

Treball de Final de Carrera
Universitat Oberta de Catalunya

Desenvolupament d'aplicacions per a dispositius mòbils
amb plataforma Android

Memòria

Autor : Artur Dinaret Dorca
Tutor : Marc Domingo Prieto
Juny 2012

Índex de Continguts

1. Introducció	4
1.1 Motivació del Projecte	4
1.2 Objectius	5
1.3 Recursos	6
1.3.1 Hardware	6
1.3.2 Software	7
1.4 Estructura del projecte	7
1.4.1 Activitats	7
1.5 Calendari del Projecte	8
1.5.1 Diagrama de Gantt	9
1.6 Contingut de la memòria	10
2. Estat de l'Art	11
2.1 Dispositius mòbils	11
2.1.1 Classificació dels dispositius mòbils	12
2.1.2 Sistemes operatius per dispositius mòbils	13
2.2 Estat actual del mercat, aplicacions d'estalvi	14
3. La plataforma Android	16
3.1 Arquitectura	17
3.2 Components d'una Aplicació	19
3.2.1 Activity	19
3.2.2 Broadcast Intent Receiver	20
3.2.3 Service	21
3.2.4 Content Provider	21
3.3 Cicle de vida de les aplicacions android	22
3.4 Eliminació de processos	24
3.5 API Demos	25
3.6 Contingut d'un projecte Android	26
3.6.1 Carpeta \src	26
3.6.2 Carpeta \res	28
3.6.3 Carpeta \bin	30

3.6.4 Arxiu AndroidManifest.xml	30
3.7 Interfícies d'usuari en Android	32
4. Where's my Money? : Aplicació d'estalvi.....	34
4.1 Anàlisis i disseny de la aplicació.....	34
4.1.1 Casos d'ús.....	35
4.2 Desenvolupament e implementació.....	37
4.2.1 Package del projecte	38
4.2.2 Característiques Principals	39
4.2.3 Idiomes.....	45
5. Proves i resultats	46
5.1 Funcionalitat Ingrés	47
5.2 Funcionalitat Consulta	54
5.3 Funcionalitat Històric	56
5.4 Funcionalitat Estalvi.....	58
6. Conclusions i possibles ampliacions.....	60
6.1 Conclusions generals	60
6.2 Ampliacions futures.....	61
7. Bibliografia i referències.....	62
7.1 Llibres	62
7.2 Llocs Web	63

1. Introducció

En les següents línies es fa una breu introducció al present projecte, exposant quina ha estat la seva motivació, quins són els objectius que persegueix i quins són els continguts d'aquesta memòria.

1.1 Motivació del Projecte

Aquest projecte està format per dues parts. Per una banda, l'aplicació desenvolupada neix del moment econòmic en el que ens trobem i de les necessitats creixents de la gent per controlar la seva despesa diària. Donada la tessitura econòmica actual l'usuari potencial de l'aplicació és aquell que necessita controlar la seva despesa diària i vol saber en tot moment quin és el balanç econòmic diari en el seu dia a dia.

Actualment és molt important portar un control de les nostres finances diàries, ja que qualsevol excés pot repercutir en el bon funcionament de la nostra economia particular, per això l'aplicació "Where is my Money?" està orientada al petit estalviador que vol saber en tot moment de quins recursos disposa i quina millor manera que implementar una aplicació on l'usuari pot tenir accés en tot moment a la consulta o actualització d'aquesta informació. Per això els dispositius mòbils, com les tables i sobretot els smartphones, són una eina fonamental ja que de forma habitual portem aquests dispositius constantment amb nosaltres i la seva facilitat d'accés afavoreixen una actualització constant d'aquestes dades sense l'esforç d'haver d'accedir a un ordinador o un full de càlcul després de molta estona d'haver fet l'ingrés o la despesa en el balanç personal, amb el risc d'oblidar d'aquestes dades.

L'altre part del projecte es basa en un petit recorregut a forma d'introducció de les principals característiques de la plataforma Android i dels seus components tant a l'hora de l'execució com en el seu desenvolupament.

És creixent el nombre de terminals que utilitzen aquest sistema i com a desenvolupador considero important el conèixer aquest nou sistema operatiu que tantes possibilitats dona

a l'hora de crear les nostres pròpies aplicacions.

D'aquesta manera qualsevol persona que llegeixi aquest treball sigui capaç d'entendre una mica millor com funciona el procés d'execució i de creació d'una aplicació d'Android.

Android es llança sota una llicència de software lliure, el que implica que qualsevol desenvolupador té accés complert al SDK del sistema, incloses totes les seves API, documentació i emulador per proves.

A més aquesta llicència permet als desenvolupadors publicar les seves creacions o distribuir-les de pagament ocultant el codi font. Aquest nou sistema introdueix també interessants conceptes com la composició d'aplicacions mitjançant mòduls o blocs bàsics, segons la naturalesa de l'aplicació, o la delegació en el sistema de determinades accions per a que siguin altres aplicacions instal·lades les que es facin càrrec d'elles.

1.2 Objectius

L'objectiu d'aquest Treball de Final de Carrera es centra sobre dos eixos bàsics. Per una banda es pretén fer un apropament a l'estudi de l'entorn de desenvolupament d'Android, al sortir al mercat sota una llicència de software lliure, l'SDK està a disposició de qualsevol desenvolupador que vulgui descarregar-lo. Aquest inclou moltes i variades ajudes per començar a fer-lo servir, des de les API de totes les classes fins a eines de programació inclòs un emulador per a realitzar els tests. Tots aquests elements seran objectes d'estudi i d'explicació en aquest treball.

Per altre banda, una vegada conegudes les característiques del sistema i el seu entorn de desenvolupament, es realitzarà una aplicació que aprofitarà algunes de les seves característiques fonamentals. Mitjançant aquest desenvolupament juntament amb una documentació detallada, el lector ha de poder comprendre el funcionament de les aplicacions per Android i conèixer els passos per crear les seves pròpies.

En quant a l'aplicació a desenvolupar, rep el nom de Where's My Money. Es tracta d'una

app per android que pretén ser útil i funcional, amb una interfície senzilla i amigable, de gran usabilitat, orientada al seu ús diari i que té com a principal objectiu el control de la petita i gran despesa de l'usuari durant el seu dia a dia, donant la opció de poder classificar-les per famílies, veient en tot moment quins són els seus moviments tant en positiu com en negatiu. Amb la possibilitat d'oferir reports de les activitats, ja sigui les referents al mes en curs com en anteriors, poden consultar un històric en tot moment de mesos enrederits que ofereix, aquesta informació dividida per famílies. També és una aplicació amb la voluntat de fomentar l'estalvi, per tant l'usuari podrà en tot moment configurar quin tant per cent dels seus ingressos desitja destinar a l'estalvi cada mes rebent alertes en el moment en que s'aproximi a aquesta quantitat i la sobrepassi. L'aplicació també tindrà la capacitat de recomanar-nos un tant per cent ideal segons quins siguin els ingressos de l'usuari i aconsellar-nos en quina de les famílies tenim més despesa i podríem reduir aquesta quantitat.

D'aquesta manera es pretén que l'usuari prengui el control dels seus diners anotant on se'ls gasta i controlant els seus deutes en el moment exacte en que passen, gràcies a que una eina tant potent com un mòbil sempre la portem amb nosaltres.

1.3 Recursos

En aquest apartat es destaquen els recursos que es faran servir per al desenvolupament del projecte.

1.3.1 Hardware

- Equip : Ordinador Sobretaula amb processador AMD Athlon 7750 Dual Core 2,7GHz
- 3 Gb de memòria ram amb disc dur de 320 Gb
- Tarjeta Gràfica GeForce 8200 de 1Gb de memòria compartida amb DirectX 10.0
- Sistema Operatiu Microsoft Vista Home Premium de 32 bits

1.3.2 Software

- Paquet Android SDK per Windows
- Entorn de desenvolupament Eclipse Indigo v.3.7.0
- Smartphone LG P970 Optiums Black amb versió de sistema operatiu Android 2.2 Froyo

L'elaboració de tots els documents es portarà a terme amb el paquet d'aplicacions ofimàtiques Open Office v3.3

1.4 Estructura del projecte

1.4.1 Activitats

Planificació del Projecte – PAC 1. *Dates : 01/03/2012 fins 12/03/2012*

Es lliura un document, que serveix per una banda per presentar el projecte, la seva voluntat i les seves funcionalitats. Donant una visió de les tasques a realitzar i el temps en que es divideix aquesta realització.

Anàlisis de l'entorn de desenvolupament Android – PAC2. *Dates: 13/03/2012 fins 09/04/2012*

Es pretén donar una visió tècnica de les opcions que ens ofereix SDK d'Android, el treball no només pretén el desenvolupament d'una aplicació per Android si no que pretén fer un anàlisis dels conceptes més rellevants dins d'aquest nou paquet de desenvolupament. Els temes a tractar seran :

- Arquitectura
- Components d'una aplicació (Activity, Broadcast Intent Receiver, Service, Content Provider)
- Cicle de vida de les aplicacions Android
- Eliminació de processos
- Gestió de la informació
- API demos

- Funcions de l'emulador
- Contingut d'un projecte Android
- Interfícies d'usuari amb XML

Desenvolupament de l'aplicació Where's My Money – PAC 3. Dates:

10/04/2012 fins 21/05/2012

Explicació pas a pas del desenvolupament de Where's My Money, aplicació amb la voluntat de donar la possibilitat a l'usuari de controlar la seva despesa. L'objectiu d'aquest desenvolupament, al marge de l'interès en fer una aplicació usable i original que aportï noves funcionalitats dins del camp de les aplicacions d'estalvi, és el d'il·lustrar de forma pràctica alguns dels conceptes esmentats en la primera part del treball (PAC 2). Aquesta feina requereix les següents activitats :

- Construcció de les classes identificades per l'aplicació
- Realització de proves unitàries de cada classe construïda
- Realització de les proves integrades del funcionament general de l'aplicació
- Revisió del codi desenvolupat per comprovar que segueix els patrons de programació orientada a objectes així com la realització de la seva documentació.

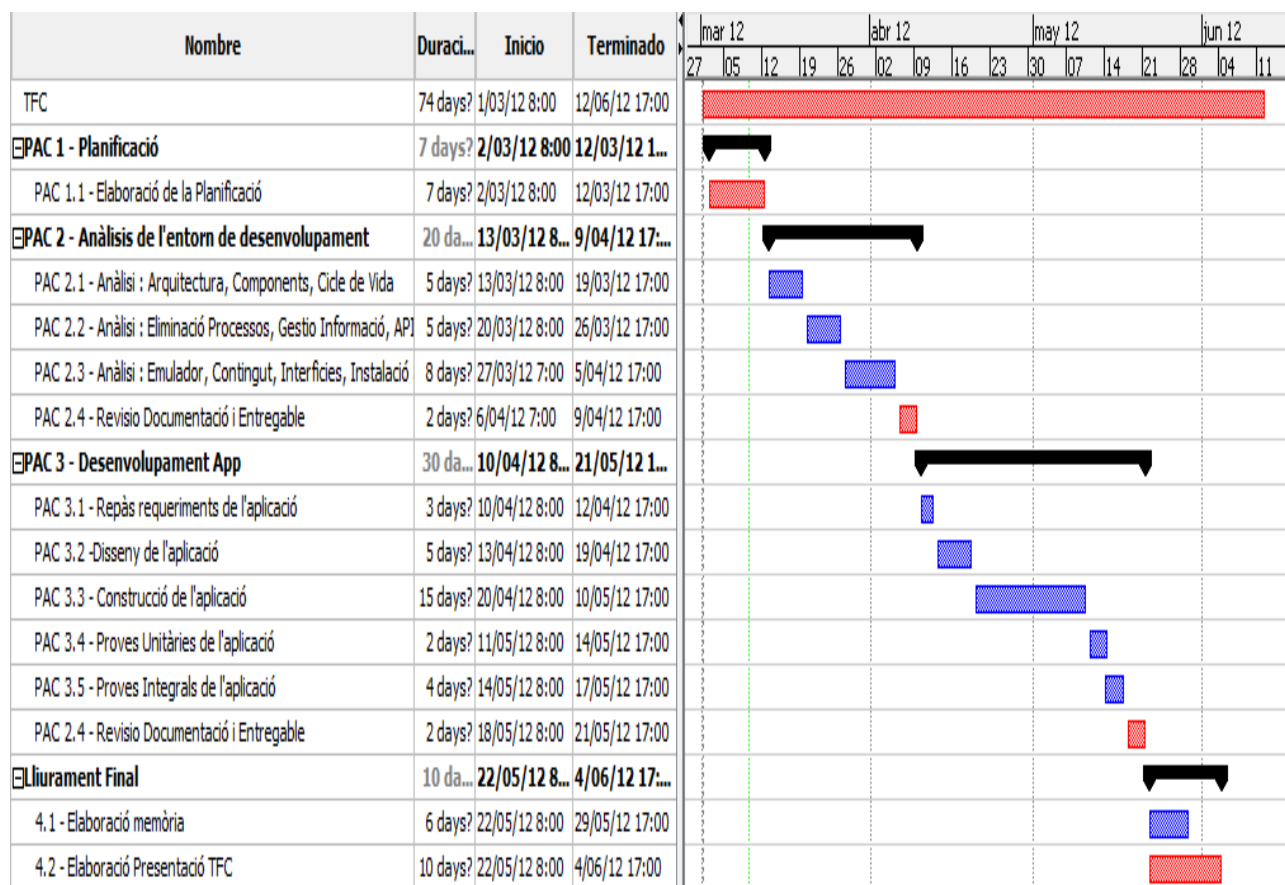
Lliurament Final. Dates: 22/05/2012 fins 12/06/2012

Aquesta tasca engloba la realització i l'entrega dels documents finals per fer la presentació d'aquest Treball Final de Carrera

1.5 Calendari del Projecte

Degut a la meua feina com a programador a IBM, la meua disponibilitat es variable durant les setmanes que dura el projecte i em veig amb dificultats de fixar una jornada de dedicació fixa. De totes maneres es fa una aproximació en un rang de entre 4 y 7 hores de dedicació setmanal. Contant que aproximadament la duració del treball es de 14 setmanes, fem el següent diagrama de gantt utilitzant l'eina de programari lliure Open Project

1.5.1 Diagrama de Gantt



1.6 Contingut de la memòria

A continuació s'explica breument el contingut de cada capítol inclosos en aquesta memòria :

Capítol 1, Introducció : s'exposa la motivació del present projecte, els objectius que persegueix i la visió general dels continguts de la memòria

Capítol 2, Estat de l'art : ofereix descripció general d'alguns dels aspectes tècnics relacionats amb aquest projecte, com els dispositius mòbils a més d'un breu anàlisi de l'estat del mercat en quan a aplicacions d'estalvi.

Capítol 3, La plataforma Android : s'exposa al lector les característiques bàsiques del sistema operatiu Android, el seu disseny, la arquitectura i el funcionament així com un petit exemple amb la implementació d'una aplicació "Hola mundo".

Capítol 4, Where is my Money? : es presenta l'aplicació desenvolupada durant el present treball final de carrera fent un anàlisi del disseny i les seves funcionalitats. S'explica el desenvolupament de les seves classes i components bàsics

Capítol 5, proves i resultats : es posa a prova l'aplicació esmentada, realitzant funcions habituals, exposant i analitzant els resultats obtinguts

Capítol 6, conclusions i possibles ampliacions : s'exposen i analitzant els resultats obtinguts al finalitzar del treball, comparant-los amb els objectius marcats inicialment, indicant possibles línies de millora futures en el desenvolupament.

Capítol 7, bibliografia : Llistat detallat de la bibliografia utilitzada durant el projecte

2. Estat de l'Art

2.1 Dispositius mòbils

Definir l'estat actual dels dispositius mòbils és una part important d'aquest treball ja que no només els smartphones són els únics que poden muntar un sistema Android on correrà la nostra aplicació, encara que siguin els més comuns, tenim un ventall de possibilitats cada vegada més amplis on els desenvolupadors poden aprofitar nous elements per les seves creacions.

No existeix un consens clar a l'hora de definir que es realment un dispositiu mòbil i que no ho és. Es freqüent avui en dia utilitzar aquest terme per designar únicament certs models de telèfons mòbils amb majors o menors prestacions. Malgrat això, un dispositiu mòbil no te perquè ser només un dispositiu d'àmbit telefònic.

Si intentem ser més rigorosos, es podria designar un dispositiu mòbil com tot aquell aparell electrònic que compleix unes característiques fonamentals :

- És de mides reduïdes, i per tant fàcilment transportable
- Té una certa capacitat de computació i emmagatzematge de dades
- Incorpora elements d'entrada i sortida bàsics, pantalla i alguna mena de teclat

Més enllà d'aquestes característiques comuns, els dispositius mòbils formen actualment un grup extremadament heterogeni que pot incorporar gairebé qualsevol component de hardware o software que amplia la seva funció inicial, el més comú es la connexió telefònica o/i a Internet.

Igualment són habituals altres components com càmera fotogràfica o/i vídeo, pantalla tàctil, teclat qwerty, ràdio, bluetooth, memòria re movable , gps, etc. Des de el punt de vista de software poden incorporar també un ampli ventall d'aplicacions com programes

ofimàtics, reproductors àudio/vídeo, jocs, navegadors web, clients de correu, etc.

2.1.1 Classificació dels dispositius mòbils

De la mateixa manera que tenim problemes a l'hora de definir que és un dispositiu mòbil, la classificació que es pot fer d'aquest aparells es subjecta a diferents valoracions i a vegades no existeix un acord ampli per ubicar un dispositiu en una determinada família.

Durant la dècada dels anys noranta, punt de naixement dels primers dispositius mòbils, establir classificacions era menys o mes senzill degut a que cada aparell estava clarament definit per una funció determinada. L'augment de les prestacions i funcionalitats d'aquests aparells en l'actualitat dificulta el poder agrupar-lo dins d'un conjunt determinat, si posem un exemple, un smartphone representa una evolució del telèfon mòbil tradicional, es a dir la seva funcionalitat principal es la de oferir comunicació telefònica; però la realitat es molt diferents ja que aquest aparells tenen altres serveis addicionals com la connexió a Internet i les aplicacions, possibilitats mes pròpies en un ordinador. També ofereixen la possibilitat de reproduir pel·lícules o música, i fins i tot jugar a videojocs cada vegada mes elaborats i que poc tenen que envejar a les consoles portàtils.

Totes aquestes funcionalitats s'estan estenent cada vegada més ràpids a altres dispositius de recent aparició com son les tablets, aquests nous elements inclouen totes les funcionalitats dels smartphones però les amplien en característiques donar que corren en un component de hardware mol més potent que un telèfon convencional. Tenen més resolució, mes memòria i mes processador, entre d'altres, i fan que les possibilitats en les aplicacions augmentin. La diferència principal entre uns i els altres es dona en la funcionalitat d'establir trucades telefòniques que si inclouen els smartphones i no els tablets.

Donat aquesta visió del panorama general dels dispositius mòbils les possibilitats d'un desenvolupador android són molt grans i augmenten cada dia més ja que qualsevol dispositiu que corri un sistema Android es susceptible de ser objectiu de instal·lar aplicacions fetes per android i la diversitat del hardware que porten pot donar lloc a noves APIs aprofitin aquestes funcionalitats i obrint les portes de noves possibilitats a l'hora d'implementar les apps.

2.1.2 Sistemes operatius per dispositius mòbils

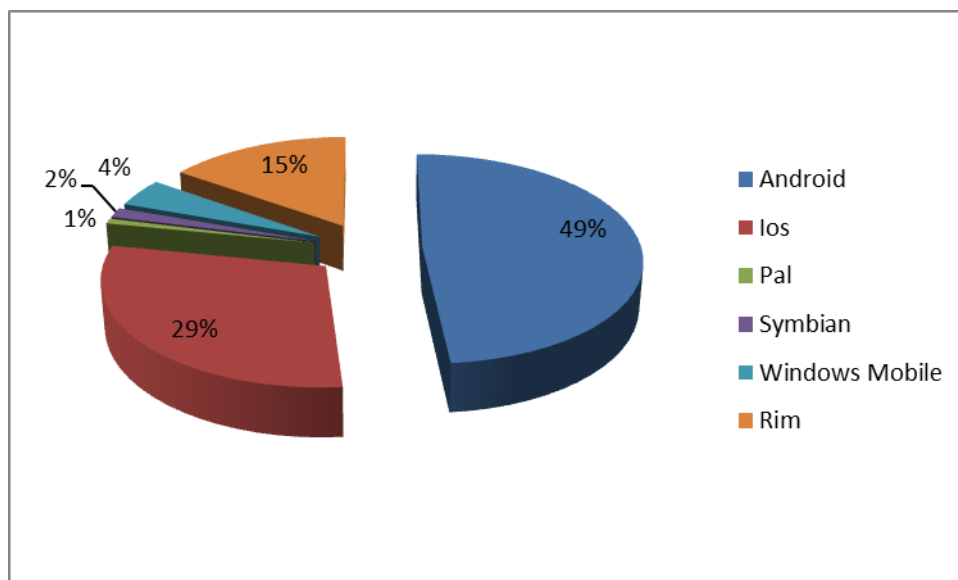
El sistema operatiu dissenyat per funcionar en un dispositiu mòbil requereix de dues característiques imprescindibles, ha de ser fiable i a de tenir una gran estabilitat, ja de d'una altra manera, les incidències habituals i tolerades en ordinadors personals com reinicis o caigudes no tenen cabuda en un dispositiu d'aquesta mena. A més ha d'adaptar-se a les limitacions de memòria i processament de dades que generalment tenen aquests dispositius proporcionant a més una execució ràpida a l'usuari.

Aquest sistemes han d'estar perfectament testejats i lliures d'errors abans d'incorporar-se definitivament a la seva comercialització. Les possibilitats que existeixen en els ordinadors normals de realitzar actualitzacions i reinstal·lar versions millorades per cobrir possibles errades o deficiències són més limitades en un dispositiu mòbil.

No és d'estranyar que un aparell d'aquesta naturalesa estigui funcionant ininterrompudament durant setmanes o mesos abans de ser apagat i reiniciat, a diferència amb el que passa amb un ordinador personal. De tal manera, el consum d'energia també és un tema molt delicat : es important que el sistema operatiu faci un ús racional de la bateria, ja que aquesta es limitada i la percepció d'autonomia per part de l'usuari és un fet exigent.

Tots aquest aspectes i d'altres han de ser tinguts en compte a l'hora de desenvolupar un sistema operatiu que sigui competent dins el mercat, atractiu pels fabricants i que permeti a l'usuari aprofitar al màxim les possibilitats del seu dispositiu.

Android forma part d'aquest grup selecte de sistemes operatius, tot i que comparteix la quota de mercat amb d'altres, ell s'emporta la major part.



Quota de mercat de diferents SSOO per smartphones (Gener – 2012)

2.2 Estat actual del mercat, aplicacions d'estalvi

L'estalvi per moltes persones suposa un gran repte i més en la situació econòmica actual en la que ens trobem. Per moltes persones s'ha tornat imprescindible porta un control de la seva despesa per tal d'equilibrar els seus balanços personals.

Per això dins del mercat podem trobar diverses opcions que en els últims temps estan augmentat donat una demanda creixent, cadascuna d'elles té unes particularitats que la fan diferent de l'anterior. Encara que comparteixen el mateix objectiu i en el fons parteixin del mateix principi, són els petits detalls els que fan que una aplicació sigui la favorita dels usuaris i líder de descàrrega.

Abans de començar el projecte vaig voler realitzar un test d'usabilitat en algunes d'elles, les escollides van ser, “Mis Cuentas” , “Easy Money”, “Money Wise” o “Mi dinero”.

Vaig agafar una mostra de cinc usuaris potencials amb perfils ben diferents, en tant a nivell d'estudis com a les seves capacitats i habilitats amb les noves tecnologies, de tal manera que aportessin al test una visió el més diferenciada possible.

Se'ls van proposar un seguit d'accions bàsiques i complexes amb l'aplicació com per

exemple, afegir ingressos o despeses, consultar els seus balanços general i per famílies, o configurar un ingrés fix en concepte de nòmina.

Durant la realització de les proves, els subjectes anaven comentant en veu alta les seves impressions i opinions. L'avaluador, en aquest cas jo mateix, anava prenent notes de cara a un disseny de "Where is my money?"

Les conclusions que es van extreure són força interessants i han influir en bona part en el disseny final de la aplicació.

- Una de les primeres opinions, que va ser comú en molts dels usuaris del test, va ser que l'aparença gràfica no era gaire intuïtiva i que existien problemes a l'hora de situar-se i trobar la forma de fer les accions.
- En molts casos, els literals que hi apareixien no eren representatius de les funcions que s'anaven a realitzar i confonien l'usuari, incrementant el temps de realització de les accions demanades
- Per incrementar la sensació de confusió, moltes d'elles només estaven en anglès, cosa que era pràcticament insalvable per les persones amb un grau de formació més baixa. En aquest cas, l'únic sistema de guiatge per l'aplicació eren les icones, moltes vegades poc descriptives. En aquest apartat podem destacar a "Easy Money" una de les aplicacions d'estalvi més descarregades
- Alguns dels usuaris enquestats, van destacar que moltes de les funcionalitats que proporcionava l'aplicació eren poc útils, i no veien en quin context es podrien utilitzar. Moltes vegades aquest excés de funcionalitats complicaven el seu ús

Al fer una recerca m'he adonat que n'hi moltes de molt completes com "Mis Cuentas", "Easy Money", "Money Wise" o "Mi dinero". Però presenten alguns problemes que al meu pare les fan poc usables.

Al acabar el test, i després d'haver analitzat les notes que vaig prendre durant la prova, vaig poder concloure que un dels problemes que tenen aquest tipus d'aplicacions són el seu alt grau de complexitat i la seva falta d'usabilitat.

Moltes d'elles actuen amb un sistema de comptes, que si bé pot ser molt útil als usuaris més experimentats o que hagin treballat més amb l'aplicació fan que per el gran públic siguin poc accessibles ja que la corba d'aprenentatge necessària és molt elevada i moltes vegades inclouen funcionalitats poc pràctiques o rarament usades.

Alguns errors i bugs estan presents en aquestes aplicacions que podem trobar al market d'Android i en molts casos no disposen de funcionalitats multi idioma i només estan disponibles en anglès.

Per molts d'aquest motius vaig concloure que la falta d'èxit definitiu d'una aplicació d'aquestes característiques dins el mercat residia en bona part en la falta d'equilibri entre usabilitat i funcionalitats. Que sigui agradable i fàcil d'usar, però que tingui la potencia suficient com per realitzar totes aquelles tasques que li demanem a una aplicació d'estalvi.

En aquest panorama “Where is my Money ?” ha estat pensada sobretot perquè sigui accessible per aquells usuaris menys experimentats. S'ha buscat una aplicació atractiva visualment, fàcil d'utilitzar i amb una corba d'aprenentatge molt reduïda. S'ha intentat dotar a l'aplicació d'un disseny molt intuïtiu i d'unes accions realment funcionals, deixant de banda opcions que en molts casos compliquen la utilització a l'usuari.

Algunes d'aquestes aplicacions existents en el mercat estan orientades al control d'economies més especialitzades i complexes, el cas de “Where is my Money?” s'orienta al control de la petita economia del dia a dia.

3. La plataforma Android

Android és un paquet de software pensat especialment per a dispositius mòbils que engloba tant un sistema operatiu, com *middleware* i diverses aplicacions d'usuari.

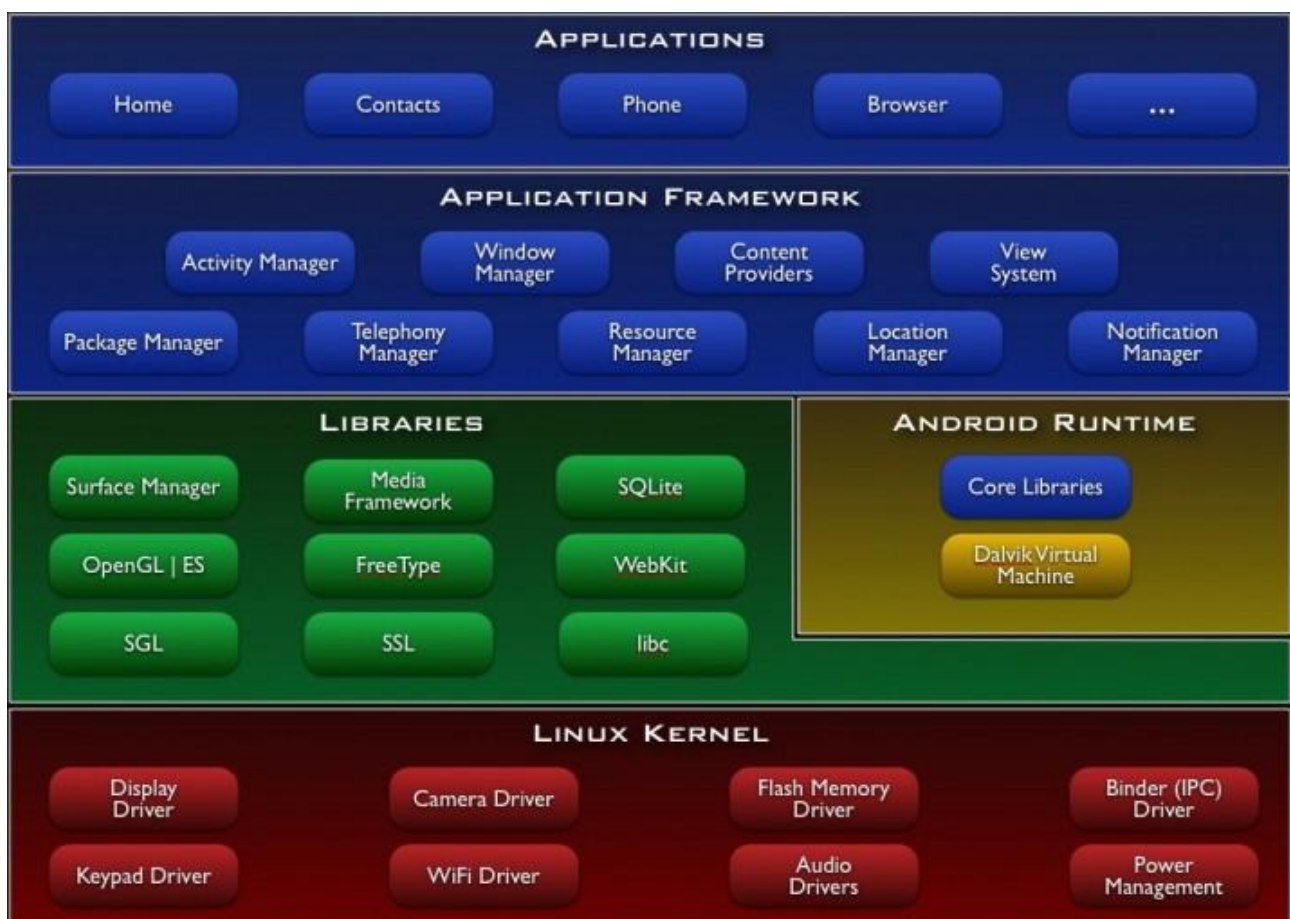
Es tracta d'un llenguatge creat per Google i la Open Handset Alliance (OHA), que engloba fabricants de hardware, operadors i empreses de software, amb companyies tan importants com LG, Samsung, Intel i Telefónica entre d'altres. El principal objectiu de la seva creació es el desenvolupament d'estàndards oberts per la telefonia com a mida per incentivar el desenvolupament i la millora de les expectatives dels usuaris.

Amb Android es buscar reunir en una mateixa plataforma tots els elements necessaris per permetre al desenvolupador aprofitar al màxim totes les funcionalitats que ens proporciona un *smartphone*, amb aquesta estratègia Google busca millorar i estandarditzar el desenvolupament d'aplicacions per a qualsevol dispositiu mòbil.

El cicle de vida d'aquestes aplicacions no es basa en flux en que l'aplicació s'executa i acaba, sinó que Android permet que aquesta s'integri dins l'entorn general. Un entorn que es basa en eines i coneixements del llenguatge Java, el que redueix considerablement la seva corba d'aprenentatge i afavoreix al seu desenvolupament amb la facilitat i la seguretat d'un llenguatge gestionat.

3.1 Arquitectura

En primer lloc, veiem un diagrama del sistema d'arquitectura d'Android recuperat del lloc web: <http://developer.android.com/guide/basics/what-is-android.html>



Esquema de l'arquitectura

La capa més immediata correspon al nucli d'Android, que utilitza el nucli de Linux com a capa d'abstracció pel hardware disponible en els dispositius mòbils. Aquesta capa conté

els drivers necessaris per a que qualsevol component de hardware pugui ser utilitzat mitjançant les crides oportunes. Quan un fabricant inclogui un nou element de hardware, el primer que ha de realitzar per a que pugui ser utilitzat des de Android es crear les llibreries de control o drivers necessaris dins d'aquest kernel de Linux.

El motiu pel qual s'ha escollit Linux es basa en dos punts principals :

Per una banda, la seva naturalesa de codi obert i lliure s'ajusta al tipus de distribució que google buscava per Android.

El segon motiu atent a que aquest kernel de Linux inclou nombrosos drivers, a més contempla la gestió de memòria, gestió de processos, mòduls de seguretat, comunicació en xarxa i altres funcionalitats pròpies d'un sistema operatiu.

La següent capa d'aquesta arquitectura es correspon amb les llibreries utilitzades per Android. Aquestes llibreries han estat implementades en C/C++, ja que és un llenguatge més propers als sistemes operatius i proporcionen a Android la major part de les seves característiques i funcionalitats.

Al mateix nivell que la llibreria d'Android es situa l'entorn d'execució que esta constituït per les *Core Libraries*, que son llibreries amb multitud de classes Java i la màquina virtual Dalvik.

Els dos últims nivells de l'arquitectura estan escrits íntegrament en Java. El framework d'aplicacions representa fonamentalment el conjunt d'eines de desenvolupament de qualsevol aplicació. Totes les aplicacions que es desenvolupen per Android utilitzen el mateix conjunt d'API i el mateix framework representat en aquest nivell. De forma general podem destacar algunes de les API mes importants ubicades aquí : *Activity Manager, Window Manager, Telephone Manager, Content Providers, View System, Location Manager, Notification Manager...etc)*

A l'últim nivell representat en aquest esquema trobem les aplicacions, aquí s'engloben tant les incloses per defecte en Android com aquelles que l'usuari va afegint posteriorment. Totes aquestes aplicacions utilitzen els serveis, les API i les llibreries dels nivells inferiors.

3.2 Components d'una Aplicació

Totes les aplicacions d'Android es poden descomposar en quatre tipus de blocs o components principals. Aquestes aplicacions seran una combinació d'un o més d'aquests components, que hauran de ser declarats d'una manera explícita en un fitxer amb format XML anomenat "AndroidManifest.xml", juntament amb altres dades associades com valors globals, classes que implementa, permisos, etc. Aquest fitxer és bàsic en qualsevol aplicació Android i permet al sistema executar correctament l'aplicació.

Aquest blocs que componen l'aplicació són :

3.2.1 Activity

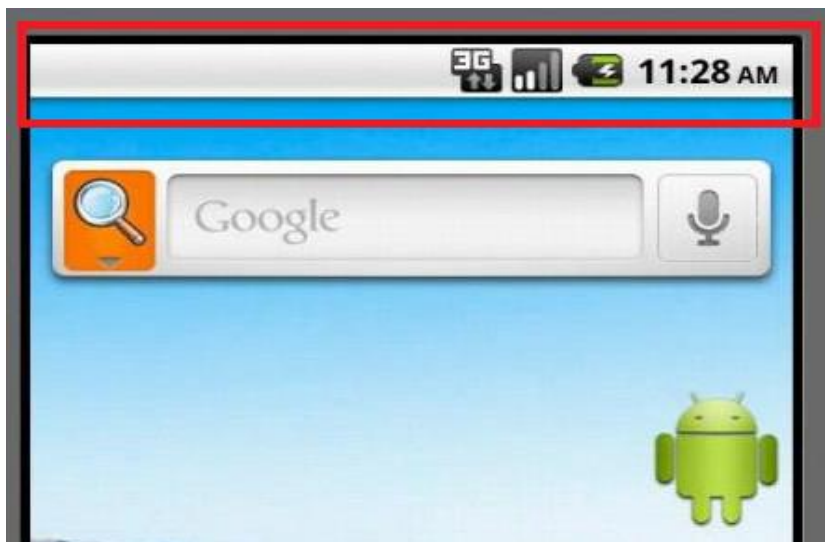
Es tracta del component més habitual en les aplicacions. Un component Activity reflexa una determinada activitat portada a terme per una aplicació i que porta associada una finestra o interface d'usuari. És important senyalar que no només es tracta d'una aparença gràfica, sinó que aquest component gràfic forma part del component Activity a través de vistes representades per classes com View i les seves derivades.

La majoria d'aplicacions permeten la execució de diverses accions mitjançant l'existència d'una o més pantalles, cadascuna d'aquestes pantalles hauria d'estar representada a través d'un component Activity, de tal manera que realitzar una navegació d'una pantalla a una altra implica llançar una activitat o posar-ne una altra en *background*, cosa que se'n permet fer gràcies a que Android permet controlar per complet el cicle de vida d'aquests components Activity.

Molt vinculat a aquest component es troben els *Intents*. Un intent consisteix bàsicament a la voluntat de realitzar una determinada acció, generalment amb dades associades, de tal manera que una aplicació pot delegar el treball a una altra.

3.2.2 Broadcast Intent Receiver

Un component *Broadcast Intent Receiver* s'utilitza per llançar alguna execució dins de l'aplicació actual quan un determinat esdeveniment es produeixi, com per exemple, rebre una trucada entrant. Pot utilitzar l'API Notification Manager, anomenada anteriorment, per avisar a l'usuari de l'esdeveniment produït mitjançant la barra d'estat del dispositiu mòbil.



Barra d'estat d'un dispositiu Android

Aquest component s'implementarà a través d'una classe anomenada `BroadcastReceiver`. És important destacar que perquè el `Broadcast Intent Receiver` funcioni la aplicació en qüestió ha de ser l'aplicació activa en el moment de produir-se l'esdeveniment.

3.2.3 Service

Un component *Service* és una aplicació sense interface d'usuari, i que generalment s'executa en segon pla mentre d'altres estan actives. Està implementat per la classe del

mateix nom *Service*.

Es el cas per exemple d'un reproductor de música, en el moment en que un usuari reproduïx un cançó, aquesta s'executa mentre l'usuari continua navegant pels menús de l'aplicació de reproducció, inclús es possible que l'aplicació que es troba en primer pla sigui una de diferent i no tingui res a veure amb el reproductor de música.

La interface d'usuari del reproductor jugaria el paper de component *Activity* però la reproducció de la música seria un component *Service* que s'executa en segon pla.

3.2.4 Content Provider

Amb aquest component, qualsevol aplicació d'Android pot emmagatzemar dades en un fitxer, en una base de dades SQLite o en qualsevol altre format.

Aquestes dades poden ser compartides entre diverses aplicacions. Qualsevol classe que implementi el component *Content Provider* haurà de tenir una sèrie de mètodes que permetin emmagatzemar, recuperar, actualitzar i compartir les dades d'aquella aplicació.

3.3 Cicle de vida de les aplicacions android

La gestió de les aplicacions en Android es fa d'una manera particular i diferent a d'altres sistemes operatius per smartphones. Cada aplicació s'executa en el seu propi procés, cosa que aporta grans beneficis com la seguretat, la gestió de la memòria o la ocupació de la CPU. Android s'ocupa de llançar i parar tots els processos que s'executen, els gestiona i s'encarrega de decidir que fer en funció dels recursos de que disposa i de les accions que rep per part de l'usuari.

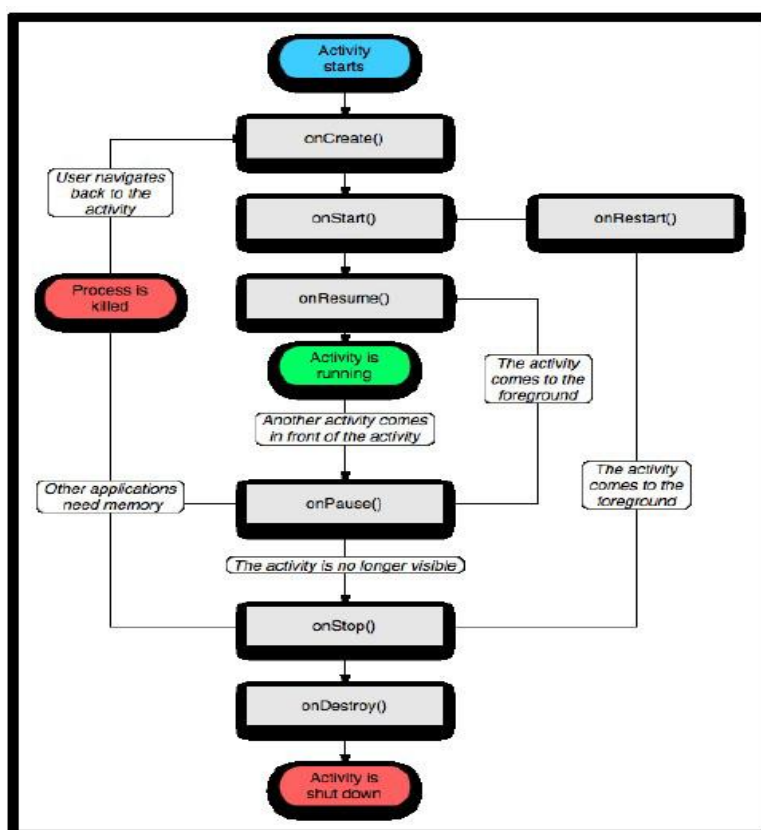
Aquest comportament es absolutament inherent per l'usuari i aquest tan sols s'ha de preocupar de fer clic per executar una aplicació i poder tornar a l'anterior en qualsevol moment amb una agilitat digne dels ordinadors personals més potents. En cap moment es preocupa quina es l'aplicació activa, quanta memòria està consumint i quants recursos li

queden per utilitzar, tot això ho gestiona el sistema operatiu.

Android llençarà tants processos com siguin possibles amb els recursos de que disposa, cada un d'aquests processos està format per una o varies Activitats que conformen l'aplicació. Quan l'usuari navega d'una activitat a l'altre o fa l'acció d'obrir una nova aplicació, el sistema dorm aquests processos i realitza una còpia del seu estat per a poder recuperar-la mes tard.

Cada component bàsic d'Android te un cicle de vida ben definit, d'aquesta manera un desenvolupador pot controlar en cada moment en quin punt es troba la seva aplicació i realitzar les accions que siguin mes oportunes en cada instant.

Veiem per exemple el cicle de vida del component Activity :



Cicle de vida d'un objecte Activity

Si analitzem aquest diagrama podem extreure les següents conclusions sobre el cicle de

vida :

- Els mètodes **onCreate()**, **onDestroy()** engloben tot el cicle de vida representant el principi i el final de l'activitat.
- Els mètodes **onStart()**, **onStop()** representen la part visible del cicle de vida, en el punt en que comença (onStart()) l'activitat es farà visible per l'usuari, encara que no tingui el focus en aquell moment, fins a onStop() on l'activitat deixarà de ser visible. Son mètodes que poden ser cridats múltiples vegades.
- Els mètodes **onResume()**, **onPause()** representen la part útil del cicle de vida de l'aplicació, no només l'aplicació esta activa durant el temps englobat entre aquests dos mètodes si no que a més te el focus i l'usuari pot interactuar amb elles.

3.4 Eliminació de processos

Una vegada explicat que les aplicacions Android s'executen en el seu propi procés i que aquest procés continuarà existint fins que l'aplicació finalitzi o el sistema necessiti utilitzar part dels recursos d'aquell procés per un altre que considera més important.

Per aquest motiu es important que un desenvolupador conegui que els components d'una aplicació (*Activity*, *Broadcast Intent Receiver*, *Service* y *Content Provider*) i que l'ús incorrecte dels mateixos pot comportar que l'aplicació sigui destruïda en el moment en que realitza una acció important per l'usuari.

D'aquesta manera Android estableix una jerarquia de processos on evalua els components que estan executant i l'estat en que es troben. L'ordre d'importància es el següent :

1. Processos en primer pla : són aquells processos necessaris per l'acció que l'usuari esta duent a terme en aquell moment. Entraran en aquesta categoria si almenys compleix un dels següents requisits :
 - Te un component Activity executant-se, amb el que l'usuari està interactuant

- Te un component Broadcast Intent Receiver executant-se
- Ha llençat algun altre procés que te un component Service executant-se

2. Processos visibles : son aquells que contenen un component Activity visible en pantalla però no tenen el focus de l'activitat en aquell moment

3. Processos de servei : es tracta d'aquells processos que tenen un component Service i s'està executant en segon pla, com em dit abans, encara que no son visibles per l'usuari estan fent una acció que si es pot percebre.

4. Processos en segon pla : son aquells amb un component Activity que no son visibles per l'usuari. No tenen una importància directa per l'usuari en aquell moment

5. Processos buits : processos que no executen cap activitat per que Android els manté en memòria per agilitzar una nova crida per part de l'usuari.

En base a aquesta jerarquia descrita Android decideix quins processos han de ser eliminats per alliberar recursos del sistema i fer que s'executin nous processos, de tal manera que un procés buit serà el primer en ser eliminat i un procés que s'executi en primer pla serà l'últim.

3.5 API Demos

Juntament amb el SDK d'Android, Google ha inclòs nombroses classes d'exemples dins d'un bloc anomenat "APIDemos". Aquestes classes exemplifiquen de forma pràctica moltes de les característiques i possibilitats que s'ofereixen amb Android. D'aquesta manera faciliten les coses al desenvolupador novell i es molt útil per fer-se una idea general del tipus d'aplicacions que el sistema pot suportar. Podem trobar tota la informació a : <http://developer.android.com/resources/samples/ApiDemos/index.html>

Alguns exemples son :

- APIDemos\app : inclou exemples de com treballar amb els principals components, com per exemple, Activity o Services amb algunes aplicacions senzilles com a exemple.
- APIDemos\View : aquí es poden trobar exemples de diferents tipus de vistes i dissenys per les interfícies d'usuari. Podem veure exemples de creació de formularis, distribució d'elements, scrolls, etc...
- APIDemos\Media : exemples de com utilitzar formats multimèdia (vídeo, audio o imatges)
- APIDemos\Graphics : exemples de la capacitat gràfica que ofereix Android a l'hora de crear figures geomètriques bàsiques o definicions d'imatges amb moviment.

3.6 Contingut d'un projecte Android

Com ja s'ha comentat anteriorment, per desenvolupar aplicacions amb Android es fa necessari dues coses : tenir instal·lat l'sdk i l'Eclipse. L'entorn de desenvolupament per les nostres aplicacions serà Eclipse . Al crear un nou projecte Android en eclipse es crea una carpeta amb el nom del projecte i un seguit de subcarpetes que conformen l'anatomia del projecte.

Anem a centrar-nos amb un projecte hipotètic anomenat “HolaMundo” que agafarem com a exemple per mostrar aquesta estructura de carpetes.



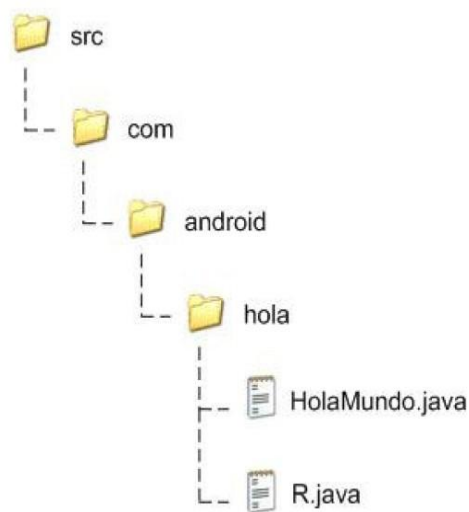
Jerarquia en el workspace d'Eclipse

Las carpetas i fitxers de “HolaMundo” han estat creades pel plug-in ADT i les seves característiques dependrà del tipus de projecte que s’estigui construint, en aquest cas un projecte Android.

Els principals elements que podem trobar son els que passem a descriure a continuació

3.6.1 Carpeta \src

La carpeta src conté els arxius .java del nostre projecte, es aquí on es situen totes les classes java encarregades de la lògica del nostre programa. L’estructura que segueix depèn del *Package* que li assignem al nostre projecte, d’aquesta manera en el nostre exemple em decidit “com.android.hola” i la jerarquia de carpetes quedaria com en la següent figura :



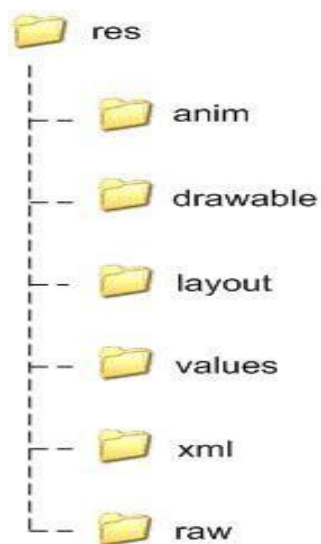
Estructura de la carpeta \src

3.6.2 Carpeta \res

A la carpeta \res podem trobar tots els recursos utilitzats pel projecte, es a dir, qualsevol fitxer extern al projecte que contingui dades referents al projecte i que ha de ser compilat

juntament amb el codi font.

Aquí podem incloure diferents tipus de recursos com XML, JPEG o PNG entre d'altres. Cadascun d'aquest recursos ha d'estar ubicat dins de la seva carpeta corresponent en funció de quina sigui la seva naturalesa, per això la carpeta \res té la següent estructura



Estructura de la carpeta \res

Cada subcarpeta alberga diferents recursos, de tal manera que :

- \anim : conté fitxers XML que descriuen una animació per determinats elements.
- \drawable : conté recursos que poden ser dibuixats en pantalla com per exemple JPG, PNG o GIF
- \layout : conté dissenys que poden ser utilitzats per construir interfícies, aquets dissenys estan descrits en fitxers XML
- \values : aquesta carpeta tindrà fitxers XML on es declaren valors de diferent tipus que poden ser accessibles per l'aplicació. En aquest cas per a cada tipus de recurs existirà un fitxer que faci referència a ell com per exemple : "strings.xml", "colors.xml" o "integers.xml"
- \xml : en aquesta carpeta es troben tots els fitxers xml genèrics que poden ser parsejats desde el codi java

- `\raw` : en aquesta carpeta s'inclouen tots aquells recursos que son afegits a la compilació directament sense processar, de tal manera que l'aplicació podrà llegir-los com a fluxos de bytes. Un exemple típic d'aquesta carpeta son els arxius d'àudio

Tots els recursos que es declarin aquí quedaran reflexats un fitxer anomenat “`R.java`”. L'usuari no s'ha de preocupar per editar aquest fitxer ja que es genera i sincronitza de forma automàtica en temps de compilació segons variï el contingut de `\res`. Mitjançant aquest fitxer, tots els recursos es poden utilitzar i referenciar directament en el codi de la aplicació com si fossin variables globals, la forma més habitual de fer això es mitjançant el mètode `getResources()` de la classe `Context`.

3.6.3 Carpeta `\bin`

Com en d'altres entorns de desenvolupament, la carpeta `\bin` conté els arxius binaris del projecte generats a partir dels arxius font. De la mateixa manera que `\src` es manté la mateixa jerarquia de subcarpetes que les representades en el nom del paquet.

3.6.4 Arxiu `AndroidManifest.xml`

Es tracta d'un element imprescindible i fonamental en totes les aplicacions Android. Aquest arxiu es genera de forma automàtica pel plug-in d'eclipse i descriu d'una forma genèrica cadascun dels components que formen l'aplicació, com per exemple, `Activity` o `Content Provider`. També conté permisos i polítiques de seguretat que afecten a tota l'aplicació.

Veiem, a mode d'exemple, l'arxiu `Manifest` que es genera en l'exemple de “`HolaMundo`” :

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.hola">

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

        <activity android:name=".HolaMundo" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>

    </application>

</manifest>

```

Manifest d'exemple "HolaMundo"

Veiem com en un primer moment es declara el document com un XML vàlid i s'adjunta un espai de noms adequat en l'element <manifest>, i el el nom del paquet al qual pertany amb l'atribut Package.

En la següent línia existeix un element <application> on es declaren els components que formen l'aplicació complerta. Únicament hi haurà un element application per Manifest. L'atribut android:icon en aquest apartat ens indica que existeix un recurs del tipus "drawable", que es troba a la carpeta HolaMundo\res\drawable, i que es diu "icon", aquest "icon" serà la icona representativa de la nostra aplicació. L'altre atribut que se'ns mostra en aquest apartat, "app_name" segueix la mateixa lògica que la descrita amb l'atribut anterior per aquesta vegada ens indica que el que busquem es un string i que farà referència a l'etiqueta de text visible per l'usuari.

Dins de l'element <application> es declara un únic component que constitueix l'aplicació, un component Activity. L'element <activity> declara un únic atribut android :

name que designa el nom de la classe que implementa aquesta activitat, també conté android:label que dona un altre vegada nom visible.

Per últim, aquesta Activity declarada porta associada un Intent Filter, que descriu com i quan s'ha de llençar aquesta activitat. Amb el Tag <action> informa que la activitat ha de ser considerada com la classe principal de l'aplicació i amb el tag <category> indica que la activitat s'ha d'executar quan l'usuari llença l'aplicació.

Un exemple d'altres elements importants que es poden trobar en l'arxiu AndroidManifest.xml i que no queden reflectits en aquest exemple son :

- <permission> : defineix permisos utilitzats per restringir als components de la aplicació l'accés a certs serveis d'Android.
- <intent-filter> : encarregat de descriure quan i on es pot executar la Activity de la que penja, permet la inclusió de diferents atributs per a concretar el seu funcionament
- <action> : acció que es suportada per la Activity
- <category> : categoria de la Activity
- <data> : dades acceptades per la activity
- <receiver> : element que representa un component Broadcast intent Receiver, encarregat de llençar una acció de resposta a un esdeveniment.
- <service> : element que representa un component Service, encarregat d'executar una acció en segon pla.
- <provider> : element que representa un Content Provider, utilitzat en la aplicació per emmagatzemar i compartir dades.

3.7 Interfícies d'usuari en Android

Les aplicacions Android més complexes poden arribar a tenir milers de línies de codi dedicades només a l'apartat gràfic. Aquest codi acaba generalment vinculat a la resta de codi que no té res a veure amb la interfície gràfica, de tal manera que un sol canvi en l'aspecte d'una aplicació pot suposar moltes hores de treball i realitzar codi nou completament.

Amb aquesta premissa, Android pretén independitzar d'una manera senzilla la forma de realitzar el contingut gràfic de tal forma que afecti el menys possible a la part de codi de la aplicació.

La solució per la que s'ha optat ha estat en utilitzar fitxers escrits amb XML que representa l'aparença gràfica i tots els aspectes relacionats de l'aplicació. Aquest fitxer estarà vinculat al projecte simplement pel fet de ubicar-ho en la carpeta "res/layout" i fent una referència a ell al codi, veiem un exemple d'una interfície gràfica escrita amb XML :

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="Hello, Android"/>
```

Exemple d'una interfície amb XML

Veiem que en aquest cas, el que es fa en primer lloc, d'igual manera que en el cas del Manifest, es declarar el document XML com un document vàlid, a continuació s'indica l'únic element que contindrà la interfície, en aquest cas un TextView amb la definició dels seus atributs.

Si analitzem aquest atributs d'una forma senzilla :

- `xmlns:android` : declara l'espai de noms Android vàlids
- `android:layout_width` : indica quina quantitat d'amplada en la pantalla ha d'ocupar, "fill-parent" indica que s'ocupi tota la pantalla.
- `android:layout_height` : indica quina quantitat d'altura en la pantalla s'ha d'ocupar, el valor es el mateix que en l'apartat anterior
- `android:text` : indica el text que es mostra en aquest element View

4. Where's my Money? : Aplicació d'estalvi

L'aplicació que presento a continuació busca dos objectius, la part fonamental es la del desenvolupament d'una aplicació pròpiament dita, una aplicació basada en la senzillesa del seu ús que pugui facilitar l'acció de control de la despesa a un usuari seguint uns patrons definits anteriorment. Per una altre banda, amb el desenvolupament de l'aplicació es busca plasmar alguns dels conceptes més fonamentals d'Android, com les "activities" o els "intents", descrits a l'apartat anterior. De tal manera que aquells lectors menys experimentats en la programació per dispositius mòbils puguin veure a mode d'exemple el codi d'una aplicació d'aquest tipus.

4.1 Anàlisis i disseny de la aplicació

Mitjançant les conclusions extretes del test d'usabilitat realitzat i descrit en el punt 2,2 d'aquest mateix document, es van establir unes bases abans de començar la implementació i el disseny i s'ha intentat seguir aquesta línia durant tot el seu desenvolupament.

Per una banda s'ha donat molta importància a la facilitat d'ús de la aplicació, de tal manera que aquells d'usuaris mes inexperts tinguin facilitats per utilitzar-la en un breu període de temps ja que el disseny ha intentat disminuir la corba d'aprenentatge.

D'altre manera, s'ha intentat dotar a la aplicació de unes funcionalitats prou potents com per que l'usuari pugui du a terme totes aquelles accions necessàries pel control de la seva despesa, eliminant accions innecessàries o poc pràctiques que en molts casos fan que l'usuari es confongui.

Moltes funcionalitats pensades prèviament han estat excloses de la implementació final donada la manca de temps, per això en l'apartat possibles ampliacions de la present memòria s'acabarà de comentar quines son totes les funcionalitats que hauria de tenir l'aplicació.

4.1.1 Casos d'ús

En aquest apartat s'utilitzen els casos d'us per apropar al lector a les funcionalitats de l'aplicació "Where is my money?", però no es pretén aprofundir massa en les possibilitats que ofereix aquesta tècnica, si no que es limitarà només en aquells aspectes que son mes il·lustratius i concisos per al lector i que ajudin a comprendre millor que es el que realitza l'aplicació.

A mode de recordatori direm que en enginyeria del software un cas d'ús representa un us típic que se li dona al sistema. La tècnica de casos d'ús permet capturar i definir els

requisits que deu complir una aplicació, i descriu les interaccions entre l'usuari i aquesta. Aquesta tècnica es utilitza en molts casos pels enginyers de software per, entre d'altres coses, ensenyar al client de forma clara i descriptiva les accions que podrà realitzar el futur sistema.

A continuació es mostra un diagrama amb els casos d'ús associats a "Where is my money?"

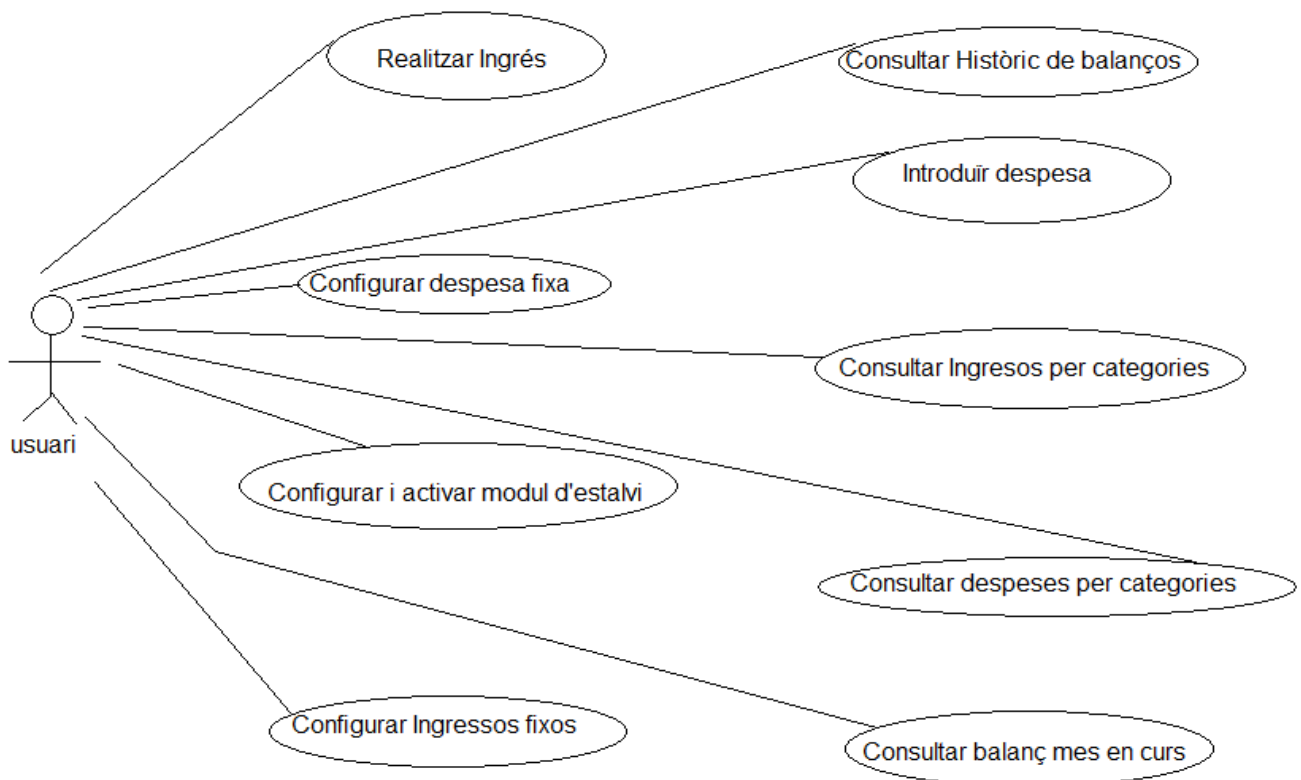


Diagrama de casos d'ús de "Where is my money?"

Cada cas d'us representa un cercle amb un nom descriptiu de l'acció en el seu interior, fora del sistema es troba l'usuari (actor) que pot interactuar amb ell. A continuació s'ofereix una breu descripció de cadascun dels casos mostrats en el diagrama :

- **Realitzar ingrés** : permet realitzar la introducció d'una quantitat de saldo positiu en una data determinada corresponent a una categoria concreta

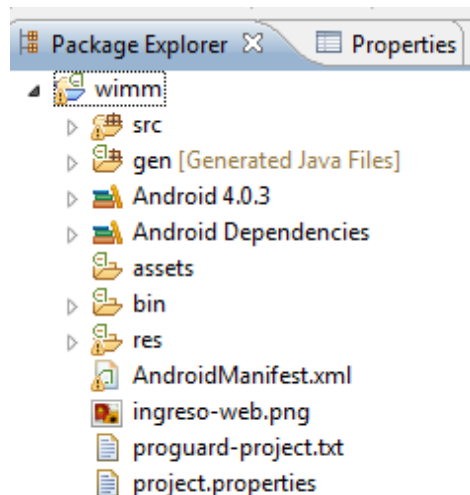
- **Introduir despesa** : permet realitzar la introducció d'una quantitat de saldo negatiu en una data determinada corresponent a una categoria concreta
- **Consultar històric de balanços** : permet la consultació del balanç d'un mes determinat diferent a l'actual, també es pot consultar l'històric per categories
- **Configurar despesa fixa** : permet realitzar la configuració de la introducció d'una quantitat de saldo negatiu corresponent a una categoria determinada en una data concreta de forma automatitzada per part de l'aplicació
- **Configurar ingress fix** : permet realitzar la configuració de la introducció d'una quantitat de saldo positiu corresponent a una categoria determinada en una data concreta de forma automatitzada per part de l'aplicació
- **Consultar ingressos per categories** : consulta les quantitats de saldo positiu introduïdes en el mes en curs per categories
- **Consultar despeses per categories** : consulta les quantitats de saldo negatiu introduïdes en el mes en curs per categories
- **Consultar balanç mes en curs** : consulta el balanç de quantitats positives – negatives del mes en curs.
- **Configurar i activar mòdul d'estalvi** : introdueix una tant per cent d'estalvi en relació als ingressos produïts aquell mes i activa el mòdul d'estalvi de tal manera que cada vegada que es consulti el balanç aparegui un missatge informatiu en cas de que es superi la quantitat d'estalvi escollida

4.2 Desenvolupament e implementació

Una vegada exposats i descrits les característiques principals de l'aplicació, així com les decisions preeses pel seu disseny, a continuació es desglossen els aspectes més importants relacionats amb la seva implementació.

4.2.1 Package del projecte

A continuació es mostra desglossat i explicat com s'ha distribuït el projecte durant la seva implementació en relació a les carpetes que el formen

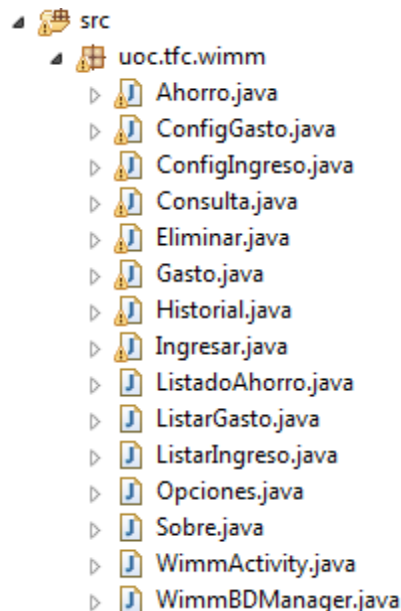


Distribució de carpetes

En la figura anterior podem veure el conjunt de carpetes que formen l'aplicació, cadascuna d'elles conté elements rellevants per la seva implementació, donem especial importància a la carpeta `src\`, que conté les classes java, i a la carpeta `res\` que conté els recursos de l'aplicació. També podem destacar per la seva importància en el funcionament de l'aplicació l'arxiu `AndroidManifest.xml` on es descriu el comportament global de “Where is my money?”

4.2.2 Característiques Principals

Carpeta SRC\ :



Conjunt de classes java que formen la carpeta src

En aquesta carpeta trobem el cor de la implementació, aquí tenim totes les classes java que contenen el conjunt de les funcionalitats de l'aplicació.

Cadascuna d'elles extenen de la classe pare Activity sobreescrivint el seu constructor e implementen la interfície “On click listener” de tal manera que han de codificar els mètodes d'aquesta per obtenir les propietats necessàries per cliackar en els elements mostrats en pantalla i realitzar les accions pertinents.

Veiem per exemple la capçalera de classe inicial de l'aplicació :

```
public class WimmActivity extends Activity implements OnClickListener
```

Menció apart mereix la classe WimmBDManager que no segueix el patró esmentat anteriorment ja que en cap moment es presenta per pantalla pel que no es necessari, la funcionalitat d'aquesta classe resideix en la creació de les bases de dades necessàries per l'acumulació de les dades durant el procés normal de funcionament, d'aquesta manera veiem que la capçalera de la classe és :

```
public class WimmBDManager extends SQLiteOpenHelper
```

AndroidManifest.xml ens marca quina serà la classe principal a l'hora d'executar l'aplicació

```
<activity
    android:name=".WimmActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

En aquest cas veiem que es tracta de WimmActivity, que serà l'encarregada, entre d'altres coses de mostrar el menú principal i portar-nos a d'altres classes mitjançant un element fonamental en la programació d'android com son els intents, veiem un exemple :

```
Intent i_ing = new Intent (this, Ingresar.class);
startActivity(i_ing);
```

En aquest cas, les línies de codi pertanyen a una estructura switch que discrimina en el moment de realitzar un click sobre quin element s'ha realitzat i en conseqüència realitza l'acció pertinent, en aquest cas iniciar la classe Ingresar.class. La mateixa estructura de selecciona d'elements en pantalla està present en la resta de classes.

Cada nom de classe s'ha intentat posar de tal manera que descrigui amb prou transparència les accions que s'esperen d'aquella classe, cadascuna d'elles inclouen en el seu codi la lògica de l'aplicació per les funcionalitats que han estat pensades, així per exemple, la classe Ingresar.java, a part de carregar xml corresponent a la interfície gràfica conté els mètodes necessaris per obtenir els elements introduïts per pantalla per part de l'usuari i realitzar la creació e insercions a les bases de dades mitjançant una API d'android que permet fer-les de forma parametritzada i minimitzar els errors a l'hora de realitzar queries. Per exemple el mètode InsertarTablaIngreso que s'executa després que l'usuari hagi introduït les dades i hagi premut el botó Acceptar :

```
private boolean insertarTablaIngreso(double cant, String con, int fecha) {
    boolean rc = false;
    WimmBDManager wimmBd =new WimmBDManager(this, "BDwimm", null, 1);
    SQLiteDatabase db = wimmBd.getWritableDatabase();
    if(db != null){
```

```

        ContentValues nuevoRegistro = new ContentValues();
        nuevoRegistro.put("fecha", fecha);
        nuevoRegistro.put("cantidad", cant);
        nuevoRegistro.put("concepto", con);
        db.insert("Ingresos", null, nuevoRegistro);
        db.close();
        rc=true;
    }
    return rc;
}

```

Un altre aspecte interessant a destacar de la implementació de les classes es la inclusió d'un menú contextual que s'executarà a l'hora de prem el botó físic del dispositiu mòbil “menú”. Aquest menú s'ha inclòs en totes les classes que es presentin gràficament a l'usuari, en molts casos contenen varies opcions relatives a funcionalitats de l'aplicació, però el denominar comú en totes elles es la inclusió de la opció sortir que farà que immediatament l'aplicació acabi, veiem, a mode d'exemple, el menú que implementa la classe principal :

```

public boolean onCreateOptionsMenu (Menu menu){
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
}

public boolean onOptionsItemSelected(int featureId, MenuItem item){
    super.onOptionsItemSelected(featureId, item);
    switch(item.getItemId()){

        case R.id.sobre:
            startActivity(new Intent(this, Sobre.class));
            break;
        case R.id.salir:
            Historial h = new Historial();
            h.setEsHistoric(false);
            finish();
            break;
        default:
            break;
    }
    return true;
}

```

Un altre aspecte a destacar de les classes que formen el projecte es la inclusió, en les que la lògica d'aplicació ho requereix, de mètodes capaços de mostrar diàlegs

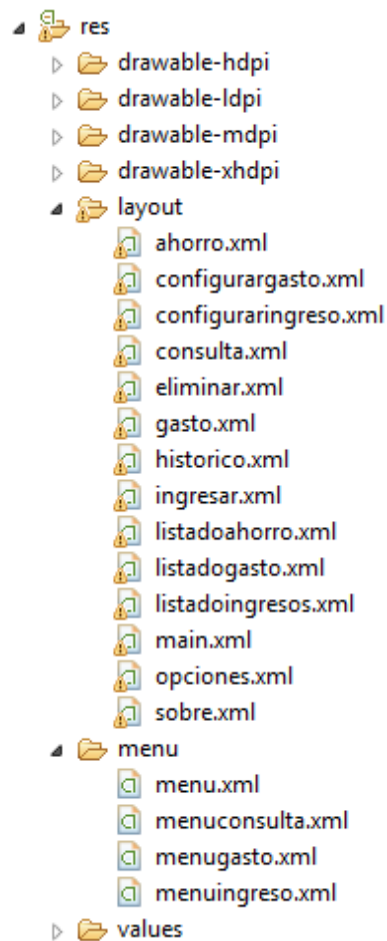
d'informació a l'usuari, com per exemple la confirmació o errors en les accions que es duen a terme o el control d'errors, veiem una estructura d'aquest tipus :

```
private void openDialogOK() {
newAlertDialog.Builder(this).setTitle(R.string.txt_FijOk).setItems(R.array.insercionOK,
new DialogInterface.OnClickListener(){
    public void onClick(DialogInterface dialoginterface, int which)
    {
        dialoginterface.cancel();
    }
})
.show();
}
```

La relació entre instàncies de les classes es constant durant tota l'execució de la aplicació, així que s'ha decidit que la majoria de mètodes siguin mètodes amb modificador d'accessos públics ja que d'aquesta manera son transparents entre elles i permeten, entre d'altres coses accedir a variables d'una classe que modifiquen el comportament d'altres, com per exemple el cas del “flag” de control de l'activació de la classe d'estalvi, que necessita ser visible per la classe que mostrarà el diàleg en cas de que estigui activada i compleixi uns determinats requisits :

```
public boolean getEsAhorroActivado(){
    return esAhorroActivado;
}
```


Carpeta RES\ :



Conjunt d'arxius xml que formen els recursos de la aplicació

En aquesta carpeta i les seves subcarpetes trobem tots els recursos necessaris que mostrarà l'aplicació. En les carpetes res\drawable.... tenim totes les imatges que conté l'aplicació distribuïdes segons la seva resolució, totes les imatges i els recursos es van referenciant durant l'aplicació mitjançant la classe R que fa apuntadors a tots els recursos que anem introduint d'una forma automàtica. Així per exemple, si durant l'execució del codi volem fer referència a algun recurs concret, des de les classes .java farem :

R.id.*btn_historial*

I desde un xml farem :

android:src="@drawable/ing"

A les carpetes `res\layout` trobem les implementacions gràfiques de la part visual, es habitual l'us del conegut com a layouts per col·locar els elements en pantalla i que d'aquesta manera mantinguin la relació d'aspectes entre ells, ja que en un altre cas i a diferència d'altres formes d'implementar per dispositius mòbils, Android s'executa sobre molts tipus diferents de smartphones de diferents fabricants amb característiques diferents entre ells, i son precisament aquestes característiques en resolució de pantalla las que poden provocar que l'aspecte visual varii respecte a una implementació inicial.

Aquestes implementacions en arxius `.xml` defineixen cadascuna de les pantalles de l'aplicació i son cridades des de les classes java mitjançant una sobreescrita del mètode constructor d'Activity de la forma :

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.sobre);  
}
```

Ja s'ha comentat anteriorment, pero tornem a destacar l'accés als recursos android mitjançant la classe R : `R.layout.sobre`

`sobre.xml` defineix la pantalla d'informació inicial de l'aplicació, accessible des de el menú contextual de la classe principal i te una estructura del tipus :

```
<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:padding="10sp">  
  
    <LinearLayout  
        android:id="@+id/sobre"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:text="@string/about_text">  
  
        <TextView  
            android:id="@+id/textView1"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:text="@string/about_text" />  
  
    </LinearLayout>  
  
</ScrollView>
```

Menció especial mereixen alguns dels xml que es presenten a l'usuari no de la forma habitual com una pantalla si no com un diàleg. Això es defineix en el moment de mostrar l'Activity i es fa en l'arxiu AndroidManifest.xml, per exemple veiem aquesta pantalla sobre com es presentaria en el moment en que es invocada per la classe Sobre.java

```
<activity android:name=".Sobre"
    android:label="@string/lbl_sobre"
    android:theme="@android:style/Theme.Dialog">
</activity>
```

En l'arxiu manifest i mitjançant la línia `android:theme="@android:style/Theme.Dialog` definim que l'estil de la presentació serà com un diàleg.

4.2.3 Idiomes

Avui en dia tenir una aplicació en un sol idioma no n'hi ha prou. Cal pensar que un usuari pot ser de qualsevol país i per tant, tenir l'aplicació en un sol idioma pot suposar una barrera.

Es podria fer l'aplicació en Anglès, que és el més estès i en el que solen estar la majoria de programes, però això suposaria un disseny poc universal, degut a que potser hi ha usuaris que no el saben i no es defensen prou bé amb ell.

Una aplicació, si està amb el màxim d'idiomes possibles, ofereix un mercat més ampli i una millor acceptació per part dels usuaris, els quals no acabarien dient el clàssic "No està en Català?".

Android ofereix un bon potencial per a que l'aplicació pugui ser multi idioma amb molta facilitat. Per tal de canviar l'idioma del terminal, tan sols cal anar a "*ajustes->Idioma y teclado*". Depenent de la versió d'*Android* i de la companyia telefònica on s'hagi adquirit el terminal o del fabricant, els idiomes que el terminal puguin contenir.

Per exemple, es pot trobar el terminal *Huawei Sonic*, el qual només conté els idiomes Anglès i castellà, tot i tenir la versió 2.3.3 de *Android*, la qual conté una llarga llista d'idiomes. En canvi, el LG-P500 amb la versió d'*Android* 2.2, conté l'Anglès, el Català, el Castellà, Euskera, Gallec, Francès, Alemany, Italià i Portuguès Així doncs, tot dependrà de quina compilació del sistema operatiu s'ha instal·lat en el terminal.

Per tal de poder traduir una aplicació de forma paral·lela, cal que totes les cadenes de text estiguin en el fitxer *strings.xml*. Aquest fitxer conté un llistat de tots els textos utilitzats per a l'aplicació, evitant així, posar les cadenes de text directament al codi. En cas de posar el

text directament al codi, es podrien tenir problemes per a canviar-lo degut a que s'hauria de revisar les línies de codi fins a trobar-lo i tenir molts problemes per a traduir el sistema de l'aplicació complet.

El fitxer *strings* conté una estructura “camp-valor” per cada cadena de text de la forma :

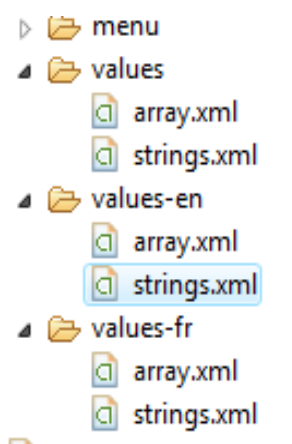
```
<string name="ingreso">Ingreso</string>
```

Aquests fitxers externs, es pot accedir amb facilitat des del codi amb els mètodes proporcionats per la classe *Resource*. O bé amb l'atribut text en el element corresponent del *Layout* de la *Activity*.

```
getResources().getString(R.string.NomDelString));  
android:text="@string/NomDelString"
```

D'aquesta manera, tindrem totes les cadenes de text en la carpeta */res/values/* i en un sol idioma. Un cop arribats a aquest punt, caldrà fer l'aplicació multi-idioma amb un senzill pas i tenint en compte els codis de representació dels noms dels idiomes.

Si es vol un nou idioma, tan sols cal agregar una nova carpeta */res/values/* afegint en el nom de la carpeta */res/values/*, el nom de la representació de l'idioma. Per exemple, si es vol el Francès, caldrà afegir la carpeta */res/values-fr/* i dins de la carpeta, cal que hi contingui els fitxers *strings* i *arrays* amb els valors de les cadenes traduïdes en el idioma corresponent.



Respecte al canvi d'idioma, aquest el fa automàticament el sistema *Android*. *Android* consulta quin és el idioma del terminal i seguidament mira en el sistema de carpetes de

l'aplicació si hi ha una carpeta `/res/values-idioma/` expressament per a l'idioma per tal d'anar a buscar el valor de les cadenes de text en ell. En cas de que no trobi l'idioma indicat, es farà servir el que hi hagi per defecte `/res/values/`.

5. Proves i resultats

Durant tot el projecte s'han anat fent múltiples proves per tal de poder testejar el correcte funcionament de l'aplicació. Per la realització de les proves s'ha utilitzat l'emulador que Android incorpora en el seu SDK (permet emular dispositius mòbils amb el SO d'Android incorporat sense necessitat d'utilitzar un dispositiu físic) i un terminal LG P970.

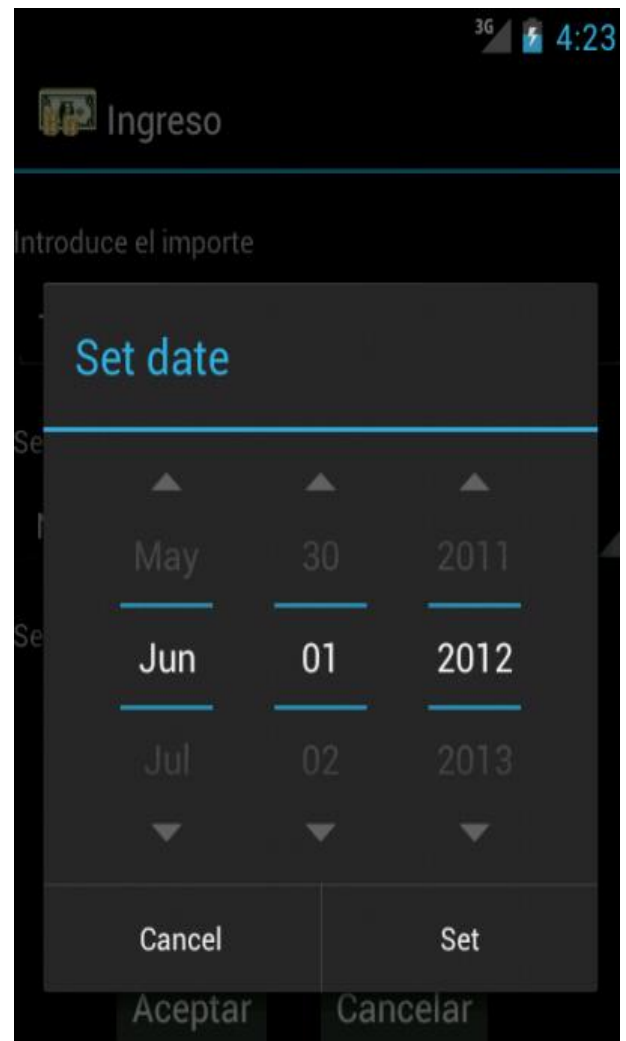
A continuació es detallen algunes de les proves més significatives realitzades i els resultats obtinguts.

5.1 Funcionalitat Ingrés

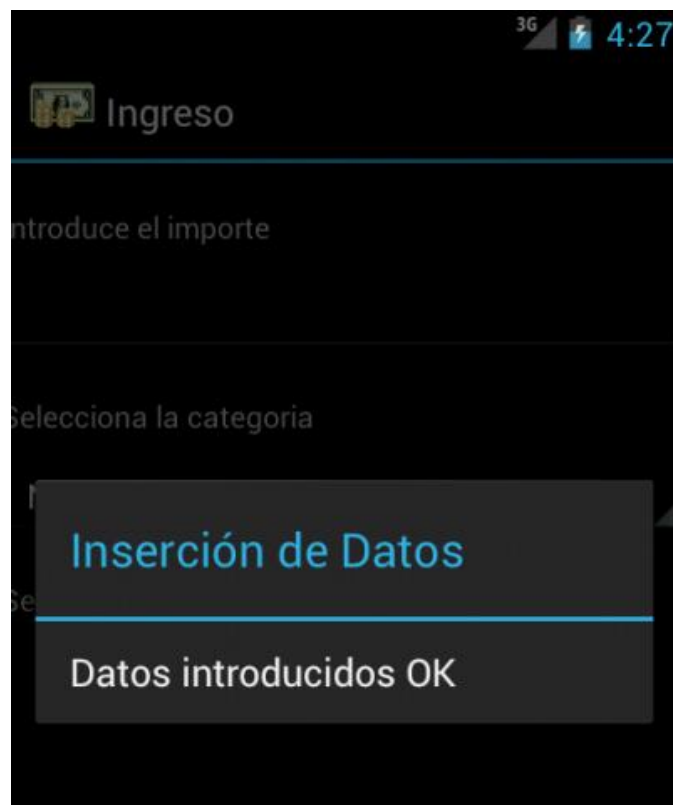
- En la pantalla inicial seleccionem realitzar un nou ingrés :



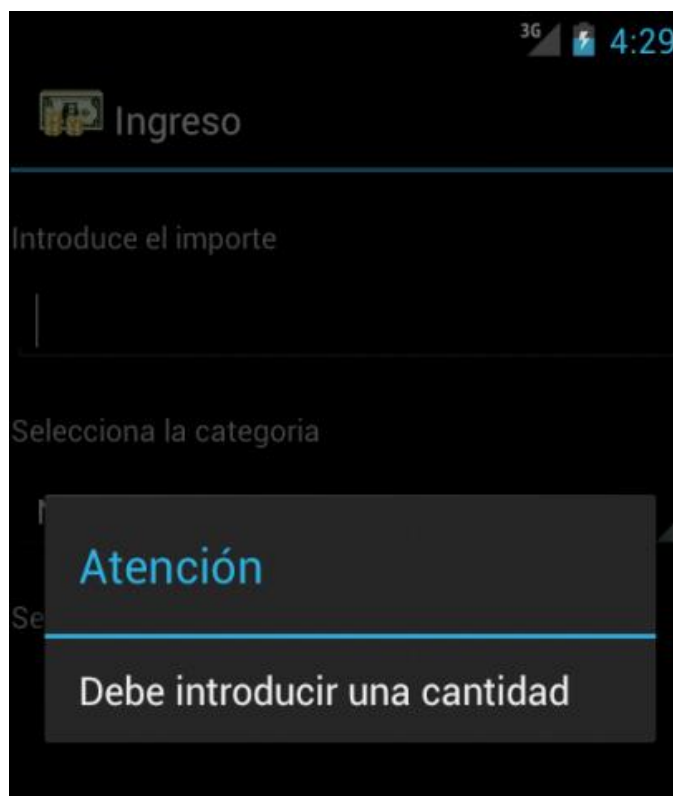
- Introduïm un import de 1000 euros i despleguem el “spinner” per seleccionar la categoria del nostre ingrés, seleccionem una nova data pitjant sobre el calendari, apareixerà un “Date Picker” que ens permetrà fer la selecció



- Un cop introduïdes les dades cliquem en Aceptar i esperem un missatge de confirmació



- En el cas que cliquéssim acceptar sense haver introduït una quantitat d'ingrés, apareixerà un missatge d'error informant a l'usuari



- Aquestes proves son extrapolables pel cas d'introduir una despesa

5.2 Funcionalitat Consulta

- Des de el menú principal de l'aplicació seleccionem Consulta, prèviament s'han introduïts moviments relatius a ingressos i despeses.



- El resultat esperat es una consulta del mes en curs sobre el total ingressat i gastat, si seleccionem la “check box” podem escollir que aquest balanç se'ns mostri segons la categoria seleccionada

3G 4:36

 Consulta

Consulta mes Junio



Ingresado : 0.0

Gastado : 100.0

Total : -100.0

☒ Mostrar por categoria

Alimentación Consultar

 Listar Ingresos  Listar Gastos

- Si cliquem en les icones de la llista d'ingressos o despeses obtindrem un llistat de les quantitats en relació a la seva categoria durant el mes en curs

3G 4:37

 Consulta

Listado de Ingresos Junio

Nómina	1000
Deudas Cobradas	0
Paga Extra	0
Ventas	0
Alquileres	200
Ingresos Extraordinarios	0



3G 4:38

 Consulta

Listado de Gastos Junio

Alimentación	100
Suministro	0
Hipoteca / Alquiler	600
Gasolina	0
Transporte publico	0
Ocio	0
Restaurante	0
Compras	0
Gastos Extraordinarios	0



5.3 Funcionalitat Històric

- Des de el menú principal de l'aplicació seleccionem Històrics, prèviament s'han introduïts moviments relatius a ingressos i despeses per algun mes o any anterior



- El resultat esperat es que s'obri una pantalla de consulta relativa al mes o any demanat que contingui les informacions sol·licitades

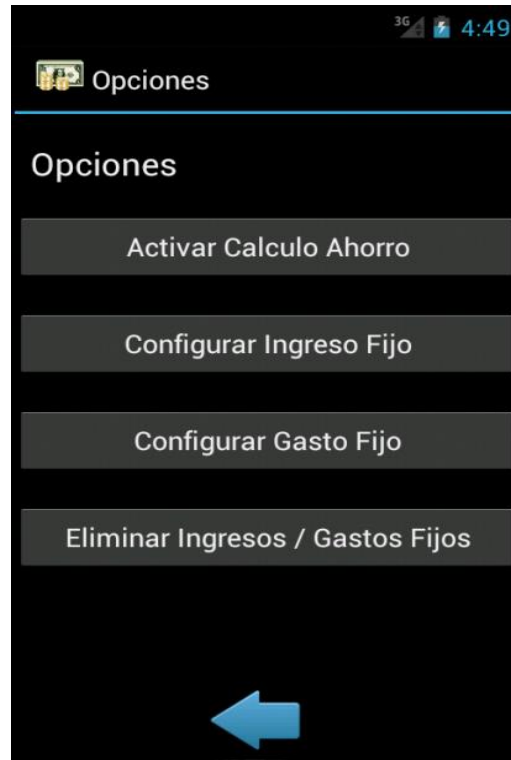


- Les funcionalitats a partir d'aquí seran les mateixes que en el cas de la consulta normal, es important que a partir de clicar la fletxa de “back” les dades del mes de consulta es reiniciïn i per tant la pròxima vegada que es cliqui consulta, aquesta mostri la del mes en curs i no la del històric, veiem que el funcionament es correcte

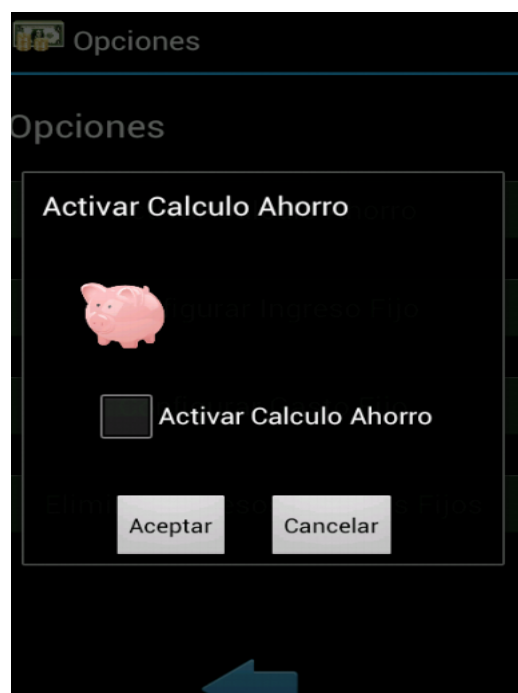


5.4 Funcionalitat Estalvi

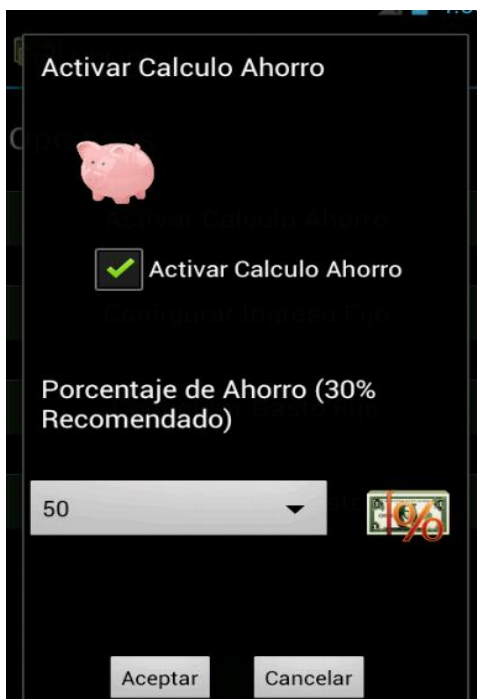
- La opció d'estalvi s'activa des de la pantalla d'opcions per tant des de el menú principal seleccionem opcions



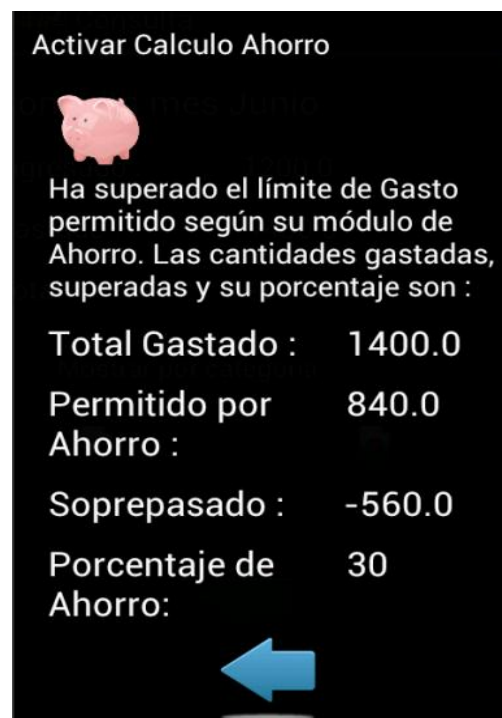
- Al seleccionar la primera opció relativa al calcul de l'estalvi se'ns obrirà un diàleg on podrem activar i configurar aquesta funcionalitat



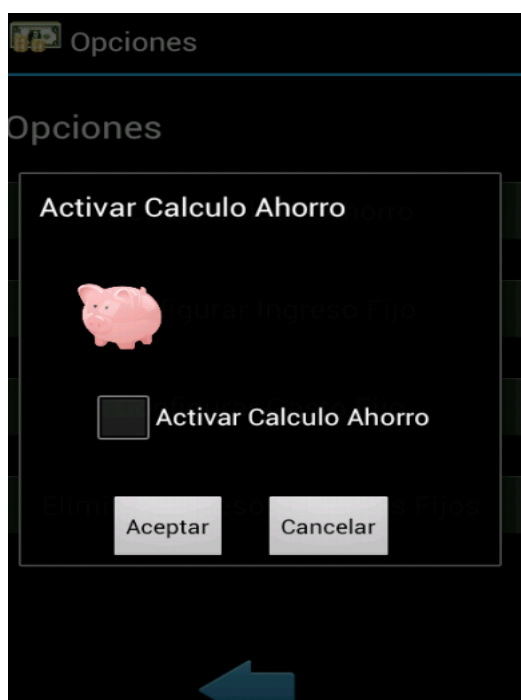
- Al clicar en la “check box” despleguem la funcionalitat i la podem configurar, al clicar aceptar aquesta s'activarà i ens tornarà a la pantalla d'opcions, si tornem a pitjar veurem com la opció activar esta marcada i per tant vol dir que la funcionalitat està activada



- Ara si anem a realitzar una consulta des de el menú principal i em superat aquest limit marcat d'estalvi veurem un diàleg que ens alerta que estem sobrepassant l'estalvi que havíem configurat



- Si des de el menú d'opcions desactivem la opció de l'estalvi, el resultat esperat es que no se'ns mostri el diàleg la pròxima vegada que realitzem la consulta.



6. Conclusiones i possibles ampliacions

6.1 Conclusiones generals

Els objectius principals del projecte s'han assolit. S'ha implementat una solució seguint el disseny universal i que funciona, realitzant correctament les funcionalitats per les quals ha estat pensada l'aplicació. La interfície d'usuari es prou entenedora i després de realitzar algunes proves amb gent de diferent perfil el resultat ha estat bo.

No només durant el procés de la implementació si no també durant l'elaboració sobre la documentació de la plataforma Android s'han assumit els principals conceptes d'aquesta arquitectura i la documentació generada ha estat prou entenedora i específica perquè qualsevol persona amb uns mínims coneixements de programació pugui llegir el treball i entendre una mica millor com funciona un projecte amb aquesta tecnologia.

Alguns problemes que han sorgit durant la implementació que han pogut ser salvats en la seva gran majoria. Però el principal de tots ha sigut la manca de temps, durant la elaboració del projecte han sorgit idees que fan que la aplicació prengui un nou caire i sigui encara molt més completa, aquestes propostes es comentaran en el següent punt.

També m'hagués agradat donar-li una volta més al “look and field” general de l'aplicació i en alguns punts concrets acabar de polir-ho.

El que em porta a pensar que potser era un projecte massa ambiciós donat el temps del que es disposava i que sense la documentació prèvia i més centrat en el desenvolupament de “Where is my money?” hagués obtinguts un resultat molt millors, de totes maneres, i al tractar-se d'un treball de final de carrera considero que era fonamental fer una explicació de la eina amb que es treballava i donat que el resultat de l'aplicació ha estat bo, considero que els objectius s'han assolit satisfactòriament.

6.2 Ampliacions futures

A continuació es presenten alguns punts de millora i possibles ampliacions que han anat sorgint duran la implementació de la aplicació.

- **Introducció de categoria personalitzada** : l'usuari, apart de disposar de les categories prefixades, té la possibilitat de poder personalitzar les seves pròpies categories segons les seves necessitats, i que aquestes es trobin com una part integrada dins de la aplicació
- **Compte compartit i avisos** : Al ser una aplicació orientada a l'estalvi domèstic, va sorgir la idea realitzar una funcionalitat en la que dos usuaris o més poguessin compartir un mateix balanç mensual amb dos o més dispositius mòbils, de tal manera que els ingressos i les despeses introduïdes fossin compartida i s'envies un avis perquè tots els usuaris que comparteixen l'aplicació sàpiguin que hi ha agut

un moviment. De tal manera que els balanços també fossin compartits i així es pugues dur un control de una economia compartida.

Aquesta funcionalitat requeriria la construcció d'un servlet i la introducció de l'arquitectura java J2EE que no s'ha esmentat durant el treball. Seria necessari que el telefon mòbil tingues obligatòriament connexió a Internet i la base de dades en comptes de estar en local hauria d'estar en remot per realitzar una arquitectura de la aplicació mes adient

- **Gràfics** : Es un dels elements que es va començar a implementar i per falta de temps es va haver de descartar, considero fonamental en una següent ampliació de la aplicació que els balanços es mostrin mitjançant gràfics que es facin mes entenedors pels usuaris
- **Reports per dies del mes** : una nova funcionalitat que ens comunicui no només la despesa per mesos sinó també per dies del mes en curs, augmentant així el nivell de detall i control de la nostre economia
- **Selecció de divises** : L'aplicació s'ha pensat per funcionar amb euros, però de cara a fer-la mes internacionalitzada crec que seria bo que es donges la opció a l'usuari de personalitzar les quantitats mostrades en la divisa que el cregui convenient.
- **Aparença colors** : no s'ha configurat un arxiu de colors i no s'han introduïts colors personalitzats en les diferents etiquetes de l'aplicació, sense ser una cosa imprescindible crec que si hi hagués mes temps dedicaria uns quants dies a escollir millor una combinació de colors i d'aquesta manera acabar de mostrar l'aplicació amb un aspecte mes professional

7. Bibliografia i referències

7.1 Llibres

□ Ableson, Frank. Collins, Charlie. Sen, Robi. Android. Guía para desarrolladores “Unlocking Android. A developer’s Guide”. Traducció de José Luis Gómez Celador. 1ª Edició. Ediciones Anaya Multimedia, 2010.

□ Burnette, Ed. Android. “Hello android. Introducing Google's Mobile Development Platform”. Traducció de Pedro Dans Alvarez de Sotomayor. 1ª Edició. Ediciones Anaya Multimedia, 2011.

7.2 Llocs Web

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

Android Developers.

Creació al 2007. Escrit en Anglès.

Pàgina oficial d'*Android*, on apareix tota la API, demos, tutorials i totes les novetats que van incorporants els desenvolupadors d'*Android*.

□ <http://stackoverflow.com/>

StackOverflow. Escrit en Anglès.

Pàgina que ofereix una sèrie de serveis gratuïts, la qual és una barreja entre wiki, blogs i forums i Diig/Reddit. Aquesta pàgina ofereix als programadors un lloc on poder realitzar les seves preguntes sobre programació Android, etc... La resta de programadors usuaris de la pàgina, proposen la resolucions fins a trobar-ne una que funcioni.

□ <http://www.android-spa.com/>

Android-Spa. Escrit en castellà.

És la comunitat oficial de *Android* en Español, on ofereixen un espai obert a tota la comunitat de desenvolupadors Android, amb tutorials, temes de debat i preguntes, notícies de les noves tecnologia *Android*.

□ <https://groups.google.com/group/desarrolladores-android>

desarrolladores-android. Creació al 2007. Escrit en castellà.

Grup d'ajuda per a desenvolupadors d'android, on tenen tutorials i explicacions de resolucions de problemes.

□ <http://www.elandroidelibre.com/>

El Androide Libre. Escrit en castellà.

Web de referència Android en Espanyol. Hi ha notícies, tutorials, etc.. Android i

smartphones de Google.