



Universitat
Autònoma
de Barcelona



(2876-3: ONTHEBUS)

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Daniel del Arco Gómez
i dirigit per
Jordi Roig de Zárate
Bellaterra, 12 de Setembre de 2011.

El sotasignat, Jordi Roig de Zárate

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Daniel del Arco Gómez

I per tal que consti firma la present.



Signat: Jordi Roig de Zárate

Bellaterra, 12 de Setembre de 2011.

Índex

1	Introducció.....	5
1.1	Motivació	5
1.2	Que volem fer, on volem arribar	6
2	Estudi de viabilitat	7
2.1	Estat de l'art.....	7
2.1.1	Estat actual del mercat	7
2.1.2	Sistemes de guiatge.....	7
2.1.3	Sistemes de ajuda a l'ús de transport públic.....	8
2.1.4	Sistemes d'ajuda al discapacitat.....	8
2.2	Tecnologia a emprar	9
2.2.1	El sistema operatiu Android	9
2.2.2	Serveis de Cloud Computing.....	13
2.2.3	Subversion	14
2.2.4	Entorn de desenvolupament	14
2.3	Planificació del Projecte	16
2.3.1	Diagrama de Gantt.....	18
2.4	Disseny Universal.....	18
2.5	Anàlisi de Terminals.....	23
2.5.1	Botons físics en dispositius Android	23
2.5.2	Diferenciació entre botons físics i tàctils.....	23
2.5.3	Disposició dels botons	24
2.5.4	Anàlisi.....	24
2.5.5	Properes tendències.....	28

2.6	Conclusions estudi de viabilitat	29
3	Mòdul de Dades (BDD)	30
3.1	Informació ciutats.....	30
3.1.1	Informació general	30
3.1.2	Obtenció de la informació	32
3.2	Base de dades.....	36
3.2.1	Base de dades SQLite.....	36
3.2.2	Bases de dades del projecte	37
3.3	Gestió de memòria del telèfon mòbil.....	51
3.3.1	Gestió memòria interna.....	52
3.3.2	Gestió memòria externa.....	53
3.4	Temps Espera Parada	54
3.4.1	Informació dels serveis	54
3.4.2	Mètodes de consulta al servei:.....	55
3.4.3	Algoritme de consulta al servei:	56
4	Proves i resultats	58
4.1	Proves Unitàries.....	58
4.1.1	Característiques de les bases de dades (Model Ciutats)	58
4.1.2	Rendiment de les bases de dades en els diferents mòbils	59
4.1.3	Favorits estudi	61
4.1.4	Temps espera parada	61
4.2	Proves i resultats de camp.....	65
4.2.1	Proves i Resultats de Localització	65
4.2.2	Proves i Resultats de Guiatge	67
5	Conclusions i possibles ampliacions	71
5.1	Conclusions Personals	71

5.2	Conclusions Generals.....	72
5.3	Possibles Ampliacions Mòdul de Dades	73
5.4	Possibles Ampliacions de l'aplicació.....	74
6	Bibliografia.....	75

1 Introducció

1.1 Motivació

Aquest projecte neix de les creixents necessitats de mobilitat actuals. Cada dia ens movem més i més lluny, per estudiar, treballar, comprar, visites al metge, oci, etc. Aquestes necessitats són també presents en els grans nuclis urbans, on pot ser necessari cobrir grans distàncies per poder arribar a un determinat destí. Moltes persones utilitzen el transport privat com a forma de desplaçament, però cada cop més gent opta per el transport públic.

Si bé els mitjans de transport públic eren anys enrere considerats una forma de transport per persones amb pocs recursos econòmics, com estudiants o jubilats, cada dia més gent opta per aquest tipus de transport per diversos motius, que poden ser econòmics o ecològics d'entre d'altres. Els parcs automobilístics de les ciutats no paren de créixer, i es comencen a fer patents problemes derivats de d'aquest augment de vehicles, com, l'augment de la pol·lució ambiental, la dificultat per aparcar en nuclis molt concorreguts o la saturació d'algunes vies. És per això que les autoritats cada cop són més conscients de la necessitat de pacificar el trànsit creant zones peatonals, zones trenta i impulsar el transport públic com a forma preferent de desplaçament, creant noves línies de metro i bus a les ciutats.

Aquest increment en el número de línies d'autobús ha provocat que actualment sigui molt complex determinar quina és la línia d'autobús que s'ha d'agafar per arribar al nostre destí, determinar quins transbords podem fer o quina es la disponibilitat horària de cada línia. Per això, les companyies d'autobús que operen en cada zona, proporcionen tota aquesta informació en una pagina web, o en marquesines especialment ubicades a les parades. Les pagines web però, són de difícil consulta des del carrer, especialment si opera més d'una companyia a la zona, i la informació no es troba unificada en una única pàgina. Les marquesines en canvi, tenen més aviat una funció de validació, ja que podem determinar si a la parada on ens trobem hi circula la línia que volem agafar, encara que aquestes marquesines poden ser vandalitzades. El problema d'accedir a aquesta informació, es pot agreujar més encara en persones que no estan familiaritzades amb la zona o persones discapacitades, així com persones d'edat avançada.

Per sort, en el món actual, la presencia de terminals mòbils amb accés a Internet és tota una realitat. Aquestes plataformes, a més de disposar de connexió a Internet, incorporen tota

mena de sensors, com brúixoles, giroscopis, acceleròmetres i GPS, una molt bona combinació per poder desenvolupar aplicacions que solucionin aquestes necessitats creixents.

1.2 Que volem fer, on volem arribar

El que és vol aconseguir, es unificar tota la informació referent a les línies d'autobús de diverses ciutats en una única aplicació, amb una interfície universal de fàcil accés i ús, per tal de poder guiar al usuari fins a la parada d'autobús, donar informació dels temps d'espera, i informar-lo durant el trajecte amb l'autobús, indicant a l'usuari quan ha de baixar de l'autobús per arribar a un determinat destí.

Es vol aconseguir una aplicació intuïtiva, fàcil d'utilitzar, tant per aquelles persones acostumades a fer servir dispositius mòbils, com per aquelles persones que hi tenen algunes dificultats, com les persones grans.

Degut a que els terminals mòbils actuals no disposen de software adaptat per a persones amb discapacitats visuals, volem utilitzar el disseny universal per fer el més accessible possible la nostre aplicació, perquè incloses aquestes persones puguin fer-la servir de forma fàcil, senzilla i pràctica.

No es vol que la aplicació es quedi al nivell d'una prova pilot en una ciutat concreta, volem estendre la possibilitat d'utilitzar l'aplicació en diverses ciutats del món, per presentar una solució realista al problema del guiatge en autobús dins les ciutats.

2 Estudi de viabilitat

2.1 Estat de l'art

En aquesta secció es comentarà l'estat de l'art de les aplicacions de guiatge en transport públic accessible.

En una primera part s'explicarà l'estat actual d'aquest tipus d'aplicacions, per a passar després a detallar l'estat de cadascuna de les vessants.

2.1.1 ESTAT ACTUAL DEL MERCAT

En els últims anys, els sistemes de guiatge personal, han cobrat especial importància, es poden trobar nombrosos desenvolupaments de sistemes de guiatge, tant gratuïts com de pagament, així com exemples de implementacions senzilles de sistemes de seguiment de rutes de codi obert.

No obstant això, aquests sistemes solen estar orientats al desplaçament autònom (sense considerar l'ús de sistemes de transport públic), i no s'han trobat referències a sistemes de guiatge per a persones discapacitades.

Amb l'arribada de dispositius de cost relativament reduït, amb la tecnologia necessària per a implementar aquests sistemes de guiatge, és d'esperar que el nombre d'aplicacions d'aquest àmbit augmentin amb rapidesa, sobre tot tenint el compte els costos reduïts, per al seu desenvolupament, així com l'ampli ventall d'usuaris.

2.1.2 SISTEMES DE GUIATGE

Els sistemes de guiatge, que estan àmpliament difosos en les plataformes, es poden prendre com a referència gran nombre d'aplicacions com GoogleMaps, TomTom o AndNav.

Aquestes aplicacions es basen en la utilització de sistemes de cartografia externs al dispositiu que es descarreguen a través de serveis online utilitzant la connexió a Internet del dispositiu o bé incorporen mapes juntament amb l'aplicació, tant gratuïts com de pagament, així com el sistema civil de GPS (Geo Posicionament per Satèl·lit).

En aquest àmbit a l'hora de utilitzar tant els sistemes cartogràfics, com el sistema de GPS s'han trobat diversos problemes, alguns d'ells àmpliament coneguts i d'altres amb escassa

literatura disponible, que han fet que s'hagin hagut de modificar diverses especificacions de desenvolupament. Totes aquestes problemàtiques seran àmpliament detallades en els respectius punts.

2.1.3 SISTEMES DE AJUDA A L'ÚS DE TRANSPORT PÚBLIC

Els últims dos anys s'ha incrementat la demanda d'ajuda a l'hora de desplaçar-se per la ciutat en transport públic, degut a l'alt volum de tràfic produït pel transport privat i les despeses associades a ell, així com per la poca informació de qualitat proporcionada per les pròpies companyies de transport.

Aquesta mala gestió per part de les companyies de transport, està essent lentament suplida mitjançant aplicacions proporcionades per companyies de desenvolupament de software (UrbanStep), projectes universitaris (ValenciaBus), o aplicacions desenvolupades per la pròpia companyia de transports (TMB Virtual).

Paradoxalment, per tal implementar els sistemes d'ajuda a l'ús del transport públic ha sigut necessari fer mineria de dades amb la informació proporcionada per les companyies de transport públic, ja que la gran majoria de companyies no proporcionen la informació en un format fàcil de manipular informàticament. Per exemple, en alguns casos ha sigut necessari programar un software per tal d'extreure automàticament les dades de les línies i parades d'autobusos a partir de la informació mostrada en la pàgina web de la companya, o ha calgut fer una transformació dels sistemes de coordenades utilitzats per les companyies a un sistema únic per a que totes les dades funcionin correctament a OnTheBus.

2.1.4 SISTEMES D'AJUDA AL DISCAPACITAT

Els sistemes d'ajuda al discapacitat en dispositius basats en *Android* es troben en fases molt verdes de la seva evolució. Tot i que varies funcionalitats que són necessàries per a implementar un sistema d'aquest tipus es troben disponibles mitjançant la pròpia arquitectura proporcionada pel sistema, aquestes manquen de la estabilitat i maduresa necessàries per a fer aplicacions comercials per a col·lectius de discapacitats.

D'altra banda les limitacions no sempre venen donades per a l'arquitectura, sinó que pot ser que sigui el propi dispositiu el que dificulti l'accessibilitat, degut a la utilització de components més assequibles i de baixa qualitat per a abaratir el preu final del dispositiu.

2.2 Tecnologia a emprar

En aquest apartat es defineixen les tecnologies requerides per al desenvolupament d'aquest projecte.

2.2.1 EL SISTEMA OPERATIU ANDROID

Android és un sistema operatiu per a dispositius mòbils en general, com poden ser telèfons, tablets, reproductors MP3, etc, encara que recentment s'està expandint el seu ús per a tot tipus de dispositius, com poden ser televisors o fins i tot microones o rentadores.

Inicialment, la companya *Android Inc.* va començar el seu desenvolupament, i posteriorment va ser comprada per Google a l'any 2005. La seva llicència és *Apache Software Licence*, de codi obert.

Google, a l'any 2007, va fundar la *Open Handset Alliance (OPH)*, que es tracta d'una aliança comercial formada per 78 companyies que desenvolupen estàndards oberts per dispositius mòbils. En formen part, entre d'altres, *HTC*, *Motorola*, *NVIDIA* i *Samsung*.

Al novembre de 2007, es va publicar la primera versió de la API d'*Android*, que és una interfície de programació d'aplicacions que conté el conjunt de procediments en forma de llibreria de software que permet utilitzar els components dels terminals *Android* (senyors, pantalla, SO, etc) per desenvolupar aplicacions.

Existeixen altres sistemes operatius per a dispositius mòbils, com per exemple *IOS* de l'empresa *Apple*, *BlackBerry OS* de *RIM (Research in Motion)*, *Symbian* de *Nokia* o *Windows Phone* de *Microsoft*. Un dels motius de que s'hagi escollit el sistema operatiu *Android* és que ofereix més possibilitats, facilitat de programació i garanties d'èxit que qualsevol dels altres esmentats.

Per exemple, en el cas de *IOS* el *kit* de desenvolupament només està disponible per a ordinadors amb sistema operatiu *MAC OS*, també fabricat per *Apple*, i no tots els integrants del grup de OnTheBus en posseeixen un. *Symbian* era el sistema operatiu que més quota de mercat posseïa en el moment de la decisió, però en els últims anys havia anat perdent usuaris i quedant-se antiquat: fins i tot la companya que el desenvolupa, *Nokia*, havia començat a desenvolupar d'altres sistemes operatius per substituir-lo. *Windows Phone* era un sistema

operatiu massa nou i no existia molta literatura per poder aprendre'n ràpidament el llenguatge i sistema de programació.

Però el que va acabar decidint que es faria servir *Android* és el ràpid creixement i acceptació que estava tenint, que feia presagiar que en pocs mesos seria el sistema operatiu més utilitzat. A més a més, al ser un sistema operatiu ofert a terminals de diferents marques, existeix una àmplia varietat de dispositius de diferents preus, pel que es va pensar que seria molt més fàcil arribar a tots els tipus d'usuaris que es pretenia: d'elevada edat, estudiants, persones amb problemes d'accessibilitat o no.

2.2.1.1 Versions de la API d'Android

Actualment existeixen 13 versions de la API d'*Android*. Cada vegada que apareix una nova versió, incorpora noves funcionalitats i s'actualitza constantment. A la següent *Taula 1* es mostren les diferents versions i el seus noms.

Platform Version	API Level	VERSION_CODE
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Taula 1 - Versions de la API d'Android

2.2.1.2 Justificació de la versió de la API utilitzada

Durant les primeres reunions de grup, es va veure que els seus membres ja havien començat a fer projectes de prova per anar aprenent a programar per *Android*, però no tots havien fet servir la mateixa versió. Es va investigar i es va veure que en principi no hi havia cap problema amb qualsevol de les versions disponibles en aquell moment, ja que, malgrat que a la API s'indicava que algunes funcions no estaven disponibles per les primeres versions, en principi cap d'elles semblava que anés a suposar un inconvenient per al desenvolupament del projecte. Però davant de tantes versions diferents, calia posar-se d'acord en utilitzar la mateixa versió tots els components del grup, per evitar problemes que poguessin sorgir més endavant.

Es van mirar les estadístiques de les versions que incorporaven els terminals a la pàgina web de la API, on s'observà que gairebé el 80% dels terminals tenien una versió igual o superior a *Android* 2.1. A més a més, es va veure que molts fabricants havien anunciat actualitzacions per els terminals que tenien al mercat amb les versions 1.5 i 1.6.

Aleshores, es va prendre la decisió de que la versió mínima de la API que suportaria *OnTheBus* seria *Android* 2.1 en un inici.

A la següent *Figura 1* i *Taula 2*, es poden veure les estadístiques actuals de les versions utilitzades.

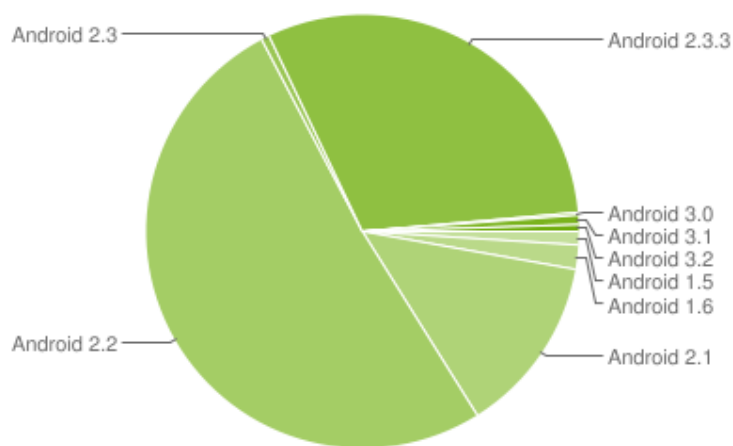


Figura 1- Gràfic circular de les versions d'Android que incorporen els terminals a Setembre de 2011.

Platform	Codename	API Level	Distribution
Android 1.5	Cupcake	3	1.0%
Android 1.6	Donut	4	1.8%
Android 2.1	Eclair	7	13.3%
Android 2.2	Froyo	8	51.2%
Android 2.3 - Android 2.3.2	Gingerbread	9	0.6%
Android 2.3.3 - Android 2.3.4		10	30.7%
Android 3.0	Honeycomb	11	0.2%
Android 3.1		12	0.7%
Android 3.2		13	0.5%

Taula 2 - Resum dels percentatges de terminals de cada versió d'Android a Setembre de 2011

Sumant els percentatges de les versions majors a 2.1, es pot observar que són el 97.2% dels terminals.

Per tant, la decisió inicial va ser encertada, ja que molts terminals s'han anat actualitzant amb el temps i els nous terminals que han anat sortint al mercat ja incorporen les últimes versions. A la següent *Figura 2* es pot veure aquesta tendència, on destaca l'estancament de les versions inferiors a *Android 2.1* i també el ràpid creixement de la versió 2.3 *Gingerbread*, que al inici del desenvolupament d'aquest projecte, encara no estava disponible.

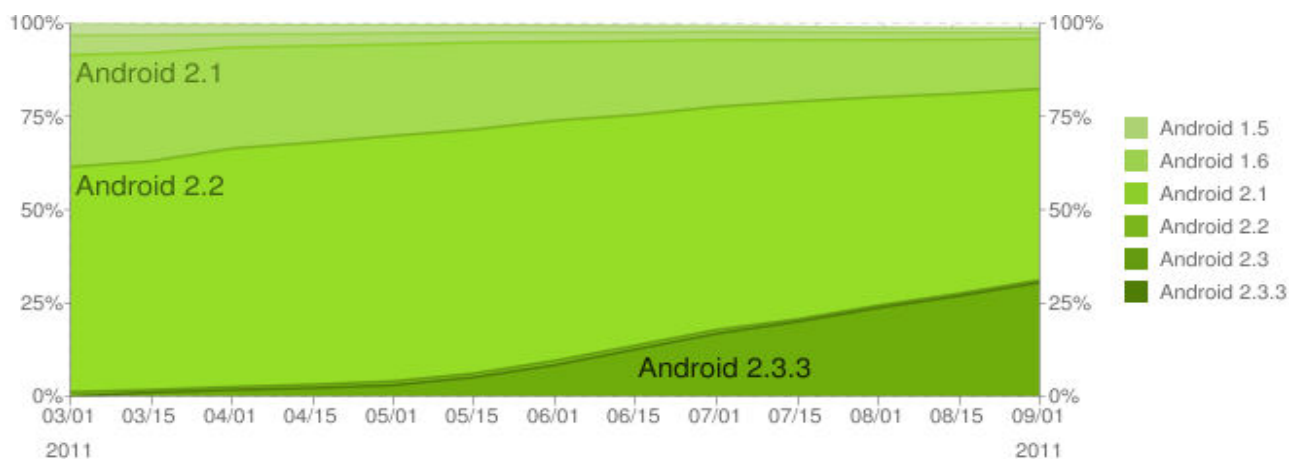


Figura 2 - Evolució de les versions dels terminals entre els mesos de Març i Setembre de 2011

2.2.2 SERVEIS DE CLOUD COMPUTING

Cloud Computing (computació en el núvol o núvol de còmput) és una tecnologia informàtica que presta un servei de computació en servidors remots o centres de dades emmagatzemades a Internet.

L'usuari del servei no ha de tenir coneixements en la matèria del càlcul ni ha de crear i mantenir la infraestructura que requereix: l'únic requisit per utilitzar-lo és tenir connexió a Internet.

En aquest projecte, hi ha tres càlculs que no és possible fer-los per diversos motius.

- Calcular les rutes caminant a partir de dos punts, amb indicacions pas a pas.
- Aconseguir el temps d'espera del bus en temps real (quan l'usuari arriba a la parada) de totes les línies i parades d'autobús de sis ciutats diferents.
- Saber l'adreça completa (país, ciutat, carrer i número) a partir de dues coordenades geogràfiques, i també el pas invers.

Aleshores s'ha decidit que es faran servir serveis de computació en el núvol en aquests tres casos. Per tant, és necessari utilitzar un servei de *Cloud Computing* que calculi la ruta caminant entre dos punts, indicant el sentit de cada gir i la orientació (Nord, Sud, etc) de cada part de la ruta. També és necessari un altre servei que faci les funcions de geocodificació (procés que permet obtenir l'adreça completa a partir d'un parell de coordenades geogràfiques) i geocodificació inversa (obtenir les coordenades exactes a partir del nom del carrer, número i ciutat). Per últim, és necessari consultar a les companyies d'autobusos de les diferents ciutats el temps d'espera en temps real, sempre i quan la companyia ho permeti.

2.2.3 SUBVERSION

Subversion és un sistema de control de versions amb llicència lliure [GNU/GPL](#) que automatitza la tasca de guardar, recuperar i barrejar versions de fitxers, utilitzant el protocol [SVN](#).

Permet que diferents usuaris utilitzin el mateix repositori on s'hi allotja el codi font del programa, amb la possibilitat d'actualitzar les modificacions de qualsevol usuari o restaurar l'aplicació a un estat anterior en cas de que sigui necessari. A més a més, permet crear línies de desenvolupament paral·leles amb noves funcionalitats i més endavant, quan es tingui la seguretat de que funcionen correctament, incorporar-les a la versió principal i estable de l'aplicació.

Hi ha d'altres sistemes de control de versions, com [Mercurial](#) o [Git](#), aquest últim molt utilitzat en l'àmbit del software lliure i en el projecte [AOSP \(Android Open Source Project\)](#), on col·laboren diferents fabricants de dispositius i processadors per millorar el sistema operatiu [Android](#). Però es va decidir utilitzar [Subversion](#) perquè ja se sabia instal·lar i configurar, i l'utilització d'un altre sistema hagués suposat invertir més temps en aprendre a instal·lar-lo i utilitzar-lo, temps que va ser preferible invertir en aprendre a programar en [Android](#).

2.2.4 ENTORN DE DESENVOLUPAMENT

En aquest apartat es descriu el software necessari per poder desenvolupar OnTheBus.

Eclipse IDE i Java Standard Development Kit (SDK): [Eclipse](#) és un entorn de desenvolupament integrat que ens permet programar en diferents llenguatges de programació, entre ells el llenguatge [Java](#), compilar el nostre codi i executar-lo. És necessària també la instal·lació del SDK de [Java](#), que és un kit de desenvolupament pel llenguatge [Java](#). Hi ha altres entorns per programar en Java perfectament vàlids per aquest projecte, com per exemple [NetBeans](#), però [Eclipse](#) és l'*IDE* recomanat per [Google](#) i per al que existeixen les últimes versions dels [plugins](#) que es descriuran seguidament, i és l'*IDE* que els integrants del grup estan acostumats a fer servir i dominen.

Android SDK: instal·lant aquesta eina podrem connectar el nostre terminal Android, ja que ens proporciona els drivers del dispositiu per al nostre sistema operatiu (està disponible per Windows, Linux i MacOS). Si no tenim un dispositiu [Android](#), podem simular el sistema operatiu mitjançant un emulador.

ADT Eclipse plugin: com el seu nom indica, és un afegit d'*Eclipse*. És molt complet, ens permet descarregar la versió d'*Android* que desitgem, crear emuladors amb les característiques que necessitem simular (acceleròmetre, *GPS* i altres tipus de sensors, resolució de pantalla, emmagatzemament extern, memòria *RAM*, etc), veure el *Log* d'informació i errors del terminal o emulador que estem executant, dissenyar la nostra interfície gràfica amb un editor gràfic, simular el nostre geoposicionament i fins i tot un recorregut introduint les coordenades geogràfiques de cada punt en un fitxer, entre d'altres coses interessants pel projecte. S'ha fet servir aquest perquè és l'únic que existeix que permet emular el sistema i, per tant, poder provar OnTheBus sense necessitat de tenir un dispositiu físic.

Subclipse: Es tracta d'un *plug-in* per *Eclipse* que permet connectar al repositori *SVN* de l'aplicació i efectuar les operacions modificació de codi i actualització del codi dels altres usuaris dins el mateix entorn de desenvolupament integrat *Eclipse*.

2.3 Planificació del Projecte

El projecte s'ha dividit en diferents etapes de dissenys i implementació definint fites a la finalització de cadascuna de les etapes i dependència directe entre la finalització i inici d'algunes d'elles.

1. Anàlisi de requeriments: En aquesta etapa es posa de manifest què es vol desenvolupar i es defineix a alt nivell que es farà.

2. Disseny i implementació objectes comuns: En aquesta etapa es dissenya i implementa els objectes de cada mòdul que seran visibles per la resta.

3. Disseny i implementació dels mòduls: Durant aquesta etapa es desenvolupen les funcionalitats que ha d'oferir cada un dels mòduls.

4. Proves unitàries: Durant aquesta etapa es realitzen proves a nivell modular tant en emulació com en terminal físic.

4. Disseny i implementació de comunicació entre mòduls: Durant aquesta etapa es conjunten tots els mòduls definint la informació a traspasar entre els mateixos i com es traspassa.

5. Proves de l'aplicació: Durant aquesta etapa es realitzen diferents proves de l'aplicació tant en emulació com en terminal físic.

6. Proves de camp (UAT): Durant aquesta prova es realitzen proves de camp fent ús de l'aplicació en escenaris reals. També es realitzen unes proves d'acceptació d'usuari que es duen a terme amb la col·laboració del director del projecte.

La durada de cada una de les etapes així com la seva planificació es pot veure en la següent taula (*Figura 3*):

		Nombre	Duració	Inici	Terminado
1		Gestió del projecte	1.062 days	22/10/10 16:00	28/07/11 17:00
2		Dinamització	1.062 days	22/10/10 16:00	28/07/11 17:00
9		Gestió general	164 days	10/12/10 8:00	27/07/11 17:30
10		Documentació general	842,667 days	1/11/10 17:00	9/06/11 17:30
16		Definició document de requeriments	277,333 days	25/10/10 8:00	4/01/11 17:30
24		Desenvolupament	442,667 days?	5/01/11 8:00	29/04/11 17:30
25		Mòdul algoritme de cerca	256,633 days	5/01/11 16:03	14/03/11 17:00
26		Disseny / Elecció de l'algorisme	85,967 days	5/01/11 16:03	27/01/11 17:30
30		Implementació	170,667 days	28/01/11 8:00	14/03/11 17:00
33		Test de l'algorisme	117,633 days	5/01/11 16:03	4/02/11 16:30
43		Mòdul core	111,833 days	15/03/11 8:00	12/04/11 16:45
44		Disseny	26,167 days	15/03/11 8:00	21/03/11 16:15
52		Implementació	53,667 days	21/03/11 16:15	4/04/11 16:45
56		Test del Controlador	32 days	4/04/11 16:45	12/04/11 16:45
60		Mòdul BDD	187,222 days	5/01/11 16:03	23/02/11 16:53
61		Buscar informació del servei d'autobusos	17,037 days	5/01/11 16:03	28/01/11 16:06
62		Aprovisionament dades	18,519 days	28/01/11 16:06	23/02/11 16:53
63		Estudi i Disseny E/R Base dades "Autobús"	2,222 days	5/01/11 16:03	7/01/11 16:23
64		Normalitzar E/R e implementar Base dades "Autobús".	4,444 days	7/01/11 16:23	13/01/11 17:03
65		Omplir Base de Dades "Autobús" amb les dades del ser...	2,667 days	14/01/11 8:00	18/01/11 17:00
66		Implementar (dintre de l'aplicació) la Base de Dades (SQ...	1,481 days	19/01/11 8:00	20/01/11 16:43
67		Generar els mètodes per les diferents connexions, acce...	8,148 days	5/01/11 16:03	17/01/11 16:16
68		Parsejar la informació dels temps que triguen els difere...	11,111 days	5/01/11 16:03	20/01/11 16:13
69		Mòdul Interfície d'usuari	442,667 days?	5/01/11 8:00	29/04/11 17:30
70		Esbossar un disseny	213,333 days?	5/01/11 8:00	1/03/11 17:00
75		Prototipatge	149,333 days	2/03/11 8:00	8/04/11 17:00
80		Mòdul de so	43,3 days	5/01/11 16:03	17/01/11 17:03
90		Mòdul de text	21,967 days	5/01/11 16:03	11/01/11 17:03
93		Mòdul Gestures	53,333 days	5/01/11 16:03	19/01/11 16:03
97		Mòdul de Vibració	21,333 days	5/01/11 16:03	11/01/11 16:03
100		Interfície final	80 days	11/04/11 8:00	29/04/11 17:30
105		Mapa	165,967 days	5/01/11 16:03	17/02/11 17:00
106		Estudi de la API de Google Maps	32,633 days	5/01/11 16:03	13/01/11 17:03
107		Control i visualització del Mapa	32 days	14/01/11 8:00	21/01/11 17:00
111		Posicionament en Pantalla	31,667 days	24/01/11 8:00	31/01/11 16:30
114		Rutes al Mapa	69,667 days	31/01/11 16:30	17/02/11 17:00
117		Mòdul Geoposicionament	272 days?	18/02/11 8:00	29/04/11 17:00
118		Geoposicionament GPS	208 days	18/02/11 8:00	13/04/11 17:30
119		Obtenció posició actual	37 days	18/02/11 8:00	28/02/11 16:30
122		Obtenció geoposició d'una adreça	37,667 days	28/02/11 16:30	9/03/11 17:00
127		Guiat rutes a peu - Backend	74,667 days	10/03/11 8:00	29/03/11 17:00
132		Guiat rutes en autobus - Backend	58,667 days	30/03/11 8:00	13/04/11 17:30
136		Obtenció rutes caminades	64 days?	13/04/11 17:00	29/04/11 17:00

Figura 3- Planificació temporal del projecte

Com es pot observar, la data prevista de finalització és el 28 de juliol de 2011, superant el primer termini d'entrega del projecte, tenint com a marge per a la resolució de problemes el mes d'agost.

S'ha de tenir en compte que s'ha definit que la dedicació de cada un dels recursos és de unes 4 hores dia.

2.3.1 DIAGRAMA DE GANTT

A continuació podem veure un diagrama de Gantt (*Figura 4*) amb el calendari del projecte així com les dependències entre les tasques globals.

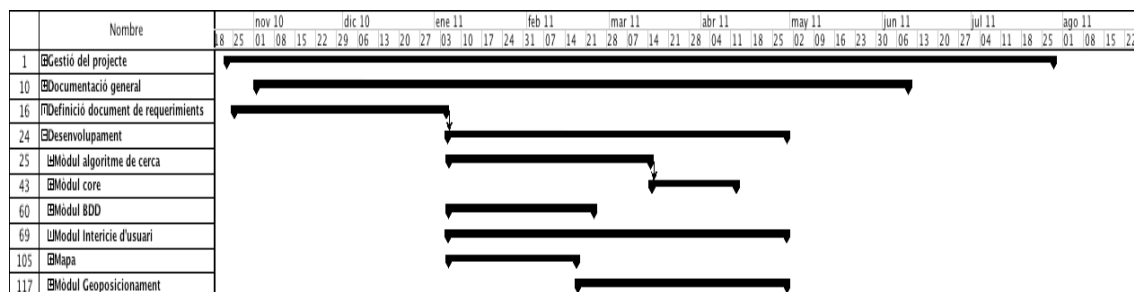


Figura 4 - Diagrama de Gantt

2.4 Disseny Universal

The opposite of useful, the opposite of something that is well designed is... useless” – by @JHockenberry

“No es pot pensar en la discapacitat de forma general, perquè cadascuna té les seves necessitats específiques. A cada problema, una solució” – Per @Carlos Martínez, vicepresident d'ASOCIDECAT, l'Associació de Sord-Cecs de Catalunya

La idea de disseny universal, va començar a prendre forma als voltants dels anys 90 com a evolució lògica del disseny sense barreres, accessible i de la tecnologia assistida de recolzament, amb la intenció d'aprofitar-se en un rang ampli de disciplines.

El Disseny Universal es refereix a un conjunt de principis acordats per un grup d'arquitectes, dissenyadors de productes, enginyers i investigadors del disseny ambiental que serveixen com a guia per l'obtenció de productes i entorns que siguin el més flexibles, equitatius, intuïtius possibles entre d'altres característiques, per poder ser utilitzats pel nombre màxim possible de persones, sense necessitats d'adaptacions o de disseny

especialitzat. No ens s'ha d'oblidar que hi ha diversos problemes d'accessibilitat per a un percentatge d'usuaris, els quals normalment es deixen de banda a l'hora de dissenyar *software*.

En essència són 7 principis que a més de servir per futurs desenvolupaments, també tenen funció avaluadora per dissenys actuals.

Al ser d'àmbit general, és possible que segons el tipus de disseny, algun d'aquests principis es pugui obviar o no ser del tot rellevant en el context de l'aplicació.

Els 7 principis, dels quals s'ha basat el disseny de l'aplicació, són:

1. **Ús equitatiu:** El disseny ha de ser fàcil d'usar i adequat per a totes les persones independentment de les seves capacitats i habilitats. Així doncs, cal evitar segregar a qualsevol usuari i proporcionar les mateixes formes d'ús per a tots. Idèntiques quan sigui possible i equivalents quan no hi hagi cap altre alternativa. Per tant, ha de ser utilitzable i a un preu raonable independent de les problemes d'accessibilitat dels usuaris.

2. **Ús Flexible:** El disseny ha de poder adequar-se a un ampli rang de preferències i habilitats individuals. Ha d'oferir opcions per tal que la interfícies es pugui utilitzar de diverses formes, tan si l'usuari es esquerrà com dretà i que sigui el més adaptable possible al ritme de l'usuari.

3. **Ús Simple o intuïtiu:** El disseny ha de ser fàcil d'entendre independentment de l'experiència, els coneixements, les habilitats o el nivell de concentració de l'usuari. Cal dissenyar una interfície sense complexitat innecessària i que sigui molt intuïtiva i guiable per a l'usuari, independentment del seu grau de coneixement amb les tecnologies i la seva edat. Es pot proporcionar d'informació complementària per tal de que sigui més guia, ja sigui ajudes, o avisos retroalimentats durant la tasca.

4. **Informació fàcil de percebre:** El disseny ha de ser capaç d'intercanviar informació amb l'usuari, independentment de les condicions ambientals o les capacitats sensorials del mateix. Per tal de que sigui fàcil de percebre, caldria utilitzar diferents mitjans, ja siguin sons, vibracions, visuals i lumínics, per a que la presentació sigui més completa. La informació ha de ser llegible i els elements s'han de diferenciar entre si per a que es puguin percebre per si sols.

5. **Tolerància a l'error:** El disseny ha de minimitzar les accions accidentals o fortuïtes que poden tenir conseqüències fatals o no desitjades. Per a que hi hagi aquesta tolerància, s'ha fer un ampli testeig del disseny, cal tenir en compte totes les màximes accions que pot arribat a fer un usuari per tenir en compte que no comportin errors. Així doncs, cal advertir dels errors

de forma clara, que s'entenguin els errors i per a que s'han produït, fet que l'usuari al entendre-ho no repetiria certes accions per tal de no conduir al mateix error.

6. **Mínim esforç físic:** El disseny s'ha de poder utilitzar de forma còmoda, eficaç i amb el mínim esforç possible. Cal que es pugui utilitzar minimitzant certs aspectes, com ara les accions repetitives, amb el mínim d'esforç físic constant, i en una posició neutral. En cas de que calgui un esforç, que la força utilitzada de la operació sigui raonable.

7. **Dimensions apropiades d'ús i aproximació:** Les mides i espais han de ser apropiats per l'abast, manipulació i ús per part de l'usuari independentment de la seva mida, posició i mobilitat. Proporcionar una interfície, on la mida dels elements siguin proporcionals a la mida del terminal. A més a més, que sigui lo suficientment grans com per a que siguin llegibles i de fàcil accés independentment de la mida dels dits dels usuaris o de la seva mobilitat.

L'objectiu principal d'aquesta filosofia és aconseguir una societat on totes les persones puguin participar, independentment de la seva condició, incloses les persones amb discapacitats (ja siguin limitació funcional visual, auditiva, cognitiva o de mobilitat), gent gran, de diferents grups de poblacions, etc...

Una manera de millorar el disseny universal és treballar directament amb un grup d'usuaris amb problemes d'accessibilitat, els quals et poden explicar quins problemes troben en els dissenys d'aplicacions i en l'avaluació de prototips realitzats.

D'altra banda, també cal sensibilitzar als dissenyadors. Per exemple el propi dissenyador pot simular certs problemes d'accessibilitat per veure en primera persona, quins problemes es pot trobar. Una manera és simular les limitacions de mobilitat, limitant els moviments amb objectes, embenatges, barres de ferro per immobilitzar articulacions, etc...

D'aquesta forma canviaria la manera de veure les conseqüències que poden tenir un mal disseny per als grups d'usuaris amb problemes d'accessibilitat.

En el propi context particular, s'ha trobat dificultats en el fet de que actualment, la gran majoria de telèfons mòbils amb *S.O. Android* instal·lat pateixen fragmentació, és a dir, hi ha molta diversitat de models, amb dispositius diferents, i cada companyia ha decidit els dissenys del seus aparells, amb o sense teclat, amb o sense botons i ni tan sols s'han posat d'acord en el nombre estàndard de botons a utilitzar.

Aquests detalls han portat a haver de cercar informació de tots o gairebé tots els dispositius mòbils amb *Android* per tal de poder concretar uns criteris perquè el disseny de la interfície fós el més universal possible.

A l'actualitat, s'ha pogut establir un nombre mínim de botons, que són:

- Els botons de Home, per minimitzar la *App*.
- El botó de Menú, per obtenir més opcions.
- El de *Back*, per tornar a la pantalla anterior.

Un altre factor important, ha sigut el fet de realitzar una aplicació de geo-localització. Aquest fet ha condicionat totalment en el moment de prendre les decisions més importants, en base a la realització de la interfície de guiatge en temps real i la selecció de destinació. El disposar d'un mapa i de la capacitat de representar informació sobre ell, juntament amb altres factors estètics, ha portat a dissenyar dues interfícies paral·leles i equivalents (principi [1]) per tal de poder treure el màxim rendiment a cada perfil.

La primera d'elles està destinada a persones que no necessiten cap tipus d'accessibilitat especial. Basant-se en el principi [2], es dona a l'usuari la opció d'activar/desactivar la forma en que rebrà la informació, ja sigui per veu o per text i la possibilitat de seleccionar el seu mode d'entrada de dades per defecte, encara que un cop a la pantalla corresponent es pot triar un altre.

Aquesta interfície utilitza *layouts* (fitxer que conté la distribució dels elements en una pantalla) amb botons virtuals, taules per sintetitzar la informació de les rutes, texts aclaridors i funcionalitats bàsiques amb els botons del dispositiu. D'aquesta manera s'aconsegueix accomplir el principi [3].

La segona interfície està més adaptada per tal de poder proporcionar la millor experiència a l'usuari que tingui una necessitat d'accessibilitat, solucionant el problema de la localització dels botons i de la informació, substituint-los per una barra lateral (per dretans o esquerrans, segons la configuració) que s'encarregarà de fer saber a l'usuari a on es troba en cada moment.

S'ha afegit ajudes vibratòries, visuals i sonores per informar a l'usuari sobre l'estat del guiatge i que no es trobi desorientat [1].

Per tal de poder intercanviar informació de manera eficient (principi [4]), els integrants del mòdul d'interfície gràfica han trobat formes apropiades per transmetre informació, com s'ha comentat abans i per poder rebre informació per part de l'usuari, s'han fet servir mètodes de reconeixement de veu, teclat virtual, *gestures*, historial, etc...

Els mètodes emprats en les dues interfícies, s'han hagut de testejar per poder minimitzar les seves fallades, fent l'aplicació més robusta, basant-nos en el principi [5]. També

s'ha hagut de tenir en compte les mides dels botons i dels texts [7] perquè siguin entenedors i no cal dir que pel fet de ser una aplicació mòbil, no cal fer cap esforç físic [6].

Per poder arribar al nombre màxim d'usuaris, s'ha implementat l'aplicació de manera que sigui multilinguatge, aprofitant les facilitats que ens dona el SDK *d'Android* i pel moment podem oferir l'aplicació en 3 idiomes: anglès, castellà i català. No obstant, està preparada per incorporar altres llenguatges en un futur.

La gran demanda de telèfons amb aquest S.O. ha portat a les companyies a dissenyar aparells per tal de poder arribar al nombre màxim d'usuaris i això ha portat a fer una classificació general de 3 tipus en quant als preus:

Gamma baixa: ZTE Blade, Huawei Ivy, Huawei Sonic, etc...

Gamma mitjana: Samsung Galaxy SCL, HTC Desire, etc...

Gamma alta: Samsung Galaxy S2, LG Optimus 2x, HTC Sensation, HTC Desire HD, etc...

Aquesta diferenciació de gammes, també ha suposat una diferenciació en els components dels aparells i això ha portat per exemple, a haver de realitzar una interfície preparada pel màxim nombre de resolucions i densitats de píxels (*Figura 5*). Per aquesta raó s'ha fet

	Low density (120), <i>ldpi</i>	Medium density (160), <i>mdpi</i>	High density (240), <i>hdpi</i>
Small screen	• QVGA (240x320), 2.6"-3.0" diagonal		
Normal screen	• WQVGA (240x400), 3.2"-3.5" diagonal • FWQVGA (240x432), 3.5"-3.8" diagonal	• HVGA (320x480), 3.0"-3.5" diagonal	• WVGA (480x800), 3.3"-4.0" diagonal • FWVGA (480x854), 3.5"-4.0" diagonal
Large screen		• WVGA (480x800), 4.8"-5.5" diagonal • FWVGA (480x854), 5.0"-5.8" diagonal	

densitat de píxels

servir com a unitats els dp (*density pixels*) per les mides dels layouts, i sp per les mides de les tipografies.

2.5 Anàlisi de Terminals

2.5.1 BOTONS FÍSICS EN DISPOSITIUS ANDROID

Els terminals mòbils han evolucionat molt durant els últims anys. Els primers terminals tenien botons físics, els quals es podien localitzar amb facilitat sense mirar, cosa que feia que fossin fàcilment accessibles pels usuaris amb problemes d'accessibilitat visuals. Avui en dia la majoria d'aquests botons han passat a ser tàctils i per tant ja no els podem diferenciar com abans.

Per tal de veure aquesta clara tendència a la desaparició dels botons físics per botons tàctils, s'ha realitzat un petit estudi dels diversos terminals que hi ha per al sistema operatiu *Android*.

El primer terminal va aparèixer els 2008, l'anomenat *HTC Dream*. Aquest terminal compta amb un teclat *qwerty* físic amagat sota la pantalla, a més a més, té una bola de navegació (*trackball*) amb la tecla enter per seleccionar els elements de la pantalla situada a la part central inferior del terminal. Conté 5 botons físics, dos a cada costat del *trackball* i un a sobre d'ell. Aquests botons, ordenats d'esquerra a dreta són el trucada, *home*, menú, *back* i penjar trucada/encendre i apagar el terminal. A part d'aquests botons visibles en la cara de la pantalla, en els laterals s'hi troba les tecles de volum i el disparador de la càmera. Així doncs, es pot veure que inicialment els terminals anaven carregats de botons físics.

Com s'ha comentat en l'apartat anterior, el tema de la fragmentació s'ha tingut ben en compte en el procés de disseny de la interfície. S'han estudiat nombrosos models de telèfons mòbils i cercat documentació al respecte i s'ha pogut observar que actualment, només hi han 3 botons que predominen a tots els telèfons amb S.O. *Android*. Ni tan sols el botó de la càmera, o cercar han estat respectats en alguns models, dificultant les interfícies i les implementacions associades.

2.5.2 DIFERENCIACIÓ ENTRE BOTONS FÍSICS I TÀCTILS

S'ha realitzat un anàlisi sobre dispositius de diferents companyies i s'ha seleccionat uns models que destaquen sobre la resta en lo referent al seu disseny a nivell de botons fent la classificació en botons físics, tàctils i digitals creats per *S.O.*

2.5.3 DISPOSICIÓ DELS BOTONS

En els dispositius analitzats tot i que porten botons semblants en els dispositius que porten botons semblants, s'ha pogut veure que la disposició d'ells és diferent, augmentant la dificultat en la implementació i en les decisions de disseny. Per exemple, fent servir la interfície 2, s'ha vist que en molts dispositius, la barra de desplaçament coincidia amb la posició del botó *Back*, fent inestable el seu funcionament en cas d'un desplaçament erroni fora de la pantalla. Per aquesta raó es va decidir bloquejar el botó per *software*. De la mateixa manera, es va mirar de bloquejar la resta de botons que habitualment es situen sota la pantalla i es va trobar informació en la que es parlava de la impossibilitat de bloquejar el botó de *Home* en les versions actuals del S.O. per no posar a l'usuari en una situació en la que no pogués avortar una aplicació, encara que en possibles versions, potser no serà així. Al ser impossible el poder bloquejar totalment els botons, es va realitzar una doble vibració als extrems de la barra per tal de mantenir sempre orientat a l'usuari.

2.5.4 ANÀLISI

A continuació es pot veure una petita, però significativa, mostra de diferents tipus de dispositius mòbils amb *Android*. S'ha triat dispositius de les principals marques per ser lo més representatius possibles. Es posa especial èmfasi en la disposició dels botons físics i tàctils. En l'*Annex A* si pot trobar una taula amb una extensa mostra de dispositius mòbils, els quals han servit per a realitzar l'anàlisi complet.

2.5.4.1 Motorola

MILESTONE



<u>Teclat</u>	<u>Analògic extern lliscant</u>
<u>Trackpad/Trackball</u>	<u>Trackpad analògic òptic</u>
<u>Botons tàctils</u>	<u>Back, Menú, Home, Cerca, Càmera</u>
<u>Botons físics</u>	<u>Volum</u>
<u>Extres</u>	<u>No</u>

Taula 3- Especificacions Motorola Milestone

2.5.4.2 HTC

MAGIC



Teclat	No
Trackpad/Trackball	Trackball
Botons tàctils	No
Botons físics	Back, Menú, Home, Cerca, Trucar, Penjar, Volum
Extres	No

Taula 4 - Especificacions HTM MAGIC

DESIRE Z



Teclat	Analògic extern lliscant
Trackpad/Trackball	Trackpad analògic òptic
Botons tàctils	Back, Menú, Home, Cerca
Botons físics	Volum, On/Off, Càmera
Extres	No

Taula 5 - Especificacions HTC DESIRE Z

CHACHACHA



Teclat	Analògic extern lliscant
Trackpad/Trackball	No
Botons tàctils	Menú, Home, Back, Cerca
Botons físics	Back, Facebook, volum
Extres	No

Taula 6 - Especificacions HTC CHACHACHA

2.5.4.3 SONY ERICSSON

XPERIA PLAY



Teclat	Analògic extern lliscant
Trackpad/Trackball	TouchPad analògic capacitiu
Botons tàctils	No
Botons físics	Back, Menú, Home, Cerca, Volum
Extres	Control Jocs

Taula 7 - Especificacions SONY XPERIA PLAY

2.5.4.4 SAMSUNG

GALAXY SII i9100



Teclat	No
Trackpad/Trackball	No
Botons tàctils	Back, Menú
Botons físics	On/Off, Volum, Home
Extres	No

Taula 8 - Especificacions SAMSUNG GALAXY SII

GALAXY SII amb teclat físic (variant SGH-I927)



Teclat	Analogic extern lliscant
Trackpad/Trackball	No
Botons tàctils	Back, Menú, Home, Cerca
Botons físics	On/Off, Volum
Extres	No

Taula 9 - Especificacions SMASUNG GALAXY SII (TF)

2.5.5 PROPERES TENDÈNCIES

S'ha cercat informació sobre futurs models de telèfons mòbils amb aquest [S.O.](#) i fins i tot es pot trobar algun futur model que no tindrà cap botó físic. Per part d' [Android](#), també s'ha vist la mateixa tendència. En properes versions del Sistema Operatiu, la totalitat dels botons físics seran opcionals, deixant la decisió en mans dels fabricants. Possiblement aquesta decisió s'ha degut a l'objectiu de fusionar les versions 2.3 per [Smartphones](#) i [Honeycomb](#), per [tablets](#).

S'hauria d'estudiar si aquesta tendència continuarà en augment o no, per tal de decidir si caldria afegir com a futures millores la remodelació de la interfície.

Un altre punt important, conseqüència de la desaparició dels botons físics, és la inutilització de l'actual revisor de pantalla d'[Android](#), el [Talkback](#), el qual el seu funcionament és donar una descripció dels elements de la pantalla per als quals es van seleccionant. Aquesta acció es pot a dur a terme, si el terminal mòbil conté un [Trackball](#) o [Trackpad](#). Com es pot veure, si la tendència dels terminals és que no continguin aquests elements, deixa inaccessible el terminal per part d'aquest grup d'usuaris i per tant, aquests hauran de buscar terminals on hi hagin el màxim nombre de botons físics.



2.6 Conclusions estudi de viabilitat

Un cop realitzat l'estudi de viabilitat del projecte es pot decidir la seva viabilitat. Si es decideix que no és viable s'haurà de buscar alternatives i realitzar un nou estudi des de l'inici d'altra banda si és viable es durà a terme.

En primer lloc el que s'ha realitzat és un estudi de les aplicacions de guiatge i d'ús dels transport públics que es poden trobar en el mercat i s'ha determinat que la aplicació cobreix certes mancances front a les aplicacions existents. Les grans aportacions de l'aplicació és l'accessibilitat que ofereix amb diferents tipus d'entrada de dades i un alt nivell de personalització de l'aplicació sense influir en la seva senzillesa.

D'altra banda s'ha realitzat un estudi dels costos que suposa el desenvolupament del projecte i es conclou que són assumibles donat que els costos materials són mínims i el cost més elevat és el cost per persona/hora en el desenvolupament de l'aplicació.

La planificació que s'ha realitzat dona un calendari raonable encara que el marge per a possibles endarreriments és força ajustat.

Finalment i tenint en compte tots els estudis realitzats es decideix que el projecte és viable i es procedeix a la seva realització.

3 Mòdul de Dades (BDD)

El mòdul de BDD (o de dades) principalment s'encarrega de recol·lectar i subministrar a l'aplicació, la informació necessària dels diferents serveis d'autobusos que hi ha a les ciutats, a més de gestionar altre informació interna.

3.1 Informació ciutats

En aquest apartat es definirà el grup de ciutats que s'integren en l' aplicació, detallant la informació definida per cada una d'elles i els mètodes per obtenir-la.

3.1.1 INFORMACIÓ GENERAL

Primerament es van definir els requisits mínims de l'aplicació fent un estudi sobre quina informació bàsica es necessitava per al seu funcionament, i després es va analitzar quina és viable aconseguir. Partint d'aquesta base, es va fer una extensa cerca de dades per poder definir un conjunt obert (ja que es pot anar ampliant en el futur) de ciutats que compleixen els requisits i que per tant, formaran part del projecte, a més de fixar la informació definitiva amb la que treballar.

3.1.1.1 Requisits mínims

Els requisits principals a l'hora d'incloure una ciutat a l'aplicació bàsicament eren dos, que es pogués aconseguir les coordenades de les parades d'autobús i que l'empresa que gestionés el servei d'autobusos tingués disponible (i accessible per obtenir dades) el servei web de "Temps d'espera a la parada".

La resta d'informació com recorreguts i característiques de les línies, també era i és important obtenir-la pel desenvolupament de l'aplicació, però és una informació molt fàcil d'aconseguir, per tant, tot i ser un requisit pel desenvolupament, no és una característica excloent a l'hora d'afegir ciutats.

3.1.1.2 Ciutats del projecte

Les ciutats que formen part de l'aplicació actualment són les següents:

- Barcelona
- Madrid
- València
- Saragossa
- Roma
- Hèlsinki

Després hi ha un llarg llistat de ciutats que, després d'estudiar la seva inclusió dintre de l'aplicació, han quedat descartades per diferents motius, tot i que en el futur poden ser incloses en cas de poder complir els requisits mínims del projecte.

Ciutats espanyoles

Sense servei de temps d'espera de l'autobús a la parada (o disponible per servei mòbil i no web).

- Girona, Tarragona, Lleida, Bilbao, Vitòria, Osca, Terol, Alacant, Castelló, Palma Mallorca, Menorca, Eivissa, Pamplona, Logronyo, Oviedo, Gijón, A Corunya, Santiago de Compostel·la, Pontevedra, Ourense, Vigo, Lugo, Àvila, Burgos, Lleó, Palència, Salamanca, Segòvia, Sòria, Zamora, Ciutat Real, Conca, Guadalajara, Múrcia, Cadis, Jaen, Càceres, Las Palmas.

Amb servei de temps d'espera de l'autobús a la parada i a falta de trobar les coordenades de les parades o poder parsejar la informació del servei de temps d'espera.

- Sevilla, Santander, San Sebastià, Valladolid, Albacete, Toledo, Almeria, Granada, Màlaga, Huelva, Badajoz, Santa Cruz Tenerife.

Ciutats de la resta del món

- París, Berlin, Londres, Manchester, Liverpool, Amsterdam, Milà, Tokio, New York, Washington.

Les raons per descartar-les de moment es que, o bé no disposen o s’han trobat les coordenades de les parades (cas *París*), o no disposen del servei d’espera a la parada de l’autobús. A *New York* per exemple estan començant a implementar el servei i només és accessible per una determina línia. En el cas de *Washington*, tot i tenir el servei de “Temps d’espera” i estar disponibles les coordenades de les parades a la plataforma *Google Transit* no s’ha implementat ja que els codis de parada de *Google Transit* no corresponien als codis de parada de la seva web, necessaris pel temps d’espera, llavors no podem gestionar de moment amb certesa aquest servei i s’ha optat per no incloure-la de moment.

3.1.1.3 Informació obtinguda

A banda dels requisits mínims, la informació que es creia necessària pel correcte funcionament de l’aplicació i que s’ha pogut aconseguir és la següent:

- Informació de les companyies d’autobusos:

Nom de la companyia, adreça postal, adreça electrònica, i número de telèfon.

- Informació de les línies:

Identificador, nom, horaris, diverses característiques de la línia (accessibilitat, tipus, línia circular o no) i el seu recorregut per les parades.

- Informació de les parades de les línies:

Identificador, nom i coordenades geogràfiques de la parada.

- Altres:

Dies festius, nom i telèfons d’algunes companyies de taxis.

3.1.2 OBTENCIÓ DE LA INFORMACIÓ

Tota la informació s’ha obtingut a través d’Internet i les diferents pàgines webs de les companyies d’autobusos o de les entitats que els gestionen. Per poder accedir a la informació i començar a treballar i gestionar-la, en alguns casos s’han pogut descarregar diferents arxius de text i en d’altres, s’han desenvolupat diversos parsers que accedien al contingut web i permetien obtenir tota la informació necessària, guardant-la en els formats de fitxer més

adients per poder gestionar-la en el futur. Com cada ciutat/companyia té la seva pròpia pàgina web, s'ha generat un parser diferent per a cadascuna, ja que el codi reaprofitable era mínim (només la part de connexió a les pàgines web).

En el cas de *Roma*, al no poder obtenir les coordenades de les parades en les diferents recerques d'informació pública fetes, es va contactar amb el responsable de la planificació de transport del seu servei de mobilitat (*Roma Servizi per la Mobilità*) per veure si podia donar suport a l'aplicació, ja que per la seva història i la dificultat d'implementar el servei de metro, *Roma* té una de les xarxes d'autobús més grans del món. A més, el seu servei de temps d'espera a la parada és dels més complets, tal i com es veurà més endavant, en la secció 3.4.1). El resultat va ser molt satisfactori ja que va proporcionar la informació demanada i va mostrar força interès en el projecte.

A mida que es va fer la recol·lecció de la informació, es va procedir també a la normalització d'aquesta abans d'inserir-la en els nostres fitxers. Això permet tenir les diferents dades d'un mateix tipus o model en un format prèviament establert i així evitar incongruències un cop aquestes hagin sigut inserides en les bases de dades.

Per tal de poder realitzar la normalització, s'han implementat convertidors de dades en els casos que ha sigut necessari, com per exemple, en les coordenades cartogràfiques de les parades d'autobús d'una ciutat. Es va trobar que aquestes coordenades poden estar disponibles en projecció *UTM* (Sistema de Coordenades Universal Transversal de Mercator), *Gauss-Boaga*, o *Datum WGS84* (World Geodetic System 84). Aquest últim, és el sistema amb el que s'ha treballat durant tot el projecte, ja que és el sistema que utilitza el *Sistema de Posicionament Global* (GPS) i les diferents aplicacions de *Google*. Això ha implicat convertir les coordenades recol·lectades en format *UTM* a *WGS84* o *Gauss-Boaga* a *WGS84*.

Per trobar més informació sobre els tipus de projeccions utilitzades en el projecte, veure [Annex C](#).

3.1.2.1 Convertidors de coordenades

Només utilitzats en els casos en que les coordenades de les parades d'una determinada ciutat no siguin coordenades geogràfiques *WGS84*.

- Transformació coordenades *UTM* a geogràfiques *WGS84*:

Aquest convertidor s'ha utilitzat en les coordenades de les ciutats de Barcelona i Madrid. S'ha generat a partir de la unió i adaptació de dos convertidors diferents que vam trobar a Internet:

- *UTM(ED50)* a *UTM(WGS84)*.
- *UTM(WGS84)* a coordenades geogràfiques (*WGS84*).

Abans d'utilitzar aquets convertidors, es va utilitzar un que feia la transformació directament de coordenades *UTM* amb *Datum ED50* a coordenades geogràfiques (*WGS84*), però amb aquesta conversió es va trobar que hi havia un error de 150-200 m en la localització de la parada. Per aquest motiu és necessari realitzar prèviament la transformació de coordenades *UTM* de *Datum ED50* a *UTM* amb *Datum WGS84* i després a coordenades geogràfiques (*WGS84*).

Així doncs, la transformació total de les dades és la següent:

UTM (ED50) → UTM (WGS84) → Geogràfiques (WGS84)

En els dos convertidors ha estat necessària fer una mateixa adaptació, ja que aquets només estaven implementats per fer la conversió d'una dada en format *UTM* a *WGS84* i es necessitava poder fer la conversió de totes les dades disponibles en un sol cop.

- Transformació coordenades en format *Gauss-Boaga* a geogràfiques *WGS84*:

Convertidor utilitzat en les coordenades cartogràfiques de *Roma*. Com s'ha comentat abans, des del servei per la mobilitat de Roma van enviar les bases de dades en un arxiu *Shapefile*¹, dintre del qual les coordenades estaven en format *Gauss-Boaga*.

Per poder fer la conversió de coordenades es va buscar a veure si hi havia algun programa que ens pogués fer la conversió directament de *Gauss-Boaga* a *WGS84* a partir del fitxer *Shapefile* proporcionat, però tot el software trobat, només deixava convertir una

¹ format d'arxiu que s'utilitza per l'emmagatzemament digital de la localització dels elements geogràfics i atributs associats a ells.

coordenada cada cop i no hi havia cap opció per poder modificar els programes per poder fer la conversió de totes les coordenades directament.

Després de fer una exploració en el diferent software trobat que treballava amb el format *Shapefile*, es va veure que combinant les característiques de dos programes diferents, es podia fer la conversió de dades satisfactòriament:

Primer de tot es va utilitzar un programa que permetia fer l'exportació de totes les dades carregades del *Shapefile* (coordenades de les parades e id d'aquestes) a un fitxer de bases de dades en format *".adp"*. Aquest fitxer després és carregat per un altre programa en el qual es van poder fer les conversions de dades de *Gauss-Boaga* a *WGS84*. El principal problema del software que permetia fer diverses conversions de coordenades a l'hora, era que no admetia fitxers de tipus *Shapefile*, en canvi si que permetia fer la càrrega de dades en altres fitxers de base de dades, per això es va haver d'utilitzar dos programes diferents.

3.2 Base de dades

L'aplicació treballa amb un conjunt de bases de dades per tal de poder consultar/accedir a diferent informació necessària per dur a terme el projecte, per tant, es necessita definir un sistema que gestioni les bases de dades. Android incorpora "SQLite" com a motor de base de dades i dóna suport complet per base de dades SQLite, per tant, és el sistema de gestió utilitzat durant tot el desenvolupament del projecte.

3.2.1 BASE DE DADES SQLITE

SQLite és una llibreria de software de domini públic, programada en llenguatge C i s'implementa sense necessitat de configuració un motor transaccional de base de dades SQL, ja que la pròpia SQLite és un motor de base de dades SQL encastat.

Les aplicacions que utilitzen la biblioteca de *SQLite* poden utilitzar bases de dades sense necessitat d'instal·lar un motor de base de dades extern. A diferència de la majoria d'altres base de dades *SQL*, *SQLite* no té un procés de servidor independent, ja que llegeix i escriu directament en els arxius de disc normal. En un arxiu únic de disc és continguda una base de dades *SQL* completa, amb taules, índex, triggers i vistes.

SQLite és una llibreria compacta. Amb totes les característiques habilitades, el tamany de la biblioteca pot ser inferior a 300KiB, depenent de la configuració de l'optimització del compilador. Si les característiques opcionals s'ometen, el tamany en la llibreria *SQLite* pot reduir-se per sota de 180KiB. *SQLite* també es pot executar en un espai mínim de pila (4KiB) i en un molt petit heap (100KiB), el que fa que sigui una gran opció com a motor de base de dades sobre gadgets amb memòria limitada, como telèfons mòbils, PDAs i reproductors MP3.

A la plataforma *Android* Totes les bases de dades es creen a la carpeta `/data/data/<package-name>/databases` de la memòria interna del telèfon mòbil, i es poden importar o exportar les bases de dades de les aplicacions.

Android retorna el resultat de les consultes a la base de dades en un objecte *Cursor*, que no és més que un punter a un conjunt de dades. Per defecte les bases de dades son privades a l'aplicació, per compartir dades entre aplicacions s'utilitzen els *Contents Providers*.

Per la naturalesa de l'aplicació, no s'implementa cap *Content Provider* per gestionar els mètodes d'accés a les dades ja que en principi no hi havia previsió de comunicar l'aplicació

amb unes altres. En el cas que un futur l'aplicació formi part d'un possible grup d'aplicacions que comparteixin informació, caldria adaptar el gestor de mètodes a un *Content Provider*.

3.2.2 BASES DE DADES DEL PROJECTE

En l'aplicació es fa servir un gran volum de dades que són emmagatzemades en diferents bases de dades com s'ha introduït anteriorment. Al tenir diferents tipus de funcionalitats, informació, accés, etc, s'han implementat tres models diferents de base de dades, on un model pot tenir diverses bases de dades del mateix tipus però canviant la informació.

Segons el model, les bases de dades són internes (implementades i creades internament per l'aplicació) o externes (creades e implementades des de fora de l'aplicació) les quals s'han d'integrar a l'aplicació abans de fer l'entrega d'aquesta al usuari.

3.2.2.1 BD Ciutats

En aquest model de base de dades s'emmagatzema tota la informació relativa a les ciutats incorporades al projecte. Pel millor funcionament de l'aplicació es generarà una base de dades o mòdul diferent per cada ciutat, això permetrà fer-ne de nous en el futur i facilitarà intercanviar-los més fàcilment quan es necessiti.

Cada base de dades (mòdul) serà generada externament a l'aplicació, ja que crear la base de dades corresponent e insertar tota la informació que es té disponible sobre les ciutats en qualsevol moment de l'execució de l'aplicació, no és gens òptim pel gran volum de dades que cada base de dades d'aquest model manega. A més, això permetrà un major control de consistència a l'hora de futures actualitzacions.

3.2.2.1.1 Disseny conceptual i relacional

L'objectiu es obtenir una bona representació de dades, amb independència dels usuaris o les diferents aplicacions.

A partir de les especificacions i la informació recol·lectada, es dissenya un esquema Entitat-Relació, que com el seu nom indica, consta d'una col·lecció d'entitats (qualsevol

objecte del món real, distingible, a representar en la base de dades) i relacions (vinculen diferents entitats, associant un valor entre aquestes) descrits en un diagrama Entitat-Relació.

Les entitats contenen diverses característiques que les identifiquen o les defineixen de forma única. Aquestes característiques són els atributs, els quals també poden formar part d'una relació.

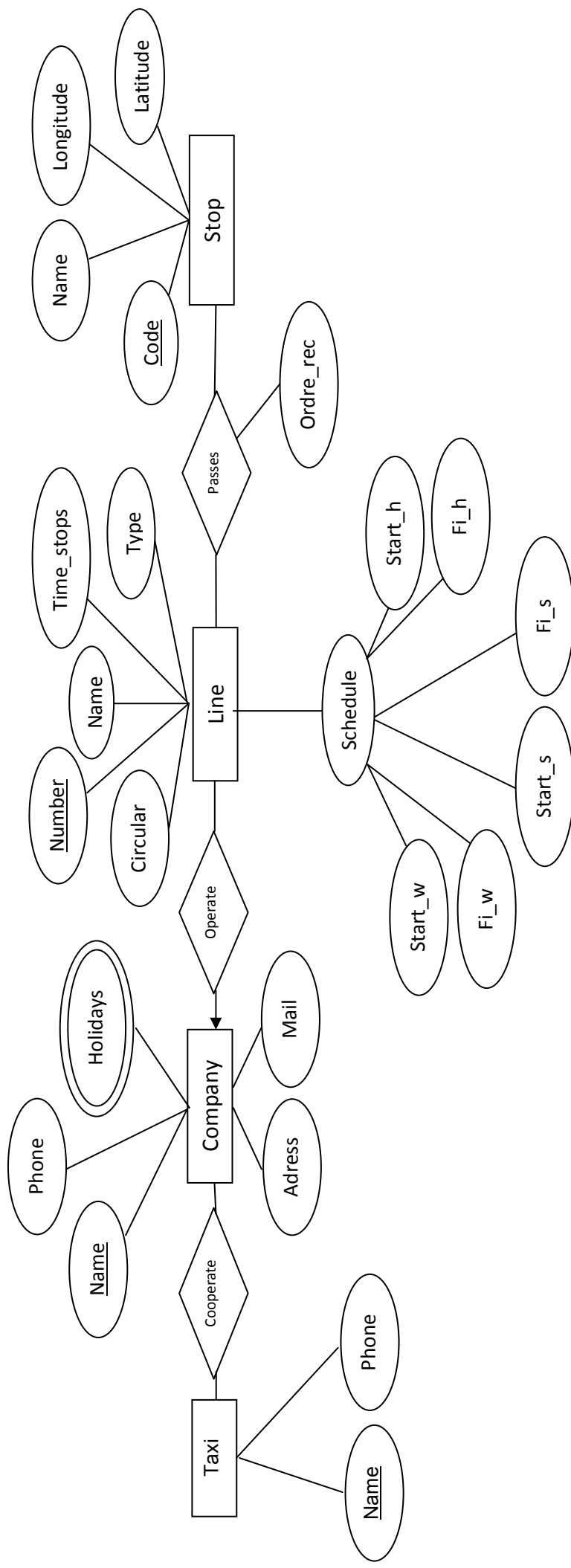


Figura 6. Diagrama Entitat-Relació base de dades del model ciutat

Com es pot veure en el diagrama de la *figura 6*, hi han quatre entitats: “Companyia”, “Línia”, “Parada” i “Taxi”. A continuació es descriurà cada entitat amb els seus atributs:

Company (Companyia)

- Name: Clau primària de l’entitat. Nom de la companyia d’autobús.
- Phone: Telèfon de la companyia.
- Adress: Adreça de la companyia.
- Mail: Direcció electrònica de la companyia d’autobús.
- Holidays: Atribut multivalorat. Dies que son considerats festius a la ciutat que pertany la BD.

Line (Línia)

- Number: Clau primària de l’entitat. Número o identificador de la línia.
- Name: Nom de la línia.
- Time_stops: Temps aproximat que hi ha entre una parada i la següent en el recorregut.
- Type: Es refereix a si la línia es operativa pel dia o per la nit i a més si és adaptada per discapacitats o no.
- Circular: Defineix si la línia és circular (té un únic sentit) o pel contrari té dos, anada i tornada, i fa una parada (i cal tornar a introduir el bitllet) al final de cada sentit.
- Schedule: Atribut compost;
 - Start_w: Hora en que la línia comença a ser operativa en dies laborables.
 - Fi_w: Hora en que surt l’últim autobús de la parada capçalera en dies laborables.
 - Start_s: Hora en que la línia comença a ser operativa els dissabtes.
 - Fi_s: Hora en que surt l’últim autobús de la parada capçalera els dissabtes.

- Start_h: Hora en que la línia comença a ser operativa en diumenges i dies festius.
- Fi_h: Hora en que surt l'últim autobús de la parada capçalera en diumenges i dies festius.

Stop (Parada)

- Code: Clau primària de l'entitat. Número o identificador de la parada.
- Name: Nom de la parada.
- Longitude: Coordenada "longitud" (WGS84) de la parada en sexadecimal.
- Latitude: Coordenada "latitud" (WGS84) de la parada en sexadecimal.

Taxi (Parada)

- Name: Clau primària de l'entitat. Nom de la companyia de taxis.
- Phone: Número de telèfon de la companyia de taxis.

A més de les entitats, el diagrama E-R conté tres interrelacions: "Hi Operen", "Passa per" i "Cooperen":

Operate (Hi operen):

Interrelació binària 1-n entre les entitats "Companyia" i "Línia". Expressa que una companyia pot tenir n línies i una línia només pertany a una companyia.

Passes (Passa per):

Interrelació binària n-n entre les entitats "Línia" i "Parada". Expressa que una línia pot tenir n parades i una parada pot està dintre del recorregut de n línies. Conté un atribut, `Ordre_rec` (Posició en que una parada es recorreguda per la línia).

Cooperate (Cooperen):

Interrelació n-n entre les entitats “Companyia” i “Taxi”. Expressa que n companyies de taxi pot cooperar o ser accessibles a n companyies d'autobús, i a l'inrevés.

3.2.2.1.2 Disseny lògic

El disseny lògic d'una base de dades consisteix en la conversió de l'esquema E-R generat en l'etapa anterior, a un esquema relacional, basat en taules (relacions i atributs), a partir del qual es referirà el gestor de bases de dades triat.

Per fer la conversió del esquema entitat-relació a relacions, s'han seguit aquest passos:

- Conversió d'entitats fortes.
- Conversió d'interrelacions.
 - Combinació d'interrelacions.
- Conversió d'atributs compostos.
- Normalització de les relacions.
 - Primera Forma Normal (1NF).
 - Segona Forma Normal (2NF).
 - Tercera Forma Normal (3NF).
 - Forma Norma Boyce-Codd (BNF).
 - Quarta Forma Normal (4NF).

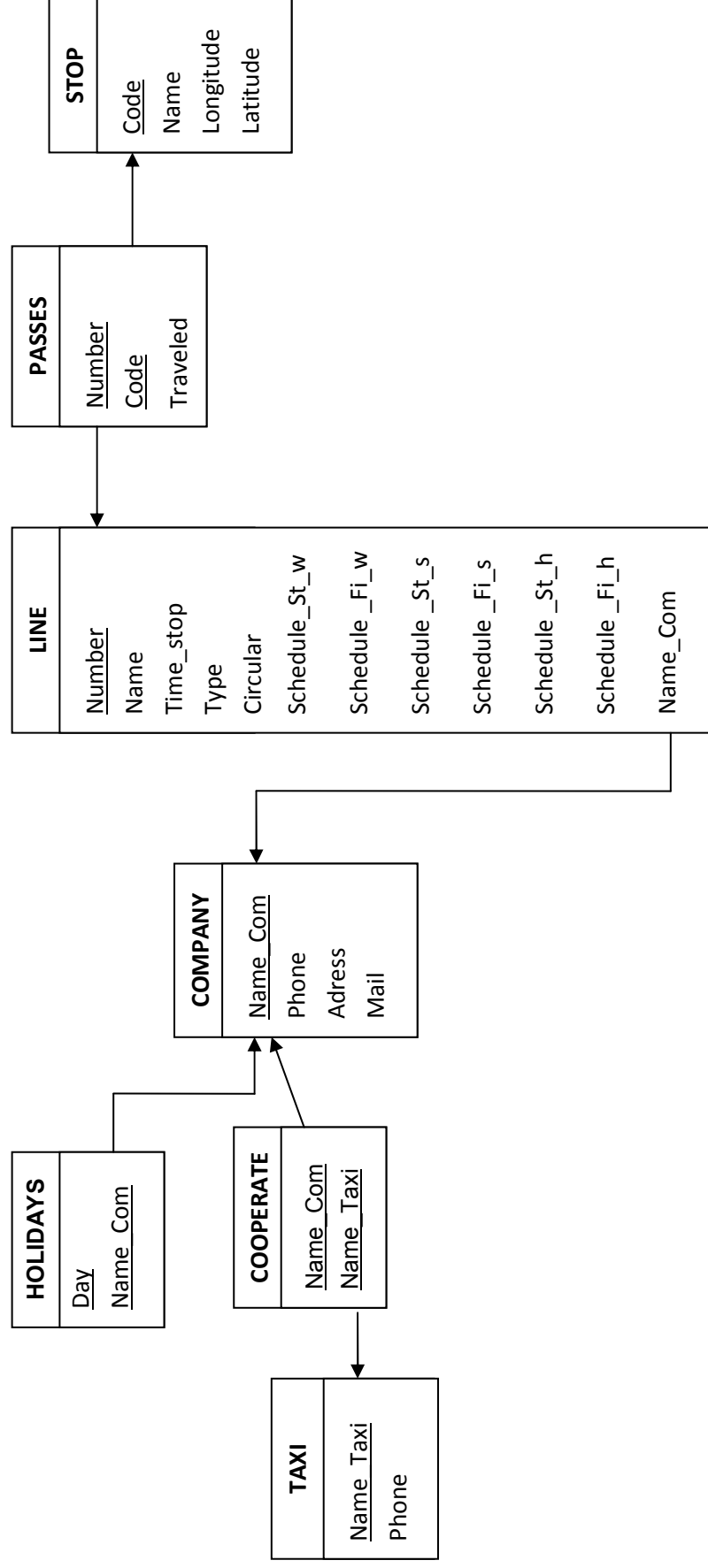


Figura 7. Diagrama Conceptual base de dades del model ciutat

3.2.2.1.3 Disseny Físic: Implementació en SQLite

Per gestionar les bases de dades de les ciutats s'ha utilitzat durant *SQLite Database Browser*, que és una eina de software lliure implementada per crear, dissenyar i editar arxius de base de dades compatibles amb *SQLite*. Com a funcionalitats més generals, conté una petita consola per generar executar codi *SQL* i dos visors, un per veure l'estructura de la base de dades carregada i l'altre per veure les seves taules i tot el seu contingut.

Mitjançant aquest programa es genera el fitxer de bases de dades de cada ciutat i es creen les taules que s'han vist en el disseny lògic de la base de dades (secció anterior). Tot i que l'eina proporciona una interfície força intuïtiva per crear les taules, aquestes (i els seus atributs, claus primàries, etc) han sigut creades a partir de l'execució en la consola d'un codi *SQL* generat per nosaltres.

Per omplir les bases de dades s'ha implementat un únic programa en llenguatge *Java* que serveix per cada ciutat/mòdul. Per cada taula de la base de dades, es carrega el fitxer on es tenen guardades totes les dades normalitzades preparades per inserir i es processa per tal d'introduir la informació a les taules mitjançant consultes *SQL*.

Cal fer un petit incís en el fet de tenir les dades i els fitxers normalitzats en uns formats establerts, ja que això facilita molt les coses a l'hora de fer portables tots els mòduls i sobretot de cara a possibles actualitzacions o generacions de nous mòduls en el futur.

També és important generar uns mecanismes de consistència de les bases de dades, ja que a l'hora de recollir tota la informació que anirà dintre d'aquestes, per una mateixa ciutat es pot haver agafat la informació de diferents fonts o fins i tot podem trobar que en la mateixa font (pàgina web), la informació estigui en dos seccions diferents i una estigui més actualitzada que l'altre i això indueix a errors e inconsistències. Sobretot repercuteix en els recorreguts de les línies, on poden haver-hi parades "fantasma", és a dir, que ja no son operatives però la font que publica la informació no ha actualitzat la part de recorregut de les línies (on només ens mostra el identificador de parada i l'ordre en que serà recorreguda pel autobús) i en canvi si la secció de la informació de les parades (on obtenim el nom i les coordenades de les parades, a més del seu identificador).

Per dur a terme aquest tractament, segurament seria millor fer-lo abans d'inserir la informació a les bases de dades, és a dir, en els fitxers on tenim tot, però es realitzat un cop es tenen les bases de dades omplertes perquè així és molt més fàcil tractar i modificar els possibles errors trobats amb l'ajuda de diferents consultes *SQL*.

3.2.2.1.4 Accés BD Ciutats: Consultes SQL

Per tal de proporcionar a l'aplicació totes les dades, s'han generat diverses consultes SQL que accedeixen a la informació emmagatzemada a les bases de dades. A l'hora d'implementar les consultes, es va establir una comunicació amb els mòduls que havien d'accedir a les bases de dades per tal d'adaptar la informació retornada als paràmetres que es necessitaven.

A continuació s'exposaran els mètodes més importants:

API Android

- Package: *android.database.sqlite*

Classes principals del Package:

- **SQLiteDatabase**: Classe que proporciona mètodes per gestionar una base de dades SQLite.
- **SQLiteCursor**: Implementa un Cursor (Interfície que permet l'accés al resultat d'una consulta a una base de dades) per poder exposar resultats derivats d'una consulta a una base de dades SQLite.
- **SQLiteOpenHelper**: Classe que permet crear i gestionar les bases de dades i les seves versions. Té un constructor (el qual no es sobreescriu en aquest cas) i dos mètodes abstractes principals que normalment cal implementar, *onCreate()* i *onUpgrade()*. Per poder utilitzar la classe i els seus mètodes cal estendre-la primer i després obtenir un *handler* sobre aquesta.

```
private static class DatabaseHelper extends SQLiteOpenHelper {  
    DatabaseHelper(Context context) {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    }  
}
```

```
myDbHelper = new DataBaseHelper(context); //Handler
```

- `OnCreate()`: Mètode per crear la base de dades per primera vegada. Es sobreescriu
- `OnUpgrade()` Aquest mètode és llençat automàticament pel sistema quant

en el futur es vulgui actualitzar les bases de dades de l'aplicació. Es sobreescriu per tal d'adaptar-lo a les exigències del projecte.

Rep per paràmetres la versió actual de les bases de dades que hi ha a l'aplicació i la nova versió a la que es vol actualitzar. Cal remarcar que quan s'actualitza, la versió serà equivalent a totes les bases de dades, és a dir, tot i que només s'actualitzi un dels mòduls, la versió s'aplicarà a la resta. El fet de tenir varies bases de dades complica fer una actualització automàtica, i el fet d'assignar a totes una mateixa versió, facilita les coses per desenvolupar.

Si la versió actual de bases de dades es anterior a la sol·licitada, es cridarà automàticament al mètode `onUpgrade()`, on es cercaran les bases de dades carregades a l'aplicació (tant a la memòria SD com a la memòria interna) i s'esborraran de la memòria. Això permetrà que la propera vegada que l'usuari carregui un mòdul de ciutat, ja es carregui la base de dades actualitzada a partir de l'actualització de l'aplicació.

API OnTheBusSQLite

- **`getLines()`**: És la classe més important del API. Retorna al Mòdul de Cerca totes les línies d'autobús disponibles (per tema horari i preferències d'usuari) per tal que pugui generar el graf. Cada línia de bus conté una llista amb la informació de totes les parades que formen part del seu recorregut.
- **`getNearStops()`**: Classe que retorna les parades d'autobús properes a un punt. A més de les coordenades del punt, rep com a argument el radi màxim a partir del qual es volen obtenir la llista de parades. Per fer els càlculs de distàncies s'utilitza la distància *Manhattan*.

3.2.2.2 BD Favorits

Els usuaris disposen en l'aplicació d'una secció de favorits on podran guardar els seus destins. Per administrar aquesta informació, s'han estudiat les diferents vies d'emmagatzematge de la informació proporcionada pel usuari i es va decidir que l'opció més recomanable és utilitzar una base de dades creada internament en l'aplicació, després de veure incompatibilitats amb una possible sincronització amb les destinacions preferides que l'usuari té guardades al seu compte de *Google*.

3.2.2.2.1 Estudi sincronització Google Maps

Google Maps disposa de diverses funcionalitats per gestionar els mapes o destins favorits dels seus usuaris. Aquestes funcionalitats permeten guardar, modificar i consultar tots els mapes personalitzats pel usuari.

Dintre de les funcionalitats es troba la capa "*Els Meus Mapes*" que permet afegir, modificar, eliminar i classificar en grups tots els favorits del usuari. A més, inclou la funció "*Elements Destacats*" que permet la gestió de favorits de manera més ràpida i simple, però sense poder classificar-los en grups. El major problema de la capa "*Els Meus Mapes*" és que la gestió completa dels favorits únicament és accessible a través d'un ordinador i en les versions *Android* més actuals únicament es pot consultar les dades, però no afegir.

Per la gestió de la funcionalitat "*Els Meus Mapes*" en *Android*, *Google* va desenvolupar l'aplicació *My Maps Editor*, que permetia als usuaris crear i editar mapes personalitzats amb el telèfon mòbil, però aquesta aplicació només és compatible amb versions inferiors a la versió 1.6 d'*Android* (aquesta inclusivament), pel que no es pot fer ús en l'aplicació *OnTheBus*, ja que està basada en la versió 2.1.

"*Elements Destacats*" permet guardar la marca de posició determinada pel usuari dintre del mapa i afegir informació. El major problema es que aquesta informació/marques no es poden classificar, si no que es guarden en una llista ordenada per antiguitat d'inserció. A favor, té que des del dispositiu *Android* és fàcilment accessible ja que ve implementat en les funcionalitats principals, pel que podem guardar en la nostre compte de *Google* i consultar els nostres favorits sense cap problema a través del nostre dispositiu *Android*.

Un cop estudiats els pros i els contres de poder sincronitzar l'aplicació amb *Google Maps* per gestionar la funcionalitat dels favorits, s'ha decidit descartar de moment aquesta sincronització i optar per una base de dades interna a l'aplicació. No obstant això, en futures actualitzacions de l'aplicació no es descarta utilitzar la sincronització de dades amb *Google Maps* si s'actualitza l'aplicació *My Maps Editor* a versions més actuals d'*Android* o si es publica en l'*API* les dades necessàries per realitzar la implementació.

3.2.2.2 Disseny conceptual i relacional

Per implementar internament la base de dades "Favorits" s'han estudiat dues possibilitats, generar una base de dades "Favorits" única i englobant tots els favorits de les ciutats o una base de dades exclusiva per cada ciutat.

Després de fer diferents proves sobre l'espai que ocupen les diferents bases de dades (veure resultats, punt 4.1.3), es va arribar a la conclusió de que es millor utilitzar una base de dades única. Per la naturalesa d'aquesta funcionalitat, es pot trobar que la ciutat de residència del usuari pot tenir força registres, en canvi, en altres ciutats el registres poden ser mínims ja que només pot estar allà una setmana o uns dies, pel que es tindria en memòria diverses bases de dades pràcticament obsoletes i amb un accés mínim. En canvi, al utilitzar una mateixa base de dades per guardar els favorits, es va trobar que el tamany d'aquesta en la seva creació no serà molt diferent de la mateixa base de dades després d'afegir els diferents registres. Això és degut a que SQLite ja fa una previsió d'espai de memòria pels registres i aquest no augmenta si no s'introdueixen un gran volum de dades. La base de dades "Favorits" es crearà quan l'usuari utilitzi per primer cop la funcionalitat.

3.2.2.3 Model Entitat – Relació

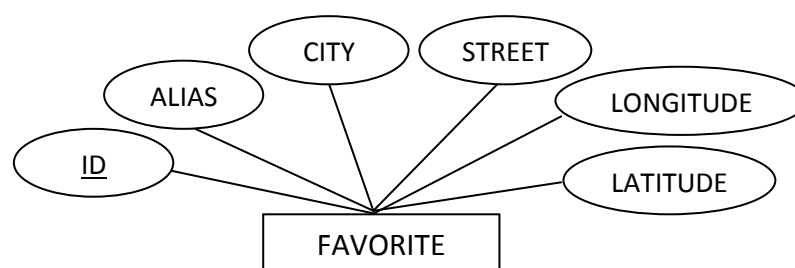


Figura 8. Diagrama Entitat-Relació base de dades model Favorit

1 entitat: FAVORITE.

Favorite:

Id: Clau Primària de l'entitat. Identificador/número de favorit registrar a la BD. Tipus enter.

Alias: Nom que l'usuari dona al seu contacte favorit.

City: Ciutat a la que pertany l'usuari.

Street: Adreça postal del favorit.

Longitude: Coordenada "longitud" (WGS84) en sexadecimal a la que pertany l'adreça del favorit .

Latitude: Coordenada "latitud" (WGS84) en sexadecimal a la que pertany l'adreça del favorit.

3.2.2.2.4 Disseny Lògic

FAVORITE
<u>Id</u>
Alias
City
Street
Longitude
Latitude

Figura 9. Diagrama Conceptual base de dades model Favorit

3.2.2.3 BD Historials

Per tal de facilitar als usuaris la cerca de possibles destinacions, s'ha implementat una funció d'autocompletar dades, que mostra als usuaris les destinacions que ha cercat últimament. Per tal de guardar aquesta informació s'ha implementat una petita base de dades de dades interna que manté tots els destins visitats (sense repeticions) sempre que l'usuari no esborri les dades en una de les funcionalitats del projecte.

3.2.2.3.1 Disseny conceptual i relacional

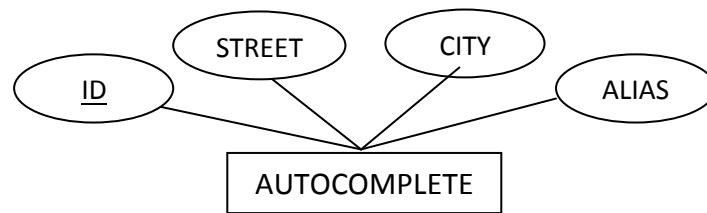


Figura 10. Diagrama Entitat-Relació base de dades model Autocompletar

1 entitat: AUTOCOMPLETE.

Autocomplete:

Id: Clau Primària de l'entitat. Identificador intern de destí registrat a la BD.

Street: Destinació insertada pel usuari.

City: Ciutat o població a la que pertany la destinació del usuari.

Bd: Base de dades a la qual pertany el destí.

3.2.2.3.2 Disseny Lògic

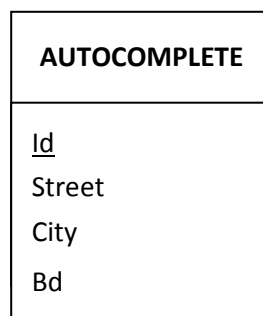


Figura 11. Diagrama Conceptual base de dades model favorit

3.3 Gestió de memòria del telèfon mòbil

Al utilitzar bases de dades generades externament, es necessita importar-les a l'aplicació mitjançant algun mètode. Donat que no es disposa o disposarà de cap servidor web per poder descarregar les dades quan s'executa l'aplicació, s'han d'adjuntar totes les dades en el .apk per poder-ne fer ús en temps real.

Tots els arxius auxiliars al projecte es guarden a la carpeta "Assets" dels projectes *Android*. Aquesta carpeta no es pot modificar en temps real, és a dir, no es poden generar ni eliminar les dades guardades en la carpeta, però si es poden utilitzar els arxius continguts. El fet de guardar els arxius auxiliars en la carpeta "Assets" fa que el tamany definitiu de l'aplicació augmenti en consonància amb la grandària dels arxius.

En el projecte s'importen tots els mòduls de base de dades necessaris en la carpeta "Assets", per després poder-los carregar en l'aplicació i poder-ne fer ús de les dades.

Existeix una particularitat en *Android 2.1* i en versions anteriors, i és que no es poden utilitzar fitxers amb una grandària major a 1MB. Per tal de poder adjuntar i després carregar els mòduls de base dades majors d'1MB, cal dividir el fitxer de base de dades en diferents fitxers menors de la grandària límit.

Per carregar el mòdul de base de dades de la ciutat seleccionada, hi han dues opcions, carregar les dades a la memòria interna del telèfon o, carregar les dades a la targeta *SD* de l'usuari i treballar directament allà, sense haver de fer ús de la memòria interna. En qualsevol dels casos el que no es pot fer es eliminar la informació guardada dintre de la carpeta "Assets" del projecte, tal i com s'ha comentat anteriorment, ja que no hi ha permisos d'inserció o eliminació de dades en aquesta carpeta.

Abans de carregar la base de dades en una de les dues memòries, cal mirar si prèviament no s'ha carregat el mòdul de dades en una de les anteriors interaccions que ha fet l'usuari en l'aplicació, ja que si ja està carregada, el següent i últim pas seria obrir la base de dades per treballar amb ella. En cas que no hagi estat carregada anteriorment, verifiquem si la *SD card* està present (i es permet la lectura i escriptura) o si aquesta té espai suficient

per poder carregar les nostres dades. En cas que sigui així, es treballarà sobre aquesta opció i en cas negatiu s'escolliria la memòria interna del telèfon, en la qual també es verifica si hi ha espai a la memòria per guardar la base de dades pertinent.

Per mirar l'espai disponible a la memòria interna o a la *SD Card*, es fa ús de la classe *statFs* disponible a l'*API d'Android*, que proporciona informació general sobre l'espai en un sistema d'arxius. A partir d'aquestes dades, es fa un càlcul entre el tamany dels blocs i els blocs disponibles de la memòria seleccionada per poder aconseguir l'espai lliure que hi ha al sistema.

Un cop escollida la memòria, cal comprovar si la base de dades del mòdul seleccionat ha estat dividida en diversos fitxers menors d'1MB tal i com s'ha explicat anteriorment. En aquets cas, es generaria la base de dades a partir dels fitxers i es guardaria i obriria en la memòria seleccionada. En el cas de que no hagi estat dividida, es descarregaria i obriria directament per començar a treballar amb ella.

3.2.1 GESTIÓ MEMÒRIA INTERNA

Quan s'emmagatzemen fitxers a la memòria interna s'ha de tenir en compte les limitacions d'espai que tenen molts dispositius mòbils, ja que aquesta memòria acostuma a ser relativament limitada i no es aconsellable emmagatzemar en ella fitxers de gran tamany.

En general, les bases de dades de l'aplicació ocupen entre 100 i 500 kB, això comporta un petit problema ja que realment ocupen força espai si es compara amb el tamany que acostumen a tenir diverses aplicacions (sobretot aplicacions més simples). A més, per tal de tenir un accés més ràpid a les dades en propers usos de l'aplicació, no s'esborren mai les bases de dades de la memòria (a excepció que s'actualitzin a una nova versió), el que agreuja més la limitació d'espai que solen tenir els dispositius *Android*.

Per això a l'hora de guardar/carregar les bases de dades de les ciutats a memòria, es va decidir que la memòria interna no sigui la primera opció a tenir en compte, sinó que només s'utilitzi quan hi ha problemes amb la memòria externa. En canvi, les bases de dades de favorits i historials, són guardades a la memòria interna per que el seu tamany és molt petit.

3.2.2 GESTIÓ MEMÒRIA EXTERNA

El gran avantatge de la memòria externa és la seva gran capacitat d'emmagatzemament, on normalment no acostuma a ser ocupada en gran part (depenent l'usuari), per tant això permet tenir força llibertat a l'hora de guardar les bases de dades en aquesta memòria i treballar-hi directament.

En canvi, a diferència de la memòria interna, la memòria externa (*SD Card*) pot ser que no estigui present en el dispositiu mòbil, per tant, abans de començar a treballar amb la memòria externa cal assegurar-se sempre de que estigui present i disponible pels accessos de lectura i escriptura. Una altre desavantatge respecte la memòria interna és la velocitat, ja que la memòria externa és el tipus de memòria més lent de les presents en el dispositiu.

3.4 Temps Espera Parada

El mòdul de dades també s'ocupa d'obtenir la informació que les companyies d'autobusos ofereixen sobre el seu servei de "Temps d'espera a la parada", un servei que informa sobre el temps que trigarà el proper autobús que vol utilitzar l'usuari en arribar a la parada.

Aquest és un servei en temps real, per tant, no és una informació predefinida ni constant, si no que s'haurà d'obtenir quan l'usuari de l'aplicació ho demani. Depenent de la ciutat i de la companyia d'autobusos, aquest servei varia en la informació que mostra (pot donar més informació sobre l'autobús a banda del temps que triga) i en l'accés a les dades. Això implica que a l'hora d'obtenir les dades per mostrar-les a l'aplicació, s'implementarà un parser diferent per a cada ciutat. El parser implementat farà una consulta a la pàgina web que subministri el servei en cada ciutat i obtindrà tota la informació necessària. Un cop obtinguda, es processarà a normalitzar-la per poder-la utilitzar entre els diferents mòduls del projecte.

3.4.1 INFORMACIÓ DELS SERVEIS

- La informació que es pot trobar en els diferents serveis web és la següent:
 - Hora d'arribada de l'autobús a la parada o temps que falta en minuts.
 - Si l'autobús és accessible per persones amb mobilitat reduïda.
 - Incorporació d'aire condicionat.
 - Emissió de bitllets dintre de l'autobús.
 - Incorporació de pantalles informatives.
 - Destinació de la línia (hi han línies que poden finalitzar en dos destins diferents).
- Informació a la que es pot accedir en funció de les ciutats del nostre projecte:

	Temps	Accessible	A.Condicionat	Bitllets	Pantalles	Destinació
Barcelona	(**)	(*)	(*)	×	×	×
Madrid	✓	(*)	(*)	×	×	×
València	✓	(*)	×	×	×	✓
Saragossa	✓	(*)	×	×	×	×
Roma	✓	✓	✓	✓	✓	✓
Hèlsinki	✓	(*)	(*)	×	×	×

Taula 10. Informació Temps d'espera de les ciutats del projecte

(*) Informació no facilitada en el servei de "Temps d'espera a la parada" que disposa la companyia a les consultes web, però en canvi, si mostrada dintre de les especificacions de la flota.

(**) A Barcelona hi ha un particularitat amb el seu servei de "Nitbus", ja que es proporcionat per diverses companyies d'autobusos i cap d'elles ofereix el servei de "Temps d'Espera a la Parada" (l'empresa que opera amb els autobusos diürns no forma part). Per tant, pels autobusos que operen només de nit, s'ha decidit calcular (veure següent punt) el temps que falta en funció de la informació disposada. Cal dir que aquesta informació és aproximada i que no obté un grau d'efectivitat igual a la informació obtinguda en temps real.

3.4.2 MÈTODES DE CONSULTA AL SERVEI:

Dintre de l'aplicació és fa diferents usos d'aquesta funcionalitat, per tant el mode d'accedir i entrar les dades al servei varia en funció de l'activitat que cridi al servei:

- **Routing:** Servei de guiatge de la nostre aplicació. S'accedeix al "Temps d'Espera" cada vegada que l'usuari arriba a una parada en la que ha d'agafar l'autobús. En aquest cas, tenim internament totes les dades per fer la consulta al servei web (codi de parada i línia que cal utilitzar) i es mostra automàticament el temps al usuari, per tant aquest no ha de fer res per saber el temps que falta per a que arribi el seu bus.
- **Temps Espera Bus:** Activitat allotjada al "Menú Principal" de l'aplicació. Per la naturalesa de l'activitat si que cal interactuar amb l'usuari per poder fer la consulta web. Dintre d'aquesta funcionalitat hi han dues opcions, consulta del temps mitjançant la inserció manual per part de l'usuari en un box, gestures, etc., del codi de la parada i la línia (en aquest cas, l'usuari ha de saber o tenir a mà aquestes dades) o consulta del temps en funció de la proximitat de l'usuari amb les parades que té al voltant.

Per saber si l'usuari està a l'entorn d'una/es parada/es, es fa ús de la localització actual que tingui (interacció amb el *Nucli de l'Aplicació*) i es calcula la distància que hi ha entre la localització del usuari i la localització que tenen totes les parades de la ciutat corresponent (informació guardada a la base de dades). En funció dels radis (distància usuari - parada) definits prèviament, s'obtidran més o menys parades

properes. En el cas que el resultat sigui que l'usuari només està en l'entorn d'una parada, i que en aquesta parada només hi opera una línia, es mostraria automàticament el codi, línia i el temps que falta (a més de les diferents característiques de l'autobús) per a que arribés l'autobús desitjat. Si en canvi hi operen diferents línies, es mostraria al usuari les línies corresponents i s'esperaria a que aquest introdueixi la línia que vol consultar per poder fer la consulta a la web i mostrar les dades obtingudes.

Si pel contrari, l'usuari està a l'entorn de diferents parades, per cada una es mostraria el codi de parada i un llistat de línies que operen per aquella parada, repetint per tant, els passos comentats anteriorment per poder obtenir la informació del següent autobús.

3.4.3 ALGORITME DE CONSULTA AL SERVEI:

Per cada ciutat es té un mòdul per gestionar el servei que l'empresa d'autobusos ofereix per consultar el "Temps d'Espera". Cada mòdul segueix un mateix patró definit prèviament, on primer es genera la connexió amb Internet i a partir d'aquesta s'obté la informació que ofereixen els diversos serveis.

Per gestionar la consulta a Internet, es realitza una connexió a la url del servei que accedeix via web passant-li els paràmetres pertinents. Segons el servei de l'empresa d'autobús, de vegades fa falta passar-li el id de línia i parada, i d'altres vegades amb l'id de parada ja n'hi ha prou per fer la consulta.

Un cop s'ha accedit a la pàgina que mostra la informació, es parseja per obtenir les dades que interessin i aquestes es guardem internament, prèvia normalització, per poder mostrar-la en l'opció de l'aplicació que faci la petició. Cal normalitzar la informació perquè aquesta no sempre és present en un mateix format, com per exemple quan es mostra el temps que queda pel següent autobús, on de vegades els diferents serveis indiquen els minuts restants i d'altres només mostren la hora en la que passarà. Per tal de que sigui més clar i en relació amb altres funcionalitats del guiatge que també utilitzen aquest servei, l'aplicació sempre mostrarà al usuari el temps en minuts, per tant, quan es rep l'hora, es fa el càlcul pertinent per extreure els minuts. Una altre exemple referent a la informació del temps restant és el cas de que l'arribada del bus sigui imminent, és a dir que quedin segons per a que aquest arribi. Les diferents webs en aquets casos acostumen a mostrar diferents missatges

com “Arribada imminent” o “Autobús en entorn de parada”, en l’aplicació i seguint el patró esmentat abans, mostrarem els minuts, en aquest cas 0.

A més del temps restant hi ha vegades que el servei web també facilita alguna informació extra. Aquesta informació també es recol·lectada per després gestionar-la, però hi ha vegades que aquestes dades no són accessibles o presents en el servei web però en canvi, a partir de la recerca de dades feta al principi de tot per gestionar i omplir les bases de dades de cada ciutat/mòdul, es pot saber que són disponibles, en aquets casos s’ha optat per mostrar al usuari aquets extrems quan demanin la informació, com es pot veure a la [Taula 10](#).

Com s’ha comentat en el punt 3.6.1, hi ha un cas especial a la ciutat de *Barcelona* amb la consulta del temps d’espera amb els autobusos que operen només de nit, ja que aquest temps s’obté a partir de la implementació de diferents càlculs:

Per una banda a la nostre base de dades de *Barcelona* hi ha disponible tota la informació sobre els horaris d’inici i final de cada una de les línies que hi operen, el recorregut en ordre que fa la línia per cada parada i un valor que indica el temps aproximat que triga l’autobús entre parada i parada. A més, la pàgina web de l’*EMT* de l’àrea metropolitana de *Barcelona* indica que cada autobús surt de la seva capçalera de línia cada 20 minuts mentre el servei estigui disponible, per tant, amb aquesta informació es pot fer un càlcul força aproximat del temps que falta pel següent bus.

En el cas que l’usuari consulti el temps d’espera abans de la primera sortida (horari inici) de l’autobús des de la seva capçalera, només s’ha de calcular el temps que falta en la parada del seu recorregut. Per fer el càlcul d’aquest temps, s’obté en quina posició la parada és recorreguda i se li aplica els temps entre parades, així es pot saber quant temps trigarà en ser recorreguda una parada en condicions inicials. Aquest temps després serà aplicat al horari de començament de servei, per tal d’adaptar-lo amb les dades correctes i finalment obtenir els minuts comparant-ho amb l’hora actual.

Si pel contrari la consulta es fa un cop ja està el servei en marxa, s’han d’aplicar iteracions (de 20 minuts que es la freqüència de sortida de les capçaleres) al temps d’inici per a que aquest passi en temps al temps actual. Un cop es detecta que el temps és superior al actual i encara estigui la línia disponible (no hagi sobrepassat el horari de final de línia), es fa el càlcul per obtenir els minuts.

4 Proves i resultats

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

A continuació es detallen algunes de les proves més significatives realitzades i els resultats obtinguts. Es divideix en dues parts, primerament les proves unitàries(o de mòdul en cas que hi hagin) i en el segon punt les proves i resultats de camp a la ciutat de *Barcelona*.

4.1 Proves Unitàries

4.1.1 CARACTERÍSTIQUES DE LES BASES DE DADES (MODEL CIUTATS)

Ciutat	Tamany	#Línies	#Parades	#Trajectes
Barcelona	491 kb	244	3144	7742
Madrid	684 kb	432	4603	10996
València	184 kb	106	1123	2821
Saragossa	111 kb	73	752	1794
Hèlsinki	303 kb	192	1884	5563
Roma	1543 kb	885	8175	27407

Taula 11. Característiques de les bases de dades (Model Ciutats) de l'aplicació.

A la *Taula 11* es detalla el tamany de totes les bases de dades de ciutats presents en el projecte, a més del número de línies, parades i trajectes (Taula “passes” del diagrama conceptual de la *Figura 7*), que són els atributs que més influeixen en el tamany final de la base de dades al tenir més registres. Tot i que el número de línies i parades influeix, la quantitat de trajectes és la que acaba predominant, com es pot veure si es compara *Madrid* i *Roma*. En el cas de *Madrid*, si es doblés el número de línies i parades, es pot comprovar que superaria o igualaria als mateixos registres de *Roma*, però el tamany de la base de dades seria menor (el registre trajectes també és menor respecte si doblem el de *Madrid*).

4.1.2 RENDIMENT DE LES BASES DE DADES EN ELS DIFERENTS MÒBILS

Barcelona (491 kbs; 244 línies)

Telèfon	T.Obtenir línies
HTC Wildfire	25 s
HTC Magic	20 s
Huawei Ivy	29 s
LG P500	9 s
LG Optimus 2x(2.2)	2 s
LG Optimus 2x(2.2)	2 s

Madrid (684 kbs; 432 línies)

Telèfon	T.Obtenir línies
HTC Wildfire	28 s
HTC Magic	28 s
Huawei Ivy	31 s
LG P500	12 s
LG Optimus 2x(2.2)	5 s
LG Optimus 2x(2.3)	4 s

SARAGOSSA (111 kbs; 73 línies)

Telèfon	T.Obtenir línies
HTC Wildfire	5 s
HTC Magic	4 s
Huawei Ivy	7 s
LG P500	2 s
LG Optimus 2x(2.2)	1 s
LG Optimus 2x(2.3)	0,5 s

VALÈNCIA (184 kbs; 106 línies)

Telèfon	T.Obtenir línies
HTC Wildfire	7 s
HTC Magic	10 s
Huawei Ivy	10 s
LG P500	4 s
LG Optimus 2x(2.2)	1 s
LG Optimus 2x(2.3)	1 s

Helsinki (303 kbs; 192 línies)

Telèfon	T.Obtenir línies
HTC Wildfire	11 s
HTC Magic	9 s
Huawei Ivy	12 s
LG P500	4 s
LG Optimus 2x(2.2)	2 s
LG Optimus 2x(2.3)	1,5 s

Roma (1543 kbs; 895 línies)

Telèfon	T.Obtenir línies
HTC Wildfire	70 s
HTC Magic	68 s
Huawei Ivy	80 s
LG P500	40 s
LG Optimus 2x(2.2)	8 s
LG Optimus 2x(2.3)	7 s

Per tal de fer l'estudi del diferent rendiment de les bases de dades de ciutats a l'aplicació s'han utilitzat mòbils de gam

ma baixa (*HTC Wildfire* i *Magic*, *Huawei Ivy*), mitjana (*LG P500*) i alta (dos *LG Optimus 2x*, un amb la versió 2.2 d'*Android* i l'altre amb la 2.3).

Com es pot veure, la diferència de rendiment és molt gran, sobretot si es compara els mòbils de gamma baixa amb els de gamma alta en bases de dades de grans dimensions, com és el cas de Roma i en menor mesura Madrid i Barcelona. En canvi, en bases de dades amb poc volum d'informació continua existint la diferència de rendiment però no és tan pronunciada.

Amb els mòbils de gamma mitjana passa més o menys el mateix. En bases de dades de petites dimensions obtenen un rendiment semblant als de gamma alta, rendiment que no difereix molt quan ha de carregar les bases de dades de Madrid i Barcelona, que tenen un tamany més gran. En canvi al tractar amb una base de dades amb molts registres com la de Roma si que es pot veure que el seu rendiment baixa força.

És interessant analitzar el rendiment que pot donar un mateix mòbil però amb dues versions d'*Android* diferent, com el cas dels *LG Optimus 2x*. Com es pot veure, la versió més actual gairebé millora sempre el rendiment, tot i que les diferències només acostumen ser d'un segon. Això segurament pot ser degut a la diferent gestió que es fa entre una versió i l'altre de les diferents característiques del mòbil, com la memòria RAM o el processador.

4.1.3 FAVORITS ESTUDI

Com s'ha indicat en el punt 3.2.2.2.2, es va realitzar un petit estudi per veure l'espai que ocupen les diferents bases de dades "Favorits", per veure si es podia generar una que englobés tots els favorits de les ciutats o una base de dades exclusiva per cada ciutat. Els resultats són els següents:

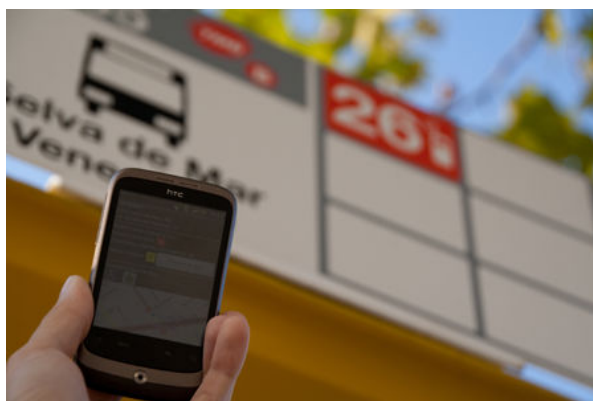
# Registres BD	Tamany BD
1	5 kB
25	5 kB
50	5 kB
75	7 kB
100	7 kB

Taula 12. Estudi Base de Dades Favorits

Es pot concloure que al crear la base de dades es reserva espai per futures insercions, i que aquest no augmenta si no s'insereixen molts registres, per tant, per la naturalesa de la funcionalitat, és millor tenir una base de dades de favorits única en tota l'aplicació.

4.1.4 TEMPS ESPERA PARADA

S'ha generat un conjunt de test per veure el nivell de precisió real del temps d'espera a la parada proporcionat per TMB.



- Test 1: Precisió Correcte.

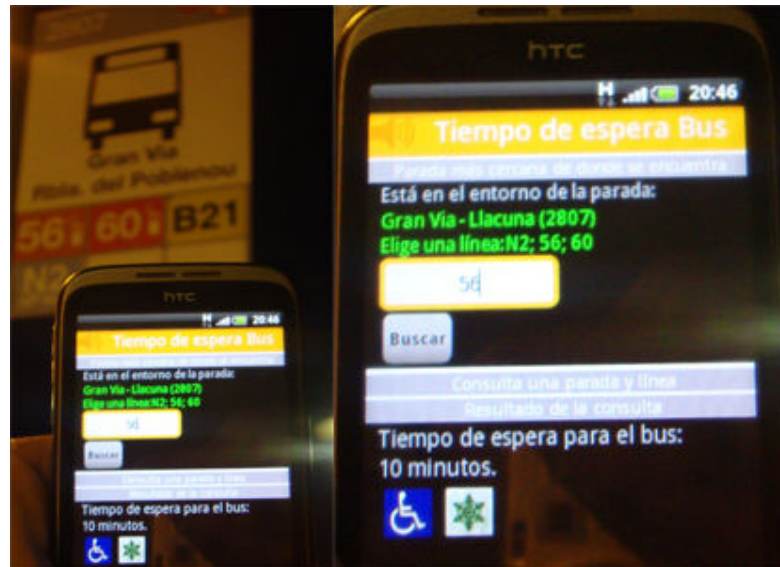


Figura12. Precisió correcta.

Com es pot veure en la *Figura 12*, l'aplicació detecta automàticament la parada en la que l'usuari està esperant el bus (codi 2807). En aquest cas, al haver-hi diferents línies l'usuari ha d'introduir la línia que vol utilitzar.

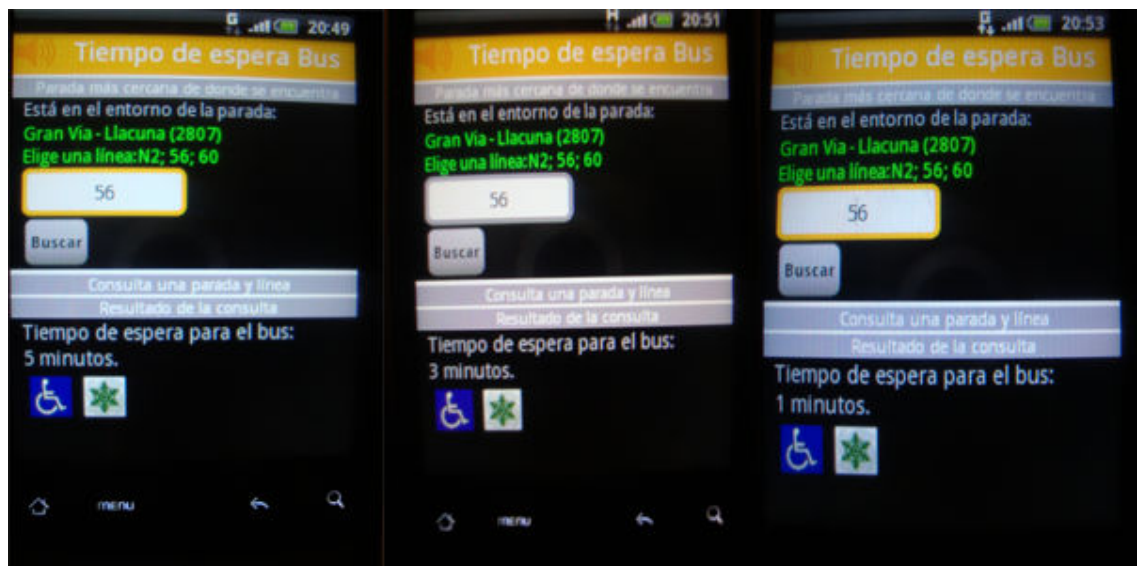


Figura 13. Successió

Si s'observa el temps que marca l'aplicació i l'hora que marca el telèfon mòbil, es pot veure que l'autobús sembla que arriba sense cap retard a la parada.

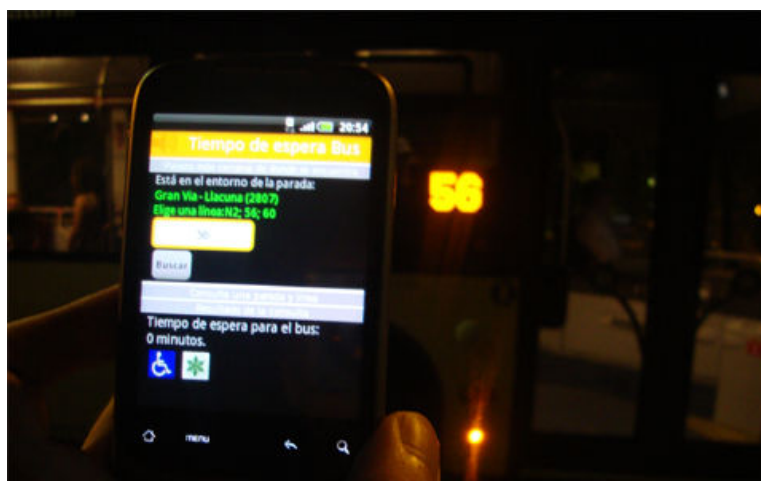


Figura 14. Precisió correcte.

Finalment, es pot apreciar que l'autobús, ha arribat a l'hora prevista.

- Test 2: Error per falta d'informació.

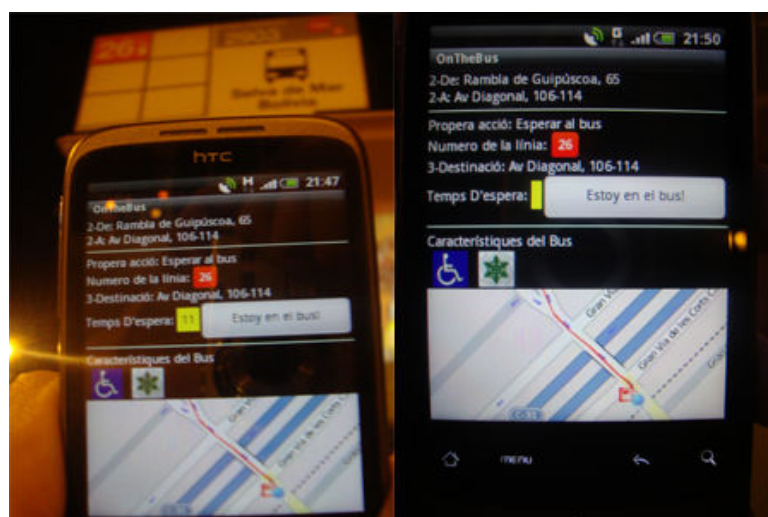


Figura 15. Successió Téms Espera errònia. Falta d'informació.

Hi han casos en que la informació del temps d'espera no és disponible ja que falla la consulta a Internet per la connexió o perquè el proveïdor del servei no té les dades operatives, com es pot veure en la [Figura 15](#), on passem de mostrar els minuts restants a no poder mostrar res.

- Test 3: Retard en l'arribada del autobús.

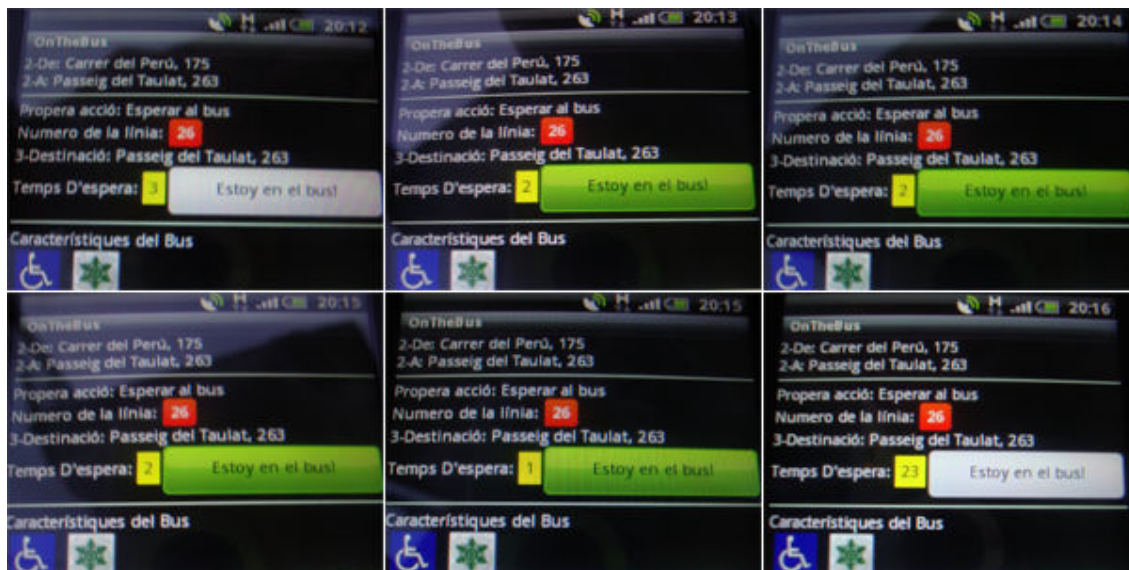


Figura 16. Successió Temps Espera. Retard arribada.

A la [Figura 16](#) es pot observar que l'autobús ha petit un retard en el seu trajecte i trigarà més dels tres minuts que marcava la consulta inicial. Durant tres minuts de rellotge (20:13 – 20:15), el temps d'espera marcat per l'aplicació és el mateix, dos minuts. També s'hi produeix una imprecisió quant està a punt d'arribar a la parada, ja que es passa d'un minut restant a vint-i-tres, és a dir, hauria d'haver arribat i ja s'està esperant al següent.

En aquesta sèrie de testos es pot veure que de vegades el temps que falta pel proper autobús potser no concorda del tot amb la seva arribada, tot i tenir una precisió força alta en global. En el test 3 per exemple, es passa d'un minut restant a vint-i-tres, tot i que el bus no ha passat per la parada i pot estar parat en un semàfor o a uns metres. Els minuts que falten s'obtenen a través d'una consulta al servei web de la companyia en qüestió, per tant, depenem exclusivament d'aquest servei i no es pot generar un control en cas de diferències entre la informació que mostrem i l'arribada del bus.

4.2 Proves i resultats de camp

Proves realitzades a Barcelona. Per la resta de mòduls (ciutats) consultar l'Annex B. Les proves fan referència al perfil sense discapacitat. El sistema de guiatge és el mateix per tots els perfils, només varia la interfície, per tant, s'obtingran els mateixos resultats.

4.2.1 PROVES I RESULTATS DE LOCALITZACIÓ

En aquest apartat es mostraran els diferents resultats obtinguts al obtenir la localització actual del usuari:

- Localització Precisa:

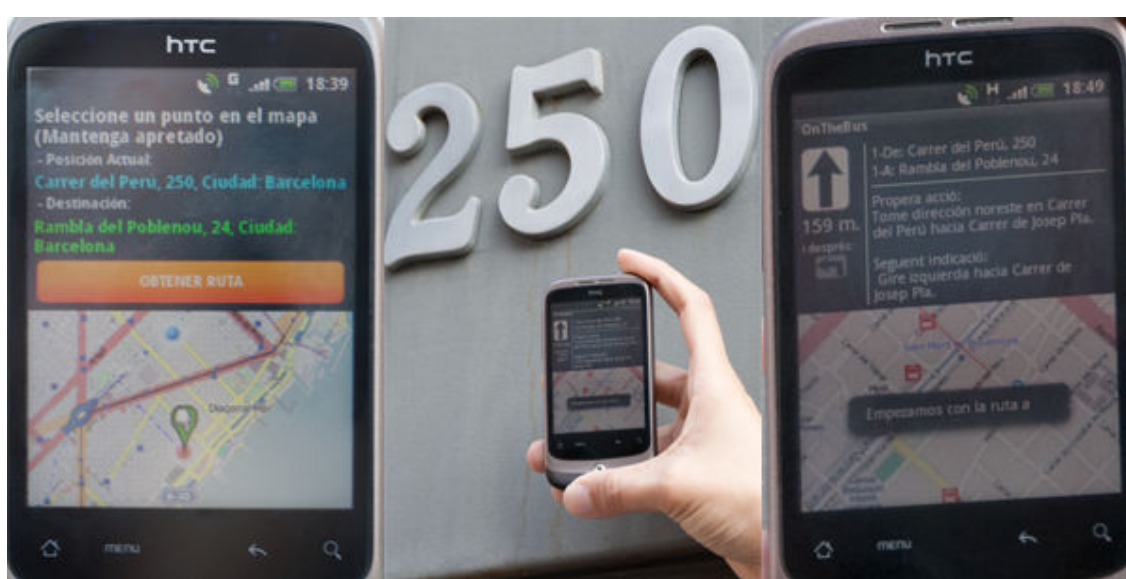


Figura 17. Localització amb perfecte precisió.

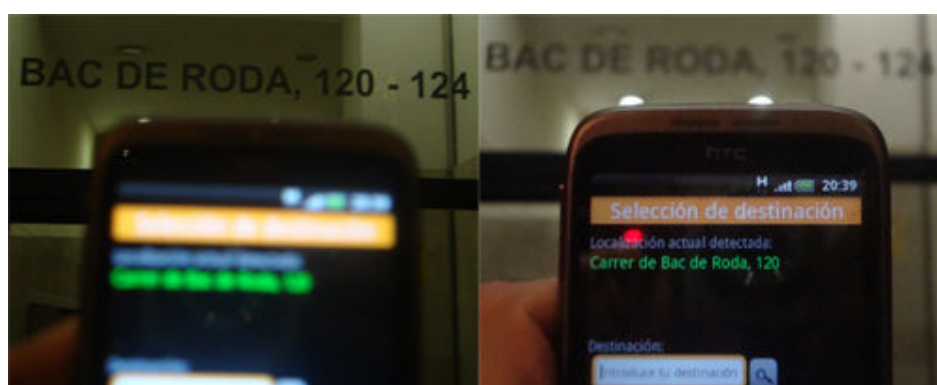


Figura 18. Localització amb perfecte precisió.

- Errors de localització:

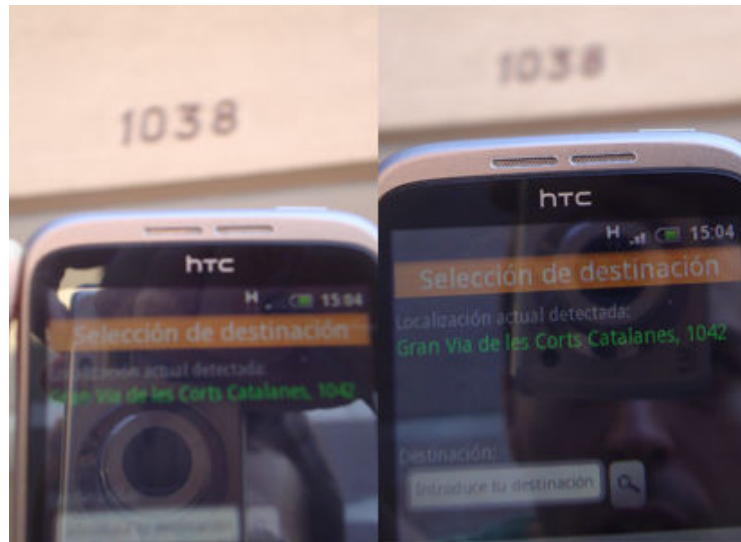


Figura 19. Localització errònia.

Es aquest cas es pot comprovar que hi ha un imprecisió. L'usuari està al carrer *Gran Via*, número *1038*, però la localització marca el número *1042* del mateix carrer. Això és degut a que el *Geocoder* no mostra tots els números d'un carrer, sinó que només mostra uns quants en un segment determinat i per tant ens treu precisió a l'hora d'obtenir i mostrar l'adreça actual del usuari.

Un error d'imprecisió semblant i derivat del *Geocoder* el trobem en la *Figura 20*. En aquest cas es pot veure que l'usuari està una mica lluny de la seva posició actual i a més que el *Geocoder* retorna un conjunt de números en els quals pot estar l'usuari, per tant, la precisió en cas d'estar en el conjunt de números, tampoc serà total.

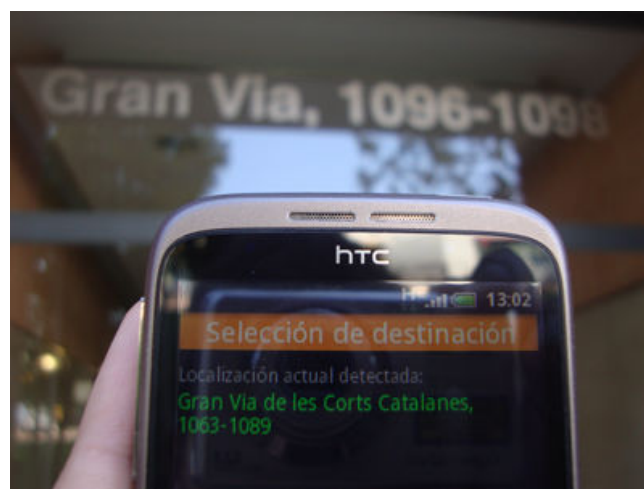


Figura 20. Localització errònia, agrupació de números.

4.2.2 PROVES I RESULTATS DE GUIATGE

4.2.2.1 Guiatge WALK

- Guiatges



Figura 21. Exemple simple de gir.

En la *Figura 21* es pot veure com avança l'usuari pel camí i la indicació marcada. Just en el punt en que l'usuari ha de girar per un carrer, li es mostrada la indicació de gir sense cap tipus d'error. En les següents il·lustracions (*Figura 22*) es mostrarà uns exemples de guiatge amb un bona precisió de girs.

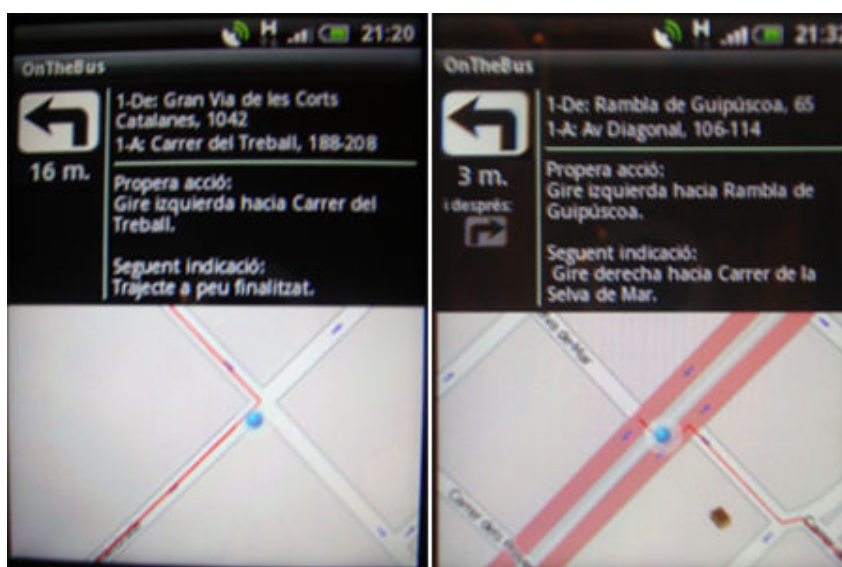


Figura 22. Guiatge amb un bona precisió de girs

- Recalcular la ruta

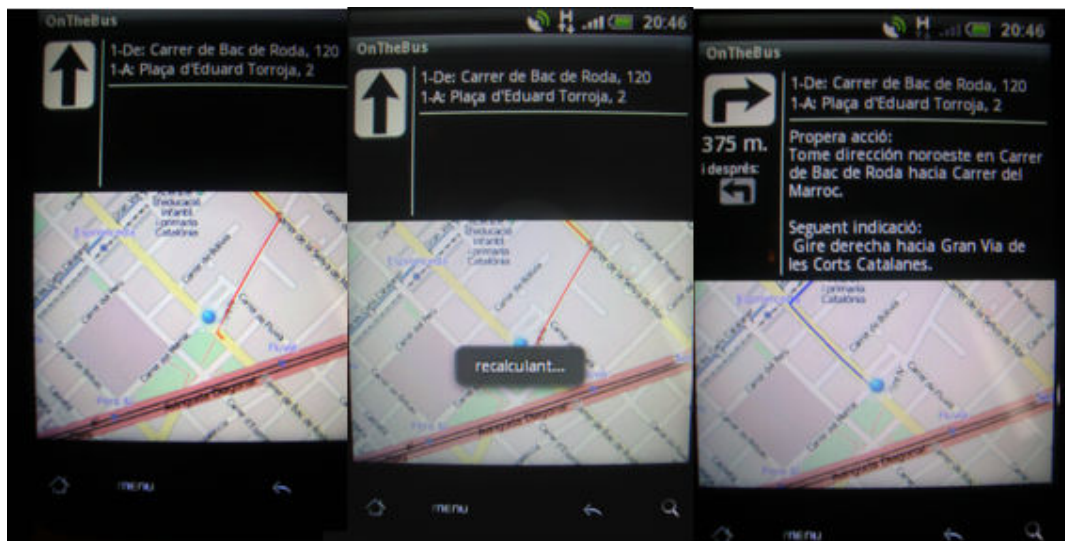


Figura 23. Recalcular Ruta.

Es pot apreciar en la primera imatge de la [Figura 23](#) que l'usuari ha iniciat el guiatge de caminar però que no li arriben indicacions, ja que mentres es calculava la ruta ha mogut la seva posició (es pot apreciar la diferència entre l'inici del camí vermell i el punt de localització). En les dues següents imatges es pot veure com l'aplicació detecta l'error i automàticament recalcula la ruta de caminar fins al proper punt de guiatge o destí de l'usuari.

- Errors
 - GPS

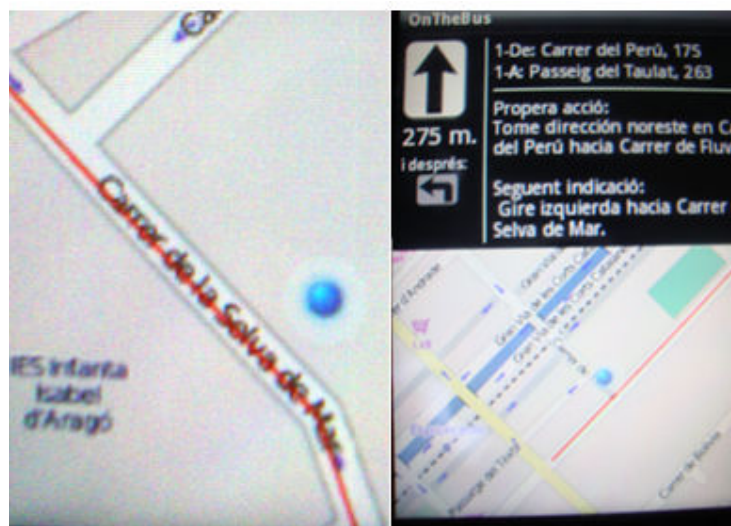


Figura 24. Guiatge Walk. Error localització GPS.

En la [Figura 24](#) es pot apreciar que el punt de localització està lluny del camí que hauria de seguir l'usuari (marcat en vermell) i que a més, el punt marca que l'usuari està dintre d'un edifici. Aquest error de localització passa de tant en tant i és molt difícil de controlar, ja que deriva de l'error de GPS que tenen els dispositius mòbils.

- Errors Mapa

Hi ha vegades que el mapa que utilitzem de *OpenStreetMap* no s'acaba d'aproximar amb l'estat actual dels carrers, i això ens indueix principalment a dos tipus d'error:

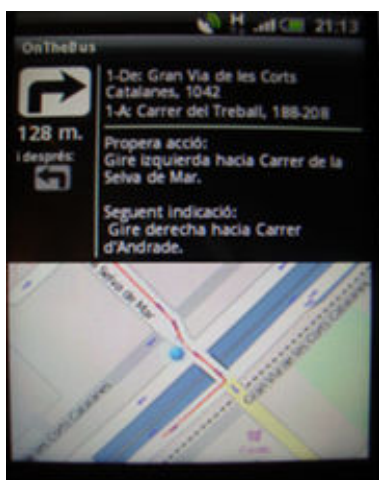
- Cantonades



Sovint el mapa mostra cantonades amb girs de 90° (amb forma de “quadrat”, com es pot veure a la [Figura 25](#)) que no s'aproximen amb la realitat, ja que moltes han sigut adaptades en forma de diagonal per aprofitar millor l'espai públic. Aquest fet indueix a imprecisions en el guiatge ja que ens ajustem al mapa proporcionat per poder implementar el servei. Per tal de solucionar-ho s'ha augmentat el radi de la *proximity alert* que ens avisa que l'usuari ha de girar, per a que aquest ho pugui fer correctament, com es pot veure en l'exemple.

Figura 25. Guiatge Walk. Cantonades

- Voreres



És el cas més complicat i de moment impossible de controlar degut a la seva complexitat i dificultat. El servei proporciona una ruta e indicacions que son mostrades al usuari en el format indicat. El problema deriva en que moltes vegades quan hi ha un canvi de vorera no se'ns proporciona aquesta informació, i per tant, no es pot indicar al usuari que canviï de vorera quan es fa el gir. En el mapa de la [Figura 26](#) es pot veure com es realitza un canvi de vorera però no s'informa en cap moment en les indicacions.

Figura 26. Guiatge Walk. Voreres

4.2.2.2 Guiatge BUS

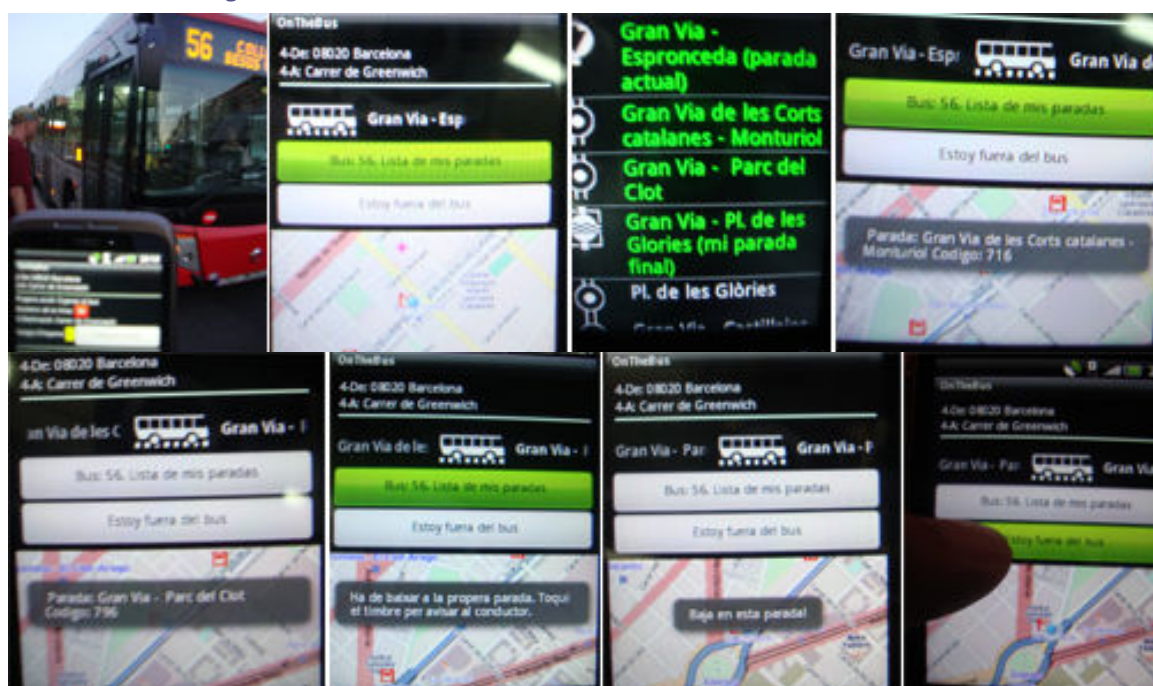


Figura 27. Guiatge Bus. Etapes de Guiatge

Com es pot veure, un cop l'usuari puja a l'autobús pot consultar les parades que recorrerà, i cada cop que passi amb aquest per una parada rebrà un avís on l'indicarà la parada actual del recorregut. Durant el trajecte entre la penúltima i última parada, es mostra un missatge cada cert temps alertant a l'usuari de que ha de baixar en la següent parada i finalment al arribar a la última parada, s'informa al usuari de que ha de baixar. Un cop ho hagi fet, l'usuari confirma a l'aplicació que ha deixat l'autobús per continuar amb el següent guiatge.

- Problema: Passar-se de Parada

En aquest cas, l'aplicació avisa a l'usuari de que no ha baixat en la parada correcta i que cal recalculer la ruta. Un cop l'usuari baixa del autobús, es recalcula automàticament la ruta fins el destí (línia blava de la última imatge) i comença de nou el guiatge.

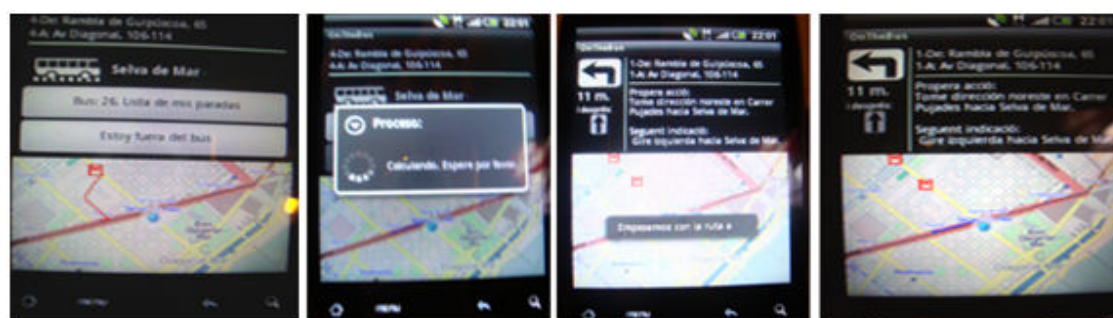


Figura 28. Guiatge Bus. Exemple de passar-se de parada

5 Conclusions i possibles ampliacions

5.1 Conclusions Personals

L'objectiu principal del projecte era desenvolupar un sistema de guiatge sobre el sistema operatiu *Android*, que permeti a una persona desplaçar-se per una ciutat mitjançant el transport públic, i crec que s'ha pogut assolir correctament.

Tot i no haver obtingut una fiabilitat total en alguns casos particulars de guiatge degut a alguns problemes derivats del error de localització de *GPS* o els problemes durant el guiatge amb les cantonades i les voreres (veure punt 4.2.2.1 a “Proves i Resultats”), s'ha obtingut bons resultats i l'aplicació és funcional, és a dir, està preparada per exemple per a que els usuaris puguin interaccionar amb ella sense problema.

Com es pot veure en tot el conjunt de memòries dels mòdul que formen el projecte, s'han implementat i afegit a l'aplicació diverses funcionalitats que complementen al sistema de guiatge i d'altres fora d'aquest, relacionades amb els serveis d'autobusos amb les que no hi comptaven al principi i que fan que l'aplicació sigui molt completa. Això ens pot servir per guanyar terreny enfront d'altres aplicacions que hi al mercat amb una funcionalitat semblant.

Aquest projecte té la particularitat de que no és un projecte individual, sinó que està integrat per un grup de set persones on s'ha buscat simular el funcionament d'una empresa de desenvolupament. L'experiència crec que ha sigut molt bona, tothom s'ha involucrat amb el projecte i ha hagut molta comunicació, coordinació i treball en equip, tal i com suggereix el model de treball que es vol implantar actualment a la majoria d'empreses. Com a mostra, durant pràcticament tot el temps de desenvolupament del projecte, s'han realitzat reunions setmanals per analitzar els avanços i les futures accions a realitzar.

El fet de treballar per mòduls ha sigut força interessant i no excloent, és a dir, tot i tenir assignat un mòdul dintre del projecte, he donat suport a la resta de companys i així he pogut ampliar els meus coneixements sobre *Android* (que era un dels meus objectius principals a l'hora d'escollir aquest projecte), i he pogut participar en la realització de la resta de l'aplicació i no només d'una part.

5.2 Conclusions Generals

Els objectius principals del projecte s'han assolit. S'ha implementat una solució seguint el disseny universal i que funciona, realitzant correctament el guiatge des del punt inici on es troba l'usuari fins un punt destí fent ús de la xarxa de transport d'autobusos per a diferents ciutats d'arreu del món.

També s'ha aconseguit gestionar diferents grups homogenis en el desenvolupament d'un projecte de dimensions importants creant sinergies entre aquest grups per a assolir l'objectiu principal.

Alguns dels inconvenients que han aparegut han estat relacionats amb la precisió del dispositius GPS que incorporen els mòbils que s'han usat durant les proves de l'aplicació, donant diferents resultats cadascun d'ells. Això ha fet que s'allargués la duració del projecte per sobre del que s'havia previst inicialment. Aquesta desviació ha suposat aproximadament cinc setmanes de retràs, que han fet que l'entrega es realitzi en setembre.

5.3 Possibles Ampliacions Mòdul de Dades

Una possible ampliació força interessant en l'aplicació seria el fet de tenir totes les bases de dades de les diferents ciutats en un servidor d'Internet (Cloud Computing). El fet de poder utilitzar un facilitaria molt les coses en quant a emmagatzematge i actualització, en canvi caldria fer una comparativa sobre la velocitat i la consistència de les consultes entre les bases de dades guardades al telèfon i les bases de dades allotjades al servidor.

A l'hora d'actualitzar de les bases de dades, les actualitzacions serien més constants, ràpides i fàcils, tant per els desenvolupadors, com per l'usuari. De cara als desenvolupadors això permetria fer múltiples actualitzacions per petites que siguin i per tant, tenir les bases de dades sempre actualitzades amb els últims canvis. No caldria esperar a un volum determinat de canvis per fer una actualització generant i pujant constantment diferents versions al *Android Market*.

De cara al usuari hi haurien dues opcions, per una banda no descarregar-se cap base de dades i fer totes les consultes a les que hi ha allotjades al servidor, el que permetria no ocupar espai en la memòria interna del mòbil o en la *SD Card*. I per l'altre fer l'actualització cada cop que accedeixi a l'aplicació, ja que al iniciar-la es descarregaria sempre la última versió de la base de dades corresponent i no caldria que fes absolutament res més.

D'altre banda es deixa obert el número de mòduls o ciutats presents en el projecte. Tal i com està preparada l'aplicació, en qualsevol moment es pot afegir una nova ciutat en el cas que aquesta compleixi els requisits mínims marcats en el punt 3.1.1.1. Sobretot seria interessant ampliar l'aplicació amb més ciutats europees o de la resta del món, aprofitant que l'aplicació dona suport a diverses llengües i així donar a conèixer l'aplicació en més llocs.

Una altre ampliació que es pot fer en el futur és la sincronització dels favorits de l'aplicació amb els favorits de Google Maps, tal i com s'ha pogut veure en el punt 3.2.2.2.1.

5.4 Possibles Ampliacions de l'aplicació

A continuació es presenten algunes de les possibles ampliacions a realitzar en l'aplicació. Es tracta de noves funcionalitats afegides, que resten pendents d'anàlisi i disseny.

Cerca per punts d'interès (POI): Es tractaria d'afegir la geolocalització de diferents punts d'interès per tipologia (comissaries, farmàcies, restaurants, etc.). Encara que queda pendent d'anàlisi aquesta modificació implicaria la modificació de la majoria del mòduls de l'aplicació.

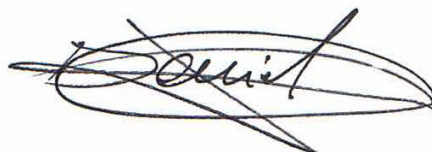
Rutes turístiques: Es tractaria de tenir rutes complertes d'inici a fi amb un o més punts turístics. Es podria enllaçar amb l'ampliació anteriorment esmentada per tal d'emmagatzemar els punts turístics com a punts d'interès.

Recorregut Invers: Es tractaria de poder indicar-li a l'aplicació que emmagatzemi el punt inici i final de ruta per a en un breu espai de temps demanar que es guïï de tornada al punt inicial. És una ampliació interessant per a quan et desplaces puntualment a un destí.

Cercar un pla de negoci: Consistiria en l'addició d'algun tipus de publicitat en l'aplicació.

- Desenvolupament
 - Api Android: <http://developer.android.com>
 - SQLite: <http://www.sqlite.org/>
 - SQLiteDatabrowser: <http://sqlitebrowser.sourceforge.net/>
 - Ableson Frank; Collins Charlie; Sen Robi. *Android Guía para Desarrolladores*. 1a ed. Madrid: Anaya Multimedia, 2010. 464p.
- Pàgines web relaciones amb els serveis d'autobusos de les ciutats:
 - Àrea Metropolitana de Barcelona: <http://www.emt-amb.com>
 - Transports Metropolitans de Barcelona : <http://www.tmb.cat>
 - Empresa Municipal de Transportes de Madrid: <http://www.emtmadrid.es>
 - Empresa Municipal de Transportes de València: <http://www.emtvalencia.es>
 - Transportes Urbanos de Zaragoza: <http://www.tuzsa.es>
 - Azienda per la mobilità Roma: <http://www.atac.roma.it>
 - Helsinki Region Transport: <http://www.hsl.fi>
 - Google Transit: <http://www.google.com/transit>
- Altre informació:
 - Wikipedia: <http://wikipedia.com>
 - Cartesia: <http://www.cartesia.org/>
 - Convertidors de coordenades: <http://www.gabrielortiz.com/>

Memòria del Projecte realitzat per

A handwritten signature in black ink, appearing to read 'Daniel', enclosed within a large, stylized oval loop.

Daniel del Arco Gómez

Bellaterra, 12 de setembre de 2011

El projecte OnTheBus pretén ser un sistema de guiatge gratuït, destinat per a la plataforma mòbil Android, per desplaçar-se d'un punt origen a un punt destí dins de la ciutat utilitzant la xarxa de transport metropolità.

Aquesta memòria s'enllaça amb sis memòries més, dels companys que han realitzat el projecte OnTheBus. Es descriu el Mòdul de Dades (BDD), el qual recol·lecta i proporciona a l'aplicació tota la informació necessària del diferents sistemes d'autobusos, per a que aquesta pugui assolir el guiatge correctament.

El proyecto OnTheBus pretende ser un sistema de guiado gratuito, destinado para la plataforma móvil Android, para desplazarse de un punto de origen a un punto destino dentro de la ciudad, utilizando la red de transporte metropolitano.

Esta memoria se enlaza con seis memorias más, de los compañeros que han realizado el proyecto OnTheBus. Se describe el Módulo de Datos (BDD), el cual recolecta y proporciona a la aplicación toda la información necesaria de los diferentes sistemas de autobuses, para que esta pueda realizar el guiado correctamente.

The OnTheBus Project pretends to be a guiding free system, designed for the Android platform, with the target to go from one source point to a destination point in a chosen city, using the metropolitan transport network.

This study links with another six works from my partners who we have been developing this project. It is described the Data Module(BDD), which collects and provides the application with all the useful information regarding the different bus systems, in order to enable it to perform an accurate guidance.