



TICOCUPAT

TFC

Memòria Final TICOCupat
10 de juny de 2013

Consultors: Jordi Almirall López i Marc Domingo Prieto

La nostra recompensa es troba en
l'esforç i no en el resultat. Un esforç total
és una victòria completa.

(Mahatma Gandhi)

Així doncs vull dedicar aquest esforç a la
meva família , la meva filla i la meva dona,
ja que realment són elles les que han realitzat
l'esforç de la comprensió completa amb tot
el temps robat a estar tot plegats.

Índex de continguts

1. Descripció del projecte	5
1.1. Motivació i justificació del projecte	5
1.2. Objectius	5
1.3. Funcionalitats de TICOcupat.....	6
1.4. Recursos	6
1.4.1. Maquinari	6
1.4.2. Programari	6
1.4.3. Dispositius mòbils.....	6
1.5. DAFO	7
1.6. Planificació.....	7
1.6.1. Fase per la definició del treball i planificació	8
1.6.2 Fase per la definició dels requisits	8
1.6.3. Fase d'anàlisi i disseny de l'aplicació	8
1.6.4. Fase d'Implementació.....	9
1.6.5. Fase de proves	9
1.6.6. Fase de preparació de la documentació final.....	9
1.6.7 Diagrama de Gantt	9
2. Fase per la definició dels requisits	11
2.1. Detall de les tècniques d'anàlisi.....	11
2.1.1. Observació i investigació contextual.....	11
2.1.2. Shadowing	12
2.1.3. Mètode de diari	12
2.1.4. Anàlisi competitiu (Benchmarking).....	12
2.1.5. Entrevistes en profunditat.....	12
2.2. Anàlisi utilitzat.....	13
2.2.1. Anàlisi Shadowing	13
2.2.2. Anàlisi d'entrevistes.....	13
2.2.3. Anàlisi Benchmarking	15
2.2.3.1. Call Blocker	15
2.2.3.2. Reject Call Answer.....	16
2.2.3.3. Calls Blacklist.....	16
2.2.4. Identificació d'usuaris.....	17
2.2.5. Usabilitat	17
2.2.6. Conclusions	18
2.3. Funcionalitats de l'aplicació.....	18
3. Fase de disseny de l'aplicació	18
3.1. Diagrama de navegació	19
3.2. Diagrama de classes	19
3.3. Persistència	21
3.4. Diagrama de casos d'ús.	21
3.4.1 Especificació dels casos d'ús	22
3.4.2. Especificació dels escenaris	26
3.5. Prototipatge de baix nivell.....	29
4. Fase d'implementació.....	31
4.1. Model Vista Controlador	31
4.2. Definició Android Manifest.....	32
4.3. Interface Definition Language (AIDL).....	35
4.4. Concepte de Content Provider.....	36
4.5. Interfícies i activitats gràfiques.....	36

4.5.1. Pantalla de presentació	37
4.5.2. Pantalla de menú.....	37
4.5.3. Pantalla de telèfons.....	39
4.5.4. Pantalla d'històric.....	40
4.5.5. Pantalla de configuració	42
4.5.6. Pantalla d'ajuda	43
4.6. Suport per a llenguatges.....	44
4.7. Activitats	46
4.7.1. Activitat Intro.....	47
4.7.2. Activitat TICOcupat	48
4.7.3. Activitat AfegirTelefon	50
4.7.4. Activitat Historic	51
4.7.5. Activitat Preferencies	51
4.7.6. Activitat Ajuda	52
4.7.7. Activitat Accio.....	53
4.7.8. Classe MeuAdaptadorAfegir.....	54
4.7.9. Classe MeuAdaptadorHistoric	55
4.7.10. Classe ServeiTrucada	55
4.7.11. Classe Trucada.....	56
4.7.12. Classe Usuari.....	56
4.8. Persistència	57
4.8.1. Classe BaseDades	57
5. Fase de proves.....	59
5.1. Tests realitzats	59
5.2. Modificacions del producte final.....	60
5.3. Avaluació de l'aplicació.....	60
6. Producte final.....	60
6.1. Dificultats.....	61
6.2. Implementacions futures.....	61
7. Conclusions finals	61
8. Bibliografia	62
9. Webgrafia	62

Índex de figures

Fig. 1: Logotip de l'Aplicació	5
Fig. 2: Cicle de vida en cascada.....	7
Fig. 3: Diagrama de Gantt amb la planificació del projecte.....	10
Fig. 4: DCU	11
Fig 5: Iteració DCU	11
Fig 6: Gràfic pregunta 1	14
Fig 7: Gràfic pregunta 2	14
Fig 8: Gràfic pregunta 3	14
Fig 9: Gràfic pregunta 4	14
Fig 10: Gràfic pregunta 5.....	15
Fig 11: Interfície Call Blocker	15
Fig 12: Interfície Call Answer	16
Fig 13: Interfície Call Blacklist.....	16
Fig 14: Esquema de navegació de TICOcupat	19
Fig 15: Diagrama de classes de TICOcupat.....	20
Fig 16: Diagrama de cassos d'ús.....	22

Fig 17: Pantalla de benvinguda	29
Fig 18: Pantalla principal	29
Fig 19: Pantalles de visualització de telèfons.....	30
Fig 20: Pantalla de visualització històrica	30
Fig 21: Pantalla de configuració	31
Fig. 22: Arquitectura Android MVC	32
Fig. 23: Arquitectura TICOcupat MVC	32
Fig 24: Estructura projecte i AndroidManifest	33
Fig 25: AndroidManifest.xml.....	34
Fig 26: Broadcast receiver	34
Fig 27: receiver BD.....	34
Fig 28: activies	35
Fig 29: Estructura projecte i AIDL	35
Fig 30: AIDL ITelephony.....	35
Fig 31: Estructura gràfica del projecte.....	36
Fig 32: Pantalla de presentació.....	37
Fig 33: Pantalla de menú.....	38
Fig 34: Pantalla d'afegir telèfons	39
Fig 35: Esquema llista afegir.....	40
Fig 36: Pantalla d'històric	41
Fig 37: Esquema llista històric.....	41
Fig 38: Esquema llista històric	42
Fig 39: Esquema llista històric	43
Fig 40: Directoris de suport d'idiomes	44
Fig 41: Exemple definició strings.xml	44
Fig 42: Exemple de referència a strings.xml	45
Fig 43: arrays.xml	45
Fig 44: styles.xml	45
Fig 45: Estructura Activitat	46
Fig 46: Cicle de vida d'una activitat	46
Fig 47: Exemple d'execució d'activitat definint un Intent	47
Fig 48: Mètode onCreate de l'activitat Intro	47
Fig 49: Mètode personalitza de l'activitat Intro	48
Fig 50: Mètode onClick de l'activitat TICOcupat.....	49
Fig 51: Mètode Concedesco de l'activitat TICOcupat.....	49
Fig 52: Mètode onCreate de l'activitat AfegirTelefon	50
Fig 53: Mètode EliminaLlista i ModificaLlista de l'activitat AfegirTelefon	50
Fig 54: Mètode EliminaLlista i ModificaLlista de l'activitat AfegirTelefon	51
Fig 55: Mètode onCreate de l'activitat Preferencies.....	52
Fig 56: Mètode refresh de l'activitat Preferencies.....	52
Fig 57: Mètode onCreate de l'activitat Ajuda.....	53
Fig 58: Mètode enviarSMS de l'activitat Accio	53
Fig 59: Mètode eliminarLog de l'activitat Accio	54
Fig 60: Mètode penjaTelefon de l'activitat Accio.....	54
Fig 61: Inflat de la vista.....	55
Fig 62: Pantalla afegir	55
Fig 63: exemple d'inserció d'icona amb accions.....	55
Fig 64: Mètode onReceive de l'activitat ServeiTrucada.....	56
Fig 65: Mètode comprovaTelefon de l'activitat ServeiTrucada.....	56
Fig 66: Arquitectura SQLITE	57
Fig 67: Creació de taules BD	58
Fig 68: Afegir dades BD.....	58
Fig 69: Actualització dades BD.....	58
Fig 70: Eliminació dades BD	58
Fig 71: Aplicació TICOcupat en funcionament.....	60

1. Descripció del projecte

1.1. Motivació i justificació del projecte

En el món de les relacions comercials cada vegada més es dona molta importància a conceptes tals com la experiència de client o la comunicació proactiva com a eina necessària per una diferenciació competitiva.

Dins d'aquest entorn és primordial atendre correctament, en tot moment i en qualsevol lloc als nostre interlocutors, malgrat que es produeix una situació molt complexa de solucionar en situacions de mobilitat.

Les empreses amb un ADN purament comercial solen tenir molts recursos en mobilitat que solen atendre clients i que han d'executar una atenció excel·lent. Però que passa quan el gestor comercial està reunit? Com percep un client una trucada no atesa pel seu comercial?.

TICOcupat dona resposta a aquesta problemàtica responen de una forma automatitzada, mitjançant un SMS, als números de telèfon designats com a usuaris *VIP*¹, aportant un valor extra a la relació del comercial amb el seu client. D'igual forma *TICOcupat* permetrà definir uns números *Prohibits* que l'aplicació rebutjarà per defecte i que filtrarà aquelles trucades que el comercial no vulgui rebre en cap circumstància.

El nom de *TICOcupat* és un simpàtic joc de paraules, i que podem interpretar-lo de varies formes. D'una banda com a una elisió d'algunes lletres de *Estic*, però fent un picada d'ull al concepte tecnològic de *TIC*², que finalment s'uneix a la paraula *Ocupat*.



Fig. 1: Logotip de l'Aplicació

1.2. Objectius

L'objectiu del projecte té com a principals fites:

- Aprofundir en els principals conceptes de l'Enginyeria del programari, el desenvolupament de programari i els cicles de vida.
- Adquirir coneixements per al desenvolupament de programari per a dispositius mòbils amb les seves diferències amb el programari tipus desenvolupat al llarg de la Enginyeria tècnica.
- Assolir les habilitats de gestió i planificació d'un projecte de programari.
- Finalment aconseguir un programari útil i operatiu que faciliti la tasca d'atenció en mobilitat.

(1) VIP – Acrònim de "Very Important Person" (Persona molt important).
(2) TIC – Tecnologies de la informació i la comunicació.

1.3. Funcionalitats de TICOcupat

Les principals funcionalitats de l'aplicació seran les següents:

- Rebutjarà les trucades no desitjades que l'usuari haurà definit prèviament.
- Enviarà missatges de cortesia als Vips introduïts per l'usuari.
- Mostrar històric de trucades.
- Afegir, eliminar, editar i consultar els diferents números introduïts.

1.4. Recursos

Per desenvolupar el projecte i l'aplicació TICOcupat s'han utilitzat una sèrie de recursos que es detallaran en aquest apartat. D'altra banda no s'estan definint uns requisits mínims, si no tan sols els utilitzats durant la vida del projecte.

1.4.1. Maquinari

- Equipo portàtil *Intel Core I7 2.10GHz*.
- 8GB de memòria cau i disc dur de 720GB.
- Tarja gràfica *NVIDIA Geforce GT 640M LE*.
- Sistema Operatiu *Windows 7 Professional 64bits*.

1.4.2. Programari

- Entorn de desenvolupament *SDK de Android*.
- *IDE Eclipse JUNO*.
- *Microsoft Office 2010*.
- *Microsoft Project 2010*.
- *MagicDraw UML Personal Edition*.
- *Balsamic Mockups*.

1.4.3. Dispositius mòbils

- *Samsung Galaxy GT-S5570I amb SO v2.2 (Froyo)*.
- *HTC Desire HD amb SO v2.3 (Gingerbread)*.

1.5. DAFO

L'objectiu d'aquest anàlisi és detectar les debilitats, amenaces, fortaleeses i oportunitats del projecte i utilitzar aquesta informació per la seva millora, posicionament i correcció del producte final.

Debilitats	Amenaces
<ul style="list-style-type: none"> • Nou entorn de programació. 	<ul style="list-style-type: none"> • Espai temporal limitat. • Planificació força ajustada.
Fortaleeses	Oportunitats
<ul style="list-style-type: none"> • IDE conegut • Estructura de programació JAVA. • Ràpida corba d'aprenentatge. 	<ul style="list-style-type: none"> • Aplicació en català.

1.6. Planificació

Les fites i fases que realitzarem durant el cicle de vida del projecte són les següents:

- Fase per la definició del treball i planificació.
- Fase per la definició dels requisits.
- Fase d'anàlisi i disseny de l'aplicació.
- Fase d'Implementació.
- Fase de proves.
- Fase de preparació de la documentació final.

Com podem observar s'ha utilitzat un cicle de vida en cascada ja que és molt senzill d'aplicar en les situacions en el que es té molt clar el que es vol fer i com s'ha de fer. En definitiva els requisits estan molt clars i ben definits per l'usuari final, així com pel dissenyador que construirà l'aplicació.

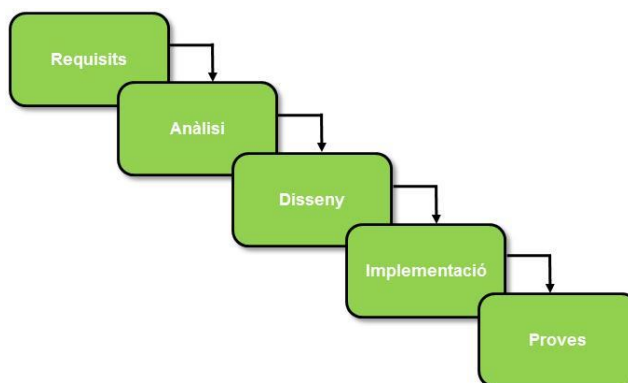


Fig. 2: Cicle de vida en cascada

1.6.1. Fase per la definició del treball i planificació

En aquesta fase es determina l'aplicació a crear, el disseny d'un pla de treball que permeti assolir les diferents fites temporals, a més de l'anàlisi de debilitats, amenaces, fortaleses i oportunitats del projecte.

Fita	Observacions	Data inici	Data final
Tria del projecte	Cercar la idea d'aplicació	28-02-13	01-03-13
Objectius	Definir l'objectiu i abast del projecte	02-03-13	03-03-13
Planificació temporal	Planificar totes les tasques i fites a assolit en el temps	04-03-13	05-03-13
Entorn de treball	Identificació de les eines necessàries i instal·lació	06-03-13	07-03-13
DAFO	Identificar les debilitats, amenaces, fortaleses i oportunitats del projecte	08-03-13	08-03-13
Documentació PAC1	Preparació del pla de treball -PAC1-	09-03-13	10-03-13
Lliurament PAC1	Lliurament del document del pla de treball -PAC1-	11-03-13	11-03-13

1.6.2. Fase per la definició dels requisits

Fase clau de recollida dels requisits complets, estudi de les necessitats i entorn del mercat. Al desenvolupar en un entorn acotat i amb un cicle de vida convencional els requisits han de ser excel·lents en la seva definició per evitar les problemàtiques pròpies d'aquest disseny lineal.

Fita	Observacions	Data inici	Data final
Recollida dels requisits	Recopilar i analitzar tots els requisits de l'aplicació	12-03-13	14-03-13
Estudi i recollida de necessitats	Identificar tots els recursos i paràmetres necessaris pel bon desenvolupament de l'aplicació	15-03-13	17-03-13
Estudi de mercat de App's similars	Realitzar una prospecció d'aplicacions semblants per identificar les característiques clau	18-03-13	20-03-13

1.6.3. Fase d'anàlisi i disseny de l'aplicació

En aquesta fase es dissenyarà tota l'estructura necessària per a l'aplicació, així com la creació i preparació de la documentació final de la PAC2.

Fita	Observacions	Data inici	Data final
Definició de l'estructura de classes	Disseny de les classes i diagrama	21-03-13	24-03-13
Definició dels casos d'ús	Disseny dels casos d'ús de l'aplicació	25-03-13	27-03-13
Definició de la persistència	Disseny de la BD que guarda els paràmetres de l'aplicació	28-03-13	30-03-13
Disseny navegació i interfícies	Disseny de la navegació entre classes i les seves interfícies gràfiques	31-03-13	02-04-13
Elaboració de la documentació	Preparació de la documentació sobre l'anàlisi i disseny de l'aplicació	03-04-13	07-04-13
Lliurament PAC2	Lliurament del document d'anàlisi i disseny -PAC2-	08-04-13	08-04-13

1.6.4. Fase d'Implementació

Durant la fase de la implantació i desenvolupament del programari és a on finalment obtindrem un producte funcional fruit de la codificació del programari.

Fita	Observacions	Data inici	Data final
Implementació interfícies	Elaboració de les interfícies gràfiques	09-04-13	16-04-13
Implementació classes	Elaboració de les classes i la seva navegació entre interfícies gràfiques	17-04-13	01-05-13
Implementació persistència	Elaboració de la BD que emmagatzemarà les dades de l'aplicació	02-05-13	09-05-13

1.6.5. Fase de proves

A aquesta fase es realitzaran totes les proves i correccions necessàries per la consecució del producte acabat. Finalment es redactarà la documentació d'implementació i proves del projecte – PAC3-.

Fita	Observacions	Data inici	Data final
Test d'estrès de l'aplicació	Test de l'aplicació per detectar les possibles errades i disfuncions	10-05-13	11-05-13
Correcció errors detectats	Correcció dels errors detectats en la fita del test d'estrès	12-05-13	14-05-13
Documentació PAC3	Preparació de la documentació sobre la implementació de l'aplicació	15-05-13	19-05-13
Lliurament PAC3	Lliurament del document d'implementació -PAC3-	20-05-13	20-05-13

1.6.6. Fase de preparació de la documentació final

Fase final de creació de tota la documentació i recursos necessàries del projecte, així como la creació d'una presentació multimèdia que permeti una presentació funcional de les àrees clau de l'aplicació al tribunal del Treball de Final de Carrera.

Fita	Observacions	Data inici	Data final
Preparació de la memòria	Elaboració de la memòria del projecte amb les conclusions, problemes i millores del projecte	21-05-13	30-05-13
Preparació de material de suport	Elaboració de la presentació multimèdia que acompanyarà al projecte	31-05-13	09-06-13
Lliurament Final	Lliurament dels documents i presentacions finals	10-06-13	10-06-13

1.6.7 Diagrama de Gantt

Aquest apartat detalla en un diagrama temporal tota la planificació dels punts anteriors amb la identificació de totes les fites a assolir.

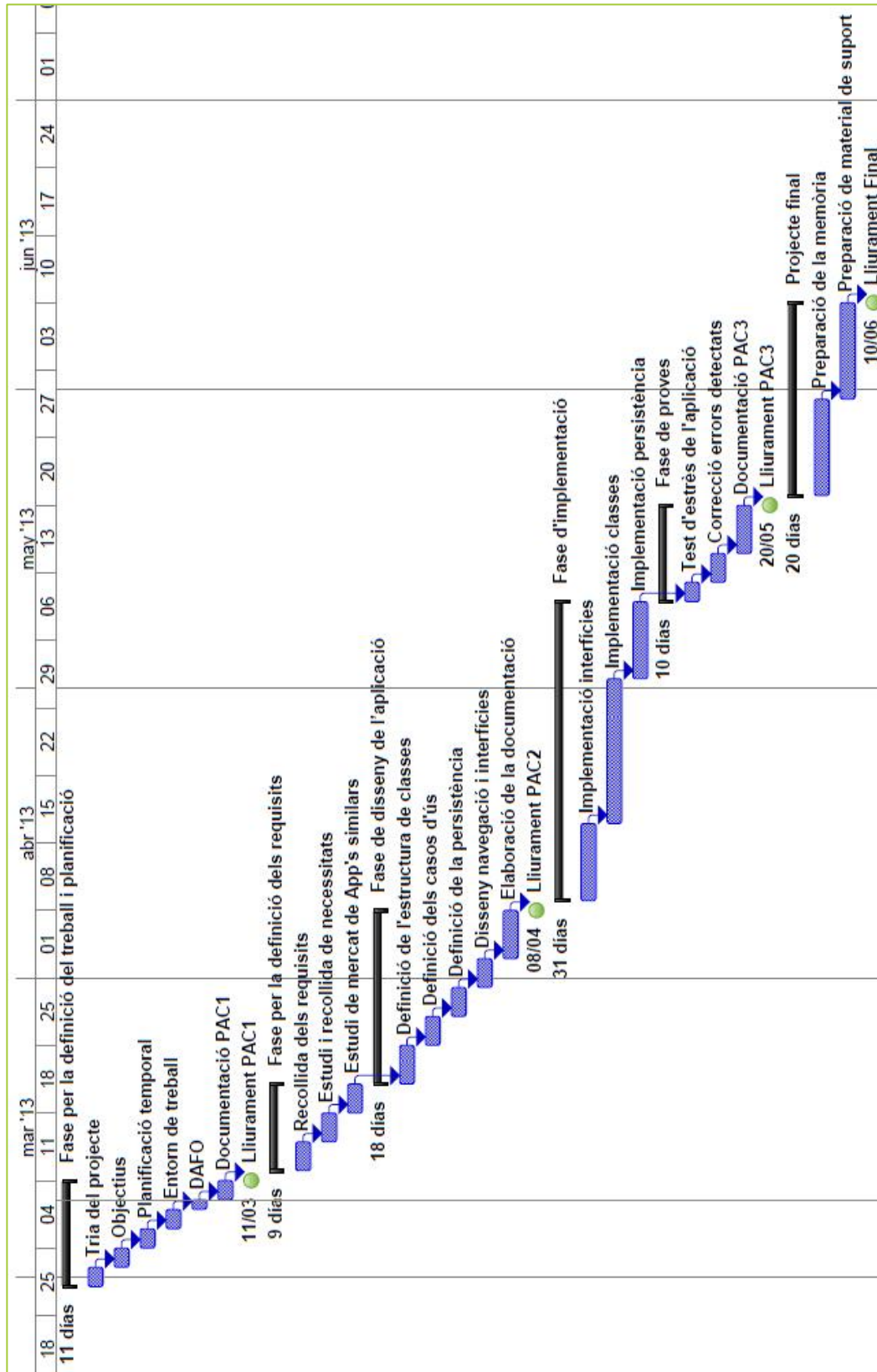


Fig. 3: Diagrama de Gantt amb la planificació del projecte

2. Fase per la definició dels requisits

En aquesta fase es detallaran totes les necessitats que haurà de complir TICOcupat, sense especificar o definir els diferents mitjans tècnics que es faran servir en el seu desenvolupament.

Per obtenir un producte final de qualitat farem servir l'anomenat **Disseny Centrat en l'usuari (DCU)**, el qual pretén orientar-se a les necessitats finals de l'usuari, per fer-lo partícip del procés i per satisfer les màximes expectatives en quan a utilitat, usabilitat, comoditat i facilitat d'aprenentatge de l'aplicació, en definitiva assolir una bona experiència d'usuari.

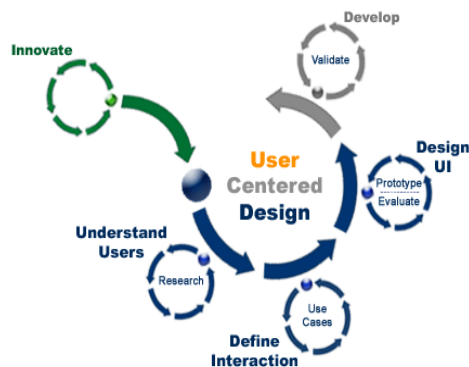


Fig. 4: DCU

Font: SAP - <http://barbellido.com/ofacw>

Malgrat que DCU pot semblar un procés oposat als cicles de vida tradicional, ja que pot implementar un cicle iteratiu i de disseny de prototips, segons podem veure a la figura 4, perfectament pot encaixar-se en la fase de recollida de requeriments i disseny segons el llibre *Técnicas Cuantitativas para la Gestión en la Ingeniería del Software* de Javier Tuya, Isabel Ramos Román i José Javier Dolado Cosín.

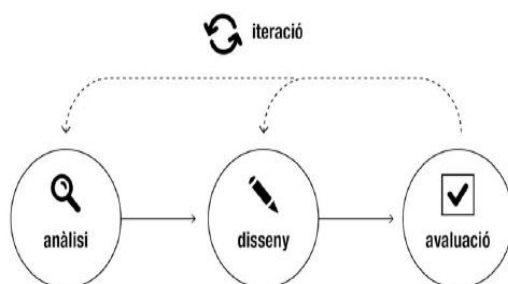


Fig 5: Iteració DCU

Font: UOC

El procés que es farà servir per detallar els requisits de TICOcupat pertany a la primera fase del procés iteratiu de DCU i que anomenem anàlisi. Aquesta fase té com a objectiu indagar i obtenir la informació necessària per passar a la fase de disseny. Informació com quines són les necessitats dels usuaris, objectius, actituds i context de l'usuari, i que es recopilarà fent servir una sèrie de tècniques que introduïrem a continuació.

2.1. Detall de les tècniques d'anàlisi

2.1.1. Observació i investigació contextual

Aquesta tècnica permet conèixer l'entorn de l'usuari i l'ús que en fa de l'aplicació. Amb aquest entorn, a on l'usuari interacciona amb el producte, el dissenyador podrà entendre les dificultats i motivacions que mouen la relació del sistema a estudiar amb el seu usuari, a més de facilitar-li el poder dirigir qüestions de caire específic per polir el seu disseny final.

2.1.2. Shadowing

El *Shadowing* es un mètode de recerca qualitatiu en que el dissenyador observa a un usuari en les seves activitats diàries, seguint-lo de la forma més discreta possible per no coartar les seves accions quotidianes i així poder recollir les dades de la forma més fidedigne possible.

La tècnica permet reconèixer accions, patrons de comportament, hàbits i rutines en el context en el que l'usuari es mou, pel que a ser una tècnica intrusiva molt freqüentment topa amb problemàtiques tan diverses com la col·laboració del propi usuari o la manca de recursos per fer un estudi en condicions reals.

2.1.3. Mètode de diari

Aquest mètode fa que el sigui el mateix usuari el que recopili la informació sobre les activitats a investigar tals com: les accions, comportaments, sensacions, etc.

En contraposició al mètode de *Shadowing* el mètode de diari no és intrusiu però sí que manifesta una sèrie de limitacions que poden donar una informació errònia o lleugerament allunyada dels objectius de l'estudi.

2.1.4. Anàlisi competitiu (Benchmarking)

La tècnica del Benchmarking³ consisteix en analitzar productes que comparteixen similituds amb el producte que es vol dissenyar. Amb l'anàlisi resultat es poden incloure funcionalitats similars a la resta de productes o intentar cercar característiques que permetin una diferenciació competitiva de la competència més estreta.

Alguns dels objectius primaris de la tècnica són els següents:

Descobrir oportunitats de mercat.

Identificar patrons que fan que l'aplicació sigui un èxit o un fracàs i els seus punts de millora.

Obtenir un punt de partida per inspirar-se en l'aplicació final que es vol dissenyar.

Identificar l'abordament de certes problemàtiques en cada aplicació.

2.1.5. Entrevistes en profunditat

Les entrevistes faciliten la recaptació directa de les motivacions, problemàtiques i necessitats de l'usuari, aquestes poden realitzar-se de les formes més tradicionals possibles com poden ser les dinàmiques de grup, Focus, enquestes individuals; o bé utilitzant eines online que permeten recollir d'una forma més massiva aquests informació.

(3) Benchmarking – Procés pel qual es recopila informació de l'entorn competiu per comparar-los amb la situació pròpia de l'empresa o producte.

El nou paradigma de les xarxes al món 2.0 ha facilitat l'aparició dels conceptes de Cocreació⁴ o Crowdsourcing⁵ en el que les entrevistes acaben per integrar en el procés de creació del producte final a l'usuari. D'aquesta forma es pretén augmentar el percentatge d'èxit del producte final i que l'usuari identifiqui que la solució s'ajusta a les seves necessitats. Necessitats que d'altra banda ell ha identificat, ha desenvolupat i que ha portat a la pràctica.

2.2. Anàlisi utilitzat

De totes les tècniques descrites en l'apartat anterior s'ha utilitzat el *Shadowing*, les entrevistes i el Benchmarking com a eines principals per analitzar l'entorn, l'usuari i les seves motivacions, i que són ampliades a continuació.

2.2.1. Anàlisi Shadowing

Com s'ha comentat a l'inici del treball i en l'apartat de motivació del projecte, l'objectiu era la creació d'una aplicació que permetés el facilitar la tasca d'atenció al client per part de professionals de diferents sectors.

L'estudi s'ha portat a terme en diferents reunions directives on hom ha observat els següents comportaments que detallem a continuació:

- Els terminals mòbils estan molt temps amb el to de trucada activat, provocant moltes interrupcions de les dinàmiques grupals a realitzar.
- Les constants interrupcions per trucades, i la seva atenció, aturen les reunions i distreuen dels objectius, converses, etc. que s'estan duen a terme.
- Malgrat que els terminals estiguin en silenci, els usuaris responen per SMS, correu o *Whatsup*⁶ a les persones que els truquen.

2.2.2. Anàlisi d'entrevistes

En l'anàlisi s'ha utilitzat l'aplicació online *SurveyMonkey* per realitzar les entrevistes als usuaris. De la mateixa manera s'ha fet servir el concepte de *Crowdsourcing* mitjançant les diferents xarxes socials per obtenir uns mínims resultats i que la mostra fos el més rica possible i sense segmentar-la en cap cas, i per sobre de tot amb les mínimes preguntes possibles per evitar el rebuig de l'usuari entrevistat.

La introducció de l'entrevista era el següent: "Recopilar informació sobre el interés en el desarrollo de una App móvil que permita filtrar llamadas indeseadas así como de avisar a los usuarios de diversos estados del usuario."

(4) Cocreació – Estrategia de negoci que genera activitats conjuntes entre l'empresa i els seus clients.

(5) Crowdsourcing – Externalització multitudinària de tasques que fins aleshores es produïen de forma interna a l'empresa.

(6) Whatsup – Aplicació mòbil per comunicar-se per missatgeria via dades.

Les preguntes bàsiques i els seus resultats gràfics són aquests:

¿Estarías interesado en instalar una App que facilite la labor comentada en el resumen?

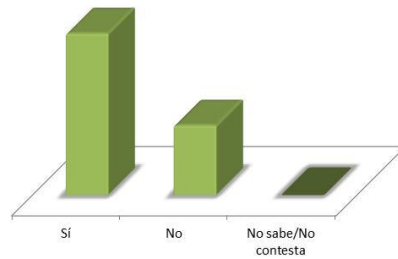


Fig 6: Gràfic pregunta 1

¿Sería interesante disponer de un filtrado de llamadas y de avisador de cortesia de diversos estados del usuario en la misma herramienta?

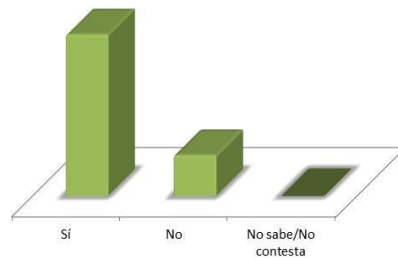


Fig 7: Gràfic pregunta 2

Valora las diferentes opciones de la App del 1 al 10:



Fig 8: Gràfic pregunta 3

¿Estarías dispuesto a comprarla con un micropago?

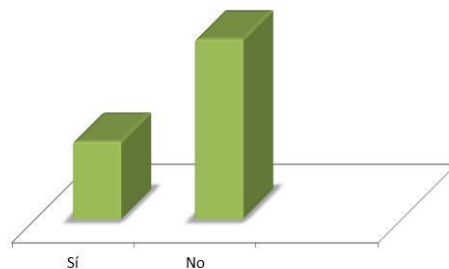


Fig 9: Gràfic pregunta 4

¿Se te ocurre alguna funcionalidad extra que te gustaría añadir?

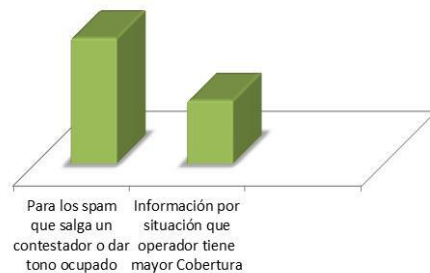


Fig 10: Gràfic pregunta 5

2.2.3. Anàlisi Benchmarking

2.2.3.1. Call Blocker



Desenvolupador: NQMobile

Web: <https://play.google.com/store/apps/details?id=com.netqin.mm&hl=en>

Anàlisi



Fig 11: Interfície Call Blocker

De les tres aplicacions analitzades Call Blocker és la que permet la més variat d'opcions configurables. Atractiva i senzilla d'utilitzar es pot customitzar al detall amb moltes funcionalitats extra.

La interfície és atractiva i molt comprensible, qual cosa permet no tenir que recórrer a l'ajuda o a un manual bàsic de funcionament.

La usabilitat i accessibilitat són bones poden accedir sense cap problema a totes les opcions configurades.

En resum té opcions interessants entre les que destaquem el que es pot d'escollir els números/contactes per afegir al telèfon des de les llistes del sistema.

Funcionalitats

- Espai privat protegit amb contrasenya.
- Bloqueja Trucades i SMS.
- Regles de bloqueig.
- Preconfigurats missatges per defecte.
- Interfície molt atractiva.
- Backup de contactes.
- Esborra les missatges i trucades del telèfon.

2.2.3.2. Reject Call Answer



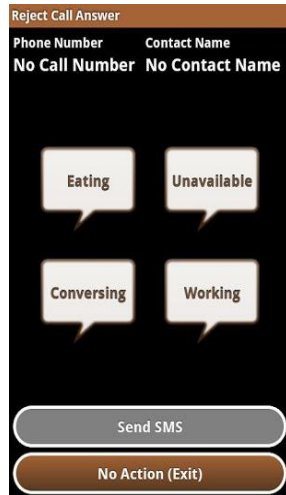
Desenvolupador: AkostaApps

Web:

<https://play.google.com/store/apps/details?id=akostaapps.rejectcallanswer.free&hl=en>

2

Anàlisi



En l'estructura base *Reject Call Answer* compleix a la perfecció la missió per la qual ha estat creat, és a dir, rebutjar trucades i missatges.

Centrada molt més que les altres *Apps* en el tractament de missatges, on podem configurar-los de moltes formes, no ofereix l'automatització al detall del rebuig de trucades de forma autònoma com seria d'esperar segons l'expectativa del programari a crear en el projecte.

Respecte a la interfície és senzilla, poc atractiva i inclou publicitat en la versió gratuïta, donant una sensació molt pobre del conjunt.

Fig 12: Interfície Call Answer

Funcionalitats

- Customització de botons i del comportament.
- Possibilitat de rebutjar trucades desconegudes.

2.2.3.3. Calls Blacklist

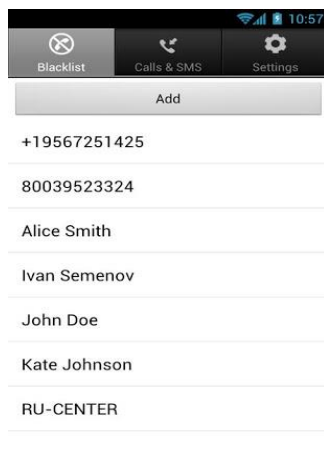


Desenvolupador: Vlad Lee

Web:

<https://play.google.com/store/apps/details?id=com.vladlee.easyblacklist&hl=en>

Anàlisi



Malgrat el seu aspecte inicial *Calls Blacklist* es una aplicació molt funcional i fàcil d'entendre. Disposa de poques opcions configurables i això fa que aquesta sigui la funcionalitat que penalitza en gran mesura a l'aplicació.

D'altra banda no ofereix la possibilitat d'enviar missatges definits als usuaris d'una llista definida, i només treballa amb el concepte de llista negra per rebutjar a petició de l'usuari les diferents trucades i missatges

Usabilitat correcta, té una gran accessibilitat a les diferents opcions del programa.

En resum, compleix la missió de rebuig

Fig 13: Interfície Call Blacklist

Funcionalitats

Protegit amb contrasenya.
Només rebutja les trucades

2.2.4. Identificació d'usuaris

Amb tots el estudis realitzats i si es té en compte el *target* objectiu de l'aplicació podríem reduir els perfils d'usuaris a dos classes:

Perfil professional. Persona que farà servir l'aplicació de forma intensiva, amb freqüència amb poc temps de interaccionar amb la solució i que requereix de unes funcionalitats molt definides com és la possibilitat de activar ràpidament els missatges predefinits.

Perfil residencial. Aquest perfil explotará les capacitats de rebuig de les trucades i podria tenir temps de entrar en molt detall en el seu ús.

Perfil mixt. Persona que podria fer un ús mixt de l'aplicació aprofitant al màxim les seves funcionalitats tant les requereixen de un ús ràpid com les que permeten una major configuració i adaptabilitat.

2.2.5. Usabilitat

El disseny final ha de facilitar que els diferents usuaris puguin utilitzar l'aplicació de una forma senzilla, clara i sense gaires complicacions. Per això es tindran en compte les indicacions per als desenvolupadors de les guies d'interfícies humanes (HIG), i si tenim em compte les funcionalitats, els estudis de Benchmarking i la identificació usuaris de *TICOcupat* haurem de complir els següents criteris:

Simplificar la navegació. L'usuari ha de tenir clar la navegació entre pantalles per no confondre'l i que l'experiència sigui el més satisfactòria possible.

Dissenyar una interfície vistosa. Atreure l'atenció de l'usuari és vital per obtenir una bona resposta i acceptació de l'aplicació. Per aquest motiu es potenciaran al màxim la inclusió de imatges o icones que identifiquin fàcilment l'opció a triar en cada moment.

Tamany adequats a l'aplicació i usuari. L'aplicació estarà dissenyada per dispositius en que la seva pantalla oscil.la entre 2.7" i 3.4" per lo que hem de distribuir correctament els diferents recursos en pantalla alhora de tenir cura que sigui del tamany correcte per que siguin fàcilment accessibles amb els dits de la majoria dels usuaris.

Agilitzar l'activació. La millor experiència que un usuari pot tenir en l'ús de l'aplicació es poder accedir i activar-la de la forma més ràpida i àgil, per aquest motiu és simplificarà al màxim el disseny de la interfície gràfica per facilitar aquesta funcionalitat.

2.2.6. Conclusions

Amb tota la informació recollida dels estudis anteriors i després de l'anàlisi de funcionalitats detectem que existeixen certes mancances en l'aspecte visual i d'usabilitat de les aplicacions analitzades o bé unes grans opcions de configuració que poden ser complicades d'entendre.

Per obtenir una bona experiència d'usuari tindrem en compte els següents punts:

Donar un aspecte visual simple i atractiu alhora.

No incrustar cap element que distregui l'atenció de l'usuari com publicitat, informació comprimida, etc.

2.3. Funcionalitats de l'aplicació

Fruit dels estudis anteriors i concretament del Benchmarking realitzat a continuació definirem les funcionalitats que extra que formaran part del programari de projecte i que especificuem a continuació:

Rebutjar les trucades.

Missatges preconfigurats.

Possibilitat de petició d'una contrasenya d'accés.

Diferents idiomes a seleccionar entre els que s'inclou el català.

3. Fase de disseny de l'aplicació

En aquesta fase es detallaran els components que formaran l'aplicació per guiar al creador en les tasques tals com: implementar, escriure el codi del programari final i realitzar els requisits de l'arquitectura, algorismes a utilitzar, estructures de dades que s'implementaran i les representacions d'interfícies gràfiques de l'aplicació.

3.1. Diagrama de navegació

L'esquema de navegació ens mostra la seqüència de pantalles per les quals navegarà l'usuari i que s'han definit a les etapes anteriors. En resum aquestes reflexen les funcionalitats bàsiques i les pautes d'usabilitat comentades a l'apartat corresponent.

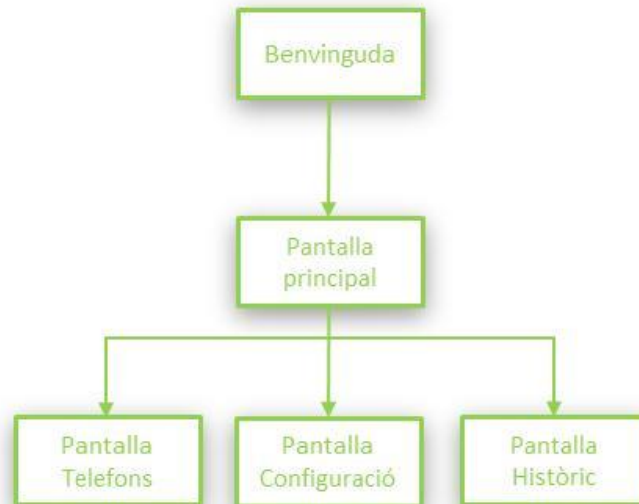


Fig 14: Esquema de navegació de TICOccupat

3.2. Diagrama de classes

El diagrama de classes ofereix una visió estàtica de les classes, objectes, interfícies, etc. que posteriorment facilitaràn al programador el codificar l'aplicació *TICOccupat*. En aquest diagrama gràfic s'especificaran les classes que intervenen en el desenvolupament de l'aplicació i com es relacionen entre elles.

Intervindran les següents classes:

- Intro.** Classe on presentarem el missatge de benvinguda de l'aplicació.
- TICOccupat.** Classe principal de l'aplicació on es visualitzarà la pantalla principal i interactuarà amb totes les opcions o pantalles.
- Acció.** Classe des d'on es realitzaran totes les accions de funcionalitats i que s'ha definit com a classe per a futures noves implementacions, i així reutilitzar la classe.
- AfegirTelefon.** Classe/pantalla on s'afegiran els diferents telèfons a tractar per l'aplicació.
- Ajuda.** Pantalla d'ajuda bàsica de l'aplicació
- BaseDades.** Classe que crea i tracta la persistència en format *SQLite*.
- Historic.** Classe/pantalla on es veurà l'historic de trucades rebut i el seu tractament.

MenuAdaptadorAfegir. Adaptador específic per mostrar una llista personalitzada.

MenuAdaptadorHistoric. Adaptador específic per mostrar una llista personalitzada.

Preferencies. Classe/Pantalla per a la personalització de les preferències.

ServeiTrucada. Classe des d'on es tracta els serveis bàsics del telèfon i que està a l'espera de rebre trucades per a posteriorment passar a l'acció.

Trucada. Classe trucada amb els seus atributs.

Usuaris. Classe Usuaris amb els seus atributs.

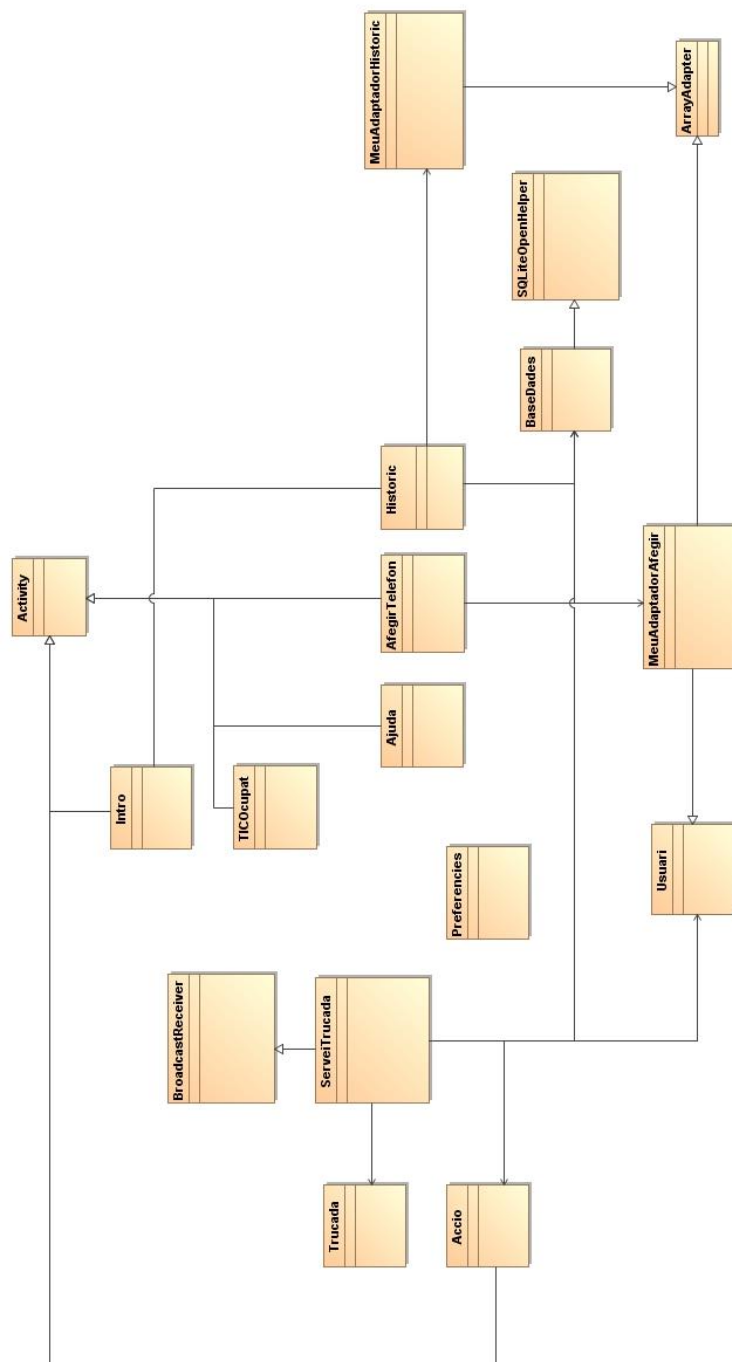


Fig 15: Diagrama de classes de *TICOcupat*

3.3. Persistència

La persistència permetrà a *TICOcupat* emmagatzemar els valors de la configuració del sistema així com els telèfons i històric de l'aplicació. Per facilitar la senzillesa del sistema es farà servir un Base per albergar els valors dels telèfons i històric i la persistència pròpia del sistema *Android*, concretament la heretada de *PreferenceActivity*, per guardar les configuracions necessàries per la funcionalitat desitjada.

Les entitats a utilitzar són les següents:

telefonos

Id1, telefon, checkvip

historic

Id2, telefon, tipus, data

Pel que fa a les configuracions del sistema la classe *PreferenceActivity* proveeix la gestió de les preferències d'aplicació i implementa la interfície de preferències que es faran servir. La mateixa classe és la encarregada de mantenir l'emmagatzematge de les opcions i no es requereix una BD o relació específica per tenir en compte aquesta funcionalitat.

Les opcions a guardar amb *PreferenceActivity* són les següents:

Enviar missatges SMS.

Rebutjar Trucades desconegudes.

Opcions predefinides dels missatges.

Text de missatge personalitzar.

3.4. Diagrama de casos d'ús.

El diagrama mostra les diferents operacions que s'esperen de l'aplicació *TICOcupat* i com es relaciona amb el seu entorn, tant a nivell d'usuaris com amb la resta d'elements que interactuïn amb la mateixa.

Si tenim en compte aquesta breu descripció el diagrama de casos d'ús de l'aplicació TICOcupat és el següent:

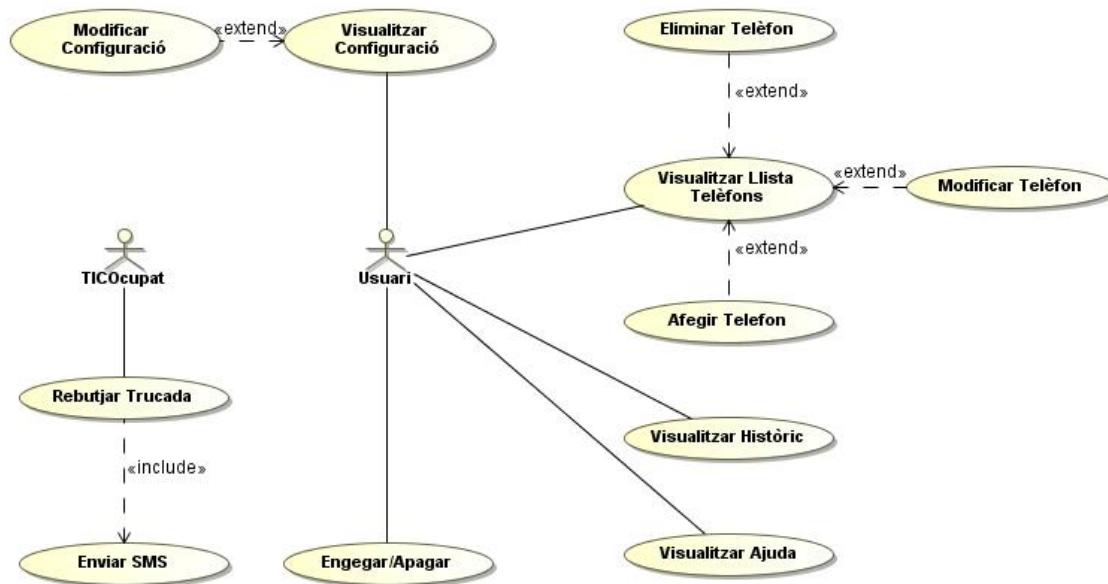


Fig 16: Diagrama de casos d'ús.

3.4.1 Especificació dels casos d'ús

En aquest apartat detallarem i especificarem els casos d'ús identificats i que ha estat reflexats al gràfic de l'apartat diagrama de casos d'ús,

Visualitzar Configuració	
Resum de la funcionalitat	Permet configurar les funcionalitats de l'aplicació
Actors principal	Usuari
Cas d'ús relacionat	Modificar configuració
Descripció del cas	L'usuari selecciona la opció de configuració i es mostra en pantalla
Precondició	En la primera execució del programa els enviaments de SMS està activat i els missatges predefinitos en la opció: "Estic ocupat. Trucaré més tard"
Postcondició	En el cas d'activació de missatge personalitzat s'activa la opció: "Personalitzar text"
Excepcions	Cap
Observacions	Cap

Modificar Configuració

Resum de la funcionalitat	Permet modificar opcions de la configuració
Actors principal	Usuari
Cas d'ús relacionat	Cap
Descripció del cas	L'usuari polsa la opció de missatges predefinits
Precondició	Cap
Postcondició	Check seleccionat de les opcions desplegadas
Excepcions	Cap
Observacions	Aquest cas d'ús es crida per triar una sèrie de configuracions específiques de missatges personalitzats

Engegar/Apagar

Resum de la funcionalitat	Permet activar i desactivar el mode reunió
Actors principal	Usuari
Cas d'ús relacionat	Cap
Descripció del cas	L'usuari polsa la opció d'engegar i apagar.
Precondició	En la primera execució la opció està en apagada
Postcondició	Si està apagada i es polsa l'opció passa a estar activada (timbre desactivat). Pel contrari si està engegada i es polsa passa a apagada (timbre activat).
Excepcions	Cap
Observacions	Cas d'us que activa l'opció de mode de reunió i que activa el nucli i objectiu de l'aplicació TICOcupat

Visualitzar Llista de Telefons

Resum de la funcionalitat	Visualitza els telèfons introduïts per l'usuari
Actors principal	Usuari
Cas d'ús relacionat	Eliminar Telèfon, Modificar Telèfon i Afegir Telèfon
Descripció del cas	L'Usuari polsa en l'opció visualització. A continuació es mostra tots els telèfons introduïts en sistema amb la seva categorització
Precondició	Cap
Postcondició	Cap
Excepcions	Cap
Observacions	Visualitza els telèfons introduïts per l'usuari i permet afegir i modificar als telèfons mitjançant la mateixa pantalla

Eliminar Telèfon

Resum de la funcionalitat	Elimina el telèfon del llistat
Actors principal	Usuari
Cas d'ús relacionat	Visualitzar Llista de Telefons
Descripció del cas	L'usuari polsa la icona que està al costat del número telèfon al llistat visualitzat, per posteriorment via una confirmació eliminar-lo
Precondició	Han d'existir telèfons al llistat
Postcondició	Telèfon eliminat de la llista alhora que de la base de dades que els emmagatzema
Excepcions	Cap
Observacions	Elimina el telèfon del llistat

Modificar Telèfon

Resum de la funcionalitat	Modificar les dades del Telèfon introduït
Actors principal	Usuari
Cas d'ús relacionat	Visualitzar Llista de Telefons
Descripció del cas	L'usuari polsa la icona que està al costat del número telèfon al llistat visualitzat, per posteriorment via una confirmació modificar-lo
Precondició	Han d'existir telèfons al llistat
Postcondició	Telèfon modificat de la llista alhora que de la base de dades que els emmagatzema
Excepcions	Cap
Observacions	Es poden modificar els números de telèfon i el seu tipus (VIP/Rebutjar)

Afegir Telèfon

Resum de la funcionalitat	Afegir telèfons
Actors principal	Usuari
Cas d'ús relacionat	Visualitzar Llista de Telefons
Descripció del cas	L'usuari polsa la icona d'afegir, introdueix el número de telèfon i la seva categorització (VIP/Rebutjar)
Precondició	Cap
Postcondició	Telèfon afegit a la llista alhora que de la base de dades que els emmagatzema
Excepcions	Cap
Observacions	Es poden afegir els números de telèfon i el seu tipus (VIP/Rebutjar)

Visualitzar Històric

Resum de la funcionalitat	Visualitzar l'històric
Actors principal	Usuari
Cas d'ús relacionat	Cap
Descripció del cas	L'usuari polsa l'opció d'històric per accedir a un llistat de les darreres trucades
Precondició	Cap
Postcondició	Si han hagut trucades categoritzades es mostraran a la pantalla
Excepcions	Cap
Observacions	Visualitza l'històric de les trucades. Dins del llistat es mostraran les trucades dels VIPS que hem rebut durant l'activació del mode reunió, així com les rebutjades o filtrades des de la seva execució.

Visualitzar Ajuda

Resum de la funcionalitat	Visualitza l'ajuda principal
Actors principal	Usuari
Cas d'ús relacionat	Cap
Descripció del cas	L'usuari polsa l'opció d'ajuda per accedir a un llistat de les funcionalitats
Precondició	Cap
Postcondició	Cap
Excepcions	Cap
Observacions	Visualitza l'ajuda principal

Rebutjar Trucada

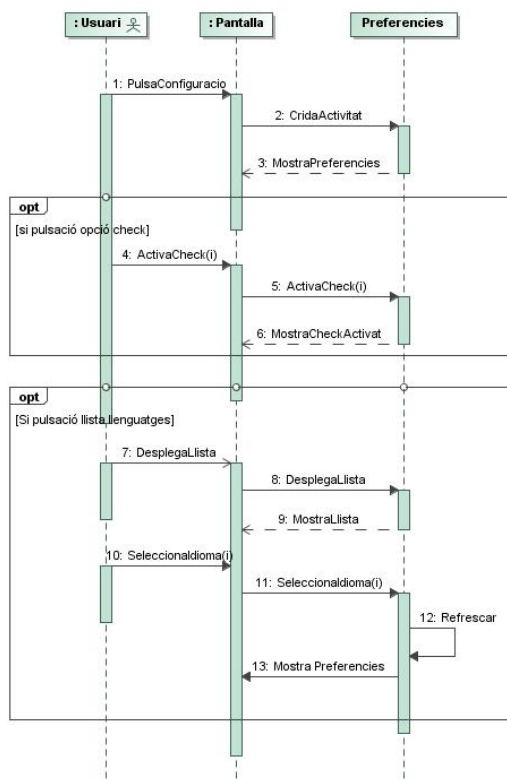
Resum de la funcionalitat	Rebutja les trucades
Actors principal	TICOcupat
Cas d'ús relacionat	Cap
Descripció del cas	TICOcupat rebutjarà les trucades dels VIP si està activada l'aplicació i les filtrades sempre
Precondició	L'aplicació té que estar activada o bé TICOcupat a rebut un número a rebutjar
Postcondició	Es rebutja la trucada si la precondició és certa
Excepcions	Cap
Observacions	Cas d'us nucli i objectiu de l'aplicació TICOcupat

Enviar SMS	
Resum de la funcionalitat	Envia SMS
Actors principal	TICOcupat
Cas d'ús relacionat	TICOcupat envia el SMS si està escollit a l'opció de configuració
Descripció del cas	
Precondició	El telèfon rep la trucada, identifica un usuari a la llista i es prepara per enviar SMS
Postcondició	Envia SMS i informa a l'usuari del seu enviament
Excepcions	L'aplicació podria informar de que missatge no s'ha pogut ser enviat
Observacions	Cas d'us nucli i objectiu de l'aplicació TICOcupat

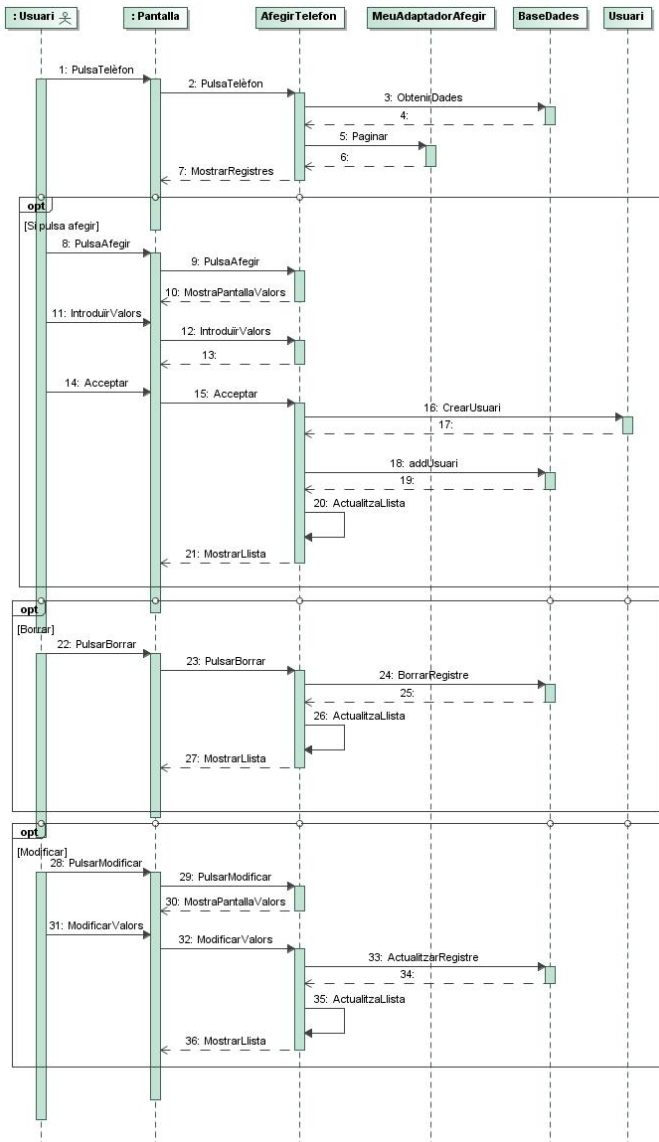
3.4.2. Especificació dels escenaris

Anteriorment s'ha comentat que existien dos o tres perfils diferents d'usuaris. Realment podríem assegurar que es tracta d'un únic perfil amb varies funcionalitats d'ús diferents si es mira des de l'òptica de l'especificació dels diferents escenaris, per aquest motiu només es mencionarà en aquest apartat l'actor genèric d'usuari.

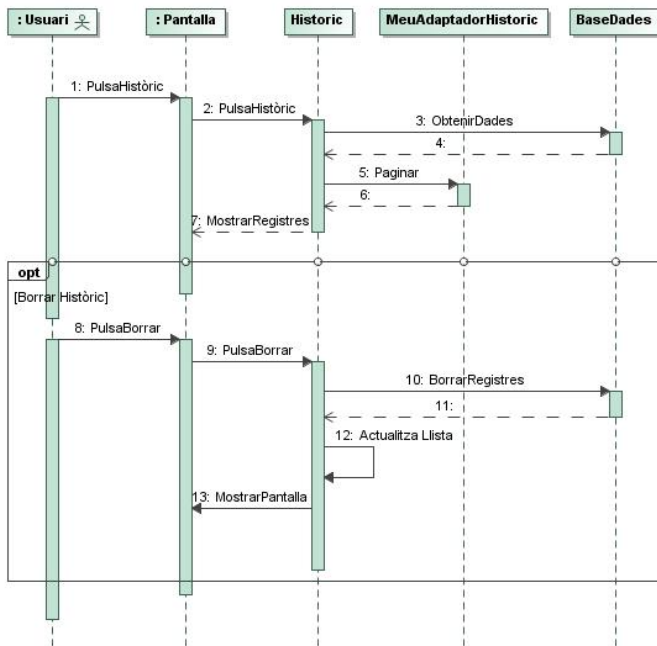
Preferències



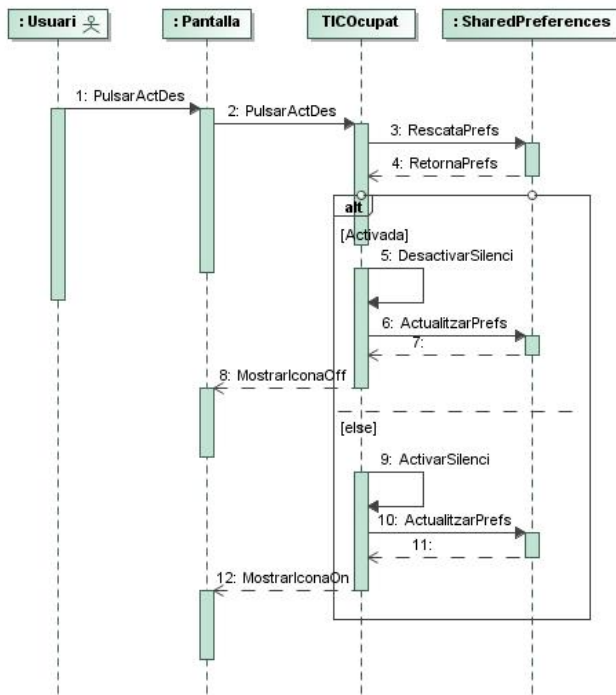
Afegir Telèfon



Històric



Engagar/Apagar



3.5. Prototipatge de baix nivell

Amb l'objectiu de validar la informació recollida durant les fases d'anàlisi i disseny, i pel que respecta a la interfície gràfica, s'ha dissenyat un prototipus de baix nivell que posteriorment es traduirà en un d'alt nivell o interfície que finalment serà el producte a lliurar al finalitzar el projecte.

Pantalla de benvinguda

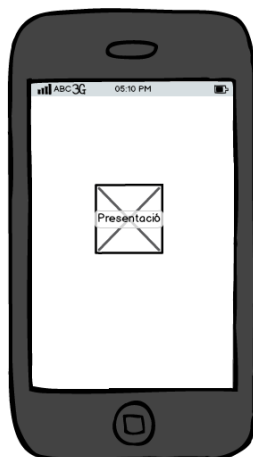


Fig 17: Pantalla de benvinguda

Pantalla d'inici on es mostrarà un logotip de benvinguda de l'aplicació utilitzant una imatge o imatges que donin a aquesta pantalla el millor aspecte visual i atractiu possible.

Pantalla principal



Fig 18: Pantalla principal

Pantalla principal a on es mostraran les diferents opcions de l'aplicació i que serà l'inici de totes les accions de l'usuari tals com: Visualitzar telèfons, afegir-ne, consultar l'històric de trucades rebudes, configuració i activació/desactivació de l'aplicació.

Pantalla telèfons

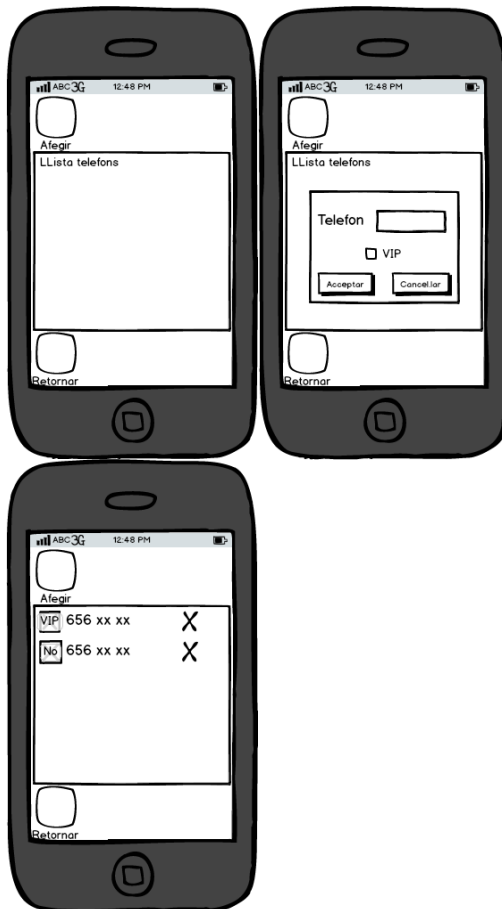


Fig 19: Pantalles de visualització de telèfons

Pantalla de visualització i addició de telèfons al sistema. Inicialment i sense cap registre en persistència veurem la pantalla inicial totalment buida. Al polsar la icona d'afegir ens demanaran les dades mínimes d'introducció (segona Pantalla) per arribar a la pantalla 3 on podem veure 2 telèfons introduïts en l'aplicació.

Pantalla històric

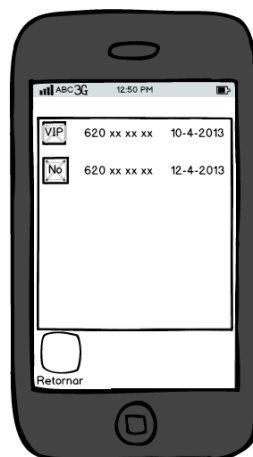


Fig 20: Pantalla de visualització històrica

Pantalla on no es permet cap interacció amb l'aplicació, a part de la navegació a la pantalla principal. En aquesta visualització podem veure 2 trucades en el sistema: una procedent d'un usuari *Vip* i l'altre d'un usuari que es vol rebutjar.

Pantalla configuració

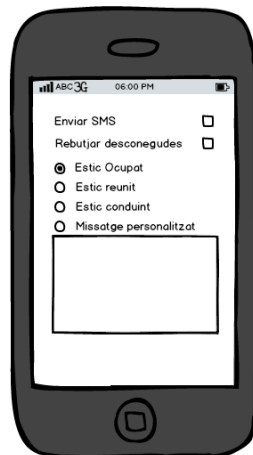


Fig 21: Pantalla de configuració

Pantalla de configuració de l'aplicació. Es poden observar les diferents opcions que el propi sistema emmagatzemarà i que dotaran de les diferents funcionalitats a TICOcupat.

4. Fase d'implementació

Durant la fase d'implementació s'escriurà el codi, la documentació necessària, i el producte executable final. Aquest codi ha de seguir les indicacions de les fases d'anàlisi i disseny, i en el cas de *TICOcupat* s'utilitzarà el llenguatge Java, tanmateix la persistència es codificarà utilitzant *SQLITE*⁷. Durant aquest procés es tindrà en compte la màxima reutilització del codi per futures implementacions, així com la verificació final de que tots els components compleixen els requisits unitaris per ser reutilitzats.

També es important destacar que en la implementació del programari s'ha utilitzat l'IDE *Eclipse* i que totes les figures d'estructura del projecte fan referència a l'estructura que pren el projecte dins el directori d'arbres de *Eclipse* i que coincideix amb l'estàndard d'*Android*.

4.1. Model Vista Controlador

Conegut com a patró de disseny MVC té com a objectiu establir un model distribuït dels diferents components del disseny i que estructura l'aplicació per facilitar la posterior reutilització, alhora que afavoreix una millora en la qualitat i la posterior depuració del codi.

Segons aquesta arquitectura es defineixen 3 nivells:

Model. És la lògica de l'aplicació, incloent els objectes que ens permeten emmagatzemar i manipular les dades i que tenen la particularitat de permetre reutilitzar i facilitar l'escalabilitat de l'aplicació. En el cas de *TICOcupat*, i en aquest nivell, podem esmentar la interacció amb la SGDB *SQLITE* que emmagatzema les dades de les trucades i els telèfons a tractar.

(7) *SQLITE* – sistema de gestió de base de dades continguda en una llibreria de petit tamany.

Vista. Es tot allò que pots veure a l'aplicació i amb el que es pot interactuar (botons, text, etc.). A TICOcupat són les diferents interfícies que s'han construït en XML.

Controlador. Aquí incloem totes les classes que ajuden a donar vida a les interfícies i que ajudaran a consumir la informació que prové de l'usuari o bé que s'ha de servir finalment a aquest. Podríem dir que és el core de l'aplicació.

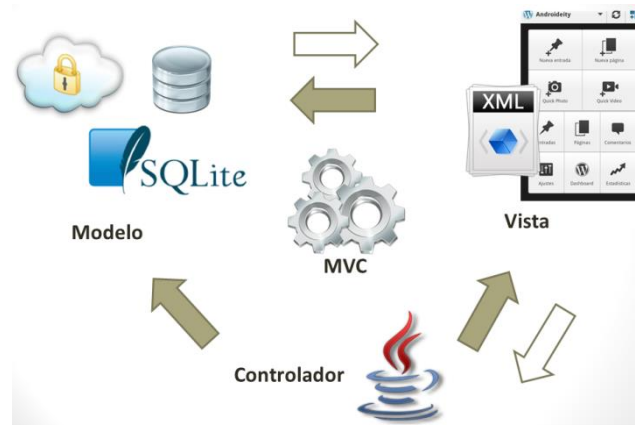


Fig. 22: Arquitectura Android MVC
Font: Androideity

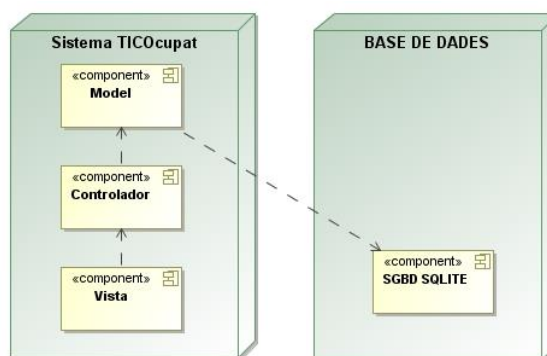


Fig. 23: Arquitectura TICOcupat MVC

4.2. Definició Android Manifest

Tota aplicació Android ha de tenir obligatòriament un fitxer AndroidManifest.xml en la seva arrel del projecte. Aquest fitxer representa la informació necessària que l'aplicació ha de tenir en el sistema Android. D'afegit el manifest també té algunes de les següents missions:

- Descriure els components de l'aplicació tals com: activitats, serveis, *broadcastreceivers* i proveïdors de continguts de l'aplicació.

- Declara els permisos dels components a fer servir.
- Llista les llibreries amb les que l'aplicació connectarà.
- Determina el mínim nivell de l'API⁸ d'Android que farà servir l'aplicació.
- Dona nom al paquet de Java de l'aplicació.
- Determina els processos que albergarà l'aplicació.

Com es pot observar a la imatge Fig.24 el *AndroidManifest* està allotjat a l'arrel del projecte *Eclipse*. D'altra banda és important destacar que en la creació del projecte es crearà automàticament un manifest per defecte que s'haurà de enriquir amb les serveis, processos i permisos necessàries per un funcionament correcte de l'aplicació a desenvolupar.

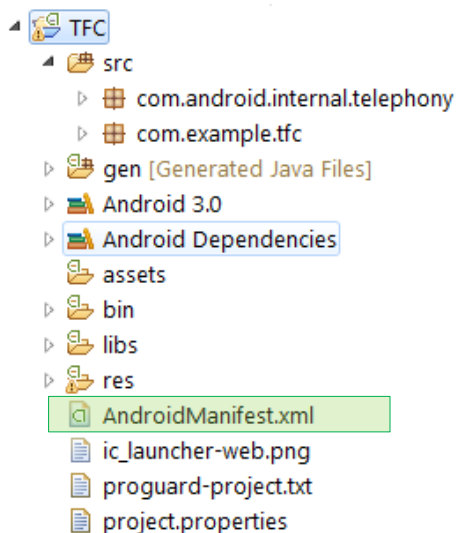


Fig 24: Estructura projecte i *AndroidManifest*

L'estructura del fitxer Manifest de TICOcupat és la següent:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.tfc"
  android:versionCode="1"
  android:versionName="1.0" >

  <uses-sdk
    android:minSdkVersion="10"
    android:targetSdkVersion="15" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.MODIFY_PHONE_STATE" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.SEND_SMS" />

```

(8) API – interfície per a la programació d'aplicacions

```

<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>

<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <activity
        android:name=".Intro"
        android:label="@string/title_activity_ticocupat"
        android:configChanges="locale" >

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

    <receiver android:name=".ServeiTrucada">
        <intent-filter>
            <action android:name="android.intent.action.PHONE_STATE"/>
        </intent-filter>
    </receiver>

    <receiver android:name=".BaseDades"></receiver>
    <activity android:name=".TICOcupat" android:label="TICOcupat"> </activity>
    <activity android:name=".AfegirTelefon" android:label="AfegirTelefon"> </activity>
    <activity android:name=".Historic" android:label="Historic"></activity>
    <activity android:name=".Preferencies" android:label="Preferencies"
        android:theme="@android:style/Theme.Black.NoTitleBar"></activity>
    <activity android:name=".Ajuda" android:label="Ajuda"></activity>
    <activity android:name=".Accio" android:label="Accio"></activity>

</application>
</manifest>

```

Fig 25: AndroidManifest.xml

A la primera part de l'estructura es pot observar que la versió mínima d'API es la 2.3.3 (versió 10) i detallem els permisos que ens permetran interactuar amb el telèfon. A la segona part es veu com es defineix l'activitat d'entrada del sistema (.Intro), així com els diferents serveis i activitats de l'aplicació i que detallarem a continuació:

```

<receiver android:name=".ServeiTrucada">
<intent-filter>
<action android:name="android.intent.action.PHONE_STATE"/>
</intent-filter>
</receiver>

```

Fig 26: Broadcast receiver

Declara i executa automàticament el Broadcast receiver (ServeiTrucada) per rebre Intents-missatges- sobre l'estat del telèfon i que és un component d'Android que permet registrar-lo en el sistema.

```

<receiver android:name=".BaseDades"></receiver>

```

Fig 27: receiver BD

Declara la classe BaseDades per interactuar amb ella.

```

<activity android:name=".TICOcupat"android:label="TICOcupat">
</activity>
<activity android:name=".AfegirTelefon" android:label="AfegirTelefon">
</activity>
<activity android:name=".Historic" android:label="Historic"></activity>
<activity android:name=".Preferencies" android:label="Preferencies"
android:theme="@android:style/Theme.Black.NoTitleBar"></activity>
<activity android:name=".Ajuda" android:label="Ajuda"></activity>
<activity android:name=".Accio" android:label="Accio"></activity>

```

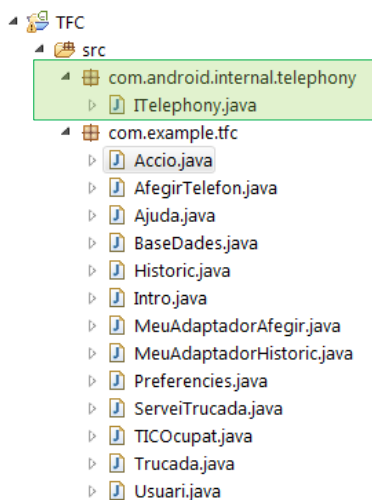
Declara totes les activitats de TICOcupat que hem d'introduir en el manifest per poder navegar entre elles.

Fig 28: activies

4.3. Interface Definition Language (AIDL)

AIDL (*Android Interface Definition Language*) permet als desenvolupadors definir una interfície de programació que el client i servidor usen per comunicar-se utilitzant el sistema de comunicacions entre processos (IPC). És important tenir en compte que a *Android* els processos no poden normalment accedir a la memòria d'un altre procés, d'igual forma han de descomposar el seus objectes en primitives que el sistema operatiu pot interpretar.

A *TICOcupat* s'ha utilitzat un AIDL/interfície anomenada *ITelephony* que facilita els serveis bàsics d'interacció amb el telèfon i concretament el de penjar les trucades.



Com es pot observar a la imatge Fig.29 el AIDL necessari per dotar de les capacitats bàsiques a *TICOcupat* per interaccionar amb el telèfon està sota la estructura del paquet anomenat *com.android.internal.telephony*.

Fig 29: Estructura projecte i AIDL

A continuació es mostra el paquet i la interfície *ITelephony* que ofereix la finalització de les trucades de l'Aplicació *TICOcupat*:

```

package com.android.internal.telephony;

public interface ITelephony {
    boolean endCall();
}

```

Fig 30: AIDL ITelephony

4.4. Concepte de Content Provider

Per definició una aplicació *Android* no pot accedir directament a les dades d'una altra aplicació. Aquesta part és un principi molt important anomenat *sandboxing*⁹ i del com s'aïlla unes aplicacions d'altres. Quan es necessari accedir a certes dades d'un altra aplicació, incloent les dades del propi telèfon, *Android* facilita unes interfícies a les dades anomenats *Content Providers* – proveïdors de continguts-.

A *TICOcupat* aquest concepte és força útil per que permet, per exemple, accedir a certes parts de la programació del telèfon com per exemple els *logs* de trucades.

4.5. Interfícies i activitats gràfiques

La part gràfica com ja s'ha comentat a l'anàlisi és una part força important per dotar d'una bona experiència d'usuari, a més de contribuir en una millor acceptació del programa en qüestió. Per aquest motiu s'ha fet un esforç important en definir un aspecte visual atractiu i alhora fàcil d'utilitzar.

Les icones que han estat utilitzades formen part d'un repositori gratuït i en la definició de pantalles s'ha utilitzat el llenguatge propi XML.

XML - sigles en anglès de eXtensible Markup Language- és un llenguatge de marques desenvolupat pel *W3C-World Wide Web Consortium*- que s'està convertint en un estàndard per a l'intercanvi d'informació estructurada entre diferents plataformes.

L'estructura del projecte permet definir unes pantalles i icones que s'adaptin als diferents formats de pantalla per que es puguin veure en diferents terminals (els més comuns del mercat). En el cas de *TICOcupat* i ja que el *target* d'aparells ha estat acotat a un tamany de pantalla de 2,7" fins a 4,3" s'ha definit un únic estil visual per les pantalles i icones.

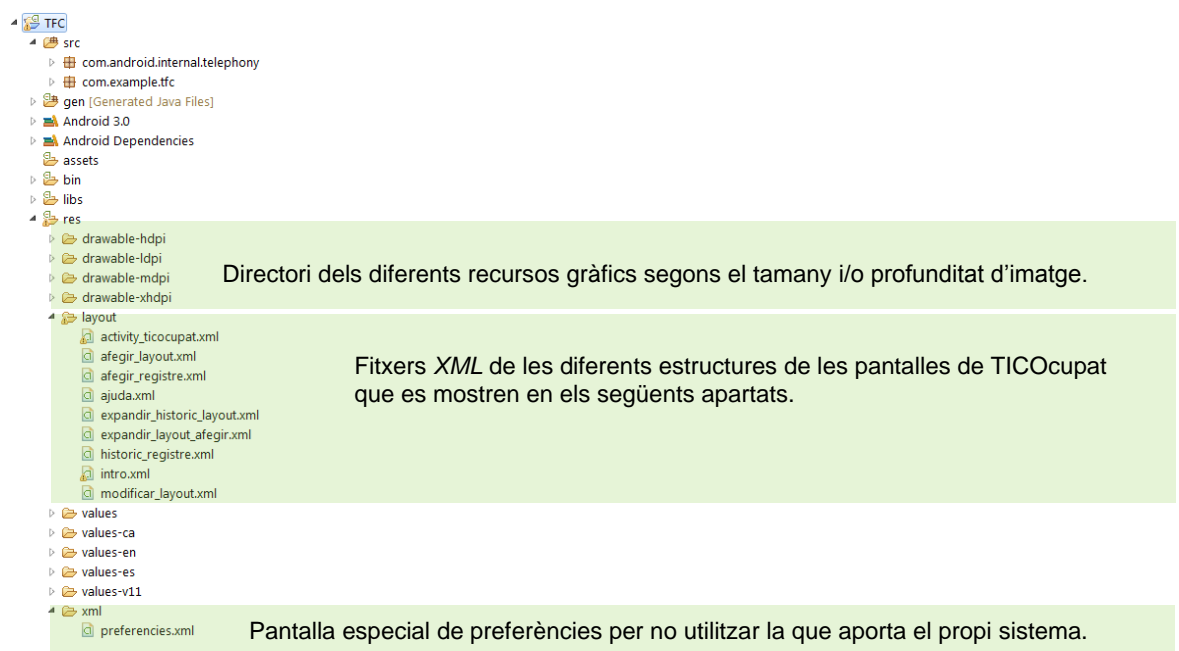


Fig 31: Estructura gràfica del projecte

(9) Sandboxing - procediment i/o mecanisme que permet aïllar i separar programes.

4.5.1. Pantalla de presentació

Pantalla simple i introductòria de la presentació de l'aplicació. Només estarà activa durant breus segons per passar posteriorment a l'activitat *TICOcupat*. Mentre es produeix l'espera es pot veure un gràfic *-ProgressBar-* que facilita que l'usuari interpreti que l'aplicació s'està carregant.



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rootRL"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/intro"
    android:orientation="vertical" >

    <ProgressBar
        android:id="@+id/widget37"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="33dp" />

</RelativeLayout>
```

Fig 32: Pantalla de presentació

4.5.2. Pantalla de menú

Pantalla principal de l'aplicació *TICOcupat* i que mostra les diferents opcions programades. Algun dels aspectes importants a destacar és la utilització de taules per paginar correctament la imatge i l'ús dels atributs adequats per que s'interpreti que les imatges tenen la funcionalitat per ser pulsades - *android:onClick="onClick"* -.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000"
    android:orientation="vertical">

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <TableRow
            android:layout_weight="1"
            android:gravity="center"
            android:paddingLeft="20dp"
            android:paddingBottom="10dp">

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:gravity="center"

                android:layout_weight="1"
                android:paddingRight="20dp">

                <ImageView
                    android:id="@+id/botoAfegirRegistre"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:onClick="onClick"
                    android:src="@drawable/afegirregistre" />

                <TextView
                    android:id="@+id/textAfegirRegistre"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/afegirregistre"
                    android:textColor="#ffffff"
                    android:gravity="center"
                    android:textSize="10sp"
                    android:textStyle="bold" />

            </LinearLayout>

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:gravity="center"

                android:layout_weight="1"
                android:paddingRight="20dp">

                <ImageView
                    android:id="@+id/botoHistoric"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:onClick="onClick"
                    android:src="@drawable/historic" />

                <TextView
                    android:id="@+id/texthistoric"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/historic"
                    android:textColor="#ffffff"
                    android:textSize="10sp"
                    android:textStyle="bold" />

            </LinearLayout>

        </TableRow>

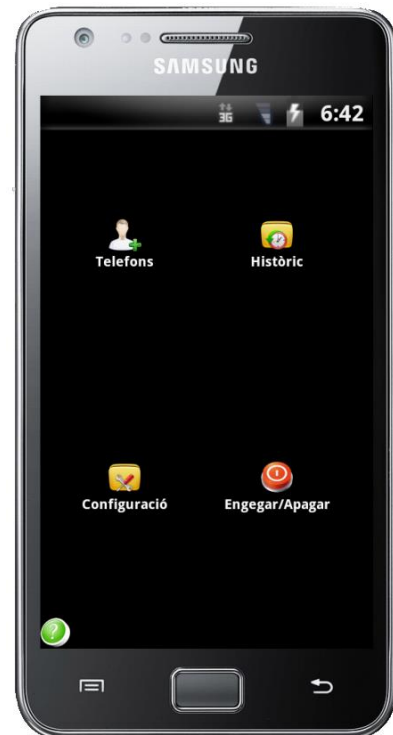
        <TableRow
            android:layout_weight="1"
            android:gravity="center"
            android:paddingLeft="20dp"
            android:paddingBottom="10dp" >

            <LinearLayout
                android:orientation="vertical"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:gravity="center"

                android:layout_weight="1"
                android:paddingRight="20dp">

                <ImageView
                    android:id="@+id/botoAfegirConfiguracio"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:onClick="onClick"
                    android:src="@drawable/configuracio" />

```



```

<TextView
    android:id="@+id/textConfiguracio"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/configuracio"
    android:textColor="#ffffff"
    android:textSize="10sp"
    android:textStyle="bold" />
</LinearLayout>
<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:layout_weight="1"
    android:paddingRight="20dp">
    <ImageView
        android:id="@+id/botoOnOff"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick" />
    <TextView
        android:id="@+id/textOnOff"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/engegar"
        android:textColor="#ffffff"
        android:textSize="10sp"
        android:textStyle="bold" />
</LinearLayout>
</TableRow>
</TableLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <ImageView
        android:id="@+id/sobre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:onClick="onClick"
        android:src="@drawable/sobre" />
</LinearLayout>
</LinearLayout>

```

Fig 33: Pantalla de menú

4.5.3. Pantalla de telèfons

La pantalla d'aquest apartat mostra els números de telèfon del sistema i la seva tipificació de la classe d'usuari: VIP o a rebutjar. Inclou una llista personalitzada amb un adaptador específic que permet mostrar elements visuals per cada element de la llista.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <ImageView
            android:id="@+id/BotoAfegirTelefon"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:onClick="onClick"
            android:src="@drawable/afegirtelefon" />
        <TextView
            android:id="@+id/pulsarAfegir"
            android:layout_width="170dp"
            android:layout_height="match_parent"
            android:layout_weight="0.14"
            android:gravity="right"
            android:textSize="10sp"
            android:text="@string/pulsarAfegir"
            android:textColor="#ffffff" />
    </LinearLayout>
    <ListView
        android:id="@+id/lLista"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="0.80" >
    </ListView>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <ImageView
            android:id="@+id/BotoTornar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:onClick="onClick"
            android:src="@drawable/tornar" />
        <TextView
            android:id="@+id/pulsarTornar"
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:gravity="right"
            android:textSize="10sp"
            android:text="@string/pulsarTornar"
            android:textColor="#ffffff" />
    </LinearLayout>
</LinearLayout>
```

Fig 34: Pantalla d'afegir telèfons

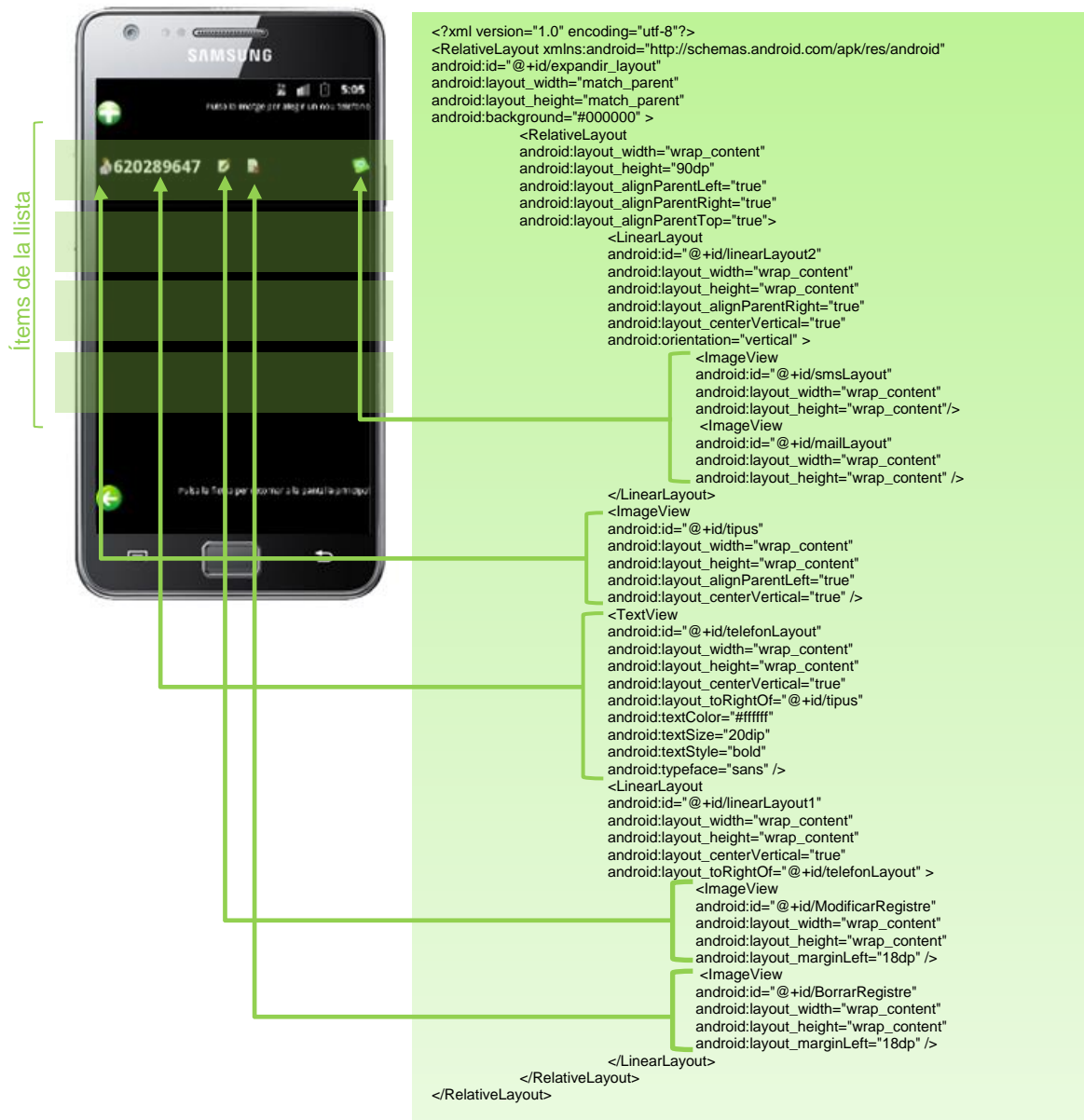


Fig 35: Esquema llista afegir

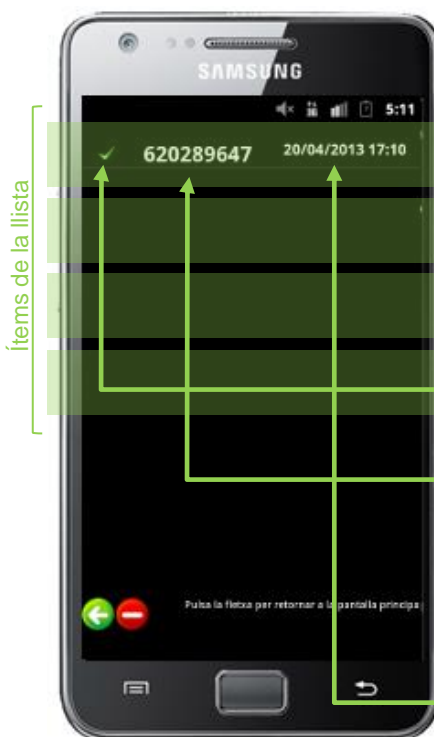
4.5.4. Pantalla d'històric

La plantilla/pantalla d'històric detalla les trucades rebudes mentre l'aplicació ha estat funcionant. A més destacarà el tipus de trucada rebuda, és a dir, si pertany a un usuari VIP o que *TICOcupat* ha rebutjat. D'igual forma que a la llista d'afegir usuaris/telèfons s'ha fet servir una llista personalitzada o un manipulador específic pel cas i que mostra la llista de trucades.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="15dp" >
    </LinearLayout>
    <ListView
        android:id="@+id/lIstahistoric"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="0.73" >
</ListView>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <ImageView
        android:id="@+id/BotoTornar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:onClick="onClick"
        android:src="@drawable/tornar" />
    <ImageView
        android:id="@+id/BotoborraHist"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:src="@drawable/borrahist" />
    <TextView
        android:id="@+id/pulsarTornar"
        android:layout_width="213dp"
        android:layout_height="60dp"
        android:layout_weight="0.60"
        android:gravity="right"
        android:textSize="10sp"
        android:text="@string/pulsarTornar"
        android:textColor="#ffffff" />
</LinearLayout>
</LinearLayout>
```

Fig 36: Pantalla d'historic



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/expandir_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000" >
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true" >
        <LinearLayout
            android:id="@+id/linearLayout1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="15dp" >
            <ImageView
                android:id="@+id/historicEstat"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"/>
            <TextView
                android:id="@+id/historicNumero"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:layout_marginLeft="27dp"
                android:textColor="#ffffff"
                android:textSize="20dip"
                android:textStyle="bold"
                android:typeface="sans" />
            <TextView android:id="@+id/historicData"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerVertical="true"
                android:layout_marginLeft="29dp"
                android:layout_toRightOf="@+id/linearLayout1"
                android:textColor="#ffffff"
                android:textStyle="bold"
                android:typeface="sans" />
        </LinearLayout>
    </RelativeLayout>
</RelativeLayout>
```

Fig 37: Esquema llista històric

4.5.5. Pantalla de configuració

La pantalla de configuració permet triar les opcions i paràmetres de funcionament de *TICOcupat*. Com es pot veure en la jerarquia de plantilles-4.5- s'ha fet servir un fitxer XML per personalitzar el format de les preferències del sistema en comptes de fer servir el propi *d'Android*, d'aquesta forma l'aspecte visual és més homogeni i permet una millor parametrització visual.



```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
  <PreferenceCategory>
    <CheckBoxPreference
      android:title="@string/EnviarSMS"
      android:summary="@string/EnviarSMS1"
      android:defaultValue="true"
      android:key="checkboxPref1" />
    <CheckBoxPreference
      android:title="@string/RebuigDesconegudes"
      android:summary="@string/RebuigDesconegudes1"
      android:defaultValue="false"
      android:key="checkboxPref1" />
    <ListPreference
      android:title="@string/missatgeDefinit"
      android:summary="@string/missatgeDefinit1"
      android:key="listPref"
      android:defaultValue="Estic ocupat. Trucaré més tard"
      android:entries="@array/Opcions"
      android:entryValues="@array/Valors" />
    <CheckBoxPreference
      android:title="@string/missatgePerso"
      android:summary="@string/missatgePerso1"
      android:defaultValue="false"
      android:key="checkboxPref2" />
    <EditTextPreference
      android:summary="@string/missatgePerso2"
      android:defaultValue=""
      android:dependency="checkboxPref2"
      android:key="editTextPref" />
    <ListPreference
      android:title="@string/idiomes"
      android:key="listPref1"
      android:defaultValue="ca_ES"
      android:entries="@array/idiomes_opcions"
      android:entryValues="@array/idiomes_valors" />
    <CheckBoxPreference
      android:title="@string/activacio"
      android:defaultValue="false"
      android:key="Activacio" />
  </PreferenceCategory>
</PreferenceScreen>
```

Fig 38: Esquema llista històric

4.5.6. Pantalla d'ajuda

La pantalla d'ajuda que es presenta al pitjar el botó del menú és la mostrada, tant en l'esquema XML com en la pantalla. Com es pot observar s'ha utilitzat un *ScrollView* per poder desplaçar la pantalla amunt i avall i poder veure tota l'ajuda de les funcionalitats. Sense ser objectiu d'aquest apartat, en la visualització es fa servir un procés que mostra el text com a format HTML¹⁰, d'aquesta forma és fàcil en tot moment adaptar el text amb nous components visuals o funcionalitats pròpies del format.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000"
    android:orientation="vertical">
    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="0dip"
        android:layout_weight="1.23" >
        <TextView
            android:id="@+id/textAjuda"
            android:layout_width="match_parent"
            android:layout_height="0dip"
            android:text="@string/ajuda"
            android:textColor="#ffffff" />
    </ScrollView>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.06" >
        <ImageView
            android:id="@+id/BotoTornar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:contentDescription="@string/none"
            android:gravity="center"
            android:onClick="onClick"
            android:src="@drawable/tornar" />
        <TextView
            android:id="@+id/pulsarTornar"
            android:layout_width="0dip"
            android:layout_height="60dp"
            android:layout_weight="0.60"
            android:gravity="right"
            android:text="@string/pulsarTornar"
            android:textColor="#ffffff" />
    </LinearLayout>
</LinearLayout>
```

Fig 39: Esquema llista històric

(10) HTML – HyperText Markup Language («lenguaje de marcat hipertextual»).

4.6. Suport per a llenguatges

Una de les funcionalitats definides com a molt importants és que l'aplicació permeti i ofereixi la possibilitat d'intercanviar l'idioma, alhora que oferir per defecte el català com a idioma per defecte.

Android permet afegir suport per diferents idiomes creant una estructura de directoris afegida de diferents valors per a cada idioma. Els valors a utilitzar per l'aplicació seran usats depenent del llenguatge configurat i triat.

En el cas de *TICOcupat* es donarà suport al català (idioma per defecte), castellà i anglès, i segons la estructura de directoris següent:

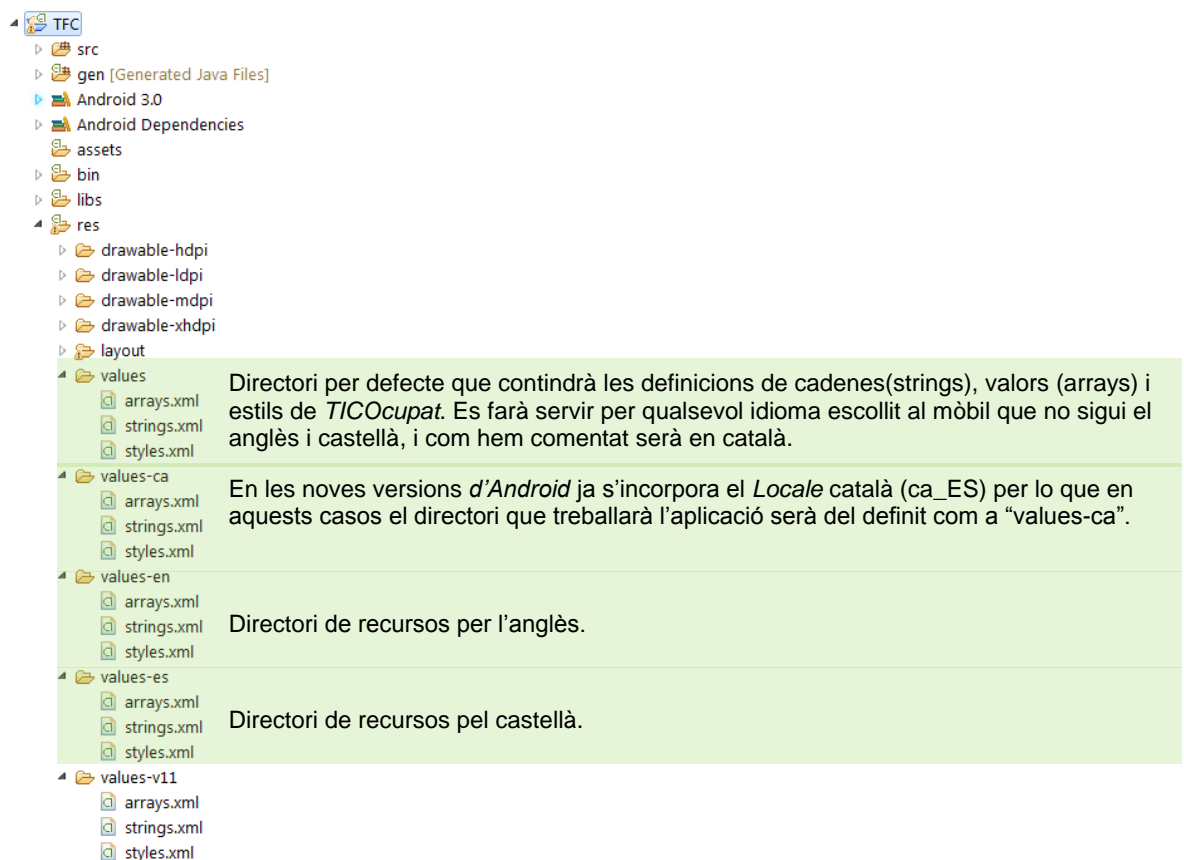


Fig 40: Directoris de suport d'idiomes

El contingut tipus del fitxer strings.xml és l'annexat en la figura següent:

```
<resources>
  <string name="SMSEnviat">SMS Enviat!!!</string>
  <string name="SMSError">El SMS nos s'ha pogut enviar!</string>
</resources>
```

Fig 41: Exemple definició strings.xml

La forma de referenciar a les etiquetes definides en els diferents *strings* d'idiomes es pot observar en els ombrejats del mètode *enviarSMS* :

```
//Mètode que permet enviar un missatge SMS
public void enviarSMS(String numero){
    try {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(numero, null, sms, null, null);
        Toast.makeText(context, R.string.SMSEnviat,Toast.LENGTH_LONG).show();
    } catch (Exception e) {
        Toast.makeText(context,R.string.SMSError,Toast.LENGTH_LONG).show();
    }
}
```

Fig 42: Exemple de referència a *strings.xml*

D'igual forma també es personalitza els valors dels *arrays* del programa per cada llenguatge. *TICOcupat* fa servir un *array* per contenir diversos valors de la llista de preferències de missatges predefinits i llenguatges:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="Opcions">
        <item name="Estic ocupat. Trucaré més tard">@string/missatge1</item>
        <item name="Estic reunit. Trucaré més tard">@string/missatge2</item>
        <item name="Estic conduint. Trucaré més tard">@string/missatge3</item>
    </string-array>
    <string-array name="Valors">
        <item name="Estic ocupat. Trucaré més tard">@string/missatge1</item>
        <item name="Estic reunit. Trucaré més tard">@string/missatge2</item>
        <item name="Estic conduint. Trucaré més tard">@string/missatge3</item>
    </string-array>
    <string-array name="idiomes_opcions">
        <item>Español</item>
        <item>Català</item>
        <item>English</item>
    </string-array>
    <string-array name="idiomes_valors">
        <item>es_ES</item>
        <item>ca_ES</item>
        <item>en_EN</item>
    </string-array>
</resources>
```

Fig 43: *arrays.xml*

I per finalitzar també es té en compte que els estils es poden també personalitzar per a cada idioma, malgrat que realment *TICOcupat* no tindrà en compte aspectes visuals diferents per a cada idioma:

```
<resources>
    <style name="AppTheme" parent="android:Theme.Light" />
    <style name="PreferencesTheme">
        <item name="android:background">#000000</item>
    </style>
</resources>
```

Fig 44: *styles.xml*

4.7. Activitats

Les activitats en *Android* són classes públiques que representen cada una de les diferents pantalles amb les que l'usuari interactua i que hereten de la classe base *android.app.Activity*. En definitiva i en el cas de *TICOcupat* conformen la base de les diferents pantalles i del codi de les accions que es poden executar a cadascuna d'elles.

L'estructura tipus mínima d'una activitat és la següent:

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

public class nombreActividad extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nom del layout);
    }
}
```

Fig 45: Estructura Activitat

Al iniciar-se l'activitat s'inicia el cicle de vida de creació i es mostra la plantilla gràfica de l'aplicació. És força important tenir clar el concepte de cicle de vida ja que posteriorment, a part de que totes les activitats sempre seran iniciades amb el mètode *onCreate*, es farà servir el cicle de vida d'una activitat per executar unes instruccions o altres segons es requereixi en cada cas.

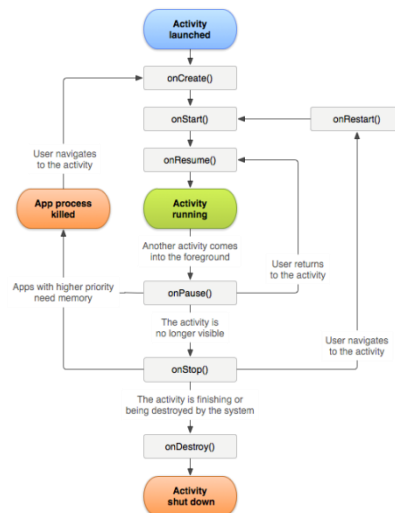


Fig 46: Cicle de vida d'una activitat

Una part important a destacar és el com s'executa una nova activitat des d'una altra en concret, ja que quan es navega per diferents activitats cal que s'implementi utilitzant Intents. En definició un Intent és la descripció abstracta d'una operació que s'executarà, és a dir, fent una crida als mètodes particulars de la classe *Activity* amb els paràmetres declarats de l'intent.

Es faran servir doncs un intent implícit quan es cridi a una nova activitat amb la instrucció *StartActivity*.

De la mateixa manera és condició indispensable que al fitxer *AndroidManifest* aquestes activitats estiguin ben declarades per poder ser cridades amb els Intents, en cas contrari obtindrem un error del sistema i l'aplicació finalment morirà durant aquestes crides o inicis d'activitat.

```
startActivity(new Intent(Intro.this, TICOcupat.class));
```

Fig 47: Exemple d'execució d'activitat definint un Intent

En els apartats següents es detallaran les diferents activitats, el seu objectiu i els mètodes més representatius que aporten les diferents funcionalitats d'una forma breu per mostrar el com s'han implementat en cada cas.

4.7.1. Activitat Intro

És l'activitat que mostra la pantalla d'inici d'entrada a l'aplicació i que ensenya un petit gràfic temporitzat durant uns segons per prosseguir posteriorment amb l'activitat principal *TICOcupat*.

Aquí hem de tenir en compte un aspecte fonamental d'una de les característiques base de l'aplicació i que hem definit durant el procés d'anàlisi, es a dir, el suport per diferents idiomes i que recaptarem durant aquesta activitat per definir, o bé l'idioma per defecte, o pel contrari l'idioma triat en execucions anteriors i que ja havia estat predefinit a les preferències de *TICOcupat*.

Com en totes les activitats heretades de *Activity* sempre es disposa d'un mètode sobrecarregat *onCreate* que serà sempre l'inici/creació de l'activitat i que en aquest cas és interessant mostrar breument per mostrar les característiques del com s'implementa l'orientació i format de la pantalla:

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);

    setContentView(R.layout.intro);
}
```

Fig 48: Mètode *onCreate* de l'activitat *Intro*

Com és pot observar en la introducció no s'ensenyarà la barra superior del telèfon (*FEATURE_NO_TITLE*), l'orientació triada sempre serà vertical i a pantalla completa. Finalment s'observa la crida al fitxer *XML* (*intro*) que representa la pantalla gràfica i esquema de l'activitat.

- Personalitza

Aquest mètode recull les preferències (*SharedPreferences*) de l'aplicació i en cas de que sigui la primera execució, evidentment encara no s'hauria configurat que idioma, es defineix el català per defecte. Finalment és refrescarà la configuració de l'idioma per que al continuar amb l'activitat *TICOcupat* ja es mostri correctament l'idioma a visualitzar.

```
private void personalitza(){
    SharedPreferences pref =PreferenceManager.getDefaultSharedPreferences(this);
    String idioma=pref.getString("listPref1","ca_ES");
    Locale locale = new Locale(idioma);
    Locale.setDefault(locale);
    Configuration config = new Configuration();
    config.locale = locale;
    getBaseContext().getResources().updateConfiguration(config,getBaseContext().getResources().getDisplayMetrics());
}
```

Fig 49: Mètode personalitza de l'activitat Intro

4.7.2. Activitat TICOcupat

La classe *TICOcupat* conforma el nucli principal de l'aplicació, i bàsicament conté el menú, les crides a les diferents activitats/classes i la funcionalitat d'activació i desactivació de l'aplicació amb la seva casuística particular.

El mètode d'inici del cicle de vida únicament mostra la pantalla gràfica i no aporta més funcionalitat que aquesta. Tenint en compte els diferents cicles de vida que hem comentat en l'apartat introductor i s'ha utilitzat *onStart* per executar unes instruccions al recuperar l'activitat provinent d'un altra. Concretament es miraran les preferències per saber si l'aplicació està connectada o desconnectada i mostrarà la icona pertinent. D'igual forma es farà servir *onRestart* per obligar a l'activitat a refrescar l'idioma

En aquest mòdul destacarem només els següents mètodes representatius:

- *onClick*

En aquest mètode detectarem les pulsacions a les icones de les diferents funcionalitats del sistema, posteriorment executarem *StartActivity*- diferents *Intents* per canviar d'activitat segons l'opció triada.

```
public void onClick(View v) {
    switch(v.getId())
    {
        case R.id.botoAfegirRegistre:
            startActivity(new Intent(TICOcupat.this, AfegirTelefon.class));
            break;
        case R.id.botoHistoric:
            startActivity(new Intent(TICOcupat.this, Historic.class));
            break;
        case R.id.botoAfegirConfiguracio:
            startActivity(new Intent(TICOcupat.this, Preferencies.class));
            break;
        case R.id.sobre:
            startActivity(new Intent(TICOcupat.this, Ajuda.class));
            break;
    }
}
```

```

        case R.id.botoOnOff:
            SharedPreferences pref1
            =PreferenceManager.getDefaultSharedPreferences(TICOcupat.this);
            SharedPreferences.Editor editor = pref1.edit();
            if (MiroPreferencies()){
                editor.putBoolean(Activacio, false);
                Conecdesco(Apagat);
            }else {
                editor.putBoolean(Activacio, true);
                Conecdesco(Ences);
            }
            editor.commit();
            break;
    }
}

```

Fig 50: Mètode *onClick* de l'activitat *TICOcupat*

El mètode rep com a paràmetre un objecte de tipus *View*, i que amb el seu *Id* correspondrà amb la identificació de la icona pulsada que s'identificarà i executarà l'activitat requerida. En aquest mètode es pot observar com es canvia el paràmetre d'activació de *TICOcupat* i es guarda a les preferències del sistema.

- **Conecdesco**

Mètode que té com a missió canviar la icona d'activació i desactivació de l'aplicació, a més de posar el telèfon en silenci –*AudioManager*–.

```

public void Conecdesco(String uri){
    AudioManager so;
    so= (AudioManager) getSystemService(Context.AUDIO_SERVICE);
    if (uri.contains(Ences))
        so.setRingerMode(AudioManager.RINGER_MODE_SILENT);
    else so.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
    ImageView estat = (ImageView)findViewById(R.id.botoOnOff);
    int imageResource = this.getResources().getIdentifier(uri, null, this.getPackageName());
    Drawable imatge = this.getResources().getDrawable(imageResource);
    estat.setImageDrawable(imatge);
}

```

Fig 51: Mètode *Conecdesco* de l'activitat *TICOcupat*

- **onBackPressed**

S'ha introduït el mètode *onBackPressed* per controlar quan pitgem la tecla de retornar enrere. Aquesta tecla navega a l'anterior activitat, i per motius obvis, no té sentit retornar a l'activitat *Intro* des de l'activitat *TICOcupat*, és per això que s'ha codificat per interceptar aquesta pulsació. Al codi del mètode, no obstant, no s'ha introduït cap acció ja que no es vol realitzar acció alguna, si no tot el contrari que no en faci cap, i menys retornar enrere, veritable motiu de la inclusió.

4.7.3. Activitat AfegirTelefon

Aquesta activitat permet afegir telèfons, definir la seva tipologia VIP o a rebutjar, modificar un telèfon de la llista introduïda, o per finalitzar eliminar-lo de la llista.

La utilització d'un adaptador específic enriqueix l'aspecte visual de la llista, en la que la inserció d'elements gràfics en cada subnivell o element de la llista ofereix un millor aspecte final de *TICOcupat*.

A continuació es mostren els trets més característics i importants de la implementació:

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    this.context=this;
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.afegir_registre);

    BaseDades BD=new BaseDades(this);
    usuari=BD.getUsuaris();
```

```
listViewUsuari = ( ListView ) findViewById( R.id.llista);
adaptador=new MeuAdaptadorAfegir(this, R.layout.expandir_layout_afegir, usuari );
listViewUsuari.setAdapter(adaptador);
}
```

Fig 52: Mètode onCreate de l'activitat AfegirTelefon

A la figura 52 es pot observar l'ús de la plantilla-layout- de registre, la qual té incorporada una llista –R.id.llista-. Així mateix es veu clarament la definició del adaptador i el com envia a l'adaptador els paràmetres de la plantilla de cada usuari, a més de la llista dels telèfons emmagatzemats en sistema.

Alguns dels mètodes més interessants i la seva implementació són:

- **EliminaLlista i ModificaLlista**

Aquests mètodes modifiquen la BD d'usuaris, amb l'acció que és clarament identificativa – Elimina i Modifica-, per posteriorment notificar a l'adaptador que les dades han canviat i que s'ha de refrescar la pantalla de la llista.

```
public void EliminaLlista(int pos){
    usuari.remove(pos);
    adaptador.notifyDataSetChanged();
}
public void ModificaLlista(Usuari user,int pos){
    usuari.set(pos, user);
    adaptador.notifyDataSetChanged();
}
```

Fig 53: Mètode EliminaLlista i ModificaLlista de l'activitat AfegirTelefon

- **AfegimTelefon**

Mètode en el que s'afegeix un número a la BD del sistema. Per introduir la dada s'ha utilitzat una *Layout* per afegir les dades necessàries del telèfon – número, *checks*, etc.-

que s'obre amb una finestra diàleg, per posteriorment tancar la finestra i refrescar la llista i afegir el telèfon a la BD.

```

public void AfegimTelefon(){
    final BaseDades BD=new BaseDades(this);
    BD.getWritableDatabase();
    final Dialog dialog = new Dialog(context);
    dialog setContentView(R.layout.afegir_layout);
    dialog.setTitle(R.string.titolAfegir);
    Button botoafegir = (Button) dialog.findViewById(R.id.botoafegirtelefonlayout);
    Button botocancelar = (Button) dialog.findViewById(R.id.botocancellalayout);
    botoafegir.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            String txtTelefon=((TextView)dialog.findViewById(R.id.Afegirtelefonlayout)).getText().toString();
            Boolean vip=((CheckBox)dialog.findViewById(R.id.checkBox_viplayout)).isChecked();
            Boolean valida=Validar(dialog,txtTelefon,vip);
            if(valida){
                BaseDades BD=new BaseDades(context);
                Usuari usu=new Usuari(txtTelefon,vip);
                BD.addUsuari(usu);
                usuari.add(usu);
                adaptador.notifyDataSetChanged();
                dialog.dismiss();
            }else{
                dialog.dismiss();
            }
        }
    });
    botocancelar.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            dialog.dismiss();
        }
    });
    dialog.show();
}
}

```

Fig 54: Mètode EliminaLlista i ModificaLlista de l'activitat AfegirTelefon

4.7.4. Activitat Historic

A l'activitat històric es mostrarà una llista personalitzada amb un adaptador, molt semblant a l'activitat anterior en la seva estructura i que no aporta més complexitat que la de mostrar la llista, a excepció d'esborrar tots els registres emmagatzemats en sistema. Tenint en compte que els mètodes d'aquesta activitat no aportem diferències d'implementació i que aquests no aporten valor, s'ha decidit no comentar-ne cap.

4.7.5. Activitat Preferències

Activitat que gestiona les preferències de l'aplicació i que està personalitzada mitjançant una plantilla específica- *R.xml.preferencies*-.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    addPreferencesFromResource(R.xml.preferences);

    ListPreference idioma = (ListPreference) findPreference("listPref1");

    idioma.setSummary(idioma.getEntry());

    idioma.setOnPreferenceChangeListener(new Preference.OnPreferenceChangeListener() {

        public boolean onPreferenceChange(Preference preference, Object o) {
            Locale locale = new Locale(o.toString());
            Locale.setDefault(locale);
            Configuration config = new Configuration();
            config.locale = locale;
            getBaseContext().getResources().updateConfiguration(config, getBaseContext().getResources().getDisplayMetrics());
            refresh();
            return true;}
    });
}

```

Fig 55: Mètode onCreate de l'activitat Preferencies

A la figura 55 es mostra com s'afegeix una plantilla en forma XML com a preferències del sistema. A més es control·larà la llista dels idiomes i s'implementarà un mètode – *onPreferenceChange*– que detectarà el canvi d'idioma i defineix l'idioma que ha de servir *TICOcupat*.

- Refresh

Aquest mètode refresca l'activitat després de canviar d'idioma per defecte, i per que es visualitzi correctament en aquesta activitat.

Inicialment es finalitza l'activitat per cridar-la de nou amb una creació d'objecte *Intent* i un inici de l'activitat.

```

private void refresh() {
    finish();
    Intent Intent = new Intent(Preferencies.this, Preferencies.class);
    startActivity(Intent);
}

```

Fig 56: Mètode refresh de l'activitat Preferencies

4.7.6. Activitat Ajuda

A ajuda es presentarà l'ajuda mínima de l'aplicació. S'ha adoptat la solució de mostrar-la en format *HTML* amb l'objectiu de que aquesta informació pugui ser fàcilment actualizable i enriquida a futur amb elements dinàmics i/o imatges.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN);

    setContentView(R.layout.ajuda);
    Spanned sp = Html.fromHtml( getString(R.string.ajuda));
    TextView text = (TextView)findViewById(R.id.textAjuda);
    text.setText(sp);
}
```

Fig 57: Mètode onCreate de l'activitat Ajuda

Aquesta ajuda es visualitzarà a pantalla completa, sense barra superior de telèfon i en sentit vertical. Així mateix es defineix l'objecte HTML emmagatzemat a la cadena ajuda que s'insertarà en un objecte tipus text.

4.7.7. Activitat Accio

A aquesta activitat es centren totes les accions que l'aplicació executarà segons les especificacions de les funcionalitats i les configuracions escollides. Concretament com a accions tipus s'executarà l'enviament de missatges de cortesia en resposta als usuaris VIP, eliminarà la trucada del *log*¹¹ del telèfon i penjarà la trucada rebuda.

- **enviarSMS**

Mètode d'enviat d'un missatge SMS. Es farà servir el tipus *SmsManager* per enviar el missatge segons el text definit a les preferències, que poden ser personalitzades o les aportades per la pròpia aplicació. Finalment apareixerà un missatge tot informant – *Toast*– de que s'ha enviat correctament o pel contrari que hi han hagut problemes d'enviament.

```
public void enviarSMS(String numero){
    try {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(numero, null, sms, null, null);
        Toast.makeText(context, R.string.SMSEnviat,Toast.LENGTH_LONG).show();
    } catch (Exception e) {
        Toast.makeText(context,R.string.SMSError,Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
}
```

Fig 58: Mètode enviarSMS de l'activitat Accio

(11) Log – Registre informàtic d'events durant un temps acotat

- **eliminarLog**

EliminarLog utilitzarà, com s'ha comentat prèviament, el concepte de *Content Provider* per accedir al propi *log* de trucades del telèfon. L'objectiu serà esborrar les trucades malicioses del propi registre *-log-* del telèfon.

```
public void eliminarLog(String numero){
    @SuppressWarnings("unused")
    Cursor cursor = null;
    Uri uri=Uri.parse("content://call_log/calls");
    try{
        cursor = context.getContentResolver().query( uri,null,null,null,"DATE DESC");
        context.getContentResolver().delete(CallLog.Calls.CONTENT_URI,CallLog.Calls.NUMBER+"="+numero+"",null);
    } catch(Exception ex){
    }
}
```

Fig 59: Mètode *eliminarLog* de l'activitat *Accio*

- **penjaTelefon**

Mètode que executarà el penjat del telèfon al rebre la trucada quan està activada l'aplicació o pel contrari al rebre una trucada definida com a no desitjada i en estat desactivat. Per realitzar l'acció invocarem a l'acció *endCall* definida a AIDL/Interfície definida al codi i comentada en punts anteriors de l'apartat d'implementació.

```
public void penjaTelefon(){
    TelephonyManager tm = (TelephonyManager)
    context.getSystemService(Context.TELEPHONY_SERVICE);
    try {
        @SuppressWarnings("rawtypes")
        Class c = Class.forName(tm.getClass().getName());
        @SuppressWarnings("unchecked")
        Method m = c.getDeclaredMethod("getITelephony");
        m.setAccessible(true);
        com.android.internal.telephony.ITelephony telephonyService = (ITelephony) m.invoke(tm);
        telephonyService.endCall();
    } catch (Exception e) {
    }
}
```

Fig 60: Mètode *penjaTelefon* de l'activitat *Accio*

4.7.8. Classe **MeuAdaptadorAfegir**

Aquesta classe és probablement la més complexa i elaborada de totes les que componen l'aplicació. Bàsicament el seu objectiu és inflar la vista, dins un layout específic, que es rebuda pel mètode i amb cada component de la llista, en un de més personalitzat amb icones específiques per a cada tipologia de telèfon a presentar.



Fig 62: Pantalla afegir

```
public View getView(int position, View convertView, ViewGroup parent) {
    final int pos=position;
    LayoutInflater inflater = (LayoutInflater)
    context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View rowView = inflater.inflate(R.layout.expandir_layout_afegir, parent,false);
    TextView textView = (TextView) rowView.findViewById(R.id.telefonLayout);
    Usuari usuari = getItem( position );
    textView.setText(usuari.getnumero().toString());
    ImageView tipologia = (ImageView) rowView.findViewById(R.id.tipus);
    uri = "drawable/" + usuari.getImagetipus();
    int imageResource = context.getResources().getIdentifier(uri, null,
    context.getPackageName());
    Drawable imatge = context.getResources().getDrawable(imageResource);
    tipologia.setImageDrawable(imatge);
    if(usuari.getvip()){
        tipologia = (ImageView) rowView.findViewById(R.id.smsLayout);
        uri = "drawable/sms";
        imageResource = context.getResources().getIdentifier(uri, null,
        context.getPackageName());
        imatge = context.getResources().getDrawable(imageResource);
        tipologia.setImageDrawable(imatge);
    }
}
```

Fig 61: Inflat de la vista

```
ImageView tipologia1 = (ImageView) rowView.findViewById(R.id.BorrarRegistre);
uri = "drawable/borrar";
imageResource = context.getResources().getIdentifier(uri, null, context.getPackageName());
imatge = context.getResources().getDrawable(imageResource);
tipologia1.setImageDrawable(imatge);
tipologia1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setMessage(R.string.titolEliminar)
        .setCancelable(false)
        .setPositiveButton(R.string.Si, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                BaseDades BD=new BaseDades(context);
                Usuari us=RecuperarUsuari(pos);
                BD.borrarUsuari(us);
                AfegirTelefon elimina=new AfegirTelefon();
                elimina.EliminaLista(pos);
            }
        })
        .setNegativeButton(R.string.No, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }
});
```

Fig 63: exemple d'inserció d'icona amb accions

4.7.9. Classe MeuAdaptadorHistoric

En essència aquesta classe és força semblant en la implementació que l'anterior. Rep la llista històric i la inflarà tenint en compte el *layout* personalitzat tal com ho fa l'anterior adaptador.

4.7.10. Classe ServeiTrucada

La classe ServeiTrucada hereta de *BroadcastReceiver*, classe la que com s'ha dit anteriorment actua com un component que permet registrar events del sistema i que traslladarà les seves capacitats a ServeiTrucada, per així poder detectar quan existeix un event de trucada (trucada, penjat, etc.).

Al heretar de *BroadcastReceiver* conté el mètode necessari següent:

```
public void onReceive(Context context, Intent intent) {
    this.context=context;
    BD=new BaseDades(context);
    bundle = intent.getExtras();
    if(null == bundle)
        return;
    String Estat=bundle.getString("state");
    if(Estat.contains("RING")){
        String NumeroTelefon=bundle.getString("incoming_number");
        Usuari user=comprovatelefon(NumeroTelefon);
        GeneraAccio(user);
    }
}
```

Fig 64: Mètode *onReceive* de l'activitat *ServeiTrucada*

Cada vegada que es produeix una trucada, penjat o rebuig s'executarà el mètode *onReceive*, a on es mirarà si l'event produït es tracta d'una trucada entrant. En el cas afirmatiu s'extraurà el número de telèfon d'entrada, es comprovarà si existeix un registre al sistema amb aquest número i es cridarà al mètode per generar les accions corresponents segons l'estat de l'aplicació.

- **comprovaTelefon**

Mètode que obtindrà una llista de telèfons de la BD cridant al mètode *getUsuaris* i que comprovarà si existeix un registre al sistema.

```
private Usuari comprovaTelefon(String numeroTelefon) {
    Usuari usuari;
    if (numeroTelefon!=null){
        usuari=new Usuari(numeroTelefon,false);
        List<Usuari> llistaUsuaris = new ArrayList<Usuari>(BD.getUsuaris());
        for (Usuari user: llistaUsuaris)
            if(user.getnumero().toString().contains(numeroTelefon))
                usuari=user;
        }else {
            usuari=new Usuari("Desconegut",true);
        }
    return usuari;
}
```

Fig 65: Mètode *comprovaTelefon* de l'activitat *ServeiTrucada*

- **GeneraAccio**

GeneraAccio revisarà l'estat de les preferències i de l'aplicació per iniciar les diferents accions definides a *TICOcupat* i que seran cridades com a un objecte/acció de la classe acció. La classe en si no implementa cap novetat i es limita a condicionar segons les variables establertes per cridar a l'acció corresponent.

4.7.11. Classe Trucada

Classe que moldejarà un objecte del tipus *trucada* i que posteriorment es farà servir per emmagatzemarà al registre històric de l'aplicació.

4.7.12. Classe Usuari

Classe que moldejarà un objecte del tipus *usuari* i que posteriorment es farà servir per emmagatzemarà al registre d'usuaris de l'aplicació.

4.8. Persistència

Com ja s'ha comentat anteriorment a *TICOcupat* es faran servir dos sistemes diferents per implementar la persistència: les preferències del sistema i SQLITE.

D'una banda la classe *SharedPreferences* ofereix un *framework*¹² que permet salvar i recuperar parells de valors persistent de tipus de dades primitives (booleans, enters, cadenes, etc.), oferint que les dades siguin persistents inclòs si l'aplicació mor. Aquesta tipologia de persistència s'utilitza per emmagatzemar les diverses opcions del sistema i per mitja de la classe *Preferències*-punt 4.5.5- que hereta de *PreferenceActivity*, i que no comentarem en detall ja que es tracta en un apartat anterior ja comentat.

Per a implementar la classe de Base de dades s'ha utilitzat SQLITE el qual és un sistema de gestió de base de dades relacional força conegut. Una de les grans virtuts, i per això es utilitza en la programació de dispositius mòbils, és el seu petit tamany (~275 Kbyte); d'igual forma la velocitat amb la que s'accedeix, mitjançant rutines i funcions, a les dades està bastant optimitzada. Tal com podem observar en la figura l'arquitectura és modular i l'API de la interfície permet fer crides que posteriorment els sistema modular permeabilitzarà a la resta de capes de servei.

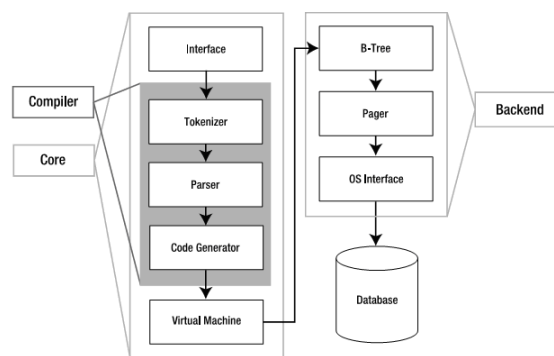


Fig 66: Arquitectura SQLITE

4.8.1. Classe BaseDades

Classe amb la missió de emmagatzemar tots el telèfons i històric de l'aplicació. *TICOcupat* farà servir una única base de dades per emmagatzemar en dos taules els diferents telèfons definits per l'usuari amb les seves tipologies, així com l'històric de trucades. Finalment destacar que no es necessari, segons requisits, relacionar en cap cas les diferents dades pel que la funcionalitat bàsica de SQLITE serà guardar els valors.

A continuació es mostraran alguns mètodes representatius de la classe i que alguns són semblants a les dues taules utilitzades en l'aplicació.

- **onCreate**

Al crear la classe heretada de *SQLiteOpenHelper* es crearan les taules en cas de que sigui la primera execució.

(12) Framework – Estructura de suport prèviament definida per que una aplicació pugui fer servir els seus mètodes ja escrits

```

public void onCreate(SQLiteDatabase db) {
    try{
        db = SQLiteDatabase.openDatabase(NOMBASE, null,0);
    } catch(Exception e)
    {
        String CREAM_TAUOLA1 = "CREATE TABLE " + TAULATELEFONS + "(" + ID1 + "
INTEGER PRIMARY KEY," + TELEFON1 + " TEXT," + CHECKVIP + " INTEGER)";
        String CREAM_TAUOLA2 = "CREATE TABLE " + TAULAHISTORIC + "(" + ID2 + "
INTEGER PRIMARY KEY," + TELEFON2 + " TEXT," + TIPUS + " TEXT," + DATA + "
TEXT)";
        db.execSQL(CREAM_TAUOLA1);
        db.execSQL(CREAM_TAUOLA2);
    }
}

```

Fig 67: Creació de taules BD

- **addUsuari**

El mètode *Addusuari* afegirà un objecte de la classe usuari a la taula *TAULATELEFONS*.

```

public void addUsuari(Usuari usuari) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues valors = new ContentValues();
    valors.put(TELEFON1, usuari.getnumero());
    valors.put(CHECKVIP, usuari.getvip());
    db.insert(TAULATELEFONS, null, valors);
    db.close();
}

```

Fig 68: Afegir dades BD

- **updateUsuari**

Aquest mètode actualitzarà un usuari quan es modificat per l'aplicació.

```

public int updateUsuari(Usuari usuari) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues valors = new ContentValues();
    valors.put(TELEFON1, usuari.getnumero());
    valors.put(CHECKVIP, usuari.getvip());
    return db.update(TAULATELEFONS, valors, ID1 + " = ?",
        new String[] { String.valueOf(usuari.getID()) });
}

```

Fig 69: Actualització dades BD

- **borrarUsuari**

BorrarUsuari permetrà eliminar de la BD a l'usuari triat en l'aplicació.

```

public int updateUsuari(Usuari usuari) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues valors = new ContentValues();
    valors.put(TELEFON1, usuari.getnumero());
    valors.put(CHECKVIP, usuari.getvip());
    return db.update(TAULATELEFONS, valors, ID1 + " = ?",
        new String[] { String.valueOf(usuari.getID()) });
}

```

Fig 70: Eliminació dades BD

Finalment i sense entrar en més detalls s'implementaran mètodes que retornen una llista d'usuaris, així com tots els anteriors referint-se a les trucades i el seu històric.

5. Fase de proves

En aquesta etapa del procés de creació de l'aplicació TICOcupat es verificarà el correcte funcionament de totes les funcionalitats de l'aplicació i la seva correcció per obtenir un producte final adaptat als requisits del disseny.

Per verificar que totes les funcionalitats funcionen correctament s'ha realitzat una triple verificació:

Test d'emulació en dispositiu virtual –AVD-.

Test en dispositiu físic Android –Samsung Galaxy Mini-.

Lliurament de l'aplicació a dos usuaris per a la verificació de funcionalitats.

5.1. Testes realitzats

A continuació detallarem els diferents tests realitzats i els seus resultats:

Navegació pels diferents menús de la interfície principal. **Resultat esperat**

Afegir registre de telèfon. **Resultat esperat**

Modificació i eliminació del registre. **Resultat esperat**

Navegació endarrere des de l'opció Afegir. **Resultat esperat**

Eliminar tots els registres històrics. **Resultat esperat**

Navegació endarrere des de l'opció Històric. **Resultat esperat**

Activació de les diferents opcions de preferències. **Resultat esperat**

Navegació endarrere des de l'opció de preferències. **Resultat esperat**

Canvi d'idioma. **Resultat esperat**

Activació de l'aplicació. **Resultat esperat**

Bateria de trucades amb les diferents opcions de les preferències. **Resultat esperat a excepció d'una incidència en la recepció de trucades amb telèfon ocult.**

5.2. Modificacions del producte final

Fruit de la triple verificació comentada en l'apartat anterior s'han reportat una sèrie d'incidències o comentaris que han estat incloses en el producte final i que detallem a continuació:

Lletres del menú principal que han de ser adaptades als diferents tamanyes de pantalla.

Necessitat d'introduir numeracions menors o superiors a 9 xifres.

Comportaments impredecibles per trucades desconegudes.

5.3. Avaluació de l'aplicació

Després de fer les correccions dels errors detectats en l'etapa de verificació, i de forma iterativa, s'ha tornat a avaluar l'aplicació per corroborar pels diferents provadors que les incidències corregides són les detectades i que *TICOcupat* funciona correctament amb les modificacions denunciades.

6. Producte final

El funcionament del producte final es pot observar en el vídeo lliurat amb la memòria del *TFC*, de totes formes annexem les principals imatges de *TICOcupat* ja funcionant en un dispositiu real i fruit de la consecució de totes les fases detalles en aquest projecte.



Fig 71: Aplicació TICOcupat en funcionament

6.1. Dificultats.

Les majors dificultats en el disseny del projecte han estat curiosament les associades a la maquetació de les interfícies gràfiques, tant en la definició d'usabilitat com en l'adaptació i programació utilitzant el llenguatge *XML*.

D'altra banda, destaquem també com a dificultat la poca documentació trobada sobre la interacció dels processos del telèfon i que en l'aplicació *TICOcupat* estan presents en la detecció de trucades, etc.-.

La planificació temporal també ha comportat una dificultat en la codificació de les funcionalitats definides inicialment, i s'ha tingut que eliminar la de menys pes -accés amb contrasenya- per ajustar-se a les fites temporals del projecte.

6.2. Implementacions futures

Dins d'aquest apartat d'implementacions futures detallarem les evolucions futures que oferirà l'aplicació, que per la limitació temporal del projecte no han pogut ser implementades en primera instància i que són:

Enviament de correus electrònics de cortesia. Actualment l'aplicació respon amb un missatge SMS i aquesta funcionalitat permetria definir si volem enviar, d'afegitó, un *email via Internet* informant de l'ocupació de l'usuari.

Resposta de trucades amb missatges de veu programats. L'aplicació permetria amb aquesta funcionalitat respondre als telèfons VIP d'una forma més amigable amb la veu del propi usuari del telèfon.

Finalment el suport gràfic per a resolucions superiors a 240x320.

7. Conclusions finals

A l'inici del projecte es plantejaven una sèrie d'objectius i reptes que amb tota humilitat s'han assolit de forma satisfactòria, tant els propis del disseny de l'aplicació com els que estan relacionats amb l'aprenentatge de les habilitats necessàries per dur a terme tots els processos associats amb el seu disseny.

La posada en pràctica dels coneixements adquirits sobre els cicles de vida en la creació del programari ha permès identificar la criticitat d'aquest fet com a punt més important en la gestió d'un projecte d'aquest tipus. Planificar, identificar correctament els requisits inicials dels recursos i estructurar correctament totes les tasques a realitzar són vitals per assolir amb èxit les fites de qualsevol projecte.

Finalment ha estat important comprendre l'aproximació al usuari amb la metodologia DCU i com portar-la a terme durant tot el procés de la creació de l'aplicació. Com intervé l'usuari, que aporta, quina informació és rellevant, etc. són informacions que esdevenen molt útils durant la codificació final del programari.

En resum, l'aprenentatge de tots aquest punts comentats aporten un coneixement extra al que ja es tenien a l'inici del projecte i puc assegurar la meua satisfacció general del resultat final, cicle de vida de tot el projecte i del treball realitzat.

8. Bibliografia

Lee, Wei-Meng. Beginning Android 4. Application Development. Indianapolis, IN 46256. Ed. John Wiley & Sons, Inc . 2012. 564 p. ISBN: 978-1-118-19954-1 [data de consulta: 10/5/2013].

Rogers, Rick. Android Application Development. Sebastopol. Ed. O'Reilly Media, Inc. 2009. 336 p. ISBN: 978-0-596-52147-9 [data de consulta: 10/5/2013].

Steele, James. The Android developer's cookbook. Boston. Ed. Pearson Education, Inc. 2011. 360 p. ISBN-13: 978-0-321-74123-3. [data de consulta: 10/5/2013].

Collins, Charlie. Android in Practice . Shelter Island, NY 11964. Ed. Manning. 2012. 647 p. ISBN: 9781935182924. [data de consulta: 10/5/2013].

Haseman, Chris. Android Essentials. Berkeley, CA 94705. Ed. Apress. 2008. ISBN: 13 (electronic): 978-1-4302-1063-4. [data de consulta: 10/5/2013].

9. Webgrafia

Expert Android and Eclipse development knowledge. [en línia].
<http://www.vogella.com/android.html> [data de consulta: 10/5/2013].

Curso de Programación en Android. [en línia].
http://www.sgoliver.net/blog/?page_id=2935 [data de consulta: 10/5/2013].

Android developing. [en línia]. <http://developer.android.com/guide/components/index.html> [data de consulta: 10/5/2013].

StackOverFlow. [en línia]. <http://stackoverflow.com> [data de consulta: 18/5/2013].

Android Service Tutorial. [en línia].
<http://www.vogella.com/articles/AndroidServices/article.html> [data de consulta: 18/5/2013].

Interface ITelephony. [en línia].
http://www.androidjavadoc.com/1.0_r1_src/com/android/internal/telephony/ITelephony.html [data de consulta: 18/5/2013].

Android Desk. [en línia]. <http://androiddesk.wordpress.com/tag/itelephony-aid/> [data de consulta: 10/4/2013].