversos en cas que s'hagi tallat, fent tots ells referència al mateix *track*.

Per últim, cada segment tindrà diversos punts retornats pel receptor de gps, on s'emmagatzemarà la latitud, longitud, alçada...

<sup>6</sup> Es tracta d'un sistema de domini públic, per la qual cosa és molt utilitzat per diversos fabricants.

### 3.4.2 Disseny relacional

Així doncs, un cop definits aquests requeriments es pot fer la següent descripció del model relacional de la base de dades.

**TRACKPOINTS** ( pointId, segmentId, trackId, latitude, longitude, elevation, time, fix )

segmentId és clau forània cap a la taula TRACKSEGMENTS

**TRACKSEGMENTS** ( segmentId, trackId )

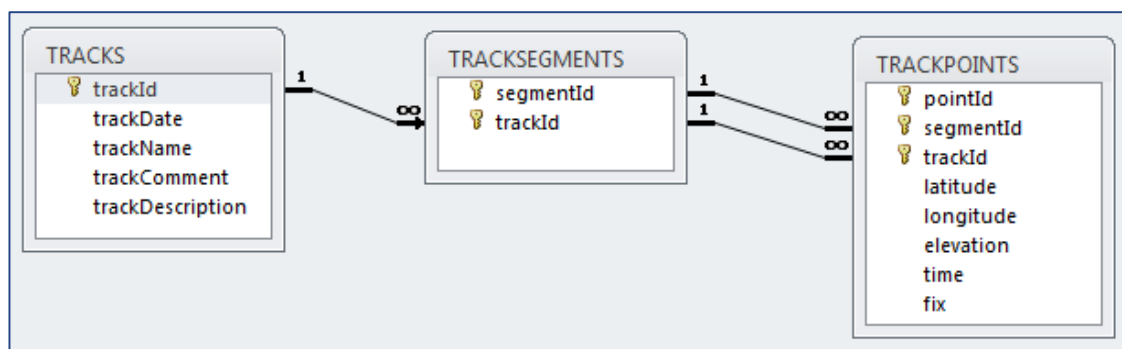
trackId és clau forània cap a la taula TRACKS

**TRACKS** ( trackId, date, name, comment, description )

Il·lustració 39. Disseny relacional de la base de dades

### 3.4.3 Diagrama de la base de dades

Segons el que s'ha dit en el punt anterior s'obté el següent diagrama.



Il·lustració 40. Diagrama de la base de dades

Les relacions que hi ha són de 1..n per a *tracks* i *segments*, i de 1..n per a *segments* i *punts*. D'aquesta manera s'obté que un *track* té un o diversos segments, i cada segment té un o diversos punts.

Mitjançant aquesta senzilla estructura es poden emmagatzemar les dades més importants de les rutes que es generes des del dispositiu, i a més es poden exportar fàcilment a format gpx.

### 3.5 Llibreries a utilitzar

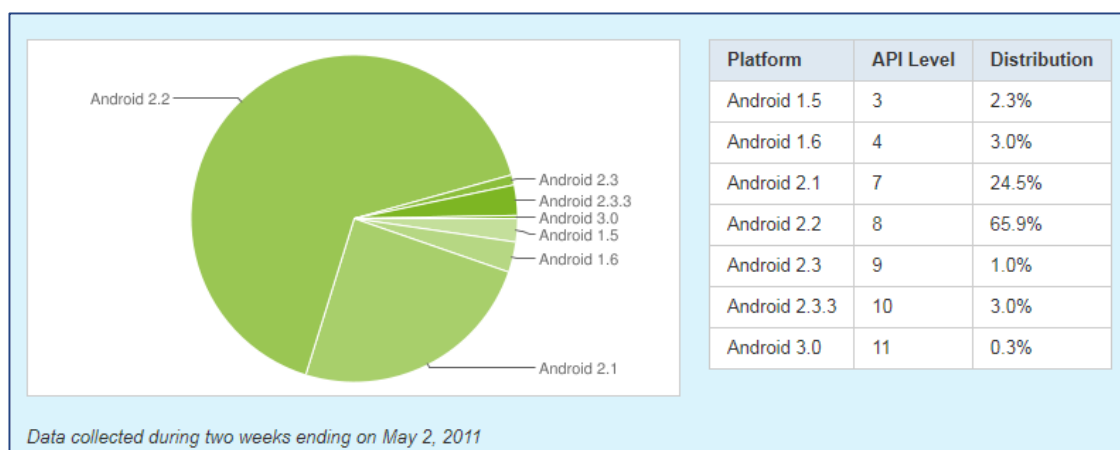
Per a desenvolupar l'aplicació per a *Android* s'utilitzen un conjunt de llibreries que venen incorporades amb el paquet de SDK d'*Android*.

Aquestes llibreries formen el que s'anomenen API de desenvolupament. Hi ha diverses versions d'aquestes llibreries, segons la versió del sistema *Android* amb el que són compatibles.

Hi ha dos tipus d'API, les genèriques d'*Android* que contenen totes les funcionalitats per a relacionar-se amb el sistema, i les de *Google Maps*. La diferència és que aquestes segones porten incorporades les llibreries per al tractament de mapes de *Google Maps*, a més de totes les funcionalitats bàsiques que ja incorporen les genèriques.

Per aquest projecte, com que està completament dirigit cap a l'ús dels mapes, s'han utilitzat les API de *Google Maps*. Concretament s'ha optat per utilitzar la versió 8, revisió 2, que equivalen a la versió 2.2 –Froyo– d'*Android*.

S'ha decidit utilitzar aquesta versió perquè avui en dia és la més estesa, i la que incorporen la majoria de terminals que han sortit entre 2010 i 2011. Així s'arriba a un equilibri entre compatibilitat i novetat. Segons un informe del dia 2 de maig de 2011, publicat a la web oficial d'*Android*<sup>7</sup>, aquesta versió és la que tenen el 65,9% dels dispositius que han accedit a l'*Android Market* durant els 15 dies anteriors.

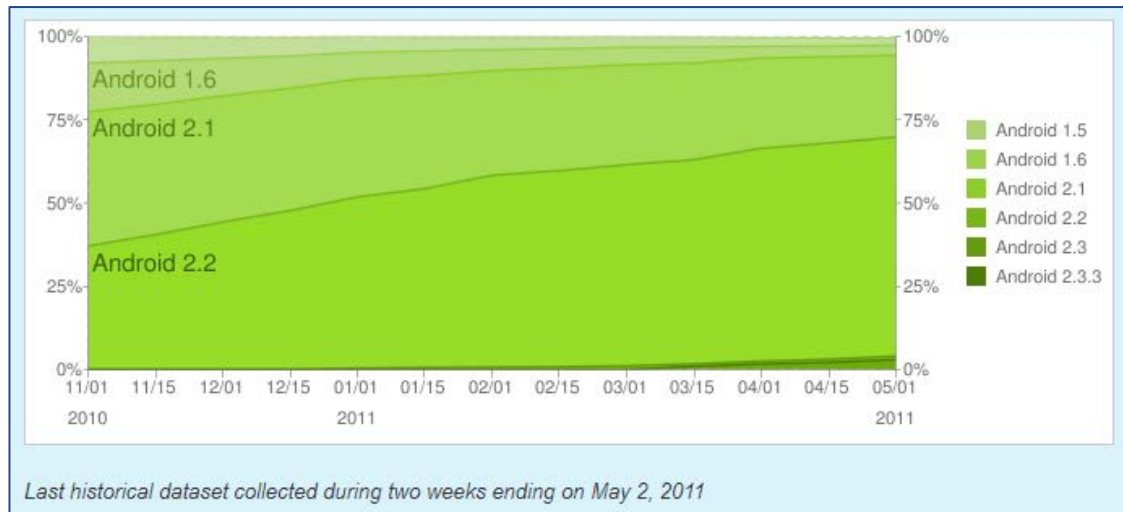


Il·lustració 41. Percentatges d'ús de les diverses versions d'Android

Veient el gràfic de la il·lustració 41 queda clar que la millor opció actual era desenvolupar-lo per a la versió 2.2, ja que és compatible amb la majoria dels dispositius que hi ha al mercat avui en dia.

Si es para atenció a l'evolució històrica dels darrers mesos, mostrada a la il·lustració 42, es constata el que ja s'ha comentat abans, el nombre de dispositius amb versió 2.2 continua creixent, mentre que els que tenen versions anteriors es van reduint. Així mateix, comença a augmentar el nombre de dispositius amb versió 2.3 i 2.3.3 –*Gingerbread*–, però el percentatge en el total de dispositius és, a dia d'avui, encara testimonial.

<sup>7</sup> <http://www.android.com>



Il·lustració 42. Evolució històrica de les diverses versions d'Android

A més d'aquestes llibreries s'ha decidit incorporar al projecte unes llibreries externes per a tenir compatibilitat amb la creació de dates i hores amb el calendari gregorià. Això s'ha hagut de fer perquè les llibreries d'*Android* –almenys en el nivell d'API 8- no incorporen aquesta funcionalitat que és necessària per a poder llegir i gravar fitxers del tipus gpx.

Aquestes llibreries s'anomenen "it.tidalwave.android.javax.xml.datatype".

De la mateixa manera, en un principi es tenia pensat serialitzar l'objecte de tipus GpxType directament cap a un fitxer gpx, evitant així la generació de codi per a transformar-lo. Aquesta opció, però, no ha estat possible ja que les API no incorporen aquesta facilitat per a objectes complexos. Així doncs, s'ha optat per desenvolupar una lògica de negoci que a partir de l'objecte complex GpxType generi un fitxer xml.

## 4 IMPLEMENTACIÓ

### 4.1 Decisions d'implementació

En el pla de treball del projecte es va posar èmfasi en el fet que es volia fer que l'aplicació permetés descarregar els mapes al dispositiu abans de començar la ruta. Malauradament aquesta funcionalitat no s'ha pogut implementar, ja que les API de *Google Maps* no permeten aquesta funcionalitat actualment.

En la descripció d'aquest projecte ja s'ha comentat que en el seu dia es va valorar la opció d'utilitzar unes llibreries de tercers que sí que ho permetessin, però es va descartar la idea pel fet que utilitzen mapes del projecte *OpenStreetMap* que, en tractar-se d'un projecte col·laboratiu on la gent ha de dibuixar els mapes de manera desinteressada, no té cobertura de mapes complets a gran part del territori.

Així doncs en aquest punt es va prendre la decisió de descartar aquesta funcionalitat, donant preferència a la qualitat dels mapes en front de la possibilitat de tenir-los descarregats al dispositiu.

A causa del temps reduït del que es disposava per dur a terme el projecte es va decidir fer una aplicació amb les utilitats mínimes que es van posar com a objectius en el moment de plantejar el projecte. Així doncs, el projecte compleix amb gairebé tots els requeriments que s'havien definit, tot i que amb més temps es podrien afegir noves funcionalitats i refinaments.

### 4.2 Els fitxers gpx

Per a poder complir amb les especificacions dels fitxers gpx definits per *Topografix* és necessari que els fitxers que es generen tinguin uns mínims imprescindibles, i que l'aplicació sigui capaç de llegir qualsevol fitxer en aquest format.

```
<gpx
  version="1.1 [1] ?"
  creator="xsd:string [1] ?">
  <metadata> metadataType </metadata> [0..1] ?
  <wpt> wptType </wpt> [0..*] ?
  <rte> rteType </rte> [0..*] ?
  <trk> trkType </trk> [0..*] ?
  <extensions> extensionsType </extensions> [0..1] ?
</gpx>
```

Il·lustració 43. Esquema principal d'un fitxer gpx

Tot fitxer gpx ha de tenir informada la versió i el creador obligatòriament.

La resta d'elements són opcionals, però és clar que algun n'ha de tenir si es vol representar alguna dada.

Per aquest projecte s'ha descartat l'escriptura i lectura

de les metadades i els punts d'interès, ja que està encaminada a la representació de les rutes, i per tant no és necessari. Tampoc es tenen en compte les extensions que s'hi puguin haver afegit.

Així doncs, mai no es representaran els objectes *<metadata>*, *<wpt>* i *<extensions>*. Els objectes que sí que es llegeixen són els de tipus ruta i track *<rte>* i *<trk>*.

A l'hora d'escriure, només s'escriuen objectes de tipus track, que permeten representar qualsevol ruta.



### 4.3 Paquet edu.uoc.narinand.gpx

Com ja s'ha vist en la descripció del disseny, el paquet edu.uoc.narinand.gpx conté tots els objectes que componen l'estructura d'un fitxer de tipus gpx, i a més, dos components per a llegir i escriure un fitxer d'aquest tipus.

Aquests components són completament independents del desenvolupament per a *Android*, és a dir, no estenen ni implementen cap de les classes pròpies d'aquest sistema. Així doncs són completament compatibles i reutilitzables per a qualsevol aplicació escrita en llenguatge *Java*.

#### 4.3.1 Definicions del tipus de dades GPX

De les setze classes que formen part d'aquest paquet, catorze implementen objectes per carregar a memòria les dades d'un fitxer amb estructura gpx.

Així doncs aquests catorze components són definicions d'objectes amb un conjunt de variables de diversos tipus i els seus corresponents mètodes per assignar i obtenir els valors.

Segons l'estructura d'un fitxer gpx, els diversos tipus que s'han generat són:

- GpxType
- MetadataType
- WptType
- RteType
- TrkType
- ExtensionsType
- TrksegType
- CopyrightType
- LinkType
- EmailType
- PersonType
- PtType
- PtsegType
- BoundsType

El tros de codi representat a la il·lustració 44 permet fer-se una idea de la implementació dels diversos tipus enumerats anteriorment.

```
package edu.uoc.narinand.gpx;

import java.util.ArrayList;

/**
 * Tipus definit a l'esquema del format GPX per a contenir les dades del fitxer
 * gpx
 */
public class GpxType {

    /** Dades contingudes en el fitxer gpx */

    protected MetadataType metadata;
    protected List<WptType> wpt;
    protected List<RteType> rte;
    protected List<TrkType> trk;
    protected ExtensionsType extensions;
    protected String version;
    protected String creator;

    /** Getters i setters del tipus */

    public void addTrk(TrkType trk) {
        if (this.isSetTrk())
            this.trk.add(trk);
        else {
            this.trk = new ArrayList<TrkType>();
            this.trk.add(trk);
        }
    }

    public String getCreator() {
        return creator;
    }

    public ExtensionsType getExtensions() {
        return extensions;
    }
}
```

Il·lustració 44. Exemple de codi dels tipus de dades que formen un objecte gpx

L'estructura en tots els casos és la mateixa, i consta de la definició de les diverses variables que representes les dades del tipus que es defineix, i els respectius mètodes que permeten accedir-hi, ja sigui per obtenir-ne el resultat, com per assignar-n'hi un.

### 4.3.2 Component GPXReader

Mitjançant la instanciació d'aquesta classe es pot carregar el contingut d'un fitxer gpx a la memòria –a un objecte de tipus GpxType-.

Per qüestions de prioritat en el desenvolupament del projecte –a causa del poc temps disponible per a la seva realització- no s'han implementat els mètodes de càrrega de tots els tipus de dades, ja que l'aplicació no en fa ús de molts d'ells.

Així doncs, s'ha centrat l'esforç en la implementació dels mètodes que carreguen les dades imprescindibles pel funcionament correcte de l'aplicació - *Waypoints, Tracks, Routes...*-.

```
/**
 * Mètode per a carregar l'objecte TrksegType des del fitxer gpx
 *
 * @return TrksegType trkseg
 */
private TrksegType trkseg() {
    TrksegType trkseg = new TrksegType();

    try {
        // Mentre no s'arribi al final del document i no es trobi l'etiqueta
        // de tancament de l'element trkseg
        while (eventType != XmlPullParser.END_DOCUMENT
            && (eventType != XmlPullParser.END_TAG || xpp.getName()
                .compareTo("trkseg") != 0)) {
            // Es van llegint els subelements de trkseg
            if (eventType == XmlPullParser.START_TAG) {
                if (xpp.getName().compareTo("trkpt") == 0) {
                    if (!trkseg.isSetTrkpt())
                        trkseg.trkpt = new ArrayList<WptType>();
                    trkseg.trkpt.add(readWpt("trkpt"));
                }
            }
            // S'obté el següent element
            eventType = xpp.next();
        }
        return trkseg;
    } catch (XmlPullParserException xpe) {
        Utils.logger.log(Level.ALL, this.getClass().getName()
            + " trkseg error: " + xpe.getMessage());
        return null;
    } catch (IOException ioe) {
        Utils.logger.log(Level.ALL, this.getClass().getName()
            + " trkseg error: " + ioe.getMessage());
        return null;
    }
}
```

Il·lustració 45. Mètode per llegir el tipus TrksegType d'un fitxer gpx

En el tros de codi representatiu mostrat en la il·lustració 45 es pot veure que per llegir els diversos tipus d'un fitxer es fa servir el parsejador *XmlPullParser*, molt eficient i ràpid, i per això ideal per a desenvolupaments per a dispositius mòbils.

### 4.3.3 Component GpxWriter

Mitjançant la instanciació d'aquesta classe es pot serialitzar el contingut d'un objecte GpxType cap a un fitxer gpx ubicat a la memòria externa del dispositiu.

De la mateixa manera que s'ha fet amb la opció de llegir, en la d'escriure també s'han deixat sense implementar els mètodes que no són obligatoris per a generar un fitxer gpx amb estructura correcta.

```
/**
 * Mètode per a serialitzar l'objecte link
 *
 * @param LinkType
 *         link
 */
private void writeLink(LinkType link) {
    try {
        // Es crea l'etiqueta link i els seus elements fill en cas que
        // existeixin
        serializer.startTag(null, "link");
        if (link.isSetHref()) {
            serializer.attribute(null, "href", link.getHref());
        } else {
            serializer.attribute(null, "href", "");
        }
        if (link.isSetText()) {
            serializer.startTag(null, "text");
            serializer.text(link.getText());
            serializer.endTag(null, "text");
        }
        if (link.isSetType()) {
            serializer.startTag(null, "type");
            serializer.text(link.getType());
            serializer.endTag(null, "type");
        }
        // Es tanca l'etiqueta link
        serializer.endTag(null, "link");
    } catch (Exception e) {
        Utils.logger.log(Level.ALL, this.getClass().getName()
            + " writeLink error: " + e.getMessage());
    }
}
```

Il·lustració 46. Mètode per serialitzar el tipus LinkType a un fitxer gpx

Per a gravar a un fitxer s'utilitza un component de tipus *XmlSerializer* que ajuda a crear un fitxer xml a partir d'un objecte. Un exemple del seu funcionament es pot veure en la il·lustració 46.

## 4.4 Paquet edu.uoc.narinad

Els components que formen part d'aquest paquet sí que són exclusius per a dispositius *Android*, i implementen i estenen diverses classes d'aquest sistema.

En primer lloc s'expliquen tot el conjunt de fitxers de configuració i disseny que pot tenir qualsevol projecte *Android*. Pel que fa al projecte *Narinand* s'han utilitzat els que es detallen tot seguit.

### 4.4.1 AndroidManifest.xml

Un fitxer indispensable per al funcionament d'una aplicació per a *Android* és el fitxer de manifest, on es defineixen diversos paràmetres de l'aplicació.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.uoc.narinand" android:versionCode="1" android:versionName="1.0">

    <!-- Versió mínima d'SDK i versió preferida -->
    <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="8" />

    <!-- Permisos que ha de tenir l'aplicació sobre el dispositiu -->
    <!-- Per accedir als sistemes de localització GPS -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_POSITION" />
    <uses-permission android:name="android.permission.ACCESS_GPS" />
    <!-- Per accedir a internet -->
    <uses-permission android:name="android.permission.INTERNET" />
</manifest>
```

Il·lustració 47. Fragment del fitxer AndroidManifest.xml

En aquest tall del fitxer de manifest es pot veure que l'aplicació està pensada per dispositius amb nivell d'API 8, i alguns dels permisos que es requereixen.

Més avall vindria la part on es defineixen les diverses activitats que formen part de l'aplicació.

### 4.4.2 Dissenys de pantalla o *Layouts*

Tota activitat necessita tenir el seu *Layout* o disseny de quins elements ha de contenir, i de com s'han de visualitzar. Tot i que es poden construir dinàmicament per codi, en aquest cas s'han definit estàticament mitjançant fitxers xml per a donar independència entre la capa de presentació i la lògica de negoci.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Definició de la vista per a mostrar el menú principal de l'aplicació -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
    <RelativeLayout android:id="@+id/titleBar"
        android:layout_width="fill_parent" android:layout_height="wrap_content">
        <ImageView android:id="@+id/icon" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:paddingLeft="5px"
            android:src="@drawable/icon">
        </ImageView>
        <RelativeLayout android:id="@+id/titleText"
            android:layout_width="fill_parent" android:layout_height="wrap_content">
            <TextView android:id="@+id/title" style="@style/Title"
                android:layout_toRightOf="@id/icon" android:padding="5px"
                android:text="@string/app_name">
            </TextView>
        </RelativeLayout>
    </RelativeLayout>
```

Il·lustració 48. Fragment del fitxer de layout de la pantalla principal

Hi ha diferents maneres de presentar una vista o *layout*, organitzant els seus elements de forma lineal, relativa, en forma de taula...

Per a la implementació de les diverses activitats d'aquest projecte s'ha utilitzat principalment el tipus relatiu, que permet col·locar els diversos components en posicions relatives d'uns respecte als altres. D'aquesta manera es pot controlar molt millor la manera com es col·loquen a la pantalla.

En les il·lustracions 47 i 48 es poden veure dos fragments de la definició d'una vista per al menú principal de l'aplicació i per a la pantalla de nova ruta respectivament.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Definició de la vista que mostra el mapa i permet veure i generar rutes -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/newroute" android:layout_height="fill_parent"
    android:layout_width="fill_parent">
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/newrouteButtons" android:layout_height="wrap_content"
        android:layout_width="fill_parent" android:layout_alignParentBottom="true">
        <ImageButton android:id="@+id/playPause"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:layout_alignParentLeft="true" android:src="@drawable/play" />
        <ImageButton android:id="@+id/stop" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:layout_alignParentRight="true"
            android:src="@drawable/stop" />
        <TextView android:id="@+id/cronoTxt"
            android:layout_toRightOf="@id/playPause" android:paddingLeft="5px"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            style="@style/BottomBar" android:text="0:00:00">
        </TextView>
        <TextView android:id="@+id/speedTxt" android:layout_toLeftOf="@id/stop"
            android:paddingRight="5px" android:layout_width="wrap_content"
            android:layout_height="wrap_content" style="@style/BottomBar"
            android:text="Vel: 0.0 km/h">
        </TextView>
        <TextView android:id="@+id/distanceLbl"
            android:layout_toRightOf="@id/playPause" android:layout_below="@id/cronoTxt"
            android:paddingLeft="5px" android:layout_width="wrap_content"
            android:layout_height="wrap_content" style="@style/BottomBar"
            android:text="@string/distanceLbl">
        </TextView>
        <TextView android:id="@+id/distanceTxt"
            android:layout_toLeftOf="@id/stop" android:paddingRight="5px"
            android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:layout_below="@id/speedTxt" style="@style/BottomBar"
            android:text="0 m">
        </TextView>
    </RelativeLayout>
</RelativeLayout>
```

Il·lustració 49. Fragment del fitxer de layout de la pantalla de nova ruta

En aquests fitxers s'hi defineixen les propietats de la vista en general, i de cadascun dels elements que en formen part en particular.

Algunes d'aquestes propietats són:

- **Id.** Nom que se li assigna a l'element i que pot ser referenciat des de qualsevol altre.
- **Layout\_width** i **layout\_heigh**. Per indicar l'alçada i l'amplada que ha d'ocupar un element.
- **Src.** Recurs on hi ha la imatge que s'ha de mostrar.
- **Text.** Text que ha de mostrar l'element.

- **Above, below, toLeftOf, toRightOf.** Propietats que són exclusives de la vista relativa, i que indiquen la posició d'un element respecte un altre de la mateixa vista.
- ...

#### 4.4.3 Menú de l'activitat *NewRoute*

Quan es vol que una activitat mostri un menú en el moment en què es prem la tecla "Menu" característica de tots els dispositius *Android*, s'ha de definir l'aspecte i els components que aquest menú ha de tenir.

Tot i que també es pot definir per codi, en aquest cas s'ha definit mitjançant un fitxer xml per conservar la independència entre capes de presentació i de negoci.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Opcions de menú que apareix quan es prem la tecla "Menu" del mòbil mentre s'està fent una ruta -->
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/loadRoute" android:icon="@drawable/loadroute"
    android:title="@string/loadRouteTxt" />
  <item android:id="@+id/settings" android:icon="@drawable/settings"
    android:title="@string/preferencesTxt" />
  <item android:id="@+id/satellite" android:icon="@drawable/satellite"
    android:title="@string/satelliteTitle" />
  <item android:id="@+id/about" android:icon="@drawable/about"
    android:title="@string/aboutTxt" />
</menu>
```

Il·lustració 50. Fitxer routemenu.xml

En aquest cas es pot veure que s'han definit quatre ítems per a aquest menú, cadascun amb un text per mostrar a la pantalla *-title-*, i amb una imatge *-icon-*.

Amb aquesta senzilla definició s'aconsegueix construir un menú, al qual després cal dotar de funcionalitat implementant les accions en la lògica de negoci.

#### 4.4.4 Pantalla de preferències

En *Android*, hi ha una manera molt senzilla de dissenyar una pantalla per a definir ajustos de l'aplicació. Com moltes de les coses que es defineixen en aquest sistema, es pot fer creant un senzill fitxer xml que contingui la informació necessària dels elements que han d'aparèixer a la pantalla de preferències.

En aquest projecte s'han definit una sèrie de paràmetres que poden ser variables i que per tant es poden configurar al gust de l'usuari final.

Les diverses opcions que es poden escollir són:

- **Temps de captura.** Per definir cada quants segons es demana la posició al dispositiu gps. Es tracta d'una llista d'opcions predeterminades, on cal escollir-ne una.
- **Distància de captura.** Per definir cada quants metres es demana la posició al dispositiu gps. Es tracta d'una llista d'opcions predeterminades, on cal escollir-ne una.
- **Directorí gpx.** Directorí del dispositiu on s'emmagatzemen i es llegeixen els fitxers gpx. Es tracta d'un quadre de text editable.
- **Directorí base de dades.** Directorí del dispositiu on s'emmagatzema la base de dades. Es tracta d'un quadre de text editable.
- **Directorí fitxer de log.** Directorí del dispositiu on s'emmagatzema el fitxer de log. Es tracta d'un quadre de text editable.



- **Màxima distància alarma.** Separació en la qual sonarà l'alarma d'allunyament de ruta. Es tracta d'una llista d'opcions predeterminades, on cal escollir-ne una.
- **Alarma per vibració.** Indica si el dispositiu ha de vibrar quan s'allunya de la ruta marcada. És un *checkbox* que es pot marcar o desmarcar.
- **Alarma sonora.** Indica si el dispositiu ha d'emetre un so quan s'allunya de la ruta marcada. És un *checkbox* que es pot marcar o desmarcar.
- **Vista satèl·lit.** Per escollir entre la vista topogràfica o de satèl·lit. És un *checkbox* que es pot marcar o desmarcar.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Aquest fitxer defineix les opcions de configuració que tindrà el programa -->
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
  <ListPreference android:key="PREF_CAPTURE_TIME"
    android:title="@string/captureTime" android:summary="@string/captureTimeDesc"
    android:entries="@array/captureTimeOptions" android:entryValues="@array/captureTimeValues"
    android:dialogTitle="@string/captureTime" android:persistent="true"
    android:defaultValue="1" />
  <ListPreference android:key="PREF_CAPTURE_DIST"
    android:title="@string/captureDist" android:summary="@string/captureDistDesc"
    android:entries="@array/captureDistOptions" android:entryValues="@array/captureDistValues"
    android:dialogTitle="@string/captureDist" android:persistent="true"
    android:defaultValue="1" />
  <EditTextPreference android:key="PREF_CAPTURE_DIR"
    android:title="@string/captureDir" android:summary="@string/captureDirDesc"
    android:persistent="true" android:defaultValue="@string/captureDefaultDir" />
</PreferenceScreen>
```

Il·lustració 51. Fragment del fitxer Settings.xml

En cadascun dels elements es defineixen diversos aspectes, com el títol, la descripció, els valors possibles –en cas que en necessiti-, el valor per defecte...

#### 4.4.5 Fitxers de recursos

Quan es desenvolupa una aplicació és molt important poder definir els diversos recursos en fitxers separats, independents del codi i dels altres fitxers de configuració.

Si es fa aquesta separació de manera correcta és molt fàcil poder corregir possibles errors ortogràfics que poden aparèixer a diversos quadres de diàleg o opcions de menú, i sobretot, dóna la possibilitat de poder fer la localització de l'aplicació per a diversos idiomes sense haver de fer diferents compilats.

Per al projecte *Narinand* s'han creat tres fitxers de recursos diferents, cadascun amb una finalitat concreta, perseguint sempre la màxima independència de la capa de presentació en front de la lògica de negoci.

#### 4.4.5.1 Fitxer de recursos Strings.xml

En aquest fitxer s'hi ha definit tots els textos que mostra l'aplicació, ja sigui en opcions de menú, quadres de diàleg, etiquetes de les activitats...

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Aquest fitxer defineix els diversos textos que mostra l'aplicació. Amb
aquest mètode es pot aconseguir la localització del programa molt fàcilment -->
<resources>
  <string name="aboutTxt">Quant a...</string>
  <string name="app_name">Narinand</string>
  <string name="appDescription">Treball final de carrera</string>
  <string name="author">\u00A9 Xavier Bach-Esteve Burch</string>
  <string name="awayAlarmMessage">T'has allunyat de la ruta marcada</string>
  <string name="awayAlarmTitle">Fora de ruta</string>
  <string name="cancel">Cancel·la</string>
  <string name="captureDefaultDir">/sdcard/Narinand/gpx</string>
  <string name="captureDir">Ubicació de fitxers</string>
  <string name="captureDirDesc">Ubicació de la memòria externa on es guarden els fitxers GPX</string>
  <string name="captureDist">Distància de captura</string>
  <string name="captureDistDesc">Distància entre captures de punts des del GPS</string>
  <string name="captureTime">Temps de captura</string>
  <string name="captureTimeDesc">Temps entre captures de punts des del GPS</string>
  <string name="commentLbl">Comentaris</string>
  <string name="course">Curs 2010-11</string>
  <string name="cronoLbl">Temps: </string>
  <string name="current_dir">.</string>
</resources>
```

Il·lustració 52. Fragment del fitxer Strings.xml

L'estructura d'aquest fitxer és molt senzilla. Es defineix un nom pel recurs i el valor que ha de tenir, en aquest cas textos que apareixen en algun dels elements de l'aplicació.

Aquests recursos són accessibles des de l'aplicació mitjançant la classe que es crea automàtica i dinàmicament anomenada R.java. Així doncs, per a obtenir el nom de l'aplicació des de qualsevol classe, només cal fer la següent crida:

```
String appName = getString(R.string.app_name);
```

Il·lustració 53. Exemple de lectura d'un recurs

#### 4.4.5.2 Fitxer de recursos Arrays.xml

En aquest fitxer s'hi defineixen conjunts de dades que han de formar part d'un mateix element. En aquest projecte s'ha aprofitat per definir-hi els diversos valors que han de tenir les llistes desplegable del menú de preferències, i també les opcions del menú contextual que es construeix a l'activitat utilitzada per veure la llista de rutes guardades.

Per a cadascun dels elements amb diverses opcions per escollir s'han creat dues llistes, una amb els textos que mostra per escollir –Options-, i l'altra amb els valors –Values- corresponents per a cada elecció.

De la mateixa manera, s'ha creat un element que conté les diverses opcions del menú contextual. Els menús contextuais apareixen després de mantenir la pressió sobre un element de la llista mostrada per pantalla durant dos segons aproximadament.

El comportament de cada opció és tractat dins la lògica de negoci de l'activitat corresponent, ja que aquest fitxer de recursos només s'encarrega de definir la presentació.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Aquest fitxer defineix les opcions dels elements de preferències que -->
<resources>
  <string-array name="captureTimeOptions">
    <item>Automàtic (recomanat)</item>
    <item>2 segons</item>
    <item>5 segons</item>
    <item>10 segons</item>
  </string-array>

  <string-array name="captureTimeValues">
  </string-array>

  <string-array name="captureDistOptions">
  </string-array>

  <string-array name="captureDistValues">
  </string-array>

  <string-array name="maxDistAlarmOptions">
  </string-array>

  <string-array name="maxDistAlarmValues">
  </string-array>

  <!-- Opcions de menú contextual que apareix amb les rutes guardades -->
  <string-array name="loadMenu">
    <item>Carrega</item>
    <item>Exporta a GPX</item>
    <item>Esborra</item>
  </string-array>
</resources>
```

Il·lustració 54. Fitxer Arrays.xml

#### 4.4.5.3 Fitxer de recursos Styles.xml

Aquest fitxer s'utilitza per definir els estils de presentació que s'utilitzaran per als diversos elements definits en els fitxers de *layout*.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Aquest fitxer defineix els diversos estils de text que utilitza l'aplicació -->
<resources>
  <style name="aboutText">
    <item name="android:textColor">#f9faae</item>
    <item name="android:layout_gravity">center</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
  </style>
  <style name="BottomBar" parent="@android:TextAppearance.Small">
  </style>
  <style name="FileExplorer" parent="@android:TextAppearance.Medium">
  </style>
  <style name="MainMenuText">
    <item name="android:textColor">#8cbc4a</item>
    <item name="android:textStyle">bold</item>
    <item name="android:layout_gravity">center</item>
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
  </style>
  <style name="RouteExplorer" parent="@android:TextAppearance.Small">
  </style>
  <style name="Title" parent="@android:style/WindowTitle">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:textColor">#8cbc4a</item>
    <item name="android:textStyle">bold</item>
    <item name="android:textSize">30sp</item>
    <item name="android:shadowColor">#ffffff</item>
    <item name="android:shadowRadius">1</item>
    <item name="android:shadowDx">2</item>
    <item name="android:shadowDy">2</item>
    <item name="android:layout_centerInParent">true</item>
  </style>
</resources>
```

Il·lustració 55. Fitxer Styles.xml

En *Android* els estils poden utilitzar herència, per la qual cosa a partir d'un estil base es poden modificar diversos aspectes per obtenir un resultat diferent. Això s'aconsegueix mitjançant l'ús de l'etiqueta *parent* on s'indica l'estil des del qual s'hereta.

En un estil es poden definir tot tipus de paràmetres sobre la presentació per evitar haver-los de replicar en cada element de cada *layout*. Així doncs l'ús d'estils facilita la feina de modificació de tipus de visualització, ja que des d'un sol lloc es pot modificar el tipus, color, mida... de lletra de totes les activitats que formen una aplicació.

#### 4.4.6 Activitat *Narinand*

Es tracta de l'activitat *-Activity-* principal de l'aplicació. És la que mostra la pantalla que l'usuari veu un cop llança el programa, i conté el menú principal amb les diverses tasques que l'usuari pot dur a terme.

##### 4.4.6.1 Layout

Com ja s'ha comentat anteriorment, s'ha definit un *layout* per a cadascuna de les activitats que formen part d'aquest projecte. En el cas del menú principal s'ha definit a través d'un *TableLayout* que permet ubicar diversos objectes de manera tabular.

```
<TableLayout android:id="@+id/menuTable"
  android:layout_width="fill_parent" android:layout_height="fill_parent"
  android:paddingTop="20px" android:stretchColumns="*"
  android:layout_below="@id/titleBar">
  <TableRow android:id="@+id/rowImages1" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:gravity="center">
    <ImageButton android:id="@+id/newRouteBtn" android:src="@drawable/newroute"
      android:layout_width="wrap_content" android:layout_height="wrap_content"
      android:layout_gravity="center">
    </ImageButton>
    <ImageButton android:id="@+id/loadRouteBtn" android:src="@drawable/loadroute"
      android:layout_width="wrap_content" android:layout_height="wrap_content"
      android:layout_gravity="center">
    </ImageButton>
  </TableRow>
```

Il·lustració 56. Fragment de *TableLayout* del fitxer *main.xml*

En el fragment de codi mostrat es pot veure com es defineixen les files mitjançant les etiquetes *TableRow*, i cadascun dels elements que s'inclouen dins d'aquestes files formen una columna.

També es pot veure que es defineix l'alineació de les columnes mitjançant l'atribut *layout\_gravity*.

##### 4.4.6.2 Lògica de negoci

Aquesta activitat ha de mostrar les diverses opcions que es poden dur a terme pel programa d'una manera clara i amigable per a l'usuari. Així doncs, l'única funcionalitat que ha de tenir és la de mostrar un menú d'imatges i gestionar els events de clic que es puguin donar.

En primer lloc es defineix quina és la vista que cal mostrar fent referència a la classe que conté tots els recursos definits, en aquest cas el *layout* anomenat *main*.

Per cadascun dels botons d'aquesta activitat es crea un objecte de tipus *View.OnClickListener* que controla els events de clic sobre aquests.

En la implementació de l'event es crea un objecte de tipus *Intent* que s'utilitza per engegar l'activitat que li correspon.

Cal fixar-se en què la definició del botó es cerca mitjançant la classe "R", que conté referències a tots els recursos definits per a l'aplicació.

```
// Es mostra la vista main
setContentView(R.layout.main);

// Control del botó de crear una nova ruta
ImageButton btnNew = (ImageButton) findViewById(R.id.newRouteBtn);
// Es crea un "clickListener" per escoltar events de click sobre el botó
btnNew.setOnClickListener(new View.OnClickListener() {
    /**
     * Override del mètode onClick. Es crida al moment de prémer sobre
     * el botó
     *
     * @param View
     *          v
     */
    @Override
    public void onClick(View v) {
        // S'obre l'activitat NewRoute
        Intent myIntent = new Intent();
        myIntent.setClassName("edu.uoc.narinand",
            "edu.uoc.narinand.NewRoute");
        startActivity(myIntent);
    }
});
```

Il·lustració 57. Codi per mostrar un botó amb imatge a la vista principal

#### 4.4.6.3 Captures de pantalla

La pantalla principal de l'aplicació té l'aspecte que es pot veure en la il·lustració 58.

Mitjançant el seu fitxer de *layout* s'ha definit una barra de títol on hi ha la icona del programa i el nom de l'aplicació.

Per sota d'aquestes dades hi ha les quatre opcions que permet el programa tabulades en matriu de 2x2.

Es tracta de quatre botons de tipus imatge amb un text de descripció a sota.

D'aquesta manera l'usuari es fa una idea clara de quines opcions té i què significa cadascuna d'elles.



Il·lustració 58. Pantalla principal

#### 4.4.7 Activitat *NewRoute*

Aquesta activitat és la responsable de la major part de funcionalitat del programa, ja que mostra el mapa i les diverses possibilitats d'interactuar-hi, ja sigui mostrant la ruta carregada, la posició actual, el camí recorregut, les dades de temps, velocitat, alçada...

##### 4.4.7.1 Layout

Es tracta d'una organització de tipus relatiu –*RelativeLayout*– on s'hi col·loquen diversos objectes. El principal és el mapa de *Google Maps*. Referent a aquest mapa es pot veure l'atribut *apiKey* –distorsionat per conservar la privacitat– que conté una clau que s'ha de demanar a *Google* per poder utilitzar les seves vistes de mapes. El procés de registre és molt senzill i s'obté la clau de forma instantània. Sense aquest element no és possible utilitzar els mapes d'aquesta companyia.

```

<ImageButton android:id="@+id/stop" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_alignParentRight="true"
    android:src="@drawable/stop" />
<TextView android:id="@+id/cronoTxt"
<TextView android:id="@+id/speedTxt" android:layout_toLeftOf="@id/stop"
<TextView android:id="@+id/distanceLbl"
<TextView android:id="@+id/distanceTxt"
    android:layout_toLeftOf="@id/stop" android:paddingRight="5px"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_below="@id/speedTxt" style="@style/BottomBar"
    android:text="0 m">
</TextView>
<TextView android:id="@+id/elevationLbl"
<TextView android:id="@+id/elevationTxt"
    android:layout_toLeftOf="@id/stop" android:paddingRight="5px"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_below="@id/distanceTxt" style="@style/BottomBar"
    android:text="0 m">
</TextView>
</RelativeLayout>
<com.google.android.maps.MapView
    android:id="@+id/map" android:enabled="true" android:clickable="true"
    android:apiKey="..."
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:layout_above="@id/newrouteButtons" />
</RelativeLayout>
    
```

Il·lustració 59. Fragment del fitxer *newroute.xml*

A més del mapa hi ha altres elements, com ara les etiquetes que mostres informació de la velocitat, alçada... –*TextView*–, o els botons per controlar la ruta.

##### 4.4.7.2 Lògica de negoci

Aquesta activitat conté la major part de la lògica de negoci de tota l'aplicació, ja que és on es desenvolupen la majoria d'accions.

```

if (locationListener != null) {
    // Es registra el listener amb el locationManager per rebre
    // actualitzacions de la posició
    locationManager.requestLocationUpdates(Utils.provider,
        GlobalData.getCaptureTime(), GlobalData.getCaptureDist(),
        locationListener);
}
    
```

Il·lustració 60. Registre de *locationManager*



Una part important és quan es registra el *locationManager* per rebre actualitzacions de posició des del dispositiu gps.

De la mateixa manera, es presenta un tros de codi en el qual, després de rebre la informació de la localització gps, s'utilitza per actualitzar la informació visual: posició actual, línia de ruta seguida...

```
// Es dibuixa la marca que indica la posició actual
Drawable pin = this.getResources().getDrawable(R.drawable.pin);
// Si ja existeix s'esborra
if (mapOverlays.contains(currentPosItOverlay))
    mapOverlays.remove(currentPosItOverlay);
mapController.animateTo(point);
// Es torna a afegir actualitzada
currentPosItOverlay = new NarinandItemizedOverlay(pin);
currentPosItOverlay.addNewItem(point, "", "");
mapOverlays.add(currentPosItOverlay);

// Si és el primer punt es dibuixa el símbol d'inici de ruta
if (isFirstLocation) {
    lastLocations = new ArrayList<Location>();
    Drawable start = this.getResources().getDrawable(
        R.drawable.start);
    if (mapOverlays.contains(startPosItOverlay))
        mapOverlays.remove(startPosItOverlay);
    mapController.animateTo(point);
    startPosItOverlay = new NarinandItemizedOverlay(start);
    startPosItOverlay.addNewItem(point, "", "");
    mapOverlays.add(startPosItOverlay);
    isFirstLocation = false;
}

// S'afegeix el punt a la llista de darreres localitzacions
lastLocations.add(location);

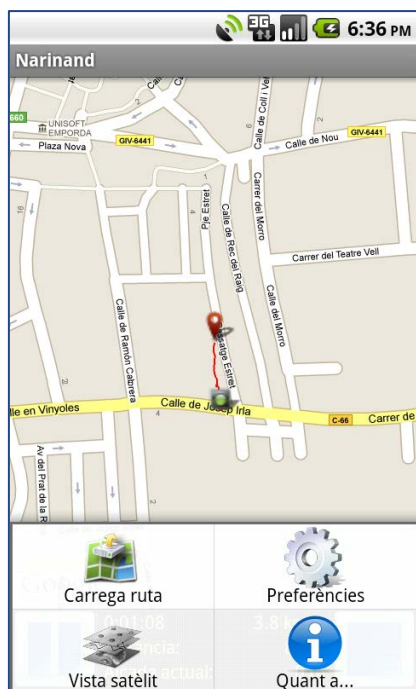
// S'afegeix el punt a la capa de la ruta actual
if (routeOverlay.isSetSegment()) {
    routeOverlay.addPoint(point, routeOverlay.getLastSegmentId());
} else {
    routeOverlay.addPoint(point, -1);
}
// Es dibuixa la ruta recorreguda fins aquest moment
routeOverlay.draw(canvas, mapView, false);
```

#### Il·lustració 61. Gestió de les capes del mapa

Hi ha moltes més funcionalitats implementades en aquesta activitat, que es poden consultar a fons, amb els comentaris pertinents, en el codi font.



### 4.4.7.3 Captures de pantalla



Il·lustració 63. Pantalla NewRoute amb menú



Il·lustració 62. Pantalla NewRoute amb vista aèria

En les imatges anteriors es pot veure la vista estàndard del mapa, amb les opcions de menú desplegadas, i la vista aèria del mapa amb les dades referents a la ruta actual.

A sota es pot veure el diàleg que demana si es vol guardar la ruta, i els ítems d'inici i final de ruta dibuixats un cop acabada.



Il·lustració 65. Pantalla NewRoute. Diàleg de guardar fitxer



Il·lustració 64. Pantalla NewRoute. Ruta finalitzada

#### 4.4.8 Activitat *LoadRoute*

Aquesta activitat és la que permet a l'usuari indicar la ubicació d'un fitxer gpx per a poder-lo carregar a l'aplicació.

S'encarrega d'iniciar l'activitat per explorar el sistema de fitxers, i també de carregar un fitxer gpx concret i iniciar l'activitat de fer una ruta.

##### 4.4.8.1 Layout

Es tracta d'una organització de tipus relatiu –*RelativeLayout*– on s'hi col·loquen entre d'altres, els botons d'acció i un quadre de text per escriure la ruta del fitxer gpx.

```
<RelativeLayout android:id="@+id/loadRouteLayout"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:paddingTop="20px" android:layout_below="@id/titleBar">
    <RelativeLayout android:id="@+id/searchRouteLayout"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:layout_below="@id/loadRouteLayout">
        <TextView android:id="@+id/fileLabel" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:padding="5px"
            android:text="@string/fileTxt" android:layout_centerVertical="true">
        </TextView>
        <EditText android:id="@+id/fileTxt" android:layout_height="wrap_content"
            android:layout_width="fill_parent" android:layout_toRightOf="@id/fileLabel"
            android:layout_centerVertical="true">
        </EditText>
    </RelativeLayout>
    <Button android:id="@+id/searchFileBtn" android:text="@string/searchFile"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_below="@id/searchRouteLayout"
        android:layout_alignRight="@id/searchRouteLayout">
    </Button>
```

Il·lustració 66. Fragment del fitxer loadroute.xml

Es pot observar que per entrar text s'utilitza un element de tipus *EditText* que es pot referenciar des del codi a través de l'identificador amb el qual es defineix aquí.

##### 4.4.8.2 Lògica de negoci

Un dels aspectes importants que utilitza aquesta activitat, és el fet de poder passar informació entre diverses activitats a través del que s'anomena *bundle*.

En *Android*, l'objecte *bundle* emmagatzema dades en format clau-valor. Els valors poden ser de diversos tipus –String, int, double, float...–.

Mitjançant aquest element es poden passar tot tipus de dades des d'una activitat a l'altra amb el que s'anomenen *Extras*.

En el fragment de codi següent queda molt clar el funcionament d'aquest objecte per a enviar dades de l'activitat actual a la que s'està cridant.

Com que es tracta de l'activitat *LoadRoute*, en aquest cas es passa un String amb el valor de la ruta que s'ha introduït en el quadre de text, i que per tant s'ha d'anar a cercar.

```
@Override
public void onClick(View v) {
    // Es comprova que el directori entrat sigui en la ruta
    // predefinida
    if (!filePath.getText().toString()
        .startsWith(GlobalData.getCaptureDir())) {
        // En cas negatiu es mostra una alerta dient-ho
        showDirDialog();
    } else {
        // En cas positiu es comprova que existeix el directori on
        // s'ha de guardar. En cas que no existeixi es crea
        Utils.chekDirToSave(filePath.getText().toString());
        // S'obre una nova activitat per navegar per sistema de
        // fitxers
        Intent myIntent = new Intent();
        myIntent.setClassName("edu.uoc.narinand",
            "edu.uoc.narinand.FileBrowser");
        // Es passa per paràmetres a la nova activitat el directori
        // escollit
        Bundle bundle = new Bundle();
        bundle.putString("dir", filePath.getText().toString());
        myIntent.putExtras(bundle);
        // S'inicia l'activitat esperant que retorni un resultat
        startActivityForResult(myIntent, 0);
    }
}
```

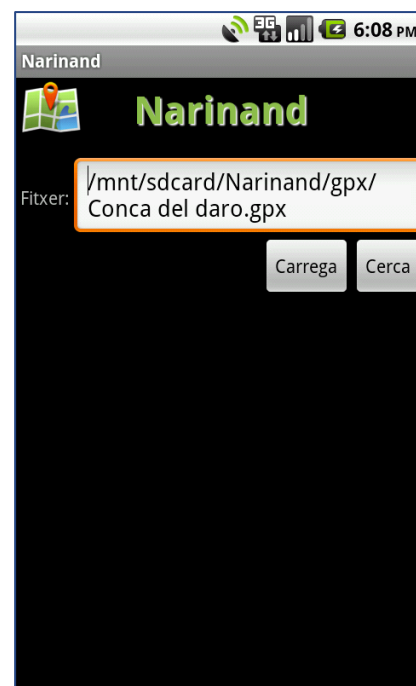
Il·lustració 67. Pas de paràmetres entre activitats

#### 4.4.8.3 Captures de pantalla

En la imatge adjunta es pot veure la senzillesa d'aquesta pantalla, on només hi ha una barra amb el logotip i nom de l'aplicació, i un quadre de text per poder escriure una ruta on cercar fitxers gpx.

Mitjançant els dos botons de text se li dóna la funcionalitat, un per obrir l'explorador de fitxers en la posició indicada en el quadre de text, i l'altre per carregar el fitxer indicat en el quadre de text.

D'aquesta manera es té un mòdul per a gestionar i carregar els fitxers gpx.



Il·lustració 68. Pantalla LoadRoute

#### 4.4.9 Activitat *FileBrowser*

Aquesta activitat s'encarrega de mostrar una llista de directoris i fitxers i permet que l'usuari hi pugui navegar per accedir al fitxer desitjat.

##### 4.4.9.1 Layout

Es tracta del *layout* més senzill de tot el projecte, ja que només conté un element. Es tracta d'un quadre de text o *EditText*, que ha de contenir la descripció dels directoris o fitxers dels quals s'està navegant.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Definició de l'element que forma part de cadascun dels elements de la
llista de directoris -->
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
style="@style/FileExplorer" />
```

Il·lustració 69. Fitxer *file\_row.xml*

##### 4.4.9.2 Lògica de negoci

En aquesta part s'ha desenvolupat tota la lògica de control de navegació pel sistema de fitxers.

En aquest cas, però cal fixar-se especialment en la manera de vincular la llista de directoris i fitxers amb el *layout* presentat en l'apartat anterior. Aquest lligam és molt senzill de dur a terme, simplement amb l'ús del que s'anomena *ListAdapter*.

```
ArrayAdapter<String> directoryList = new ArrayAdapter<String>(this,
R.layout.file_row, this.directoryEntries);
this.setAdapter(directoryList);
```

Il·lustració 70. Fragment de codi per carregar un *ListAdapter*

Primer es crea un *ArrayAdapter* en el qual se li indica a quin *layout* ha de vincular-se, i una llista de cadenes per a vincular-hi. Tot seguit s'assigna aquest *array* a el *ListAdapter*.

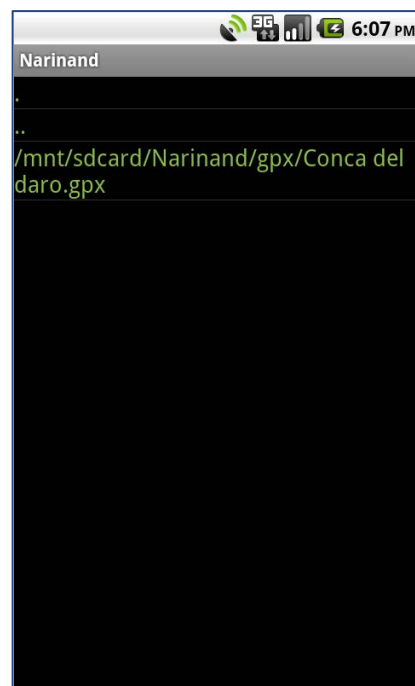
##### 4.4.9.3 Captures de pantalla

La pantalla és molt simple. Sempre conté els directoris "." i "..", aquest darrer mentre no s'estigui al directori arrel.

A més d'aquests dos, es va carregant amb la llista de directoris i fitxers que hi ha a cada moment.

Quan l'usuari clica sobre un directori, s'entra a aquest nivell i es mostra la llista de fitxers i directoris que conté.

Quan es clica sobre un fitxer, es retorna a l'activitat de càrrega de ruta passant-li la ruta del fitxer completa.



Il·lustració 71. Pantalla del navegador de fitxers

#### 4.4.10 Activitat ViewRoutes

Aquesta activitat permet navegar per la llista de rutes guardades amb el dispositiu, i interactuar-hi. De la mateixa manera que l'explorador de fitxers mostra una llista de directoris, aquesta activitat mostra una llista de rutes guardades.

Aquestes rutes les va a cercar a la base de dades del programa i n'extreu les dades per poder-les mostrar a l'usuari.

##### 4.4.10.1 Layout

En aquest cas s'han creat dos *layouts* diferents per a aquesta activitat.

El primer conté un element de tipus *ListView* que és el que conté la llista de les rutes, lligada a través d'un *ListAdapter*. També conté un element de text que es mostra quan no hi ha cap dada per carregar.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Definició de la vista per a mostrar les rutes guardades al dispositiu -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ListView android:id="@+id/list" android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <TextView android:id="@+id/empty" style="@style/RouteExplorer"
        android:text="@string/noData" />
</LinearLayout>
```

Il·lustració 72. Fitxer viewroutes.xml

Per altra banda, s'ha creat un altre fitxer on es detallen les diverses dades que ha de mostrar en cada element de la llista definida en el *layout* anterior.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Definició dels elements de cadascuna de les rutes guardades que es mostren
des de l'activitat de veure les rutes guardades -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/name" style="@style/RouteExplorer" />
    <TextView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/date" android:layout_below="@id/name" style="@style/RouteExplorer" />
    <TextView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/desc" android:layout_below="@id/date" style="@style/RouteExplorer" />
</RelativeLayout>
```

Il·lustració 73. Fitxer viewrouteselements.xml

Es pot veure que per cada element de la llista hi haurà tres elements de tipus text que contindran diverses dades de la ruta: nom, data i descripció.

Així doncs, mitjançant aquesta estructura és possible carregar un *ListAdapter* amb diverses dades a la vegada per construir una llista complexa.

##### 4.4.10.2 Lògica de negoci

Una de les parts importants d'aquesta activitat, i que la diferencia de les altres, és l'ús que fa del menú contextual. Quan un usuari prem durant uns segons sobre un element de la llista ha d'aparèixer un menú amb opcions relatives a l'element escollit.



```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {
    // Si la vista és la llista d'elements
    if (v.getId() == R.id.list) {
        // Es defineix el títol del menú contextual
        menu.setHeaderTitle(R.string.options);
        // Es defineixen les diverses opcions de menú
        String[] menuItems = getResources()
            .getStringArray(R.array.loadMenu);
        // S'afegeixen totes les opcions al menú contextual
        for (int i = 0; i < menuItems.length; i++) {
            menu.add(Menu.NONE, i, i, menuItems[i]);
        }
    }
}
```

Il·lustració 74. Creació del menú contextual

Mitjançant l'ús dels fitxers xml per definir el menú, és possible mostrar el menú amb la funció que es mostra.

Cal fixar-se que s'obtenen les opcions del fitxer de recursos, i a partir d'això es poden muntar totes les opcions d'aquest menú.

Per que fa a la gestió dels events del menú es pot accedir a la opció escollida per l'usuari mitjançant l'ús de l'event *onContextItemSelected*, que dona informació sobre quin dels elements s'ha clicat i permet desenvolupar la lògica d'actuació.

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    // S'obté informació del menú contextual
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item
        .getMenuInfo();
    // S'obté l'id de l'item que s'ha clicat del menú
    int menuItemIndex = item.getItemId();

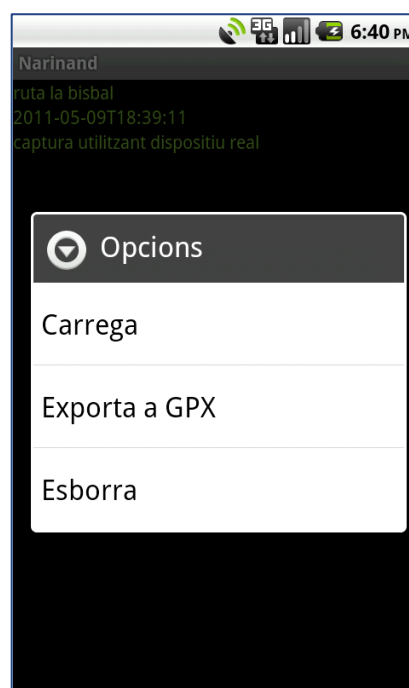
    // Es crea una instància de la classe de control de la base de dades
    DictionaryOpenHelper doh = new DictionaryOpenHelper(Utils.context);
    // Segons quina opció del menú s'ha escollit...
    switch (menuItemIndex) {
        case 0:
            // Opció de carregar ruta
    }
}
```

Il·lustració 75. Gestió d'events del menú contextual

#### 4.4.10.3 Captures de pantalla

En la captura de pantalla es pot observar que en el fons es veu la llista d'elements amb les diverses dades que s'han definit al *layout* –nom, data i descripció–.

En primer pla es mostra el menú contextual que s'ha creat amb les tres opcions definides.



Il·lustració 76. Menú contextual

#### 4.4.11 Activitat *Preferences*

A través d'aquesta activitat es poden consultar i/o canviar els ajustos de l'aplicació. Així doncs, qualsevol usuari pot personalitzar-se els diversos paràmetres mitjançant un senzill menú.

##### 4.4.11.1 Layout

Ja s'ha comentat anteriorment que les aplicacions *Android* tenen una manera molt simple de definir menús de preferències. Només definint el fitxer xml amb cadascun dels elements que es volen permetre ajustar, el sistema ja s'encarrega de mostrar-ho correctament.

```
<EditTextPreference android:key="PREF_LOG_DIR"
  android:title="@string/logDir" android:summary="@string/logDirDesc"
  android:persistent="true" android:defaultValue="@string/logDefaultDir" />
<ListPreference android:key="PREF_MAX_DIST_ALARM"
  android:title="@string/maxDistAlarm" android:summary="@string/maxDistAlarmDesc"
  android:entries="@array/maxDistAlarmOptions" android:entryValues="@array/maxDistAlarmValues"
  android:dialogTitle="@string/maxDistAlarm" android:persistent="true"
  android:defaultValue="20" />
<CheckBoxPreference android:key="PREF_VIBRATION_ALARM"
  android:title="@string/vibrationTitle" android:summary="@string/vibrationSummary"
  android:defaultValue="true" />
```

Il·lustració 77. Fragment del fitxer Settings.xml

Per a l'aplicació *Narinand* s'han utilitzat tres tipus d'elements, els de text lliure, els de llista de preferències, i els de *checkbox*.

##### 4.4.11.2 Lògica de negoci

En aquest cas la lògica de negoci és molt simple, ja que el sistema s'encarrega

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // S'obtenen les preferències des de la definició als recursos
    addPreferencesFromResource(R.xml.settings);
}
```

de tota la gestió.

Il·lustració 78. Codi font per carregar pantalla de preferències

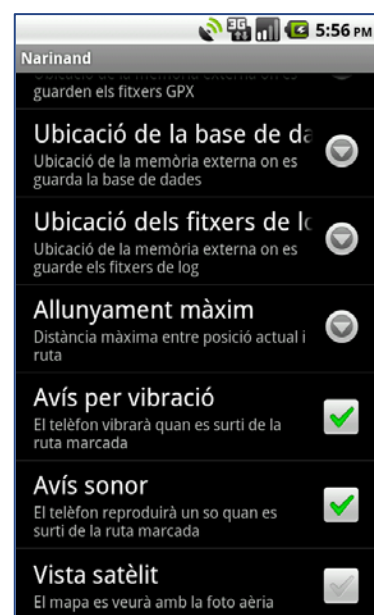
Simplement cal indicar-li a l'activitat quin és el fitxer de recursos que conté les opcions de preferències. Això es fa en l'event de creació de l'activitat.

##### 4.4.11.3 Captures de pantalla

En aquest cas es tracta de la típica pantalla de preferències que es mostra en qualsevol aplicació del sistema *Android*.

En la imatge es pot veure clarament que hi ha diversos tipus d'element que es comporten de manera diferent al rebre l'event de clic.

Aquest comportament és controlat pel sistema i no cal desenvolupar-ne cap lògica de negoci.



Il·lustració 79. Pantalla de preferències



#### 4.4.12 Altres elements

A més dels elements que estenen la classe *Activity*, n'hi ha d'altres que formen part de l'aplicació. Tot seguit s'enumeren els principals.

##### 4.4.12.1 Classe DictionaryOpenHelper

Mitjançant aquesta classe s'ha implementat la persistència. D'aquesta manera es pot crear, llegir i escriure la base de dades que conté la informació necessària de l'aplicació.

Com ja s'ha explicat, *Android* utilitza el sistema gestor de bases de dades *SQLite*. Així doncs per interaccionar-hi s'han d'executar sentències SQL.

```
/** Sentència SQL per a crear la taula de rutes o tracks */  
private static final String TRACKS_TABLE_CREATE = "CREATE TABLE "  
    + TRACKS_TABLE_NAME + " (" + trackId  
    + " INTEGER PRIMARY KEY AUTOINCREMENT, " + date + " TEXT, " + name  
    + " TEXT, " + cmt + " TEXT, " + desc + " TEXT" + ");";
```

Il·lustració 80. Sentència SQL de creació de taula TRACKS

Així doncs, per accedir o modificar qualsevol de les dades es construeixen sentències SQL que realitzen una acció o retornen un cursor que apunta als resultats per poder-hi interactuar.

```
public Cursor getSavedRoutes(SQLiteDatabase db) {  
    String sql = "select " + trackId + " as _id, " + name + ", " + desc  
        + ", " + date + " from " + TRACKS_TABLE_NAME + " ORDER BY "  
        + date + " desc;";  
    String params[] = {};  
    // Retorna els resultats de la consulta  
    return db.rawQuery(sql, params);  
}
```

Il·lustració 81. Mètode de consulta a la base de dades SQLite

##### 4.4.12.2 Classe MyRouteOverlay

Es tracta d'una classe molt important per a l'aplicació que s'ha desenvolupat, ja que és la que permet dibuixar la línia que segueix la ruta.

Aquesta classe estén la classe *Overlay* definida a les API de *Google* i permet, de forma senzilla dibuixar elements com a capes del mapa.

Per dibuixar elements utilitza els mètodes de l'objecte *Canvas*. En particular, com que per aquesta aplicació es necessita dibuixar la línia de ruta, s'utilitza el mètode *drawLine*.

Cal tenir en compte que el dispositiu obté els punts en format *GeoPoint* amb la corresponent latitud i longitud, i que cal fer la transformació a píxels per representar la posició correctament al mapa. D'això se n'encarrega l'objecte *Projection* de l'element *MapView*.

Així doncs, per a dibuixar la ruta es recorren els diversos punts que s'han recollit, calculant-ne la projecció, i es van unint amb línies ajudant-se amb el mètode adequat de la classe *Canvas*.

```
@Override
public void draw(Canvas canvas, MapView mapView, boolean shadow) {
    // S'obé la projecció del mapa (per a convertir punts geogràfics a punts
    // pintables sobre el mapa
    Projection projection = mapView.getProjection();

    // Es defineixen algunes propietats del Paint
    paint.setAntiAlias(true);
    paint.setFakeBoldText(true);

    if (shadow == false) {
        // Es recorre la llista de segments
        for (ArrayList<GeoPoint> trkseg : segments) {
            boolean first = true;
            Point myLastPoint = new Point();
            // Es recorre la llista de punts de cada segment
            for (GeoPoint point : trkseg) {
                Point myPoint = new Point();
                // Es calcula la projecció del GeoPoint cap a Point
                projection.toPixels(point, myPoint);
                if (!first) {
                    // Si no es tracta del primer punt es dibuixa una línia
                    // des del darrer a l'actual
                    canvas.drawLine(myLastPoint.x, myLastPoint.y,
                                    myPoint.x, myPoint.y, paint);
                    myLastPoint = myPoint;
                } else {
                    // Si és el primer punt no es dibuixa res
                    first = false;
                    myLastPoint = myPoint;
                }
            }
        }
    }
}
```

Il·lustració 82. Mètode per dibuixar una línia a un mapa

#### 4.4.12.3 Classe NarinandItemizedOverlay

Aquesta classe estén la classe *ItemizedOverlay* de les API de *Google*, i s'utilitza per mostrar elements gràfics mitjançant capes en un mapa.

Mitjançant l'ús d'aquesta classe és molt senzill mostrar elements gràfics sobre un mapa de *Google*.

```
public void addNewItem(GeoPoint location, String markerText, String snippet) {
    // Afegix l'item
    items.add(new OverlayItem(location, markerText, snippet));
    // Processa la vista
    populate();
}
```

Il·lustració 83. Mètode per afegir un ítem al mapa

Es van afegint els diversos ítems segons la seva posició.

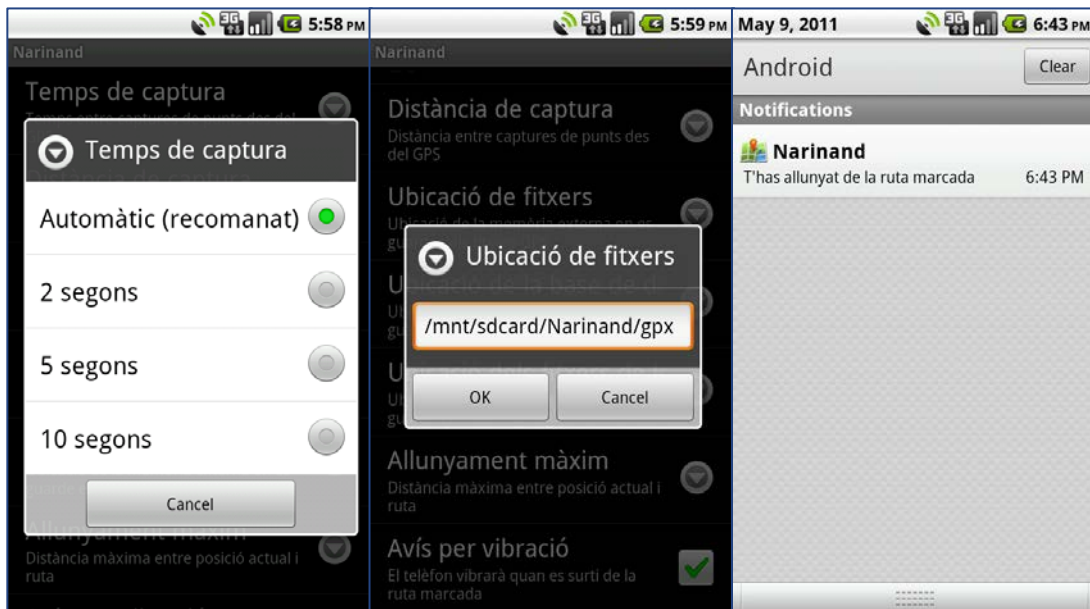
Quan s'instancia la classe se li indica quin és l'objecte que s'haurà de dibuixar en la capa mitjançant un *Drawable*. Els objectes d'aquest tipus estan definits al fitxer de recursos i són accessibles a través de la classe R.java.

```
public NarinandItemizedOverlay(Drawable defaultMarker) {  
    // Se li indica quin punt del marcador va sobre el punt  
    super(boundCenterBottom(defaultMarker));  
    // Es crea una nova llista d'items  
    items = new ArrayList<OverlayItem>();  
    // Processa la vista  
    populate();  
}
```

Il·lustració 84. Constructor de la classe NarinandItemizedOverlay

#### 4.4.13 Altres captures de pantalla

Tot seguit es mostren diverses captures de pantalla per fer-se una idea més acurada de les diverses funcionalitats implementades.



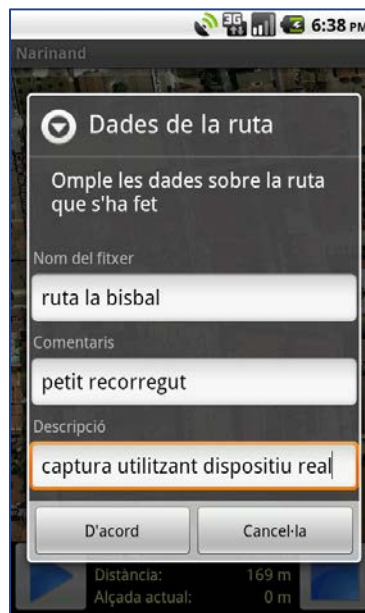
Il·lustració 87. Definició temps de captura

Il·lustració 86. Definició ubicació de fitxers

Il·lustració 85. Avís d'allunyament de ruta



Il·lustració 88. Informació de l'aplicació



Il·lustració 89. Entrada de dades de la ruta per guardar

## 5 CONCLUSIONS

La realització d'aquest projecte ha estat un molt bon final per a la carrera d'Enginyeria Informàtica, ja que amb el treball dedicat durant un semestre s'han aplicat i consolidat molts dels coneixements adquirits al llarg de tota la carrera.

S'han utilitzat i treballat els coneixements adquirits sobre la definició i gestió de projectes informàtics, ja que s'ha hagut de definir, planificar i executar un projecte complex. S'ha hagut de treballar també amb la gestió dels riscos, ja que com amb tot projecte informàtic, han sorgit incidències i dificultats mentre aquest s'executava.

Per altra banda s'han consolidat i posat en pràctica els coneixements adquirits sobre desenvolupament d'aplicacions i ús d'entorns de desenvolupament. Durant tot el projecte s'ha treballat amb l'IDE *Eclipse* i el llenguatge de programació *Java*.

Finalment, i principalment, s'ha conegut i estudiat a fons la tecnologia dels dispositius mòbils *Android*. Aquesta tecnologia està a l'ordre del dia i es preveu que tindrà un futur molt bo. D'aquesta manera s'ha pogut aprendre una tecnologia que pot ser molt important per al món empresarial.

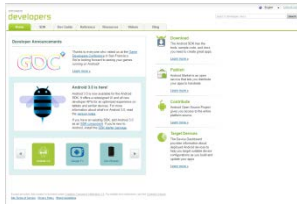
Així doncs, l'experiència ha estat completament enriquidora i motivant, a més d'ajudar a consolidar els coneixements adquirits durant els darrers anys i aprenent noves tecnologies.

Aquesta aplicació, però, podria millorar-se en diversos aspectes si es disposés de més temps per a dedicar-hi. Entre d'altres es podria incidir en els següents aspectes:

- Millora de la interfície gràfica per fer-la més amigable.
- Desenvolupament de la càrrega de tots els possibles continguts dels fitxers gpx.
- Mostrar més informació a l'usuari: alçada màxima de la ruta, velocitat mitjana...
- Millorar l'algorisme que discrimina els punts capturats pel gps per tenir un equilibri bo entre nombre de punts i fiabilitat d'aquests.

## 6 REFERÈNCIES

Per a la realització d'aquest projecte s'han consultat diverses fonts d'informació que es detallen tot seguit.



### **Android Developers**

Lloc web de referència per a desenvolupadors Android

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



### **The Busy Coder's Guide to Android Development**

CommonWare Library

Mark L. Murphy



### **Stackoverflow**

Fòrum de consultes sobre desenvolupament

<http://stackoverflow.com/>