

# **Escola Universitària Politécnica de Mataró**

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Enginyeria Tècnica de Telecomunicacions: Especialitat Telemàtica**

## **Aplicació per a dispositius mòbils en “Android”: Geoshows**

**Memòria**

**LLUIS DIAZ DE UGARTE  
PONENT: ANTONI SATUE**

PRIMAVERA 2011



**TecnoCampus  
Mataró-Maresme**

## **Dedicatòria**

Dedicat a totes les iniciatives de programari lliure arreu del món.



## **Agraïments**

A l'Alba, per els seus dissenys gràfics i el seu suport. A en Jordi, per introduir-me al món dels *smartphones*, i el món del software lliure. A en Josep, per fer-me entendre les estructures de la programació en general. Als meus pares, per ajudar-me a arribar on sóc en aquest moment.

## **Resum**

L'objectiu del projecte és crear una aplicació sota el sistema operatiu *Android* per a dispositius mòbils que permeti l'accés i escriptura a una base de dades, així com la utilització de eines que ens brinden les noves tecnologies en terminals, com la connectivitat a Internet i el posicionament mitjançant GPS. Sota aquestes premisses, ha nascut *GeoShows*, una aplicació capaç d'interactuar amb una base de dades i amb el posicionament d'esdeveniments en un mapa.

## **Resumen**

El objetivo del proyecto es crear una aplicación bajo el sistema operativo *Android* para dispositivos móviles que permita el acceso y escritura en una base de datos, así como la utilización de las herramientas que nos brindan las nuevas tecnologías en terminales, como la conectividad a Internet y el posicionamiento mediante GPS. Bajo estas premisas, ha nacido *GeoShows*, una aplicación capaz de interactuar con una base de datos y el posicionamiento de eventos en un mapa..

## **Abstract**

The objective of this project is to create an application under *Android's* operative system for mobile phones that allows access and write to a database, and to use new technologies on these mobile phones as Internet connection and GPS positioning. Under this premises, *GeoShows* has born, an application capable of interacting with a database, and map positioning of events.

## Índex.

Índex de figures.....	IV
Índex de taules.....	VI
Glossari de termes.....	VII
1. Objectius.....	1
1.1. Propòsit.....	1
1.2. Finalitat.....	1
1.3. Objecte.....	1
1.4. Abast.....	1
2. Android: sistema, recursos, i dispositius.....	3
2.1. Què és Android?.....	3
2.1.1. Context.....	3
2.1.2. Open source o freeware?.....	4
2.1.3. Arquitectura del sistema.....	4
2.2. Entorn de treball.....	6
2.2.1. Eines necessàries per al desenvolupament d'aplicacions.....	6
2.2.2. Software.....	7
2.3. Google APIs.....	8
2.3.1. Google Maps.....	8
2.3.2. Google SpreadSheets i GSS_Lib.....	9
3. Instal·lació de l'entorn de programació sota Windows 7 (x64).....	11
3.1. Sistema operatiu base.....	11
3.2. Instal·lació de la màquina virtual Java SE.....	11
3.3. Instal·lació de l'SDK de Android.....	14
3.4. Instal·lació de l'Eclipse.....	17
3.5. Aplicació de prova "HelloWorld".....	19
4. Estructura d'un projecte Android.....	21
4.1. Arbre del projecte.....	21
4.2. La classe Activity.....	22

4.3. La classe Intent. ....	25
4.4. Layouts.....	26
4.4.1. Linear Layout .....	26
4.4.2. Table Layout. ....	27
4.4.3. Absolute Layout. ....	27
4.4.4. Relative Layout .....	28
4.4.5. Frame Layout. ....	28
4.5. L'arxiu AndroidManifest.xml.....	29
5. Estudi i consideracions prèvies al disseny d'una aplicació per Android....	31
5.1. Aprenentatge. ....	31
5.2. Eines disponibles. ....	33
5.3. Necessitats. ....	34
5.4. Conclusió de l'estudi. ....	34
6. "GeoShows", el buscador de concerts. ....	37
6.1. Què és GeoShows?. ....	37
6.2. Funcionament de l'aplicació.. ....	38
6.2.1. Base de dades. ....	39
6.2.2. Obtenció de concerts .....	40
6.2.2. Pujar concerts .....	42
6.3. Us de l'aplicació. ....	43
6.3.1. Veure pròxims concerts.....	43
6.3.2. Informar d'un nou concert.....	44
7. Estudi econòmic. ....	45
7.1. Cost de l'aplicació .....	45
7.1.1. Cost del material.....	45
7.1.2. Costos de recursos humans. ....	45
7.1.3. Amortització equips, instrumental i software. ....	46
7.1.4. Despeses indirectes .....	46
7.1.5. Cost de l'aplicació.....	47
7.2. Preu de venda en el mercat .....	47
7.2.1. Càlcul del cost total per unitat.....	47
7.2.2. Càlcul de benefici.....	48

8. Conclusions.....	49
8.1. Punts forts i fluixos de l'aplicació .....	50
8.1.1. Punts forts.....	50
8.1.2. Punts fluixos.....	50
8.2. Perfil de l'usuari.....	51
8.3. Possibles millores. ....	51
9. Referències.....	53



## Índex de figures.

Fig. 2.2.3.1. Diferents components d'un sistema operatiu <i>Android</i> .....	5
Fig. 3.2.1. Llicència de <i>JDK</i> , versió 6.....	12
Fig. 3.2.2. Selecció de ruta i utilitats del <i>JDK</i> .....	13
Fig. 3.2.3. Selecció de ruta del <i>JRE</i> .....	13
Fig. 3.3.1. Descomprimir els fitxers de l'arxiu .zip a una carpeta coneguda .....	15
Fig. 3.3.2. Instal·lació de components a l' <i>SDK</i> .....	15
Fig. 3.3.3. Configuració de l'emulador.....	16
Fig. 3.3.4. Emulador d'Android.....	16
Fig. 3.4.1. Configuració del directori de treball.....	18
Fig. 3.4.2. Instal·lació de l' <i>ADT</i> i components.....	18
Fig. 3.4.3. Ruta de l' <i>SDK</i> de <i>Android</i> a <i>Eclipse</i> .....	19
Fig. 3.5.1. Aplicació <i>Hello, World</i> .....	20
Fig. 4.1.1. Arbre d'un projecte <i>Android</i> a <i>Eclipse</i> .....	22
Fig. 4.2.1. Cicle de vida d'una <i>Activity</i> .....	23
Fig. 6.1.1. Menú principal de <i>GeoShows</i> .....	37
Fig. 6.2.1. Funcionament de l'aplicació.....	38
Fig. 6.2.1.1 Pàgina del full de càlcul d'ordenació per dates.....	39
Fig. 6.2.1.2. Interacció amb la base de dades.....	40
Fig. 6.2.2.1 Obtenció de concerts.....	41

Fig. 6.2.3.1. Pujada de concerts.....	42
---------------------------------------	----

## **Índex de taules.**

Taula 7.1.2.1 Costos de recursos humans.....	45
Taula 7.1.3.1 Amortització de l'equipament.....	46
Taula 7.1.5.1 Cost total de l'aplicació.....	47
Taula 7.2.2.1. Càlcul del benefici.....	48

## **Glossari de termes.**

ADB	Android Debug Brigde
ADT	Android Development Tools
API	Application programming interface
click	Acció de prémer un botó del ratolí
frame	quadre o marc
GPL	General Public License
GPS	Sistema de posicionament global
IDE	Integrated Development Environment
kernel	nucli (parlant d'un sistema operatiu)
SDK	Source Development Kit
smartphone	Telèfon intel·ligent (nova generació de terminals mòbils)
tablet	tableta gràfica

# **1. Objectius.**

## **1.1. Propòsit.**

Crear una aplicació per a dispositius mòbils sota el sistema operatiu *Android*, capaç d'interactuar amb una base de dades externa, així com utilitzar les noves tecnologies que brinden aquests dispositius en base al posicionament GPS. Aprendre les estructures de la programació orientada a objecte per poder servir-se'n en un futur pròxim.

## **1.2. Finalitat.**

Aprendre a dissenyar una aplicació sota el sistema operatiu *Android* explotant algunes de les seves funcionalitats, i en conseqüència, crear un producte útil per als usuaris d'aquest sistema. Oferir els coneixements adquirits en favor de la creació de nous projectes en un futur

## **1.3. Objecte.**

L'aplicació "GeoShows" compilada i llesta per la seva instal·lació, capaç d'oferir un servei de cerca i publicació d'activitats musicals arreu del món. El codi font de l'aplicació, i una memòria guia per al desenvolupament d'aplicacions sota el sistema operatiu per a terminals mòbils *Android*, on s'hi detalla l'estructura del sistema operatiu, un manual d'instal·lació, aspectes més importants de la programació i aplicació d'exemple.

## **1.4. Abast.**

Aplicació per la plataforma *Android*, per tots els dispositius mòbils amb aquest sistema operatiu, sempre i quan disposin de connexió a Internet i a satèl·lits GPS. Pretén ser *open source*, així que l'abast de l'aplicació també és oferir el codi font a disposició d'altres desenvolupadors que se'n vulguin servir, o aprendre, mitjançant com a eina de difusió internet.



## 2. Android: sistema, recursos, i dispositius.

En aquest capítol es descriurà l'entorn en què s'ha treballat, així com les eines utilitzades i el necessari per entendre el contingut del projecte.

### 2.1. Què és Android?

#### 2.1.1. Context

Android és un sistema operatiu basat en Linux per a dispositius mòbils, com *smartphones* o *tablets*, escrit en *C/C++* i *Java*. Inicialment va ser pre-incubat per *Android Inc.*, firma que al final acabaria comprada per *Google* al 2005 (contractant al personal de desenvolupament del projecte i comprant els drets). És el producte principal de la *Open Handset Alliance (OHA)*, un grup empresarial compost principalment per desenvolupadors de *hardware*, *software* i fabricants.

La primera versió d'*Android* (v1.0) surt al mercat cap a finals del 2008, i des d'aleshores, diverses versions del sistema han estat alliberades fins a arribar a la v2.3.4 per dispositius mòbils, i la v3.1 per *tablets*.

Actualment existeixen més de 200.000 aplicacions per *Android* i s'estima que els seus usuaris en són uns 300.000. Per distribuir les aplicacions, s'utilitza *Android-Market*, un medi pel qual els desenvolupadors poden posar al mercat les seves aplicacions, amb un benefici del 70% dels ingressos que aquestes produeixin.

Al contrari que altres sistemes operatius (com *iPhone*), *Android* no requereix cap llicència al programador per desenvolupar les seves aplicacions. Tot el que es necessita per començar a desenvolupar aplicacions són el coneixement necessari per programar en el llenguatge *Java*, i obtenir el *Source Development Kit (SDK)* d'*Android*, proporcionat gratuïtament per *Google* des de la pàgina web oficial de desenvolupadors d'*Android*. Aquest kit és multi-plataforma, és a dir, el podem obtenir i emprar des de diferents sistemes operatius (*Apple*, *Linux*, *Windows*).

### 2.1.2. *Open source o freeware?*

*Android* és un software de codi obert que es distribueix sota una llicència de *Apache V2*.

Aquesta llicència explicita que:

- Es pot utilitzar per qualsevol propòsit, distribuir-lo, modificar-lo, i distribuir les modificacions.
- Les versions modificades no han de ser distribuïdes com a software lliure si no es vol, ja que no disposa de cap llicència *CopyLeft*.

Així doncs, es pot determinar que *Android* és un software de codi obert, però els diferents fabricants que el distribueixen tenen reservat el dret de protegir el seu codi i les seves modificacions.

No obstant, *Android* disposa d'un nucli (*kernel*) basat en l'arquitectura de *Linux*, el qual disposa de llicència *GPL*. Això fa que aquest nucli sigui software lliure a diferència de la resta del sistema operatiu.

### 2.1.3. *Arquitectura del sistema*

*Android* és un sistema operatiu basat en llibreries de *Java* com a nucli, en el que a través d'una màquina virtual anomenada *Dalvik* (capaç de compilar en temps d'execució) s'hi implementa un conjunt d'aplicacions orientades a objecte anomenat *Application Framework* que ofereixen molta facilitat al sistema de cara al programador (*Activity Manager*, *Window Manager*, *Location Manager*, ...). També s'han implementat llibreries externes com *SQLite* per bases de dades o *OpenGL* per gràfics en *2D* o *3D*. Tot això, pot funcionar gràcies a una capa de més baix nivell: un nucli de *Linux*. Aquest conté els *drivers* necessaris per poder fer funcionar els dispositius físics que incorpora el terminal, establint una comunicació entre ells i el software de més alt nivell. Agrupant per capes podem distingir-ne cinc de diferents:

- **Aplicacions:** Capa visible del sistema operatiu, on s'hi contenen les aplicacions que com a usuari fem servir (Telèfon, Contactes, Navegador, *GeoShows*...). Estan programades en *Java*



- **Application Framework:** Aplicacions orientades a objecte (manegadors) que ofereixen facilitat al programador simplificant tasques de manegament d'objectes (*Activity manager, Notification Manager...*)
- **Llibreries:** llibreries externes en C/C++ incloses al sistema operatiu per oferir diferents funcionalitats (*OpenGL, SQLite, libc, MediaFramework...*).
- **Android runtime:** Llibreries de *Java* que donen la major part de funcionalitats al sistema, i màquina virtual *Dalvik*, encarregada d'executar les aplicacions de forma òptima per sistemes amb poca memòria ( cada procés dispondrà de la seva pròpia màquina virtual).. Aquesta màquina virtual és la encarregada de comunicar amb el nucli
- **Linux Kernel:** Capa de comunicació del software amb els dispositius físics. Basat en la versió 2.6 del *kernel* de *Linux*.

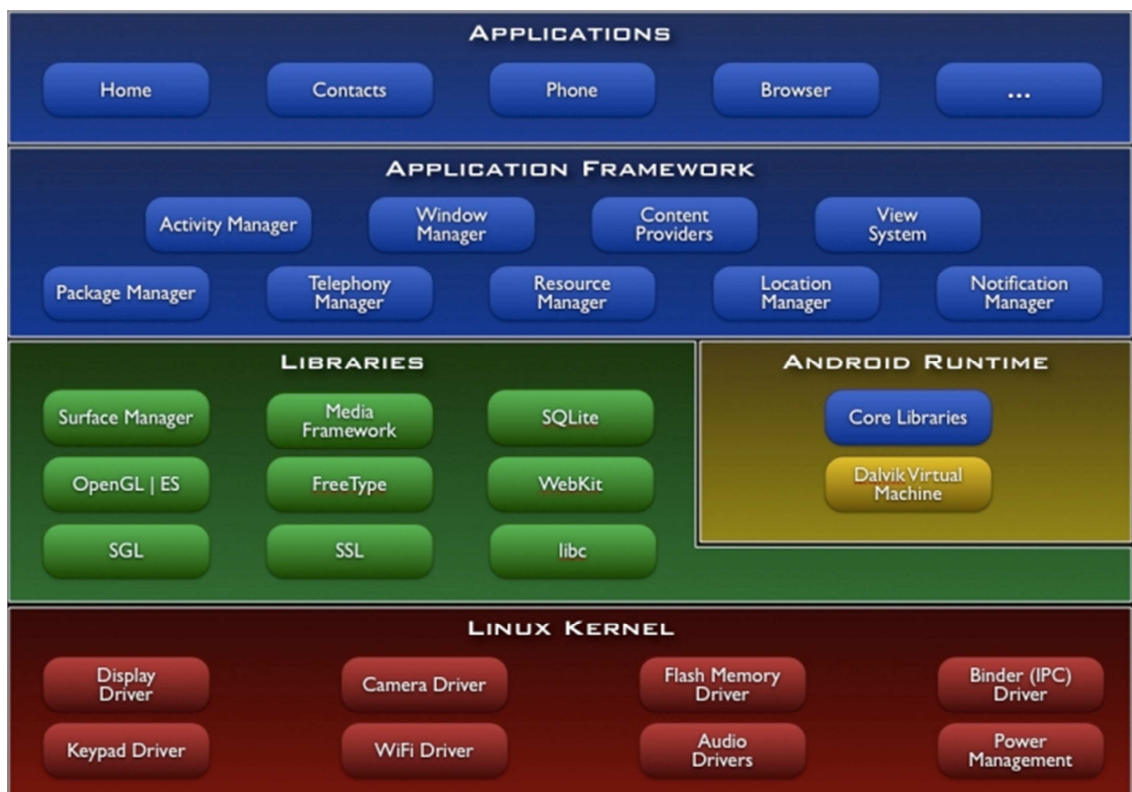


Fig. 2.2.3.1. Diferents components del sistema operatiu Android

Les aplicacions *d'Android* estan escrites en el llenguatge de programació *Java*, utilitzant recursos de disseny web com *XML*. El paquet d'eines *d'Android (SDK)* s'encarrega de compilar el codi en *Java* i crear l'arxiu amb extensió *\*.apk*, que el dispositiu mòbil usuari serà capaç d'instal·lar. Totes les aplicacions *d'Android* es regeixen per les següents característiques:

- Cada aplicació és com un usuari diferent dins del sistema *Linux*. Això proporciona independència entre aplicacions i ofereix així seguretat.
- Cada aplicació té assignat un identificador per atribuir permisos específics als diferents serveis que es poden oferir.
- Cada procés s'executa en una màquina virtual diferent, segons l'esquema de funcionament de la màquina virtual *Dalvik*. Així s'aconsegueix una forma òptima de manegar processos, i per tant, de gestió de la memòria.
- Cada aplicació executa el seu propi procés. Aquests processos són alliberats quan no es requereix la seva utilització després d'un període de temps, o quan es necessita de memòria per altres aplicacions.

## 2.2. Entorn de treball.

### 2.2.1. Eines necessàries per al desenvolupament d'aplicacions.

Per desenvolupar una aplicació en *Android* es necessiten diferents eines. Podem classificar aquestes en tres grups diferenciats: coneixements, maquinària, i software.

- **Coneixements:** és molt recomanable tenir experiències prèvies en un entorn de programació *Java*, així com en disseny web pels posicionaments gràfics. Tot i que la programació orientada a objecte i les ajudes d'auto-completat del software de programació suggereixen un fàcil inici en la programació, una mínima experiència en programació orientada a objecte pot ser de gran ajuda.

- **Maquinària:** Per programar sobre *Android* es necessitarà un ordinador, on poder instal·lar tot el software que ens permetrà desenvolupar aplicacions. Tot i que amb l'ajut d'emuladors es pot veure el resultat del codi (la aplicació funcionant), és molt recomanable disposar d'un terminal mòbil que suporti el sistema operatiu *Android* per verificar el correcte funcionament de les aplicacions.
- **Software:** Primer de tot, es necessitarà un entorn de programació on treballar. Aquest pot ser *Eclipse*, o *NetBeans*. Els dos són entorns de programació *Java* amb unes ajudes d'auto-completat de codi que faciliten molt la feina. Per utilitzar *Java*, s'ha de tenir instal·lada la versió 1.6 de *Java Sun Developers*. *Android* incorpora sobre *Java* les seves pròpies classes i repositoris, i aquestes ens les facilitarà el *SDK* de *Android* (Source Development Kit) a part de donar-nos la capacitat de crear emuladors de les diferents versions del sistema operatiu. De la comunicació entre l'emulador o terminal mòbil i l'entorn de programació se'n encarrega *Android Debug Bridge (ADB)*, gestionat si es vol per *Android Development Tools (ADT)*. L'*ADB* es pot descarregar des de la *SDK* de *Android* i *ADT* és un plug-in per *Eclipse*, que crea instàncies de *ADB*.

### 2.2.2. Llenguatges de programació.

Buscant una manera còmoda de desenvolupar aplicacions per als programadors, i que el llenguatge de programació fos molt versàtil i de fàcil instauració, es va decidir per part dels creadors de *Android* que el llenguatge de programació per aplicacions d'aquest seria *Java*. *Java* és un llenguatge molt popular, multi-plataforma (gràcies al seu funcionament en màquina virtual), i amb unes llibreries molt extenses, cosa que facilita molt la feina als programadors, i per altra banda, als creadors del sistema.

*Java* disposa de diverses versions de la seva màquina virtual: *SE (Standard Edition)*, *ME (Micro-Edition)*, i *EE (Enterprise Edition)*. Al principi, els creadors d'*Android* plantejaven utilitzar la versió *ME* de *Java* ja que era una versió orientada a dispositius mòbils amb pocs recursos de memòria, però això suposava haver de pagar llicències a *Sun Enterprises*, ja que aquesta versió no era gratuïta, i per altra banda la *SE* no s'adaptava a les necessitats d'*Android* pel que fa a recursos del sistema. Així doncs, *Google* i la *Open Handset Alliance* desenvolupen la màquina virtual *Dalvik*, optimitzada per dispositius amb poca memòria, basada en l'edició de *Java SE*, però amb bastantes diferències. *Android* no és compatible al cent per cent amb *Java SE*, ja que algunes biblioteques (*Swing*, *Awt*) no estan disponibles en *Android*, i per altra banda, el tema de la seguretat va ser del tot reinventat, ja que l'aïllament entre processos que proporciona la màquina virtual *Dalvik* (màquina que crea una màquina virtual per cada procés) ho suggeria.

Així doncs, *Java* i *Android* comparteixen llenguatge i biblioteques de programació, tot i distar en alguns aspectes.

Per altra banda, s'utilitza també *XML* com alternativa per declarar elements gràfics i posicionar-los d'una forma estàtica. Utilitzar *XML* per construir la interfície gràfica resulta molt més fàcil i còmode que realitzar-la dinàmicament amb *Java*.

## **2.3. Google APIs.**

### **2.3.1. Google Maps.**

Una font de llibreries i aplicacions molt important en *Android* és *Google APIs*. Entre les llibreries que ens ofereix *Google* per a *Android* s'hi troba *Google Maps API*.

La interfície de programació d'aplicacions (*API*) de *Google Maps* permet integrar *Google Maps* a altres aplicacions. Per fer-ho, és necessari obtenir una "Maps API Key" per l'aplicació en qüestió. Aquesta aplicació ha d'ésser degudament firmada amb un certificat digital (*MD5 Fingerprint*). Aquest certificat o firma digital pot ésser obtingut amb una aplicació anomenada "*Keytool*" que es pot trobar en l'*SDK* d'*Android*. A la pàgina oficial de *Google Android Developers* s'hi pot trobar una guia de com crear aquesta clau. Les llibreries de *Google Maps API* també es poden descarregar gratuïtament des de l'*SDK* de *Android*.

### **2.3.2. Google SpreadSheets i GSS\_Lib.**

*Google SpreadSheets* és una aplicació web de *Google Docs* que permet elaborar fulls de càlcul. Aquests poden ser molt útils, ja que estant a la web i tenint accés a lectura/escriptura remotament poden tractar-se com si fossin bases de dades.

*Gss\_Lib* és una llibreria per a *Android* que simplifica les funcions d'accés/escriptura a *Google SpreadSheets*. Agrupa funcions que permeten crear/esborrar fulls de càlcul, enumerar els fulls de càlcul residents en una conta, afegir/esborrar pàgines d'un full de càlcul, afegir/editar/esborrar registres d'una de les pàgines del full de càlcul, i llegir els registres continguts en les diferents pàgines d'un full de càlcul. El codi d'aquesta llibreria i la llibreria compilada es poden descarregar des de els repositoris oficials de Google de forma gratuïta.

Per utilitzar *Google SpreadSheets*, i per tant, *GSS\_Lib* és necessari disposar d'un compte a Google, on guardar els documents creats mitjançant Google Docs.



## **3. Instal·lació de l'entorn de programació sota Windows 7.**

### **3.1. Sistema operatiu base.**

Abans de començar a descarregar els instal·lables del software necessari per desenvolupar aplicacions en Android és necessari conèixer bé sobre quin sistema operatiu s'instal·larà l'entorn de programació. Existeixen diferents versions de cada software adequades al sistema operatiu amb el que es vol treballar. Si no s'instal·la la versió correcta pel sistema operatiu en qüestió és molt probable que s'experimentin problemes de compatibilitat, així que més val comprovar en quin sistema operatiu es treballa, i quina tecnologia fa servir (x86 o x64).

L'entorn de programació per Android és possible instal·lar-lo tant en Windows, com en Linux i Mac. Aquesta guia d'instal·lació pretén explicar com instal·lar l'entorn sota el sistema operatiu Windows 7 (x64)

### **3.2. Instal·lació de la màquina virtual Java SE.**

Java SE (Standart Edition) permet desenvolupar i executar aplicacions en Java tant en escriptoris com en servidors. Alguns dels programes que es necessiten per treballar en Android estan desenvolupats en Java, així que és necessari descarregar i instal·lar la versió sempre més actual de la seva pàgina oficial, apartat de descàrregues. Actualment, Android treballa amb la versió 1.6 de Android, així que és recomanable instal·lar aquesta versió o posterior.

Per baixar Java SE Development Kit (JDK) s'ha d'entrar a la pàgina següent:

- “<http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u25-download-346242.html>”

i descarregar la versió del JDK d'acord amb el sistema operatiu base que emprem. En el cas de tenir Windows 7 (x64) descarregarem l'arxiu [jdk-6u25-windows-x64.exe](#) i l'executem.

- Pas 1: Acceptar la llicència. Es llegeix el contracte de usuari final, i en cas d'acceptar totes les clàusules, fem *click* al botó d'acceptar (Fig. 3.2.1.).



Fig.3.2.1. Llicència de JDK, versió 6

- Pas 2: Escollir la ruta de la instal·lació del JDK i les eines que volem instal·lar. Deixant la ruta d'instal·lació per defecte n'hi ha prou. Es marca el camp *Development Tools* i es fa *click* a *Next* (Fig. 3.2.2.).



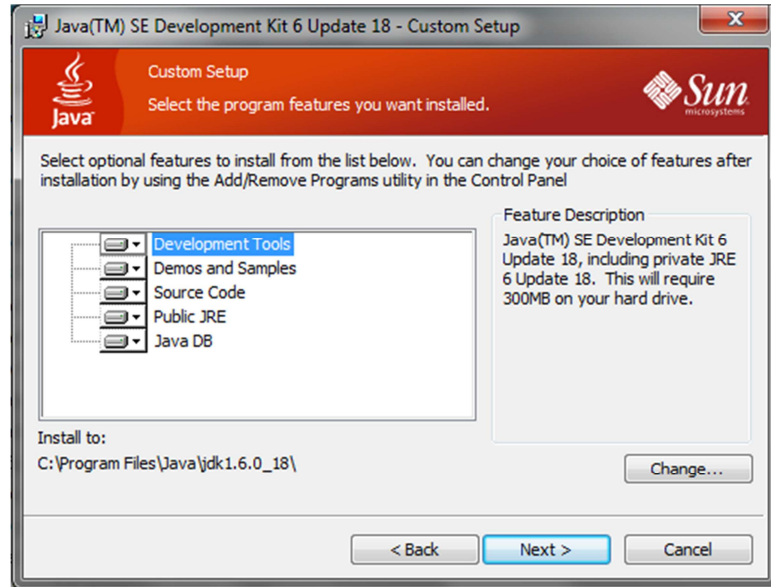


Fig.3.2.2. Selecció de ruta i utilitats del JDK.

- Pas 3: Al acabar la instal·lació del Java JDK es demanarà la ruta d'instal·lació del Java Runtime Environment (JRE) . Aquest és molt important on instal·lar-lo, ja que els programes que s'executin en Java el requeriran. Es pot deixar la ruta per defecte (Fig. 3.2.3).

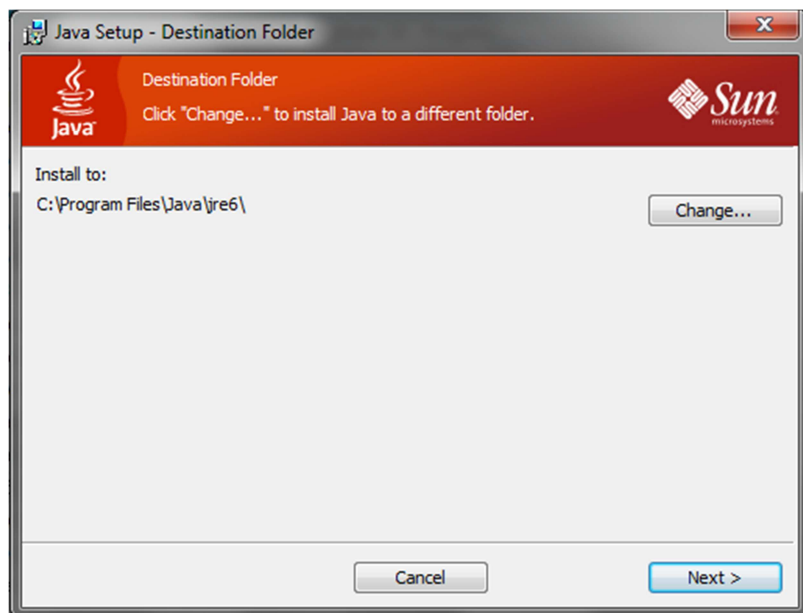


Fig.3.2.3. Selecció de ruta del JRE

Com que s'utilitzarà un software extern per compilar el codi, no serà necessari configurar les variables d'entorn de Java JRE. Si no, un cop acabada la instal·lació s'hauria d'entrar a la configuració per canviar alguns paràmetres de rutes.

Si s'han seguit bé tots els passos de la instal·lació, la instal·lació del JDK haurà conclòs, i ja es disposarà de totes les eines *Java* (llibries de *Java* 1.6 i màquina virtual *Java*) necessàries per treballar després en *Android*.

### 3.3. Instal·lació de l'SDK de Android.

*SDK* significa *Source Development Kit*, és a dir, és el conjunt d'eines que es fan servir per desenvolupar aplicacions en *Android*. De l'*SDK* d'*Android* es podran descarregar les biblioteques de codi que es faran servir alhora de programar, com per exemple la API de *Google*. També inclou la capacitat d'executar emuladors d'*Android* de diferents característiques, mitjançant un *plug-in ADB* (*Android Debug Bridge*).

El paquet es pot baixar de la pàgina oficial de *Google*:

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

Inclou varies versions per diferents sistemes operatius. En el cas de *Windows*, es baixarà l'arxiu: [android-sdk\\_r11-windows.zip](#)

L'únic que s'ha de fer en aquest moment és descomprimir-lo i deixar-lo en una carpeta coneguda del sistema (Fig. 3.3.1), per després poder accedir-hi des de l'*Eclipse*. També es pot descarregar la versió executable, que mitjançant un assistent, s'instal·larà en disc.

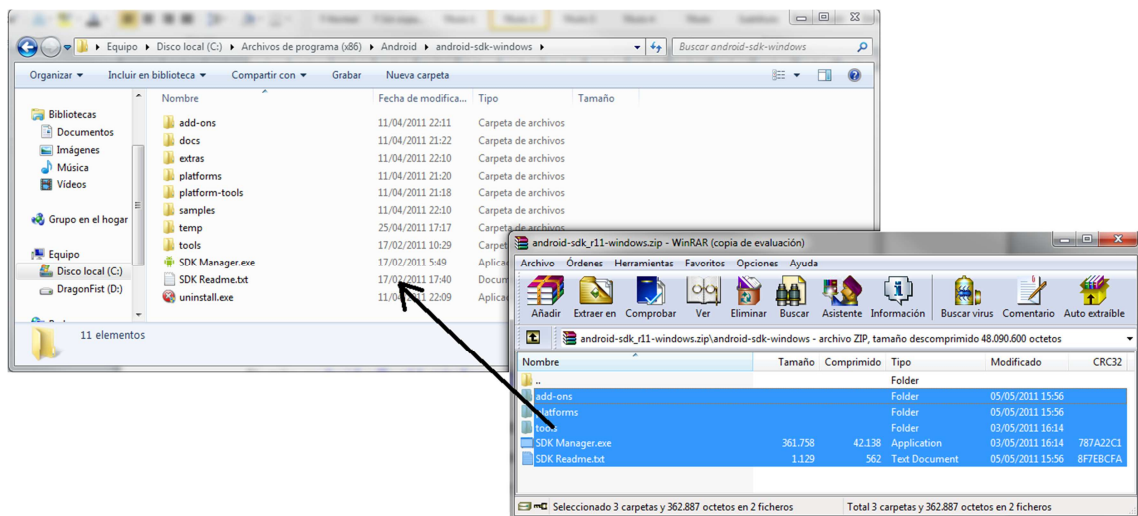


Fig.3.3.1. Descomprimir els fitxers de l'arxiu .zip a una carpeta coneguda

A continuació, es poden instal·lar els complements desitjats de *Android* mitjançant el *SDK Manager.exe* (executable contingut dins la carpeta en qüestió). Es tracta de seleccionar els paquets desitjats i pressionar el botó d'instal·lar (Fig. 3.3.2).

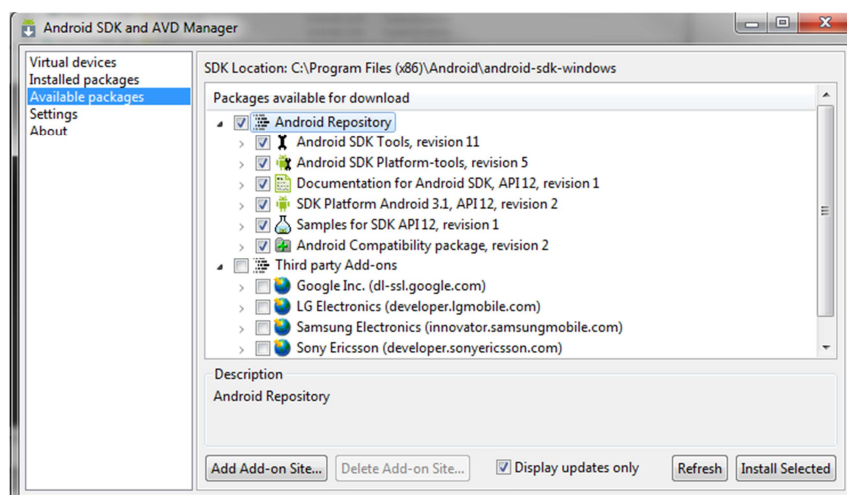


Fig 3.3.2 Instal·lació de components a l'SDK

Per últim només queda configurar un emulador. S'entra a la pestanya *Virtual Devices*, i es fa *click* a *New*. S'obrirà l'assistent de l'esquerra (Fig.3.3.3.) i es podrà configurar l'emulador. Si es vol treballar amb *Google APIs* s'haurà de seleccionar en el camp *target*, *Google Apis(Google Inc.) - API Level 11*.

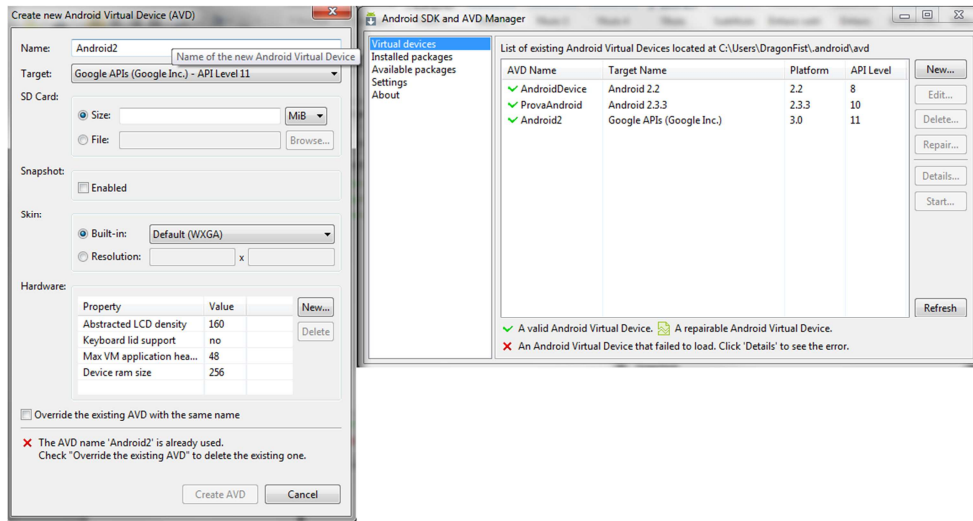


Fig 3.3.3 Configuració de l'emulador

Si s'ha fet correctament, es podrà prémer el botó *Start* i l'emulador s'executarà (Fig. 3.3.4). Es veurà l'entorn gràfic d'Android tal i com apareixeria en un dispositiu mòbil, o *tablet*. La interfície gràfica dependrà de la versió d'Android que s'hagi escollit al crear l'emulador.

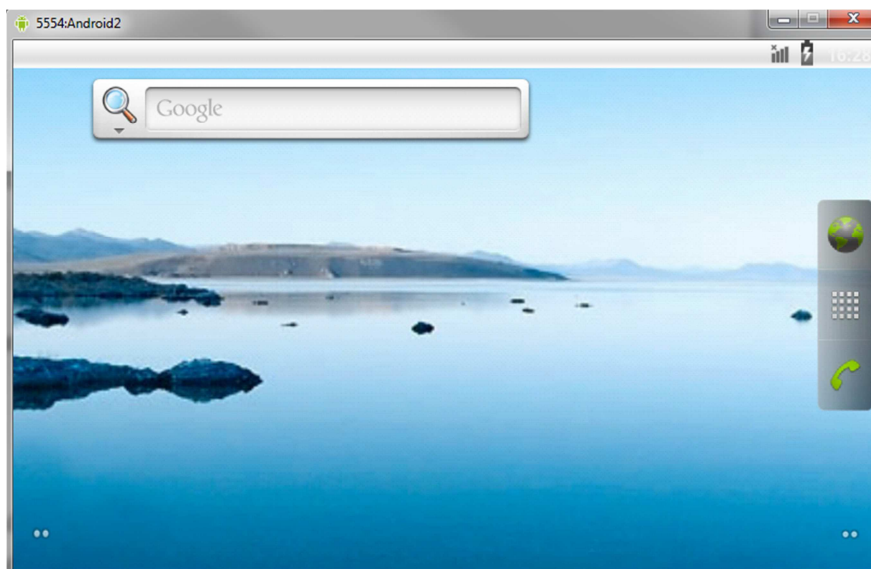


Fig 3.3.4. Emulador de Android

Possiblement s'hagi de agregar a la variable *path* de *Windows* el directori */tools* de *l'SDK*. Per fer-ho, s'obre amb el botó dret la icona de *Equip* i se selecciona el camp *Propietats*. Al camp *Opcions avançades* es selecciona *Variables d'entorn*, i un cop a dins, la variable *Path*. Es fa *click* sobre *modificar* i s'introdueix la ruta del directori */tools*.

Si s'han seguit correctament els passos es tindrà ja instal·lat el *SDK d'Android*, que permetrà l'ús d'emuladors de terminals mòbils i *tablets* i proporcionarà accés a les diferents llibreries de programació que facilita *Google* per el desenvolupament d'aplicacions *Android*.

### 3.4. Instal·lació de l'Eclipse.

*Eclipse* és un entorn de desenvolupament (*IDE*) per programar en *Java*, i a més el recomanat i oficial per *Android* ( des de *Google* ). Existeixen altres alternatives com *NetBeans* (oficial de *Java*) que també serien solucions vàlides, però es focalitzarà el tema en la instal·lació d'*Eclipse*.

L'entorn de desenvolupament *Eclipse* és un software *open source* i gratuït que proporciona *Eclipse Foundation* per a desenvolupadors de *Java*, i altres llenguatges de programació. Per programar en *Android* s'ha d'aconseguir la versió *Eclipse IDE for Java Developers*, de l'arquitectura del sistema operatiu sobre el que treballem (*x64* o *x86*). Es pot obtenir *l'IDE* desde la pàgina oficial de Eclipse:

- <http://www.eclipse.org/downloads/>

En el cas de *Windows 7 (x64)*, s'haurà de descarregar la versió de 64 bits de *Eclipse IDE for Java Developers*.

Eclipse es distribueix sense instal·lador, és a dir, és una carpeta comprimida amb el programa llest per funcionar a dins (software portable). El que s'haurà de fer és configurar-lo adequadament per treballar en *Android*.

Es descomprimeix el contingut de l'arxiu en una carpeta del sistema i s'executa *Eclipse.exe*. El primer cop que s'obri demanarà la ruta del directori de treball.

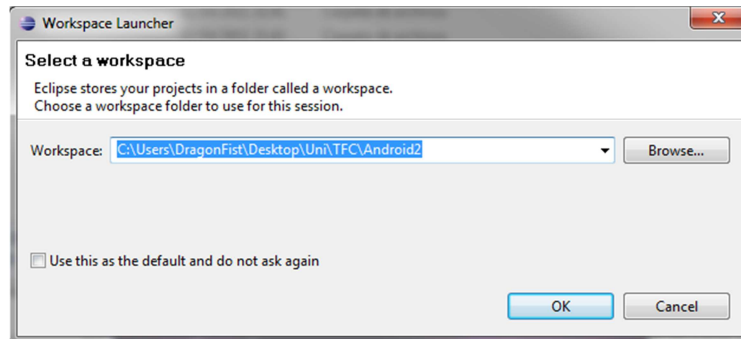


Fig 3.4.1 Configuració del directori de treball

Un cop dins d'*Eclipse* es procedeix a instal·lar *ADT (Android Development Tools)*. Es selecciona el camp *Software Updates* dins el menú *Help*, i apareixerà una finestra de complements. S'entra a *Available software* i es fa *click* a *Add site*. S'ha adjuntar la direcció "<http://dl-ssl.google.com/android/eclipse/>", cosa que permetrà descarregar els *plug-in* necessaris per programar en *Android* (Fig. 3.4.2).

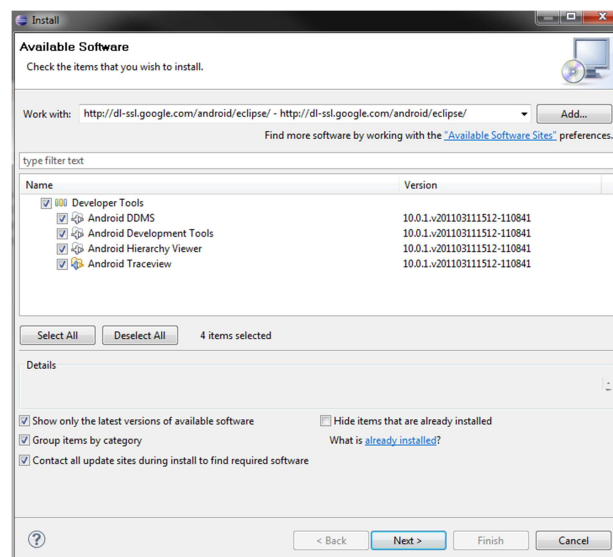


Fig 3.4.2 Instal·lació del ADT i complements.

Un cop instal·lat l'ADT i complements, s'ha de reiniciar *Eclipse*. A continuació s'obre el menú *Window* i se selecciona el camp *Preferences*. En l'apartat d'*Android* s'haurà d'introduir la ruta on s'ha descomprimit l'*SDK* d'*Android* i ja estarà llest.

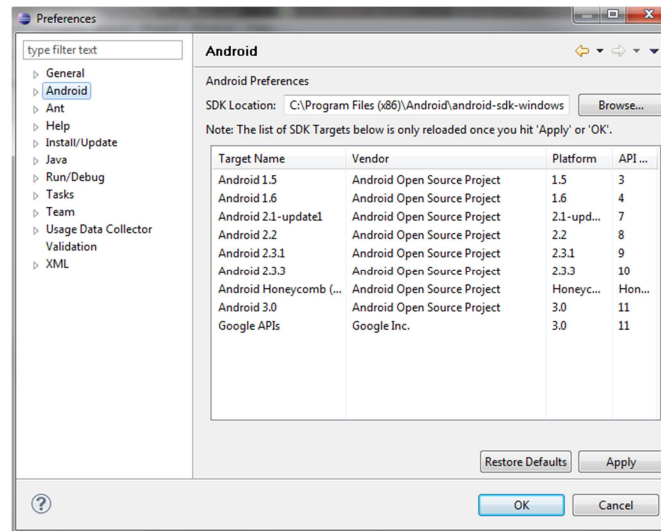


Fig 3.4.3. Ruta de l'SDK de Android a Eclipse.

### 3.5. Aplicació de prova *Hello, World*.

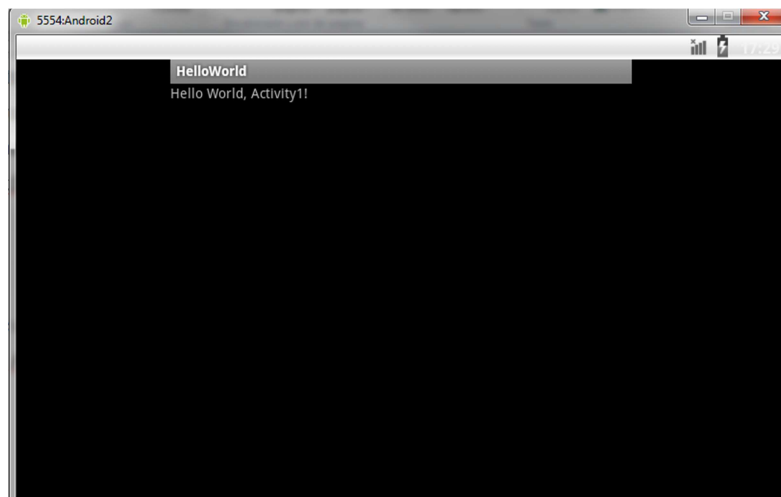
Per testejar el correcte funcionament de tot l'entorn, s'explicarà com crear el projecte Android més simple: *Hello, World*.

Amb l'*Eclipse* obert, creem un nou projecte, del tipus Android, i apareixerà un assistent de configuració d'aquest amb els següents camps que omplir:

- Nom del projecte, contingut, versió d'*Android* del projecte, nom de la aplicació, nom del paquet, nom de l'*Activity* (pantalla gràfica).

S'omplen els camps amb cura ( el paquet ha de tenir un nom únic, així que millor donar-li un nom ben llarg del tipus “com.developers.Diaz.Android.HelloWorld” ), i es fa *click* al botó *Next*.

Un cop s'ha creat el projecte, es pot provar d'executar directament amb el botó *Run* (de color verd amb un triangle apuntant a la dreta). Si el projecte ha estat creat correctament, s'obrirà una finestra on s'haurà de seleccionar el emulador que prèviament s'ha creat amb l'*SDK* de *Android*. Se selecciona, i es veu el resultat: l'emulador arranca i *Eclipse* instal·la l'aplicació recentment creada a l'emulador de forma automàtica (Fig. 3.5.1).



**Fig 3.5.1** Aplicació Hello, World



## 4. Estructura d'un projecte Android.

### 4.1. Arbre d'un projecte Android a Eclipse.

Al començar un projecte nou *Android* amb *Eclipse*, es crea una estructura de directoris i fitxers que formen l'aplicació bàsica. L'arrel d'aquesta estructura és una carpeta amb el nom del projecte, i hi conté (Fig. 4.1.1):

- **/src**: és el directori on es guarden els arxius del codi font de l'aplicació (arxius.java).
- **/gen**: carpeta que guarda arxius que *Eclipse* genera de forma automàtica, com l'arxiu *R.java*.
- **/Android**: llibreries de *Android*. Es pot desplegar i veure el contingut del *SDK* (paquets i classes)
- **/assets**: directori per guardar recursos que utilitzi la aplicació. Per utilitzar els recursos de dins d'aquesta carpeta, cal utilitzar la classe *AssetManager* (per exemple, per llegir un *stream* de *bytes*).
- **/res**: directori principal de recursos. Hi conté les imatges que utilitza la aplicació, així com els arxius *\*.xml* que defineixen la interfície gràfica, i variables de text. S'hi accedeix des de la classe *R.java*.
- **AndroidManifest.xml**: Arxiu de configuració de l'aplicació. Si es necessita utilitzar serveis o recursos físics del dispositiu mòbil, s'ha de declarar aquí. També s'hauran de declarar totes les *Activities* (és la classe que defineix una interfície gràfica).
- **Default.properties**: fitxer de propietats auto-generat per *Eclipse*. No l'hem de modificar.

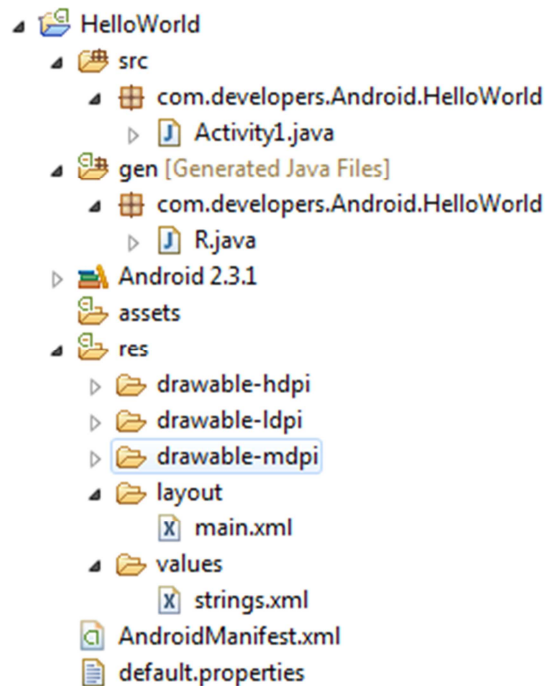


Fig 4.1.1 Arbre d'un projecte Android a Eclipse

## 4.2. La classe Activity.

En la programació orientada a objectes, com és *Java*, els dos elements més presents són les classes i els objectes. Una classe es podria definir com un conjunt d'elements que tenen el mateix comportament i característiques, i la implementació en sí d'un tipus d'objecte. Les classes es componen d'atributs, mètodes, i constructors.

La classe *Activity* és la base de la programació en *Android*. Aquesta conté els mètodes per poder crear una interfície gràfica de forma fàcil. Segons *Google*, “una *Activity* és una cosa amb un objectiu que l'usuari pot fer”. S'entén potser millor si s'imagina que *Activity* és una finestra (gràfica). Per defecte, quan es crea un projecte *Android*, es crea un arxiu a la carpeta */src* amb extensió *\*.java*. Aquest arxiu és ja la classe principal del projecte, i hereta els mètodes d'*Activity* (com a superclasse). Si es mira l'arxiu *AndroidManifest.xml*, es veurà que hi ha creada una instància d'*Activity* nombrant aquesta primera com a principal de l'aplicació.

En un projecte poden haver diferents *Activities*, o sigui, diferents pantalles. Entre elles, es gestionen en forma de pila, així que des de una d'elles es pot cridar a una altra, que si finalitza, es recuperarà l'anterior. Totes elles han d'estar declarades a *AndroidManifest.xml*.

Les *Activities*, o dit d'una altra manera, les classes que hereten d'*Activity* tenen un cicle de vida molt clar, que queda definit per l'esquema següent (Fig 4.2.1. ).

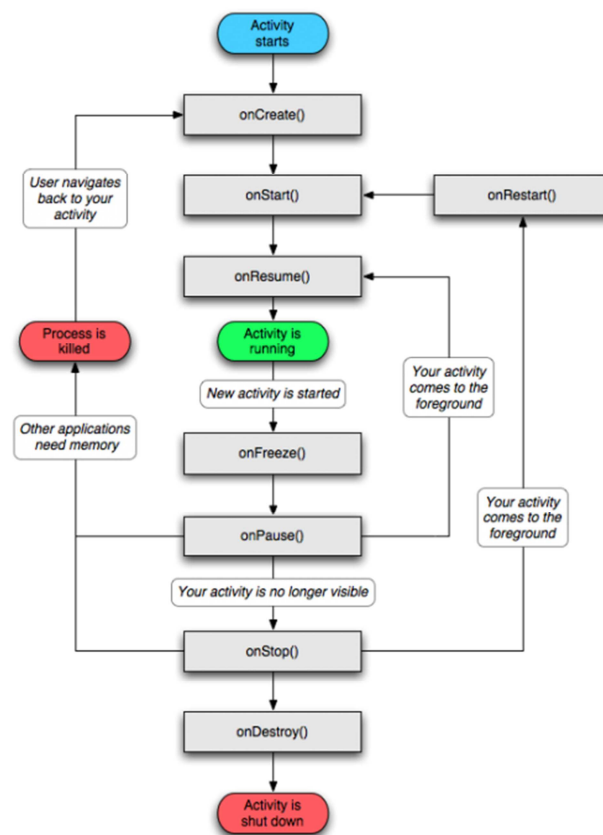


Fig 4.2.1. Cicle de vida d'una Activity

Per crear una instància d'*Activity*, s'ha de cridar primer de tot el mètode *OnCreate()*. Només fa falta cridar-lo una vegada, o quan el sistema l'elimini de memòria per falta de memòria per atendre altres processos. Després es pot recórrer un cercle de vida mitjançant els mètodes de tipus "*On\_\_\_*" (esdeveniments) fins arribar a parar la classe, i a destruir-la de memòria si es dona el cas.

En el cas d'una aplicació de prova:

```
package com.developers.Android.HelloWorld;

import android.app.Activity;
import android.os.Bundle;

public class Activity1 extends Activity {
    /** Called when the activity is first created.
    */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

- **package** defineix el paquet on es troba la classe. Un paquet pot contenir varies classes, i resulta còmode agrupar-les per paquets ja que així podem atribuir visibilitats entre elles.
- **import** fa que els mètodes i objectes de les llibreries seleccionades siguin utilitzables a la classe.
- Es declara la classe com a pública, que hereta d'*Activity*.
- Es reescriu el mètode *onCreate* per fer el que l'usuari desitgi a la classe.
- **super** s'encarrega d'invocar el mètode sobreescrit, en aquest cas, *onCreate()*.
- **setContentView** utilitza el recurs XML de la carpeta */res/layout* per construir l'aspecte gràfic que tindrà l'activitat.

Dins del mètode *onCreate()* es pot incloure codi per fer el que es desitgi com crear variables i objectes, modificar-los, destruir-los,... Tot el codi que s'ha d'executar quan es mostra aquesta pantalla ( una *Activity* és una classe que proporciona una vista).

### 4.3. La classe Intent.

Els objectes de la classe Intent s'encarreguen d'invocar les *Activities*. Segons *Google*, “un *Intent* és la descripció abstracta d'una operació que es durà a terme”. Aplicant la definició a *Activity*, un *Intent* és la classe que permet especificar quina *Activity* s'executarà. Per fer-ho, s'accedeix a un dels mètodes de la classe *Activity* (“*startActivity(Intent i)*” o una de les seves variants) i se li passa aquest *Intent* com a paràmetre, que conté la informació de quina *Activity* s'executarà.

A part de fer la feina de manegador *d'Activities*, intent permet transportar dades i categories. És una manera molt útil de passar informació entre canvis de finestra (cada classe és una finestra), o dit d'altra manera, entre classes.

Quan s'ha cridat la nova *Activity*, l'anterior queda a sobre de la pila, cosa que fa que si finalitza l'actual, tornarem a l'anterior.

Exemple pràctic:

```
Button b = (Button) this.findViewById(R.id.MapAddress);
b.setOnClickListener(new OnClickListener() {
    public void onClick(View arg0)
    {
        Intent i = new Intent (UploadConcertInfo.this,
                               TerceraActivity.class);
        if(hashRecord!=null)    i.putExtra("Event",hashRecord);

        startActivityForResult(i,TERCERAACTIVITYRCODE);
    }
});
```

El codi reflexa la implementació d'un botó que està creat a l'arxiu XML que defineix la vista de l'aplicació, i que s'està escoltant per si un usuari el prem. En aquest cas, l'activitat on es troba s'aturarà i s'executarà la nova, manegada per Intent. Quan es crea l'objecte, el constructor demana que se li passi el context en què es troba, i la *Activity* a executar. Depèn d'una condició que se li passi certa informació mitjançant el mètode “putExtra(String id,HashRecord objecte)” (en aquest cas, el que se li passa és un objecte de la classe HashRecord), i per finalitzar, es crida el mètode

“startActivityForResult(Intent i, int rcode)”, on se li passa l’Intent que manegarà el canvi d’Activities, i un codi de retorn que podrà ser capturat quan finalitzi la Activity filla.

## 4.4. Layouts

De l’anglès, la paraula *layout* significa distribució, traçat, o disseny. És així com s’anomenen les distribucions dels elements gràfics residents en una pantalla *d’Android*. Encara que el *layout* d’una aplicació es pot construir dinàmicament amb codi Java (aplicant codi directament al mètode *onCreate()* de la classe *Activity*), la manera més comú de construir-lo és utilitzar XML. Aquest arxiu XML és interpretat per la classe “R.java” (dins la carpeta “/gen” del projecte) i és capaç de crear elements gràfics (per exemple, botons i etiquetes) i distribuir-los de diferents maneres. Aquests arxius XML resideixen dins la carpeta */res/layout* per defecte, tot i que si es vol crear suport per diferents resolucions, s’ha de crear diferents carpetes penjant de “/res” amb identificadors únics (per exemple: */layout-land*, *layout-port-xhdpi-xlarge*), que *Android* identificarà i escollirà automàticament en relació al nostre terminal físic. Cada una d’aquestes carpetes contindrà un o varis *layouts* ajustat als identificadors corresponents (calen múltiples distribucions per a múltiple suport de resolucions).

Existeixen cinc maneres de distribuir els elements gràfics dins l’entorn XML: *Linear Layout*, *Table Layout*, *Absolute Layout*, *Relative Layout* i *Frame Layout*.

### 4.4.1. Linear Layout.

La distribució lineal consisteix en situar elements un darrere l’altre a la pantalla per ordre de creació. Podem decidir si aquests s’alinearan vertical o horitzontalment. Per crear una distribució lineal s’ha de introduir una capçalera XML que així ho indiqui:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
    />
</LinearLayout>
```

#### 4.4.2. Table Layout.

La distribució per taules utilitza una manera d'organitzar l'espai de la pantalla per files (*TableRow*). Dins de cada fila es poden crear els elements que es vulguin.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout android:id="@+id/TableLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TableRow android:id="@+id/fila1">
        <TextView android:id="@+id/TextView01"
            android:text="Data"
            android:width="100dp" />
        <EditText android:id="@+id/EditText01"
            android:width="200dp" />
    </TableRow>
    <TableRow android:id="@+id/fila2">
        <TextView android:id="@+id/TextView02"
            android:text="Hora"/>
        <EditText android:id="@+id/EditText02" />
    </TableRow>
</TableLayout>
```

#### 4.4.3. Relative Layout.

Sembla ser el *layout* que permet més tipus de posicionaments, tot i que és difícil estructurar-lo. Es tracta de col·locar els elements en relació a altres ja posicionats, mitjançant les eines que ens brinda aquest tipus de distribució (*below, toRightOf...*)

```

<RelativeLayout android:id="@+id/RelativeLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Data:"
        android:width="100px" />
    <EditText android:id="@+id/EditText01"
        android:layout_width="220px"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/TextView01"
        android:layout_below="@+id/RelativeLayout01" />
    <EditText android:id="@+id/EditText02"
        android:layout_width="220px"
        android:layout_height="wrap_content"
        android:layout_below="@+id/EditText01"
        android:layout_alignLeft="@+id/EditText01" />
    <TextView android:id="@+id/TextView02"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Hora"
        android:width="100px"
        android:layout_below="@+id/EditText01"
        android:layout_toLeftOf="@+id/EditText02" />
</RelativeLayout>

```

#### 4.4.4. Absolute Layout.

És el *layout* més fàcil d'utilitzar per posicionaments complexos. Però té un inconvenient molt gran, no és escalable a múltiples resolucions, ja que el posicionament és per mides absolutes (*pixels*). Actualment, *Google* el declara obsolet, tot i que funciona i es pot utilitzar.

```

<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout android:id="@+id/AbsoluteLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <EditText android:id="@+id/EditText01"
        android:layout_width="200px"
        android:layout_height="wrap_content"
        android:layout_x="12px"
        android:layout_y="12px" />
    <Button android:text="Accepta!"
        android:id="@+id/Button01"
        android:layout_width="100px"
        android:layout_height="wrap_content"
        android:layout_x="220px" android:layout_y="12px" />
</AbsoluteLayout>

```



#### 4.4.5. Frame Layout.

La distribució per *frames* consisteix en enmarcar els elements desitjats per bloquejar-los a una zona de la pantalla determinada. Amb la propietat “gravity” es pot controlar la posició dels elements dins del marc.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout android:id="@+id/FrameLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <ImageView android:id="@+id/ImageView01"
        android:src="@drawable/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="center" />

    <TextView android:text="Android"
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dip"
        android:layout_gravity="center_horizontal|bottom"
        android:padding="10dip"
        android:textColor="#AA0000" android:textStyle="bold"
        android:textSize="20px"
        android:background="#00000000" />

</FrameLayout>
```

#### 4.5. L'arxiu AndroidManifest.xml.

Totes les aplicacions d'Android duen aquest arxiu a l'arrel del projecte. Conté informació essencial sobre la aplicació en qüestió, que necessita ser atesa abans de l'execució. AndroidManifest.xml s'encarrega de:

- nombrar el paquet de l'aplicació, que servirà com a identificador d'aquesta.
- Descriure els components (*Activities*, serveis, receptors *broadcast*, i proveïdors de contingut) que l'aplicació utilitzarà i habilita el seu ús.
- Determinar quina *Activity* és la principal de l'aplicació.
- Determinar quins processos manegaran els components de l'aplicació.
- Declarar els permisos que se li donaran a l'aplicació (*Internet*, *GPS*, *Bluetooth*...)

- Llistar les llibreries externes que s'utilitzaran al projecte, i les habilita.
- Declarar la versió mínima en què l'aplicació s'ha d'executar

Exemple de l'arxiu AndroidManifest.xml:

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

  <application android:icon="@drawable/icon"
    android:label="@string/app_name">
    <activity android:name=".Activity1"
      android:label="@string/app_name">
      <intent-filter>
        <action
          android:name="android.intent.action.MAIN" />
        <category
          android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <uses-library
      android:name="com.google.android.maps">
    </uses-library>
    <uses-permission
      android:name="android.permission.INTERNET">
    </uses-permission>
  </application>
</manifest>
```

## 5. Estudi i consideracions prèvies al disseny d'una aplicació per Android.

### 5.1. Aprenentatge.

Si es vol començar a desenvolupar una aplicació per Android cal realitzar un estudi previ important per familiaritzar-se amb tot l'entorn de treball i conèixer les eines de les que es pot disposar.

Una forma d'adaptar-se a l'entorn de forma ràpida i efectiva és donar molta importància a la part pràctica. Resulta de molta ajuda començar a desenvolupar aplicacions senzilles que ajudin a conèixer l'estructura d'*Android* i el funcionament dels seus components (sempre i quan es tingui algun coneixement previ en programació).

La solució adoptada doncs per obtenir l'aprenentatge necessari per realitzar el projecte ha tingut un enfocament totalment pràctic. A través de realitzar alguns programes senzills com un convertidor de euros/pessetes s'ha aconseguit començar a entendre com funciona *Java*, i *Android*. Els aspectes més importants a estudiar de cara a adaptar-se a l'entorn, i en ordre d'aprenentatge són: escoltes dels elements creats al *layout*, moviment d'*Activities*, declaració de permisos i components a *AndroidManifest.xml*, i estructuració del *layout*.

Els primers passos consisteixen en modificar petites parts del programa com etiquetes, caixes de text, botons, i escoltes per tocar parts de codi molt concretes i fàcilment comprensibles.

Al quadre de text inferior es pot observar la implementació de dos botons i les seves escoltes que actuen sobre caixes de text editables (en aquest cas, convertint valors monetaris). També s'observa que mitjançant el mètode *findViewById(id)* es fa referència a un element creat a través de l'interfície *XML*. Aquest codi està inclòs dins el mètode *OnCreate()* de la classe *Activity*.

```

campoEuros=(EditText)findViewById(R.id.euros);
campoPesetas = (EditText) findViewById(R.id.pesetas);
Button eurosAPesetas = (Button) findViewById(R.id.euros2pesetas);
Button pesetasAEuros = (Button) findViewById(R.id.pesetas2euros);
Button salir= (Button) findViewById(R.id.salir);

eurosAPesetas.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view){
        campoPesetas.setText(""+Double.
            parseDouble(campoEuros.getText().toString()*166.386);
    }
});
pesetasAEuros.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        campoEuros.setText(""+Double.
            parseDouble(campoPesetas.getText().toString()/166.386);
    }
});

```

Després d'haver comprès el funcionament d'elements dins *Android* l'exercici següent consisteix en aprendre a manejar *Activities*. Les *Activities* són les classes Java que portaran tot el codi específic per realitzar una acció determinada (per exemple, presentar un menú i implementar les accions dels seus botons, o presentar informació recollida d'una base de dades). Ja que el moviment entre *Activities* serà constant en una aplicació, és bo entendre bé quin és el seu cicle de vida (veure Fig 4.2.1 ). Quan creem una *Activity*, haurà de ser declarada com a part de la nostra aplicació, o sigui, com un recurs utilitzable. Això porta a modificar l'arxiu *AndroidManifest.xml*. Cada *Activity* que s'inclogui a l'aplicació haurà de constar en aquest fitxer.

Exemple de crida d'una *Activity* de de un botó:

```

ImageButton e = (ImageButton)
this.findViewById(R.id.uploadConcert);
e.setOnClickListener(new OnClickListener(){
    public void onClick(View arg0)
    {
        Intent i = new Intent (Prova2View.this,
            UploadConcertInfo.class);
        startActivity(i);
    }
});

```

Si es vol incloure components addicionals o recursos variis, com per exemple disposar d'accés a Internet, GPS, o llibreries concretes, s'haurà d'accedir també a l'arxiu *AndroidManifest.xml*. En ell s'haurà de donar els permisos necessaris a l'aplicació per disposar dels recursos desitjats, mitjançant notacions *XML* concretes. Així doncs, si es necessita accedir a la posició *GPS* per mostrar-la en un formulari, s'hauria d'acreditar a

l'aplicació que té permís per utilitzar aquest recurs ( veure apartat 4.5. L'arxiu *AndroidManifest.xml*).

Des de el principi de l'aprenentatge, s'han tocat també els arxius *.xml* que conformen la visió gràfica de l'aplicació, però fins aprendre com funcionen els altres aspectes prèviament descrits resulta quasi inútil prestar excessiva atenció als *layouts*. El posicionament dels elements ha d'ésser dissenyat amb cura i hem de prestar especial atenció en quin mètode de distribució s'utilitzarà per cada finestra de l'aplicació. Calen hores de pràctica per dominar els posicionaments, i la manera d'aprendre'n és atrevir-se a desmuntar l'arxiu d'una determinada interfície i intentar reformar-lo. Existeixen varis tipus de posicionament, així que no està de més parar atenció a cada un d'ells per saber en què ens podem beneficiar al triar-ne un o altre (veure apartat 4.4.).

Un cop s'ha entès el funcionament d'aquests elements, es pot dir que es pot començar a plantejar el desenvolupament d'una aplicació. Tot i que no són tots els coneixements que es necessitaran, sí que en són els més bàsics i els que s'utilitzaran en major freqüència.

## 5.2. Eines disponibles.

Prèviament a decidir quin projecte es realitzarà, és vital saber què es podrà fer a l'entorn del que disposem.

La nova generació de dispositius mòbils disposa de moltes eines noves en aquest tipus de plataforma, com és l'accés a Internet via *3G* o *Wifi*, la *geolocalització* a través de *GPS* integrats, acceleròmetre, càmera, micròfon o *Bluetooth*, que a l'estar a l'abast d'un dispositiu petit i portable, obren noves idees i solucions a possibles demandes i necessitats.

Al tenir accés a Internet, cal encabir com a eina important l'ús i la gestió de la informació. Es poden implementar bases de dades o recollir informació de diverses fonts per donar funcionalitat a l'aplicació.

Les eines en les que s'ha centrat aquest projecte, gràcies a disposar d'un terminal el qual hi tenia accés, han estat *l'Internet* (ja sigui connectant per *3G* o *Wifi*) i el posicionament GPS.

### 5.3. Necessitats.

Contemplar quines són les necessitats que pot cobrir l'ús d'un terminal mòbil és essencial abans de començar el projecte. És necessari que un projecte sigui útil per algun sector, així que cal estudiar les possibles necessitats que els usuaris de dispositius mòbils poden tenir.

Prenent per base que un terminal mòbil és entre altres coses una eina que permet la comunicació, i que gràcies a *Internet*, també la informació, es poden ajuntar aquests dos termes en la expressió "comunicar informació". Així doncs, com a primer punt de vista el plantejament del projecte es basa en subministrar una eina que permeti comunicar i rebre algun tipus de informació. Al accedir-hi des de una plataforma mòbil, a la informació se li suma un valor afegit que és la comoditat de la seva obtenció, degut al trencament de la dependència física a un lloc determinat.

Un cop analitzades les necessitats que pot cobrir un terminal mòbil de nova generació, es pot plantejar una idea per tirar endavant un projecte.

### 5.4. Conclusió de l'estudi

A partir de l'estudi realitzat i les consideracions necessàries abans d'escollir l'aplicació a desenvolupar, es decideix crear *GeoShows*.

*GeoShows* pretén ser una aplicació client capaç de visualitzar o informar concerts o activitats musicals. Gràcies a les eines d'*Internet*, es pot obtenir accés d'escriptura i lectura a una base de dades externa on guardar registres de concerts. *GeoShows* en gran part deu el seu nom a la manera de situar els concerts. Fent un ús d'*Internet* i *GPS*

ahora, es mostra sobre un mapa subministrat per *Google Maps API* on estarà el concert que es busca, o per altra banda, la possibilitat d'informar de la posició del concert amb un sol *click* a sobre el mapa. Al ser una aplicació per terminals mòbils, ha de resultar còmode d'ús, i per això s'ha comptat amb el recurs d'utilitzar el mapa per introducció de dades o adquirir-les.

Respecte a les necessitats que cobreix la aplicació, *GeoShows* presenta una proposta alternativa a recórrer la web per buscar activitats musicals des de qualsevol lloc, sempre que es disposi d'un terminal *Android*, amb accés a Internet i *GPS*. Gràcies a la seva localització d'esdeveniments a través d'un mapa, és fàcil assimilar les distàncies de les poblacions on esdevenen aquestes activitats. L'aplicació farà ús d'una base de dades penjada a Internet, la qual els usuaris tindran accés de lectura i escriptura a través de la interfície de l'aplicació. D'aquesta manera la informació continguda a la base de dades pot créixer exponencialment a mesura que hi hagin més usuaris de l'aplicació. Per altra banda, no necessita d'un manteniment constant per banda d'un administrador, ja que qui fa viva l'aplicació són els propis usuaris.





## 6. GeoShows, el buscador de concerts.

### 6.1. Què és GeoShows?

*GeoShows* és una aplicació per *Android* que permet visualitzar i informar d'activitats musicals a arreu del món. Presenta diversa informació d'aquests actes com és la data, els grups que toquen, la localitat i la direcció, el preu, comentaris addicionals, pàgina web del/s grup/s o festival, fotografia del grup o cartell, i informant del concert. És capaç de situar els esdeveniments musicals en un mapa de *Google Maps* per a una fàcil interpretació geogràfica de la seva localització. L'aplicació és mantinguda per els mateixos usuaris, gràcies a la seva capacitat d'interacció envers a l'escriptura a la base de dades que gestiona tots els esdeveniments, cosa que fa que el seu valor depengui del nombre d'usuaris actius. Les entrades dels concerts són llegides ordenades per data (més propera) o localitat (per ordre alfabètic), orientant així la recerca de concerts per proximitat tant horària com física.

*GeoShows* planteja una interfície simple i còmode per oferir un servei d'informació guiat a la recerca d'activitats d'oci musicals, i ho fa proporcionant independència física al ser per a dispositius mòbils (Fig 6.1.1.).



Fig 6.1.1 Menú principal de GeoShows

## 6.2. Funcionament de l'aplicació.

*GeoShows* és una aplicació per a *Android* que bàsicament utilitza serveis d'*Internet* i posicionament *GPS*. *Internet* és utilitzat per accedir a la base de dades que conté la informació de tots els concerts, i als mapes subministrats per *Google* (*Google Maps*). El *GPS* ens revela la nostra posició actual a sobre del mapa per visualitzar distàncies amb els esdeveniments. La base de dades emprada és un document penjat a un compte de *Google Docs*, creat amb l'aplicació *Google SpreadSheets*. Mitjançant les llibreires *Google Maps API* i *Google SpreadSheets API* tenim accés fàcil tant als mapes com a la base de de dades. Es veuen les relacions a la figura 6.2.1

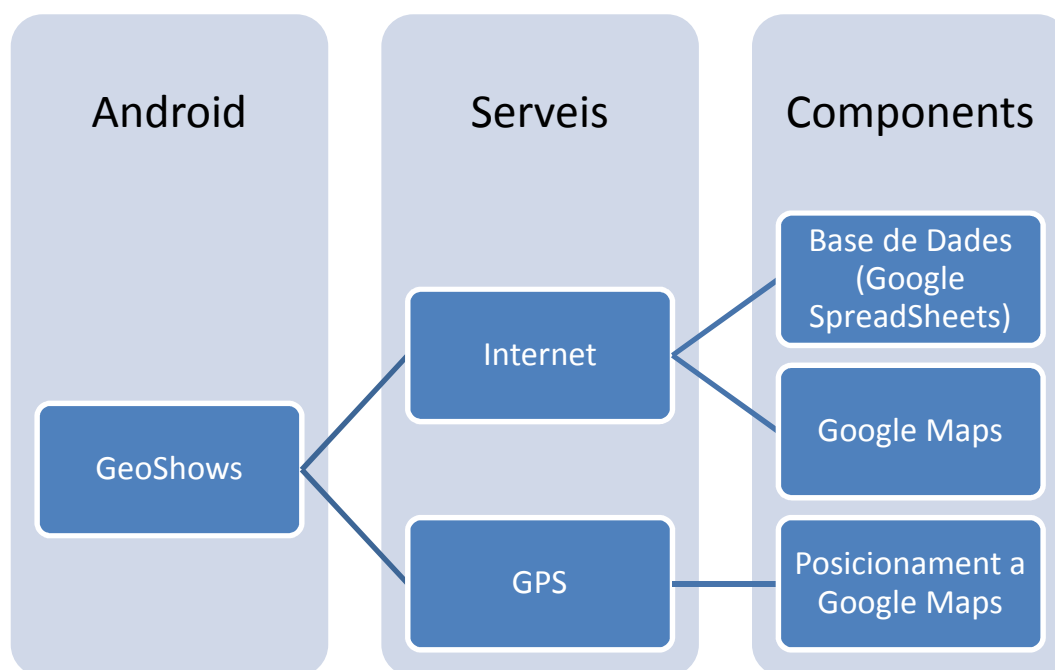


Fig 6.2.1. Funcionament de l'aplicació

*GeoShows* és una aplicació bidireccional, és a dir, per una banda és capaç de llegir registres de concerts i presentar la informació de forma ordenada, així com la posició del concert en un mapa, i per altra banda és capaç de recollir la informació sobre un

concert (entrada per un usuari), utilitzar un mapa per situar-lo, i pujar tota la informació a base de dades.

### 6.2.1. Base de dades

La base de dades que manega *GeoShows* és allotjada a *Google Docs*, utilitzant l'aplicació web *Google SpreadSheets* (veure apartat 2.2.2). La base de dades consisteix en un full de càlcul amb 3 pàgines: una pàgina per recollir la informació pujada, una per ordenar les entrades per data, i una altra per ordenar primer per localitat, i després per data.

A la primera pàgina es carreguen els esdeveniments que els usuaris creen un darrere l'altre, sense importar l'ordre en el què es guarden. Cada concert que s'informa crea un registre nou a aquesta pàgina del full de càlcul, amb els camps: *DATA*, *GRUPS*, *WEB*, *LOCALITAT*, *DIRECCIÓ*, *LONGITUD*, *LATITUD*, *PREU*, *COMENTARIS*, *INFORMANT*, i *FOTOGRAFIA*.

Mitjançant la funció *SORT* que permeten els fulls de càlcul, des de una altra de les pàgines s'accedeix a la primera i s'ordena per dates les entrades que hi conté (Fig 6.2.1.1). El mateix procediment és utilitzat per ordenar per localitats i dates, en una nova pàgina del full de càlcul. *SORT* omple la taula amb les entrades ordenades segons l'algorisme que s'hagi implementat de forma automàtica, així que des de l'aplicació el que s'haurà de fer en els procediments de lectura és accedir a les pàgines del full de càlcul ordenades.

1	A	B	C	D	E	F	G	H	I	J	K
	DATA	GRUPS	WEB GRUPS	LOCALITAT	DIRECCIÓ	LONGITUD	LATITUD	PREU	COMENTARIS	INFORMANT	FOTOGRAFIA
14											<a href="http://3.bp.blogspot.com/_RmE17b85Cw/Tt5o8QcP8I/AAAAAAAAAGRU/MiKaXITzhg/s320/ingles_ley_sin">http://3.bp.blogspot.com/_RmE17b85Cw/Tt5o8QcP8I/AAAAAAAAAGRU/MiKaXITzhg/s320/ingles_ley_sin</a>
	5/30/2011	SINDEMOCRACIA	<a href="http://www.noiesvotes.com">www.noiesvotes.com</a>	Barcelona	Cami del Bosc Can Soberano, 1-29	2.340381	41.498114	0 €	Spanish Revolution	faf	
15	5/31/2011	Slayer	<a href="http://www.slayer.com">www.slayer.com</a>	Barcelona	08338 Premià de Dalt	2.170394	41.386409	50 €	Metal Infernal	Metal Hurraca	null
16											
	6/30/2011	Death Justin	<a href="http://www.myspace.com/">www.myspace.com/</a>	Madrid	Plaza de la Puerta del Sol, 14 28014 Madrid	-3.703308	40.416898	gratis	mola	vinny	null
17											
	7/3/2011	Milencollin	<a href="http://www.milencollin.com">www.milencollin.com</a>	Premià de Mar	Carrer de Francesc Moragas, 38-40 08330 Premià de Mar	2.357579	41.490846	barato	grup de salsa	tu	null
18											
	7/3/2011	PennyWise	<a href="http://www.pennywise.com">www.pennywise.com</a>	Premià de Mar	Carrer de Francesc Moragas, 38-40 08330 Premià de Mar	2.357579	41.490846	barato	grup de salsa	tu	null
19											
	12/28/2014	Cicatriz	<a href="http://www.myspace.com/">www.myspace.com/</a>	Mataró	Carrer de Jordi Joan, 14 08301 Mataró	2.450523	41.539247	2 Lerodos	Punk del Buenol	Espinete	null
20											
	1/1/2015	Piperrak	<a href="http://www.myspace.com/">www.myspace.com/</a>	Premià de Mar	Carrer dels Capitans de Mar 08330 Premià de Mar	2.367584	41.49756	48	trauma	kabrachunga	<a href="http://www.goo.es/mq.jpg">www.goo.es/mq.jpg</a>

Fig. 6.2.1.1 Pàgina del full de càlcul d'ordenació per dates.

És important destacar que, gràcies a aquesta estructura, no s'ha de fer cap manteniment del full de càlcul ni cap càrrega manual per part d'un administrador.

A la Fig. 6.2.1.2. es pot veure un esquema que mostra les operacions d'entrada/sortida a la base de dades des de *GeoShows*.

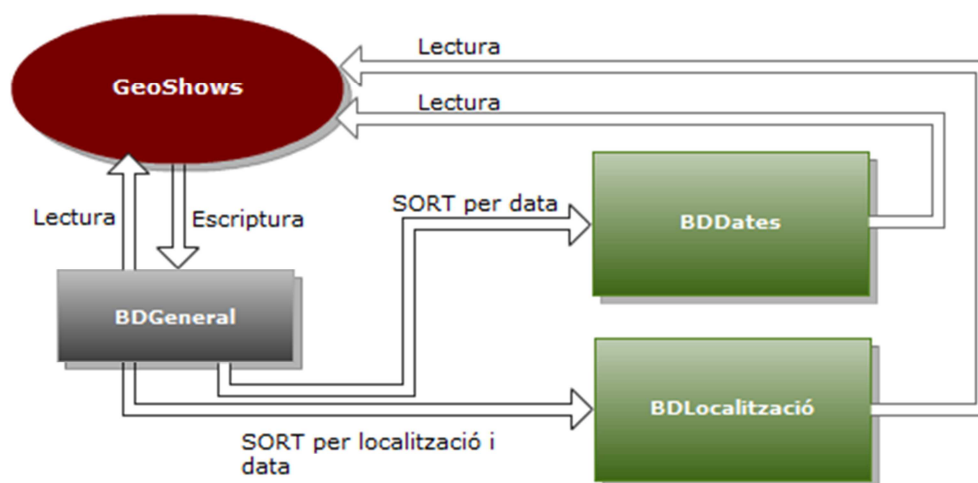


Fig. 6.2.1.2. Interacció amb la base de dades

## 6.2.2. Obtenció de concerts

Per obtenir els concerts de la base de dades, *Geoshows* utilitza la següent metodologia:

- S'ingressa a una conta concreta de *Google* per accedir a *Google Docs*.
- Es comprova mitjançant la llibreria *GSS\_Lib* si hi ha documents creats amb *Google SpreadSheets* i es guarda un índex de posició d'on es troba cadascun d'ells.
- Es recull en un manegador l'índex del full de càlcul (base de dades) que es vol utilitzar.
- Es selecciona la pàgina del full de càlcul que es vol utilitzar: *BDDates* o *BDlocalització* (veure Fig. 6.2.1.2).

- Si hi ha registres antics ( la data actual supera la data del concert) s'accedeix a BDGeneral i es borren les entrades antigues. Després es repeteix el pas anterior.
- Es representen en una llista totes les entrades de la base de dades. A la llista apareixen els camps *GRUP, DATA, i LOCALITAT*.
- En el cas de prémer un dels registres, es representen els camps d'aquell registre en una pantalla de presentació de dades, menys els camps *LATITUD i LONGITUD*.
- En cas de prémer el botó "GeoShow it!" es recullen els camps *LATITUD i LONGITUD* per presentar al mapa la localització del concert, a part de mostrar també la teva posició actual, mitjançant GPS.

A la figura 6.2.2.1 es veuen les vistes dels passos anteriors.

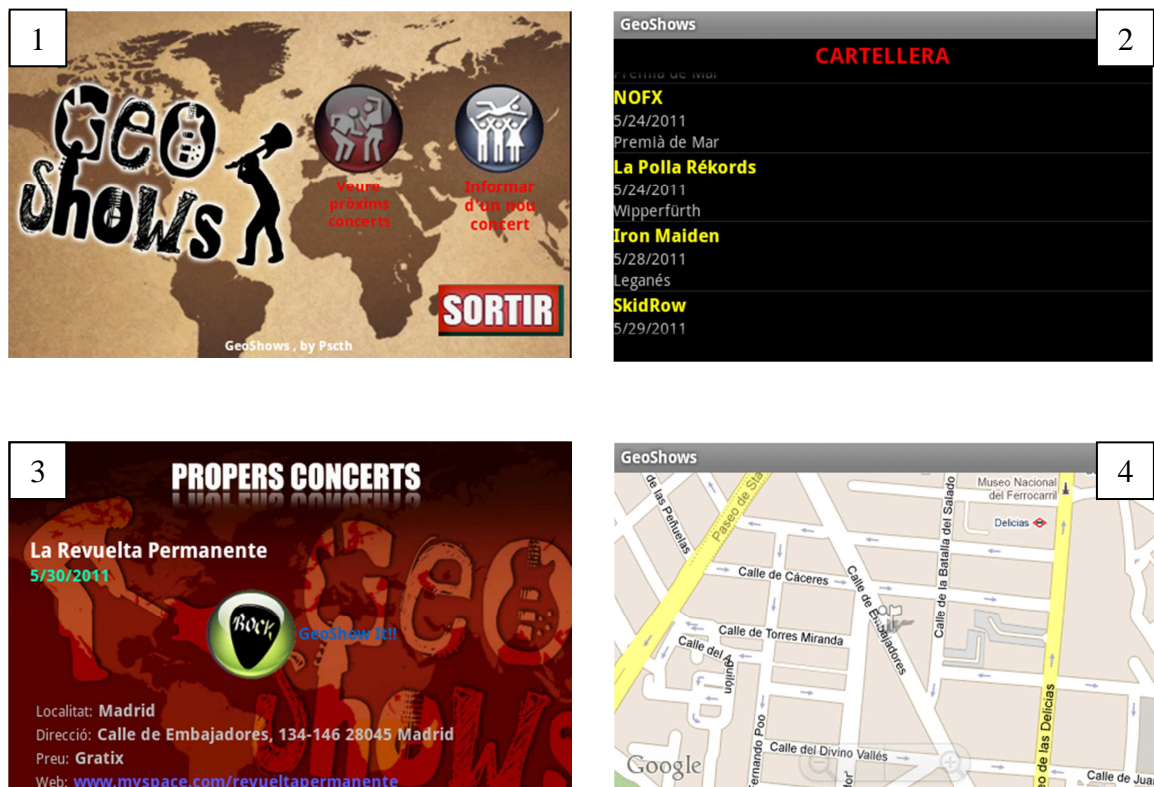


Fig 6.2.2.1. Obtenció de concerts

### 6.2.3. Pujar concerts

Per pujar un concert a la base de dades, l'aplicació sol·licita omplir un formulari amb tots els camps necessaris. És obligatori omplir tots els camps; si no, no es pot pujar el concert. Les pantalles del procés es poden veure a la figura 6.2.3.1. El procediment a seguir per pujar un concert és el següent:



Fig 6.2.3.1. Pujada de concerts

- Omplir camps del formulari. La data i la localització (localitat i direcció) tenen mètodes d'entrada exclusius ( no són editables a mà). La data s'omple gràficament amb uns quadres desplegable, i la localitat mitjançant un mapa de *Google Maps*, fent click al destí.) . Els altres camps no estan controlats.
- Es comprova que tots els camps han estat omplerts, i si no és així s'informa al usuari que ha d'acabar d'omplir-los. Si tots els camps són plens, es recull la informació, i es puja a la base de dades BDGeneral.

- S'entra al compte de Google Docs que conté la base de dades. Anàlogament al procés de obtenció de concerts (tema 6.2.2.) s'obté un manegador per controlar la BDGeneral.
- A la base de dades es crea una nova entrada amb la informació rebuda de l'usuari.
- L'aplicació confirma que el procediment de pujada s'ha completat.

## 6.3. Ús de l'aplicació

*GeoShows* es divideix en dos funcionalitats bàsiques: *Veure Pròxims concerts* i *Informar d'un nou concert*.

### 6.3.1. Veure Pròxims concerts.

Per accedir a aquesta funcionalitat, s'ha de fer *click* al botó *Veure pròxims concerts* al menú principal. (veure Fig.6.2.2.1 cas 1). Si no estem connectats a Internet sortirà un missatge que dirà que activem la connexió a Internet per accedir a l'aplicació.

Si estem connectats a Internet i es *clicka* la opció esmentada anteriorment, després d'un temps de càrrega (s'accedeix des de Internet a la base de dades de *Google SpreadSheets*), es mostrarà en una llista tots els concerts disponibles ordenats per data (veure Fig.6.2.2.1 cas 2). A la llista, de cada concert s'hi mostra el nom del grup, la data, i la localitat. Tenim la possibilitat d'ordenar per localitat les entrades de la llista prement el botó de servei *Menu* dels terminals *Android*, i seleccionant la opció *Ordenar per localitat*.

Si es fa *click* a sobre de qualsevol concert de la llista, s'obrirà una altra pantalla en la que es veurà informació detallada de l'esdeveniment marcat (veure Fig.6.2.2.1 cas 3).

Desde la finestra de presentació de dades de l'esdeveniment, s'hi troba un botó que permet veure la localització del concert de forma gràfica a través d'un mapa de *Google*

*Maps*. Allà on sigui situat el concert apareixerà una icona en forma de “home pal” subjectant una bandera (veure Fig.6.2.2.1 cas 4). El mapa permet controls de posició (desplaçar la vista del mapa) i de zoom (augmentar la vista).

### 6.3.2. Informar d'un nou concert.

Per accedir a aquesta funcionalitat, s'ha de fer *click* al botó *Informar d'un nou concert* al menú principal. (veure Fig.6.2.3.1 cas 1).

Es presentarà un formulari d'introducció de dades en el que s'haurà d'introduir els camps: *Data*, *Grups*, *Localitat*, *Direcció*, *Preu*, *Comentaris*, *Web*, *Informant* i *Fotografia* (veure Fig.6.2.3.1 cas 2). El mètode d'introducció de la data del concert és a través del botó *Data*. S'obrirà un quadre emergent on es pot introduir la data de forma gràfica. La casella de text de la data no és modificable per entrada per teclat. El mètode d'introducció de la localitat i direcció és a través del botó *Localitza!*. Al prémer el botó, s'obrirà una pantalla amb un mapa de *Google Maps*. Fent *click* a qualsevol lloc del mapa obtindrem els camps *Localitat* i *Direcció*, amb possibilitat de modificar només després el camp *Direcció* per afinar la informació (veure Fig.6.2.3.1 cas 3). La casella de text *Localitat* no és modificable per entrada per teclat. Tots els altres camps s'han d'omplir mitjançant entrada per teclat. El camp fotografia espera la introducció d'una *URL* que contingui una imatge.

A continuació es fa *click* a sobre del botó *Puja la informació*. Si no s'han omplert tots els camps, s'informarà de que falten camps per omplir (s'informa que el camp buit de més amunt a la vista no és ple). En cas contrari, es puja la informació del concert a la base de dades, i saltant a la pàgina principal de l'aplicació s'informa amb un missatge de que la informació ha sigut correctament enviada (veure Fig.6.2.3.1 cas 4).



## 7. Estudi econòmic

### 7.1 Cost de l'aplicació.

#### 7.1.1. Cost del material.

No hi ha costos materials del projecte, donat que l'aplicació es distribueix en format digital i a través del repositori oficial de *Android Market*, éssent aquest gratuït.

#### 7.1.2. Costos de recursos humans.

Els costos de recursos humans són produïts per el salari de l'enginyer que realitza l'estudi i desenvolupament de l'aplicació (desglossat en dos termes), i el càrrec administratiu encarregat de la redacció de la memòria del projecte. Les dades es poden veure agrupades a la taula 7.1.2.1.

<u>Concepte</u>	<u>Hores</u>	<u>Preu/hora (€)</u>	<u>Total (€)</u>
Estudi i documentació (Enginyer sènior)	60	30	1800
Disseny (Enginyer sènior)	150	30	4500
Redacció memòria (Càrrec administratiu)	50	15	750
<b>TOTAL RECURSOS HUMANS</b>			<b>7050</b>

Taula 7.1.2.1 Costos de recursos humans

Al ser la primera aplicació desenvolupada per l'enginyer, requereix un nombre de hores extra d'aprenentatge del llenguatge i les eines d'investigació, que fa que el cost en recursos humans s'elevi.

### 7.1.3. Amortització equips, instrumental i software.

En el projecte, s'ha utilitzat bàsicament un ordinador amb sistema operatiu *Windows 7 Professional Edition (x64)* ( tot l'altre software utilitzat és programari lliure). Les amortitzacions de l'equipament utilitzat es poden veure a la taula 7.1.3.1. Es desglossa el cost en forma de percentatge de l'amortització total del cost dels equipaments, durant la realització del projecte (quatre mesos).

<u>Equip utilitzat</u>	<u>Percentatge del cost amortitzat</u>	<u>Total</u>
<u>Equips i programari informàtic</u>		
Ordinador (1000€)	25%	250
Software Microsoft Windows 7 (x64) Professional Edition (150,90 €)	25%	37,7
<u>Instrumentació electrònica</u>		
Telèfon <i>Android</i> (600 €)	25%	150
<b>TOTAL AMORTITZACIONS</b>		<b>437.7</b>

Taula 7.1.3.1 Amortització de l'equipament

### 7.1.4. Despeses indirectes.

Les despeses indirectes, per a l'ús de les instal·lacions on s'ha desenvolupat el projecte, el consum d'energia elèctrica, aigua, lloguer del local i connexió telefònica i ADSL, per un valor del 10% del cost total del projecte.

### 7.1.5. Cost de l'aplicació total.

Els costos totals de desenvolupament de l'aplicació, desglossats a la taula 7.1.5.1.

Costos de material	0 €
Costos de recursos humans	7050 €
Costos d'amortització	437,7 €
Subtotal	7487,7 €
Despeses indirectes (10%)	748,8 €
<b>TOTAL</b>	<b>8236,5 €</b>

Taula 7.1.5.1 Cost total de l'aplicació

## 7.2. Preu de venda en el mercat.

### 7.2.1. Càlcul del cost total per unitat.

La aplicació es distribueix mitjançant repositoris de *Android Market*, sent aquests gratuïts i de propietat de *Google Inc.* La còpia i distribució no té cost.

Paral·lelament, al ser una aplicació amb llicència *GPL* pot ser copiada i distribuïda lliurement entre els usuaris.

### 7.2.2. Càlcul del benefici.

El cost obtingut per unitat és de 0 €. S'ha decidit enlloc de vendre l'aplicació, distribuïda de manera gratuïta, però amb publicitat web de *AdSense* amb beneficis per *click*. Els càlculs dels beneficis es poden veure desglossats a la taula 7.2.2.1.

Cost total per unitat	0
Cost desenvolupament	8236,5 €
<b>COST TOTAL</b>	<b>8236,5 €</b>
Benefici per click	0.07 €
Clicks anuals estimats	10000
<b>Ingrès anual</b>	<b>700 €</b>

Taula 7.2.2.1. Càlcul del benefici

El preu de les aplicacions a *Android Market* acostuma a ser bastant baix, d'entre 0 € i 5 €. Moltes aplicacions es distribueixen de manera gratuïta amb la inclusió de publicitat. El model de distribució gratuïta contribuirà a fer una base de dades més sòlida afavorint el creixement de la comunitat usuària.

El bagatge obtingut durant el desenvolupament de l'aplicació servirà per abaratir el cost de futures aplicacions en *Android*. Al ser la primera aplicació desenvolupada en aquest entorn, l'amortització total dels costos d'aquesta primera es basa en el desenvolupament continu de varies aplicacions. S'ha de deixar clar, que si la aplicació es manté, els beneficis s'han calculat de forma anual.

Hi ha altres pàgines web que ofereixen serveis similars, com [www.manerasdevivir.com](http://www.manerasdevivir.com) o [www.alternativa.cat](http://www.alternativa.cat) però a través d'una interfície web, incloent també el sistema gratuït amb ingressos per publicitat.

## 8. Conclusions.

En la memòria d'aquest projecte s'ha vist què és *Android*, quina estructura té el sistema operatiu, com s'executen les aplicacions i recursos disponibles.

Per altra banda, s'ha mostrat com és l'entorn de programació *d'Android* i què es necessita per muntar-lo, així com un manual d'instal·lació detallat de tots els seus components.

S'ha observat quins són els coneixements previs a adquirir abans de començar un projecte, i la solució adoptada després de l'aprenentatge, *GeoShows*.

El desenvolupament de l'aplicació ha suposat un aprenentatge fort en el món de la programació orientada a objecte, en aquest cas amb el llenguatge *Java* a *Android*. També ha obert una porta a conèixer les noves tecnologies per a mòbils intel·ligents (*smartphones*), una tecnologia que està emergent al mercat ara mateix.

El fet d'haver de desenvolupar un projecte en quatre mesos, amb un termini d'entrega acotat, ha fomentat l'organització i la planificació dels processos a seguir per completar-lo i acotar els seus objectius.

S'ha obviat un estudi d'impacte mediambiental, degut a que el procés de desenvolupament de l'aplicació no té repercussions importants sobre el mediambient, més que el consum energètic que representa la utilització d'un ordinador, i en tot cas, la impressió de la memòria. Les emissions de radiació dels dispositius mòbils no entren dins del projecte, ja que aquest es basa en el desenvolupament d'aplicacions, i no en els terminals en sí.

## **8.1. Punts forts i fluixos de l'aplicació.**

### **8.1.1. Punts forts.**

- Al ser una aplicació per mòbil, es pot accedir a les dades que subministra l'aplicació des de qualsevol lloc, sempre i quan hi hagi connexió a Internet.
- El fet de subministrar la localització de forma gràfica dóna valor afegit a l'aplicació, ja que es pot assimilar les distàncies entre la posició del usuari i la localització del concert molt millor, evitant que l'usuari hagi de buscar on es troba l'adreça subministrada.
- Al comptar amb una base de dades pública per els usuaris de l'aplicació (accessible només per això des de la pròpia aplicació) i automatitzada (ordenació d'esdeveniments, eliminació d'entrades antigues), es pot prescindir d'un manteniment constant d'aquesta.
- Com que els propis informants dels concerts albergats a la base de dades són els usuaris, a part d'aconseguir una aplicació desatesa per banda d'un administrador que s'encarregaria d'actualitzar les entrades i posar-ne de noves, el valor de l'aplicació creix exponencialment a mesura que n'augmenta el nombre d'usuaris actius.

### **8.1.2. Punts fluixos.**

- Introduir un concert a la base de dades a través d'un terminal mòbil pot ser més o menys feixuc. Els teclats dels terminals mòbils són petits, i la pantalla també ho és. Moltes vegades és necessari consultar certes fonts per conèixer tota la informació del concert, i tenir en pantalla tant la informació com el formulari d'introducció de dades.

- El fet que la base de dades estigui allotjada a *Google Docs*, no garanteix cap tipus de seguretat pel que fa a la destrucció del compte, dels documents inclosos a dins del mateix, o la disponibilitat de connexió als servidors.
- Si la introducció de dades de localització està limitada a l'ús dels mapes de *Google Maps*, cau la possibilitat que els servidors de *Google* no estiguin disponibles i llavors la part de l'aplicació de pujada d'informació deixi de ser operativa.
- Si la base de dades creix molt, tot i la ordenació de concerts per dates i localitats, pot ser incòmode buscar un esdeveniment concret, a causa de tenir poc espai per representar totes les dades (la pantalla d'un terminal és petita). D'aquesta manera, s'ha de recórrer un "scroll" vertical que pot arribar a ser molt llarg.

## 8.2. Perfil de l'usuari.

GeoShows és una aplicació encarada a tots els públics, en especial per aquells que surten de concert, esporàdicament o sovint. Al utilitzar noves tecnologies i estar relacionada amb activitats d'oci més aviat nocturn, l'ús de l'aplicació recau més sobre el perfil d'una persona jove o de mitjana edat.

## 8.3. Possibles millores a l'aplicació.

La primera versió de l'aplicació inclou les funcionalitats bàsiques que s'han volgut dur a terme en projecte. Tot i així, és necessari avaluar tots aquells detalls que semblen millorables, i que en un futur poden donar més qualitat a l'aplicació. Les millores que es podrien implementar són:

- Control d'identificació d'usuaris: crear una altra pàgina a la base de dades on allotjar el nom dels usuaris i les seves contrasenyes per accedir a l'aplicació.

Així es tindria un control més gran sobre qui puja els concerts, per així identificar si les fonts són fiables, amb un sistema de puntuacions per exemple.

- Afegir més camps a la base de dades com la hora del concert, o el gènere musical.
- Tapar les transicions entre *Activities*, quan es queda la pantalla en negre de mentre es carrega el *background* o s'executen fluxos de processos llargs.
- Mostrar tots els concerts de la base de dades alhora sobre un mapa, i accedir-hi mitjançant un *click* sobre l'esdeveniment.
- Sincronització amb l'aplicació *Navigation* per poder traçar una ruta des de la teva posició actual fins al concert, i guiar-te amb el sistema GPS.
- Mètode de càrrega de fotografies: disposar d'un servidor on poder allotjar les fotografies dels grups. Carregar les fotos mitjançant URL pot ser una mica engorros.
- Optimització del codi de l'aplicació en general per un funcionament més fluid.
- Correcció d'un *bug* que fa que l'aplicació es tanqui forçosament després de fer *click* al botó de servei de tornar enrere del dispositiu.
- Incloure varis idiomes per tal de fer-la de més gran abast
- Incloure un sistema de gestió de les ubicacions (recerca per províncies, països, ciutats o poblacions) per poder encarar una expansió internacional de l'aplicació.
- Millora de les implementacions dels *layouts* per adaptar l'aplicació a diverses resolucions i densitats de pantalla.



## 9. Referències.

[1] <http://developer.android.com>

Pàgina oficial d'Android on s'hi troba tota la documentació sobre classes i objectes utilitzats en desenvolupament d'aplicacions, així com també la estructura del sistema operatiu i les aplicacions

[2] <http://www.elandroidelibre.com>

Pàgina web dedicada exclusivament a Android. Hi apareixen notícies sobre aplicacions, noves versions del sistema, i terminals. També hi conté tutorials de iniciació a la programació en *Android*.

[3] <http://android-developers.blogspot.com/>

Blog dedicat a la programació en Android. Hi ha diversos tutorials força específics per realitzar accions concretes.

[4] <http://stackoverflow.com/>

Fòrum dedicat al món de la programació en general, on s'hi poden trobar moltes preguntes freqüents i casos resolts sobre la programació en Android.

[5] <http://google.com>

Motor de cerca utilitzat per buscar totes les referències utilitzades durant el desenvolupament de l'aplicació.

[6] <http://www.htcmania.com>

Pàgina web dedicada als terminals Android de qualsevol marca (tot i el nom de la pàgina), amb seccions tant de hardware, software, o programació.

[7] <http://glffy.com>

Conté un editor d'esquemes web molt potent.