



BCNg - Aplicació Android per a una millor experiència d'ús del servei de Bicing

Nom Estudiant: Andoni Carmona Chavarría

Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils

Nom Consultor/a: Eduard Martin Lineros

Professor/a responsable de l'assignatura: Carles Garrigues Olivella

06/06/2018



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

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Aplicació Android per a una millor experiència d'ús del servei de Bicing</i>
Nom de l'autor:	<i>Andoni Carmona Chavarría</i>
Nom del consultor/a:	Eduard Martin Lineros
Nom del PRA:	<i>Carles Garrigues Olivella</i>
Data de lliurament (mm/aaaa):	06/2018
Titulació o programa:	<i>Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Bicing, android, app.</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>L'objectiu d'aquest projecte és donar una alternativa a l'aplicació oficial de Bicing que suposi una millora respecte a l'actual, amb un disseny més concordant amb els actuals patrons de disseny d'Android i alguna funcionalitat afegida.</p> <p>Per a dur a terme aquest desenvolupament s'ha seguit una metodologia bastant en cascada. Primer la capta de requeriments, després el disseny, i finalment, la implementació de l'aplicació. Tot i que també en alguna de les etapes posteriors s'ha fet alguna iteració, sobretot al desenvolupament s'han iterat alguns aspectes del disseny.</p> <p>Tenint en compte els estudis previs i el disseny plantejat s'ha creat una aplicació que permet veure les estacions del servei del Bicing i el seu estat ordenades segons la posició de l'usuari (si aquest ho permet) en forma de mapa o llista. També permet la creació de "l·listes" de favorits, les quals permeten gestionar els favorits en funció de llocs que l'usuari visiti freqüentment.</p> <p>Amb aquest projecte s'ha aconseguit dur a terme un projecte en la seva totalitat, des de la capta de requeriments fins a la realització de la quality assurance del projecte. Aplicant els coneixements adquirits al màster, resultant en una aplicació apta per a ser utilitzada pels usuaris finals.</p>	

Abstract (in English, 250 words or less):

The objective of this project is to give an alternative to the official application of the Bicing which supposes an improvement respect the current one, with a design more inline with the Android design patterns and some new functionalities.

In order to realize this project, it has been followed cascade methodology. First the requirements capture, then the design and, finally, the implementation of the application. Although sometimes an iteration has been made on the previous steps.

Taking into account the previous studies and the design planned, an application that allows users to see Bicing stations and their state, ordered according to user's location (if he allows it) in the form of either map or list has been made. Also it allows the creation of lists of favourites, which allow to manage favourites according to user's more frequented locations.

With this project it has been achieved to do a project in its all totality, from the requirements capture until the quality assurance. Using the knowledge acquired in the master, resulting in an application that can be used by final users.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball	3
1.3 Enfocament i mètode seguit.....	4
1.4 Planificació del Treball	5
1.5 Breu sumari de productes obtinguts	8
1.6 Breu descripció dels altres capítols de la memòria	9
2. Anàlisi.....	10
2.1 Estat de l'art	10
2.2 Estudi de requisits	13
2.3 Usuaris i context d'ús	13
2.4 Requisits	14
3. Disseny	15
3.1 Escenaris d'ús.....	15
3.2 Arbre de navegació	16
3.3 Selecció del nom, el logotip i l'estil de l'aplicació	17
3.4 Prototip.....	18
4. Arquitectura	24
4.1 Arquitectura de l'aplicació.....	24
4.2 Estructura de les dades.....	26
5. Desenvolupament.....	27
5.1 Introducció	27
5.2 Eines utilitzades	29
5.3 Estructura del codi.....	31
5.4. Proves realitzades.....	36
6. Manual d'ús de l'aplicació	39
6.1. Vista Mapa	39
6.2 Vista llistat.....	40
6.3 Favorits	40
7. Conclusions.....	42
8. Glossari	43
9. Bibliografia	44
10. Annexos	46
6.1 Annex A – Enquesta repartida (En vermell els resultats).....	46
6.2 Annex B – Estructura de l'arquitectura de l'aplicació	48

Llista de figures

Figura 1. Estació del sistema de Bike Sharing de Toronto (Canada).	1
Figura 2. Logotip del servei de Bicing.	1
Figura 3. Captura de pantalla de l'app oficial del Bicing.	2
Figura 4. Taula que mostra els recursos humans necessaris per a dur a terme el projecte i la seva implicació en hores/persona.	5
Figura 5. Taula que mostra els recursos materials necessaris per a dur a terme el projecte.	6
Figura 6. Taula que mostra les fites i tasques del projecte, desglossades en les hores planificades per a dur-les a terme.	7
Figura 7. Diagrama de Gantt de les fites i les tasques amb les seves dependències en el temps.	8
Figura 8. Menú principal de l'aplicació oficial del Bicing.	10
Figura 9. Vista detall d'una estació de l'aplicació oficial del Bicing.	11
Figura 10. Pantalla principal de l'aplicació CityBiker.	12
Figura 11. Pantalla principal de l'aplicació Bicing Barcelona.	12
Figura 12. Diagrama de cas d'ús corresponent a quan un usuari vol saber l'estat d'una bicicleta.	15
Figura 13. Diagrama de cas d'ús corresponent a quan un usuari vol afegir una estació a favorits.	16
Figura 14. Arbre de navegació de l'aplicació.	16
Figura 15. Logotip de l'aplicació.	17
Figura 16. Pantalla principal (vista mapa) del prototip.	18
Figura 17. Pantalla principal (vista de llistat) del prototip.	19
Figura 18. Pantalla principal (vista de mapa) amb una estació seleccionada.	20
Figura 19. Pantalla principal (vista de mapa) amb una estació elèctrica seleccionada.	20
Figura 20. Pantalla de selecció de llista favorita (vista llistat).	21
Figura 21. Pantalla de selecció de llista favorita (vista mapa).	21
Figura 22. Vista lloc favorit mapa.	22
Figura 23. Vista lloc favorit llistat.	22
Figura 24. Popup selecció favorit.	23
Figura 25. Popup creació favorit.	23
Figura 26. Taula de distribució en percentatge de versions d'Android	27
Figura 27. Logotip del llenguatge de programació Kotlin	28
Figura 28. Logotip d'Android Studio	29
Figura 29. Logotip de GoogleMaps	29
Figura 30. Logotip de RxJava	30
Figura 31. Estructura del mòdul domain.	32
Figura 32. Estructura del mòdul data.	33

Figura 33. Estructura del mòdul Presentation.	34
Figura 34. Estructura del mòdul d'app	35
Figura 35. Captura de pantalla de la vista de mapa de l'aplicació	39
Figura 36. Captura de pantalla de la vista de llista de l'aplicació	40
Figura 37. Popup per a seleccionar favorit.	40
Figura 38. Popup per a afegir un favorit.	41

1. Introducció

1.1 Context i justificació del Treball

1.1.1 El Bike Sharing

El Bike Sharing^[1] és un transport que es va fer popular a finals de la dècada passada i que consisteix en la utilització de bicicletes públiques localitzades en diferents punts o estacions, que permeten esser utilitzades per al transport d'una estació a una altra.

En la majoria d'aquests tipus de transports, el servei físic ve acompanyat d'una aplicació mòbil la qual dona informació a l'usuari i li facilita la utilització del servei, indicant la localització de les estacions i la seva disponibilitat de bicicletes / slots, també si pel que sigui no estan disponibles, etc.



Figura 1. Estació del sistema de Bike Sharing de Toronto (Canada).

1.1.2 El Bike Sharing a Barcelona



Figura 2. Logotip del servei de Bicing.

A Barcelona aquest transport rep el nom de Bicing^{[2][3]}. El Bicing va esser implantat a Barcelona al 2007, finançat per l'ajuntament i gestionat per l'empresa Clear Channel.

Durant aquests 11 anys de servei el Bicing ha anat expandint-se i afegint millores al servei, des de començar amb menys de 100 estacions de bicicletes fins a les 465 estacions de bicicletes que hi ha avui en dia. També cal destacar la inclusió d'una nova modalitat de bicicleta elèctrica, la qual dona assistència al pedalejar i que té les seves pròpies estacions i modalitat de pagament.

El servei també conta amb la seva pròpia app, la qual està dissenyada per a donar informació del servei i de l'usuari entre d'altres. Al contrari que el servei físic que si que ha vist una important i constant evolució, l'aplicació mòbil no sembla haver estat participa d'aquesta. L'última actualització de l'aplicació data del 2 de març de 2017* i mai ha destacat per a esser una aplicació amb un bon disseny i usabilitat.

*A dia 02/03/2018

1.1.3 L'app del Bicing

L'app oficial^[4] del Bicing porta ja molts anys en funcionament, primer en va sortir la versió per a iOS i posteriorment en va sortir una rèplica d'aquesta en Android.

Aquesta app és funcional a nivell de que es poden cobrir els casos d'ús bàsics com veure l'estat de les estacions en temps real, guardar un llistat d'estacions favorites... Però no té una usabilitat molt bona, en aquesta pantalla per exemple podem veure que la "App bar" té ni més ni menys que 5 accions, això a part de sobrecarregar l'app bar fa que en dispositius de petit tamany l'experiència no sigui optima.

També se li podrien afegir millores funcionals com per exemple establir ubicacions preferides (ja que molts del trajectes que fan els usuaris son a les mateixes localitzacions) amb les estacions properes.

Finalment també esmentar que al esser una replica de l'app d'ios antiga (la actual a l'App Store té elements visuals nous que aquesta no té) no compleix amb els estandards de Google per al disseny de les apps, l'anomenat Material Design^[5], que dona unes pautes de disseny i també de patrons d'usabilitat que les apps d'android haurien de "seguir".

1.1.4 Altres aplicacions

Hi ha més apps a part de l'oficial que també s'ofereixen com a alternativa, la majoria però són pitjors en quant a usabilitat que la pròpia app del bicing o en el seu defecte son massa simples i només permeten el visionat de les estacions i el seu estat.

Per exemple CityBiker^[6] té un disseny més atractiu, encara que no acaba de complir amb les guies d'estil de Material, però no permet per exemple guardar estacions a favorits.

Una altra app que es presenta com a alternativa és Bicing Barcelona^[7], que té moltes més funcions però el disseny dista bastant de l'esperat en una app d'android.

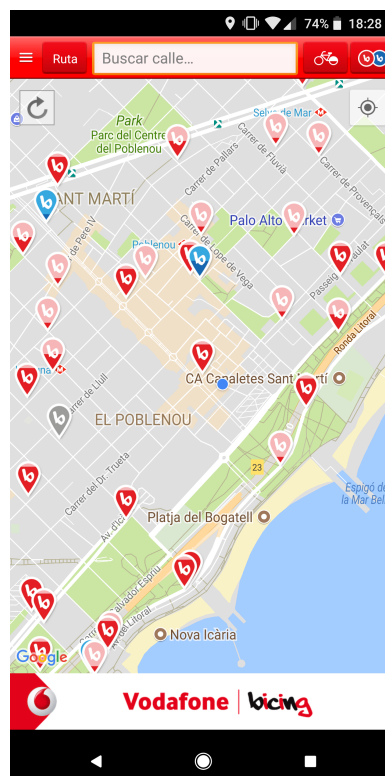


Figura 3. Captura de pantalla de l'app oficial del Bicing.

1.1.5 Justificació del Treball

Amb tot el context donat i un cop explorades les opcions als punts anteriors, la idea d'aquest treball és fer una aplicació del Bicing, que implementi els estàndards de disseny de Material Design i que permeti en primera instància, a part de poder veure l'estat de les estacions del Bicing alguna funcionalitat extra com es ara la d'una gestió de favorits en funció de llocs establerts per l'usuari. D'aquesta manera es busca donar una bona alternativa per als usuaris d'aquest servei de transport.

1.2 Objectius del Treball

Objectius del projecte:

- Implementar una aplicació en la plataforma Android que suposi una millor alternativa a la oficial.
- Que l'aplicació permeti la consulta de l'estat de les estacions del Bicing i la gestió de favorits agrupats depenen del lloc al qual siguin pròxim de l'usuari (feina, casa, gimnàs...).
- Opcionalment que es mostrin en el mapa els carrils bici disponibles a la ciutat i també qualsevol millora que pugui sorgir durant el desenvolupament d'aquest treball.
- Que l'aplicació segueixi les guies d'estil de Material i que tingui uns patrons d'usabilitat esperats per a la plataforma Android.

Objectius per als usuaris:

- Crear una aplicació atractiva tant a nivell visual com a nivell de prestacions.

Objectius personals:

- Aprofundir i desenvolupar els coneixements en la plataforma Android adquirits.
- Millorar els coneixements del llenguatge de programació Kotlin, el qual s'utilitzarà per a desenvolupar l'app.
- Aprofundir els coneixements sobre conceptes com: la clean architecture, MVP (Model Vista Presenter), Injecció de dependències.
- Opcionalment afegir tests unitaris i d'instrumentació.

1.3 Enfocament i mètode seguit

1.3.1 Enfocament

Els possibles enfocaments o maneres d'afrontar el problema que s'han trobat, són:

- Trobar un projecte open source que ja implementi el cas basic i extendre'l aplicant el disseny i els patrons d'usabilitat òptims per a l'aplicació.
- Abordar la solució fent un projecte des de zero recolzant-se en llibreries de tercers allà on faci falta (com per exemple amb els mapes s'utilitzaria google maps).

Per a avaluar quina opció escollir s'ha fet una cerca a Github^[8] i l'únic repositori que aborda de manera open source el problema a resoldre és OpenBicing ^[9]. Aquest repositori porta des del 2010 sense esser actualitzat i no utilitza cap patró dels actuals per a la capa de presentació i arquitecturització del projecte, per tant s'ha desestimat utilitzar-lo com a base per al projecte.

Vist això s'ha decidit començar un nou projecte i en tot cas avaluar i agafar idees del projecte OpenBicing en cas de necessitar-ho (donant els crèdits corresponents si escau).

1.3.2 Mètode seguit

Per a desenvolupar un projecte d'aquest tipus de nou i concordant amb les pec ja definides al campus, el projecte es dividirà en tres grans blocs, un de capta de requeriments i disseny, un altre de desenvolupament i finalment un de tancament de la memòria i elaboració de la presentació del treball.

Requeriments i Disseny

En aquesta etapa es detallaran els requeriments, s'avaluarà quin seria el disseny més òptim i es generaran tots els entregables necessaris per al posterior desenvolupament de l'aplicació.

En aquest ordre primer es farà una investigació dels usuaris mitjançant una enquesta oberta i la elaboració de diferents Persones (usuaris ficticis que representarien a perfils potencials d'ús de l'aplicació), posteriorment s'escriuran i detallaran els requeriments i se'n escriuran els casos d'us per tal d'obtenir un disseny conceptual de l'aplicació.

Finalment es definirà el flux de l'aplicació i es farà un prototip a alt nivell d'aquesta.

Desenvolupament

Aquesta etapa es divideix en els dos casos d'ús principals de l'aplicació (a no ser que hi hagin canvis substancials en el procés de capta de requeriments):

Mapa del Bicing: L'aplicació mostra un mapa de Barcelona amb totes les estacions del Bicing pintades i el seu estat, a més d'una vista de detall de la mateixa (a definir a l'etapa de disseny).

Gestió de favorits: L'aplicació permet guardar estacions com a favorites associant-les a una ubicació guardada per l'usuari (com pot esser casa, feina...).

També es tindrà en compte en aquesta etapa la creació del repositori i de les classes i arquitectura base del projecte.

Elaboració de la presentació i tancament de la memòria

Finalment un cop desenvolupada l'aplicació s'acabarà de refinar la memòria (que s'haurà anat complimentant a mesura que es desenvolupava l'aplicació) i es farà el vídeo de presentació final del projecte.

1.4 Planificació del Treball

1.4.1 Recursos

Per a dur a terme el treball es necessitaran tant recursos humans com de desenvolupament del projecte.

Recursos Humans

Recurs	Hores/persona	Propòsit
Dissenyador UI/UX	30h	Desenvolupar el disseny de l'aplicació i els millors comportaments en temes d'usabilitat.
Desenvolupador d'Android	120h	Implementar el disseny creat pel dissenyador en forma d'aplicació nativa d'Android
Project manager	35h	Gestionar el projecte, escriure la documentació i finalment preparar la presentació d'aquest.

Figura 4. Taula que mostra els recursos humans necessaris per a dur a terme el projecte i la seva implicació en hores/persona.

Recursos Materials

Recurs	Propòsit
Google forms ^[10]	Utilitzat pel dissenyador per a fer l'enquesta per la recapta de requeriments.
Sketch ^[11]	Utilitzat pel dissenyador per a crear el prototip d'alt nivell.
Android Studio ^[12]	IDE oficial per al desenvolupament en la plataforma Android, utilitzat pel desenvolupador per a desenvolupar l'aplicació.
Git ^[13]	Gestor de versions utilitzat pel desenvolupador.
Github ^[8]	Gestor de repositoris online en el que es publicarà i guardarà el projecte durant el seu desenvolupament.
Office suite ^[14]	Utilitzada pel project manager per a redactar la memòria i la presentació posterior del projecte.

Figura 5. Taula que mostra els recursos materials necessaris per a dur a terme el projecte.

1.4.2 Planificació

Per a dur a terme el projecte s'ha dividit en varies entregues que coincideixen amb les diferents entregues del Campus Virtual, cada entrega està composta per un conjunt de tasques i la seva dedicació en hores:

PEC1 - Pla de treball	10h
Contextualització i estat de l'art	4h
Descripció del projecte i definició d'objectius	2h
Enfocament i metodologia a seguir	2h
Planificació del treball	2h
PEC2 - Anàlisi i Disseny	30h
Investigació d'usuaris i recolliment de requeriments	7h
Disseny conceptual	7h
Prototipat	10h
Avaluació	6h
PEC3 - Desenvolupament	120h
Creació del repositori	2h
Clases i arquitectura base	8h
Feature: Mapa del Bicing	60h
Feature: Gestió de favorits	30h
Altres millores i refinament	20h
PEC4 - Tancament del projecte	35h
Documentació	20h
Presentació i vídeo	15h
TOTAL	195h

Figura 6. Taula que mostra les fites i tasques del projecte, desglossades en les hores planificades per a dur-les a terme.

El projecte comença el 21 de febrer del 2018 i finalitza el 6 de juny de 2018.

La dedicació del projecte serà d'1h 30m els dies laborables i 3h els festius, essent així assolible les 195h de desenvolupament respectant les dates del projecte.

Finalment s'adjunta també el diagrama de Gantt:

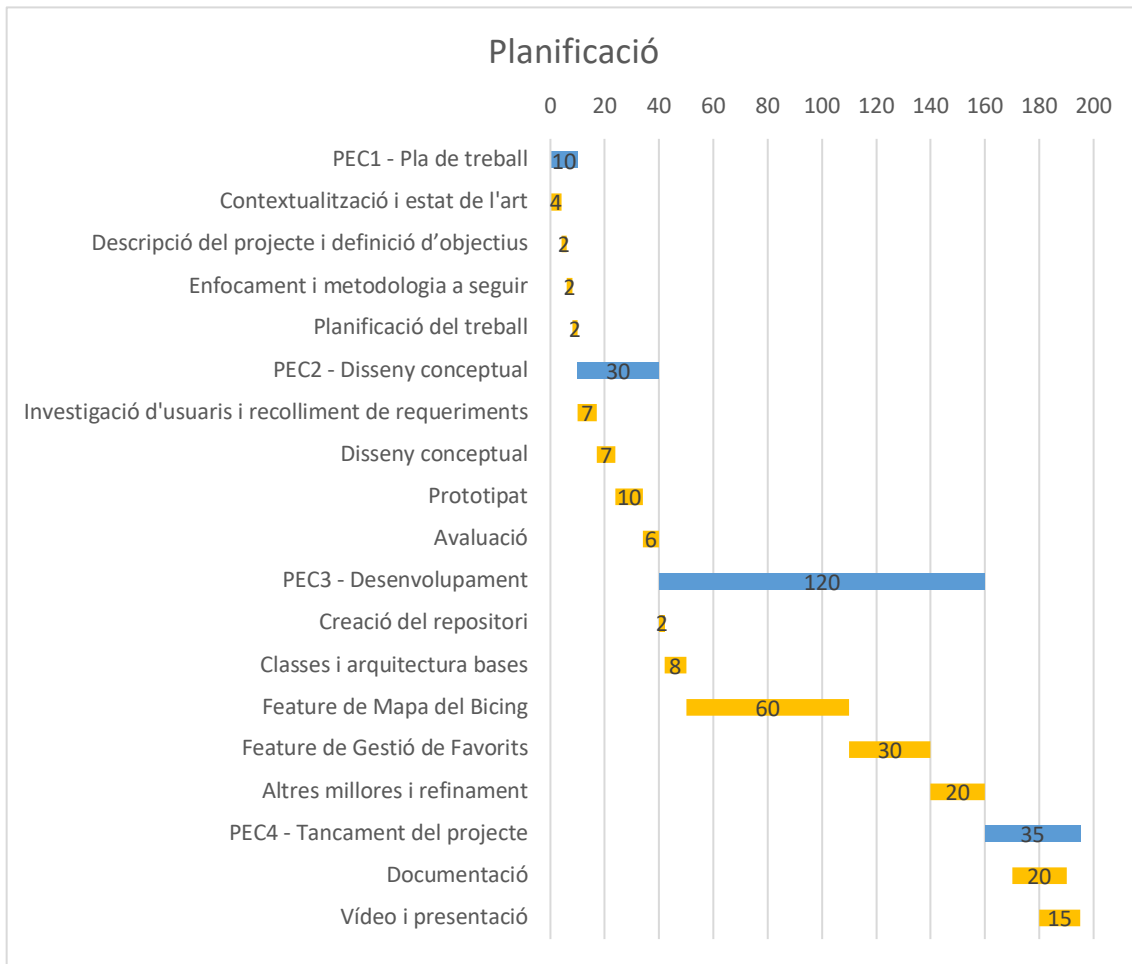


Figura 7. Diagrama de Gantt de les fites i les tasques amb les seves dependències en el temps.

1.5 Breu sumari de productes obtinguts

Els productes obtinguts d'aquest projecte seran:

- Estudi i valoració de l'enquesta de recollida de requeriments.
- Prototip a alt nivell de l'aplicació
- Codi font de l'aplicació.
- Aplicació instal·lable en format .apk.
- Memòria que documenta el projecte.
- Vídeo de presentació final.

1.6 Breu descripció dels altres capítols de la memòria

En la resta de capítols trobarem l'anàlisi de l'estat de l'art i de la capta de requeriments, on s'ha fet un reconeixement de l'aplicació oficial del Bicing, així com de les alternatives més descarregades i es presenten els resultats d'un estudi fet mitjançant una enquesta a usuaris potencials de l'aplicació. Finalment en aquest apartat es presenta la capta de requisits extrets tant de l'estat de l'art com de l'estudi.

A continuació veurem la etapa de disseny, en la qual es mostren els casos d'ús i la conceptualització i el disseny d'aquesta, realitzat amb un prototip a alt nivell i la justificació del disseny pantalla a pantalla.

Posteriorment es detallarà l'arquitectura a nivell tècnic que tindrà el projecte, veure'm l'explicació de cada una de les capes que contindrà l'arquitectura i també com modelarem les dades a base de dades i l'estructura de les dades que consumirem d'api.

A l'etapa de desenvolupament trobarem una explicació de les eines i llibreries utilitzades per a desenvolupar l'aplicació, així com la justificació i explicació de l'estructura del codi, anant mòdul per mòdul i explicant el perquè de la modularització de l'aplicació. Finalment en aquesta etapa també trobem les proves realitzades per a certificar el correcte funcionament de l'aplicació.

Per acabar trobarem un manual d'ús de l'aplicació en el que s'explica el funcionament d'aquesta i les conclusions del treball.

2. Anàlisi

2.1 Estat de l'art

Com ja hem comentat en l'apartat anterior hi ha diverses apps que ja "cobreixen" la necessitat que es pretén cobrir amb aquesta. En aquest apartat es pretén mirar detalladament a cadascuna de les alternatives i veure'n les mancances o friccions que justifiquen el desenvolupament d'una nova aplicació que intenti cobrir les necessitats i els casos d'ús que l'usuari espera amb la menor fricció possible.

Aplicació del Bicing

Aquesta és l'aplicació oficial del Bicing i és la primera alternativa i la més utilitzada que hi ha a la PlayStore^[15].

Característiques principals:

- Mapa amb totes les estacions i el seu estat, amb la possibilitat de filtrar per tipus d'estació (elèctrica o mecànica).
- Llistat de favorits.
- Possibilitat de cercar rutes entre dos punts, donant indicacions de carril bici i tenint en compte la situació de les estacions.
- Llistat amb totes les estacions, també filtrable segons el tipus d'estació.
- Secció d'ajuda en la que s'indica el numero d'atenció a incidències del servei.
- Possibilitat un cop logat de comprovar el teu ancoratge i participar en un joc per a acumular punts bescanviables per diferents premis.

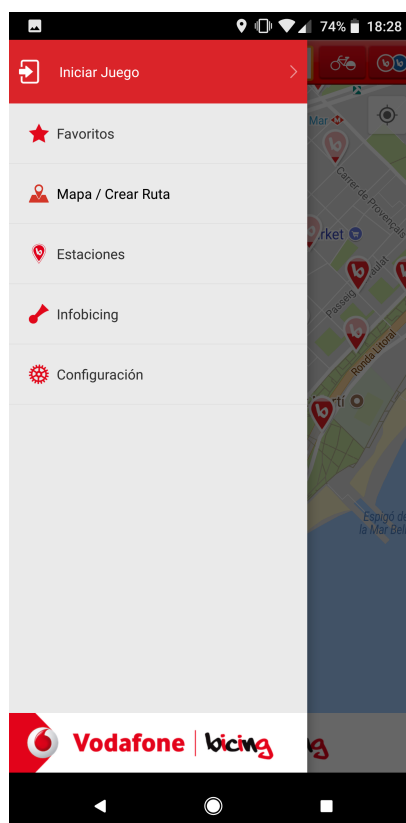


Figura 8. Menú principal de l'aplicació oficial del Bicing.

Principals problemes:

- L'aplicació només dona importància al mapa i deixa totes les seves possibilitats en una AppBar sobrecarregada en la què es difícil distingir i entendre les possibilitats que ofereix.



Figura 9. Vista detall d'una estació de l'aplicació oficial del Bicing.

- La pantalla de detall mostra molta informació com si fos una taula de dades.

- El sistema de favorits és poc intuïtiu, per a afegir un favorit s'ha d'anar a la vista detall i seleccionar un checkbox.

- Alguns fluxos són confusos, com el de veure una estació en el mapa (des de la pantalla de detall d'aquesta), en aquest cas es porta a l'usuari a una pantalla idèntica a la principal (entenent doncs que se l'ha tornat a la pantalla anterior amb la estació seleccionada) la qual no permet desplegar el menú lateral, a part es poden seleccionar altres estacions i anar acumulant vistes de detall i mapa de manera cíclica (fet que pot augmentar considerablement la memòria utilitzada per l'aplicació ja que el mapa és un element pesat i en conseqüència l'app pot esser tancada per el sistema operatiu).

-El disseny en general és molt antic i no segueix cap dels comportaments habituals d'un usuari d'android com s'especifica a la guia de Material Design que proporciona

google.

- A nivell de seguretat l'aplicació és insegura, un simple Proxy permet veure que a l'hora de logarse el password i el nom d'usuari es passen com a part de la url junt amb el token_id i el client_secret (claus crítiques de l'aplicació), a part de totes les trucades.

Exemple:

https://pubsbapi.smartbike.com/oauth/v2/token?client_id=15_3oseu6woc7ggokk4g0oscoc00sco4ocgos0440ows8os00swoc&client_secret=16l8ngutaaaooocs4wo4kswgw0owo0gw04o4o0ksoko8kkcw0o4&username=12345678X&password=XXXXXXXX&grant_type=password

Altres aplicacions – CityBiker

A part de l'aplicació oficial de Bicing, aquesta es presenta com una de les alternatives més descarregades (5000-1000 descarregues) i millor valorades amb 4,6 estrelles .

Característiques principals:

- Té una millor experiència d'usuari i segueix les directrius de Material design de Google.
- Mapa amb totes les estacions de bicing i el seu estat.
- Permet filtrar per Bicing elèctric, mecànic o ambdós.
- Té un accés directe al telèfon d'incidències del servei.

Principals problemes:

- L'aplicació és molt simple en quan a funcionalitats, no té gestió de favorits, ni permet veure molta informació sobre les estacions.

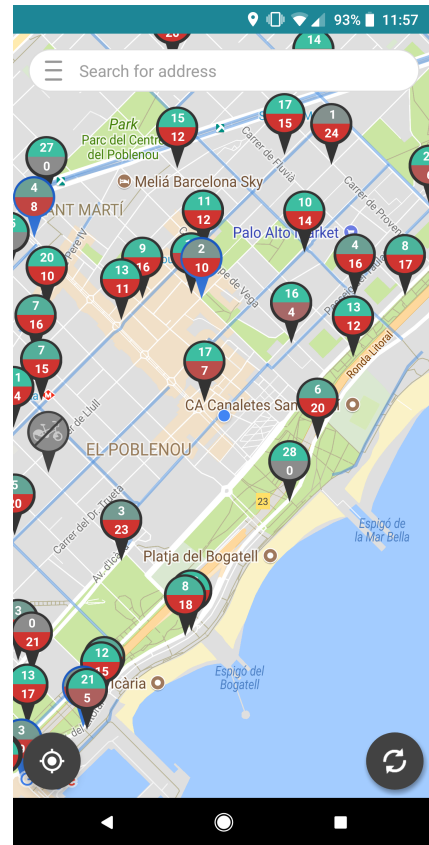


Figura 10. Pantalla principal de l'aplicació CityBiker.

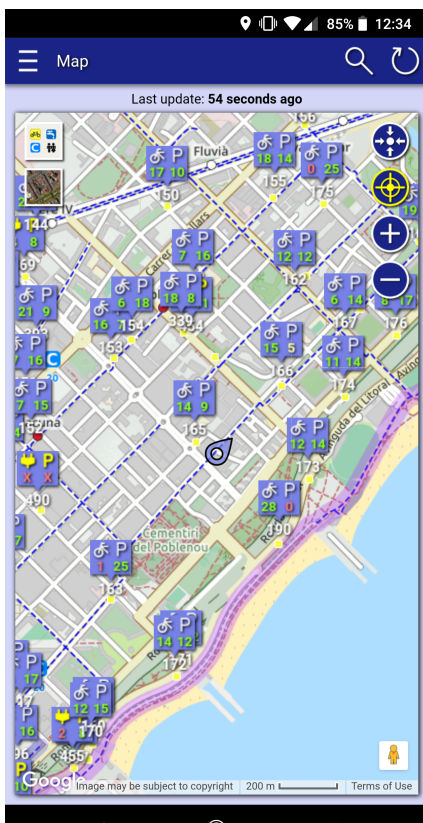


Figura 11. Pantalla principal de l'aplicació Bicing Barcelona.

Altres aplicacions – Bicing Barcelona

Aquesta aplicació amb el mateix nom que la oficial, té unes mil descarregues i una puntuació de 3,4 estrelles.

Característiques principals:

- Mapa amb totes les estacions del Bicing i el seu estat.
- Gestió de favorits amb llistes en funció de la localització (Casa, treball...).
- Llistat amb totes les estacions ordenat per proximitat.

En aquest cas el principal problema és el disseny, i la usabilitat, que com es pot apreciar no és molt atractiva a l'usuari.

2.2 Estudi de requisits

Per tal de tindre un context més acurat sobre com fan servir els usuaris l'aplicació de Bicing i d'allà extreure'n els requisits prioritaris del projecte es va realitzar una enquesta a un total de 32 persones. En l'annex A s'adjunta el model de l'enquesta i les respostes obtingudes.

2.2.1 Avaluació dels resultats de l'enquesta

Amb els resultats obtinguts s'ha pogut reafirmar que el principal requisit d'aquest treball és el mapa amb totes les estacions i el seu estat, ja que és el principal ús que se li dona a la aplicació.

També s'ha pogut extrapolar que els usuaris del Bicing utilitzen molt el servei per a transports entre les mateixes ubicacions (casa, feina, gimnàs...) i que no sabien que l'aplicació del Bicing tenia suport per a favorits (degut als problemes d'usabilitat que té), per tant aquest seria també un requisit important que s'intentarà potenciar i visibilitzar més per tal d'oferir més possibilitats a l'aplicació.

D'altra banda també s'ha constatat que la majoria d'usuaris no està content amb l'aplicació del bicing ja que ha rebut una puntuació mitja de 2,2 sobre 5, fet que junt amb el 3,2 sobre 5 de l'aplicació a Google play justifica encara més la necessitat d'una renovació i millora de la solució actual.

Finalment també s'ha extret que seria interessant tindre a part de les estacions ubicades en el mapa també un llistat ordenat per la proximitat a les estacions.

2.3 Usuaris i context d'ús

El ventall d'usuaris als que va dirigit la aplicació és una mica ampli, tenint en compte que el servei de Bicing l'utilitza gent de diferents edats. El que si que podem determinar és el seu context i que tots tenen la necessitat d'utilitzar el servei, ja sigui per a agafar una bicicleta o per a recollir-la. Podríem dir que el perfil d'usuari que utilitza el Bicing és el d'una persona mes o menys esportista, amb ganes d'utilitzar un transport alternatiu i saludable que es trobaria en majoria en el rang dels 16 als 40 anys.

En quant al context, aquest seria normalment quan vol realitzar un transport utilitzant el servei, el qual seria normalment a l'exterior i amb un ús de l'aplicació curt (ja que només voldria veure l'estació més propera amb bicicletes disponibles) i l'altre context d'ús que seria ja muntat a la bicicleta arribant al destí per a trobar la estació més propera i el seu estat, per tant també seria un context exterior i d'ús curt de l'aplicació.

2.4 Requisites

Amb la informació recaptada als apartats anteriors s'han recaptat els requisits següents per a l'aplicació.

2.4.1 Funcionals

- Mapa de l'estat del Bicing: Els usuaris han de poder consultar en un mapa les estacions més properes a la seva localització i la quantitat de bicicletes i ancoratges disponibles en aquestes.
- Llistat d'estacions pròximes: Els usuaris han de poder veure les estacions en un llistat en funció de la seva proximitat a elles, també en aquest llistat s'ha de poder veure els ancoratges i bicis dels que disposa l'estació.
- Gestió de favorits: Els usuaris han de poder guardar les seves estacions favorites en funció d'un lloc i veure'ls de manera fàcil i accessible.

2.4.2 No funcionals

- Robustesa: L'aplicació ha de ser robusta i tindrà un crashrate el més baix possible.
- Arquitectura: L'aplicació ha d'estar arquitecturitzada de manera que sigui escalable i testeable, amb mòduls el més independents possible els uns dels altres.
- Tests: La majoria de l'aplicació ha de ser testeable i contenir tests que assegurin el bon funcionament d'aquesta.
- Disseny: El disseny d'aquesta ha de seguir les directrius de Material Design el més possible.

3. Disseny

3.1 Escenaris d'ús

Tal i com s'ha comentat al punt 2.3 els principals casos d'ús de la aplicació es produiran en un context exterior i a peu o amb bicicleta (depenent de si es vol retornar o agafar una bicicleta). Amb aquest context s'han tret els següents casos d'ús.

3.1.1 Usuari busca una estació amb bicicletes

Un usuari es troba al carrer i encén l'aplicació per a veure quina és l'estació més propera a ell (filtrant o no per les seves estacions favorites en el cas d'estar-ne a una d'elles).

3.1.2 Usuari busca una estació amb ancoratges lliures

Un usuari es troba en la bicicleta, aprop de la seva destinació i obre l'aplicació per a veure l'estació més propera a ell (filtrant o no també per estacions favorites).

En ambdos casos el cas d'ús seria similar exceptuant el context d'ús i que en ambdos es busquen informacions diferents a la pantalla d'estat. Seria el següent:

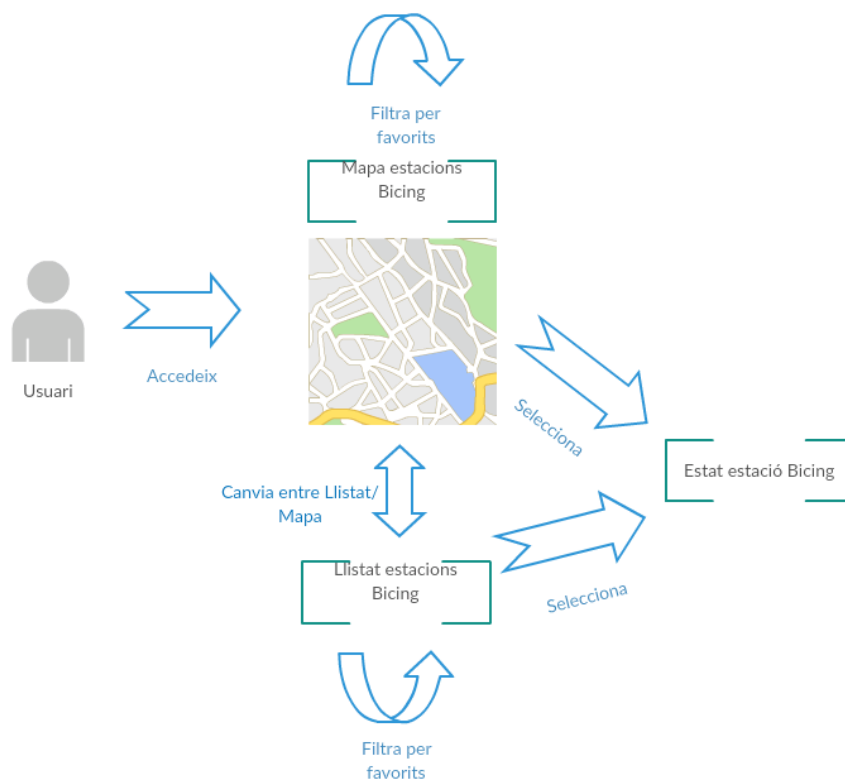


Figura 12. Diagrama de cas d'ús corresponent a quan un usuari vol saber l'estat d'una bicicleta.

3.1.3 Usuari afegeix una estació a favorits

Un usuari afegeix una estació a favorits per a poder filtrar-la més endavant, el context d'ús en aquest cas pot ser variat, ja que no és específic d'un moment o de cobrir una necessitat immediata. El diagrama seria el següent:

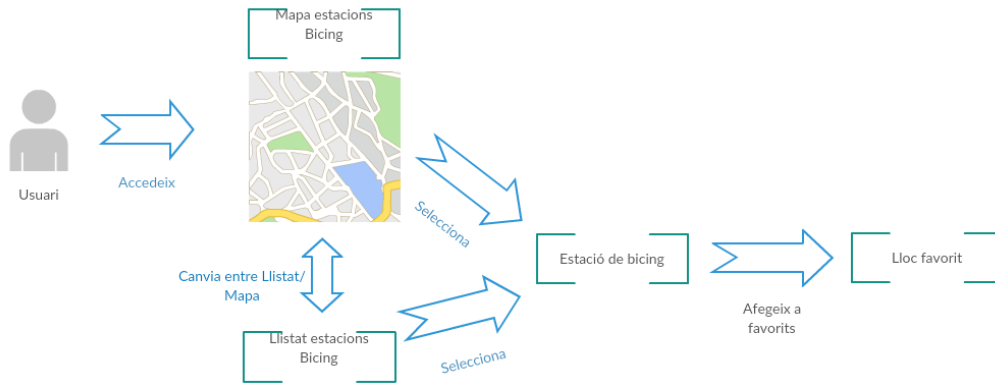


Figura 13. Diagrama de cas d'ús corresponent a quan un usuari vol afegir una estació a favorits.

3.2 Arbre de navegació

Tenint en compte els casos d'ús i la intenció de que l'aplicació sigui quan més àgil millor (en el sentit que s'arribi a l'objectiu de cada cas d'ús amb el menor número d'interaccions possibles). S'ha definit el següent arbre de navegació:

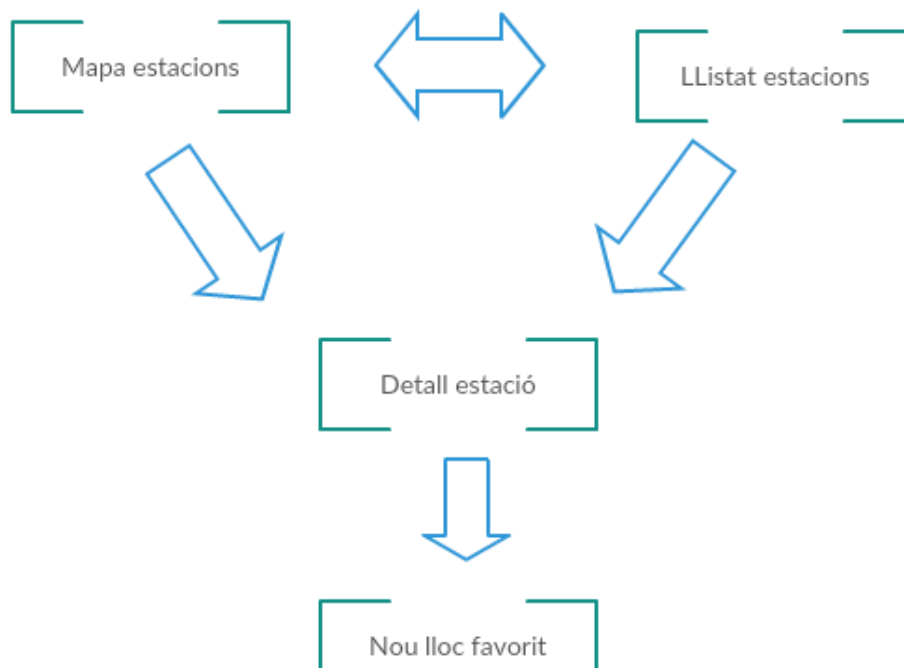


Figura 14. Arbre de navegació de l'aplicació.

3.3 Selecció del nom, el logotip i l'estil de l'aplicació

Com a fase inicial del disseny s'ha plantejat el disseny d'un logotip, el nom i de definir unes guidelines generals de l'estil d'aquesta (fonts, colors...).

3.3.2 Nom

Pel nom de l'aplicació s'ha pensat la possibilitat d'incloure parcialment el nom del servei (Bicing) ja que és una aplicació alternativa per a aquest servei, però segons les polítiques de Google Play, el nom no ha de crear confusió amb l'app original, per tant s'ha optat per el nom BCNg.

3.3.3 Logotip

S'ha optat per un logotip senzill amb una bicicleta blanca en un fons rodó amb el color vermell del Bicing.



Figura 15. Logotip de l'aplicació.

3.3.4 Estil de l'aplicació

Per a l'estil de l'aplicació s'utilitzaran els colors vermell blanc i negre, predominant el blanc i negre i deixant el vermell com a color d'accent.

La tipografia utilitzada serà la Roboto, la tipografia per defecte de Material Design.

3.4 Prototip

Tenint en compte l'estil i els fluxos definits anteriorment, s'ha dissenyat un prototip el qual es pot trobar a Invisio^[15] seguint el següent enllaç:

https://invis.io/G3GNTRJ4FWX#/288905016_Main

A continuació es troben les pantalles d'aquest prototip d'alta fidelitat junt amb l'explicació d'aquestes:

Pantalla Principal

Aquesta pantalla (la única com a tal en tota la aplicació) té dues vistes, una en forma de mapa i una altra en forma de llista.

Per intercanviar entre la vista de mapa i la de llista s'utilitza mitjançant el botó situat a la part dreta de l'app bar. S'ha optat per afegir-lo allà ja que d'aquesta manera està accessible i en un lloc destacat, també així es deixa més espai pel contingut a la resta de la pantalla.

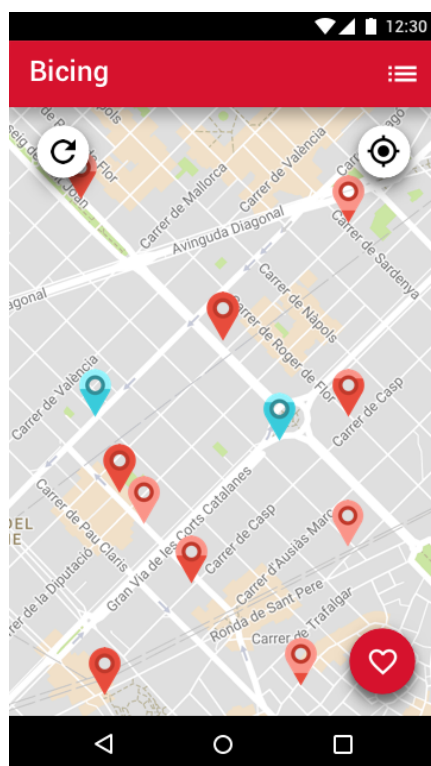


Figura 16. Pantalla principal (vista mapa) del prototip.

Mapa

En la vista de mapa es dona prioritat a la situació geogràfica de les estacions, representades per pins, els quals ja donen un indicatiu de l'estat de bicicletes i ancoratges de les estacions. En aquesta vista es pot clicar a cadascun dels pins per a veure'n l'estat més detallat de cadascuna de les estacions, que vindria a ser la informació que es mostra de cadascuna d'elles a la llista. També conta amb un botó de refresc de la informació i un altre que centra la vista en la ubicació actual de l'usuari, ambdós a la part d'a dalt del mapa.

Finalment la pantalla conta amb un botó de favorits a la cantonada inferior dreta.

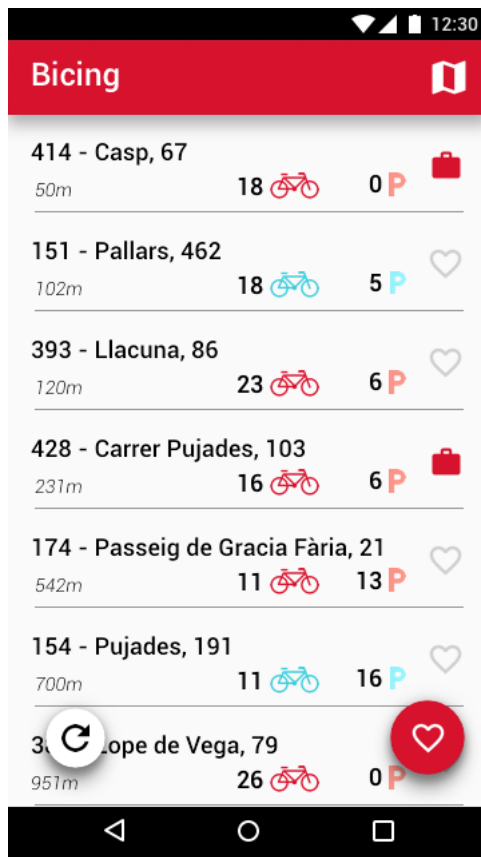


Figura 17. Pantalla principal (vista de llistat) del prototip.

Llistat

El llistat d'estacions mostra les estacions ordenades per proximitat amb la informació rellevant sobre aquesta, les quals son:

- El nom de l'estació: Ressaltat en negreta i amb una tipografia gran es mostra a la part superior esquerra, podent ocupar tota la amplada de la pantalla en cas de noms llargs.
- Distància a l'estació: Amb cursiva i sota el nom mostra la distància en metres/quilometres entre l'usuari i l'estació.
- Bicycletes i ancoratges disponibles: Aquesta informació es mostra de manera numèrica amb una icona d'una bicicleta per a les bicicletes disponibles i amb una P (de pàrquing) per als ancoratges disponibles.
- El tipus d'estació: Aquest es mostra implícit en el color dels indicatius de bicicletes/ancoratges disponibles a l'estació, s'utilitza una tonalitat de vermells per a les mecàniques i una tonalitat de blaus per a les elèctriques.
- Favorits: Finalment es mostra si l'estació forma part o no d'una llista de preferits. També clicant es pot afegir a una llista de favorits o treure-la de favorits si ja en forma part d'una.

També afegir que els botons s'amagaran al scrollejar cap a baix i es tornaran a mostrar al scrollejar cap a dalt. Aquesta decisió es fa per a una millor usabilitat, ja que al scrollejar cap a baix l'usuari busca nova informació que es veurà

primer per la part d'a baix de la pantalla, i en aquest cas els botons dificulten la tasca.

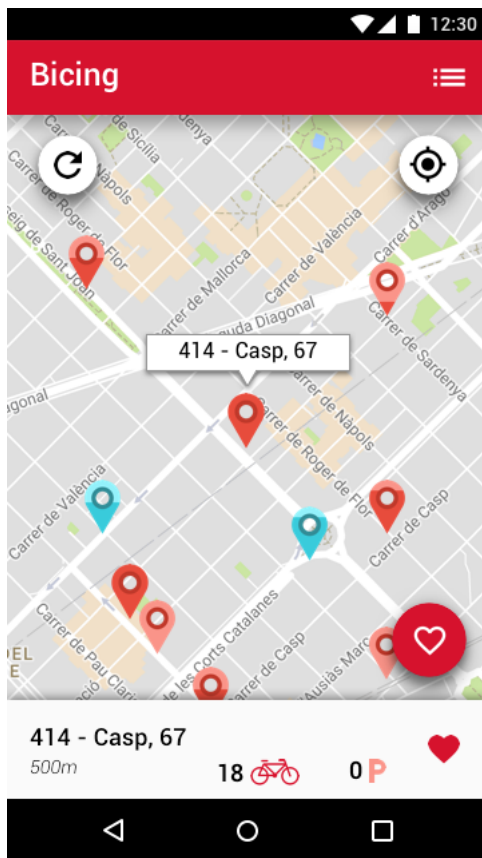


Figura 18. Pantalla principal (vista de mapa) amb una estació seleccionada.

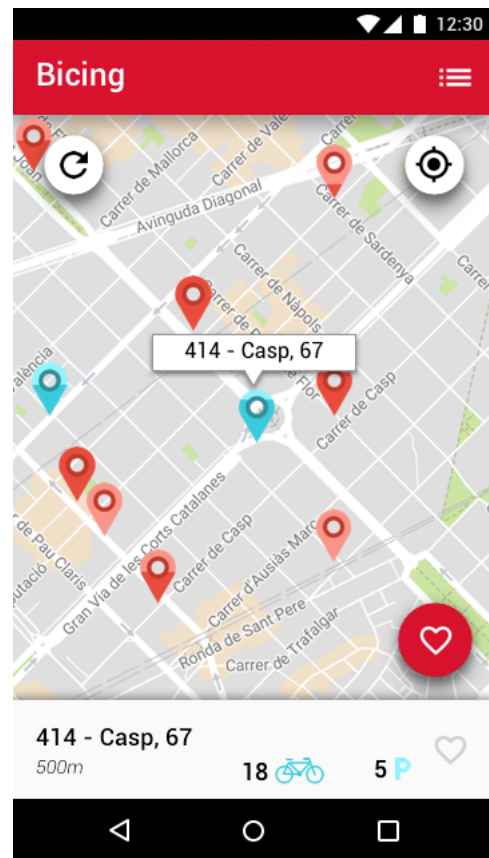


Figura 19. Pantalla principal (vista de mapa) amb una estació elèctrica seleccionada.

Detall Mapa

Com hem comentat abans, a la vista de mapa es pot clicar als pins que representen les estacions per a veure una vista de detall d'aquesta. La informació i accions disponibles d'aquesta vista són les mateixes que les dels elements del llistat, es mostra la informació de cada parada i es pot afegir aquesta a alguna llista de favorits.

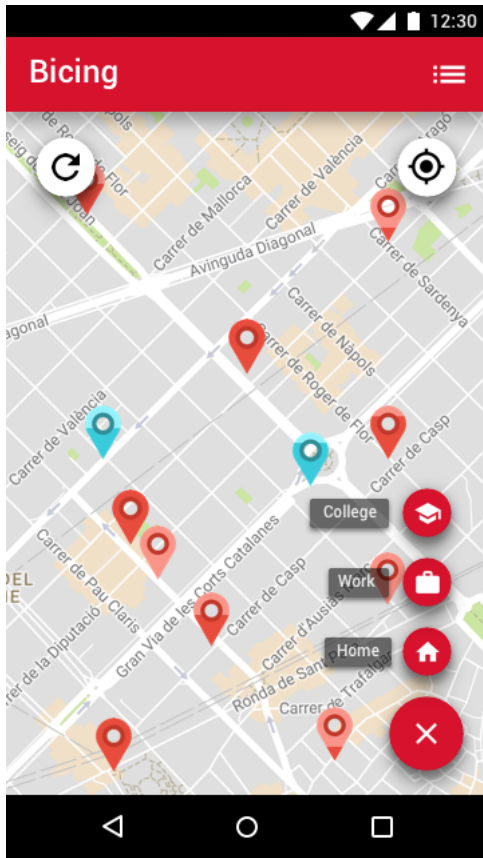


Figura 21. Pantalla de selecció de llista favorita (vista mapa).

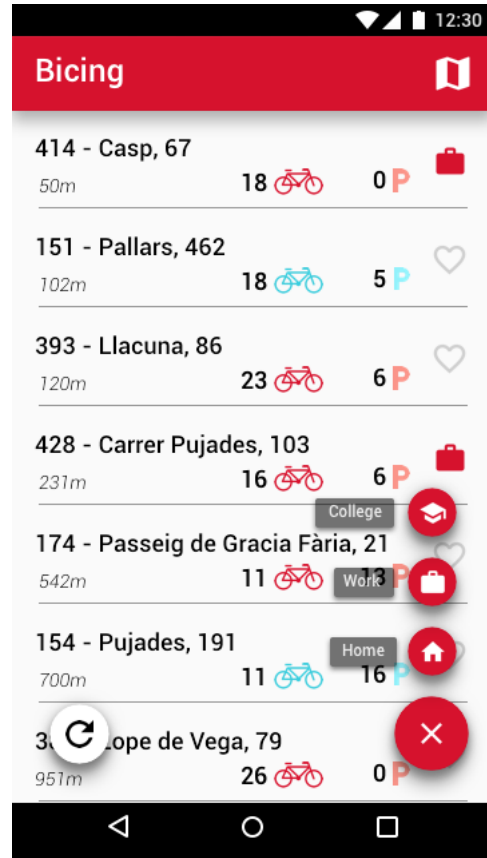


Figura 20. Pantalla de selecció de llista favorita (vista llistat).

Menú Favorits

Els favorits es poden gestionar des d'ambdues pantalles (llistat i mapa) clicant al botó vermell amb un cor situat a la cantonada inferior esquerra. Com es mostra a les figures 20 i 21, un cop clicat, es desplega un menú en vertical en el qual es pot seleccionar una de les ubicacions preferides, o clicant al botó inferior treure aquest menú.

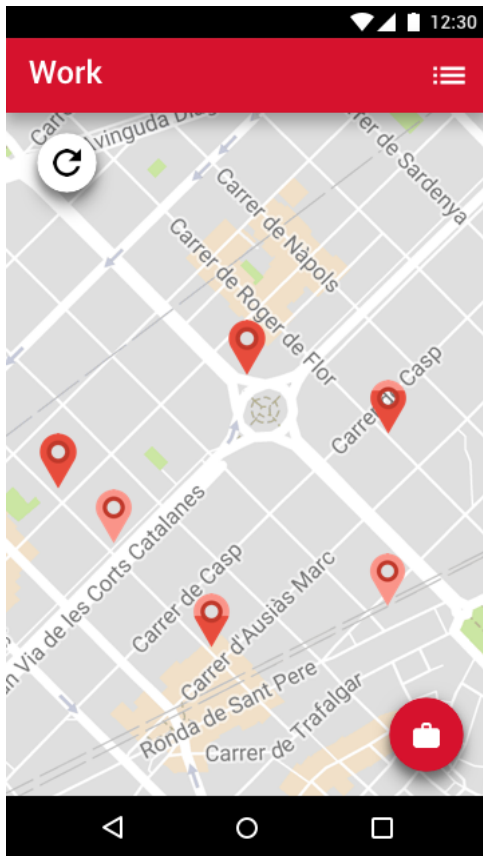


Figura 22. Vista lloc favorit mapa.

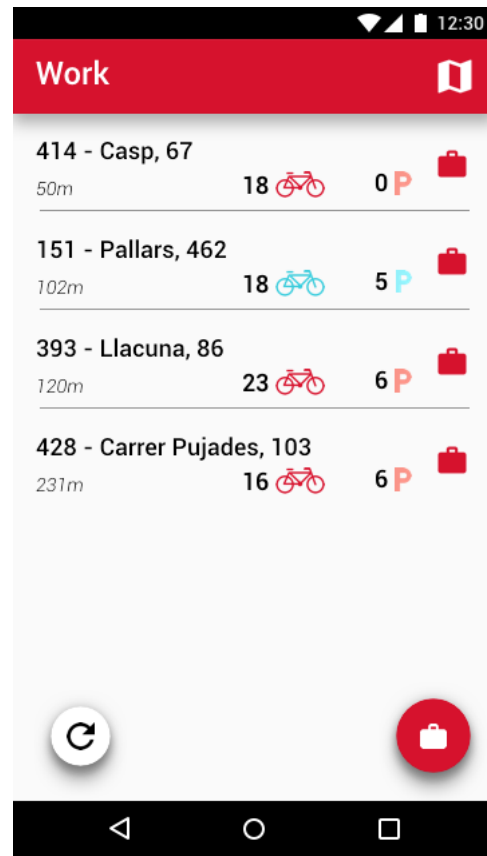


Figura 23. Vista lloc favorit llistat.

Vista favorits

Un cop seleccionada la llista (que normalment correspon a una ubicació) es mostren només les estacions pertanyents a aquella llista, en el mapa la vista es centra al mitg de tots els punts, les interaccions són les mateixes que a la pantalla sense filtrar.

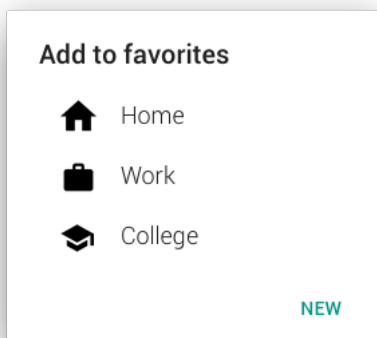


Figura 24. Popup selecció favorit.

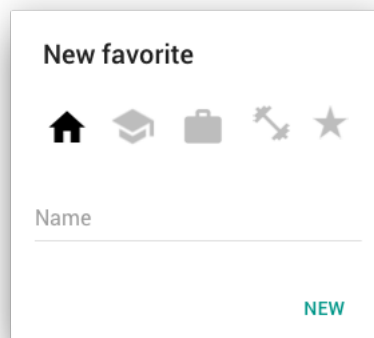


Figura 25. Popup creació favorit.

Afegir estació a favorits

Si una estació no forma part de cap llista de favorits i es clica a la icona del cor que té cada element del llistat o en la vista de mapa un cop s'ha seleccionat l'estació, aleshores es mostra un Popup amb els llocs favorits existents i un cop seleccionat un, l'estació en qüestió passa a formar part d'aquell lloc. Si l'estació que volem afegir a favorits no pertany a cap dels llocs existents, mitjançant el botó NEW se'n pot crear un de nou, clicant-lo s'obre un segon Popup que permet crear un nou lloc preferit.

4. Arquitectura

4.1 Arquitectura de l'aplicació

En quant a l'arquitectura l'estructura de l'aplicació es basarà seguint la proposta de clean architecture que es pot trobar a la següent presentació:

<https://speakerdeck.com/igorwojda/why-do-we-need-clean-architecture>.

D'aquesta manera es dividirà l'aplicació en 3 capes diferents: La capa de presentació, la de domini i la de dades.

4.1.1 Capa de presentació (Presentation)

Aquesta capa és la capa que contindrà totes les classes necessàries per al funcionament de la interfície visual de l'aplicació, en el cas específic d'Android contindrà les activities, fragments, adapters, views... així com les classes que gestionen la lògica de les activities/fragments, bàsicament presenters.

El patró de disseny que es seguirà per a aquesta capa serà el MVP (Model View Presenter), el qual separa la lògica que conté cada vista de la vista en sí.

D'aquesta manera per a cada pantalla (Activity o fragment) es crearà primer una interfície View la qual definirà quines són les possibilitats de la vista i que necessita per a dur-ho a terme, de la mateixa manera es definirà una interfície per al Presenter que servirà a la vegada de contracte també. D'aquesta manera tant el presenter com l'activity dependran de les interfícies i no de les classes concretes que tindran la lògica.

En el MVP la View correspon a la classe activity o fragment, i aquesta només conté la part de codi corresponent a la visualització i captació d'interacció amb l'usuari, essent completament dependent d'un Presenter, el qual no té el coneixement de la lògica de negoci de la vista: a on ha d'anar per a treure la informació que la view ha de mostrar, com reaccionar a les interaccions que fa l'usuari amb la vista, etc.

Aquesta capa a nivell de projecte es separarà en dos mòduls dintre del projecte per tal d'aïllar el més possible el framework d'android i assegurar encara més que els presenters no tenen coneixement de les classes concretes d'aquest framework.

4.1.2 Capa de domini (Domain)

Aquesta capa, també en un mòdul independent del framework d'Android contindrà la lògica de negoci, sabrà com filtrar i tractar les dades que li arriben de la capa de dades per a satisfer un cas d'ús que necessita la vista. Així en aquest mòdul trobarem UseCases els quals satisfaran un cas d'ús, com per exemple retreure les estacions corresponents a un lloc favorit. Els cas d'ús son utilitzats pels Presenters i aquests normalment fan ús dels repositoris de la capa de dades per a retreure la informació i després manipular-la (o modificar-la).

En aquesta capa també es defineixen les interfícies anomenades Repository que són les que s'implementaran a la capa de dades per tal de donar als UseCases la possibilitat d'accedir i modificar les dades de les fonts de dades.

4.1.3 Capa de Dades (Data)

Aquesta capa conté totes les classes necessàries per a interactuar amb les diferents fonts d'informació de les que pugui disposar la nostra aplicació (en el nostre cas base de dades i api), també es presentarà en un mòdul a part.

Les classes DataSource implementen les interfícies Repository definides a la capa de domini i permeten accedir a les diferents fonts d'informació (api, base de dades, Shared preferences...).

Quan hi ha més d'una DataSource, com és el cas d'aquest projecte, s'afegeix una capa més d'abstracció entre els UseCases i els DataSource, que seria una implementació del Repository, aquesta classe implementa en aquest cas la interfície de la capa de domini i fa ús de cada DataSource en funció de la necessitat (s'encarrega de discernir a quina DataSource o quina política es fa servir entre DataSource).

També cal esmentar que cada capa tindrà la seva pròpia classe model (classe entitat que conté un conjunt de dades) la qual estarà adaptada a les necessitats de cada capa. D'aquesta manera els models de la capa de data vindran condicionats per la estructura de l'api i de la base de dades, el de domini per la lògica de negoci i el de presentació per com es mostraran les dades a la interfície d'usuari.

Per a poder passar les dades entre capes es crearan uns mapers els quals mapejaran els models d'una capa a una altra.

Finalment tot es juntarà amb una llibreria d'injecció d'independències anomenada Dagger 2^[16].

A l'Annex B es pot trobar el diagrama de l'estructura esmentada.

4.2 Estructura de les dades

Api del Bicing

De l'api del Bicing només s'utilitzarà l'endpoint públic que proporciona l'ajuntament, el qual retorna l'estat de totes les estacions, està en format JSON i no es guardaran les dades de manera persistent.

L'endpoint de l'api és el següent: <http://wservice.viabicing.cat/v2/stations>

l'estructura d'aquest és:

```
{
  stations:
  [
    • {
      id: "1",
      type: "BIKE",
      latitude: "41.397952",
      longitude: "2.180042",
      streetName: "Gran Via Corts Catalanes",
      streetNumber: "760",
      altitude: "21",
      slots: "10",
      bikes: "18",
      nearbyStations: "24, 369, 387, 426",
      status: "OPN"
    },...],
  updateTime: 1522843148
}
```

Per tant només es crearà una classe model d'aquest que contindrà les dades del JSON de cada estació.

Base de dades

Per a la gestió de favorits es crearà una taula de dades relacional mitjançant un ORM com ara SugarORM^[17] o Room^[18].

Aquesta base de dades de SQLite contindrà una Taula de llista de favorits, la qual contindrà una primary key id (INT), un nom (TEXT) i un id de la icona que el representa (INT). També per a fer la relació Favorite place – Estació la base de dades contindrà una altra taula amb la id de favorits i la id de l'estació. Els noms de les taules i les relacions entre elles son fetes per la llibreria, per tant no s'especifiquen.

5. Desenvolupament

5.1 Introducció

Per a desenvolupar la solució dissenyada s'ha optat per a fer un desenvolupament natiu en la plataforma Android, únicament per a dispositius mòbils i què tinguin una versió superior a la SDK 19 (Android Kitkat).

S'ha optat per posar com a mínim la 19 ja que com es pot observar en la següent taula les versions inferiors a la 19 suposen un 5% del total de dispositius actius. També apuntat a versions del SDK superiors ens permet utilitzar funcionalitats més noves del sdk.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.5%
4.2.x		17	2.2%
4.3		18	0.6%
4.4	KitKat	19	10.3%
5.0	Lollipop	21	4.8%
5.1		22	17.6%
6.0	Marshmallow	23	25.5%
7.0	Nougat	24	22.9%
7.1		25	8.2%
8.0	Oreo	26	4.9%
8.1		27	0.8%

Figura 26. Taula de distribució en percentatge de versions d'Android

També com a apunt interessant en comptes de java per al desenvolupament, s'ha utilitzat el llenguatge de programació Kotlin, el qual es suporta oficialment des de la Google I/O del 2017^[19] per Google com a llenguatge de desenvolupament en Android. S'ha optat per utilitzar aquest llenguatge ja que incorpora moltes avantatges, les principals son:



Figura 27. Logotip del llenguatge de programació Kotlin

- “Null-safety”: Kotlin és un llenguatge més declaratiu en quant a la nul·litat dels objectes. Per a que un objecte pugui ser null se l'ha de declarar com a opcional, fent així que aquests objectes que no siguin opcionals mai podran esser nulls, fet que a java no és possible ja que qualsevol objecte pot ser null (i s'ha de comprovar abans d'accedir-hi per a evitar un NullPointerException). Un exemple, en kotlin podríem declarar un String de la següent manera: `var text: String, aquesta String tal i com està declarada mai podrà contenir null, per a poder contenir null se l'hauria de declarar del tipus String?`
- Extensions: Les extensions com el seu nom indica permeten d'alguna manera estendre una classe per a afegir-li més funcions, d'aquesta manera es poden crear funcions per a estendre classes com per exemple la classe View d'android per a afegir-li funcionalitats noves (sense permetre accedir a mètodes privats/protegits). Per sota aquestes extensions es compilen com a funcions estàtiques en les que el primer paràmetre és la classe que s'està extenent, però a nivell visual permet fer funcions més expressives, per exemple en el projecte s'han creat diverses d'aquestes funcions extenent la classe view per a gestionar la visibilitat d'aquestes, d'aquesta manera s'ha creat una funció `fun View.visible()` que fa que la visibilitat de la vista passi a ser visible, essent més net cridar a `view.visible()` que `view.setVisibility(View.VISIBLE)`.
- Interoperabilitat amb java: Encara que el projecte està codificat completament en Kotlin, aquest també pot coexistir en un projecte amb codi en java, podent-se cridar a funcions/classes de l'un a l'altra i a l'inrevés.

5.2 Eines utilitzades

5.2.1 Android Studio

Per al desenvolupament s'ha utilitzat l'IDE oficial per a desenvolupament en android, Android Studio en la seva versió 3.0.1.



Figura 28. Logotip d'Android Studio

5.2.2 Llibreries utilitzades

S'ha optat per utilitzar diverses llibreries que solucionen moltes de les tasques comunes, necessàries per a la majoria d'aplicacions, com ara per a la injecció de dependències o la gestió de la comunicació amb apis web. Totes elles es poden trobar en el fitxer dependencies.gradle, s'ha optat per tindre-les en un mateix fitxer per a no tindre repetits en els diferents fitxers build.gradle de cada mòdul de l'aplicació.

Android Support^[20]

S'han importat diferents mòduls d'aquesta llibreria, la llibreria de suport és una llibreria creada per l'equip d'Android que atorga retrocompatibilitat i eines bastant útils per al desenvolupament en android. De retrocompatibilitat per exemple s'ha utilitzar AppCompatActivity, Design i CardView, que permeten portar elements del disseny material a versions anteriors a Android Lollipop (SDK 21) que va esser quan es va introduir.

D'eines útils hem importat RecyclerView i ConstraintLayout. RecyclerView ens permet crear ListViews de molts elements sense tindre un gran impacte en memòria ja que com el seu nom indica recicla les vistes, instanciant només la quantitat d'elements que caben en pantalla més 2 (un per a dalt i un per a baix) i així va reciclant aquests elements a mesura que es va fent scroll. En quant a ConstraintLayout és un potent ViewGroup que permet fer quasi qualsevol interfície gràfica sense necessitar anestar Layouts.

GoogleMaps^[21]



Figura 29. Logotip de GoogleMaps

És la llibreria més famosa per a la utilització de mapes, en el nostre projecte ens permet mostrar el mapa que surt a l'iniciar l'aplicació.

Dagger 2^[22]

És la llibreria més utilitzada per a la injecció de dependències. Ens permet injectar a les activitats/fragments tots els objectes que necessitin. Es basa en crear un graf de dependències mitjançant modules (les classes que proporcionen les instàncies) i components (les interfícies que “injecten” les instàncies a les diferents classes), d’aquesta manera ens estalviem de crear els objectes a cada classe i de resoldre’n les dependències d’aquests. També com que aquests Components estan lligats a les instàncies d’Application (AppComponent), Activity(ActivitySubComponent)... es poden lligar els objectes a tindre un àmbit d’aplicació (a la pràctica serien singletons, ja que només hi hauria una instància en tota la aplicació), o a àmbit d’activity o fragment.

També en aquest projecte s’ha utilitzat AndroidInjectors els quals fan més fàcils la integració de Dagger 2 amb els elements del sistema operatiu Android (Activities i fragments).

RxJava^[23]

RxJava ens permet una gestió molt més fàcil i versàtil dels diferents threads de l’aplicació. En aquest projecte s’utilitza per a les tasques que s’executen en un background thread (bàsicament crides HTTP i a base de dades). Utilitzant el patró de disseny de l’Observer, permet a un Observa

ble emetre objectes per un thread en una espècie d’sstream i a un “Subscriber” escoltar aquests objectes en un altre thread. A més incorpora eines molt potents per a la modificació, separació i unió d’aquests streams. Un exemple interessant d’aquest paradigma es troba a GetStationsUseCase, en el que s’ajunten dos streams que provenen de repositoris diferents (el de bicicletes que ve d’una crida HTTP i el de favorits que ve de base de dades) permetent-nos associar les estacions amb els favorits en una funció zip().



Figura 30. Logotip de RxJava

Retrofit & OkHttp^[24]

Retrofit és una llibreria que ens gestiona les trucades Http amb l’endpoint en el nostre cas del Bicing. OkHttp és la llibreria del client http que fa servir Retrofit per a executar les crides. També s’importa un adaptador per a fer les crides de retrofit observables de rxJava.

FloatingActionButton^[25]

Aquesta llibreria conté els botons rodons de l'aplicació, s'ha utilitzat aquesta implementació i no la oficial d'android continguda a la support library perquè aquesta ens permet tindre Floating action button menus, o en altres paraules, botons que despleguen un menú de botons. En l'aplicació això s'utilitza quan cliquem al botó de favorits per a desplegar els favorits creats per l'usuari.

Room^[26]

És una llibreria afegida fa poc per Google, que fa més fàcil la implementació d'una base de dades de MySQLite (la que fa servir android internament). També s'ha importat com amb retrofit un adaptar per a poder utilitzar la llibreria amb RxJava.

5.3 Estructura del codi

5.3.1 Introducció

Com ja s'ha comentat a l'apartat d'arquitectura, l'app es dividirà en tres capes les quals en estructura de codi es traduirà en el mòdul d'app (el qual és l'aplicació) i 3 mòduls en forma de llibreria en kotlin, sense coneixement del framework d'android (amb l'excepció de data). D'aquesta manera el codi té els mòduls de app, presentation, domain i data, en els pròxims apartats s'explicarà la estructura que segueixen cadascun i quin és el seu paper a nivell tècnic dins de l'aplicació.

S'ha optat per a dividir la app en 4 mòduls per a fer una solució fàcilment escalable que conté classes amb un objectiu concret (Single responsibility), extendibles i amb una visibilitat acotada al que cada capa necessita (Open/close principle), amb classes que depenen d'interfaces i no de classes concretes (Dependency inversion principle), les quals són el més acotades possibles (Interface segregation principle) i qualsevol classe que implementi aquesta interface pot utilitzar-se sense alterar-ne el correcte funcionament (Liskov substitution principle), intentant complir així els principis SOLID^[26].

També tindre la aplicació dividida en 4 mòduls ens aporta temps de compilació més curts, ja que si no s'ha fet cap canvi en un mòdul, aquest no es recompila. Finalment també un gran avantatge és el de tindre les llibreries i el framework d'android acotats a allà on fan falta, estan completament desacoplades la resta de capes del framework d'android per exemple, o també tenint només la dependència de Room i Retrofit a la capa de data, essent les demès capes totalment agnòstiques de com s'estan obtenint aquestes dades.

A continuació anirem mòdul, començant per el més "intern" (domain), seguint pel mòdul de data i acabant pels mòduls de presentation i app.

5.3.2 Domain

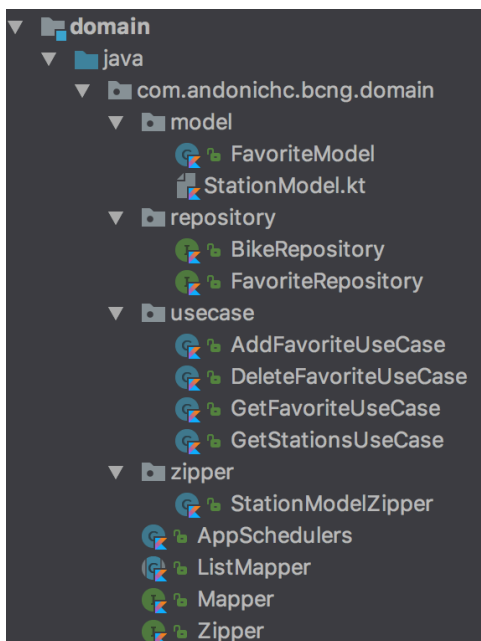


Figura 31. Estructura del mòdul domain.

Com ja s'ha comentat a l'apartat 4, domain és la capa més interna de l'aplicació, no té coneixement de cap altre mòdul o capa de l'aplicació i conté la lògica de negoci.

Aquesta lògica de negoci s'encapsula en casos d'ús (usecase) els quals seran utilitzats pels presenters a la capa de presentació per a obtenir/manipular dades amb alguna lògica o manipulació que s'escapa de la lògica de la vista.

Per exemple el GetStationsUseCase combina les dades que obté del BikeRepository (totes les estacions i el seu estat) amb les dades dels favorits de l'usuari que provenen del FavoriteRepository i d'aquesta manera la capa de presentació rep una llista

d'estacions amb la pertinença de cada estació a una llista de favorits o no. Després també tenim casos d'us per a obtenir, afegir i esborrar favorits del FavoriteRepository.

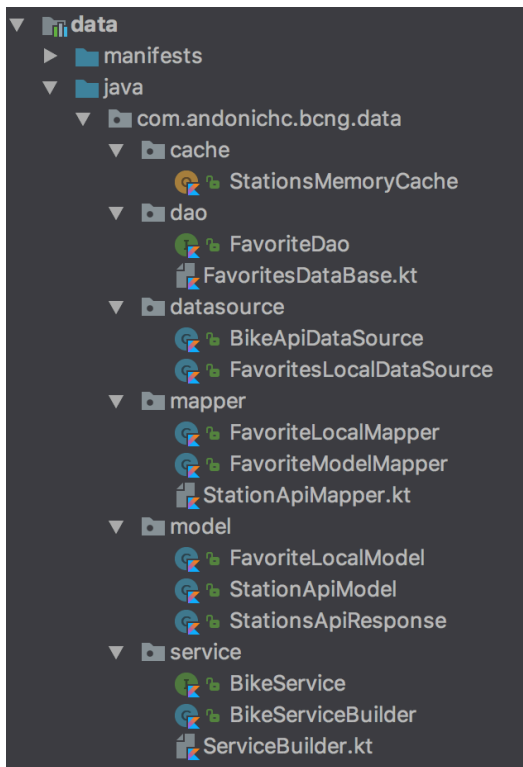
Com hem esmentat a l'exemple anterior els casos d'ús treuen les dades dels Repositoris (repository), els quals com es pot apreciar en aquesta capa no són més que interfaccs. En aquesta capa només es defineixen els contractes que hauran de seguir els repositoris per a funcionar amb els casos d'ús, és a data on es troben les implementacions concretes d'aquests repositoris.

També podem observar que té els seus propis models, cada capa té la seva pròpia classe model i aquesta es mapeja entre capa i capa, això es veurà més clar a la capa de data, el problema de tindre un model per a tota la aplicació ens obliga a lligar-lo a coses com per exemple la llibreria de peticions Http que utilitzem, o la api que estiguem consumint. Tenint una classe model per a cada capa ens permet acotar aquestes dependències només allà a on es faci falta.

Finalment podem observar que tenim les interfaces de Mapper i Zipper i un zipper que hem utilitzat a GetStationsUseCase per a unir els inputs dels dos repositoris.

Finalment això ja és més propi de RxJava, hem creat una classe que conté els diferents Schedulers de RxJava, un Scheduler és la classe que té RxJava per a especificar els threads, tenint un Scheduler per al Main Thread i un altre per a IO (input output) per exemple.

5.3.3 Data



El mòdul de data, és l'encarregat d'obtenir la informació necessària pels casos d'ús, ja sigui de Base de dades o de serveis externs. Aquesta capa té visibilitat només de domain, ja que ha de ser capaç de satisfer el contracte (interfície) dels repositoris. Com a apunt interessant aquesta és una llibreria que utilitza el framework d'android ja que el framework d'android és necessari per a la creació i gestió d'una base de dades local.

Les Fonts de dades (datasources) son les classes encarregades de implementar la interface definida a domain per a proveir les dades que necessiten els usecase. En aquest projecte tenim dues, FavoritesLocalDataSource que accedeix a la base de dades interna de favorits, i BikeApiDataSource que consumeix un endpoint Http.

Podem apreciar també que aquesta capa té els seus models propis, cada service o database retorna aquest models i a datasource fent ús dels mappers es mapejen aquests models a models de domain per tal de complir el contracte del repositori.

Figura 32. Estructura del mòdul data.

Finalment també tenim les classes de cadascuna de les llibreries d'accés a Base de dades local (contingut a dao), a l'Endpoint del bicin (contingut a service) i un object (què és bastant semblant a un singleton però ja ho prové el llenguatge en aquest cas) anomenat StationsMemoryCache que ens permet tindre guardar-nos en memòria el llistat de bicicletes per no demanar-lo al servei de Bicing cada cop que es necessita aquesta dada.

5.3.4 Presentation

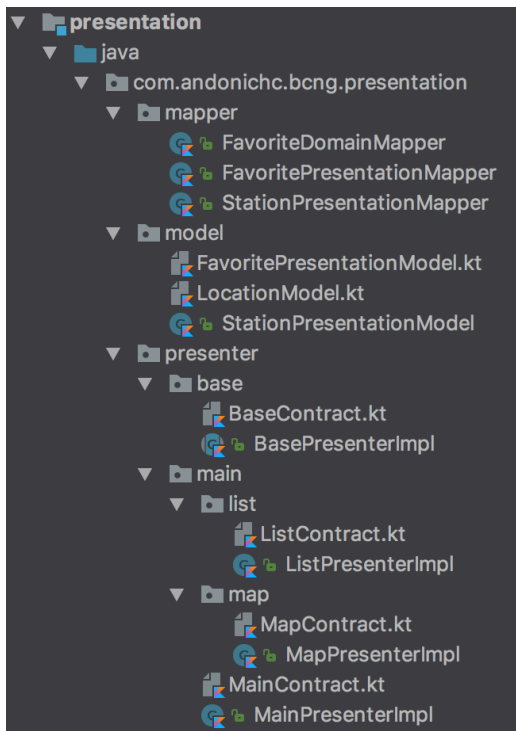


Figura 33. Estructura del mòdul Presentation.

El mòdul de presentation és l'encarregat de gestionar la lògica de presentació (lògica de les vistes). Per a això seguim el patró de disseny MVP (Model View Presenter) explicat ja a l'apartat d'arquitectura.

En aquest mòdul es troben tant els models com els contractes o interfaces de les Views i Presenters (especificats a les files Contracts) i la implementació dels presenters. La implementació de les views es troba a app ja que corresponen a classes pròpies del framework d'android (Activities i Fragments).

També podem trobar l'existència d'un contracte base que conté les interfaces i classes de les que extendran tots els demès presenters i views. Aquestes classes base especifiquen els mètodes que totes les vistes han d'implementar així com la lògica bàsica de presentació que

es podrà trobar a cada presenter.

Finalment també trobem els mappers ja que els presenters seran els encarregats de consumir el usecase amb els models de domain i mapejarlos a models de presentació, adaptats al que necessita mostrar la vista (el nom de l'estació formatejat ja de certa manera, l'estat de l'estació...).

5.3.5 App

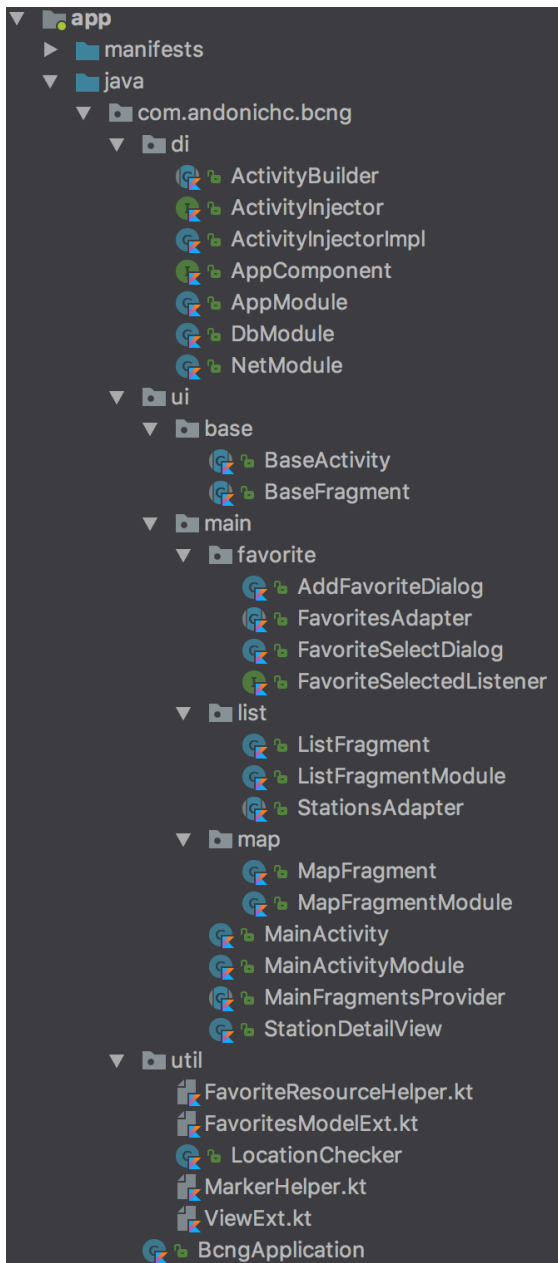


Figura 34. Estructura del mòdul d'app

El mòdul d'app és l'encarregat de connectar tots els altres mòduls i de contindre en si totes les classes del framework d'android necessàries per a que l'aplicació funcioni.

El primer package que trobem és el de di (dependency injector) en ell es poden trobar la majoria de classes necessàries per a la injecció de dependències.

A ui (interfície d'usuari) trobem separat per característica les classes corresponents a la interfície d'usuari, en aquest cas Activities i Fragments que implementen les interfaces de View corresponents i també trobem la part d'injecció de dependències específica per a cada activitat o fragment, en aquesta classe, a part d'injectar tot el que l'activity i el presenter necessiten es fa l'associació entre la classe Activity que implementa View i la implementació de la interfície Presenter corresponent, ja que ambdós guarden referència de la interface i no de la classe concreta.

A BaseActivity i BaseFragment es troba el flux base entre el presenter i l'activity així com alguns mètodes comuns entre les classes que n'hereden d'elles.

També trobem un package de utils que son files (ja que kotlin pot tindre funcions sense pertànyer a una

classe) i classes que ens serveixen d'ajuda a diferents parts de l'aplicació.

Finalment trobem la classe BicingApplication, la qual representa a l'aplicació en si i es executada a l'inici de la aplicació i en gestiona el cicle de vida, en el projecte s'utilitza per a crear el graf de dependències i acoplar-ho al cicle de vida de l'Activity.

5.4. Proves realitzades

Per a realitzar les proves s'ha provat en dos dispositius, un Google Pixel XL 2 amb android 8.1.0 (SDK 27) i un emulador amb android 4.4(SDK 19), la mínima suportada.

5.4.1 Entrar a l'aplicació

Acció	Resultat
<i>Entrar a l'aplicació amb un dispositiu major o igual a Android 23 i acceptar els permisos de localització</i>	OK L'app inicia correctament i es mostra el mapa centrat en la localització de l'usuari i el botó de centrar en la meua localització és visible.
<i>Entrar a l'aplicació amb un dispositiu major o igual a Android 23 i no acceptar els permisos de localització</i>	OK L'app inicia correctament i es mostra el mapa centrat en la localització per defecte, sense visualitzar el botó de centrar en la meua posició.
<i>Entrar a l'aplicació amb un dispositiu inferior a Android 23</i>	OK L'app inicia correctament i es mostra el mapa centrat en la localització de l'usuari i el botó de centrar en la meua localització és visible.
<i>Entrar a l'aplicació sense connexió a internet</i>	OK L'app s'inicia centrada en el punt per defecte.

5.4.2 Vista de mapa

Acció	Resultat
<i>Refrescar estacions</i>	OK Les estacions es refresquen i es mostren d'acord amb el que es rep del servei de bicing.
<i>Seleccionar una estació</i>	OK El mapa es centra en aquella estació i n'apareix la vista de detall
<i>Afegir una estació a un nou favorit</i>	OK L'estació s'afegeix al nou favorit el qual també es crea correctament.
<i>Afegir una estació a un favorit ja existent</i>	OK L'estació s'afegeix al favorit i aquest s'actualitza correctament.
<i>Esborrar una estació de favorits d'un favorit amb més estacions</i>	OK L'estació es mostra sense favorit i el favorit ja no la conté al filtrar per favorits.
<i>Esborrar una estació de favorits d'un favorit sense cap altra estació</i>	OK L'estació es mostra sense favorit i el favorit ja no la conté al filtrar per favorits.
<i>Seleccionar un favorit amb estacions</i>	OK El mapa es centra en les estacions contingudes per aquell favorit i només es mostren aquestes en el mapa.
<i>Seleccionar un favorit sense estacions</i>	OK No es mostra cap estació en el mapa i la vista no es mou.
<i>Esborrar un favorit amb estacions</i>	OK El favorit deixa de mostrar-se i les estacions que tenia com a favorites es deixen de mostrar com a favorites.
<i>Esborrar un favorit sense estacions</i>	OK El favorit deixa de mostrar-se.
<i>Tancar el menú de favorits</i>	OK La vista torna mostrar-se sense filtres.
<i>Intentar afegir més de 5 favorits</i>	OK Els 5 primers s'afegeixen correctament, al intentar marcar una estació com a favorita no es mostra la opció d'afegir-ne'n un altre.
<i>Centrar en la ubicació quan la ubicació està activa</i>	OK El mapa es centra amb un zoom per defecte en la localització del usuari.
<i>Canviar al mode llista</i>	OK La vista canvia a llista correctament.

5.4.3 Vista Llistat

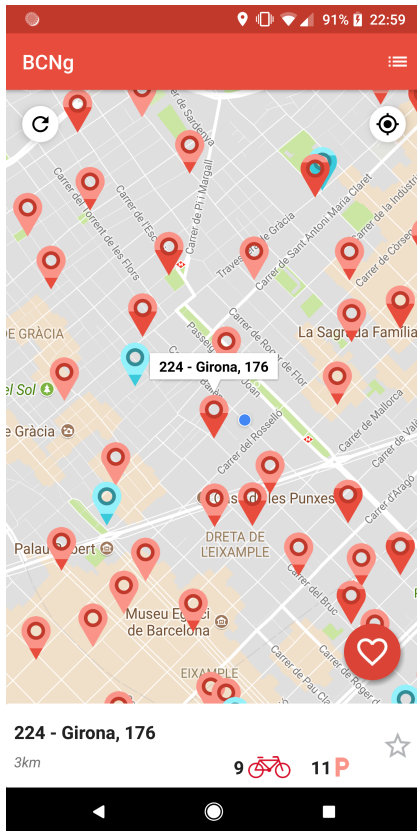
Acció	Resultat
<i>Refrescar estacions</i>	OK Les estacions es refresquen i es mostren d'acord amb el que es rep del servei de bicing.
<i>Afegir una estació a un nou favorit</i>	OK L'estació s'afegeix al nou favorit el qual també es crea correctament.
<i>Afegir una estació a un favorit ja existent</i>	OK L'estació s'afegeix al favorit i aquest s'actualitza correctament.
<i>Esborrar una estació de favorits d'un favorit amb més estacions</i>	OK L'estació es mostra sense favorit i el favorit ja no la conté al filtrar per favorits.
<i>Esborrar una estació de favorits d'un favorit sense cap altra estació</i>	OK L'estació es mostra sense favorit i el favorit ja no la conté al filtrar per favorits.
<i>Seleccionar un favorit amb estacions</i>	OK El llistat mostra només les estacions contingudes pel favorit.
<i>Seleccionar un favorit sense estacions</i>	OK No es mostra cap estació al llistat.
<i>Esborrar un favorit amb estacions</i>	OK El favorit deixa de mostrar-se i les estacions que tenia com a favorites es deixen de mostrar com a favorites.
<i>Esborrar un favorit sense estacions</i>	OK El favorit deixa de mostrar-se.
<i>Intentar afegir més de 5 favorits</i>	OK Els 5 primers s'afegeixen correctament, al intentar marcar una estació com a favorita no es mostra la opció d'afegir-ne'n un altre.
<i>Tancar el menú de favorits</i>	OK La vista torna mostrar-se sense filtres.
<i>Canviar al mode mapa</i>	OK La vista canvia a mapa correctament.

6. Manual d'ús de l'aplicació

Com ja hem vist l'aplicació conta amb dues pantalles, la de mapa i la de llistat, intercanviables entre si pel botó de la dreta contingut a l'appbar.



6.1. Vista Mapa



6.1.1 Refrescar

Per a refrescar la informació només fa falta clicar al botó amb que conté una fletxa en cercle situat a dalt a l'esquerra.



6.1.2 Centrar en la meua posició

Per a centrar el mapa en la posició actual (si aquesta està activada) clicar en el botó blanc d'a dalt a la dreta del mapa.



6.1.3 Seleccionar una estació

Per a seleccionar una estació i veure'n la informació cliqueu a qualsevol dels pins del mapa.



Figura 35. Captura de pantalla de la vista de mapa de l'aplicació

6.2 Vista llistat

6.2.1 Refrescar

Per a refrescar s'ha de fer swipe cap a baix estant a la primera posició de la llista (fer un pull to refresh).

6.3 Favorits

6.3.1 Afegir una estació a un favorit



Clicar sobre el icona d'estrella, ja sigui des de la vista del mapa o des de la llista.

Un cop clicat, s'obrirà una finestra emergent que permetrà seleccionar una de les llistes de favorits ja existents (si n'hi ha) o afegir-ne un nou.

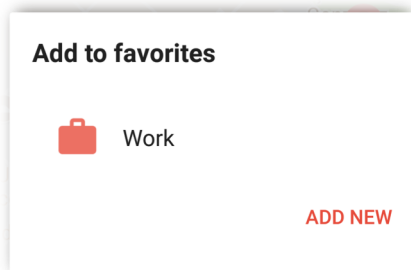


Figura 37. Popup per a seleccionar favorit.

6.3.2 Eliminar una estació de favorits



Per a eliminar una estació de favorits, només cal que cliquem al símbol que ens indica que pertany a una llista de favorits.

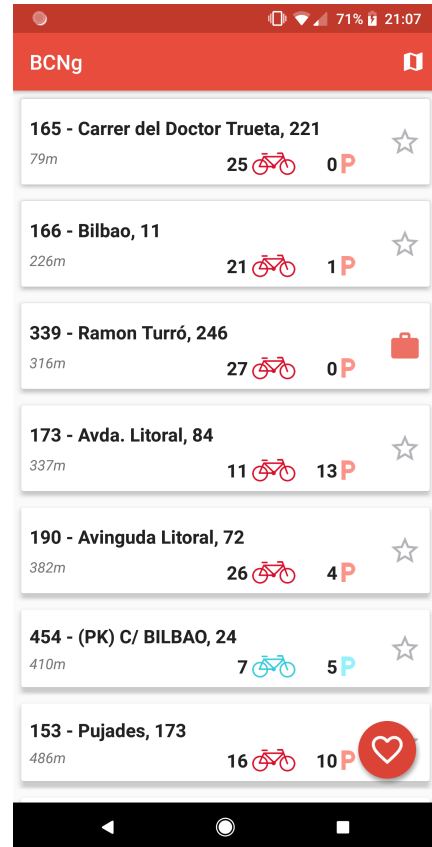


Figura 36. Captura de pantalla de la vista de llista de l'aplicació

6.3.3 Crear una llista de estacions favorites

Un cop a la finestra emergent de selecció de llista favorita accedida al punt anterior, clicar afegir nou. Un cop clicat una altra finestra emergent s'obrirà, aquesta permetrà afegir un nom, seleccionar una icona i guardar el favorit.

S'ha de tindre en compte que el màxim de llistes de favorits que pot haver-hi al mateix temps són 5.

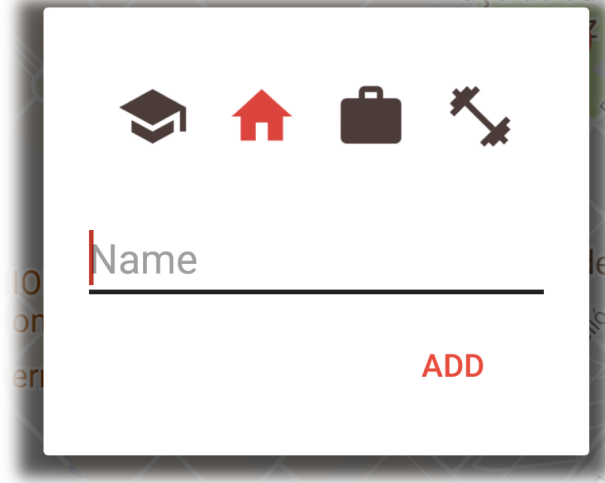


Figura 38. Popup per a afegir un favorit.

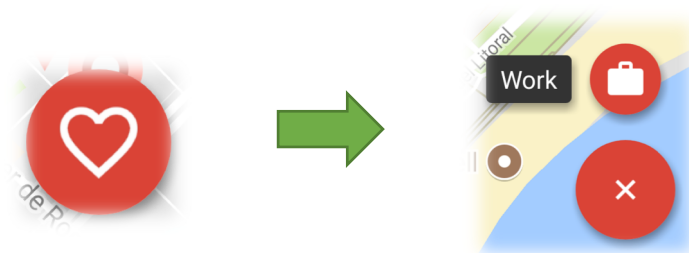
6.3.4 Eliminar un llistat de favorits

Per a eliminar un llistat de favorits, hem de clicar al botó vermell d'a baix a l'esquerra en forma de cor (des de qualsevol vista) i allà mantenir pressionat (fer un long press) sobre el favorit que volem eliminar. Ens sortirà una finestra emergent demanant-nos confirmació de l'acció, i un cop acceptada el llistat de favorits queda esborrat i les estacions que nen formaven part passen a no estar a favorits.

6.3.5 Filtrar per favorits

Tant al mapa com a la llista, per a filtrar i veure només les estacions corresponents a una llista de favorits, hem de clicar al botó vermell d'a baix a la dreta amb el símbol d'un cor.

Un cop clicat se'ns desplegarà un menú el qual utilitzarem per seleccionar el llistat de favorits pel que volem filtrar. D'aquesta manera veure'm només les estacions d'aquella llista de favorits. Per a tornar a la vista sense filtrar només farà falta clicar al botó d'a baix a l'esquerra que ara conté el símbol X.



7. Conclusions

Amb el desenvolupament d'aquest treball s'ha aconseguit desenvolupar un projecte de software des de 0. Passant per tots els passos que aquest requereix.

A l'etapa de plantejament s'ha après a partir d'una necessitat establir uns objectius i planificar i quantificar l'esforç.

A l'etapa de disseny, a partir d'un estudi de les alternatives i dels usuaris s'han elaborat uns requisits i a partir d'allà els casos d'ús i la conceptualització gràfica d'aquest.

Finalment a l'etapa de desenvolupament s'ha aprofundit més en els aspectes tècnics de desenvolupament per a tecnologies mòbils adquirits al màster i a l'experiència laboral, així com al plantejament de quines serien les proves idònies per a l'aplicació.

El treball ha aconseguit complir la majoria dels objectius establerts:

- A nivell de projecte, s'ha implementat una aplicació que millora la oficial, aportant una interfície d'usuari més nativa i amb la possibilitat d'afegir favorits. No s'ha aconseguit complir, i quedaria com a millora la inclusió dels carrils bici de la ciutat degut a que la implementació tècnica d'aquesta inflava massa el desenvolupament.
- A nivell dels consumidors, l'aplicació es mostra més atractiva a nivell personal i com hem dit en el punt anterior incorpora noves prestacions.
- A nivell personal, com ja s'ha comentat anteriorment s'ha aprofundit en els coneixements ja adquirits. No s'ha assolit el objectiu d'afegir tests unitaris i d'instrumentació, els quals queden com a millora tècnica per falta de temps.

La planificació s'ha seguit dintre de l'establert majoritàriament, sense cap alteració destacable.

Finalment les línies de millora que pot seguir aquest treball inclouen:

- Afegir els carrils bici de Barcelona al mapa.
- Incloure test unitaris i d'implementació per tal de no dependre tant del test manual.
- Filtrar per estacions mecàniques i elèctriques.
- Afegir una vista de detall amb tota la informació sobre una estació que proporciona l'api.

8. Glossari

Api - Application Programming Interface

Conjunt de subrutines, funcions i procediments (o mètodes, en la programació orientada a objectes) que ofereixen certa "biblioteca" per a ser utilitzat per un altre software com a capa d'abstracció.

Java

Llenguatge de programació orientat a objectes, nascut en el 1996, utilitzat en molts àmbits de la programació, entre elles el desenvolupament en Android.

Kotlin

Llenguatge de programació orientat a objectes, nascut en el 2012, utilitzat principalment per a desenvolupament en Android i en servidors. S'està plantejant com a substitut al llenguatge Java.

MVP - Model View Presenter

Derivat del MVC (Model Vista Controlador) és un patró d'arquitectura per a interfícies gràfiques.

App - Aplicació

Versió abreviada que fa referència a una aplicació mòbil.

Scrollejar

Verb que fa referència a l'acció de moure el dit horitzontal o verticalment per la pantalla amb la finalitat de desplaçar-se per un contingut.

Long press

Acció de pressionar i mantenir pressionat un segon o més.

SDK - Software Development Kit

Kit de desenvolupament de software en català, reuneix el conjunt d'eines necessàries per al desenvolupament en alguna tecnologia específica.

9. Bibliografía

1. Bicicletas Compartidas – Wikipedia
https://es.wikipedia.org/wiki/Sistema_de_bicicletas_compartidas
(12/03/2018)
2. Bicing – Web Oficial
<https://www.bicing.cat/es/> (12/03/2018)
3. Bicing – Wikipedia
<https://es.wikipedia.org/wiki/Bicing> (12/03/2018)
4. App Bicing – PlayStore
<https://play.google.com/store/apps/details?id=es.bcn.bicing&hl=es>
(12/03/2018)
5. Material Design Guidelines
<https://material.io/> (12/03/2018)
6. CityBiker – PlayStore
<https://play.google.com/store/apps/details?id=com.ekgdev.ekbicing&hl=es> (12/03/2018)
7. Bicing Barcelona – PlayStore
<https://play.google.com/store/apps/details?id=com.altairapps.bicing&hl=es> (12/03/2018)
8. Github
<https://github.com/> (13/03/2018)
9. OpenBicing
<https://github.com/leolchat/openbicing> (13/03/2018)
10. Google forms
<https://www.google.es/intl/es/forms/about/> (13/03/2018)
11. Sketch
<https://www.sketchapp.com/> (13/03/2018)
12. Android Studio
<https://developer.android.com/studio/index.html> (13/03/2018)
13. Git
<https://git-scm.com/> (13/03/2018)
14. Office Suite
<https://www.office.com/> (13/03/2018)

15. InVision <https://www.invisionapp.com/> (01/04/2018)
16. Dagger 2 <https://google.github.io/dagger/> (02/04/2018)
17. SugarORM <http://satyan.github.io/sugar/> (03/04/2018)
18. Room <https://developer.android.com/topic/libraries/architecture/room.html> (03/04/2018)
19. Anunci en el que Android fa oficial el suport de Kotlin com a llenguatge oficial de programació en Android. <https://android-developers.googleblog.com/2017/05/android-announces-support-for-kotlin.html> (14/05/2018)
20. Llibreria d'Android Support <https://developer.android.com/topic/libraries/support-library/> (14/05/2018)
21. Llibreria de Google Maps <https://developers.google.com/maps/documentation/android-api/?hl=es-419> (14/05/2018)
22. Dagger <https://google.github.io/dagger/> (14/05/2018)
23. RxJava <https://github.com/ReactiveX/RxJava> (14/05/2018)
24. Retrofit <http://square.github.io/retrofit/> (14/05/2018)
25. Floating Action Button <https://github.com/Clans/FloatingActionButton> (14/05/2018)
26. Solid principles <https://es.wikipedia.org/wiki/SOLID> (14/05/2018)

10. Annexos

6.1 Annex A – Enquesta repartida (En vermell els resultats)

Encuesta sobre el uso de la bicicleta

*Obligatorio

1. Dirección de correo electrónico *

2. ¿Con cuánta frecuencia utilizas la bicicleta? *

Marca solo un óvalo.

- 5 1 o 2 días por semana
9 Entre 3 y 5 días por semana
16 Diariamente
2 No la uso
0 Otro: _____

3. ¿La sueles usar para ir entre o a ubicaciones que frecuentas con asiduidad? *

Por ejemplo ir y volver del trabajo a casa.

Marca solo un óvalo.

- 28 Sí
4 No
0 Otro: _____

4. ¿Utilizas algún sistema de BikeSharing? *

Marca solo un óvalo.

- 20 Sí, el Bicing *Pasa a la pregunta 4.*
8 No, utilizo bicicleta propia. *Deja de rellenar este formulario.*
2 Otro: _____ *Deja de rellenar este formulario.*

El Bicing

5. ¿Con cuánta frecuencia utilizas el Bicing? *

Marca solo un óvalo.

- 1 1 o 2 días por semana
6 Entre 3 y 5 días por semana
13 Diariamente
0 No la uso
0 Otro: _____

6. ¿Utilizas la aplicación de Bicing? *

Marca solo un óvalo.

- 20 Sí
0 No *Deja de rellenar este formulario.*

7. ¿Cual es tu grado de satisfacción con la aplicación? *

Marca solo un óvalo.

1 2 3 4 5

2,2 (Avg)

Nada satisfecho Muy satisfecho

8. ¿Crees que la aplicación tiene una interfaz de usuario bien pensada y es fácil de entender? *

Marca solo un óvalo.

1 2 3 4 5

3,5 (Avg)

Para nada Sí, está genial

9. ¿Utilizas la función de favoritos? *

Marca solo un óvalo.

0 Sí

4 No

16 No, no sabía que tuviese una.

10. ¿Cuales son para ti los puntos fuertes de la aplicación? *

Selecciona todos los que correspondan.

20 Mapa con las estaciones

0 Favoritos

3 Buscador de rutas

5 Listado de todas las estaciones

11. ¿En qué crees que podría mejorar la aplicación?

6.2 Annex B – Estructura de l'arquitectura de l'aplicació

