



Debtoid

Una aplicació de gestió de deutes per Android

Alumne: Víctor Bueno Moreno

Enginyeria tècnica informàtica de Sistemes

Consultors: Marc Domingo Prieto i Antonio Rodríguez Gutiérrez



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

A Sònia, la meva companya de viatge. A la meva família i amics, a qui llegeixi aquestes línies. A tots vosaltres, gràcies pel temps que no us he dedicat i no podré tornar.
A Lluís «Pipo»: Gràcies per la música per a la presentació.

«No serà fàcil però valdrà l'esforç»

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Debtoid: Una aplicació de gestió de deutes per Android</i>
Nom de l'autor:	Víctor Bueno Moreno
Nom dels consultors:	Marc Domingo Prieto i Antonio Rodríguez Gutiérrez
Data de lliurament:	06/2015
Àrea del Treball Final:	Desenvolupament aplicacions dispositius mòbils – Android
Titulació:	<i>Enginyeria tècnica en informàtica de sistemes</i>

Resum del Treball:

La present memòria recull el treball de final de carrera de l'àrea Desenvolupament d'aplicacions per a dispositius mòbils Android en l'àmbit de l'enginyeria tècnica de sistemes informàtics.

Aquesta recull l'elaboració d'una aplicació per a dispositius mòbils amb sistema operatiu Android. El nom donat a l'aplicació és Debtoid.

L'aplicació Debtoid, neix com a resultat de les inquietuds personals envers el sistema operatiu Android i la possibilitat de crear una aplicació per registrar pagaments a mitges. Els objectius més destacables del projecte són crear des de zero una aplicació funcional per al sistema Android i registrar totes les fases del procés de desenvolupament, tot seguint una planificació.

S'exposa el procés de planificació, anàlisi, disseny, proves i implementació de l'aplicació tractant de justificar en cada pas les eleccions triades.

L'objectiu final de l'aplicació consisteix en un sistema que permet al dispositiu mòbil registrar els deutes i pagaments introduïts per l'usuari, amb la possibilitat d'enviar a altres usuaris notificacions mitjançant l'aplicació Whatsapp.

Abstract :

This report reflects the work of the area's final application development for mobile devices on Android on the engineering technical systems scope. This includes the preparation of an application for mobile devices with Android operating system. The name given to the application is Debtoid.

The application Debtoid, born as a result of personal concerns towards the Android operating system and the ability to create an application to register a payment register medium. The most important objectives of the project are to create from scratch a functional application for Android and record all stages of development, just following a plan.

This report tries to expose the process of planning, analysis, design, testing and implementation of the application trying to justify every step elections chosen.

The ultimate aim of the application is a system that allows the mobile device to register debts and payments by user, with the ability to send notifications to other users through the application WhatsApp.

Paraules clau (entre 4 i 8):

Android, mòbil, deutes, notificacions

Índex

Índex

1. Introducció.....	1
1.1. Context del treball.....	1
1.2. Objectius del treball.....	2
1.3. Enfocament i mètode seguit.....	3
1.4. Planificació del treball.....	3
1.5. Breu sumari de productes obtinguts.....	6
1.6. Breu descripció dels altres capítols de la memòria.....	7
2. Anàlisi.....	7
2.1. Mètodes d'indagació.....	8
2.1.1. Justificació i plantejament dels mètodes escollits.....	8
2.1.2. Desenvolupament, resultats i conclusions obtingudes.....	8
2.1.2.1. Observació i investigació.....	8
2.1.2.2. Enquestes.....	11
2.2. Perfils d'usuari detectats.....	19
3. Disseny conceptual.....	20
3.1. Persones.....	20
3.2. Escenaris d'ús.....	21
3.3. Fluxos d'interacció.....	23
3.4. Disseny:.....	25
3.4.1. Sketches:.....	25
3.4.2. Prototip horitzontal (alta fidelitat):.....	25
3.4.3. Avaluació:.....	26
4. Implementació.....	28
4.1. Eines utilitzades i justificació.....	28
4.1.1. Android Studio.....	28
4.1.2. Inkscape.....	30
4.1.3. PowerCMD.....	31
4.2. Estructura de Debtoid.....	32
4.2.1. La base de dades de Debtoid.....	33
4.2.2. Model de l'aplicació.....	35
4.2.3. Controlador de l'aplicació.....	36
4.2.4. Vistes del projecte.....	42
4.2.5. Manifest.....	47
4.3. Proves realitzades.....	48
4.4. Presentació.....	48
5. Estat actual del projecte.....	49
5.1. Futures millores.....	50
6. Conclusions.....	51
6. Conclusions.....	51
7. Glossari.....	52
7. Glossari.....	52
8. Bibliografia.....	53

8. Bibliografia.....	53
9. Annexos.....	53
9. Annexos.....	53
9.1.Instal·lació.....	53
9.2.Manual d'usuari.....	56

Llista de figures

Banner UOC.....	i
Banner Debtoid - Portada.....	i
Llicència.....	ii
Gràfic evolució de sistemes operatius.....	1
Diagrama de Gantt 1/2.....	5
Diagrama de Gantt 2/2.....	6
SplitWise 1/2.....	9
SplitWise 2/2.....	9
Conmigo 1/2.....	10
Conmigo 2/2.....	10
I Owe you 1/2.....	10
I Owe you 2/2.....	10
Enquesta - Sexe mostra.....	14
Enquesta - mostra pagaments a mitges.....	14
Enquesta - mostra amb tlf.....	15
Enquesta - Orientació.....	16
Enquesta com afegir dades.....	16
Enquesta enviament de notificacions.....	16
Enquesta Email.....	17
Enquesta SMS.....	17
Enquesta notificacions.....	17
Persones - Toni.....	20
Persones Silvia.....	20
Persones Carles.....	21
Flux interacció.....	23
Sketch 1.....	26
Sketch 2.....	26
Sketch 3.....	26
Sketch 4.....	26
Prototip 1.....	26
Prototip 2.....	26
Prototip 3.....	26
Prototip 4.....	26
Android Studio.....	29
Android Studio Estructura de projecte.....	29
Android Studio - Work in progress.....	30
Android Studio - Consola.....	30
PowerCMD.....	31
MVC.....	32

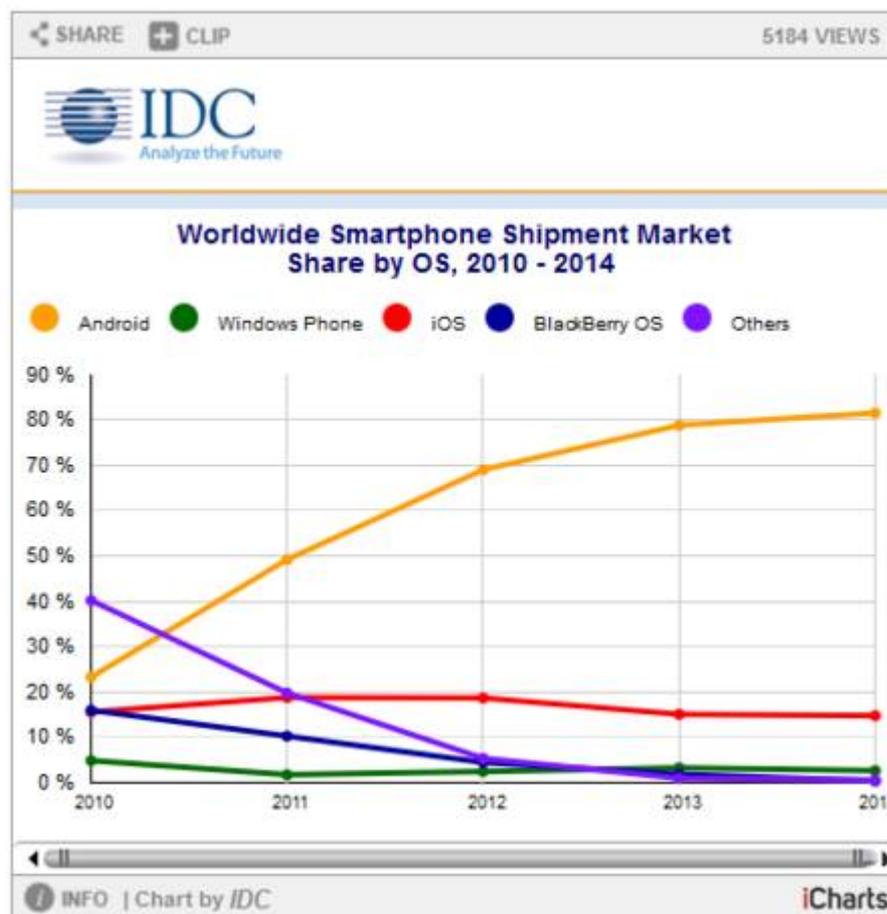
Base de dades.....	33
exemple base de dades.....	35
Exemple consultes simple.....	35
Exemple consulta a ma.....	36
cicle de vida de l'Activity.....	37
Exemple codi Total deutes.....	38
Exemple codi Adapters.....	39
Exemple classe GrupsMet.....	40
Exemple extracció URI de contacte.....	41
esquema base de dades intera de contactes 1.....	41
esquema base de dades intera de contactes 2.....	42
Exemple codi notificació Whatsapp.....	42
Exemple codi per donar per pagat.....	43
Codi XML vista Main.....	44
Vista Main a emulador.....	45
Codi exemple preferències.....	46
Vista preferències.....	46
Vista cobraments 1.....	47
Vista cobraments 2.....	47
Exemple notificació Whatsapp.....	47
Exemple codi Manifest.....	48
Camtasia.....	49
Segmentació de versions d'Android.....	51
compilació d'un projecte.....	55
Executar projecte.....	55
Seleccionar dispositiu.....	56
Pantalla principal.....	57

1.Introducció

1.1. Context del treball

Actualment el mercat de dispositius mòbils és majorment governat per tres sistemes operatius: Windows, iOS i Android. D'aquests tres, Android és el que té més quota de mercat – prop del 80%:

<http://www.xatakamovil.com/sistemas-operativos/ios-mas-fuerte-que-nunca-en-espana>



Avui en dia es disposa de milers d'aplicacions per a dispositius mòbils per cobrir pràcticament qualsevol necessitat que tinguem. Els Smartphones estan integrats al nostre estil de vida en molts aspectes, un telèfon mòbil ha deixat de ser un dispositiu per fer trucades i rebre'n, i ocasionalment enviar un missatge de text. Des de fa alguns anys el disposar d'un aparell on centralitzar gran part de les nostres activitats i comunicacions, ha esdevingut una prioritat important.

La idea per al TFC neix després d'haver detectat una necessitat a nivell personal: Portar un registre de pagaments i cobraments fets entre grups d'amics.

Sovint, es dona el cas de què, per un regal fet entre diferents persones en un grup, o una participació de loteria, un viatge, una celebració o qualsevol altre pagament a fraccionar, algú queda pendent de pagar en el mateix moment a algú altre, i no sempre és fàcil ni agradable recordar qui ha pagat què a qui.

La idea de Debtoid és disposar d'una aplicació mòbil per poder portar un registre de cada "grup de cobrament", mostrant les dades més rellevants, permetent diverses opcions com ara l'enviament de recordatoris als deutors, i notificacions a l'usuari, ja sigui deutor o cobrador.

Tot i que ja hi ha algunes aplicacions amb aquesta funció o bé de semblants, crec que encara es pot innovar i millorar més, degut a que hi ha algunes funcionalitats que no estan implementades a aquestes aplicacions. Per exemple, altres aplicacions no permeten enviar un missatge a un altre usuari, o notificar al propi usuari de deutes pendents amb una alarma periòdica.

L'objectiu general del treball, per tant, és crear una aplicació per dispositius Android mitjançant la qual es pugui portar el registre de diferents cobraments, deutes i pagaments a nivell particular, així com disposar de notificacions, aprofitant els serveis que ofereix el Sistema Operatiu.

1.2. Objectius del treball

L'objectiu general del projecte, com s'ha esmentat al punt anterior, és el de crear una aplicació on registrar, modificar i posteriorment consultar una sèrie de dades en dispositius Android. De forma opcional, s'afegiran una sèrie de funcionalitats tals com la possibilitat de notificar a usuaris, enviar missatges, o fer backup de les dades a un servidor extern.

Es pretén cobrir com a mínim els següents aspectes:

- Desenvolupar una aplicació per a mòbil Android des de zero.
- Utilitzar el llenguatge de programació Java i l'IDE Android Studio.
- Crear les icones de l'aplicació i donar un aspecte adequat a l'aplicació.
- Poder consultar i registrar tant deutes, pagaments realitzats i cobrats, tant propis com d'altres usuaris a l'aplicació.
- Portar registres per diferents grups d'usuaris.
- Notificar periòdicament a l'usuari segons es configuri .
- Enviar sota demanda notificacions a altres usuaris des de l'aplicació.
- Presentar i exposar de forma correcta l'aplicació.
- Complir amb la planificació de temps durant el projecte.
- L'aplicació ha de ser intuïtiva, per tant, ha de ser senzilla d'utilitzar.

1.3. Enfocament i mètode seguit

Donat que el projecte es realitza per una sola persona, i que el temps de dedicació serà d'entre 10 i 18 hores setmanals (en festius és possible dedicar més temps que en laborals), com a metodologia "Lean" s'utilitzarà Kanban, actualitzant cada dia els estats .

Mitjançant Kanban, es pot tenir a la vista en tot moment en què fase es troba cada part del projecte d'un cop d'ull, què hi ha pendent, què hi ha ja fet, i què és més urgent d'avançar.

Com a llenguatge de programació s'utilitzarà Java, amb l'IDE Android Studio, degut a la integració i facilitat d'ús en comparació amb altres opcions com per exemple l'IDE Eclipse. L'elecció de Java és deguda a la gran quantitat de documentació sobre aquest llenguatge. A més, actualment Android Studio és l'IDE recomanat per Google i al que li donen més suport.

S'ha decidit optar per crear una aplicació nova, ja que és més senzill que aconseguir el codi font d'una de les aplicacions ja existents i modificar-les, especialment tenint en compte que són aplicacions comercials. A més, en finalitzar el projecte es vol publicar a Play Store l'aplicació, incloent-hi publicitat per tal de monetitzar l'aplicació. Per tant, és important que sigui una aplicació pròpia.

1.4. Planificació del treball

Per tal de realitzar el projecte, són necessaris els següents elements:

HARDWARE:

- Ordinador de sobretaula, amb Windows 7
- Telèfon mòbil Android, model Xiaomi Redmi Note, per fer proves.
- Telèfon mòbil samsung Galaxy S3 Mini.
- Telèfon mòbil amb Whatsapp instal·lat.
- càmera de vídeo (per a la presentació).
- «pissarra, post-its i marcadors» - útil per implementar Kanban.

SOFTWARE:

- IDE Android Studio
- Editor d'imatges vectorials Inkscape
- Software d'edició d'àudio/vídeo Camtasia
- Conjunt d'ofimàtica LibreOffice
- Eina per diagrames Gantt: Tom's Planner (web).
- Balsamiq Mockups
- Prototyper (JustInMind)

Es pot concloure que no hi ha un cost econòmic significatiu degut a que tot el material necessari, o bé és gratuït, o bé actualment ja es disposa d'ell.

No s'ha fet ús de cap servidor extern degut a què no s'ha arribat a implementar la funcionalitat online.

L'únic risc a considerar és el temps a invertir durant el projecte.

S'esperava poder invertir entre una hora i tres hores diàries en dies laborals. Els festius i caps de setmana s'esperava invertir-hi més temps, i es van deixar alguns dies de més en la planificació per possibles imprevists.

Finalment, no s'ha pogut invertir tant de temps com estava previst.

Les dates clau han sigut les següents:

	25/02/2015	Inici de projecte
Fita 1: 14 dies	11/03/2015	Entrega PAC1
Fita 2: 28 dies	08/04/2015	Entrega PAC2
Fita 3: 42 dies	20/05/2015	Entrega PAC3
Fita 4: 32 dies	21/06/2015	Lliurament final

El que ens dona un total de 116 dies, els quals s'ha tractarà de que segueixin la següent planificació estructurada per fites:

Fita 1: Finalitzada amb la entrega de la PAC 1. En aquesta es va planificar tot el projecte i es va temporitzar i documentar tot el procés. Es van establir les fites per a les dates acordades.

Fita 2: Finalitzada amb la entrega de la PAC 2. En aquesta fase es va analitzar el problema a tractar, comparant algunes aplicacions que ja el tracten i es van buscar possibles millores. També es va realitzar el disseny gràfic i es van especificar els requisits funcionals i no funcionals. Per últim, s'esperava poder començar a codificar en aquesta fase, però no va ser encara possible, més allà de realitzar algunes vistes bàsiques.

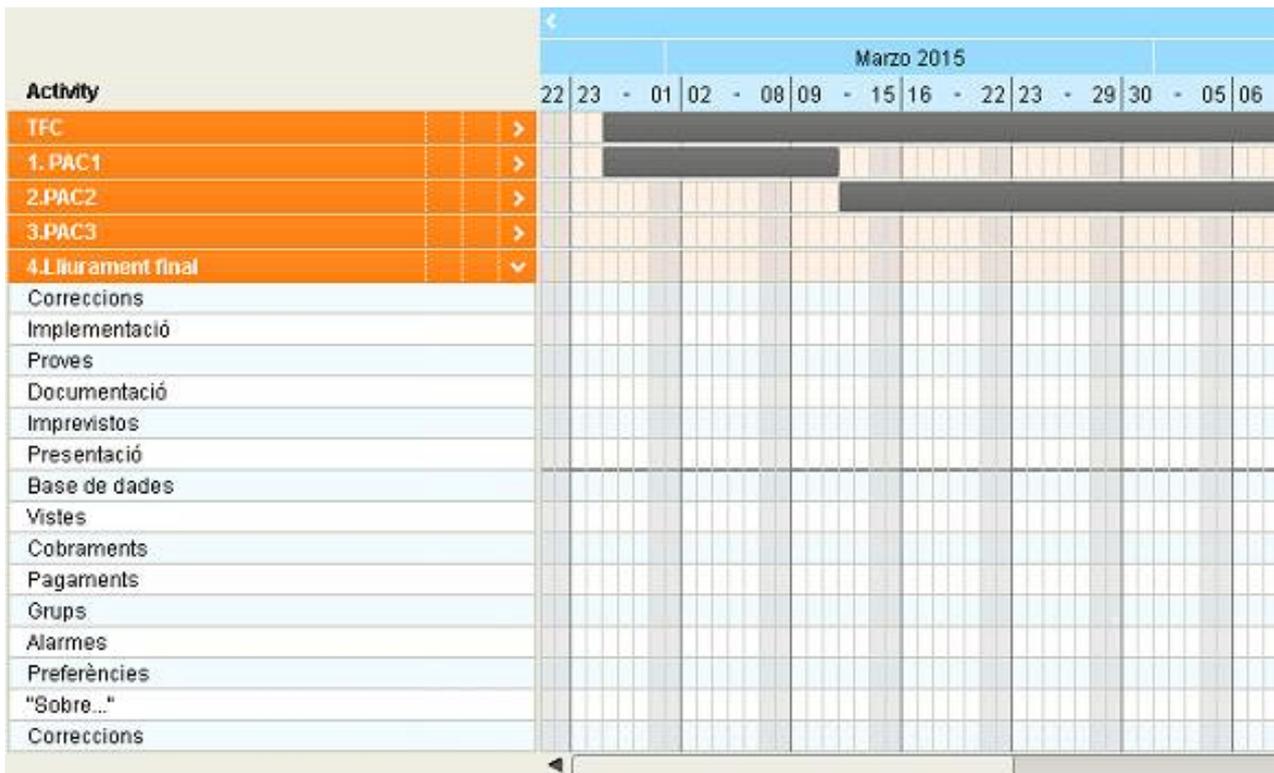
Fita 3: Finalitzada amb la entrega de la PAC 3. En aquesta fase es van començar a implementar les funcionalitats, la base de dades, etc. Paral·lelament es van anar realitzant proves segons es codificava. Com a resultat, s'esperava lliurar un prototip amb entre el 70 i el 80% de les funcionalitats i configurar el servidor per tenir una copia de les dades.

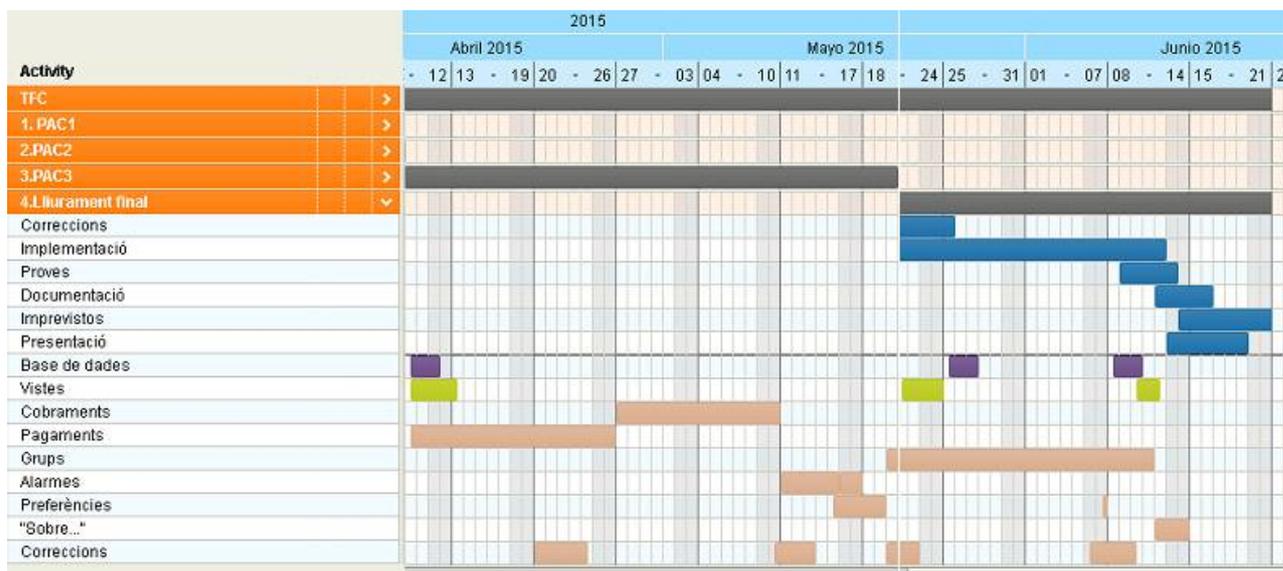
Finalment es va entregar una part molt reduïda de l'aplicació i al no realitzar-se el mode online. No s'utilitza el servidor.

Lliurament final:

Finalitzat amb el lliurament de l'aplicació ja complet, de aquesta memòria, i la presentació, així com dels materials annexos (instal·lació, codi, etc). Aquesta fase va ser semblant a la fase 3, es van avançar les tasques de les darreres setmanes, especialment la documentació i proves, per tal de poder complir amb les dates fixades. Els darrers dies es va fer la gravació de la presentació, per tant l'aplicació estava acabada.

La planificació que finalment es va seguir, es correspon amb el següent diagrama de Gantt:





Nota: En les fites 2, 3 i final, les fases d'implementació i proves coincidiran alguns dies, per poder corregir mals funcionaments. A la pràctica, és d'esperar que bona part de la implementació es vagi provant a mesura que es vagi desenvolupant: Per exemple, la comunicació amb el servidor, la base de dades, etc. tot i que les proves exhaustives es faran en les dates planificades.

Posteriorment al lliurament final, es respondrà a les preguntes del tribunal (des del 29/06/2015 fins al 03/07/2015).

1.5. Breu sumari de productes obtinguts

Al final del treball s'han entregat els següents documents:

- Memòria: Document on s'explica tot el procés de desenvolupament dut a terme.
- Aplicació: Aplicació en format .apk per instal·lar a dispositius Android. A més, s'entregarà un arxiu .zip o .rar amb la configuració del servidor.
- Presentació en vídeo: gravació explicativa sobre el treball i sobre l'aplicació.
- Altres: Altres arxius, com per exemple el codi.

1.6. Breu descripció dels altres capítols de la memòria

- Introducció: Capítol actual, on es presenta el treball, la metodologia emprada i la planificació inicial.
- Anàlisi: S'analitza el problema a tractar i es compara amb altres aplicacions actuals.
- Disseny: Es defineixen els requisits i s'especifica com funciona l'aplicació.
- Interfície d'usuari: Es defineixen els aspectes gràfics de l'aplicació.
- Implementació: S'explica com s'ha implementat l'aplicació, tot seguint el model MVC i la metodologia Kanban. S'expliquen breument ambdós conceptes.
- Conclusions: Capítol on es tracta què s'ha l'après, què ha faltat i el que es pot millorar.
- Glossari
- Bibliografia
- Annexos: Instal·lació i manual d'usuari.

2.Anàlisi

En aquest apartat es descriu com s'ha realitzat la fase d'anàlisi.

En primer lloc es va tractar d'identificar els possibles contextos d'ús de l'aplicació, els possibles usuaris i les seves necessitats.

En segon lloc es va tractar d'identificar l'usuari objectiu i les funcionalitats que necessitaria l'aplicació en funció de les dades obtingudes a la part anterior.

2.1. Mètodes d'indagació

2.1.1. Justificació i plantejament dels mètodes escollits

L'aplicació inicialment va dirigida a un públic variat. No s'espera que hi hagi un perfil professional ni acadèmic específic degut a què es podria classificar com a mig camí entre oci i finances personals.

Com s'ha comentat anteriorment, l'alumne és un futur usuari de l'aplicació, motiu pel qual pot donar una visió propera a la de l'usuari objectiu.

Per aquests factors es va optar per fer servir els dos mètodes següents:

- Observació i investigació:

L'alumne és coneixedor de la tasca que la aplicació ha de realitzar, i de fet, acostuma a realitzar-la amb altres medis. Per aquest motiu pot aportar opinions útils. A més, es va realitzar una comparativa entre algunes aplicacions semblants que ja existeixen al mercat per detectar punts forts i febles.

- Enquestes:

Es van realitzar enquestes per tal de conèixer algunes dades tant estadístiques com opinions del tipus d'usuari que pot estar interessat en l'aplicació. En aquest cas, l'enquesta va ser realitzada a persones de diferents àmbits, degut a què no hi ha un públic específic.

2.1.2. Desenvolupament, resultats i conclusions obtingudes.

2.1.2.1. Observació i investigació

Al començament de la investigació, ja es coneixia el problema a tractar, degut a què sovint s'havia d'enfrontar amb mitjans diferents i es solia veure a l'ambient familiar, d'amistats, etc.

Habitualment la majoria de la gent que l'alumne coneix no acostuma a portar una agenda a sobre, però sí que pràcticament tothom porta a sobre un telèfon mòbil, i aquest té una sèrie de funcionalitats -agenda inclosa- que per separat serveixen per solucionar d'una manera més feixuga la mateixa tasca que l'aplicació té per objectiu.

En aquest punt es podria plantejar que potser l'aplicació no és realment necessària, però després de provar durant algunes situacions el procediment emprat, es veu clar que no s'aprofiten els recursos del mòbil per falta de comoditat, sobre tot. Els dos casos observats van ser els següents:

1.- Euromillón: Es tracta d'un joc de loteria bastant conegut. El funcionament observat era el següent:

Una persona, alies «V» recollia la mateixa quantitat de tots els participants mensualment, i feia el pagament a una web de loteries amb els números prèviament escollits entre tots. Habitualment es donaven les següents situacions:

- No es pagava a V abans de pagar la loteria.
- Cada amic de V pagava al cap d'un temps (unes hores, un dia o dos, una setmana..)
- V inicialment no mantenia cap registre dels pagaments. Posteriorment va utilitzar un bloc de notes físic, després a un ordinador, i enviava recordatoris mitjançant Whatsapp als participants de la loteria. També va utilitzar una alarma al mòbil per recordar demanar als participants el pagament.
- Els participants no sempre recordaven l'import. Aquest variava en funció de si havia hagut premi el mes anterior o no.
- Alguns cops algun participant pagava una part i per qualsevol motiu fins a un altre moment no disposava de la resta.

2.- Regal aniversari entre amics:

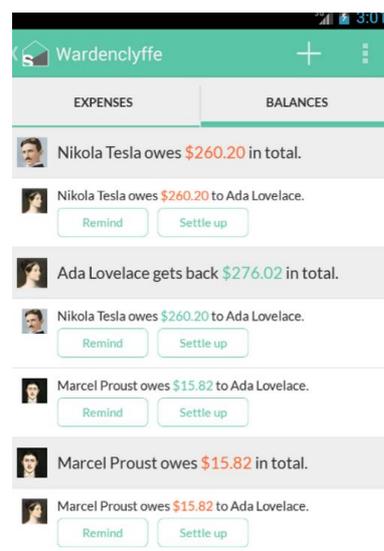
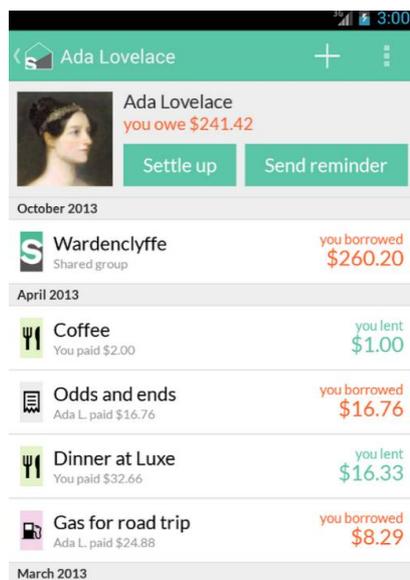
Es tracta d'un cas molt semblant a l'anterior. Varia el fet de què l'objectiu del pagament és diferent. La resta de situacions es donen de manera semblant. La única diferència rellevant és el fet de què el grup de participants no es veia a diari com en el cas anterior.

Posteriorment, l'alumne va optar per provar algunes aplicacions mòbils. Va optar per les següents:

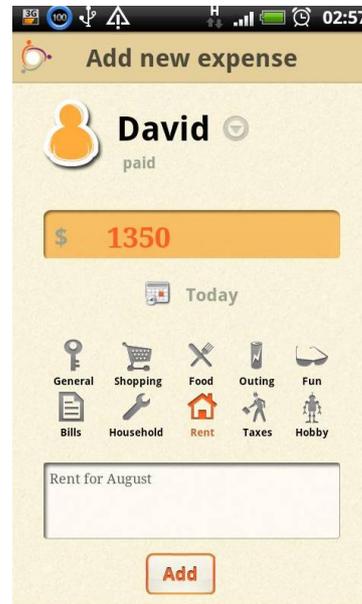
-*Splitwise*: Gràficament és bastant agradable. Permet gestionar grups o usuaris individualment. Mitjançant la mateixa aplicació permet que altres usuaris notifiquin i té bastantes opcions de configuració. És un exemple a tenir en compte. Tot i això, està molt orientat a un públic concret: Persones que comparteixen pis, sigui de vacances o normalment. També disposen d'interfície web: <https://www.splitwise.com/>

Imatges i descàrrega: [https://play.google.com/store/apps/details?](https://play.google.com/store/apps/details?id=com.Splitwise.SplitwiseMobile)

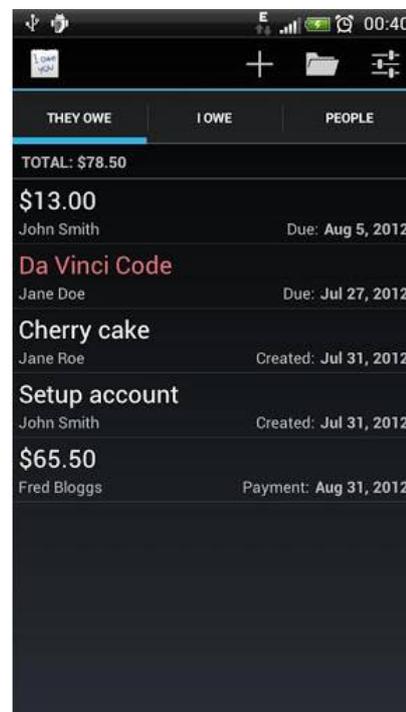
[id=com.Splitwise.SplitwiseMobile](https://play.google.com/store/apps/details?id=com.Splitwise.SplitwiseMobile)



-*Connmigo*: Aquesta aplicació es destaca en l'aspecte visual sobre les altres. No obstant, costa entendre al principi com fer-la funcionar degut a la forma en que presenta les dades. A més, no té algunes funcions com la de notificar als contactes. Web: <https://play.google.com/store/apps/details?id=com.connmigo.share&hl=es>



-*I owe you*: Té la major part de les característiques que es busquen: Diferents divises, contactes de l'agenda, deute propi i aliè, alarmes..., es podria millorar l'aspecte gràfic. No s'ha pogut provar molt a fons degut a què moltes funcions funcionen solament mitjançant micropagaments. En el cas de Debtoid, l'aplicació serà per complet gratuïta, tot i què és possible que s'arribi a comercialitzar amb publicitat. Referència: <http://ioutool.net/>



Les conclusions a les que es van arribar són:

- Es busca una aplicació amb funcionalitats similars a les de les aplicacions estudiades.
- El aspecte de l'aplicació és important, s'ha d'arribar a un compromís entre utilitat i estètica.
- Les funcionalitats han d'estar disponibles sense pagament o bloquejos addicionals.

2.1.2.2. Enquestes

A més de tenir en compte la observació i la comparació entre aplicacions, es van realitzar unes enquestes per tal de definir millor el perfil d'usuari al que es dirigirà l'aplicació. El formulari d'enquesta es va realitzar mitjançant Google Forms, i es va posar en coneixement de diferents grups de coneguts de l'alumne. L'adreça del formulari és la següent:

<http://goo.gl/forms/0QIJYnWZWW>

Es va fer el formulari de l'enquesta en castellà, degut a què hi participaven persones de diferents zones. Es van registrar enquestes de prop de 80 persones. Les preguntes es van segmentar en tres pàgines, la segona de les quals variava en funció del sistema operatiu del mòbil. Tot i que hi hagi usuaris de sistemes diferents a Android, es va creure oportú tenir en compte les seves opinions. La tercera pàgina era comú als dos tipus d'enquestat, i tractava les preferències en l'aplicació.

Les preguntes realitzades van ser les següents:

Nom: En cas de que es continués l'enquesta, o es distribuís una versió de prova de l'aplicació, seria interessant poder tornar a contactar amb els usuaris que voluntàriament s'ofereixin.

Sexe: Es va considerar important tenir en compte aquesta dada per saber si hi ha un sexe que utilitzi més aquest tipus d'aplicacions que l'altre.

Edat: És important tenir en compte l'edat predominant, tant pel fet de realitzar un disseny més orientat a una franja d'edat o a una altre, com per conèixer millor el perfil d'usuari.

Ocupació: Tot i no ser rellevant en aquesta aplicació, va semblar una dada estadística a tenir en compte.

Acostumes a fer pagaments a mitges?: Aquesta pregunta està destinada a

avaluar si els enquestats realment poden tenir necessitat d'utilitzar l'aplicació. Lògicament, si no es fan pagaments a mitges, no es té necessitat de l'aplicació.

Disposes de mòbil o tableta Android?: En funció de la resposta, es redirigeix a l'usuari a la pàgina de preguntes per a sistemes Android o sistemes diferents d'Android.

A la pàgina de preguntes per a sistemes diferents d'Android, les qüestions van ser les següents:

Disposes de mòbil o tableta?: Amb aquesta pregunta es pot esbrinar si existeix algun enquestat que no disposi de telèfon mòbil o bé de tableta.

En cas de tenir-ne, De quina marca i model és?: Es tracta de conèixer, per a futurs desenvolupaments, els dispositius i sistemes del públic objectiu.

Coneixes alguna aplicació de l'estil d'Splitwise, Conmigo, I Owe You, o d'altres semblants destinades a registrar pagaments i deutes petits?

Quina opinió en tens?: La idea d'aquesta pregunta és conèixer d'una banda la opinió dels usuaris respecte a aquestes aplicacions. Per una altra banda, és interessant conèixer altres aplicacions d'altres sistemes per agafar idees si es dona el cas de tenir-ne accés.

Utilitzes el mòbil o tableta per apuntar cobraments i/o deutes pendents a nivell particular? Si es que sí, amb quina aplicació ho fas?: Amb aquesta qüestió s'intentava esbrinar si hi ha realment un mercat potencial que ja utilitzi aplicacions d'aquest tipus. En cas d'haver-ne, es tracta de recollir quines aplicacions s'utilitzen amb aquesta finalitat.

A la pàgina de preguntes per a sistemes Android, les qüestions van ser les següents:

Coneixes la marca i el model del teu mòbil?: Algunes marques tenen funcions especials, o disposen d'una interfície diferent, per exemple Sony Xperia, o Xiaomi.

Coneixes alguna aplicació de l'estil d'Splitwise, Conmigo, I Owe You, o d'altres semblants destinades a registrar pagaments i deutes petits?

Quina opinió en tens?: La idea d'aquesta pregunta és conèixer la opinió dels usuaris respecte a aquestes aplicacions.

Utilitzes el mòbil per apuntar cobraments i/o deutes a nivell particular?

Amb quina aplicació?: Es pot utilitzar per conèixer els sistemes diferents, com per exemple blocs de notes, que utilitzen els usuaris habitualment els usuaris en lloc d'aplicacions semblants. Amb aquesta informació és possible millorar l'experiència d'usuari afegint o traient funcionalitats.

Coneixes la versió del sistema?: De cara al disseny de l'aplicació i d'especificar quines funcions del sistema es poden utilitzar i quines no, és

important establir una versió d'Android a partir de la qual l'aplicació garantirà funcionar correctament.

A la pàgina de preguntes de preferències, les qüestions van ser les següents:

Per aquest tipus d'aplicació, prefereixo que el mòbil tingui orientació: Es donava a escollir entre vertical i horitzontal. Segons es va observar, aquest tipus d'aplicació acostuma a presentar-se en vertical, però podria donar-se el cas de què els usuaris estiguin interessats en un altre format. En cas de disposar de suficient temps, es tractarà de permetre a l'aplicació funcionar en horitzontal, pel que serà necessari dissenyar en horitzontal l'aplicació.

Prefereixo afegir les dades: Es donava a escollir entre manualment o mitjançant selectors. Segons es va observar a altres aplicacions, es sol utilitzar una combinació, segons la dada a introduir. Per exemple, es pot introduir el contacte de la llista, o bé es pot introduir manualment, no obstant, el rellotge de notificacions és sempre mitjançant selectors.

T'agradaria poder enviar missatges tipus «em deus X diners en concepte de Y» mitjançant Whatsapp?: L'objectiu en aquest cas és bastant clar, esbrinar si els futurs usuaris estan disposats a utilitzar Whatsapp com a sistema de missatgeria per l'aplicació.

I mitjançant correu electrònic? I mitjançant SMS?: L'objectiu és semblant, es tracta d'observar quin dels sistemes de missatgeria els usuaris volen utilitzar i quins no.

T'agradaria poder configurar alarmes per a tu mateix a l'aplicació?: Es tractava de avaluar si els usuaris realment estan interessats en tenir un sistema d'alarmes propi integrat a l'aplicació.

T'agradaria que l'aplicació et demani confirmació per incloure't a grups i t'informi del teu estat al grup?: Donat que ser afegit pot ser molest per als usuaris, es va preguntar si preferirien confirmar abans de ser inclosos a un grup i si voldrien poder veure l'estat del grup.

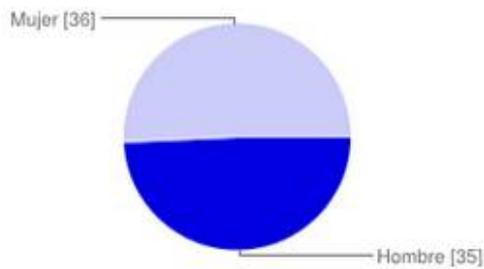
Hi ha alguna característica que voldries tenir en una aplicació d'aquest tipus?: La idea d'aquesta pregunta és rebre idees dels possibles futurs usuaris. Val a dir que la idea no era incloure tot el que demanessin els usuaris, sinó tenir en compte les seves aportacions.

Si estas interessat en continuar col·laborant provant l'aplicació, deixa'm el teu telèfon o correu electrònic: Es tractava de recollir dades de contacte voluntàries, per tal de tenir un grup d'usuaris més endavant per fer proves.

Els resultats més rellevants de les enquestes van ser els següents:

-El sexe estava bastant equilibrat, gairebé la meitat eren homes i l'altra meitat dones:

Sexo



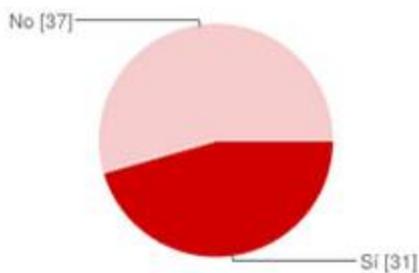
Hombre	35	49.3%
Mujer	36	50.7%

-De les franges d'edat, la predominant es troba entre 31 i 35 anys, seguida de la franja d'entre 26 i 30. Per tant, podem situar la majoria dels usuaris entre 26 i 35 anys.

-Respecte a les professions, hi van haver de diferents tipus, però es va donar el cas de què degut a l'entorn en què es van realitzar les enquestes pertanyien a l'entorn de la informàtica en la gran majoria.

-Part dels enquestats que van contestar realitza pagaments a mitges. La majoria diu no fer-ho. No obstant, hi ha una proporció suficient per tal de creure viable l'aplicació.

¿Sueles hacer pagos a medias?



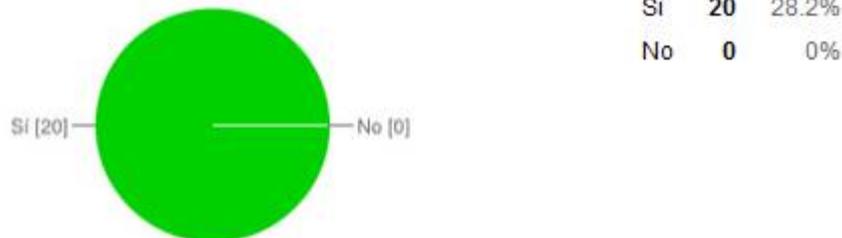
Sí	31	43.7%
No	37	52.1%

-La majoria disposava de mòbil o bé de tableta Android.

SECCIÓ NO ANDROID:

-De la resta d'enquestats, tots disposen de telèfon o bé tableta. De fet, es pot esperar que pràcticament qualsevol persona de més de 16 anys disposi d'un Smartphone.

¿Dispones de mòbil o tablet?



-Les marques i models no Android són sobre tot de la companyia Apple.

-Els usuaris coneixen les aplicacions similars següents (no s'han pogut provar):

- Expense (sailfish os)
- Una aplicació el nom de la qual l'usuari no recorda el nom, però no li va agradar que fos en entorn web.
- Paypal

-També utilitzen les aplicacions no orientades a registrament de pagaments i deutes següents:

- Excel
- Bloc de notes
- Aplicació nativa «Notas» del sistema ios.

SECCIÓ ANDROID:

-Predominen les marques Samsung i Sony en Android.

-Els usuaris que coneixen aplicacions d'aquest tipus (no tothom les coneixen), coneixen les següents:

- Tricount
- Leetchi
- Splitwise

Les valoracions de les dues darreres són bones. No es va arribar a provar Leetchi ni Tricount, tot i que sí es va provar Splitwise, compartint bones opinions amb els enquestats.

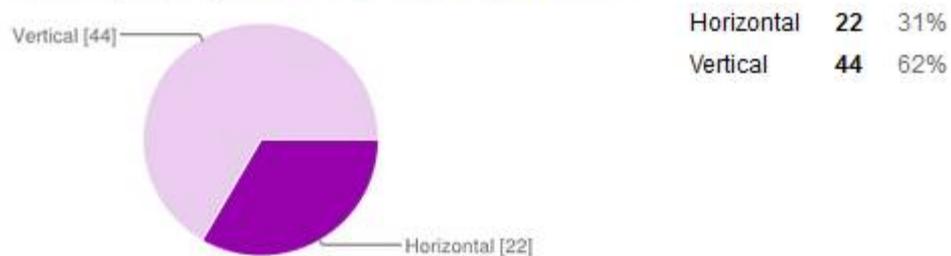
-Els usuaris que no utilitzen aplicacions orientades a aquest ús, acostumen a utilitzar les següents aplicacions:

- Google Keep
- Note everything
- Email propi
- Recordatoris (alarmes)
- Color note

-La majoria dels usuaris tenen versions posteriors a 4.0. Tot i què n' existeixen alguns amb versions anteriors.

-Els enquestats prefereixen en general tenir orientació vertical en l'aplicació:

Para este tipo de aplicación, prefiro orientación



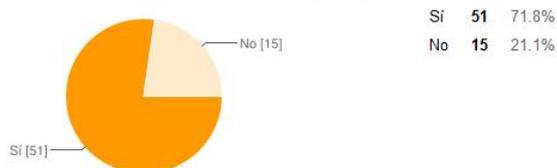
-En general, els usuaris prefereixen introduir les dades manualment en lloc de seleccionar-les d'una llista:

Prefiero añadir los datos

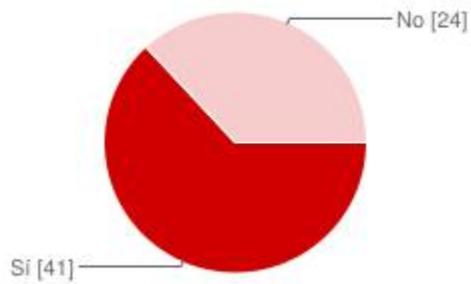


-Als enquestats els interessa el poder enviar per Whatsapp notificacions, per correu electrònic també, però per SMS no tant:

¿Te gustaría enviar mensajes recordatorios, por ejemplo: "me debes X euros en concepto de Y" a otras personas mediante whatsapp?

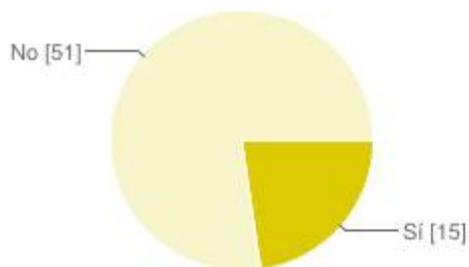


¿Y mediante email?



Sí	41	57.7%
No	24	33.8%

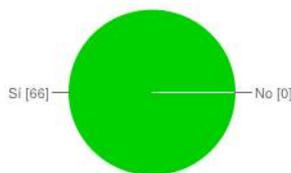
¿Y mediante SMS?



Sí	15	21.1%
No	51	71.8%

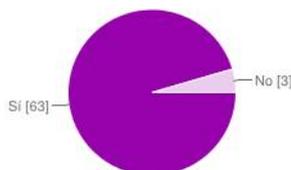
-També es va descobrir que als enquestats els interessava poder configurar alarmes i que l'aplicació els demani permís per incloure'ls en un grup, així com poder rebre informació del grup.

¿Te gustaría poder configurar alarmas para ti mismo en la aplicación?



Sí	66	93%
No	0	0%

¿Te gustaría que la aplicación te pida confirmación antes de incluirte en un grupo y te mantenga informado de tu estado en el grupo?



Sí	63	88.7%
No	3	4.2%

-De les suggerències rebudes, les que destaquen són les següents (en castellà:

- *Que destaque con colores, avisos o algo cuando algunos conceptos que el usuario elija superen ciertas cifras o límites máximos parametrizables por el usuario (para que no se le vaya de madre).*
- *Poder silenciar las alarmas o recordatorios desde el principio, sólo vibración*
- *Historial de primitiva jugado*
- *Poder escoger tipos de pagos que se deben, de una lista de preseleccionados, así como las cantidades.*
- *que quede claro con colores o de alguna manera visual que debes o te deben dinero*
- *Recordatorios*
- *El uso de emoticonos, ya que habitualmente son menos agresivos en su interpretación cuando reclamas diner*
- *Una recomendación: en los mensajes recordatorios, en vez de utilizar la palabra "debes" se podría sustituir por "pendiente pagar", por ejemplo: "tienes pendiente pagar X euros en concepto de...". El "me debes" suena muy brusco. Es mi humilde opinión.*
- *Poder mantener listas viejas de compras conjuntas, copiarlas (se suele participar en grupo con los mimos), editarlas. Poder añadir notas ("a esta persiguela que si no, no paga")*
- *Estilo facil y visual. Todo lo mas automatico posible.*
- *Preferiria que pudiese estar tanto en horizontal como en vertical*
- *Que no sea muy cansina con los recordatorios.*
- *Cómo diste el dinero, efectivo, transferencia..*
- *Que no tenga publicidad, por favor! ;)*
- *Estaría bien que se pudiera poner en un gráfico por tipo de deuda (i.e. vacaciones, regalo de cumpleaños, compras por terceros...), relación con el "moroso" (tipo familia, amigos, conocidos...) y por antigüedad (hace más de un mes, más de 3 meses, medio año, etc.)*

Tot i el llenguatge més o menys col·loquial, hi ha aportacions interessants que s'intentarà incloure al disseny. D'altres estaven ja contemplades.

-A més, s'han aconseguit aproximadament 15 adreces de correu electrònic o telèfon que podrien participar en els tests més endavant.

2.1.2.3. CONCLUSIONS

Del resultat combinat de la observació i les enquestes, s'han extret els següents conceptes que han d'ajudar a plantejar l'aplicació:

- No s'han de tenir coneixements específics per utilitzar-la. L'edat del públic objectiu s'espera que sigui de persones adultes, però joves.
- Ha de ser intuïtiva i fàcil d'entendre.
- L'aspecte gràfic és important, tot i que no és el principal.
- La versió d'Android de partida pot ser la 4.0.
- En cas de comercialitzar-se, el model serà gratuït amb publicitat poc invasiva, és a dir, que no interrompi l'aplicació.
- L'aplicació haurà d'incloure les funcionalitats que inclou «Splitwise», tot i que no s'enfocarà tant a despeses de viatge. No es tracta de fer una còpia, sinó d'oferir una millor experiència d'usuari. Per tant, s'han d'oferir funcionalitats diferents o bé més atractives.

- Inicialment l'aplicació es dissenyarà en vertical, amb la possibilitat, si hi ha temps, de permetre l'ús en horitzontal.
- De manera opcional es tractarà d'implementar, tal com es va suggerir a les enquestes, establir límits amb alarmes, per exemple, de massa deutes, massa temps sense pagar/rebre pagament, etc.
- S'inclourà un historial de despeses realitzades.
- Fent cas a una recomanació, es tindrà més cura amb el llenguatge al enviar els missatges preestablerts per Whatsapp. Així mateix, com no es pot confiar en un ús totalment responsable, es tractarà de limitar l'enviament de missatges massa seguits.

2.2. Perfils d'usuari detectats

Dels apartats anteriors s'extreuen tres perfils d'usuari. En els 3 casos tenim perfils semblants, en quant a edat, sexe, professió o estudis, però varia sobre tot la predisposició a fer pagaments en grup, i en segon lloc, la predisposició a utilitzar una aplicació mòbil per registrar-ho. Per tant podem fer la classificació següent dels perfils:

-Usuari sense interès en la aplicació ni en la problemàtica que resol la mateixa.
Es tracta de persones que no participen en pagaments entre grups normalment, i no veuen la necessitat de tenir, per tant, una aplicació específica. Probablement, tampoc utilitzaran una altra aplicació, ni tan sols una agenda. En aquest grup també tenim usuaris no interessats en utilitzar el mòbil per a portar el registre de pagaments/deutes.

-Usuari sense interès a priori en la aplicació.
Es tracta de possibles usuaris que actualment no utilitzen una aplicació per registrar pagaments/deutes, o bé utilitzen aplicacions més senzilles, com els blocs de notes, agenda, o fins i tot porten el registre a paper. A diferència del cas anterior, és un perfil al qual és possible apropar-se, sempre que se li ofereixi una aplicació senzilla i interessant. Es tracta d'un tipus d'usuari que preferirà tenir les configuracions per defecte i no explorar totes les possibilitats de l'aplicació. Es tractarà d'usuaris que estiguin en un grup, però no acostumin a gestionar-lo.

Les principals tasques que durà a terme l'usuari objectiu seran:

- Visualització, creació i gestió de deutes
- Consultes esporàdiques a grups.

-Usuari potencialment interessat en l'aplicació.
Es tracta del tipus d'usuari que explotarà l'aplicació al màxim, utilitzant totes les funcions possibles. És l'usuari que veurà una problemàtica resolta i/o substituirà una aplicació per Debtoid. Possiblement ajudarà a créixer l'aplicació publicitant-la si li agrada. És l'usuari al qual li interessin més les funcionalitats que no pas l'estètica.

A banda de les tasques del cas anterior, les quals durà a terme, les que li

correspondran són:

- Consulta de l'historial de pagaments i de grups
- Enviament de missatges de Whatsapp.
- Creació de grups
- Establiment d'alarmes.

3.Disseny conceptual

A partir dels tres perfils de l'apartat anterior, en aquest apartat i mitjançant tres persones imaginàries, es descriuran els escenaris possibles i els fluxos d'interacció de l'aplicació.

3.1. Persones

Segons el que s'ha vist a l'apartat d'anàlisi, i els tres perfils possibles, es creen tres representacions imaginàries, una per a cada perfil, i se'n descriuen les seves característiques més significatives:

Toni: Té 57 anys. Treballa a una fàbrica de roba.
«A mi el telèfon em serveix per que la família estigui tranquil·la. Això dels correus electrònics i els missatges és una pèrdua de temps.»

Font imatge:

<http://pixabay.com/es/hombre-masculina-blanco-y-negro-140547/>

OBJECTIUS:

No haver de pensar en l'aplicació

Utilitzar el mòbil per emergències i per comunicar-se amb la família.

COMPORTAMENTS:

Es correspon al primer tipus d'usuari, aquell que no està interessat en l'aplicació ni en portar un registre de pagaments a mitges. Es mostra reticent a utilitzar un telèfon mòbil per a registrar qualsevol tipus de dades. Utilitza el Whatsapp eventualment degut a què la seva família l'utilitza, però no li atrau.

No té correu electrònic.

No vol complicar-se ni aprendre a utilitzar noves tecnologies.

Silvia:

37 anys. Administrativa

«M'agrada comunicar-me mitjançant el mòbil i poder prendre notes, però no vull complicar-me massa.

Amb la meua agenda del mòbil no necessito recordatoris, però es podria millorar el funcionament»

Font imatge:

<http://pixabay.com/es/negocios-tel%C3%A9fono-trabajo-mujer-333934/>

OBJECTIUS:



Disposar de informació en tot moment, però no massa detallada
Disposar d'aplicacions senzilles per a registrar dades i pagaments/deutes
No tenir massa aplicacions.

COMPORTAMENTS:

Es correspon amb el segon perfil.

Disposada a utilitzar les noves tecnologies però sense aprofundir-hi.

No li agrada perdre temps configurant aparells.

NECESSITATS:

Necessita una aplicació configurada per defecte, on no hagi de fer ajustos.

Carles:27 anys. Tècnic de xarxes de comunicació

«Acostumo a utilitzar el mòbil, portàtil, tabletas i tot tipus d'aparell al meu dia a dia.

M'agrada configurar i explorar totes les possibilitats dels aparells»

OBJECTIUS:

Portar un registre tan detallat com sigui possible dels deutes/pagaments.

Poder rebre i enviar notificacions, preferiblement per Whatsapp i correu electrònic

Poder administrar grups

Tenir el màxim control possible sobre l'aplicació

COMPORTAMENTS:

Apassionat de la tecnologia

Li agrada configurar i explorar totes les possibilitats que ofereix qualsevol aplicació o sistema. Coneix el que toca.

NECESSITATS:

Disposar del màxim d'informació possible

Disposar d'una aplicació amb moltes possibilitats.



3.2. Escenaris d'ús.

-Escenari 1:

En Carles està amb uns amics i es plantegen regalar-li al Marc pel seu aniversari una sessió d'Spa. En total participen 6 amics i el preu és de 48 euros, per tant hauran de posar 8 euros cadascun. Resulta que no tothom porta diners a sobre, o no porten suficient o no hi ha canvi suficient, així que en Carles decideix utilitzar l'aplicació Debtoid, crea un nou grup, afegint els contactes prèviament emmagatzemats a la seva agenda del mòbil i assigna el deute partit de forma equitativa. Es troba que hi ha un dels amics que participarà però no es troba al lloc, així que també l'afegeix. L'aplicació el notificarà i li demanarà permís per incloure'l. En acabat, tot el grup d'amics té accés al grup dins l'aplicació i tots poden veure que se'ls ha generat un deute, però no veuen l'estat de la resta.

-Escenari 2:

La Sílvia ha quedat amb la Núria per menjar. Al restaurant no admeten xecs restaurant, i no porta diners a sobre. La Núria li paga, però la Sílvia decideix tornar-li els diners, ja que els xecs no li servirien. Per recordar el deute,

l'enregistra a Debtoid, a l'apartat «Deutes propis». La Núria rebrà automàticament una notificació si disposa de l'aplicació, i podrà fer-ne el seguiment i validar-ne el pagament quan el rebi.

-Escenari 3:

La Sílvia ha d'anar al caixer a treure diners per fer alguns pagaments. Recorda que està al grup creat per el Carles i què li devia diners, però no recorda la quantitat. Obre l'aplicació Debtoid i consulta en el grup la quantitat que deu. També veu el dia que es va generar el deute i s'adona de que fa massa temps del deute.

-Escenari 4:

En Joan té l'aplicació Debtoid, ja què li va instal·lar un company de feina per què van jugar una loteria tots junts. Decideix obrir l'aplicació i intentar veure quan li deu al seu company. L'aplicació li retorna un missatge que aconsegueix desxifrar, el significat del mateix és que no té dades i per tant l'aplicació no és capaç de comunicar-se. Haurà d'intentar-ho en un altre moment.

-Escenari 5:

El Joan torna a tenir dades, i casualment el seu company l'ha inclòs a un grup de pagament del qual en Joan havia pagat al moment per un regal de jubilació. En Joan rep un missatge des de l'aplicació indicant-li que està al grup i que ja ha pagat el seu deute.

-Escenari 6: (Semblant al 5)

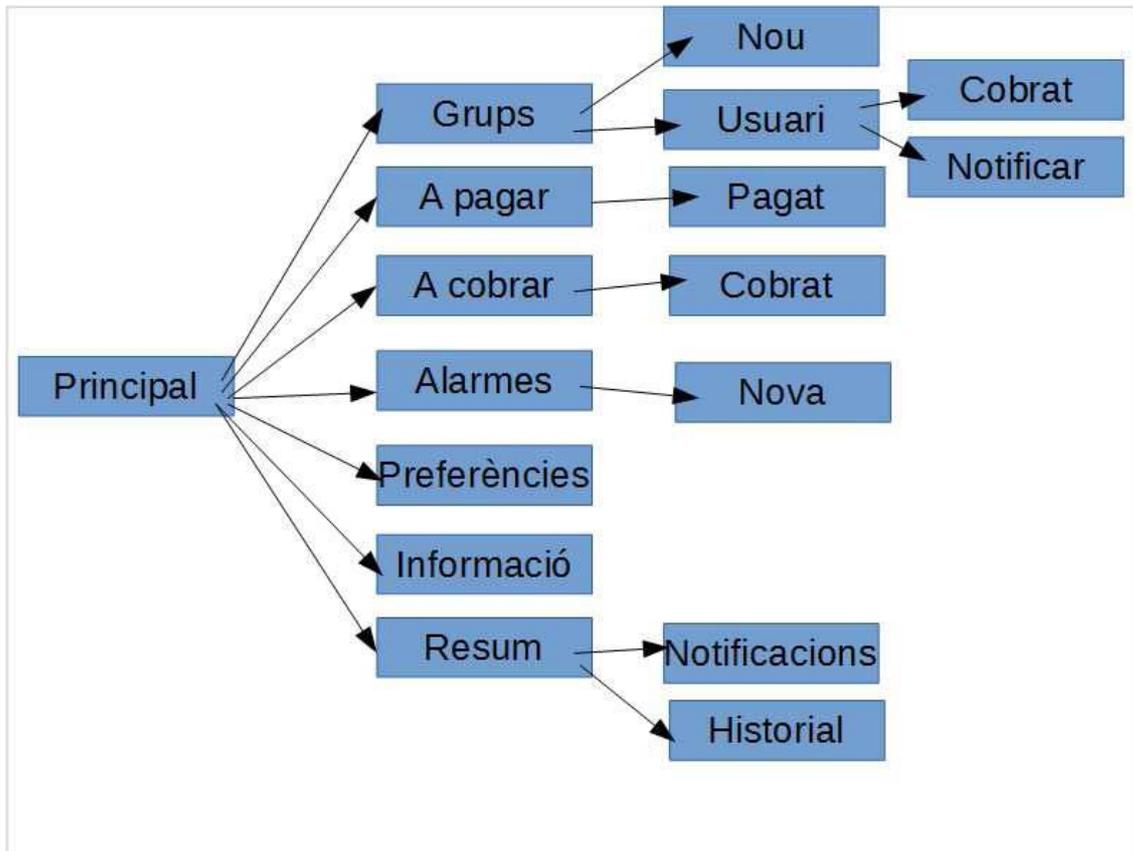
La Sílvia ha sigut inclosa a un grup de pagament, i per tant rep la notificació demanant-li confirmació. Quan la llegeix pot confirmar o cancel·lar, notificació que es rebrà al grup.

-Escenari 7:

En Carles revisa el seu historial i se'n adona que li deu 10 euros a la Maria. Casualment la veurà en dos dies, però com que no sap si se'n recordarà de portar-li el diner, decideix programar una notificació per al mateix dia des de l'aplicació.

3.3. Fluxos d'interacció

Els fluxos d'interacció de l'aplicació dissenyats seguirien una estructura com la que segueix:



A continuació, s'explica en una taula cada interacció i què es pot fer a cada una d'elles.

Opció	Accions disponibles	Repercussions
Principal	a. Seleccionar una de les opcions. b. Es mostra si hi han notificacions al botó de resum. c. Sortir	a. Accedir a les funcions b. Es mostra si hi ha alguna notificació al botó Resum. c. Tancar l'aplicació
Grups	a. Visualització dels grups dels que forma part l'usuari. b. Pitjar Nou. c. Pitjar Usuari. - Seleccionant un usuari en concret.	a. Visualitzar els grups i editar-los. b. Accés a Grups – Nou. c. Accés a Grups – Usuari. d. Torna enrere.

	d. Tornar enrere.	
Grups - Nou	a. Pantalla interactiva. b. Tornar enrere.	a. Creació de nou grup. Envia missatge de confirmació als usuaris afectats. b. Torna enrere.
Grups - Usuari	a. visualitzar dades de l'usuari. b. Pitjar Cobrat. c. Pitjar Notificar. d. Tornar enrere.	a. visualitza amb detall les dades de l'usuari seleccionat b. Marca com a cobrat l'usuari i elimina el deute amb aquest usuari. Bloqueja la opció notificar a aquest usuari. c. Envia missatge segons la opció seleccionada a l'usuari. d. Torna enrere
Grups - Usuari - Cobrat	a. Pantalla demana confirmació. b. Tornar enrere.	a. Confirma pagament, o bé el deixa en el mateix estat. b. Torna enrere
Grups – Usuari - Notificar	a. Pantalla demana confirmació. b. Tornar enrere.	a. Si s'accepta, envia missatge segons es seleccioni a l'usuari. b. Torna enrere.
A pagar	a. Mostra els deutes actuals. b. Pitjar Pagat. c. Tornar enrere.	a. Mostra els deutes . b. Accés a A pagar – Pagat. c. Torna enrere.
A pagar - Pagat	a. Pantalla demana confirmació. b. Tornar enrere.	a. Si es confirma, envia notificació a la persona afectada, que haurà d'acceptar el pagament. b. Torna enrere.
A cobrar	a. Mostra el que es deu a l'usuari. b. Pitjar Cobrat. c. Tornar enrere.	a. Mostra el que es deu a l'usuari. b. Accés a Cobrat. c. Torna enrere.
A cobrar - Cobrat	a. Pantalla demana confirmació. b. Tornar enrere.	a. Si es confirma, envia notificació a la persona afectada. b. Torna enrere.
Alarmes	a. Mostra les alarmes configurades. b. Pitjar Nova.	a. Visualitzar les alarmes configurades i editar-les b. Accés a Nova.

	c. Tornar enrere.	c. Torna enrere.
Alarmes - Nova	a. Crea nova alarma. b. Tornar enrere.	a. Crea nova alarma i la configura. b. Torna enrere.
Preferències	a. Llista de preferències b. Tornar enrere.	a. Mostra el llistat de preferències i en permet l'edició. b. Torna enrere.
Informació	a. Mostra informació b. Tornar enrere.	a. Mostra informació de l'aplicació. b. Torna enrere.
Resum	a. Mostra resum. b. Pitjar Notificacions. c. Pitjar historial. d. Tornar enrere.	a. Mostra les darreres notificacions i un resum de l'aplicació amb un balanç de despeses/pagaments. b. Accés a Notificacions. c. Accés a Historial. d. Torna enrere.
Resum - Notificacions	a. Mostra notificacions. b. Tornar enrere.	a. Mostra les notificacions i permet eliminar-les. b. Torna enrere.
Resum - Historial	a. Mostra historial. b. Tornar enrere.	a. Mostra historial amb detall b. Torna enrere.

Finalment el disseny va variar al no incloure el mode online. No hi havien notificacions i es va afegir un parell de notificacions permanents a la pantalla principal, on es veu el total degut i que es deu.

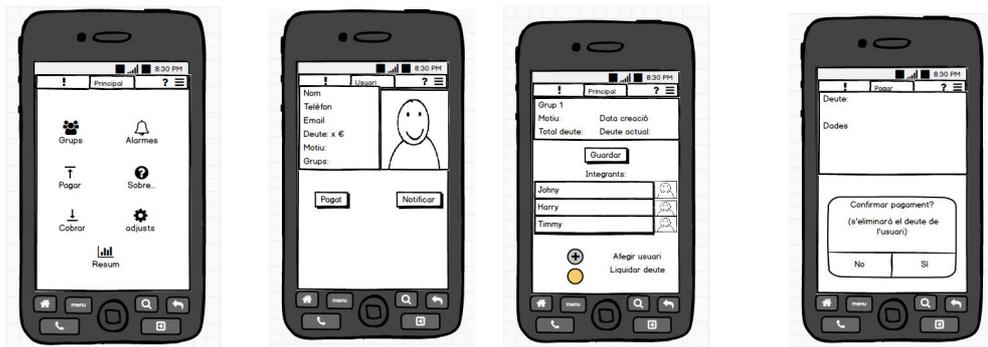
3.4. Disseny:

Es divideix la fase de prototipatge en dos parts. La primera, els Sketches, o esborranys, s'han realitzat amb l'aplicació Balsamiq Mockups, versió de prova. Per a la segona part del prototipatge, s'ha utilitzat Prototyper de Justinmind.

3.4.1. Sketches:

Es tracta d'una visió aproximada de la interfície de l'aplicació per plasmar el flux d'interaccions. En aquesta fase, no s'ha detallat massa les icones ni cap dada.

Exemples d'Sketches:



3.4.2. Prototip horitzontal (alta fidelitat):

Tenint en compte els esborranys realitzats a la fase anterior, es va realitzar un prototip amb més detall sense dades per tenir una idea aproximada del funcionament de l'aplicació.

Exemples de prototip:



3.4.3. Avaluació:

Per tal de realitzar l'avaluació del prototipat de l'aplicació, s'utilitzaria el projecte en HTML generat, se li demanaria a diferents usuaris fer una sèrie de tasques sobre l'aplicació i se li farien una sèrie de preguntes al respecte.

Al tractar-se d'un prototip, i a més en HTML, el funcionament no és real, senzillament és per fer proves de navegació entre menús.

Tot seguit es detallen les proves proposades:

Recull de preguntes:

Com en el cas de l'enquesta inicial, és interessant conèixer el perfil d'usuari al que li fem provar l'aplicació. Tot i que no n'excloem cap perfil, intentem continuar afitant el nostre "target".

Ets usuari d'Android? De quina versió?

Ens interessa conèixer si l'usuari coneix la interfície d'Android i no tindrà problemes amb els menús de l'aplicació.

Utilitzes algun tipus d'aplicació per prendre notes, anotar deutes...? Quina aplicació?

Com a l'enquesta inicial, ens interessa conèixer si els usuaris utilitzen aplicacions enfocades a registrar deutes, pagaments pendents, etc.

Utilitzes Whatsapp habitualment?

Degut a que l'aplicació enviarà missatges mitjançant Whatsapp, és interessant saber si l'usuari està acostumat.

Tasques que s'haurien de realitzar per part dels usuaris:

Les proves es basarien en fer a l'usuari que sense explicar el funcionament de l'aplicació, però explicant-li la finalitat de la mateixa, identifiqui les funcions i faci una explicació breu de per què creu que serveix cada opció. Un cop fet això, se li demanaria que realitzés (simuladament) les següents tasques:

- Crear un grup.
- Visualitzar un usuari d'un grup concret
- Modificar una alarma.
- Afegir una alarma.
- Revisar els cobraments pendents
- Donar per pagat el deute d'un usuari.
- Enviar una notificació a un usuari
- Revisar els pagaments pendents.
- Visualitzar la informació de l'aplicació.
- Canviar una preferència.
- Veure l'historial de moviments.

Un cop fetes les tasques demanades, se li plantejarien a l'usuari les següents qüestions, per tal de tornar a recomençar tot el procés amb possibles millores.

T'ha resultat difícil fer alguna de les tasques? Si es que sí, quina i per què?

Es tracta de millorar la percepció de l'usuari respecte a la usabilitat de l'aplicació en funció del que l'usuari ens digui sobre la mateixa.

Et sembla que s'hauria d'incloure més informació o menys a l'aplicació?

Tot i que no s'ha reflectit tota la informació al prototip, és important saber si l'usuari vol tenir més dades o si li sembla que hi ha massa.

Creus necessari un camp amb informació o ajuda a cada apartat?

En alguns casos, els usuaris poden no entendre l'aplicació, de ser així, una possible correcció seria aquesta.

Et semblen adequades les icones? T'agradaria l'aplicació amb més color?

En aquesta versió del prototip no s'han afegit les icones finals, ni color de fons, per exemple. En tot cas, val la pena conèixer la opinió de l'usuari del disseny general en cada iteració.

Et sembla útil l'aplicació per a la finalitat que se t'ha explicat?

Se li pregunta a l'usuari aquesta qüestió, per tal d'esbrinar si l'aplicació li sembla útil.

Afegiries o trauries alguna funcionalitat?

Aquesta pregunta és gairebé obligada. El feedback és sempre important, i sempre podem trobar idees que no hem tingut o valorat prèviament.

4.Implementació

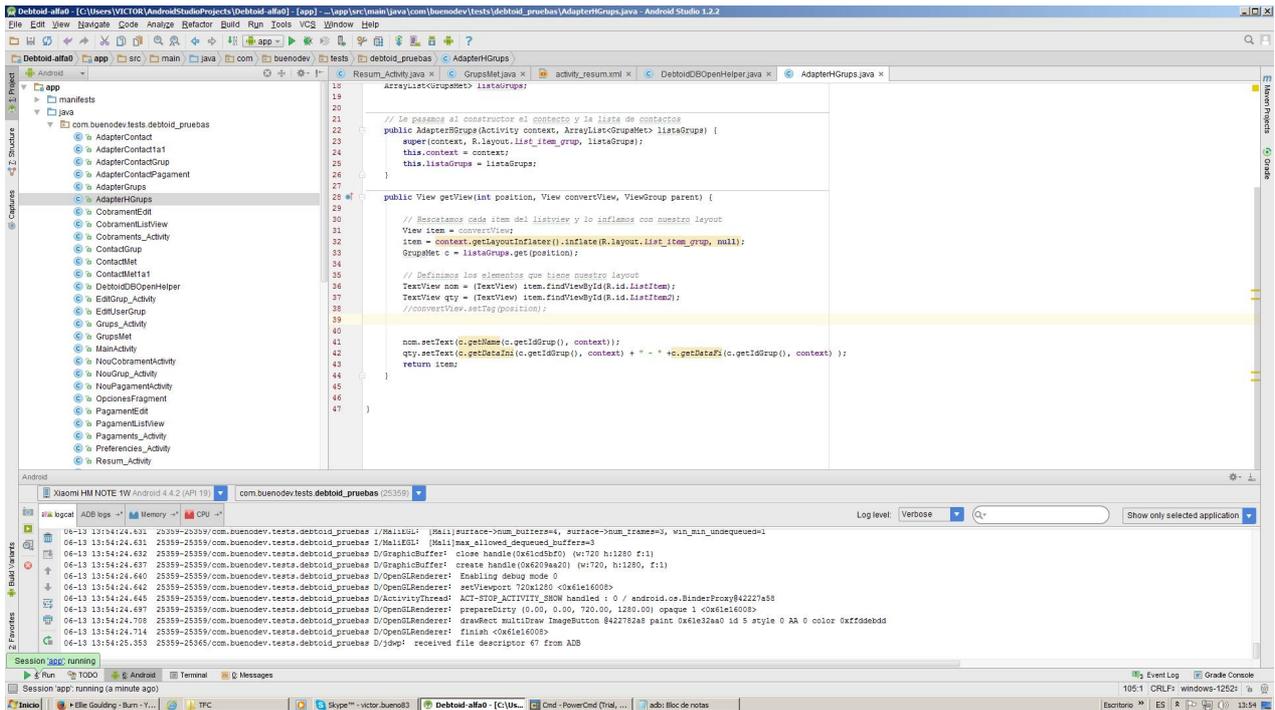
En aquest apartat es tracta com s'ha realitzat la implementació de l'aplicació, amb quines eines, i com funciona l'aplicació.

4.1.Eines utilitzades i justificació

4.1.1. Android Studio

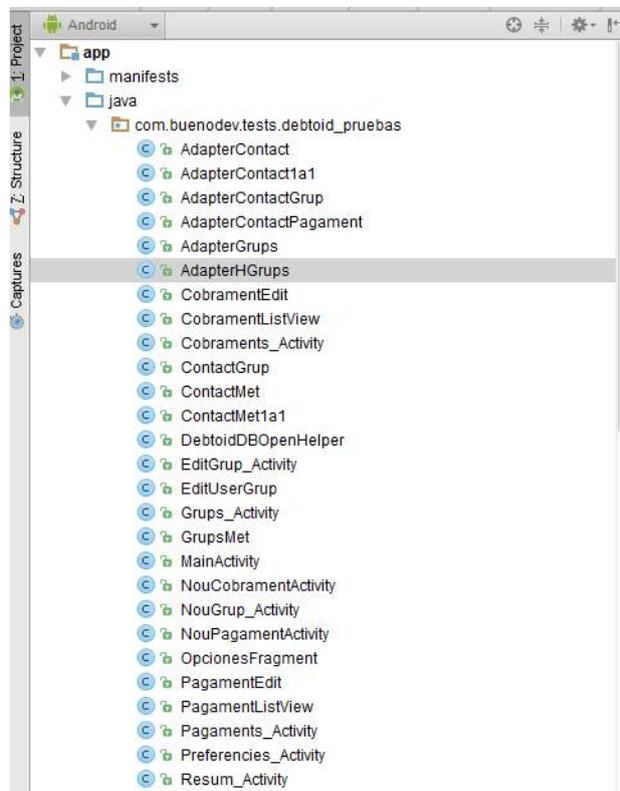
He optat per aquest IDE degut a la bona fama que ha anat adquirint en els darrers mesos. Des de la pàgina d'Android es recomana utilitzar aquest IDE. Tot i tenir una experiència mínima amb Eclipse, un cop adaptat a Android Studio, no es fa en falta cap element. De fet Android Studio proporciona algunes eines extres com, per exemple, visualitzar les interfícies en diferents configuracions de dispositiu totes alhora.

Android Studio incorpora sota un mateix framework totes les eines que normalment es necessiten per desenvolupar aplicacions en Android. Actualment és el IDE al qual Google dona suport, i sobre el que més documentació es pot trobar.



La pantalla es divideix en diferents parts, i es pot configurar gairebé tot. Les parts més importants són:

-La estructura del projecte o d'aplicació: Permet accedir als arxius sobre els que treballa l'eina, també mostra segons l'iconesi es tracta d'arxius de tipus vista o de classes Java:



-La pantalla amb l'arxiu actual. Mostra el contingut de l'arxiu sobre el que s'està treballant en el moment. En el cas dels arxius de vistes, mostra també una simulació a una pantalla virtual.

```
Resum_Activity.java x GrupsMet.java x activity_resum.xml x DebtoidDBOpenHelper.java x AdapterHGrups.java x
18 ArrayList<GrupsMet> listaGrups;
19
20
21 // Le pasamos al constructor el contexto y la lista de contactos
22 public AdapterHGrups(Activity context, ArrayList<GrupsMet> listaGrups) {
23     super(context, R.layout.list_item_grup, listaGrups);
24     this.context = context;
25     this.listaGrups = listaGrups;
26 }
27
28 public View getView(int position, View convertView, ViewGroup parent) {
29
30     // Rescatamos cada item del listview y lo inflamos con nuestro layout
31     View item = convertView;
32     item = context.getLayoutInflater().inflate(R.layout.list_item_grup, null);
33     GrupsMet c = listaGrups.get(position);
34
35     // Definimos los elementos que tiene nuestro layout
36     TextView nom = (TextView) item.findViewById(R.id.ListItem);
37     TextView qty = (TextView) item.findViewById(R.id.ListItem2);
38     //convertView.setTag(position);
39
40
41     nom.setText(c.getName(c.getIdGrup(), context));
42     qty.setText(c.getDataIni(c.getIdGrup(), context) + " - " + c.getDataFi(c.getIdGrup(), context) );
43     return item;
44 }
45
46
47 }
```

Com en la majoria de Frameworks actuals, es disposa de moltes opcions, com ara refactoritzar el codi, canviar color d'alguns elements (variables, constants, funcions, classes), o una de les més útils, que l'IDE faci les importacions de les classes necessàries.

-La consola de depuració. És un element molt útil al provar l'aplicació. Ens mostra dades en temps real del que està fent l'aplicació, si hi ha un error crític ens mostra la raó, i es pot utilitzar per a afegir registres de les funcions que va realitzant l'aplicació.

```
Android
Xiaomi HM NOTE 1W Android 4.4.2 (API 19) com.buendev.tests.debtoid_pruebas (25359)
logcat ADB logs -> Memory -> CPU ->
06-13 13:54:24.631 25359-25359/com.buendev.tests.debtoid_pruebas I/MaliEGL: [Mali]surface->num_buffers=4, surface->num_frames=3, win_min_underequed=1
06-13 13:54:24.631 25359-25359/com.buendev.tests.debtoid_pruebas I/MaliEGL: [Mali]max_allowed_dequeued_buffers=3
06-13 13:54:24.632 25359-25359/com.buendev.tests.debtoid_pruebas D/GraphicBuffer: close handle(0x61cd5bf0) (w:720 h:1280 f:1)
06-13 13:54:24.637 25359-25359/com.buendev.tests.debtoid_pruebas D/GraphicBuffer: create handle(0x6209aa20) (w:720, h:1280, f:1)
06-13 13:54:24.640 25359-25359/com.buendev.tests.debtoid_pruebas D/OpenGLRenderer: Enabling debug mode 0
```

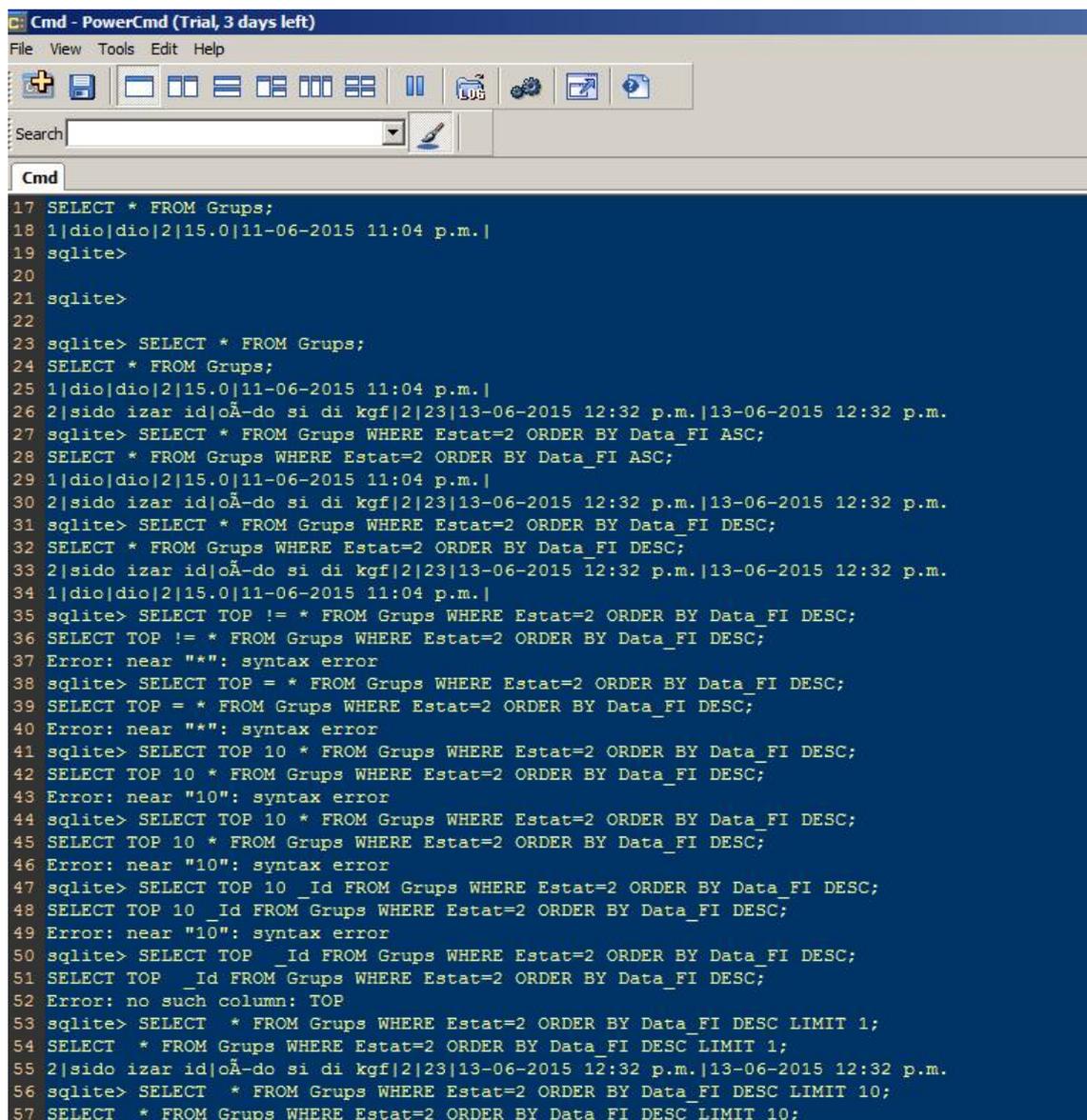
4.1.2. Inkscape

S'ha utilitzat aquesta eina tot i tenir-ne pocs coneixements de disseny gràfic pel fet de ser lliure i gratuïta. El seu ús ha sigut per dissenyar el logo de l'aplicació i fer proves de colors pels fons de l'aplicació, els colors de lletra, icones, etc. Tot i que es pot millorar molt el disseny, s'ha buscat un esquema de colors no massa cridaner, i que recordés el color típic dels bitllets al tractar-se d'una aplicació per registrar deutes.

Inkscape és una aplicació bastant coneguda i, tot i no tenir coneixements previs, es pot trobar documentació fàcilment.

4.1.3. PowerCMD

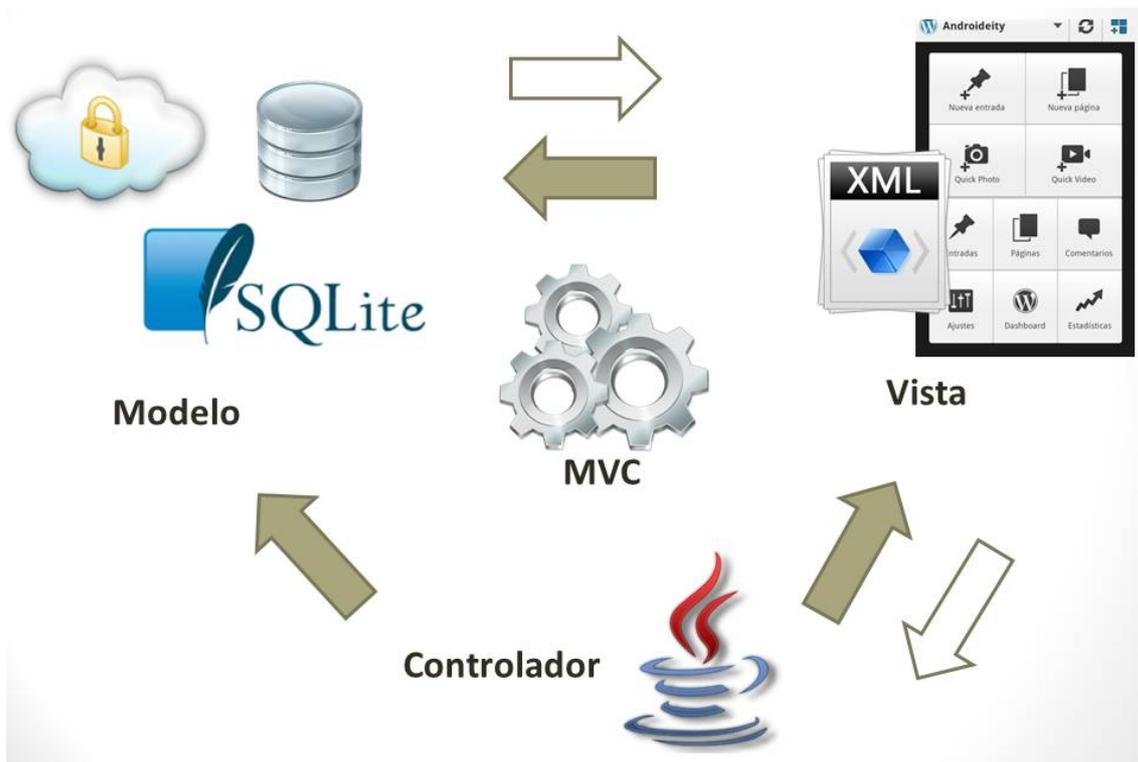
Tot i no ser una eina imprescindible, ha facilitat molt algunes tasques. Es tracta d'una consola de comandes que s'integra sobre la pròpia de Windows, afegint funcionalitats. S'ha utilitzat per connectar des del PC a la base de dades de l'aplicació al mòbil i fer-ne comprovacions.



```
Cmd - PowerCmd (Trial, 3 days left)
File View Tools Edit Help
Search
Cmd
17 SELECT * FROM Grups;
18 1|dio|dio|2|15.0|11-06-2015 11:04 p.m.|
19 sqlite>
20
21 sqlite>
22
23 sqlite> SELECT * FROM Grups;
24 SELECT * FROM Grups;
25 1|dio|dio|2|15.0|11-06-2015 11:04 p.m.|
26 2|sido izar id|oÃ-do si di kgf|2|23|13-06-2015 12:32 p.m.|13-06-2015 12:32 p.m.
27 sqlite> SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI ASC;
28 SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI ASC;
29 1|dio|dio|2|15.0|11-06-2015 11:04 p.m.|
30 2|sido izar id|oÃ-do si di kgf|2|23|13-06-2015 12:32 p.m.|13-06-2015 12:32 p.m.
31 sqlite> SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
32 SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
33 2|sido izar id|oÃ-do si di kgf|2|23|13-06-2015 12:32 p.m.|13-06-2015 12:32 p.m.
34 1|dio|dio|2|15.0|11-06-2015 11:04 p.m.|
35 sqlite> SELECT TOP != * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
36 SELECT TOP != * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
37 Error: near "!=": syntax error
38 sqlite> SELECT TOP = * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
39 SELECT TOP = * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
40 Error: near "=": syntax error
41 sqlite> SELECT TOP 10 * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
42 SELECT TOP 10 * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
43 Error: near "10": syntax error
44 sqlite> SELECT TOP 10 * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
45 SELECT TOP 10 * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
46 Error: near "10": syntax error
47 sqlite> SELECT TOP 10 _Id FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
48 SELECT TOP 10 _Id FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
49 Error: near "10": syntax error
50 sqlite> SELECT TOP _Id FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
51 SELECT TOP _Id FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC;
52 Error: no such column: TOP
53 sqlite> SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC LIMIT 1;
54 SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC LIMIT 1;
55 2|sido izar id|oÃ-do si di kgf|2|23|13-06-2015 12:32 p.m.|13-06-2015 12:32 p.m.
56 sqlite> SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC LIMIT 10;
57 SELECT * FROM Grups WHERE Estat=2 ORDER BY Data_FI DESC LIMIT 10;
```

4.2. Estructura de Debtoid

Debtoid s'ha desenvolupant utilitzant codi Java i arxius XML. S'utilitza una base de dades de tipus SQLite creada i manipulada al propi dispositiu. S'ha fet servir el patró MVC, o Model-Vista-Controlador degut a la estructura d'arxius que típicament genera Android Studio.



Font: <http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/>

Estructurant els arxius segons MVC, el que es fa amb els arxius és dividir-los en tres categories diferents, segons es tracti d'arxius de model-lat, de vista o de controlador. Cada tipus d'arxiu té una funció lògica:

-Arxius de model-lat: Aquests arxius estableixen els mètodes mitjançant els quals es crea, es consulten, s'insereix o modifiquen registres a la base de dades. En el cas de Debtoid s'han realitzat tots els mètodes a una única classe.

-Arxius de vista: Són els arxius que defineixen el que es mostra al dispositiu i alhora permet a l'usuari introduir dades o canviar a un altre menú. En el cas de Android es sol tractar d'arxius XML. Normalment un o més per cada pantalla diferent, dependent de si una vista n'inclou d'altres o no. En el cas de Debtoid, es pot veure als llistats de deutes, per exemple, com dins d'una vista es repeteixen altres vistes amb dades diferents.

-Arxius de controlador: Es tracta dels arxius que com el seu nom indiquen, controlen el funcionament de l'aplicació. Són, de fet, la part més complexa de

l'aplicació.

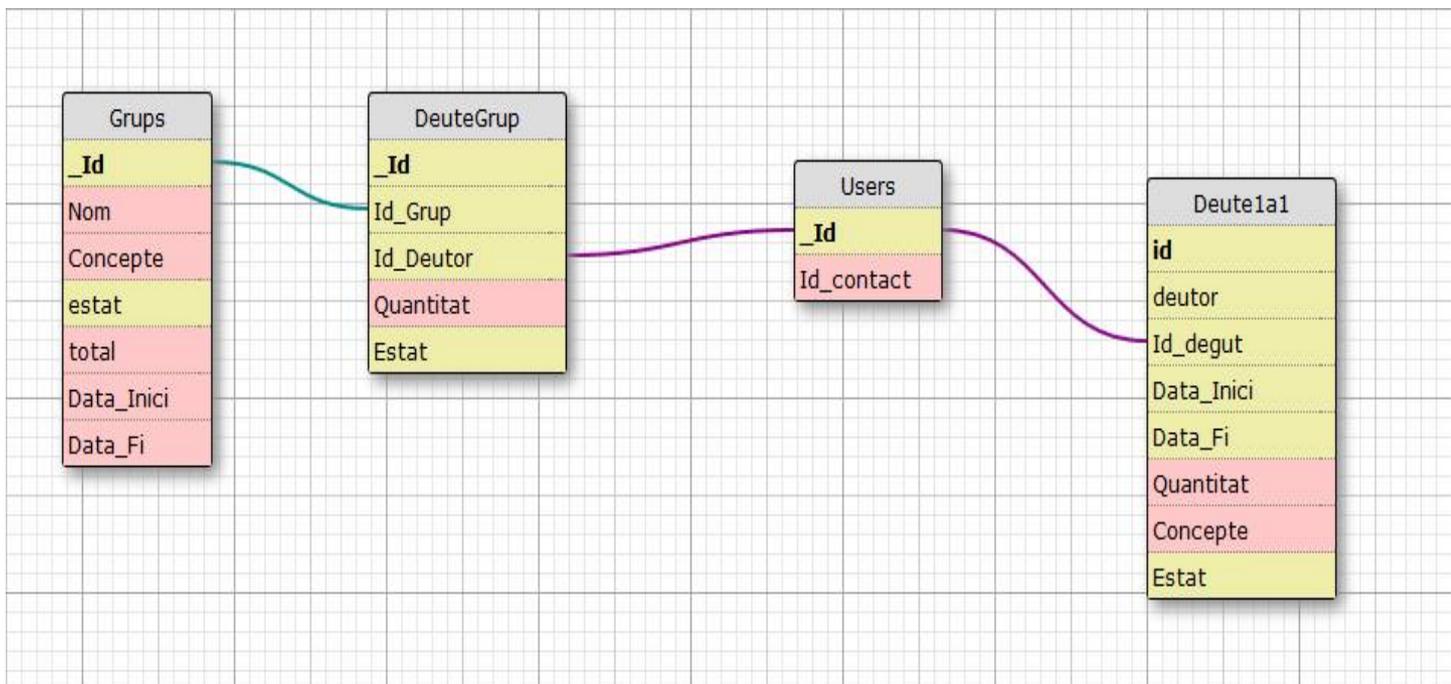
Per posar un exemple fora del context d'Android sobre el funcionament de MVC, suposem que entrem al nostre Email mitjançant una web. La web sobre la que veiem les dades seria la vista, o conjunt de vistes. Les dades serien el model, probablement allotjades en una base de dades. En darrer lloc, els controladors, serien tots els arxius amb la codificació que fa possible que veiem els correus, que n'enviem de nous. Etc.

4.2.1. La base de dades de Debtoid

Debtoid utilitza com a Base de dades un arxiu de tipus SQLite. Aquest arxiu es pot llegir des del terminal o des de fora del mateix, sempre que tinguem privilegis d'administrador al dispositiu.

SQLite és un sistema bastant senzill d'emmagatzemament, basat en SQL. SQL és bastant conegut pels programadors d'aplicacions en general. El seu funcionament a partir de taules de dades, permet inserir, editar, esborrar o visualitzar dades seguint una codificació concreta.

L'estructura de la base de dades és la següent:



Tal com es pot veure, existeixen quatre taules:

-Users: Emmagatzema les dades necessàries per mostrar a pantalla de cada usuari. Consta de dos camps: `_Id` i `Id_contact`. `Id` serveix d'índex. `Id_contact` emmagatzema el valor que necessita l'aplicació per trobar al dispositiu el contacte.

-Deute1a1: Emmagatzema les dades corresponents a cada deute individual, és a dir, de l'usuari propietari del telèfon a un contacte del mateix. S'emmagatzemen tant els pagaments com els cobraments. Part de l'històric i del deute/degut total també s'extreu d'aquí. Consta de vuit camps:

`Id`: Funciona com a índex intern de la base de dades.

`Deutor`: Serveix al controlador per saber si és un pagament o un cobrament.

`Id_degut`: Indica el `_Id` corresponent a la taula Users.

`Data_Inici`: Indica la data en què es va establir el deute

`Data_Fi`: Indica la data en què es va finalitzar el deute

`Quantitat`: Quantitat deguda

`Concepte`: Concepte pel qual es deu la quantitat.

`Estat`: Indica si el deute s'ha saldat ja o encara és actiu.

-Grups: Emmagatzema la informació relativa a cada grup. Part de l'històric i del deute/degut total també s'extreu d'aquí. Consta de set camps:

`_Id`: Funciona com a índex intern de la base de dades.

`Nom`: Nom del grup

`Concepte`: Explicació del motiu del grup.

`Estat`: Indica si el deute encara existeix.

`Total`: Indica la quantitat deguda al grup.

`Data_Inici`: Indica la data d'inici del deute de grup

`Data_Fi`: Indica la data en què es va finalitzar el deute.

-DeuteGrup: Emmagatzema la informació de cada usuari que participa al deute dins d'un grup. Consta de cinc camps:

`_Id`: Funciona com a índex intern de la base de dades.

`Id_Grup`: Indica a quin grup pertany el deute.

`Id_Deutor`: Indica quin és l'usuari deutor, dels de la taula Users.

`Quantitat`: Quantitat que deu l'usuari a dins del grup.

`Estat`: Indica si s'ha saldat el deute o bé si encara és actiu.

A banda de definir la base de dades d'una forma senzilla, s'ha tingut en compte als controladors que s'ha de mantenir la consistència de la mateixa. Un exemple clar és el fet de què un deutor d'un grup no pot introduir un deute major al del grup sencer mai.

4.2.2. Model de l'aplicació

Com s'ha esmentat, el modelat de les dades de l'aplicació es realitza sobre una única classe Java anomenada DebtoidDBOpenHelper. En aquesta es defineixen els mètodes que actuen sobre la base de dades. Per exemple, podem veure com es creen algunes de les taules al iniciar l'aplicació per primer cop:

```
//Definició de taula Deute1a1
private static final String TABLE_Deute1a1 = "Deute1a1";
public static final class Deute1a1 implements BaseColumns {
    private Deute1a1() {}
    public static final String ID_Deute1a1 = "_Id";
    public static final String Deutor = "deutor";
    public static final String ID_Degut = "Id_degut";
    public static final String Data_inici = "Data_inici";
    public static final String Data_Fi = "Data_Fi";
    public static final String Quantitat = "Quantitat";
    public static final String Concepte = "Concepte";
    public static final String Estat = "Estat";
}

//Definició de taula DeuteGrups
private static final String TABLE_DeuteGrup = "DeuteGrup";
public static final class DeuteGrup implements BaseColumns {
    private DeuteGrup() {}
    public static final String ID_DeuteGrups = "_Id";
    public static final String Id_Grup = "Id_Grup";
    public static final String Id_Deutor = "Id_Deutor";
    public static final String Quantitat = "Quantitat";
    public static final String Estat = "Estat";
}
```

Tal com es pot veure a les imatges, es defineixen per a cada taula els camps que hem vist abans.

Un altre exemple de com s'implementen les consultes (queries) a la base de dades és el següent:

```
public static void insertDeuteGrup(SQLiteDatabase db, String IdGrup, String Iddeutor, String quantitat) { //Insert
    ContentValues nuevoRegistro = new ContentValues();
    nuevoRegistro.put(DeuteGrup.Id_Grup, IdGrup);
    nuevoRegistro.put(DeuteGrup.Id_Deutor, Iddeutor);
    nuevoRegistro.put(DeuteGrup.Quantitat, Double.parseDouble(quantitat));
    nuevoRegistro.put(DeuteGrup.Estat, "1");

    db.insert(TABLE_DeuteGrup, null, nuevoRegistro);
}
```

En el qual es pot veure una funció que insereix un nou deute amb els paràmetres rebuts per el controlador.

En el cas d'aquesta classe, s'han utilitzat dos mètodes diferents per realitzar les consultes en funció de si es modificaven o s'insertien noves dades o bé si es feia una consulta sobre dades ja existents.

En el primer cas, s'ha utilitzat el sistema vist a la imatge anterior, on la consulta SQL la construeix Android Studio a partir dels paràmetres que se li subministren a la funció. En el cas de les consultes, s'ha construït cada consulta i se li passa ja construïda a Android Studio per què l'executi:

```
public static Cursor getDadesCobrament(SQLiteDatabase db, String ordre) { //Mètode que torna els IDs dels  
  
    Cursor c = db.rawQuery("SELECT Id_contact FROM Users u, Deute1al d WHERE u._Id = d.Id_degut " +  
        "AND d.Estat = 1 AND d.deutor=1 GROUP by u.Id_contact ORDER BY "+ ordre , null);  
  
    return c;  
}
```

Tot i què és millor costum realitzar les consultes com en el primer cas, al fer proves ha sigut més senzill utilitzar les consultes construïdes, al haver de utilitzar la consola de comandes. També, degut a què algunes consultes són més complicades, s'ha optat per deixar-les en aquest format.

Java té una classe anomenada Cursor, serveix per retornar resultats de les consultes, s'ha utilitzat en la majoria dels casos per poder manipular les ades més fàcilment al controlador.

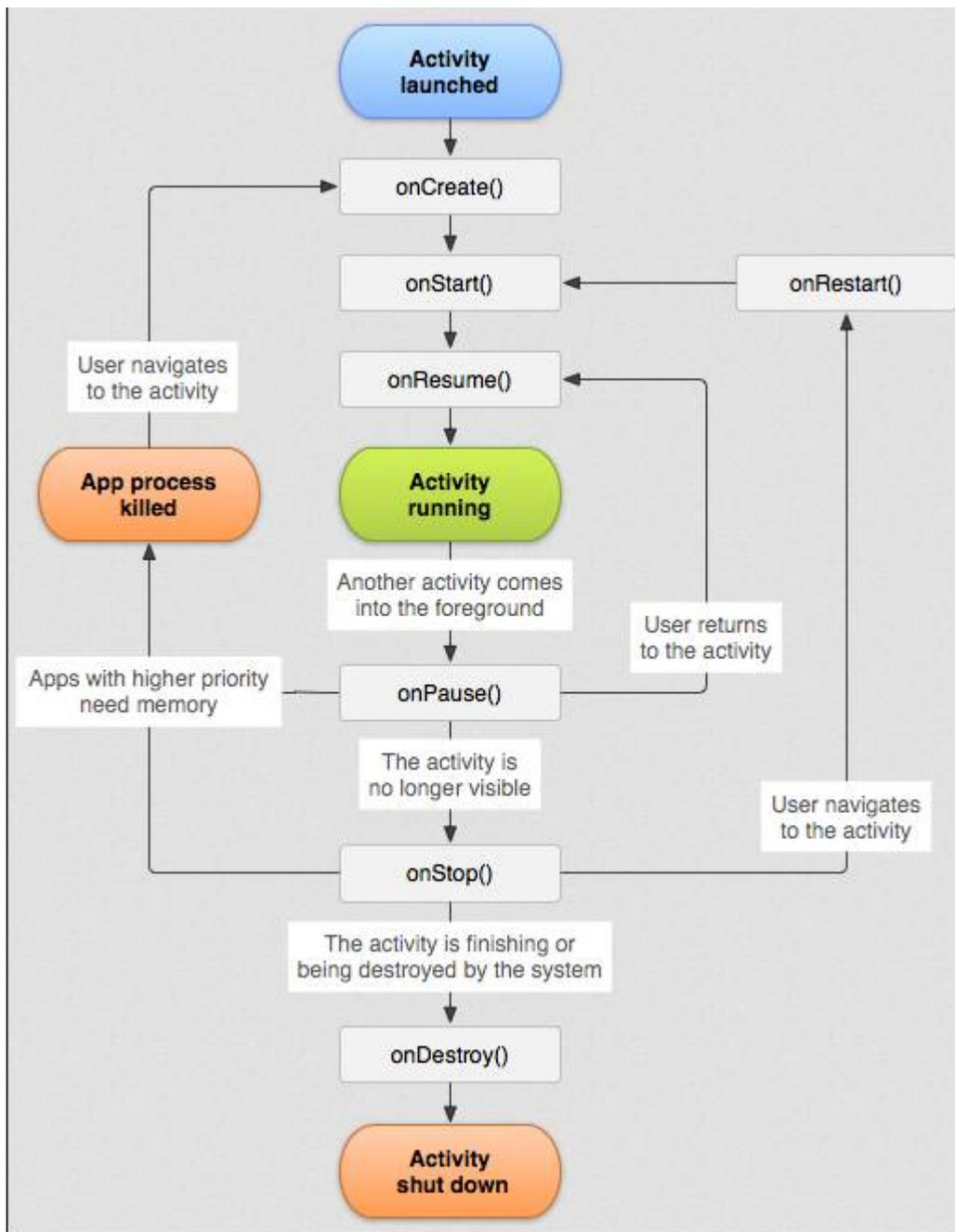
4.2.3. Controlador de l'aplicació

Com s'ha comentat, la part més extensa en quant a codi de l'aplicació, és possiblement la part dels controladors. S'explicarà en aquest apartat alguns dels més importants dins de l'aplicació. Abans d'això, i per entendre millor el funcionament d'algunes parts de l'aplicació, és important conèixer què és una activity i com funciona:

Una Activity és el conjunt de la vista i el controlador que la carrega i permet la interacció. És a dir, una vista sense el seu controlador, no tindria gaire utilitat, tot i que un controlador sense vista, sí que en pot tenir, com veurem en el cas dels «adapters». Col·loquialment una Activity es correspon amb una «pantalla».

Respecte al cicle de vida de les activitats, s'ha de tenir en compte, ja que algunes funcions s'executen solament en determinats moments. Per exemple, si anem a un menú concret, fem un canvi, i tornem al menú inicial, aquest podria necessitar recarregar les dades que es mostren a pantalla ja que aquests han canviat.

Com a resum, la següent imatge indica els principals estats d'una Activity:



Font: <http://developer.android.com/reference/android/app/Activity.html>

El cicle de vida és el següent:

- Es crida a l'Activity, aquesta es crea, mitjançant el mètode onCreate().
- S'inicia l'activitat i en cas d'haver estat pausada es resumeix, amb les dades que hi haguessin carregades.
- L'Activity es considera en execució.
- Si una altra Activity s'executa, l'actual quedarà pausada (onPause()) i posteriorment onStop().

-En cas de què el sistema necessiti recursos, pot alliberar l'Activity pausada destruint-la. També es destrueix al finalitzar-la.

Tenint en compte el funcionament de les Activities, podem examinar alguns dels controladors més importants. No s'explicaran tots, ja que el funcionament és bastant semblant en alguns casos:

-MainActivity: Es tracta del primer controlador que es crida al iniciar l'aplicació. Aquest crida a l'Activity -interfície a mostrar- i carrega els elements de la vista associada. La classe carrega per defecte unes preferències que es poden modificar.

Aquesta classe té dos mètodes que fan ús de la classe de model, per tal de mostrar el deute i el degut totals:

```
//Carrego les dades extretes d'una consulta a la base de dades per tenir la suma total del deute i del que es deu
pref.registerOnSharedPreferenceChangeListener(this);
TextDeus.setText("Deus en total: " + DebtoidDBOpenHelper.getPagaTotal(db) + " " + money);
TextDec.setText("Se't deu en total: " + DebtoidDBOpenHelper.getCobraTotal(db)+" " +money);
```

En aquest fragment de codi el que es fa és indicar al IDE que a les caixes de text indicades les hi afegeixi la frase «Deus en total:» seguida de la quantitat extreta de la base de dades i del tipus de moneda establert a l'aplicació.

Per cridar a altres Activities, s'utilitzen els anomenats Listeners (escoltadors):

Aquests resten sense fer res, però en el moment en què detecten que s'ha préss un botó, actuen segons la seva programació.

Un cas especial és el del botó alarmes, el qual llença l'aplicació del calendari d'Android en lloc d'una Activity pròpia de l'aplicació.

Tal i com es veu a la imatge, existeix un altre element, l'Intent. Aquest serveix per cridar una Activity, un Servei, accions en segon plànol, proveïdors de contingut i Broadcast Receivers. En el cas de Debtoid no s'han implementat aquestes funcions.

-Grups_Activity: Es tracta de l'Activity llençada des de MainActivity i en la que es carreguen i mostren les dades dels grups de deutes. Fa ús de les classes AdapterGrups i GrupsMet per omplir el llistat de deutes. Això és degut a què el llistat s'ha d'omplir mitjançant un tipus de vector o array amb les dades definides d'una forma no estàndard, és a dir, cada aplicació que vulgui personalitzar un llistat, necessita canviar el tipus d'adapter que utilitza amb els seus llistats. Mitjançant una consulta a la base de dades troba els grups actius, i fa una crida a AdapterGrups per cada grup, amb la fi de crear el llistat.

Permet cridar a l'Activity NouGrupActivity per crear un nou grup, o a la EditGrupActivity, per editar un grup en concret.

-AdapterGrups: Aquesta classe utilitza (estén) la classe GrupsMet. El codi és el següent:

```
public class AdapterGrups extends ArrayAdapter<GrupsMet> {  
  
    Activity context;  
    ArrayList<GrupsMet> listaGrups;  
  
    public AdapterGrups(Activity context, ArrayList<GrupsMet> listaGrups) {  
        super(context, R.layout.list_item_grup, listaGrups);  
        this.context = context;  
        this.listaGrups = listaGrups;  
    }  
  
    public View getView(int position, View convertView, ViewGroup parent) {  
  
        View item = convertView;  
        item = context.getLayoutInflater().inflate(R.layout.list_item_grup, null);  
        GrupsMet c = listaGrups.get(position);  
  
        TextView nom = (TextView) item.findViewById(R.id.ListItem);  
        TextView qty = (TextView) item.findViewById(R.id.ListItem2);  
  
        nom.setText(c.getName(c.getIdGrup(), context));  
        qty.setText(c.getQuantit(c.getIdGrup(), context));  
        return item;  
    }  
}
```

El seu funcionament es basa en retornar una vista (View) per cada element trobat a la classe que la crida, en aquest cas, Grups_Activity. Utilitza la classe GrupsMet per carregar les dades de cada element de la vista associada a cada valor.

-GrupsMet: Aquesta classe conté els mètodes per extreure les dades per cada ítem a dins del llistat dels grups. Un fragment del seu codi és el següent:

```

public static String getName(String IdGrup, Context context) { //Mètode

    String nom = "DATO";

    final DebtoidDBOpenHelper usdbh =
        new DebtoidDBOpenHelper(context, "DBDebtoid", null, 1);
    final SQLiteDatabase db = usdbh.getWritableDatabase();

    final Cursor cur = DebtoidDBOpenHelper.getNomGrup(db, IdGrup);
    for (cur.moveToFirst(); !cur.isAfterLast(); cur.moveToNext()) {
        nom = cur.getString(0);
    }
    return nom;
}

```

Com es pot veure, la classe rep l'Identificador del grup en qüestió, i un paràmetre, context, necessari per la funció per establir a quina Activity és. Mitjançant l'Identificador del grup, es pot extreure de la base de dades el nom del grup i tornar-lo a la classe que el demana, en aquest cas, AdapterGrups.

El procediment és semblant per a la resta de les dades a extreure.

Aquest conjunt de adapterX – Xmet s'ha utilitzat en diferents llistats de l'aplicació, degut a què mostren diferent informació. L'estructura és bastant semblant en tots ells, amb petites variàncies.

-Cobraments_Activity: Aquesta classe és bastant semblant a la de grups, i a la de pagaments, serveix per a llistar els cobraments pendents i ofereix la possibilitat de crear-ne de nous.

-NouCobramentActivity: Aquesta classe permet crear un nou cobrament. Inicialment carrega la vista associada i deixa els espais en buit. Observem un fragment més peculiar de codi a la pàgina següent. En ell, es pot veure com es crida a una Activity de sistema per obtenir com a resultat un paràmetre anomenat URI. URI és semblant a una adreça HTML, que en aquest cas ens indica el valor per buscar un contacte en concret a la base de dades de contactes d'Android.

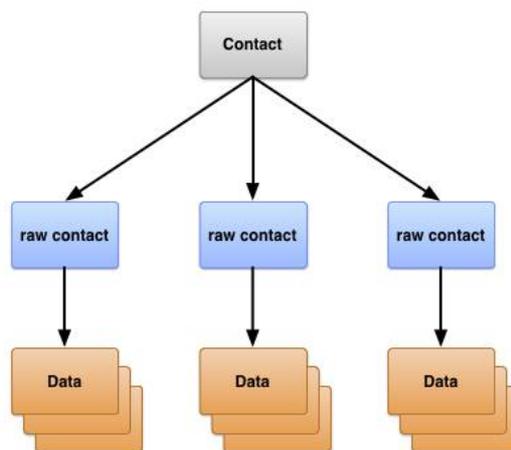
```
public void initPickContacts(View v) {
    // Intent per seleccionar contacte
    Intent i = new Intent(Intent.ACTION_PICK, ContactsContract.Contacts.CONTENT_URI);
    // Iniciar la activitat esperant resposta mitjançant el canal PICK_CONTACT_REQUEST
    startActivityForResult(i, PICK_CONTACT_REQUEST);
}

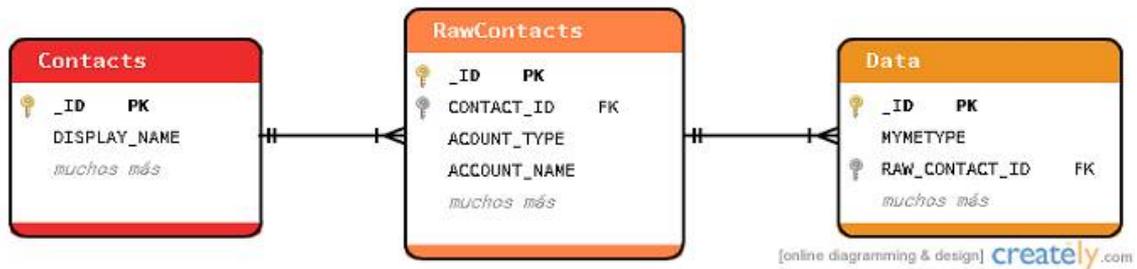
protected void onActivityResult(int requestCode, int resultCode, Intent intent) { //Al rebre resultat del

    super.onActivityResult(requestCode, resultCode, intent);
    if (requestCode == PICK_CONTACT_REQUEST) {
        if (resultCode == RESULT_OK) {
            //Capturar el valor de la Uri
            contactUri = intent.getData();
            //Processar la Uri
            Context context = this.getApplicationContext();
            contactText.setText(ContactMet.getName(contactUri.toString(), context)); //Mitjançant la
            imageContact.setImageBitmap(ContactMet.getPhoto(contactUri.toString(), context));
        }
    }
}
```

La base de dades de contactes d'Android és bastant més complexa que la de Debtoid, per aquest motiu, solament emmagatzem la URI de cada contacte amb un deute.

Segons la documentació oficial, el proveïdor de continguts dels contactes, estructura a aquests de la següent manera:





Tal com es pot veure, cada contacte té diferents taules amb camps sense especificar per trobar les dades. Per aquest motiu, s'utilitzen llibreries d'Android com per exemple CommonDataKinds, per accedir al telèfon, correu electrònic, adreces, etc d'un contacte.

-CobramentEdit: Aquesta Activity ens permet canviar les dades d'un cobrament, donar-lo per pagat, o enviar un missatge de Whatsapp a un usuari.

Alguns fragments interessants de codi són els següents:

```

notibutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        PackageManager pm=getPackageManager();
        try {

            Intent waIntent = new Intent(Intent.ACTION_SEND);
            waIntent.setType("text/plain");
            String text = "---Enviat des de DEBTOID--- Recordatori de deuta:" +
                " " + TextQuantitat.getText() + " en concepte de " +
                TextConcepte.getText();

            PackageInfo info=pm.getPackageInfo("com.whatsapp", PackageManager.GET_META_DATA);

            waIntent.setPackage("com.whatsapp");

            waIntent.putExtra(Intent.EXTRA_TEXT, text);
            startActivity(Intent.createChooser(waIntent, "Share with"));

        } catch (PackageManager.NameNotFoundException e) {
            Toast.makeText(context, "WhatsApp no Instal·lat", Toast.LENGTH_SHORT)
                .show();
        }
    }
}

```

Aquest codi permet enviar un missatge a través de Whatsapp amb el text predefinit. En cas de no trobar Whatsapp instal·lat, ho notificarà amb un missatge a pantalla, anomenat Toast.

```
String[] Dades = {
    TextConcepte.getText().toString(),
    TextQuantitat.getText().toString(),
    "2"
};

DebtoidDBOpenHelper.updateDeute1a1(db, Dades, finalDeute);
```

A la imatge es pot veure el codi que fa possible que a la base de dades es deixi un deute com a pagat. Es tracta d'emmagatzemar en estat el valor «2» en lloc de 1. Al cercar els deutes, s'estableix la condició de què l'estat del deute ha de ser 1.

Com s'ha esmentat abans hi ha una sèrie de classes amb funcionaments molt semblants i variacions petites en els funcionaments. Ja que s'entrega el codi juntament amb la memòria, no s'explicaran més classes controladores.

4.2.4. Vistes del projecte

Degut a què l'aplicació funciona a través de bastants menús, alguns d'ells bastant semblants, hi han moltes semblances entre algunes vistes. S'exposaran els detalls d'algunes de les més significatives i de la seva codificació. La codificació de les vistes es realitza mitjançant arxius XML, un llenguatge semblant a HTML, però més senzill. Android Studio ens permet veure a pantalla el disseny abans de provar-lo a un terminal físic.

Vista Main

```

<TextView
    android:layout_width="220dp"
    android:layout_height="20dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="10dp"
    android:background="#ffff2a2a"
    android:enabled="false"
    android:text="Deute"
    android:textColor="#ffddf6f4"
    android:id="@+id/TextDeus" />
<TextView
    android:layout_width="220dp"
    android:layout_height="20dp"
    android:layout_gravity="center_horizontal"
    android:background="#ff008080"
    android:textColor="#ffddf6f4"
    android:text="Pagaments pendants"
    android:id="@+id/TextDec"/>
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:baselineAligned="false"
    android:isScrollContainer="false"
    android:hapticFeedbackEnabled="false"

    android:background="#ffddebdd"
    android:layout_weight="1">

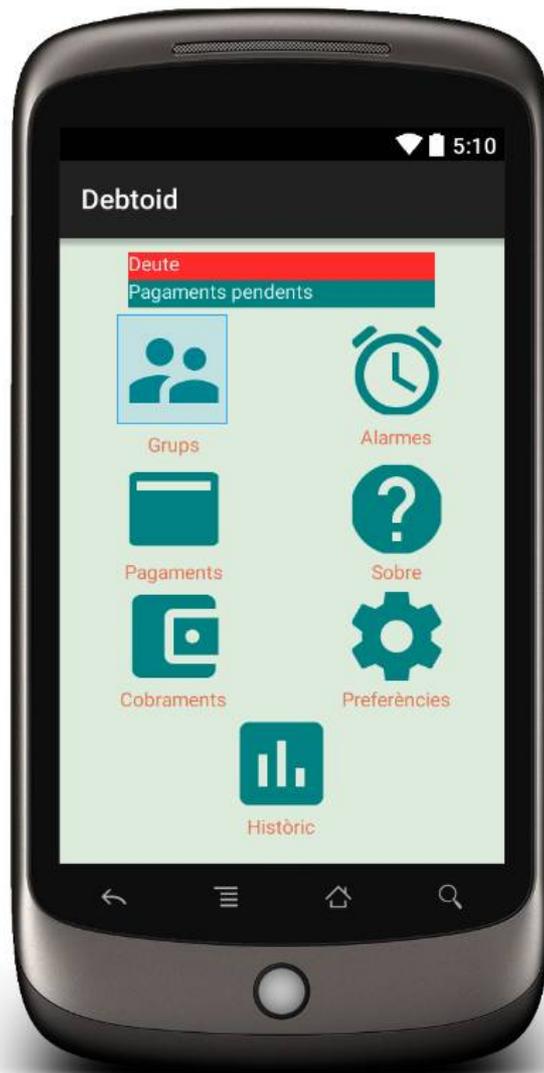
    <TableRow android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center_horizontal"
            android:orientation="vertical" >

            <ImageButton
                android:id="@+id/buttonGrups"
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"
                android:scaleType="centerInside"
                android:adjustViewBounds="true"
                android:layout_weight="1"
                android:src="@mipmap/ic_action_grupos"

```

Es correspon amb el menú d'inici de l'aplicació. Tal com es veu, el codi XML es basa en etiquetes que defineixen els objectes de la vista. Per exemple, al començament tenim «<TextView ... />», aquest codi significa que s'ha de carregar una vista de text amb les característiques indicades, ja siguin mida, color, text prefixat, etc.

La imatge que es mostra a pantalla és la següent:



Vista preferències

Es tracta d'un cas més especial de vista, ja que no permet la visualització prèvia. El motiu és que es carrega dinàmicament des del controlador, i solament permet indicar les opcions que es mostraran:

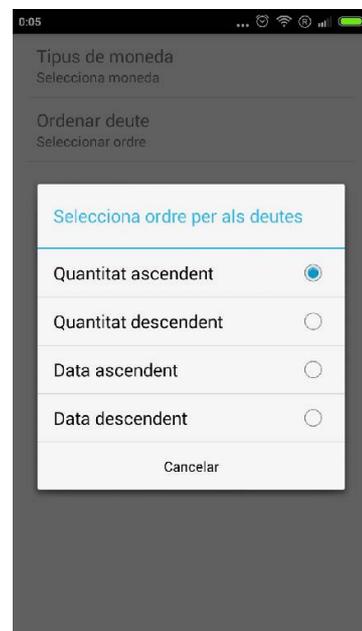
```

PreferenceScreen
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
  <ListPreference
    android:key="opciomoneda"
    android:title="Tipus de moneda"
    android:summary="Selecciona moneda"
    android:dialogTitle="Selecciona tiups de moneda"
    android:entries="@array/moneda"
    android:entryValues="@array/tipusmoneda"
    android:defaultValue="1"
  />
  <ListPreference
    android:key="opciordre"
    android:title="Ordenar deute"
    android:summary="Seleccionar ordre"
    android:dialogTitle="Selecciona ordre per als deutes"
    android:entries="@array/ordre"
    android:entryValues="@array/ordreper"
    android:defaultValue="1"/>
</PreferenceScreen>

```

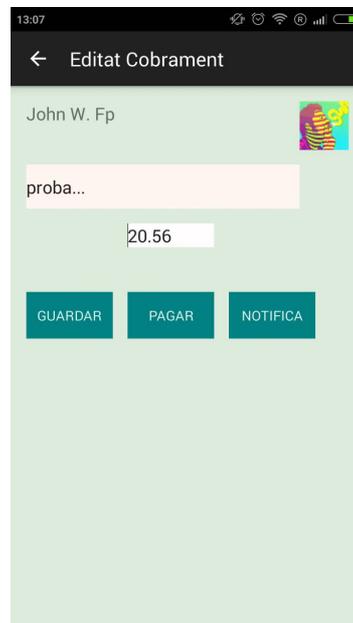
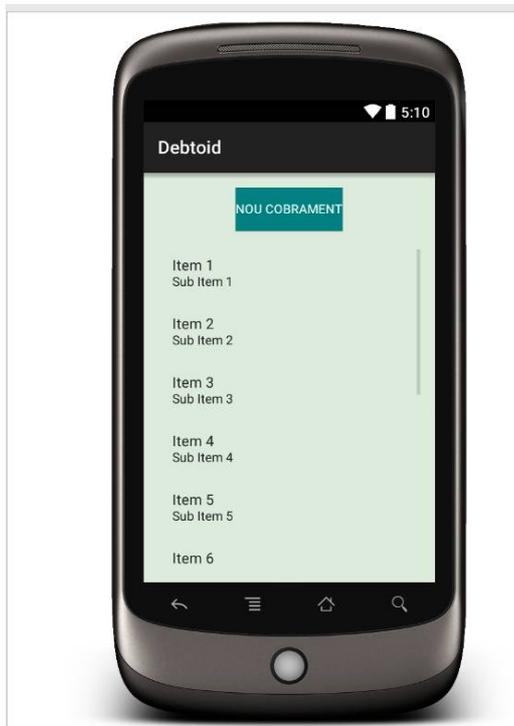
En aquest cas, s'especifica que s'han de carregar dos llistats de preferències, moneda i ordenar deute. Els texts que es mostren a cada un dels llistats es troba a un altre arxiu XML anomenat arrays.

La vista es veuria com segueix al telèfon:



Vista Cobraments:

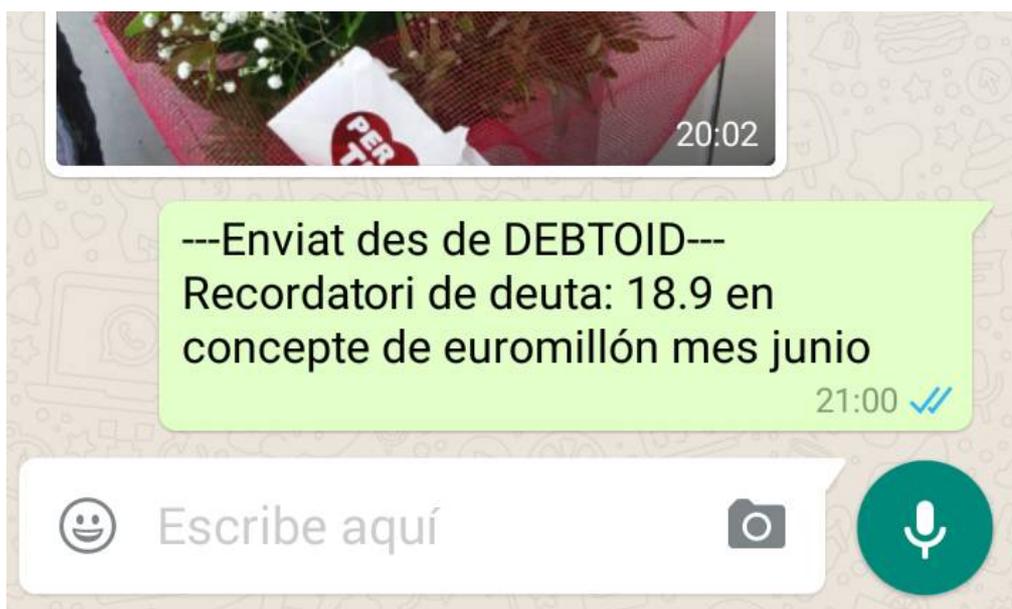
El més interessant d'aquesta vista és el fet de què disposa d'un llistat inicialment buit, que omple el controlador al cridar a la Activity corresponent.



A la imatge de l'esquerra la imatge a Android Studio. A la dreta, captura en execució a un telèfon.

Vista notificació de Whatsapp:

Com a darrer exemple, es mostra una notificació enviada per Whatsapp:



Vista Alarmes:

Aquesta vista és una excepció. Realment no existeix, degut a què el que fa l'aplicació és cridar al calendari d'android, des del qual es poden crear alarmes personalitzades. S'ha fet així per tal de simplificar l'aplicació.

4.2.5.Manifest

El manifest és un arxiu XML que no es correspon a cap dels tipus vistos fins ara. Aquest defineix l'abast de les possibles accions de l'aplicació envers el dispositiu. Podríem parlar d'una capa de seguretat a més de definir les versions mínimes i objectiu del sistema al que s'instal·larà l'aplicació.

En el Manifest s'especifiquen entre altres opcions, els permisos que l'aplicació necessita. En el cas de Debtoid, es necessita el permís per a llegir els contactes de la memòria del telèfon. Tot i que pot semblar obvi, és molt útil a l'hora d'instal·lar una aplicació nova a un dispositiu, ja que podem sospitar d'una aplicació que demani permís per exemple per fer pagaments quan suposadament no ho ha de fer, per posar un exemple.

```
<uses-sdk
    android:minSdkVersion="16"
    android:targetSdkVersion="21" />

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Debtoid"
    android:logo="@drawable/ic_launcher"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:configChanges="orientation"
        android:label="Debtoid"
```

Tal com es pot veure a la imatge, s'ha hagut de declarar què Debtoid por utilitzar els següents serveis:

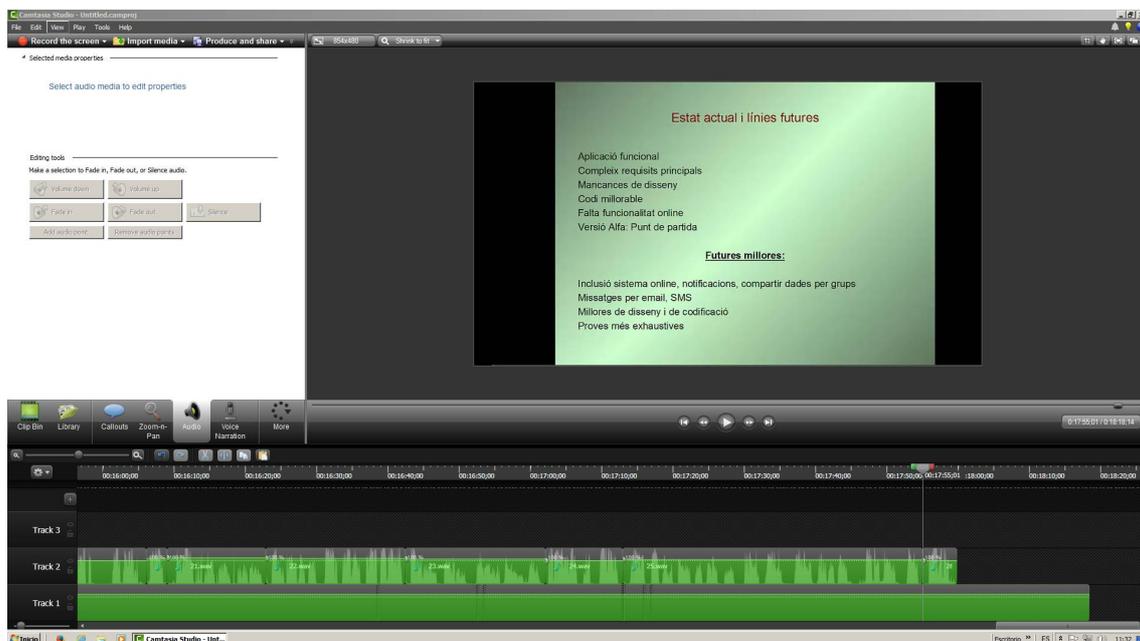
- Internet (per futures versions)
- Accés a la xarxa (per futures versions)
- Escriptura i lectura a la targeta de memòria, per poder escriure a la base de dades.
- Llegir l'estat del telèfon.
- Llegir els contactes: Aquest és el cas més obvi, ja que l'aplicació necessita extreure la informació dels contactes de l'usuari.

4.3. Proves realitzades

S'han realitzat proves manuals de l'aplicació a varis Smartphones. El resultat és satisfactori en quant a funcionament. No s'ha pogut comprovar amb emulador degut a limitacions de Hardware de l'equipament del que es disposa.

4.4. Presentació

S'ha realitzat i entregat una presentació en vídeo (format MP4), utilitzant el software Camtasia Studio (versió de prova). El qual és un programa que permet editar àudio i vídeo, així com capturar el que es fa a la pantalla de l'ordinador.



Tal com es veu a la imatge, tenim 3 àrees diferents:

L'àrea del vídeo, on podem veure el clip sobre el que treballem, ocupa la part de dalt a la dreta.

L'àrea d'edició i la línia de temps, just a sota de l'àrea de vídeo. Es on podem inserir el so, tallar la imatge, afegir efectes...

L'àrea d'opcions, a dalt a l'esquerra, ens permet gravar la pantalla, fer una narració, crear animacions, afegir text.

Tot i no tenir coneixement previs de software d'edició i captura de vídeo, es tracta d'una aplicació molt potent, malauradament, la llicència per tenir-la és bastant cara.

5. Estat actual del projecte

L'estat actual del projecte és el següent:

L'aplicació funciona correctament, tot i que el disseny s'ha de millorar i afegir algunes funcionalitats.

S'han complert les especificacions mínimes en quant a funcionament.

Es pot considerar una versió Alfa d'una aplicació, és a dir, actualment no s'hauria de publicar al Play Store. No obstant, és un punt de partida avançat per millorar l'aplicació i més endavant procedir a publicar-la.

Sobre Whatsapp: Tal com es pot comprovar, al enviar una notificació, s'ha d'escollir manualment l'usuari destinatari. Això és degut a restriccions per part de l'API de Whatsapp, la qual restringeix, possiblement per evitar enviament d'SPAM, automatitzar l'enviament del tot.

En quant al públic objectiu, segons el propi Google, al utilitzar la versió d'Android 4.1 o superior degut a algunes funcions que així ho requerien, el públic objectiu serà com a molt d'un 61,3% dels usuaris totals de la plataforma Android.

2.2	Froyo	8	99,5%
2.3	Gingerbread	10	90,4%
4.0	Ice Cream Sandwich	15	82,6%
4.1	Jelly Bean	16	61,3%
4.2	Jelly Bean	17	40,9%
4.3	Jelly Bean	18	33,9%
4.4	KitKat	19	< 0.1%
5.0	Lollipop	21	

5.1. Futures millores

Entre les futures millores que s'han pensat per a millorar l'aplicació, destaquen les següents:

La més important de totes les possibles millores seria la inclusió del sistema online, mitjançant el qual es podria avisar a un altre usuari de que té un deute o bé de que se l'ha inclòs a un grup, per exemple. També permetria implementar un sistema de usuari i contrasenya mitjançant Google+.

A més de notificar als usuaris per Whatsapp, es podria fer que es notifiqués

mitjançant SMS, correu electrònic, etc.

El disseny es pot millorar, afegir icones als botons on solament hi ha text, per exemple, posar fons a les pantalles.

Es podria afegir a la barra de l'aplicació una icona d'informació que indiqui en cada pantalla quines opcions es tenen.

A ser possible, s'hauran de fer més proves, que siguin més exhaustives sobre l'aplicació.

En referència a aquesta darrera reflexió, s'haurien de controlar millor les dades introduïdes, possibilitar la creació d'alarmes des de la pròpia aplicació en el sentit de que es puguin establir límits de deutes, avisos quan introduïm valors molt grans, etc.

6. Conclusions

Després d'haver implementat i provat l'aplicació, i coneixent l'estat actual de la mateixa, les conclusions extretes de tot el projecte són les següents

Tot i intentar seguir la planificació, no ha sigut possible seguir-la de la manera plantejada inicialment. Per tant una de les primeres conclusions és que per poder fer una estimació del temps que es triga en realitzar un projecte, és necessari tenir experiència en projectes similars.

A nivell personal, considero que l'aplicació ja es pot utilitzar, tot i que espero millorar-la en un futur.

S'han complert els objectius inicials que es van plantejar.

El projecte ha sigut molt útil per aprendre més sobre el sistema Android, i per aprendre com s'hauria de gestionar un projecte. M'ha donat la oportunitat de tenir molts errors i alguns encerts. Tot el conjunt fa la experiència positiva.

7. Glossari

- Activity**: Unió de controlador i vista, col·loquialment, la «pantalla».
- Adapter**: Vector modificat per carregar les dades segons els mètodes que si li indiquin.
- Android**: Sistema Operatiu per a dispositius mòbils adquirit per Google.
- Android Studio**: IDE (Entorn de desenvolupament integrat) per a Android.
- API**: Conjunt de llibreries que un fabricant posa a disposició dels programadors
- Array**: Vector o matriu de dades utilitzat en programació.
- Base de dades**: Arxiu o sistema d'arxius on s'emmagatzemen dades de manera relacional seguint una lògica concreta.
- Controlador**: Capa d'abstracció on s'especifiquen els mètodes i comportaments d'una aplicació i connecta la capa de vista amb la de modelat.
- Inkscape**: Software d'edició d'imatges vectorials.
- Kanban**: Sistema ideat per Toyota per poder controlar de manera visual l'estat de les tasques d'un projecte.
- Lean** (metodologia): Metodologia àgil, utilitzada per millorar temps, productes obtinguts, i reduir despeses.
- Model**: Capa d'abstracció que representa les dades sobre les que es treballa.
- MVC**: Patró de programació Basat en separar la lògica de l'aplicació en models, vistes i controladors.
- Play Store**: Tenda virtual on es poden publicar les aplicacions d'Android.
- Sistema Operatiu**: Sistema que proporciona totes les funcions per tal de que una màquina, en aquest cas un terminal mòbil, funcioni.
- SQL**: Structured Query Language – Llenguatge per a tractar les consultes a bases de dades, es pot considerar gairebé un estàndard de programació.
- SQLite**: Tipus de base de dades senzilla basada en SQL que implementa Android.
- Vista**: Capa d'abstracció lògica en la que s'especifiquen les característiques de la interfície mostrada a l'usuari.
- Whatsapp**: Aplicació de missatgeria

8. Bibliografia

Llibre: Professional Android 4 Application Development. Autor: Reto Meier, editorial Wrox. Edició 2012.

www.tomsplanner.es – Utilitzada per crear el diagrama de Gantt.

Kanban:

<http://es.wikipedia.org/wiki/Kanban>

<http://hipertextual.com/archivo/2013/11/que-es-kanban/>

<http://www.javiergarzas.com/2011/11/kanban.html>

Programació:

<http://stackoverflow.com/> - Fòrum de programadors amb preguntes i respostes.

<https://developer.android.com> – Web d'Android per a desenvolupadors.

<http://jarroba.com> – Articles de programació Android

<http://sgoliver.net> - Articles de programació Android

<http://androcode.es/category/tutoriales/> - Recursos Android variats

<http://hermosaprogramacion.com> – Tutorials sobre Android.

<http://soundcloud.es/archison> – Música de fons per la presentació, amb permís de l'autor.

<http://inkscape.org/doc/basic/tutorial-basic.es.html> - Inkscape bàsic

Recursos subministrats a l'aula del TFC de la UOC.

9. Annexos

9.1.Instal·lació

Instruccions per a compilar l'aplicació:

És necessari disposar de l'aplicació Android Studio.

Opció 1 – Crear un apk:

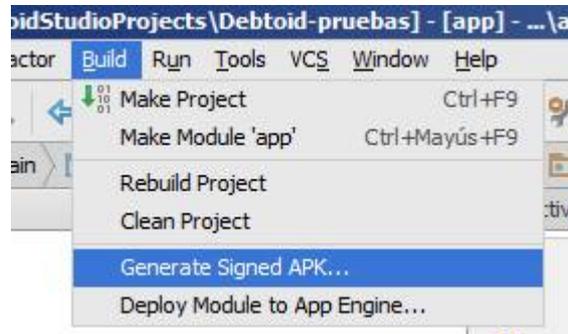
A la carpeta on tenim configurats els projectes d'Android Studio deixem una còpia de la carpeta del projecte amb els arxius de codi. Quan obrim Android Studio el podem seleccionar com a projecte per tal d'explorar els arxius o, en aquest cas, de realitzar una nova compilació.

Per tal de realitzar la nova compilació, o bé creem una nova clau, o utilitzem la clau que hi ha adjunta amb el nom d'arxiu `key_provisional_debtoid.jks`. Es tracta d'una clau provisional ja que dispenso d'una clau pròpia en cas de publicar l'aplicació al PlayStore.

Les claus per utilitzar-la són:

Alias: debtoid
Password: vbuenom

Per compilar l'aplicació i obtenir un arxiu xxx.apk, hem de clicar:

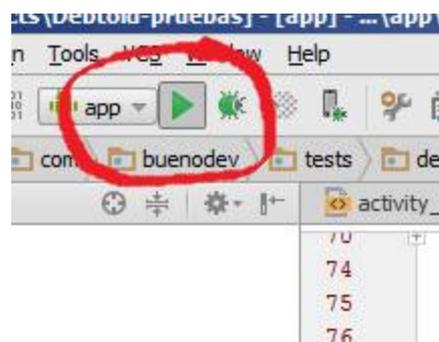


Seleccionarem importar la nostra clau, o crear-ne una de nova, un cop escollida, ens demanarà alias i password, confirmem i seleccionem la carpeta de destí, i Android Studio ens avisarà amb un missatge de que ja ha acabat el procés.

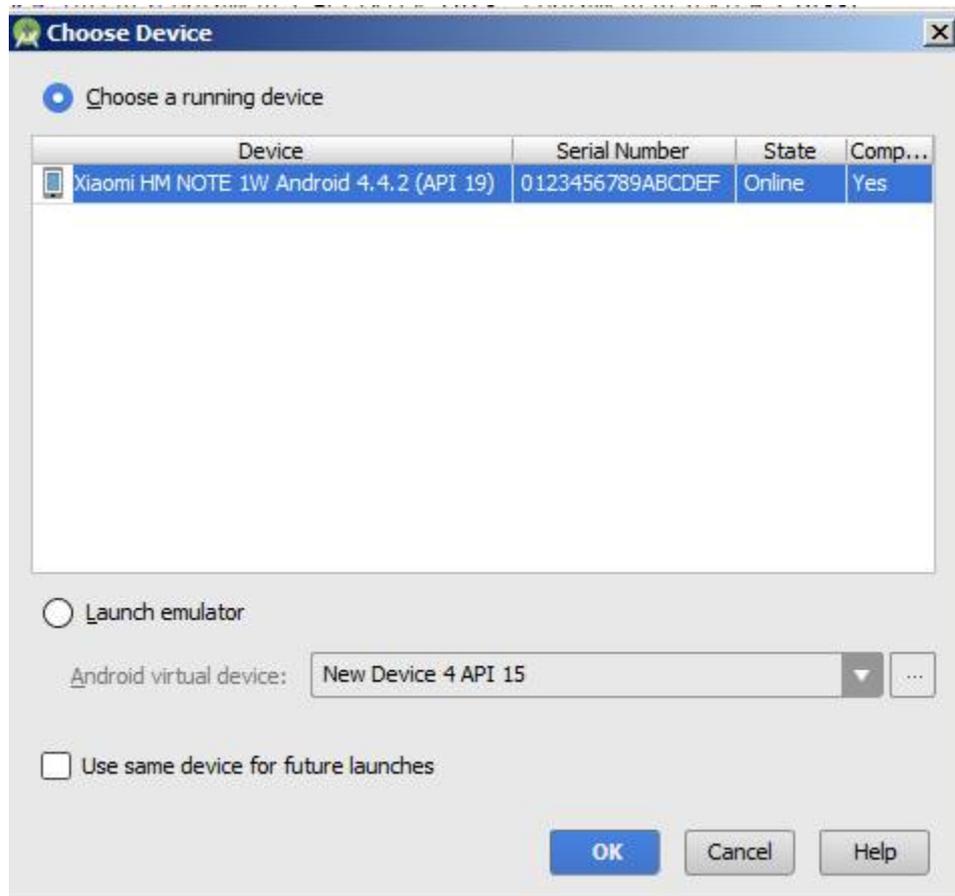
Opció 2 – Provar aplicació en un emulador o bé en terminal físic:

En el meu cas no he pogut provar l'aplicació en l'emulador, em requereix massa recursos i el meu ordinador no disposa d'acceleració Hardware entre d'altres. He fet les comprovacions mitjançant un Smartphone. En cas de testejar l'aplicació en un dispositiu físic, dins de les opcions de seguretat s'haurà de permetre el software d'origen desconegut i "debugar aplicaions per USB". En el cas de dispositiu físic, ha d'estar connectat al ordinador per realitzar les proves.

En aquest cas, un cop seleccionat el projecte, i sempre que no tinguem errors identificables (el propi IDE ens marca en vermell la pestanya de l'arxiu corresponent), farem clic a:



Quan l'IDE acabi de fer comprovacions (sol trigar una mica), mostra una pantalla semblant a la següent:



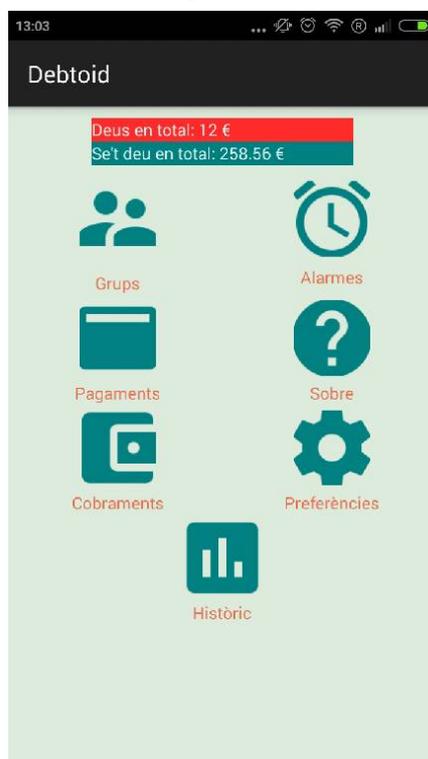
On hem d'escollir o bé el dispositiu físic, o bé iniciar l'emulador.

A partir d'aquí, l'aplicació s'executarà en el dispositiu seleccionat.

9.2.Manual d'usuari

Pantalla principal:

Al iniciar l'aplicació, apareix el següent menú:



En el qual podem veure els botons per anar a cada secció, i a la part de dalt, dos notificacions. En vermell el que l'usuari deu en total. En verd el que se li deu en total a l'usuari. Picant a cada botó, s'accedeix a la secció corresponent.

Les seccions són:

Grups: Permet crear i editar grups de contactes amb deutes comunes.

Pagaments: Permet crear deutes individuals que es tenen amb un usuari.

Cobraments: Permet crear deutes que altres contactes tenen amb l'usuari.

Històric: Mostra tres llistats amb els darrers grups i deutes tancats.

Alarmes: Crida al calendari d'Android per tal de crear alarmes i recordatoris.

Sobre: Informació bàsica de l'aplicació i com contactar amb l'autor .

Preferències: Es poden seleccionar algunes preferències generals.

Grups:

Dins d'aquesta pantalla es poden veure un llistat amb els grups que ja estan creats i actius. També podem accedir a la pantalla de creació de nous grups mitjançant un botó.

Si es pica a un dels grups del llistat, s'obrirà la pantalla d'edició del mateix, la qual mostra les dades del grup: Nom, concepte, total, i els contactes que estan dins del grup. En aquest punt es poden fer les següents accions sobre el grup:

Afegir un usuari, amb el que s'aniria a una pantalla per introduir un usuari i introduir el valor del seu deute.

Editar el deute d'un usuari: Picant a l'objecte del llistat corresponent, es carrega la pantalla d'usuari on podem canviar el deute, saldar-lo o enviar una notificació per Whatsapp.

Saldar el deute: Es pot donar per saldat el deute, Debtoid demanarà confirmació abans d'eliminar el deute.

Canviar el valor del deute: Es pot canviar el valor del deute degut a què pot ser que aquest varii en algun moment.

Notificar per Whatsapp: Es pot enviar una notificació predefinida per Whatsapp al grup o a la persona en qüestió.

Pagaments:

A la pantalla de pagaments es poden veure en un llistat els contactes amb qui es té deute i la quantitat total de cada usuari.

També es pot crear un nou deute amb el botó nou pagament. En aquest cas s'haurà d'afegir un contacte del telèfon, introduir un concepte i una quantitat i guardar.

Dins de cada deute general dels contactes trobem un llistat amb els deutes que aquest contacte té. Per exemple, un mateix usuari podria haver pagat per nosaltres un menjar i un paquet de tabac en diferents moments.

Dins de cada deute es tenen les mateixes opcions que als grups: Guardar canvis, Pagar el deute i notificar al contacte. El funcionament és el mateix que a Grups.

Cobraments:

El funcionament és igual al de Pagaments, però en aquest cas es registren els cobraments pendents.

Històric:

Es poden veure en 3 llistats els darrers 10 grups saldats, els darrers 10 pagaments saldats i els darrers 10 cobraments saldats.

Alarmes:

S'obre el calendari d'Android per veure els avisos que ja es tenen i per afegir-ne de nous.

Preferències:

Mitjançant dos selectors desplegable, es poden seleccionar el tipus de moneda per a tota l'aplicació, i l'ordre per a ordenar els deutes mostrats a pantalla.

Sobre:

Es tracta d'una pantalla amb informació bàsica respecte a l'aplicació i un correu electrònic de contacte.

Respecte a Whatsapp i les notificacions: Degut a restriccions a l'API de Whatsapp, no és possible fer un enviament directament a un usuari o a un grup. Per fer-ho, és l'usuari qui ha d'escollir el contacte o grup al que s'adreça el missatge. Per altra banda, el missatge sí que estarà ja predefinit.