

Desenvolupament d'una aplicació d'Android: Antarec

Carles Güell i Noguera

Màster Universitari en Enginyeria Informàtica
Desenvolupament d'Aplicacions sobre Dispositius Mòbils

Consultors

Jordi Ceballos Villach

Jordi Almirall López

Professor responsable de l'assignatura

Robert Clarisó Viladrosa

Data Lliurament: 09/01/2019

Copyright © 2018 Carles Güell i Noguera

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Desenvolupament d'una aplicació d'Android: Andarec</i>
Nom de l'autor:	<i>Carles Güell i Noguera</i>
Nom del consultor/a:	<i>Jordi Ceballos Villach Jordi Almirall López</i>
Nom del PRA:	<i>Robert Clarisó Viladrosa</i>
Data de lliurament :	<i>01/2019</i>
Titulació o programa:	<i>Màster Universitari en Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Desenvolupament d'Aplicacions sobre Dispositius Mòbils</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Bloc Viatges, Android, Kotlin</i>
Resum del Treball (màxim 250 paraules):	
<p>Aquest projecte és el resultat del meu treball final de màster, que consisteix a realitzar una aplicació per a <i>smartphones</i> amb sistema operatiu basat en Android.</p> <p>L'aplicació és un bloc de viatge on es pot apuntar el que s'ha fet cada dia de forma independent, adjuntar-hi arxius multimèdia i tenir-los classificats en una llista de manera que les persones usuàries puguin recordar els diferents llocs que han visitat durant un viatge o anotar de forma ràpida una ubicació que volen recordar o que els ha agradat,</p> <p>Aquesta idea va néixer després de fer un viatge amb una persona que apuntava cada dia en un bloc de notes del mòbil les diferents coses que havíem visitat. D'aquesta manera, li era més fàcil explicar-ho a la seva família. De totes maneres, al final també ens va ajudar a nosaltres mateixos a recordar què havíem fet cada dia durant tot el viatge.</p> <p>El desenvolupament i l'elaboració d'aquest projecte s'ha fet seguint el mètode tradicional en cascada. S'ha iniciat amb una fase d'anàlisi i indagació a usuaris per a poder realitzar el disseny de l'aplicació en una segona fase, seguint la tècnica del disseny centrat en l'usuari. A continuació, s'ha desenvolupat l'aplicació i, finalment, hi ha la fase de manteniment.</p> <p>L'aplicació s'ha desenvolupat amb l'entorn d'Android Studio. El llenguatge de programació escollit ha sigut Kotlin. Utilitzant API's de Google entre altres llibreries de tercers s'ha pogut elaborar el producte final proposat en aquest treball.</p>	

Abstract (in English, 250 words or less):

This project is the result of my master's final project, which consists of creating an application for smartphones with an Android-based operating system.

The application is a travel diary where you can write down what you have done each day, attach multimedia files and have them classified in a list so that users can remember the different places they have visited during a trip or save a location which they want to remember or because they liked.

This idea was born after going on a trip with a person who wrote down the different things we had visited every day in a notebook app on her mobile phone. This way, it was easier for her to explain the trip to her family. Anyway, in the end it also helped ourselves to remember what we had done each day throughout the trip.

The development and elaboration of this project has been done following the traditional waterfall model for software development. It began with a phase of analysis and user inquiry in order to perform the design of the application in a second phase, following the user centered design method. Next, the application was developed and, finally, there was the maintenance phase.

The application has been developed with the Android Studio environment. The programming language chosen was Kotlin. The final product proposed in this project was developed Using Google APIs among other third-party libraries.

Índex

1. Introducció	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball	5
1.3 Enfocament i mètode seguit	6
1.4 Planificació del Treball	8
1.5 Breu sumari de productes obtinguts	10
1.6 Breu descripció dels altres capítols de la memòria	10
2. Disseny i Arquitectura.....	12
2.1 Disseny Centrat en l'Usuari	12
2.1.1 Anàlisi.....	13
2.1.1.1 Entrevistes.....	14
2.1.1.2 Conclusions entrevistes.....	16
2.1.1.3 Enquestes.....	16
2.1.1.4 Conclusions enquestes	26
2.1.1.5 Perfils d'usuari	27
2.1.2 Disseny Conceptual	29
2.1.2.1 Fluxos d'interacció	34
2.1.3 Prototipatge.....	35
2.1.3.1 Esbossos.....	35
2.1.3.2 Prototipus alta fidelitat	39
2.1.4 Avaluació	42
2.1.5 Iteracions en el disseny.....	44
2.2 Disseny Tècnic.....	45
2.2.1 Casos d'ús	45
2.2.2 Disseny Arquitectura.....	49
2.2.2.1 Arquitectura Global del Sistema.....	49
2.2.2.2 Arquitectura MVC.....	50
2.2.2.1 Disseny de la base de dades	51
2.2.2.2 Diagrama Classes del controlador	53
3. Implementació	54
3.1 Eines utilitzades en el desenvolupament.....	54
3.1.1 Android	54
3.1.1 Android Studio.....	54
3.1.2 Kotlin.....	55
3.1.3 APIs.....	55
3.1.4 GitHub	56
3.2 Aspectes importants del desenvolupament	57
3.2.1 Projecte Android Studio	57
3.2.2 Activitats i Fragments.....	58
3.2.3 Disseny de les vistes per a diferents mides de pantalla	59
3.2.4 Vista llista viatges (<i>RelativeLayout</i>)	60
3.2.5 Fitxa restaurant	62
3.2.6 Estructura Firebase Realtime Database.....	62
3.2.7 Emmagatzematge i Recuperació de dades a Firebase Realtime Database.....	65
3.2.8 Adjuntar imatge a un dia des de la càmera	66
3.2.9 Emmagatzematge d'imatge a Firebase Storage	67
3.2.10 Recuperació d'imatges de Firebase Storage i presentació amb Glide ...	68

3.2.11	Ús de la Api de Google Maps pel Trip Header.....	69
3.2.12	Ús de Google Place Picker	69
3.2.13	Dagger Framework.....	70
3.3	Pla de Proves	71
3.3.1	Proves automàtiques	71
3.3.1.1	Proves Unitàries.....	72
3.3.1.2	Proves Integració i UI	72
3.3.2	Proves manuals.....	72
3.4	Funcionalitat de l'aplicació.....	74
3.4.1	Llista viatges	74
3.4.2	Viatge Header	74
3.4.3	Dies de un viatge.....	75
3.4.4	Adjuntar un restaurant	75
4.	Conclusions	76
4.1	Treball futur	77
5.	Glossari	78
6.	Annexos	79
6.1	Anex1 Diagrama Complet MVC	79
7.	Bibliografia	80
7.1	Recursos de text.....	80
7.2	Recursos Web	80
7.3	Recursos Software	81

Llista de figures

Il·lustració 1 Usuaris Mòbil vs Connexions Internet	1
Il·lustració 2 Connexions internet segons dispositiu	2
Il·lustració 3 Ingressos Aplicacions Mobils nivell mundial	3
Il·lustració 4 Model basat en cascada	7
Il·lustració 5 Cicle de vida del projecte	8
Il·lustració 6 Diagrama de Gantt del projecte	10
Il·lustració 7 Model DCU	12
Il·lustració 8 Preguntes 1,2,3 de l'enquesta	17
Il·lustració 9 Preguntes 4,5,6 de l'enquesta	18
Il·lustració 10 Preguntes 6,7,8,9 de l'enquesta	19
Il·lustració 11 Pregunta 10 de l'enquesta	19
Il·lustració 12 Pregunta 11 de l'enquesta	20
Il·lustració 13 Divisió per edats dels usuaris	21
Il·lustració 14 Divisió temps ús d'internet dels usuaris	21
Il·lustració 15 Dispositius més utilitzats pels enquestats per a navegar	22
Il·lustració 16 Divisió per número de viatges a l'any	22
Il·lustració 17 Divisió per motius de viatge	22
Il·lustració 18 Ús del mòbil dels usuaris quan viatgen	23
Il·lustració 19 Divisió dels usuaris segons si s'apunten coses o no durant els viatges.....	23
Il·lustració 20 Divisió segons si han fet mai un bloc de viatges	24
Il·lustració 21 Divisió usuaris si han compartit experiències viscudes durant un viatge amb persones de confiança	24
Il·lustració 22 Divisió d'usuaris segons si donen o no opinió a internet	25
Il·lustració 23 Coneixement de les webs competència dels usuaris	25
Il·lustració 24 Valoració de les funcionalitats proposades	26
Il·lustració 25 Diagrama de flux	34
Il·lustració 26 Esbós pantalla càrrega	35
Il·lustració 27 Esbós pantalla principal plena	35
Il·lustració 28 Esbós pantalla viatge nou buida.....	36
Il·lustració 29 Esbós pantalla pestanyes dies	36
Il·lustració 30 Esbós pantalla principal menú opcions	37
Il·lustració 31 Esbós pantalla pestanyes menú adjuntar	37
Il·lustració 32 Esbós pantalla pestanyes buida.....	38
Il·lustració 33 Esbós pantalla pestanyes amb 1 dia.....	38
Il·lustració 34 Prototipus alta fidelitat I	39
Il·lustració 35 Prototipus alta fidelitat II	40
Il·lustració 36 Prototipus alta fidelitat III.....	40
Il·lustració 37 Prototipus alta fidelitat IV	41
Il·lustració 38 Prototipus alta fidelitat V	42
Il·lustració 39 Prototipus alta fidelitat després iteració.....	44
Il·lustració 40 Arquitectura global del sistema.....	49
Il·lustració 41 Diagrama MVC	50
Il·lustració 42 Diagrama classes Base de Dades.....	52
Il·lustració 43 Diagrama classes del controlador	53
Il·lustració 44 Arquitectura Android OS	54
Il·lustració 45 Estructura Android Studio	57
Il·lustració 46 Estructura paquets i classes del projecte.....	58
Il·lustració 47 ConstraintLayout de la vista HeaderDetailFragment	60

Il·lustració 48	List Adapter a MainActivity	60
Il·lustració 49	Classe TripListAdapter	61
Il·lustració 50	Vista trip_list_item.xml	61
Il·lustració 51	Classe TripItemViewHolder	61
Il·lustració 52	Creació dinàmica fitxa restaurant	62
Il·lustració 53	Estructura Firebase Realtime Database	63
Il·lustració 54	Estructura Firebase Realtime Database II	64
Il·lustració 55	Referència Firebase al node <i>trips</i>	65
Il·lustració 56	Firebase guardar viatge	65
Il·lustració 57	Firebase llegir dades	66
Il·lustració 58	Crear fitxer buit per guardar imatge a la galeria	67
Il·lustració 59	Guardar imatge a FirebaseStorage	68
Il·lustració 60	Mostrar imatge utilitzant Glide	68
Il·lustració 61	Fitxer Google API Key	69
Il·lustració 62	AutocompleteFragment	69
Il·lustració 63	PlacePicker	70
Il·lustració 64	PlacePicker result	70
Il·lustració 65	Piràmide de <i>testing</i>	71
Il·lustració 66	Resultat tests unitaris	72
Il·lustració 67	Test UI	72
Il·lustració 68	Pantalla Llista de viatges	74
Il·lustració 69	Pantalla TripHeader	74
Il·lustració 70	Pantalles dies d'un viatge	75
Il·lustració 71	Procés per inserir fitxa restaurant	75

1. Introducció

1.1 Context i justificació del Treball

El ràpid avanç tecnològic que vivim gràcies a la globalització, facilita molt conèixer els esdeveniments que passen a altres països però també a abaratir els costos de dispositius com els *smartphones*. Amb dades de l'article "*Digital in 2018: World's internet users pass the 4 billion mark - We Are Social USA*", l'any 2018 hi ha 4 mil milions de persones amb connexió a internet, i 5 mil milions amb telèfons mòbils.

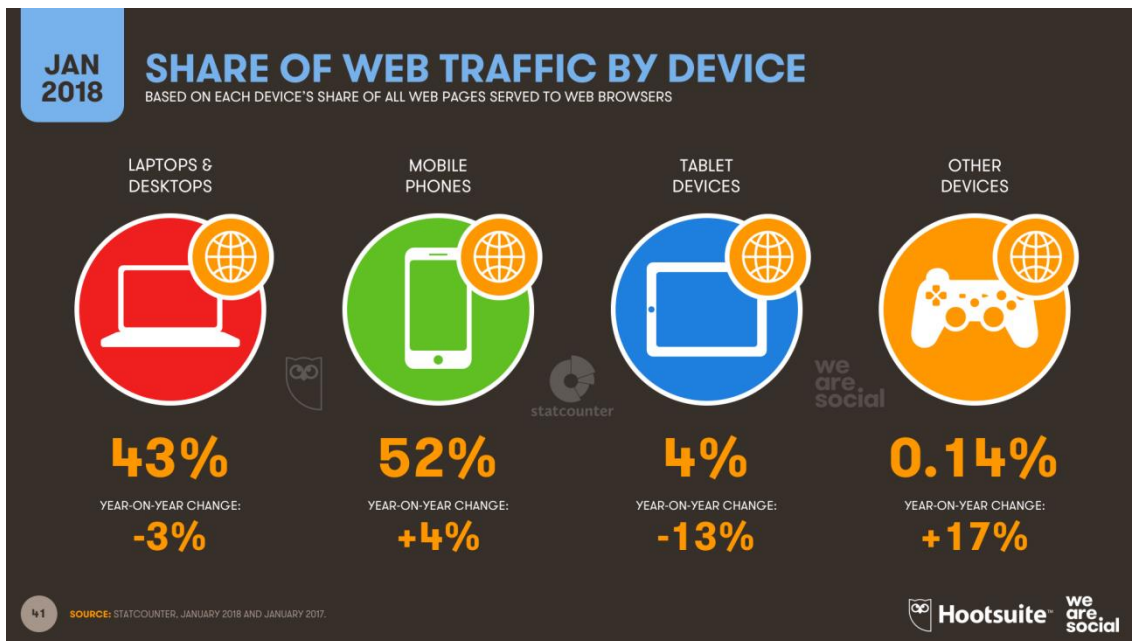
A continuació podem veure un gràfic del mateix article, on podem veure el número d'usuaris de mòbil respecte les connexions a internet realitzades amb mòbil i veiem que hi ha més usuaris de mòbil que no usuaris d'internet. També veiem que hi ha un 68% de la població usuària de telèfon mòbil, això són 2/3 de la població mundial.



Il·lustració 1 Usuaris Mòbil vs Connexions Internet ¹

En aquest altre gràfic del mateix article, podem veure que més de la meitat de connexions a internet han estat fetes amb un mòbil un 52%, superant els navegadors d'ordinadors 43% o altres dispositius menys d'un 5%.

¹ font: Digital in 2018: World's internet users pass the 4 billion mark - We Are Social USA



Il·lustració 2 Connexions internet segons dispositiu²

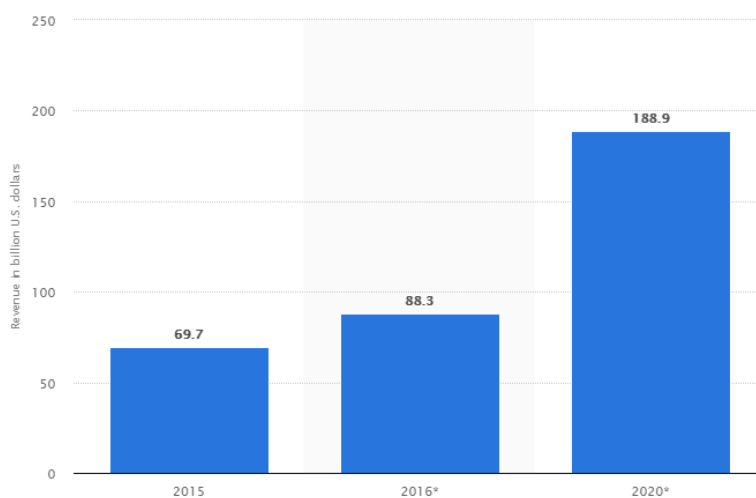
Amb totes aquestes dades, podem afirmar que l'ús dels mòbils incrementa, de fet, cada vegada hi ha dissenys nous, que trenquen amb la forma quadrada dels *smartphones* actuals, i cada vegada són més atractius.

Pel que fa als *smartphones* en concret, a l'article *Insights into the 2.3 Billion Android Smartphones in Use*, a dades de Novembre del 2017, hi ha un total de 3.1 bilions de *smartphones* a nivell mundial, del qual un 75.9% estan basats en el sistema operatiu Android. A l'any 2018 es preveu que hi hagi 300 milions més de *smartphones*.

Amb l'increment de l'ús de *smartphones* i altres dispositius, també incrementa l'ús de les aplicacions. El mercat de les aplicacions mòbils genera molts beneficis a les empreses, tal i com es mostra al gràfic següent, es preveu que per l'any 2020 hi hagi uns ingressos de quasi 189 Milions de dòlars americans.

² font: Digital in 2018: World's internet users pass the 4 billion mark - We Are Social USA

Worldwide mobile app revenues in 2015, 2016 and 2020 (in billion U.S. dollars)



© Statista 2018

II-lustració 3 Ingressos Aplicacions Mobils nivell mundial³

També estan arribant a les llars aparells com Google Home, o Alexa de Amazon, que permeten integrar la domòtica a les cases particulars. Molts d'aquests aparells es poden configurar des del *smartphone*, a part de que també disposen de aplicacions pròpies per a poder configurar-los.

Com podem veure, la globalització i l'ús dels *smartphones* estan a l'ordre del dia, com també ho està el fet de viatjar. Que ha passat de ser una cosa de classe alta, on només les persones amb poder adquisitiu alt es podien permetre, a obrir-se a un mercat *low-cost*, i assequible per a molta gent de classe mitjana.

Segons l'article *LAS TRES CLAVES DEL AUMENTO DE LOS VIAJES INTERNACIONALES EN 2025*, que es vasa en l'estudi de Visa, hi ha un creixement de la classe mitjana a nivell mundial, l'envelliment de la població i l'augment de la connectivitat, ajudaran a una classe social 'viatgera' que impulsarà un augment significatiu en els viatges internacionals a nivell mundial la pròxima dècada.

Es preveu que hi hagi un increment de un 35% a l'any 2025 de viatges amb família respecte el 2015.

Els *smartphones* ajudaran a que hi hagi aquest increment, ja que fa més fàcil que les persones usuàries puguin arribar a ofertes, permeten fer viatges més personalitzats, i poder compartir i opinar sobre llocs visitats o a visitar.

Hi ha persones que quan viatgen els hi agrada anotar cada dia els llocs visitats, fer un diari, per a poder explicar més tard a la família o altres persones què han fet aquell dia. I és aquí on neix la idea de la aplicació, que serà poder

³ font: Insights into the 2.3 Billion Android Smartphones in Use

fer un diari de viatges de forma senzilla, ja que quan una persona viatge vol poder escriure el que esta vivint sense que hi hagi moltes floritures.

Fent una recerca de les *apps* actuals que s'utilitzen com a diaris, o per opinar sobre llocs, he trobat les següents:



Journi: Aquesta aplicació permet crear un viatge, i afegir varis "moments". On cada "moment" pot ser una foto, un escrit, un vol. El disseny de l'aplicació és agradable. Permet posar una foto de fons de cada viatge, i compartir-lo amb altre gent. Però he trobat que permet fer tantes coses, que acaba sent complicat, afegir un escrit, o alguna cosa nova.



Steller: És una aplicació xarxa social, on permet crear viatges i afegir text. Després es publiquen al teu espai i la poden veure altres persones. No m'agrada que sigui xarxa social i que s'hagi de penjar una imatge per a començar escriure.



Trell: És una xarxa social, on permet crear experiències penjant una imatge amb escrit i compartir-ho a altres persones. Molt similar a l'aplicació anterior.



Evernote: És un bloc de notes que es pot compartir amb altres persones. Permet crear varis tipus de notes, ja siguin text, veu, escriptura a mà, adjuntar-hi arxius, o un recordatori. No és una app destinada al viatgers, però es pot fer servir per apuntar i com a diari.



GoogleKeep: És un bloc de notes senzill. Permet afegir una nota i escriure el que vols, en text pla, adjuntar una imatge. O redactar la nota a través de comandes de veu. També permet fer llistes.



TripAdvisor: Aplicació molt utilitzada que permet donar opinió de tots els llocs del món. Personalment l'utilitzo per informar-me de llocs que vaig, no acostumo a donar opinió. No permet fer un bloc.

De totes les aplicacions estudiades, no n'he trobat cap que permeti fer un bloc de notes, creant un viatge. O eren blocs de notes, o eren viatges amb moltes imatges. Les aplicacions que són més destinades a viatgers, fan funció de xarxa social, i és el que es vol evitar en aquest projecte. Per altra banda, les dues aplicacions bloc de notes clàssic, li faltaria alguna característica més destinada a viatgers.

El que vull aconseguir és una aplicació que permeti crear un diari de viatges, i poder-lo compartir amb les persones que es vulgui, i que aquestes també el puguin editar. D'aquesta manera una persona que viatge sola pot anar anotant com si fos un simple diari, per recordar-ho més tard, o si és un viatge en grup, totes les persones poden anar-lo construint de forma conjunta.

1.2 Objectius del Treball

A continuació trobem un llistat dels objectius del treball:

- 1- L'aplicació permetrà crear viatges entre 2 dates, i un lloc concret. També es pot afegir el número i qui són les persones que viatgen.
- 2- A cada viatge es podran afegir dies, on es podrà escollir lloc, afegir-hi fotos, i un text.
- 3- L'aplicació mostrarà cada viatge amb les dades anotades, o imatges, o ubicacions guardades a les fitxes de viatge.
- 4- L'aplicació permet buscar viatges entre la llista dels creats, i també marcar com a pendants els viatges que es faran en un futur.

Els requisits funcionals i no funcionals de l'aplicació es llisten a continuació:

RF1– Llistar viatges

- RF 1.1: El sistema ha de permetre llistar els viatges creats per l'usuari.
- RF 1.2: El sistema ha de permetre buscar un viatge concret entre tots els creats per l'usuari.
- RF 1.3: El sistema ha de permetre seleccionar afegir o esborrar viatges de la llista.

RF2 – Crear Viatges:

- RF 2.1: El sistema permet crear un viatge entre 2 dates específiques
- RF 2.2: Es pot posar la ubicació del viatge a través de la API de GoogleMaps
- RF 2.3: S'ha de permetre afegir llocs visitats.
- RF 2.4: Ha de permetre afegir llocs de restauració.
- RF 2.5: El sistema ha de permetre que es puguin afegir fotografies en el viatge.
- RF 2.6: El sistema ha de poder guardar el viatge.

RF3 – Modificar Viatge:

- RF 3.1: Quan l'usuari decideix modificar un viatge, s'ha d'obrir la pantalla del viatge seleccionat i poder fer qualsevol de les accions explicades en l'apartat de crear viatge.

RF4 – Eliminar Viatge:

- RF 4.1: Quan l'usuari decideix esborrar un viatge, ha de poder seleccionar el viatge que desitja de la llista, i poder-lo esborrar.

Els requisits no funcionals són els següents:

- RNF1: Els permisos d'accés al *backend* només es podran canviar per l'administrador d'accés a les dades.
- RNF2: L'aplicació s'implementarà per la plataforma Android 5.1 en Kotlin.
- RNF3: El temps d'aprenentatge de la aplicació per un usuari no hauria de ser més de 1 hora.
- RNF4: El sistema ha de proporcionar missatges d'error que siguin informatius i orientats a l'usuari final.
- RNF5: El número d'usuaris connectats a la vegada, està limitat a la llicència del servidor *backend*.
- RNF6: Els tests unitaris es realitzaran mitjançant JUnit, i els tests de funcionalitat es faran amb Espresso.
- RNF7: Es treballà amb GIT per al control de versions del codi, concretament la solució escollida és GitHub.
- RNF8: El servei *backend* del sistema serà un servei *cloud*, concretament Firebase de Google.

1.3 Enfocament i mètode seguit

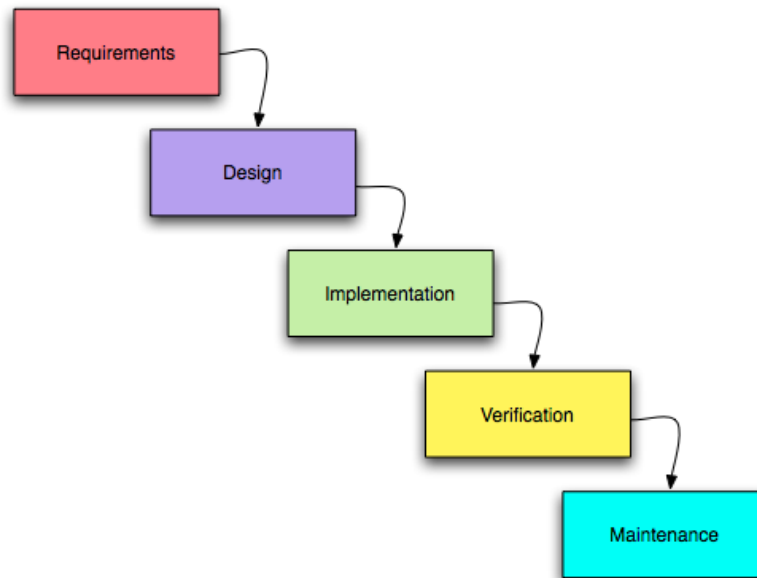
Per a la realització d'aquest treball no duem a terme cap millora a un producte ja existent. Tot i que alguna de les aplicacions analitzades ens podria servir com a base però afegir-hi les funcionalitats desitjades, aquestes són d'empreses privades i tenen el codi tancat.

Per tant, crear una aplicació nova des del principi amb l'arquitectura que més ens convingui i les característiques escollides és la millor opció. Així, podrem arribar a persones que busquen poder fer diaris de viatges de manera simple, amb una imatge i text, i que no volen formar part de cap xarxa social.

El procés que utilitzarem per aquest projecte és el model basat en el cascada pel desenvolupament de software, ja que no serà un projecte molt gran, i

només hi haurà una persona desenvolupant. Per tant es poden anar complint les diferents fases de forma organitzada, i pas a pas.

Es poden observar les diferents etapes en el següent gràfic:



Il·lustració 4 Model basat en cascada⁴

Requisits: En la primera fase és on s'identifica la necessitat que volem cobrir amb l'aplicació, i també on es defineixen els diferents requeriments que tindrà l'aplicació.

Disseny: A la fase del disseny, és on es defineix l'arquitectura de software que s'utilitzarà i els diferents components. També es dissenya l'aplicació, amb les diferents interfícies necessàries per a complir els requisits.

Implementació: Aquesta és la fase més llarga de totes, i és on es fa la programació del codi. Hi podem haver diferents entregues a mesura que es va desenvolupant codi.

Verificació: En aquesta fase es fan tests, es verifica el producte i es busquen possibles errors de codi, i *bugs*.

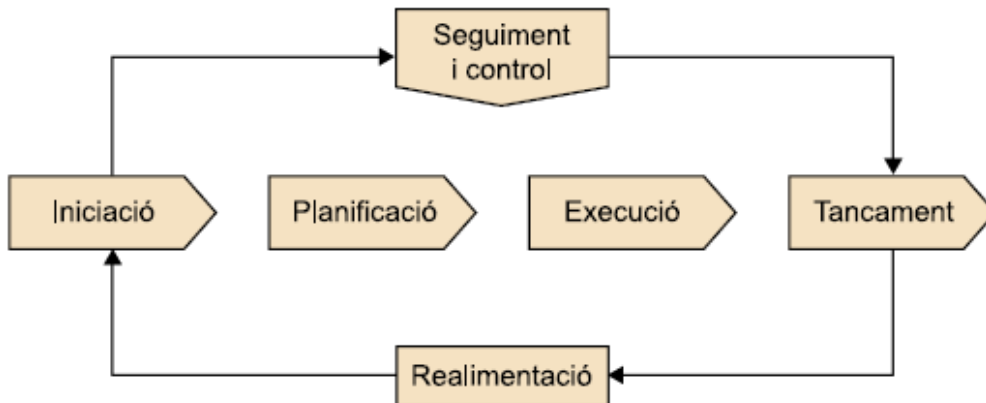
Manteniment: La última fase serveix per mantenir el producte final, corregir-lo i fer-hi millores.

A continuació mostro un gràfic del cicle de vida d'un projecte, extret dels apunts de la UOC, basat en el PMBOK, en què es pot veure, que després de la fase de tancament, hi ha una realimentació que torna al inici. Aquesta aproximació de retro alimentar s'escau més per al cicle de vida del software, ja que en la

⁴ font: Wikipedia

fase de manteniment, hi ha d'haver unes certes pautes per a les millores, i que aquestes no generin nous problemes, que es podrien donar si es fessin sense seguir cap criteri.

Figura 3. Cicle de vida del projecte



II-Il·lustració 5 Cicle de vida del projecte⁵

1.4 Planificació del Treball

Per a fer la planificació m'he basat en el calendari d'entregues de la UOC, el qual està compost de tres proves d'avaluació continua a entregar, marcades al diagrama de Gantt com a entregues parcials, i un lliurament final.

He utilitzat les entregues de la avaluació continua per definir les tasques relacionades amb aquestes i poder elaborar així una planificació completa:

- Pla de treball:
 - Definició del projecte:
 - Anàlisi de requisits
 - Planificació del projecte
 - Entrega PAC1

En aquesta primera fase, és on definim els objectius, i fem l'anàlisi de requisits. I elaborem una planificació de tot el projecte segons les tasques que tindrem.

- Disseny i Arquitectura:
 - Anàlisi funcional (DCU)
 - Disseny Arquitectura
 - Disseny tècnic de l'aplicació
 - Prototipatge
 - Entrega PAC2

En la segona fase, realitzarem les fases del DCU(Disseny Centrat en l'Usuari). Aquestes fases consten d'analitzar els usuaris i context d'ús, fer un disseny conceptual i un disseny prototip per a poder avaluar la nostra aplicació.

⁵ font: UOC - El treball final com a projecte (PID_00197248)

També analitzarem els casos d'ús de la nostra aplicació i dissenyarem quina arquitectura utilitzarem.

- Implementació:
 - Desenvolupament
 - Proves unitàries
 - Integració
 - Proves d'integració
 - Entrega PAC3

Aquesta fase és la de més duració ja que és on implementem les diferents funcionalitats a la nostra aplicació, juntament amb tots els tests i proves unitaris i d'integració.

És possible que no s'implementin totes les funcionalitats, però si les principals per poder tenir una aplicació beta.

- Lliurament final:
 - Memòria del projecte
 - Vídeo de presentació
 - Entrega final

La última fase del projecte consisteix en acabar d'implementar les funcionalitats de l'aplicació, la redacció de una memòria i realitzar un vídeo de presentació de la nostra aplicació.

Els recursos humans que hi haurà en aquest projecte, consten d'una sola persona encarregada del disseny i implementació de l'aplicació, com també de elaborar l'entregar final.

Els recursos de hardware que disposa l'equip de treball són els següents:

- Ordinador Portàtil Samsung Serie 3 amb 8gb de ram, processador Intel Core i7, i SSD Samsung Evo 500gb.
- Mòbil Vodafone SmartSpeed 8gb de memòria interna i 1gb de ram. Dimensions de pantalla 480 x 854 px (4.50") 218 ppi. Android 5.1.

Els recursos de software a utilitzar:

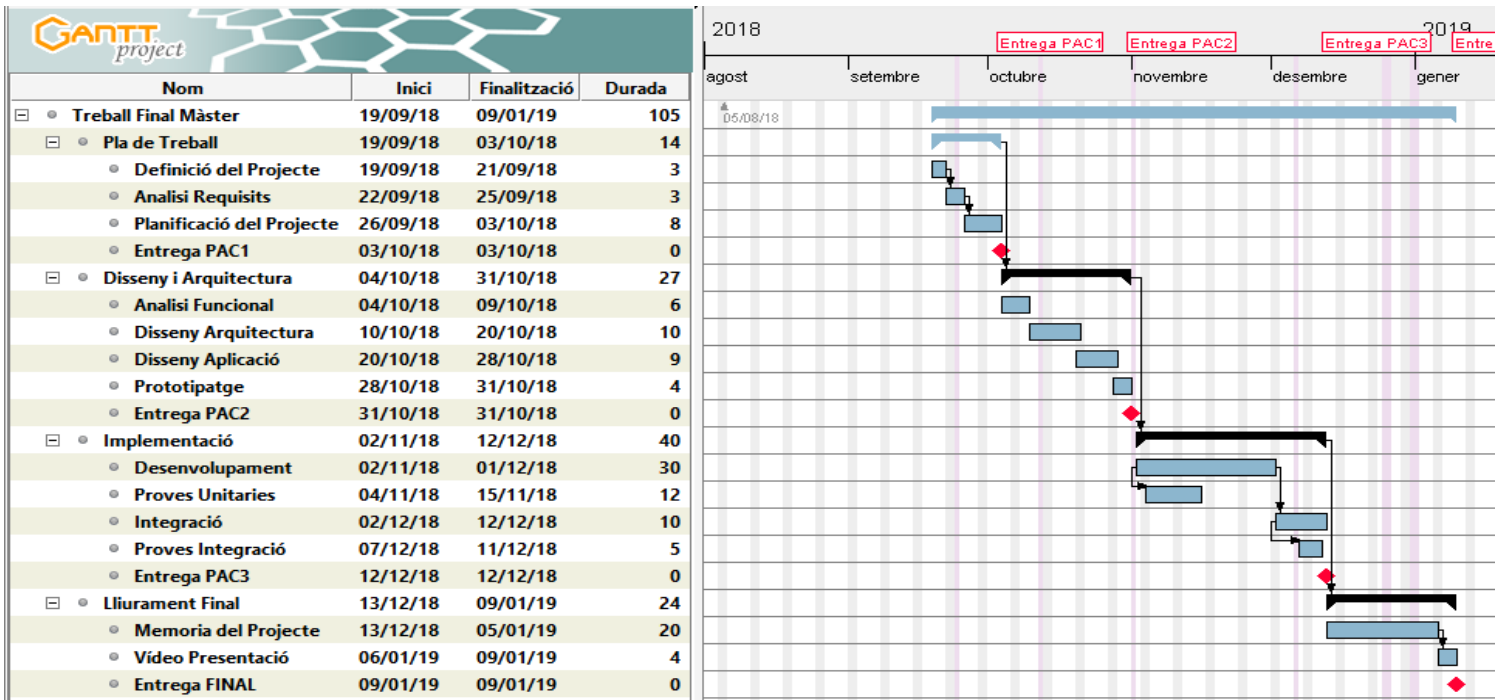
- Android Studio 3.1 amb plugin de Kotlin.
- Firebase de Google per el servidor Backend.
- GitHub amb repositori privat per la gestió de versions de codi, un cop acabat el projecte, es farà públic.

El projecte ha començat a data de 19/09/2018 i acaba el 09/01/2019, total de 113 dies.

Tenint en compte els dies festius de 24 de Setembre, 12 d'octubre, 1 de Novembre, 6,25,26,31 de Desembre i 1 de Gener, tenim un total de 105 dies.

Pel que fa a les hores dedicades al projecte, s'estima que hi hagi 3 hores de dedicació els dies entre setmana i 4 hores els caps de setmana. No es treballarà els dies festius marcats al calendari.

Durant la època de implementació les hores de cap de setmana passaran a ser 8 hores.



Il·lustració 6 Diagrama de Gantt del projecte

1.5 Breu sumari de productes obtinguts

Els productes que s'obtidran al finalitzar el projecte són els següents:

- Aplicació d'Android ja compilada (fitxer *.apk*) per a poder instal·lar a telèfons mòbils.
- Codi font del projecte, on es podrà veure la part de l'aplicació Android, com també la integració amb el *backend*.
- Memòria del projecte.
- Vídeo de presentació, i presentació de suport.

1.6 Breu descripció dels altres capítols de la memòria

En el següent capítol podem trobar la continuació de la fase del disseny iniciada en aquest primer capítol.

Hi haurà tot el temari relacionat amb el disseny centrat en l'usuari i també els diferents casos d'ús analitzats i una explicació del disseny de l'arquitectura.

A continuació trobarem tota la fase de la implementació, amb les principals característiques del projecte i justificacions de les decisions preses.

Juntament amb la implementació hi ha el pla de *testing* plantejat per aquest projecte, on es parla de les proves manuals i automàtiques que tindrà l'aplicació.

Després de la implementació hi ha el capítol de conclusions on hi consta una valoració personal del treball fet, i també si inclou un apartat de treball futur, per explicar els següents passos a fer un cop el projecte s'ha entregat.

Per acabar trobem un glossari dels termes i acrònims del projecte. També trobem els annexos del projecte i la bibliografia emprada per a l'elaboració de la memòria.

2. Disseny i Arquitectura

2.1 Disseny Centrat en l'Usuari

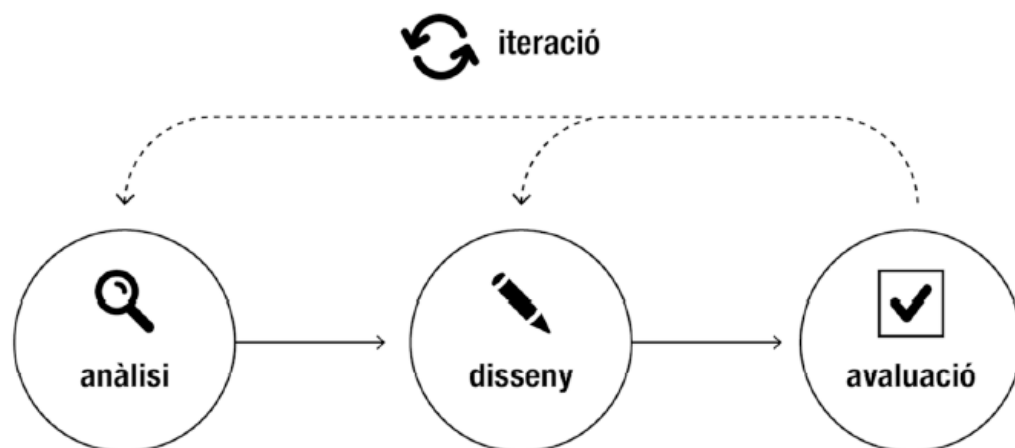
En aquest capítol ens centrem en la tècnica del disseny centrat en l'usuari (a partir d'ara DCU), la qual ens permet situar l'usuari al centre de tot el procés de disseny.

La premissa major d'aquesta metodologia és que si es vol garantir l'èxit d'un producte, s'han de tenir en compte les persones usuàries durant totes les fases del disseny.

Aquesta tècnica s'utilitza per al disseny de la majoria de pàgines web, o aplicacions de software, perquè permeten veure la usabilitat dels productes i com es pot veure en el gràfic de sota, millorar-los abans no hi hagi un producte final.

Durant aquest a fase es poden conèixer els perfils d'usuari, costums, quan fan ús del mòbil, en quines situacions utilitzen el mòbil, i en quines utilitzaran el producte que s'està dissenyant. També permet veure si allò que es dissenya s'entén, i s'utilitza per a cobrir la necessitat que hem vist que hi ha, o també permet observar noves necessitats a cobrir.

A més a més també podem arribar a saber quina experiència tenen els usuaris de productes similars, si saben com funciona el nostre producte, quines funcionalitats esperen del producte, i amb aquest *feedback* podem modificar el disseny. Ja que modificar la interfície, o l'aplicació durant aquesta fase és molt més simple i d'un cost molt més baix que fer-ho un cop ja tenim un producte final.



II-Il·lustració 7 Model DCU⁶

Com podem veure en el gràfic superior, en el DCU tenim tres fases.

⁶ font: UOC – Disseny centrat en l'usuari (PID_00176048)

- **Anàlisi:** En aquesta primera fase, es duen a terme els mètodes d'indagació que permeten obtenir informació qualitativa i quantitativa de les persones usuàries. I analitzant aquesta informació podem trobar necessitats a cobrir i fer un perfil de l'usuari el qual ens volem dirigir, o utilitzarà la nostra aplicació. Alguns d'aquestes mètodes són entrevistes personals, anàlisi de la competència (*Benchmarking*), o enquestes.
- **Disseny:** Aquesta fase es compon de dues parts, una seria la del disseny conceptual, la qual es defineixen els usuaris i els seus perfils, o es creen els escenaris. I l'altra part seria el disseny de prototipus. La qual es pot fer un disseny a mà alçada com a esbós inicial, i posteriorment realitzar una maqueta sense funcionalitat, simplement que es pugui anar d'una pantalla a una altra.
- **Avaluació:** En aquesta última fase es testeja als usuaris els quals utilitzen un prototip d'alta fidelitat i ens serveix per a obtenir informació qualitativa del nostre producte. Un dels inconvenients és que és costós però a canvi s'obté una informació molt valuosa de les interfícies i usabilitat del producte.

Una altra part de l'avaluació consta en avaluar el producte amb algun dels mètodes existents com per exemple l'avaluació de les característiques sobretot de les interfícies, o una inspecció d'estàndards portada a terme per un equip d'experts.

- **Iteració:** s'apliquen els resultats dels testos i avaluacions per a millorar i refinar el disseny, i es torna a testejar i a iterar per tal d'aconseguir un bon disseny.

2.1.1 Anàlisi

Per aquest projecte les tècniques d'avaluació emprades han estat la realització d'una enquesta a diverses persones, entrevistes en profunditat i *benchmarking*.

S'ha escollit el mètode de realitzar una enquesta perquè és una forma d'obtenir dades quantitatives i estadístiques de forma remota, i que no impliquen molt de temps a les persones enquestades, tot i que un dels inconvenients d'aquesta tècnica, és que la informació recollida pot no ser verídica.

Un altre dels mètodes emprats per a la indagació, ha sigut realitzar una entrevista personal a tres persones, per tal de poder obtenir una informació de qualitat i més real, ja que l'entrevista s'ha realitzat en entorns confortables pels entrevistats i seguint un guió poc formal. Amb les entrevistes s'ha pogut conèixer a les persones d'una manera més profunda i veient així inquietuds, formes d'actuar, o el comportament.

Per última la tècnica del *benchmarking* que consisteix en l'estudi de la competència realitzat en l'apartat 1.1 d'aquest projecte.

2.1.1.1 Entrevistes

Les entrevistes personals s'han realitzat a tres persones que han volgut col·laborar en aquest projecte. A continuació detallo el guió seguit per a l'entrevista i també quina informació he extret de les tres persones entrevistades.

- I. Informació personal de la persona entrevistada.
 - a. Nom i edat
 - b. Professió

- II. Informació sobre els seus hàbits amb l'ús de mòbil.
 - a. Quantes hores utilitzes el mòbil cada dia? i Internet?
 - b. En quins llocs utilitzes el mòbil?
 - c. Quin ús n'acostumes a fer, navegar? Xatejar? Comprar?
 - d. Utilitzes algun altre dispositiu per a navegar?

- III. Informació sobre els seus hàbits quan viatge.
 - a. Quants viatges acostumes a fer durant l'any? I quin o quins són els motius del viatge?
 - b. Fas servir el mòbil quan viatges? Per a què?
 - c. Prens nota de les bones i males experiències quan viatges? Has escrit mai un bloc?
 - d. T'agrada explicar els teus viatges amb la resta de família o amics? Com ho fas per en recordar-te de tots els llocs visitats?

- IV. Valoració de l'aplicació.
 - a. Coneixes alguna de les aplicacions següents (Tripadvisor, Trel, GoogleKeep, Evernote, Steller, Journi)? En cas afirmatiu les has utilitzat? Què n'opines?
 - b. Si tinguessis una aplicació que et servis per realitzar un diari de viatge de forma molt simple, l'utilitzaries?
 - c. Quines funcionalitats creus que hauria de tenir una aplicació per a crear diaris de viatge?

Amb aquest guió he pogut conèixer en profunditat els hàbits, costums, ús dels mòbils i d'internet de les persones entrevistades. Així com també obtenir informació sobre quines aplicacions han utilitzat, i quines funcionalitats troben que hauria de tenir una bona aplicació per a poder fer un diari de viatges.

Els resultats de les entrevistes es detallen a continuació:

La primera persona entrevistada es diu Ruth, té 23 anys, i està estudiant i exerceix de professora d'anglès.

El seu ús del mòbil és de 3,5 hores diàries i sobretot ho fa durant el trajectes cap a la feina o quan està per casa. La majoria de temps d'internet el dedica a les xarxes socials utilitzant el mòbil. A la feina utilitza una tauleta o un ordinador, i quan es tracta d'oci també fa servir la televisió per mirar vídeos.

Anualment fa uns 4 viatges i tots per motius d'oci. Durant aquests viatges fa servir el mòbil per gestions del viatge, com podrien ser les targetes d'embarg d'avions o les reserves d'hotel. També el fa servir per informar-se de llocs i restaurants, o els diferents mètodes de transport del lloc que visita.

No ha escrit mai cap bloc de viatges tot i que si que en els viatges llargs, de més de cinc dies, s'apunta el que ha fet cada dia. I en viatges curts, ha fet vídeo blocs explicant el que ha fet cada dia. Així se'n recorda. Per explicar-ho a la família utilitza les fotos que fa durant els viatges, juntament amb el que ha apuntat o gravat.

De les aplicacions que s'han estudiat coneix TripAdvisor, tot i que només l'ha fet servir en format web, i la de GoogleKeep l'ha utilitzat algun cop com a bloc de notes.

Creu que si tingués una aplicació senzilla per a fer blocs de viatge de manera simple, la investigaria com a mínim.

Les funcionalitats que creu més útils són:

Possibilitat de crear àlbums multimèdia, combinant fotos, vídeos o *gifs*.

Poder utilitzar la ubicació.

Tenir un bloc de notes per combinar les anteriors coses, i poder-ho separar per dies.

Poder compartir el bloc amb altres persones perquè ho puguin editar, només les convidades.

Possibilitat de obrir el bloc de viatges a tothom, sense que ho puguin editar.

I poder-ho compartir sense necessitat de tenir la aplicació i que les persones que ho reben puguin descarregar-se totes les fotos.

La segona persona entrevistada es diu Ramon, té 58 anys, i és tècnic de qualitat en laboratoris farmacèutics.

Utilitza el mòbil una hora i mitja al dia i el fa servir per xarxes socials. També navega a través de la tauleta o des de l'ordinador, mes per temes de feina o oci. La majoria d'ús del mòbil és des de casa.

Acostuma a fer un viatge a l'any i no ha escrit mai cap bloc de viatges. Quan explica els viatges als seus amics ho fa amb l'ajuda de les fotos que ha fet durant el viatge.

No coneix cap de les aplicacions que s'han estudiat.

Diu que ara no faria servir cap aplicació per a crear blocs, que fos senzilla, tot i que si fos mes jove sí que ho faria, en funció del viatge i la durada d'aquest.

Les funcionalitats útils que ha comentat són:

Poder escriure i penjar-hi fotos.

Que la gent pugui posar comentaris al bloc.

Que es pugui compartir amb els amics.

L'última persona entrevistada es diu Clara, té 23 anys, i és estudiant.

Fa servir el mòbil entre 3 i 4 hores cada dia, sobretot durant els trajectes fins a la facultat, i un cop a casa navega ja amb el *Macbook* o *iPad*.

El mòbil el fa servir bàsicament per a les xarxes socials, escoltar música, fer fotos i xatejar, en canvi la tauleta o l'ordinador els fa servir mes per treballar o mirar series, o informar-se de llocs.

No ha escrit mai cap bloc de viatges ni tampoc acostuma a prendre nota de les coses quan viatge.

Comenta que per explicar el viatge s'ajuda de les fotos que ha anat fent, i que ho explica tan bon punt arriba, així segur que se'n recorda de tot.

De les aplicacions estudiades coneix TripAdvisor i Evernote. La primera la fa servir per opinar i llegir opinions, i la troba molt útil. La segona la fa servir de bloc de notes.

Creu que no faria servir cap aplicació per fer blocs de viatges, però que ho hauria de provar.

Les funcionalitats que creu necessàries per a una aplicació són:

- Poder afegir viatges que es volen fer, per anotar-hi llocs a visitar. També poder distingir entre els que es volen fer, o les fets.
- Tenir un processador de textos simple.
- Possibilitat d'adjuntar-hi fotos.
- Fos organitzat a l'hora de guardar la informació, tenir les dates del viatge, no la data de quan escrius el text.
- Poder buscar en el mapa restaurants o llocs concrets per adjuntar-ho al processador de textos.

2.1.1.2 Conclusions entrevistes

Un cop realitzades les entrevistes, podem deduir que les persones joves estarien més interessades a tenir una aplicació per a realitzar un bloc de viatges.

Pel que fa a les funcionalitats, la opció de tenir un processador de textos, on poder adjuntar arxius multimèdia i poder-ho compartir amb la resta de persones, sí que seria bàsic per a la aplicació.

Les persones entrevistades han mostrat interès en les aplicacions per a crear blocs i poder-ho organitzar de forma clara i senzilla. I estarien disposades a provar l'aplicació que es realitza en aquest projecte.

2.1.1.3 Enquestes

L'enquesta realitzada per a poder obtenir informació estadística consta d'onze preguntes. D'aquestes preguntes n'hi ha que indaguen sobre l'ús i els hàbits amb *smartphone* de les persones, i n'hi ha sobre els seus hàbits a l'hora de viatjar.

A continuació es pot veure el formulari enviat:

Hàbits d'ús mòbil i viatges

Aquest formulari serveix per a recollir informació estadística per a un futur desenvolupament d'una aplicació mòbil.

* **Necessari**

Franja d'edat *

Trieu una opció ▼

Quanta estona utilitzes internet durant el dia? *

Trieu una opció ▼

Quan navegues per internet, quin dispositiu utilitzes? *

Smartphone

PC

Tablet

Phablet

Altres: _____

II·lustració 8 Preguntes 1,2,3 de l'enquesta

En aquestes dues primeres preguntes, obtenim informació dels enquestats, quanta estona utilitzen internet, i des de quin dispositiu ho acostumen a fer.

Quants viatges acostumes a fer cada any? *

- 0
- 1
- 2-3
- 4-5
- +5
- Altres: _____

En cas de viatjar, per quin motiu acostumes a fer-ho? *

- Oci
- Feina
- Altres: _____

Si vas de viatge, utilitzes el mòbil per a? *

- Informar-me de llocs a visitar
- Utilitzar el gps
- Trucar
- Fer fotos
- Treballar
- Altres: _____

II-lustració 9 Preguntes 4,5,6 de l'enquesta

Acostumes a prendre nota de les coses que t'han agradat, o no, durant els viatges? *

- Sí
- No

Has realitzat mai cap bloc de viatges? *

- Sí
- No

Comparteixes amb persones de la teva confiança experiències viscudes durant un viatge? *

- Sí
- No

Acostumes a donar la teva opinió a internet de llocs visitats, restaurants..? *

- Sí
- No
- No, només llegeixo opinions
- Altres: _____

II-lustració 10 Preguntes 6,7,8,9 de l'enquesta

Utilitzes o has utilitzat alguna d'aquestes aplicacions? *

- Journi
- Steller
- Trell
- Evernote
- GoogleKeep
- TripAdvisor
- Cap d'aquestes
- Altres: _____

II-lustració 11 Pregunta 10 de l'enquesta

La resta de preguntes, com es pot veure, són sobre els costums dels usuaris quan viatgen, si viatgen. Podem saber si fan servir el mòbil i el motiu d'ús quan estan de viatge, i també si mai han fet algun tipus de bloc per a recordar les experiències viscudes.

Valoració 1 al 5 funcionalitats útils d'una aplicació per a realitzar un bloc d'un viatge *

	1	2	3	4	5
Poder crear un viatge per dates i lloc	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Llistat amb tots els viatges creats i poder afegir-ne de nous	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Processador de textos simple	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Divisió per dies del viatge, cada dia amb el seu bloc de notes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Possibilitat d'adjuntar fotos o altres arxius multimèdia, en el bloc de notes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Compartir el viatge i que les persones el puguin editar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

II-lustració 12 Pregunta 11 de l'enquesta

La última pregunta fa referència a les principals funcionalitats que es volen implementar en la aplicació d'aquest projecte.

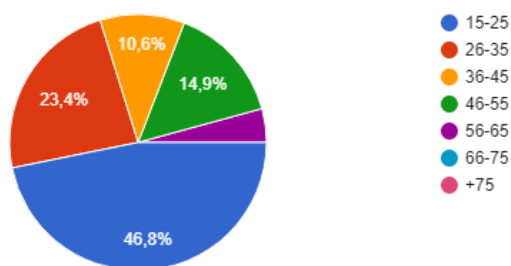
Es pot trobar el formulari clicant en el següent [enllaç](#)⁷.

Els resultats de les enquestes es detallen a continuació:

⁷ <https://goo.gl/forms/qoYbiCLMOPmceSdm2>

Franja d'edat

47 respostes

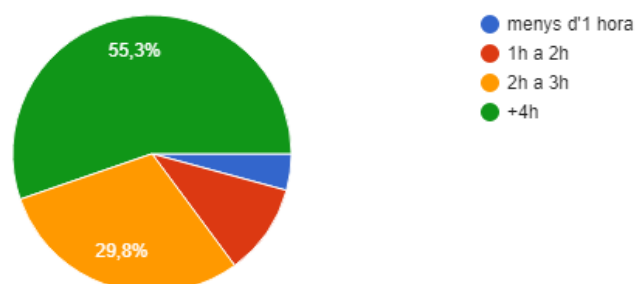


II-lustració 13 Divisió per edats dels usuaris

L'enquesta ha estat contestada per 47 persones, les edats dels quals són entre 15-25 anys la majoria, un 46,8% i un 23,4% són entre 26 i 35 anys. La majoria de gent que ha contestat l'enquesta, és gent jove de 35 anys o menys.

Quanta estona utilitzes internet durant el dia?

47 respostes

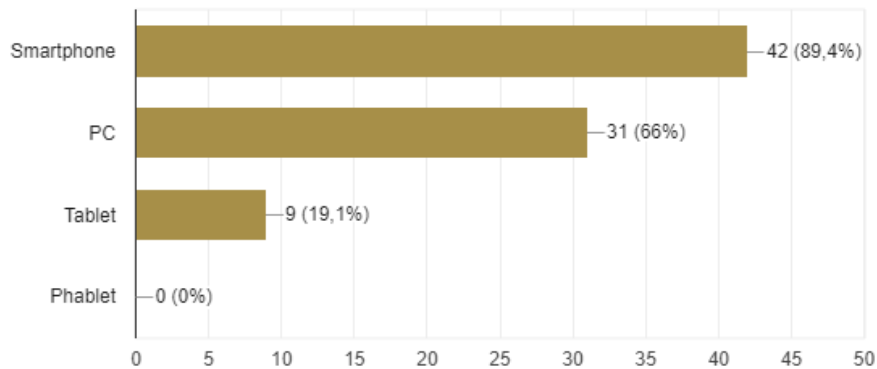


II-lustració 14 Divisió temps ús d'internet dels usuaris

D'aquestes persones la majoria utilitzen internet més de 4hores, per tant estan acostumades a l'ús d'aquest, i l'utilitzen des de diferents dispositius. Hi ha més ús d'internet a través del *smartphone* que no pas des de l'ordinador.

Quan navegues per internet, quin dispositiu utilitzes?

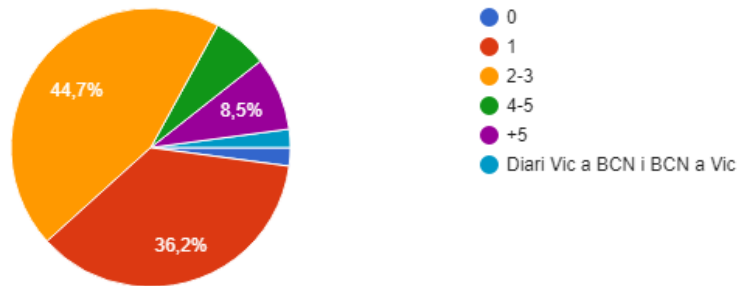
47 respostes



II-lustració 15 Dispositius més utilitzats pels enquestats per a navegar

Quants viatges acostumes a fer cada any?

47 respostes

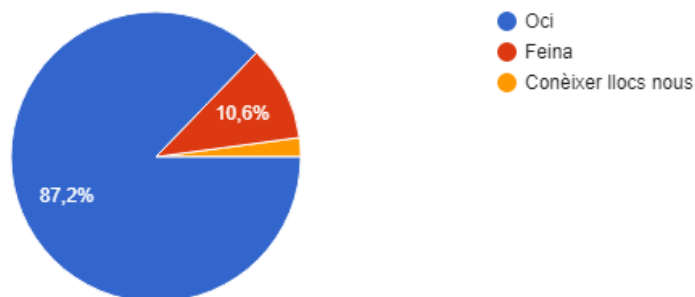


II-lustració 16 Divisió per número de viatges a l'any

Pel que fa als viatges, la majoria de persones realitzen entre 2-3 viatges l'any, o 1 viatge. Veiem que són poques les persones que no fan cap viatge, i alguns que en fan més de 5.

En cas de viatjar, per quin motiu acostumes a fer-ho?

47 respostes

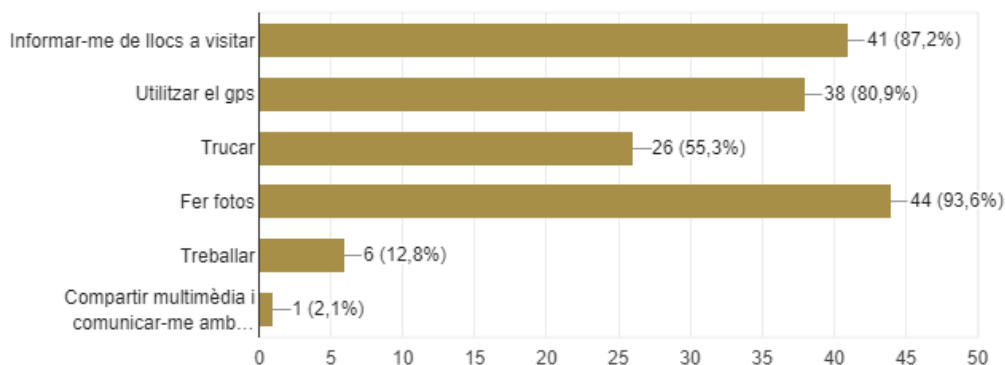


II-lustració 17 Divisió per motius de viatge

La majoria d'aquests viatges són per motius d'oci.

Si vas de viatge, utilitzes el mòbil per a?

47 respostes

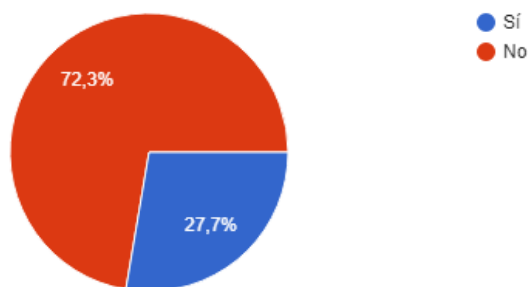


II-lustració 18 Ús del mòbil dels usuaris quan viatgen

Quan les persones viatges fan servir el mòbil per a fer fotos, informar-se de llocs a visitar, o fer servir la funcionalitat de GPS.

Acostumes a prendre nota de les coses que t'han agradat, o no, durant els viatges?

47 respostes

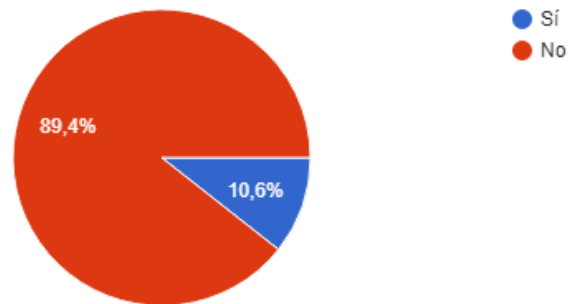


II-lustració 19 Divisió dels usuaris segons si s'apunten coses o no durant els viatges

Com es pot veure en les següents preguntes, un 27.7% pren nota dels llocs que li han agradat, o no li han agradat, mentre que una majoria no apunta res.

Has realitzat mai cap bloc de viatges?

47 respostes

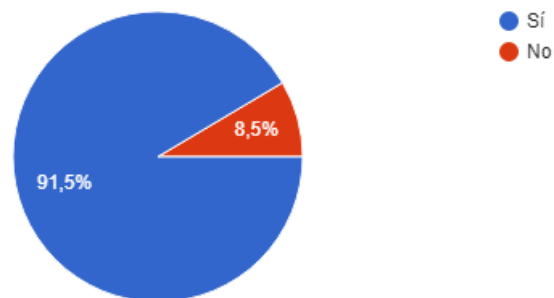


II-lustració 20 Divisió segons si han fet mai un bloc de viatges

Un 10% de les persones enquestades han realitzat blocs de viatge alguna vegada. Mentre que quasi un 90% no n'ha fet mai cap.

Comparteixes amb persones de la teva confiança experiències viscudes durant un viatge?

47 respostes

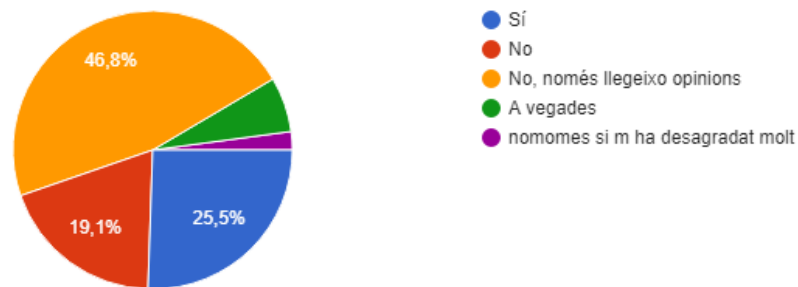


II-lustració 21 Divisió usuaris si han compartit experiències viscudes durant un viatge amb persones de confiança

Un 91% de les persones enquestades comparteix el viatge amb familiars, amics, persones de confiança.

Acostumes a donar la teva opinió a internet de llocs visitats, restaurants..?

47 respostes

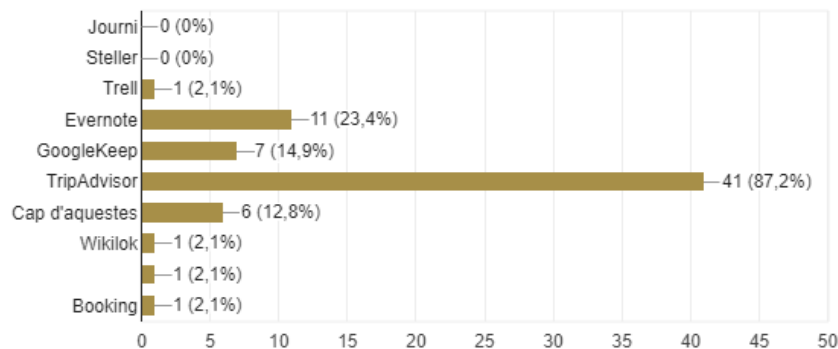


II-lustració 22 Divisió d'usuaris segons si donen o no opinió a internet

Hi ha un 46,8% de les persones enquestades que no han donat mai la opinió a internet, però sí que han llegit opinions. Un 25,5% de les persones sí que han donat la seva opinió i un 19,1% no l'han donat ni han llegit mai opinions d'internet.

Utilitzes o has utilitzat alguna d'aquestes aplicacions?

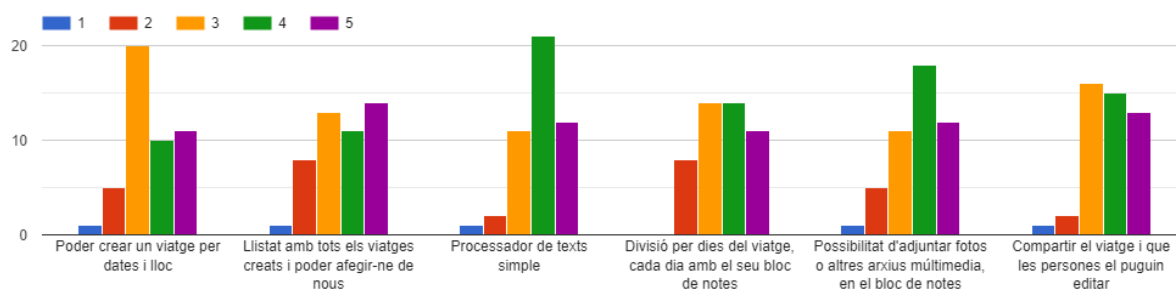
47 respostes



II-lustració 23 Coneixement de les webs competència dels usuaris

De les diferents aplicacions que es van estudiar de la competència, o que tindrien una funció similar a la realitzada en aquest projecte, la que més coneix la gent és TripAdvisor, que serveix per donar opinió de quasi tots els restaurants i llocs del món, i després les altres dues són blocs de notes més tradicionals com són Evernote i GoogleKeep. Aplicacions específiques de blocs de viatge, com Trell, Journi o Steller no les coneixen.

Valoració 1 al 5 funcionalitats útils d'una aplicació per a realitzar un bloc d'un viatge



II-lustració 24 Valoració de les funcionalitats proposades

A la última pregunta sobre funcionalitats, s'han obtingut els següents resultats:

A la pregunta de poder crear un viatge per dates i lloc, ha estat valorada majoritàriament amb un 3 sobre 5.

Pel que fa a poder tenir un llistat amb tots els viatges i crear i afegir-ne de nous, ha obtingut bons resultats, és la única en que la nota de 5 sobre 5 és la més alta.

La funcionalitat de tenir un processador de texts senzill ha obtingut un resultat de 4 sobre 5.

Poder dividir el viatge per dies i que cada un tingui el seu propi bloc de notes, ha obtingut un resultat molt similars de 3 i 4. També ha sigut la pregunta on la valoració amb un 2 ha estat la més alta.

La funcionalitat de poder adjuntar fotos o altres arxius multimèdia ha estat valorada amb un 4 sobre 5.

I la última pregunta sobre compartir el viatge amb persones i que aquestes el puguin editar, també ha estat valorada molt positivament.

2.1.1.4 Conclusions enquestes

Un cop realitzades enquestes, podem concloure que les persones naveguen per internet amb el mòbil entre 3 i 4 hores diàries. Per tant és un ús de mòbil molt elevat tenint en compte que l'horari laboral és de 8 hores i 8 hores són per descansar.

Durant els viatges, la majoria de persones no han escrit mai cap bloc de viatges però sí que algunes s'han apuntat llocs i experiències que han tingut. I fan servir les aplicacions que ofereixen opinions de llocs per informar-se, més que no pas donar la seva pròpia opinió.

Pel que fa a les funcionalitats enquestades, les persones han puntuat molt positivament totes les funcionalitats, tot i que veient la resta de preguntes semblaria que els resultats haurien hagut d'anar cap a una altra direcció. Les funcionalitats de crear un viatge, poder llistar-los, adjuntar-hi arxius multimèdia, han tingut bones puntuacions, i son les principals funcionalitats que es voldran implementar en aquest projecte.

Cal tenir en compte que les persones enquestades eren conscients de l'enquesta que estaven responent, sabien qui la feia i que era del projecte de final de màster, i això podria haver influenciat en els resultats.

Nogensmenys, el fet d'haver obtingut aquesta bona puntuació, semblaria que l'aplicació podria tenir un cert èxit.

2.1.1.5 Perfils d'usuari

A continuació es troben els dos perfils d'usuari identificats, gràcies a la recopilació de dades dels apartats anteriors.

	Usuari viatges curts
Característiques del perfil	<p>Perfil d'usuari, jove o adult, que està molt acostumat a utilitzar el mòbil i aplicacions de tot tipus. Coneix aplicacions com TripAdvisor, o blocs de notes i li agrada molt viatjar. Acostuma a realitzar entre 2 i 3 viatges cada any per motius d'oci o de feina, però solen ser de 1 a 3 dies, i sense anar a grans distàncies. Si son per motius de feina, podria fer-ne mes de 5 cada any.</p> <p>Normalment utilitza les aplicacions de viatges per a llegir opinions sobre restaurants o llocs a visitar, i no donar la pròpia, però si viu experiències que li agraden, s'ho apunta, de manera rapida perquè no té molt de temps.</p>
Contexts d'ús	<p>Quan aquest usuari farà ús de l'aplicació quan busqui prendre nota d'una experiència positiva o negativa, sense haver de fer un gran escrit, i que no suposi un gran esforç. El que busca és la immediatesa de tenir un bloc de notes, i apuntar. I que quedi guarda't i classificat de forma simple.</p>
Anàlisi de tasques	<ul style="list-style-type: none"> - Obrir l'aplicació. - Crear viatge, o editar si ja ha estat creat abans. - Crear un dia, o aprofitar algun ja creat. - Escriure el que necessiti. - Adjuntar-hi una foto, si creu que l'ajudarà.
Característiques o elements presents a la interfície	<ul style="list-style-type: none"> - Poder trobar el botó de crear un viatge fàcilment. - Poder veure la llista de viatges creats. - Permetre editar un viatge existent. - Botó de crear dia fàcilment visible.

	<ul style="list-style-type: none"> - Bloc de notes intuïtiu. - Botó per adjuntar fotos visible.
	Usuari viatges llargs
Característiques del perfil	<p>Perfil d'usuari, jove o adult, que està acostumat a utilitzar el mòbil i aplicacions de tot tipus. Coneix aplicacions com TripAdvisor, o blocs de notes i li agrada molt viatjar, i la majoria de viatges són d'oci.</p> <p>Aquest usuari acostuma a fer 1 viatge a l'any, i de llarga durada, en dies i també lluny de casa.</p> <p>És una persona que viatge amb temps, i calma, i li agrada apuntar-se tot el que viu per a poder-ho recordar bé, juntament amb les fotografies.</p> <p>És un perfil que ha realitzat blocs de viatges, i ha donat opinions a internet sobre restaurants, o llocs.</p>
Contexts d'ús	<p>Quan l'usuari estigui de viatge, i s'estigui desplaçant amb algun vehicle d'un lloc a un altre, o ja hagi arribat al hotel, o estigui en un restaurant i no tingui pressa, farà ús del mòbil per anar-se anotant, i adjuntant fotografies de tot el que ha anat veient en aquells moments. Voldrà guardar el viatge a la meitat, i acabar-lo poc a poc cada dia.</p>
Anàlisi de tasques	<ul style="list-style-type: none"> - Obrir l'aplicació. - Crear viatge, o editar si ja ha estat creat abans. - Si està creant, ús de la cerca de llocs de GoogleMaps. - Posarà les dates i dades de viatge completes. - Crear un dia, o editarà el que correspongui. - Escriure el que necessiti. - Adjuntar fotos.
Característiques o elements presents a la interfície	<ul style="list-style-type: none"> - Poder veure la llista de viatges creats. - Botó de crear un viatge. - Mapa per poder buscar el lloc pel viatge. - Calendari per poder posar les dates. - Permetre editar un viatge existent. - Botó de crear dia. - Bloc de notes intuïtiu. - Botó per adjuntar fotos.

2.1.2 Disseny Conceptual

Escenaris d'ús

En aquest apartat trobem els diferents escenaris d'ús identificats per a l'ús de la nostra aplicació:

Escenari 1:

Perfil	Usuària viatges curts
Context	<p>La Maria es troba fent un viatge per feina a París, on només hi pot passar una nit. Ja que té una reunió tot el dia dilluns, i l'avió de tornada li marxa el dimarts a les 6 del matí per arribar a Barcelona a temps per entrar a treballar cap a les 9.</p> <p>Parlant amb les companyes de feina de París li expliquen varis llocs on pot anar a menjar o visitar, però la Maria no té temps per fer-ho aquests dies, i s'ho apunta per tornar-hi de vacances amb la seva dona el proper estiu.</p>
Objectius	Apuntar-se varis restaurants per visitar a París.
Tasques realitzades per assolir els objectius	<ul style="list-style-type: none">- Crear un viatge que es digui París, sense dates, i marcar-lo com a pendent.- Buscar el lloc del viatge.- Afegir un dia.- Apuntar-hi el restaurant- Adjuntar la ubicació buscant l'adreça en el mapa.
Necessitats d'informació	<ul style="list-style-type: none">- Crear un viatge.- Buscar ciutats en el mapa de creació de viatges.- Adjuntar la ubicació del restaurant amb el mapa.- Escriure en el bloc de notes.
Com es desenvolupen les tasques	<ul style="list-style-type: none">- Obrir l'aplicació.- Crear viatge que posi París- Clicar continuar.- Escriure el nom del restaurant en el bloc de notes.- Clicar botó per adjuntar.- Escollir adjuntar ubicació.- Buscar-hi el restaurant.- Adjuntar-lo.

Escenari 2:

Perfil	Usuària viatges curts
Context	La Laila ha anat a Porto amb les companyes de la universitat per a visitar una companya que està allà d'Erasmus. Mentre estan al autobús anant cap a un parc de les afores de la ciutat, està recordant el que han vist durant el matí, i es vol apuntar el nom dels monuments que han vist per a poder buscar informació sobre aquests un cop arribi a casa.
Objectius	Apuntar-se els noms dels diferents monuments visitats durant el seu viatge.
Tasques realitzades per assolir els objectius	<ul style="list-style-type: none">- Editar el viatge que ja té creat que posa Porto.- Crear un dia nou, ja que es troba en el segon dia del viatge.- En el bloc de notes, afegir la data del dia.- Escriure els noms del monuments.- Adjuntar-hi algunes de les fotos que ha fet.
Necessitats d'informació	<ul style="list-style-type: none">- Editar un viatge.- Crear un dia nou.- Escriure en el bloc de notes.- Adjuntar imatges.
Com es desenvolupen les tasques	<ul style="list-style-type: none">- Obrir l'aplicació.- Buscar a la llista el nom de Porto.- Clicar botó per editar.- Clicar botó per afegir un dia nou.- Posar-hi la data.- Escriure al bloc de notes.- Clicar botó d'adjuntar.- Escollir adjuntar des de la galeria.- Seleccionar imatges desitjades.

Escenari 3:

Perfil	Usuari viatges llargs
Context	En Karim està de viatge al Perú amb un grup d'amics. I com cada any que fa un viatge llarg, vol escriure tot el que viu per poder-ho explicar a la família amb pels i senyals quan arribi a Catalunya un altre cop. De moment aprofita que està a l'avió per crear el primer dia, i escriure el que han anat fent, fins ara.
Objectius	Apuntar-se tot el que viu durant el viatge al Perú. Ja siguin llocs que han visitat, llocs que han menjat, o trobades amb la gent local.
Tasques realitzades per assolir els objectius	<ul style="list-style-type: none">- Crear un viatge nou a la llista de títol Perú.- Afegir les dates del viatge.- Buscar en el mapa el país.- Clicar botó continuar.- Afegir un dia nou.- Escriure el que ha fet avui.
Necessitats d'informació	<ul style="list-style-type: none">- Crear un viatge.- Escriure en el bloc de notes.
Com es desenvolupen les tasques	<ul style="list-style-type: none">- Obrir l'aplicació.- Crear viatge que posi Perú- Escriure el que ha viscut durant aquest viatge.

Escenari 4:

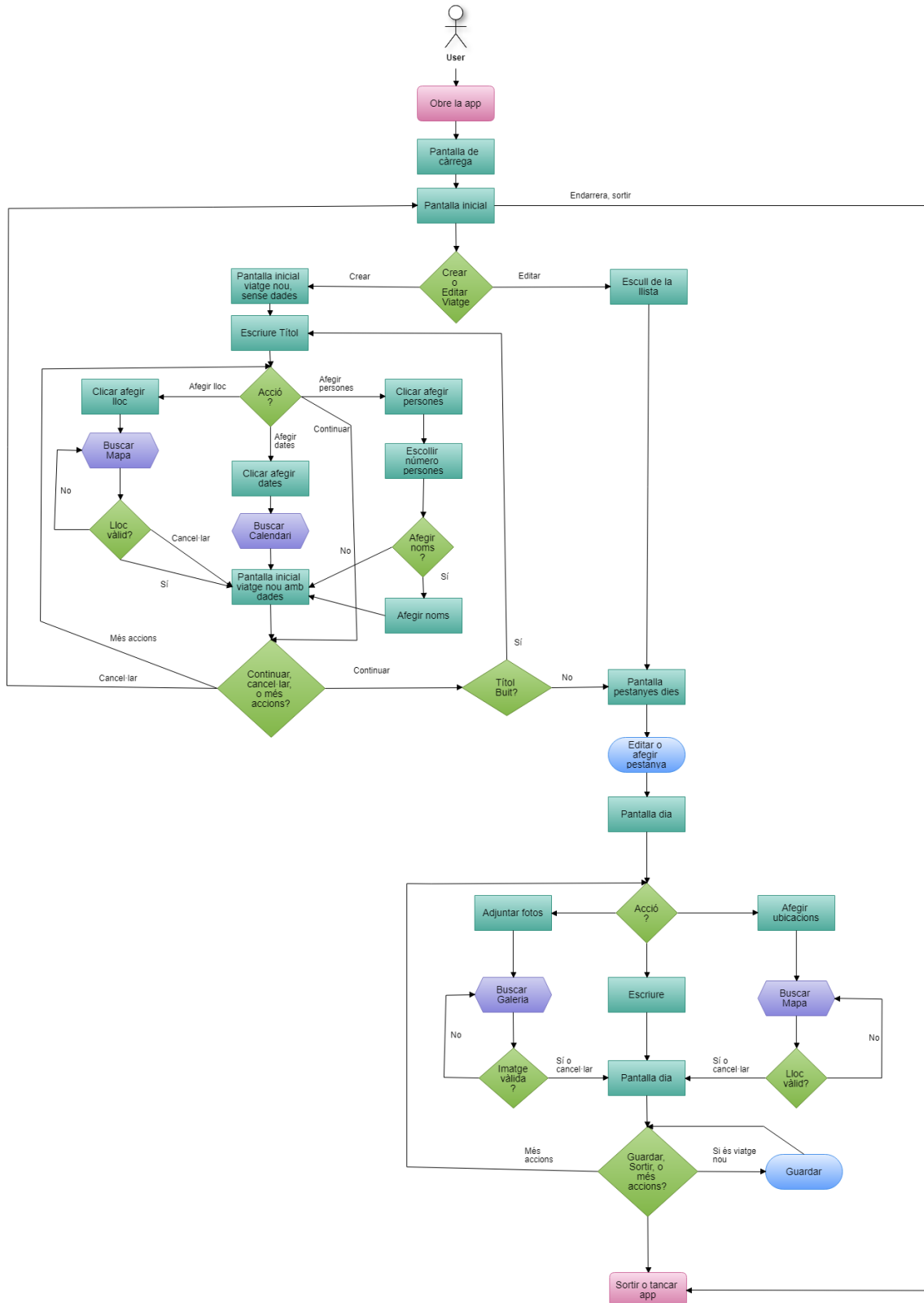
Perfil	Usuària viatges llargs
Context	La Martina està preparant el viatge per anar al Japó el proper estiu, és el primer cop que realitza un viatge sola i lluny de casa, i se'l vol preparar molt bé, deixant poques coses a l'atzar i poder planificar per veure tot el que vol.
Objectius	Crear un bloc de viatges nous, com a pendent de fer, i apuntar-se cada dia els diferents llocs que visitarà i llocs on dormirà.
Tasques realitzades per assolir els objectius	<ul style="list-style-type: none"> - Crear un viatge de nom Japó. - Afegir les dates del viatge. - Crear tants dies com duri el viatge. - En el primer dia, apunta el nom de la ciutat. - Busca en el mapa la ciutat. - S'apunta en el bloc de notes, les coses que farà el primer dia. - S'apunta el nom i adreça de l'hotel. - Adjunta direcció de l'hotel en el mapa.
Necessitats d'informació	<ul style="list-style-type: none"> - Crear un viatge nou - Afegir dates - Crear dies. - Escriure en el bloc de notes. - Adjuntar ubicació.
Com es desenvolupen les tasques	<ul style="list-style-type: none"> - Obrir l'aplicació. - Clicar botó crear viatge nou. - Posar nom viatge. - Afegir dates del viatge. - Clicar botó continuar. - Crear el primer dia - Escriure el nom de la ciutat. - Clicar botó adjuntar. - Escollir adjuntar ubicació. - Cercar la ciutat desitjada. - Clicar adjuntar. - Escriure els noms dels llocs que vol visitar aquell dia. - Escriure el nom del hotel. - Afegir enllaç de la reserva de Booking. - Escollir adjuntar ubicació. - Cercar l'adreça desitjada. - Clicar adjuntar.

Escenari 5:

Perfil	Usuàries viatges curts i llargs
Context	La Maria i la Júlia han anat de viatge a París perquè la Maria i havia estat per feina, i no havia pogut visitar la ciutat. Ja que tenia apuntats a l'aplicació el nom d'algun restaurant, recomanat per companyes de feina i és l'hora de dinar. Busca el nom del restaurant a l'aplicació, i gràcies a que l'havia apuntat adjuntant-hi la ubicació pot obrir fàcilment el GoogleMaps i iniciar la navegació.
Objectius	Buscar en un viatge creat fa temps, un nom d'un restaurant apuntat i la seva ubicació.
Tasques realitzades per assolir els objectius	<ul style="list-style-type: none"> - Editar un viatge creat. - Moure's per el bloc de notes per trobar el nom del restaurant. - Clicar sobre el mapa. - Clicar obrir a GoogleMaps. - Iniciar Navegació.
Necessitats d'informació	<ul style="list-style-type: none"> - Ordenar llistes. - Editar viatge. - Moure's pel bloc de notes. - Obrir ubicacions.
Com es desenvolupen les tasques	<ul style="list-style-type: none"> - Obrir aplicació. - Ordenar la llista per viatges "pendents". - Buscar París. - Clicar editar viatge. - Buscar en el bloc de notes el restaurant. - Clicar ubicació. - Clicar inicia navegació. - Escollir Google Maps.

2.1.2.1 Fluxos d'interacció

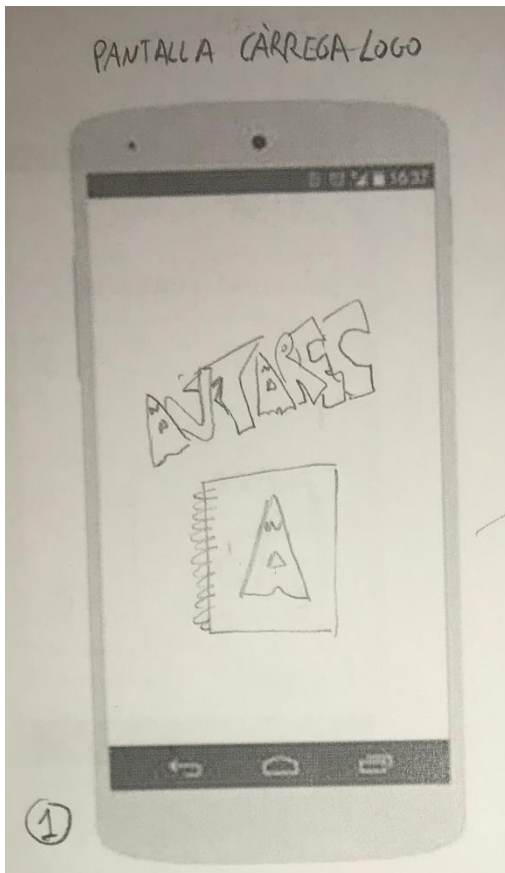
Els flux d'interacció serveix per poder veure com les persones usuàries interactuen amb la nostra aplicació i els camins que segueixen en funció el seu objectiu.



II-lustració 25 Diagrama de flux

2.1.3 Prototipatge

2.1.3.1 Esbossos



Il·lustració 26 Esbós pantalla càrrega

Primera pantalla que veurà l'usuari que serà de càrrega i tindrà el logo de l'aplicació.

Número 1.

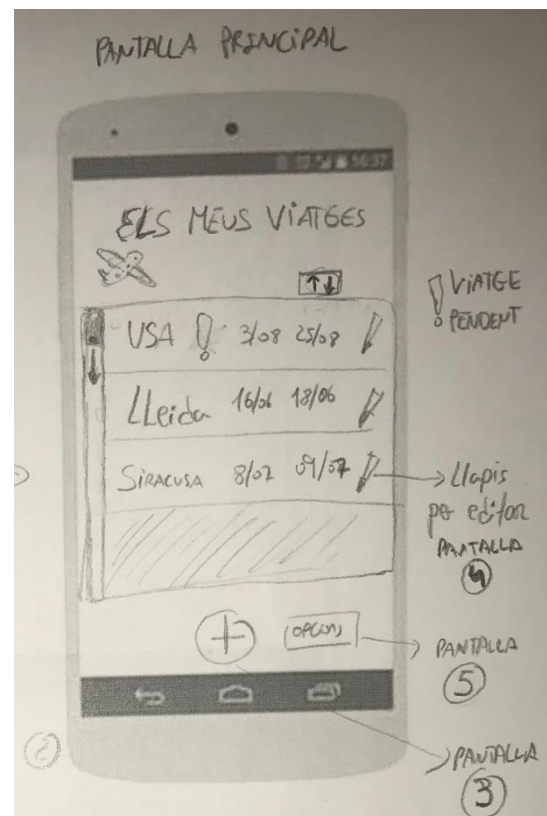
Pantalla principal, amb la llista de viatges, i els botons per ordenar, crear viatge, o el d'opcions per esborrar o compartir un viatge seleccionat.

Número 2.

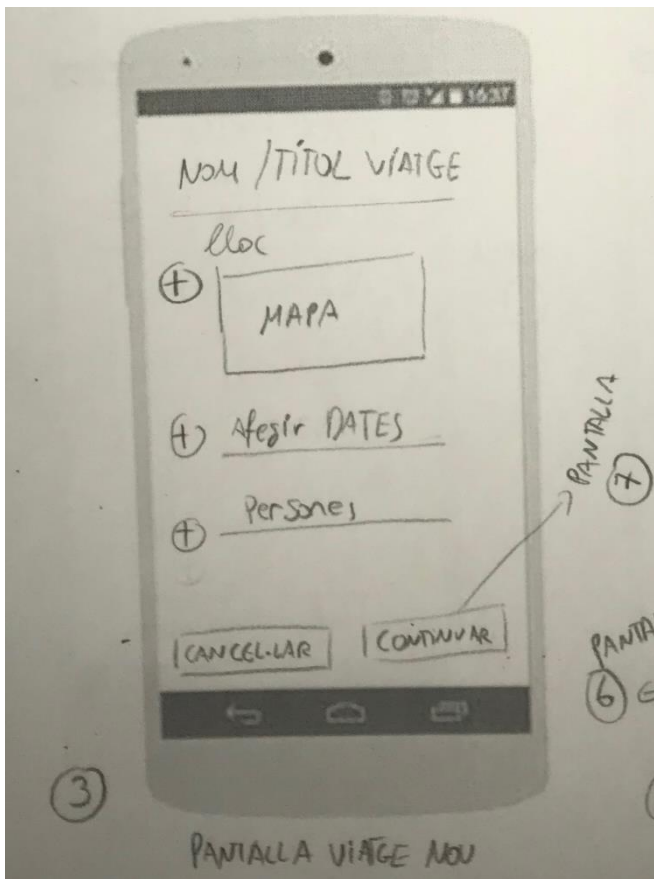
Si es clica **crear viatge** passa al número 3.

Si es clica **editar** passa al número 4.

Si es clica **opcions** passa al número 5.



Il·lustració 27 Esbós pantalla principal plena



II-l·lustració 28 Esbós pantalla viatge nou buida

Primera pantalla de creació de viatges.
Es pot afegir el títol, el lloc i la ubicació si es vol.
També les dates que es viatge.
I les persones que han viatjat.

Número 3.

Si es clica **continuar** passa al número 7.
Si es clica **cancel·lar** passa al número 2.

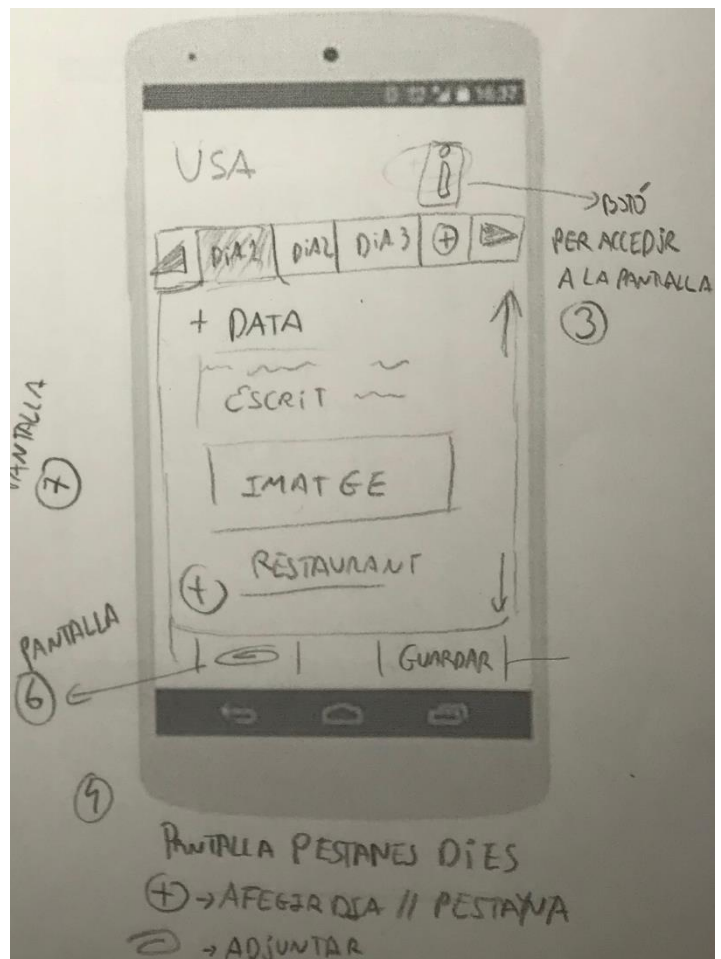
Pantalla per editar un viatge, es pot veure la informació de cada dia, els escrits, dates, imatges. Es pot canviar de dia clicant a la pestanya, del dia que toca.

Número 4.

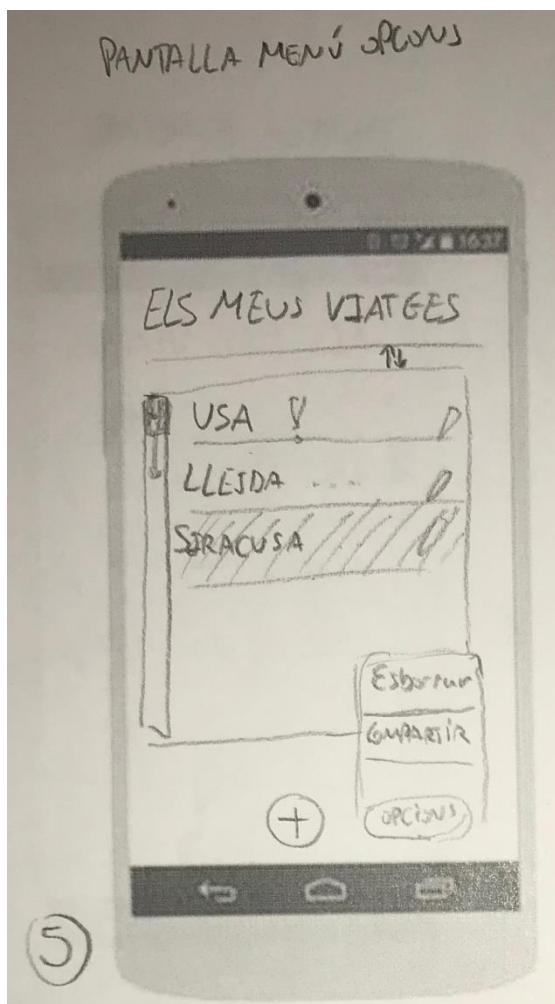
Si es clica **Informació** passa al número 3.

Si es clica **Guardar** es queda al mateix lloc, però s'ha guardat.

Si es clica **adjuntar** passa al número 6.



II-l·lustració 29 Esbós pantalla pestanyes dies



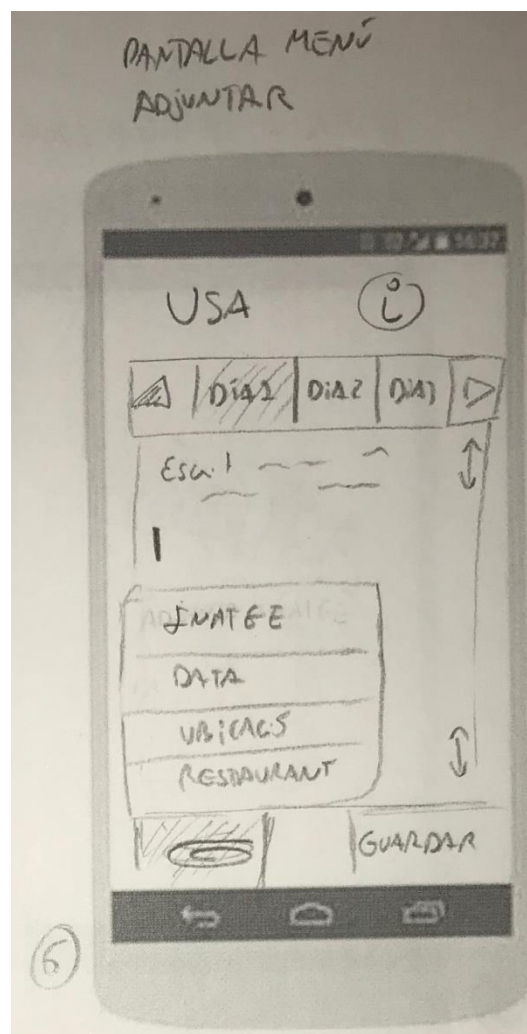
Il·lustració 30 Esbós pantalla principal menú opcions

Pantalla principal de la llista de viatges, amb el menú de opcions desplegat. Només serveix si hi ha un viatge seleccionat.

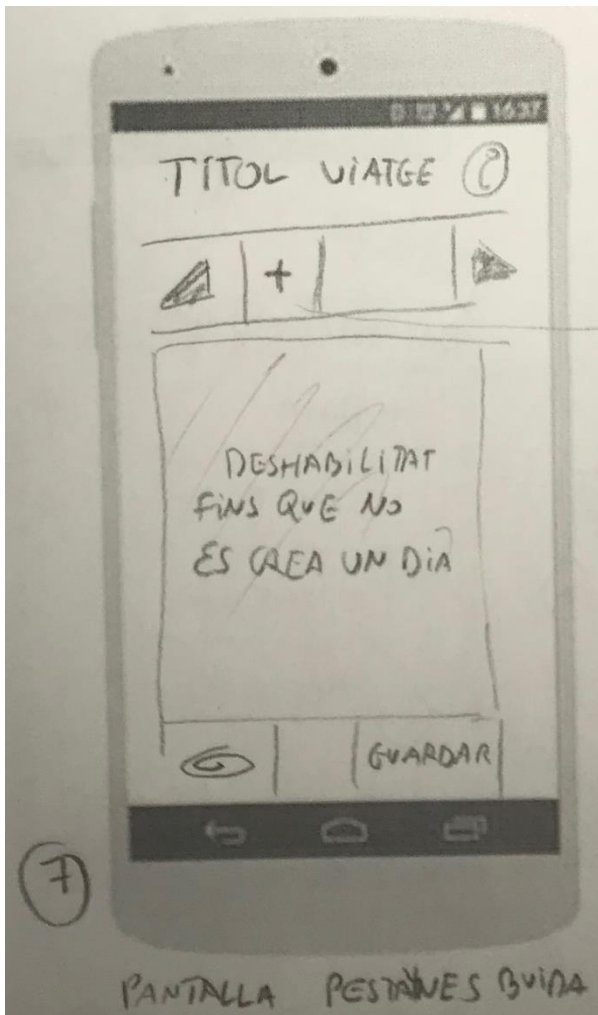
Número 5.

Pantalla de pestanyes amb la primera pestanya seleccionada, i el menú per adjuntar seleccionat.

Número 6.



Il·lustració 31 Esbós pantalla pestanyes menú adjuntar



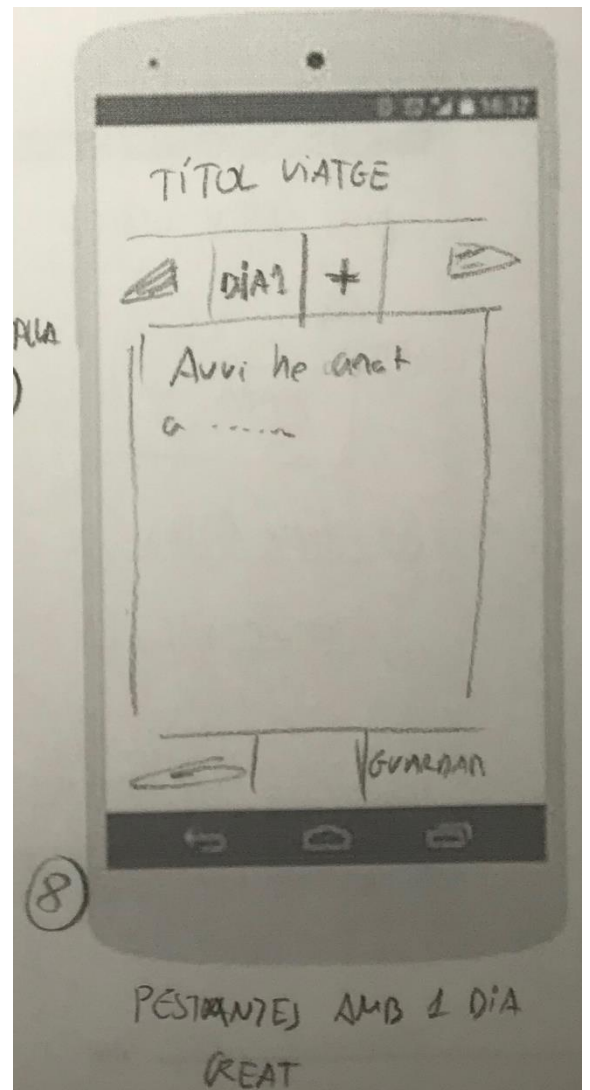
Il·lustració 32 Esbós pantalla pestanyes buida

Pantalla de pestanyes buida.

En aquest cas no hi ha cap dia creat, i per tant no es pot escriure ni afegir res, fins que no hi ha 1 dia mínim nou.

Si que es permet guardar.

Número 7



Il·lustració 33 Esbós pantalla pestanyes amb 1 dia

Pantalla de les pestanyes havent creat només el primer dia.

Es pot veure que només hi ha text, i també hi ha el botó per afegir noves pestanyes.

Número 8.

2.1.3.2 Prototipus alta fidelitat

El prototipus d'alta fidelitat que veiem en aquest apartat, s'ha realitzat mitjançant el software JUSTINMIND⁸.

Les dues primeres imatges són de la pantalla de càrrega i de la llista de viatges.

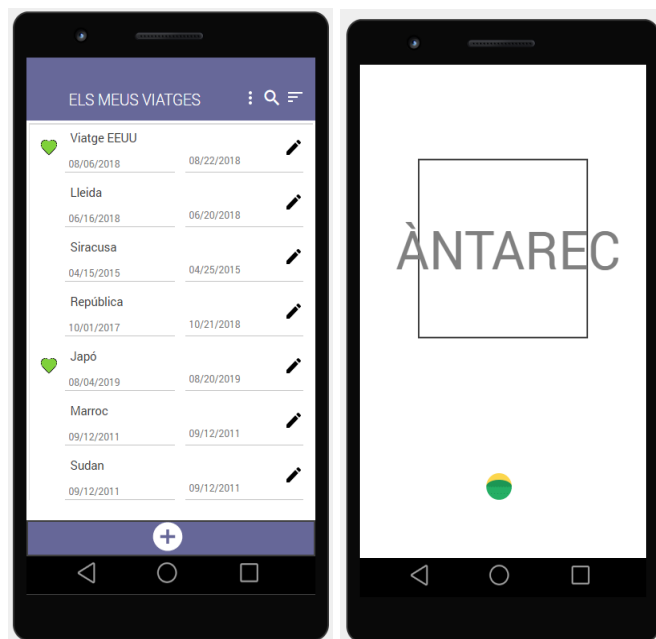
Podem veure els botó per crear viatges nous.

En el menú superior tenim els botons d'opcions, cerca dins la llista, i ordena la llista.

En els elements de la llista podem veure el títol del viatge, com també les dates en què es va fer, o es vol fer el viatge.

La icona amb el llapis permet editar el viatge.

La icona en forma de cor serveixen per indicar que el viatge està marcat com a pendent.



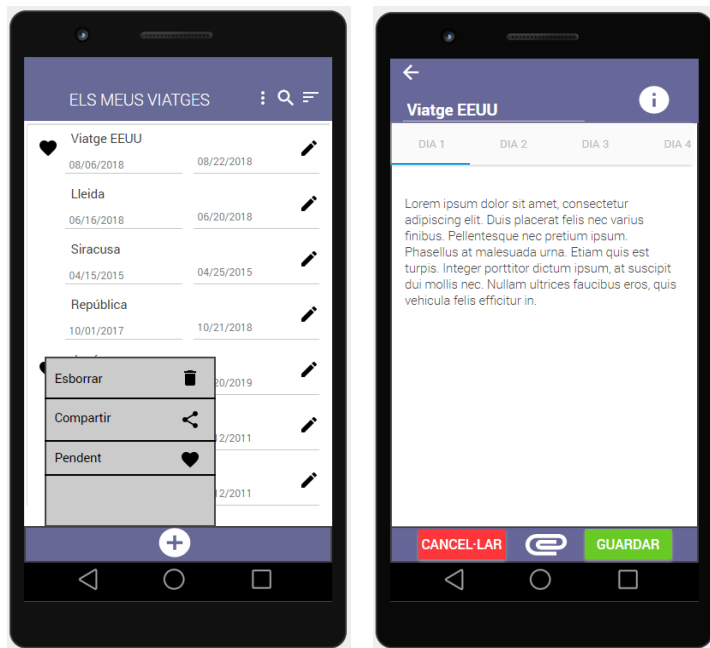
II-Iustració 34 Prototipus alta fidelitat I

En les pantalles següents podem veure les opcions desplegadas, on permet esborrar i compartir el viatge, que tenim seleccionat (a la captura de pantalla no es veu el viatge seleccionat).

La funcionalitat de compartir el viatge, no s'ha pogut fer perquè hauria sigut una tasca molt complicada pel poc temps que tenim per desenvolupar l'aplicació.

També hi ha la pantalla amb el bloc de notes del primer viatge. I es veuen els botons per guardar, cancel·lar o guardar en el menú superior. I en el menú superior tenim el botó per veure les dades de capçalera del viatge.

⁸ <https://www.justinmind.com/>



II-lustració 35 Prototipus alta fidelitat II

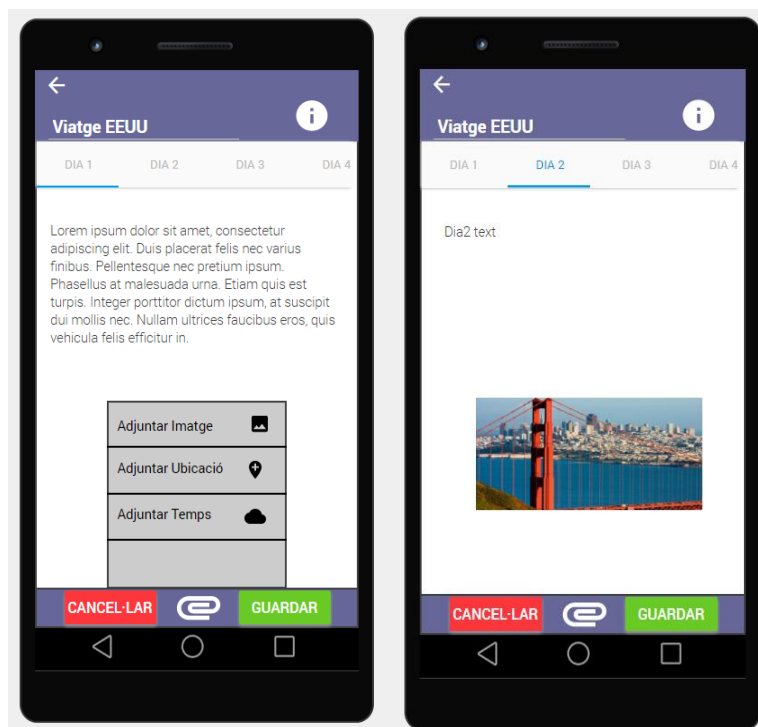
Les següents pantalles són les del menú per adjuntar desplegat, i el text del dia dos.

Veiem que ens permet adjuntar una imatge, la qual ens portaria a la nostra galeria del telèfon mòbil.

Podem adjuntar una ubicació, per si volem guardar un lloc en concret en algun punt.

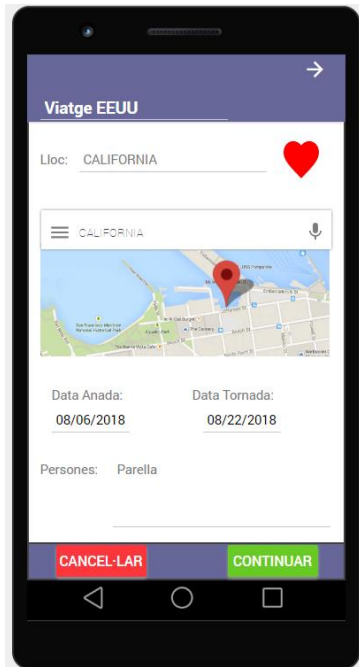
I permet adjuntar el temps que està fent en aquell moment.

Es va decidir que enlloc de tenir una pestanya amb un símbol per afegir-ne de noves, n'hi haurà tantes com dies duri el viatge, segons les dates de capçalera.



II-lustració 36 Prototipus alta fidelitat III

La següent pantalla és la del detall del viatge.
En aquest cas hem polsat la icona en forma de (i) del viatge de estats units, i ens ha portat a les dates i informació que es van escriure quan es va crear el viatge. Podem veure que hi surt el cor, perquè està marcat com a pendent, sinó no hi apareix.



**Il·lustració 37 Prototipus
alta fidelitat IV**

Les dues imatges següents corresponen en el procés de creació d'un viatge nou.

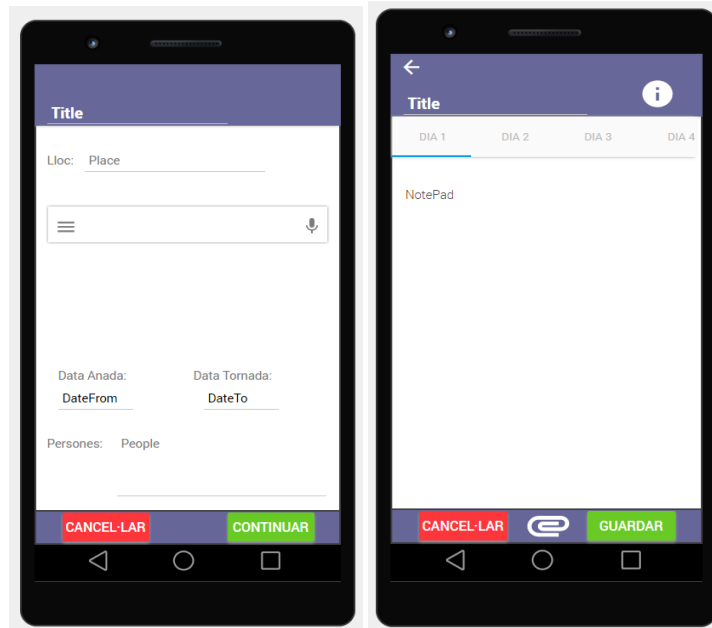
Primer veiem la pantalla de capçalera buida, per tal d'introduir les dades necessàries.

Podem posar-hi el títol que serà el que veurem en el llistat, el lloc que volem, hi ha una barra per cercar la localització. També podem afegir les dates d'anada i tornada, i un camp de text per explicar les persones que ens han acompanyat.

Un cop tenim les dades de capçalera plenes, anem al bloc de notes per tal de poder afegir informació del nostre viatge.

Si necessitem fer canvis a la capçalera, hi podem tornar amb el botó del menú superior.

Els dos botons de cancel·lar en les dues pantalles porten cap a la pantalla inicial de llista de viatges.



II-Il·lustració 38 Prototipus alta fidelitat V

2.1.4 Avaluació

El mètode d'avaluació amb usuaris ens permet testejar les interfícies dissenyades, per conèixer debilitats i punts forts d'aquestes, i poder així millorar-les en la següent iteració.

Els perfils d'usuaris escollits seran persones entre 25 i 60 anys, que estiguin acostumades a fer un ús del mòbil diari.

També es buscarà persones que acostumin a viatjar, indiferentment del número de vegades que ho fan per any, i el motiu d'aquests. Tot i que aquesta informació es recollirà en el qüestionari previ.

Les proves es realitzaran en una sala neutre, on hi hagi la persona enquestada i la persona que guia el test.

La prova es realitzarà utilitzant el prototipus d'alta fidelitat dissenyat, on es pot simular l'ús de botons, i navegació entre pantalles.

La persona enquestada haurà d'anar dient en veu alta els seus pensaments mentre està utilitzant el prototipus. I la persona encarregada de guiar, li haurà d'anar indicant quines tasques ha de realitzar.

Qüestionari previ al test

- a. Edat?
- b. Ús del mòbil diari? I lloc a on acostuma a utilitzar-lo.
- c. Viatges que acostuma a fer en un any? I motiu principal?
- d. Utilitza aplicacions per a gestionar viatges?
- e. Utilitza aplicacions per a donar opinió o llegir opinions de llocs?
- f. Ha realitzat mai cap bloc de viatges?

Tasques a avaluar

Les tasques que es demanaran als usuaris que facin seran les següents:

Tasca 1: Crear un viatge nou, a Oslo, de 3 dies, les dates que vulguin, amb 3 persones, i que no hi posi la ubicació.

Tasca 2: En el primer dia hi posi el text: "Primer dia de viatge.". I també hi adjunti una imatge d'un fiord que hi ha a la galeria.

Tasca 3: Cancel·lar el viatge sense guardar-lo.

Tasca 4: Crear un altre viatge a Nador. De quatre dies, les dates que vulgui. Amb la ubicació i sense persones.

Tasca 5:

- En el primer dia que hi escrigui: "Estic molt content/x d'haver arribat a Nador".
- En el segon dia que hi adjunti una imatge i escrigui: "Aquest matí he anat a veure les haines.
- El tercer dia que adjunti la ubicació de Fes i escrigui: "Avui he anat a passar el dia a Fes, he menjat una harira boníssima".
- En el quart dia hi adjunti el temps, juntament amb el text: "L'últim dia de viatge feia molt sol".

Tasca 6: Guardar el viatge i tornar a la llista.

Tasca 7: Buscar un viatge marcat com a pendent. Seleccionar-lo i esborrar-lo.

Tasca 8: Ordenar la llista per noms ascendents i buscar utilitzant el botó del menú, el viatge a Zurich.

Tasca 9: Un cop trobat el viatge a Zurich, editar la data d'anada per la d'un dia menys. Guardar el canvi i tornar a la llista.

Qüestionari post test

Un cop acabat el test, i amb la persona examinadora fora de la sala, es passarà el següent qüestionari perquè ens responguin les persones enquestades.

Obtenint així informació directe de les interfícies i funcionalitats (simulades) de l'aplicació, que ens servirà per a poder millorar-la, o no.

- Ha sigut senzill utilitzar l'aplicació?
- Has pogut realitzar totes les tasques que se t'han demanat?
- Quina tasca has trobat que t'ha costat més? Perquè?
- Hi ha algun element de les pantalles que t'hagi molestat, o desagradat? Perquè?
- Hi afegiries alguna cosa?
- Creus que utilitzaries aquesta aplicació en algun dels teus viatges?

- Tenint en compte el format dels viatges, si hi hagués la possibilitat de compartir-lo entre varies persones, i que totes elles el poguessin editar, ho faries servir? Perquè sí o perquè no?

2.1.5 Iteracions en el disseny

Un cop avaluat el disseny per part dels professors d'aquest treball i analitzant el feedback rebut, s'han fet una sèrie de modificacions en algunes de les pantalles, es poden veure els canvis a la part inferior:

Pantalla llista viatges:

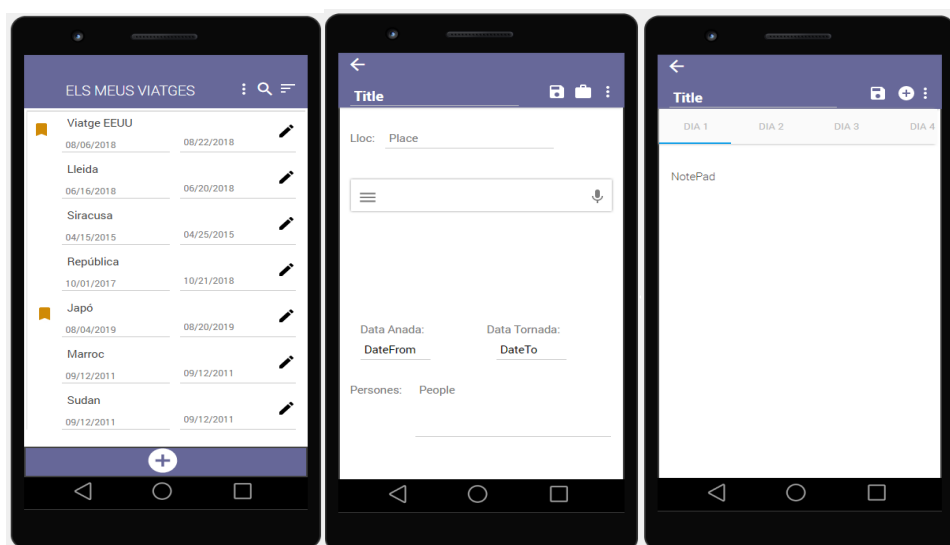
- S'ha substituït la icona que servia per marcar els viatges pendents. Inicialment era un cor i finalment si ha deixat una icona de *bookmark*. Aquest canvi s'ha fet per evitar confusions ja que normalment el cor és utilitzat per a marcar els preferits.

Pantalla crear Viatge:

- A la pantalla de crear viatge s'han tret els botons inferiors i s'ha posat a la barra superior un botó per guardar el viatge, i un botó per navegar cap als dies del viatge. No hi ha botó cancel·lar ja que es fa amb la mateixa acció d'anar endarrere.

Pantalla dies viatge:

- Els botons inferiors s'han tret, i s'han posat opcions a la barra superior.
- Les opcions per adjuntar imatges o ubicacions, s'han posat en el menú superior desplegable.
- Per poder tornar a la pantalla Header, enlloc de tenir un boto de Informació hi ha una opció en el menú que permet tornar endarrere.



II-lustració 39 Prototipus alta fidelitat després iteració

2.2 Disseny Tècnic

2.2.1 Casos d'ús

Identificador	CU-001
Nom	Crear viatge nou
Prioritat	Alta
Descripció	La persona usuària vol crear un viatge nou
Actors	Persona Usuària
Pre-condicions	La persona vol crear un viatge i disposa de un dispositiu mòbil amb la app.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none">1. Iniciar aplicació2. Esperar que es carregui l'aplicació3. Clicar botó de crear viatge4. Omplir dades de capçalera de viatge5. Omplir Títol obligatori6. Modificar dates que per defecte surten amb el dia actual.7. Omplir altres dades opcionals de lloc, persones.<ol style="list-style-type: none">7.1 Si es vol cercar una ubicació CU-0068. Clicar continuar9. Omplir dades en els dies desitjats<ol style="list-style-type: none">9.1 Si es vol adjuntar una ubicació CU-0059.2 Si es vol adjuntar una imatge CU-00410. Clicar Guardar
	Tornar a la llista i veure el nou viatge
Notes	

Identificador	CU-002
Nom	Modificar viatge existent
Prioritat	Alta
Descripció	La persona usuària vol modificar un viatge existent
Actors	Persona Usuària
Pre-condicions	La persona vol modificar un viatge i disposa de un dispositiu mòbil amb la app, amb viatges creats.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none"> 1. Iniciar aplicació 2. Esperar que es carregui l'aplicació 3. Buscar a la llista el viatge que vol modificar 4. Clicar icona de modificar 5. Modificar el viatge. 6. Clicar guardar
Post-condicions	Tornant a entrar al viatge des de la llista, es poden veure els canvis fets.
Notes	

Identificador	CU-003
Nom	Esborrar viatge existent
Prioritat	Alta
Descripció	La persona usuària vol esborrar un viatge existent
Actors	Persona Usuària
Pre-condicions	La persona vol esborrar un viatge i disposa de un dispositiu mòbil amb la aplicació, amb viatges creats.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none"> 1. Iniciar aplicació. 2. Esperar que es carregui l'aplicació 3. Seleccionar el viatge que es vol esborrar. 4. Clicar el botó de opcions 5. Seleccionar esborrar 6. Confirmar que es vol esborrar el viatge.
Post-condicions	Veure que el viatge esborrat, ha desaparegut de la llista.
Notes	

Identificador	CU-004
Nom	Adjuntar imatge
Prioritat	Normal
Descripció	La persona usuària vol adjuntar cercar i adjuntar una imatge, en un dels dies que componen un viatge.
Actors	Usuari/a
Pre-condicions	Venir de CU-001 o CU-002. Que l'usuari/a disposi d'un dispositiu mòbil amb càmera fotogràfica o amb imatges a la galeria.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none"> 1. Clicar sobre el botó d'adjuntar 2. Seleccionar adjuntar imatge 3. Escollir de on es vol treure la imatge. <ol style="list-style-type: none"> 3.1 Si es vol seleccionar imatge de la galeria. 3.2 Si es vol fer una foto en aquell instant.
Post-condicions	Continuar amb el flux del qual es venia un cop adjuntada la imatge.
Notes	Ús opcional de l'aplicació.

Identificador	CU-005
Nom	Cercar Ubicació
Prioritat	Normal
Descripció	La persona usuària està creant o modificant les dades de capçalera d'un viatge, i vol cercar la ubicació
Actors	Usuari/a
Pre-condicions	Venir de CU-001 o CU-002. Disposar d'un mòbil amb accés a GPS i internet.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none"> 1. Cercar en el quadre de text la localització geografia que es vol posar a la capçalera del viatge. 2. Continuar amb el flux que es venia-
Post-condicions	Quan l'usuari accedeix a la capçalera, pot veure la localització cercada anteriorment.
Notes	Ús opcional de l'aplicació.
Identificador	CU-006

Nom	Adjuntar ubicació
Prioritat	Normal
Descripció	La persona usuària vol adjuntar una ubicació en un dels dies del viatge.
Actors	Persona Usuària
Pre-condicions	La persona disposa de un dispositiu mòbil amb l'aplicació, té accés a internet i al GPS. Venir de CU-001 o CU-002.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none"> 1. Clicar botó adjuntar. 2. Seleccionar adjuntar ubicació. 3. Donar permisos d'accés al GPS. 4. Cercar la ubicació que es vol. 5. Acceptar un cop trobada.
Post-condicions	Continuar amb el flux del qual es venia. La persona pot veure la ubicació en el bloc de notes del dia que l'ha adjuntat.
Notes	Ús opcional de l'aplicació.

Identificador	CU-007
Nom	Marcar viatge com a pendent
Prioritat	Baixa
Descripció	La persona usuària vol marcar un viatge existent, com a pendent.
Actors	Persona Usuària
Pre-condicions	La persona disposa de un dispositiu mòbil amb la aplicació, amb viatges creats.
Iniciat per	Usuari/a
Flux	<ol style="list-style-type: none"> 1. Seleccionar viatge 2. Clicar menú. 3. Clicar opció pendent.
Post-condicions	Apareix la icona de pendent tant a la llista, com a la capçalera del viatge.
Notes	Ús opcional de l'aplicació.

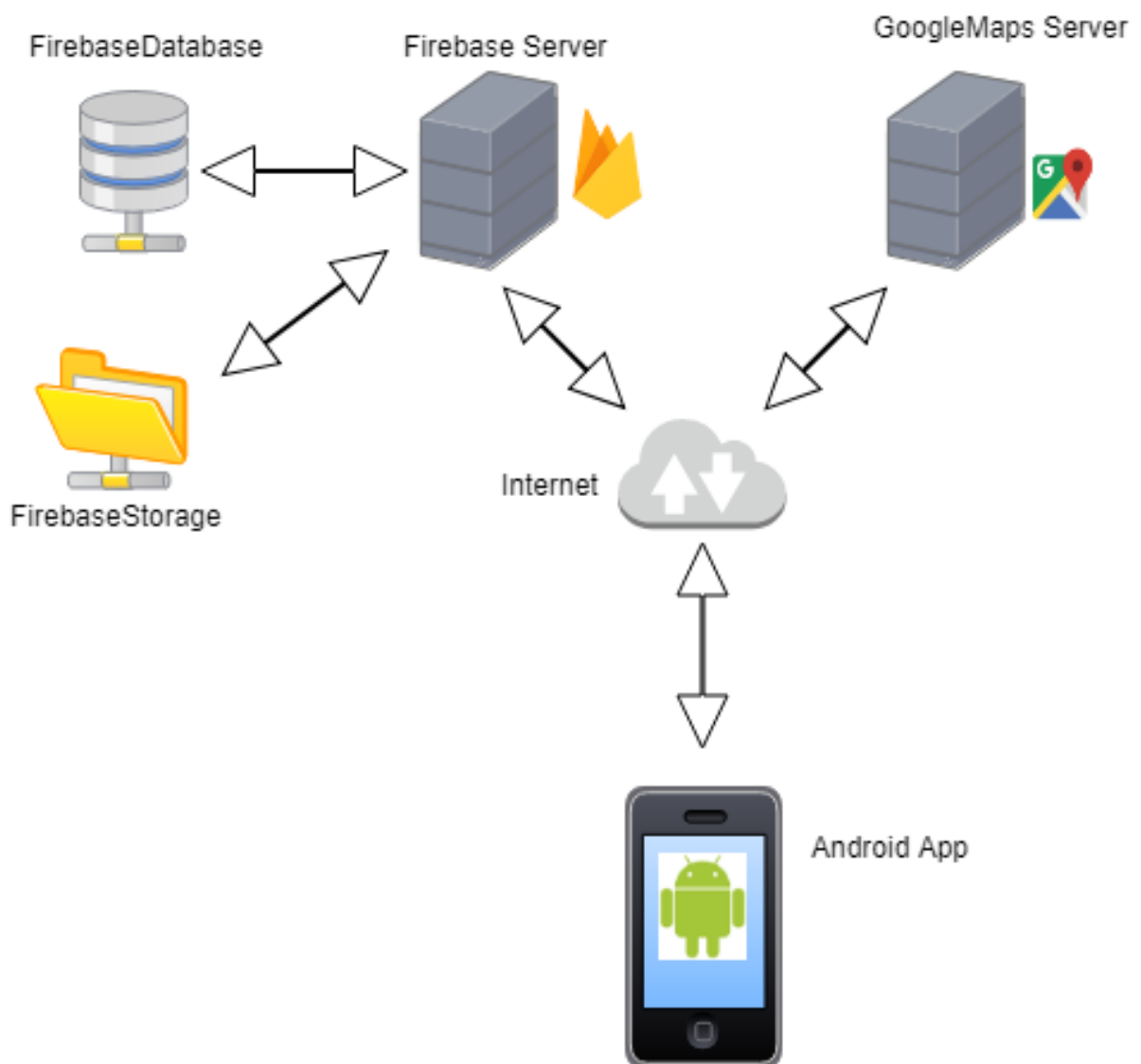
2.2.2 Disseny Arquitectura

2.2.2.1 Arquitectura Global del Sistema

En la il·lustració següent es mostra l'arquitectura global del sistema, on es veuen com interaccionen els diferents components que formen el sistema des de l'aplicació mòbil fins al servidor *backend*.

L'aplicació d'Android a través d'internet es pot connectar amb el servidor *backend* de Firebase per poder guardar dades i actualitzar-les, també hi ha la part de Storage que serveix per guardar arxius, en aquest cas només hi s'hi guarden imatges. Tota la part interna del servidor és invisible des de l'aplicació ja que aquesta només interactua mitjançant APIs amb el servidor.

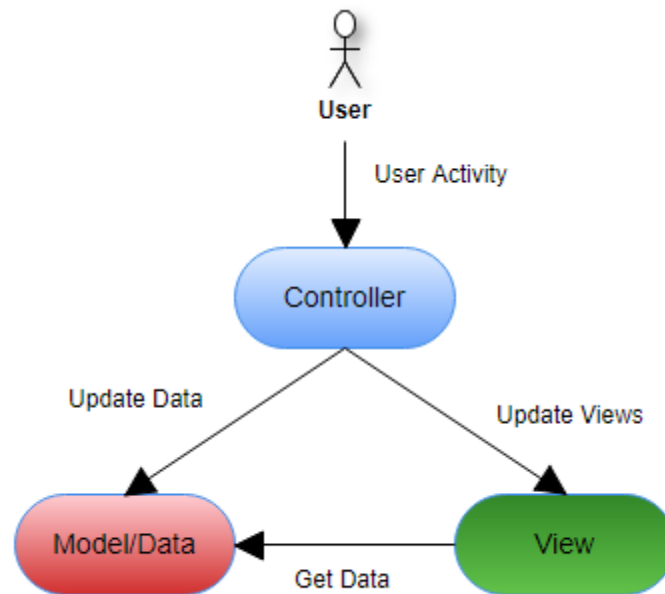
L'aplicació també es connecta amb el servidor de GoogleMaps a través d'una API per a poder utilitzar les dades que ofereixen.



Il·lustració 40 Arquitectura global del sistema

2.2.2.2 Arquitectura MVC

El paradigma utilitzat és el de Model, Vista, Controlador, a partir d'ara MVC. Amb MVC es permet separar la gestió de les dades, i la gestió de les vistes, a través de un controlador.



II-lustració 41 Diagrama MVC

A la part de Model, tindrem les classes definides en el gràfic que es pot veure a la il·lustració 42. És la part que gestiona l'accés a les dades i l'estructura d'aquestes, independentment de l'origen de les dades o la sortida d'aquestes.

A la part de Vista (View) tindrem les diferents pantalles XML de l'aplicació ja que és la part de presentació de les dades cap a les persones usuàries.

A la part del controlador, hi haurà la connexió entre el Model i la Vista. Quan es clica algun botó a alguna de les vistes o una actualització de les dades, el controlador és la part encarregada de traslladar aquests canvis en tots els sentits.

El controlador de l'aplicació està format per varies classes, ja siguin les activitats de Android, que són classes que interactuen directament amb les vistes, i també amb les Bases de dades. I també tenim classes de tipus Fragment que també controlen parts de la vista.

En el gràfic del Anex 1, es pot veure el diagrama de classes complet de l'aplicació i on es diferencien les diferents parts del model MVC.

2.2.2.1 Disseny de la base de dades

La base de dades escollida per aquest projecte és la Firebase de Google. Es tracta de una base de dades NoSQL i l'he escollit perquè al ser de Google la interoperabilitat amb Kotlin i Android Studio és molt simple, ja que només fa falta disposar d'una conta de Google i configurar el servei de Firebase que es pot fer des del mateix Android Studio.

Un altre factor important ha sigut que els accessos que realitzaré a la base de dades no són complexos i es suporten fàcilment en les bbdd NoSQLs.

Els nodes que corresponen a la base de dades són els següents, els he representat mitjançant nodes JSON, enlloc d'un diagrama d'entitats de una base de dades relacional, perquè és tal i com es guarden al servidor. A l'apartat d'implementació es pot veure un exemple real.

```
"TripHeader" : {  
  "id" : número aleatori,  
  "title" : títol del viatge,  
  "place" : lloc del viatge,  
  "from_date" : data d'anada,  
  "to_date" : data de tornada,  
  "bookmark" : Boolean,  
  "persons" : número de persones que viatgen,  
  "travelers" : informació sobre les persones que viatgen,  
  "days": número de dies que té el viatge,  
  "latitude": latitud del lloc del viatge,  
  "longitude": longitud del lloc del viatge,  
  "timestamp" : timestamp del últim canvi del viatee  
}
```

```
"Day" : {  
  "id" : identificador aleatori,  
  "header_id": id del viatge capçalera  
  "place_name": ubicació del dia,  
  "address": addressa de la ubicació,  
  "latitude": latitud del lloc de la ubicació,  
  "longitude": longitud del lloc de la ubicació,  
  "date": data del dia  
  "notes" : bloc de text del dia  
}
```

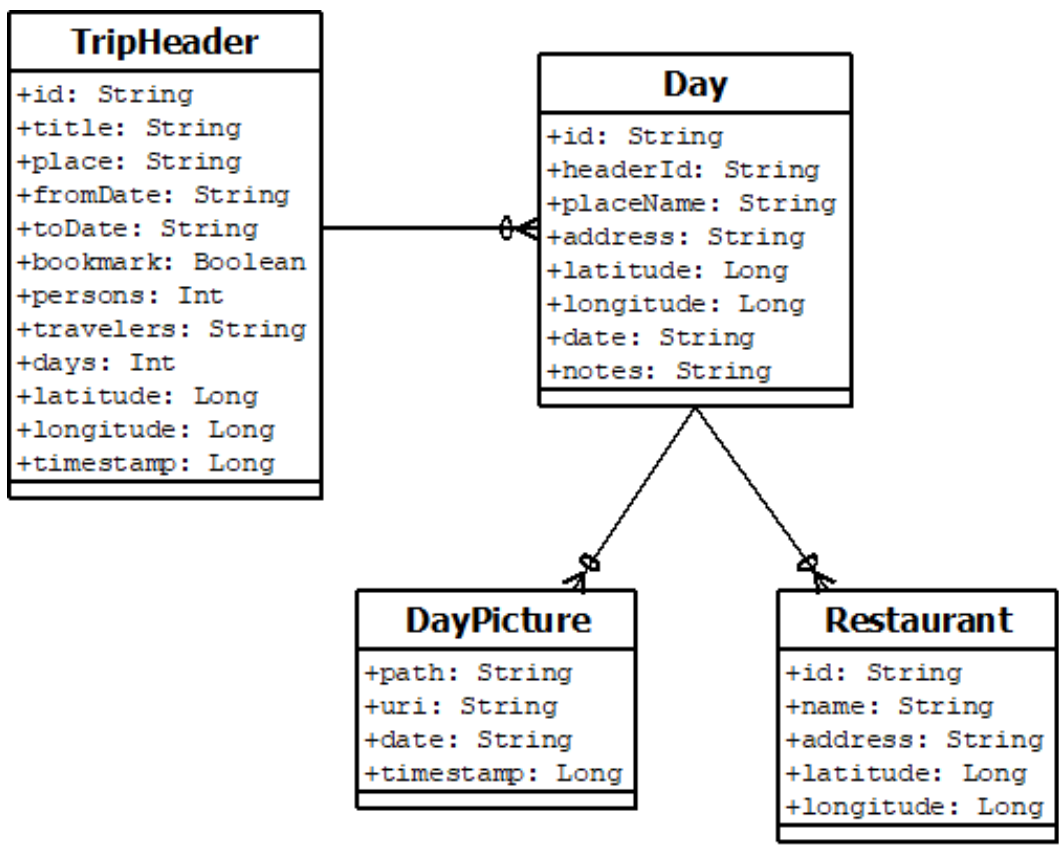
```

"DayPicture" : {
  "uri" : uri de la imatge del FirebaseStorage,
  "path" : ruta de la imatge del FirebaseStorage,
  "date": data,
  "@timestamp" : timestamp
}

"Restaurant" : {
  "id" : identificador aleatori,
  "name" : nom del restaurant,
  "address" : adreça del restaurant,
  "latitude": latitud del restaurant,
  "longitudo": longitud del restaurant
}

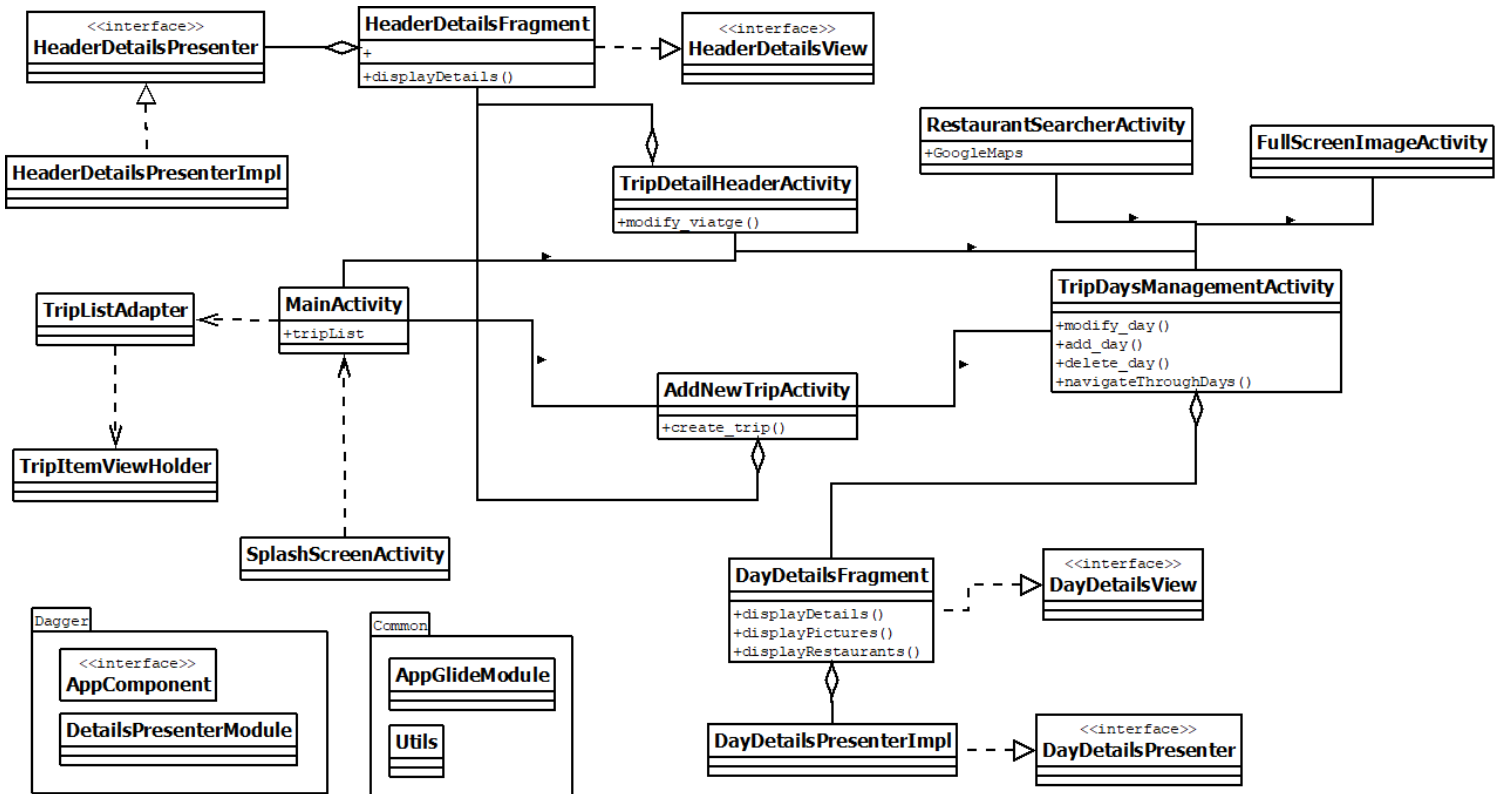
```

A continuació podem veure el diagrama de classes que corresponen a cada un dels diferents nodes



II-lustració 42 Diagrama classes Base de Dades

2.2.2.2 Diagrama Classes del controlador



II-lustració 43 Diagrama classes del controlador

En el diagrama superior, podem veure les diferents classes que componen la part del controlador de l'aplicació.

Hi ha dos paquets separats, el Common i el Dagger. El primer té una classe Utils que es fa servir a varies classes, i també té la classe AppGlideModule que s'utilitza per la gestió de imatges.

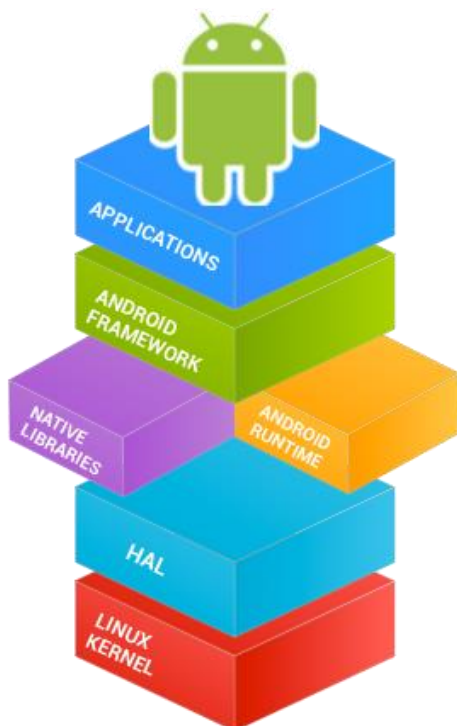
El paquet Dagger és la implementació per a poder injectar dependències en els llocs que li correspon.

3. Implementació

3.1 Eines utilitzades en el desenvolupament

3.1.1 Android

Android és el sistema operatiu per a mòbils desenvolupat per Google. Està basat en una versió modificada de Linux Kernel i altres programes *open source*. La principal característica del seu disseny és que està pensat per a l'ús de pantalles tàctils com poden ser les dels *smartphones* o tauletes, entre altres dispositius.



Il·lustració 44 Arquitectura Android OS⁹

A la il·lustració de l'esquerra podem veure els diferents components que hi ha en el sistema operatiu Android.

La capa superior (Applications) és la que veuen els usuaris i és on trobem les aplicacions, com la que creem en aquest projecte o el correu electrònic, etc.

Les capes inferiors hi trobem el *middleware*, llibreries i APIs que permeten utilitzar el dispositiu i el seu hardware i es connecten amb el Linux Kernel a través de la capa HAL.

La última capa Linux Kernel inclou les funcionalitats bàsiques com la gestió de memòria o bateria, també la connexió amb la xarxa, entre d'altres.

3.1.1 Android Studio

L'entorn utilitzat per al desenvolupament d'aquesta aplicació ha estat l'Android Studio versió 3.2.1.

Android Studio és l'entorn de desenvolupament integrat (IDE) oficial de Google pel desenvolupament d'Android. Està basat en el IDE de IntelliJ de l'empresa JetBrains, i està especialment dissenyat pel disseny amb Android.

El mateix IDE inclou un compilador Gradle, fet que facilita molt la compilació del codi, i la integració de llibreries externes.

També disposa d'emuladors Android integrats que permeten provar les aplicacions ràpidament a diferents dispositius amb pantalles de mida variades i a diferents versions d'Android

⁹ Font: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

3.1.2 Kotlin

Per a la realització de l'aplicació s'ha escollit programar amb Kotlin perquè és un nou llenguatge creat per JetBrains amb llicència Apache 2 a l'any 2012, la primera versió oficial 1.0 va sortir a l'any 2016.

l ha estat adoptat per Google com un llenguatge oficial de desenvolupament d'Android, juntament amb Java. També és un llenguatge que està completament integrat amb l'Android Studio.

Kotlin és un llenguatge tipat estàtic (que és quan la comprovació dels tipus es fa durant la compilació i no durant l'execució) i s'executa sobre la *Java Virtual Machine*. Kotlin combina la programació orientada objectes amb la programació funcional com es va començar a fer amb Java 8 o altres llenguatges com Scala. Un dels avantatges principals que té és que treballa inferència de tipus i això redueix molt la verbatim que hi pot haver amb Java. Una altra de les característiques principals de Kotlin és que és *Null Safety*. És a dir, que està orientat a evitar els errors de referències nul·les tan característics de Java.

3.1.3 APIs

Firestore: És una plataforma de serveis que ofereix Google i es troba completament al núvol es coneix com a Baas(*Backend as a service*), i d'aquesta manera no s'ha de programar res de la banda del servidor, ja que fent crides a la API de Firestore ja es guarda tot de forma automàtica al servidor.

Dels serveis oferts per aquesta plataforma he utilitzat:

- **Firestore Realtime Database:** La base de dades que ofereix és una base de dades NoSQL, de manera que es guarden les dades en format JSON, i es pot utilitzar tan amb internet, com sense. Simplement quan hi ha internet es guarden les dades al núvol directe, i quan no, es guarden al buffer i s'actualitza el servidor tan bon punt hi ha internet al mòbil. S'ha escollit aquesta base de dades per la seva simplicitat a l'hora de implementar-la i perquè així he evitat haver de desenvolupar la part de *backend*.
- **Firestore Storage:** Aquest servei ofereix emmagatzematge de tot tipus de fitxers. La forma de treballar és molt igual que la base de dades, fent crides a la API de forma adequada ja es guarden els fitxers al núvol.

Google APIs:

- **Google Maps:** Aquesta API oferta per Google, permet fer ús de Mapes basats en Google Maps, i ja s'encarrega de tota la gestió de la vista del mapa i de carregar les dades. També permet de forma molt senzilla obrir l'aplicació de Google Maps des de dins l'aplicació quan l'usuari ho requereix.

- **Google Places:** Aquesta API ens permet fer ús del cercador de llocs de Google, d'aquesta manera es poden fer cerques de llocs concrets, i també permet que s'autocompleti la cerca mentre l'usuari escriu.

Glide: Aquest *framework* de codi obert permet gestionar arxius multimèdia o càrrega d'imatges per Android de forma molt senzilla. Es pot cercar i mostrar vídeos, imatges o Gifs , independentment del servei *backend* que s'utilitzi. Inclús conté llibreries que permet interaccionar amb el servei de Firebase de forma natural, sense haver d'utilitzar les APIs de Firebase.

Anko: Aquesta llibreria de codi obert permet que el desenvolupament amb Kotlin sigui més ràpid i fàcil. Conté funcions que faciliten la navegació entre activitats, i altres característiques del desenvolupament Android, com per exemple la creació dinàmica de *layouts*. També té una llibreria per poder gestionar una base de dades SQL però que no s'ha utilitzat.

Dagger: Aquest *framework* serveix per poder utilitzar DI (*Dependency Injection*) amb Android.

La DI és un patró de disseny que permet que un objecte subministri les dependències d'un altre, injectant les dependències a un objecte dependent. D'aquesta manera permet desacoblar objectes i que si un objecte necessita ser modificat, aquests canvis no afectin als objectes dependents.

S'ha utilitzar per poder alliberar controladors de dependències a la hora de mostrar part de les dades i que tot depengui de les classes injectades.

Espresso: *Framework open source* de Google que permet implementar tests funcionals de interfície, és a dir testejar les interaccions amb les aplicacions Android. Aquest framework està lligat a Android, i no es pot utilitzar per testejar aplicacions d'altres sistemes operatius.

JUnit: Llibreria per poder implementar les proves unitàries de l'aplicació. Aquestes proves unitàries són totalment independents a Android.

Mockito: Llibreria per poder *mockejar* classes de l'aplicació per a les proves unitàries.

3.1.4 GitHub

S'ha utilitzat GitHub com a eina Git per a gestionar les versions de codi. El codi està subjecte a la llicència **GNU General Public License v3.0** obert a tothom:

<https://github.com/Charliocat/Antarec>

3.2 Aspectes importants del desenvolupament

3.2.1 Projecte Android Studio

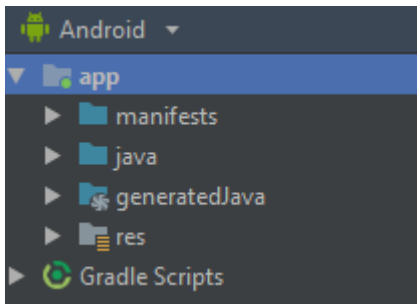
L'aplicació s'ha desenvolupat íntegrament amb l' IDE Android Studio i conté l'estructura que es veu a la il·lustració 45.

A dins del directori de **app** hi trobem la carpeta de manifests, que conté l'Android Manifest, que és un arxiu de configuració on s'apliquen les configuracions bàsiques de l'aplicació

La carpeta **java**, que conté les diferents classes de controlador i model, tot i està escrit en Kotlin la carpeta generada de forma automàtica s'anomena **java**.

A dins de la carpeta **generatedJava** hi trobem codi que es genera de forma automàtica quan es compila l'aplicació.

I a la carpeta **res** (*resources*) trobem les diferents vistes XML de l'aplicació i també altres recursos que s'utilitzen durant l'execució, com pot ser la definició dels menús, o els textos de les pantalles.



Il·lustració 45 Estructura Android Studio

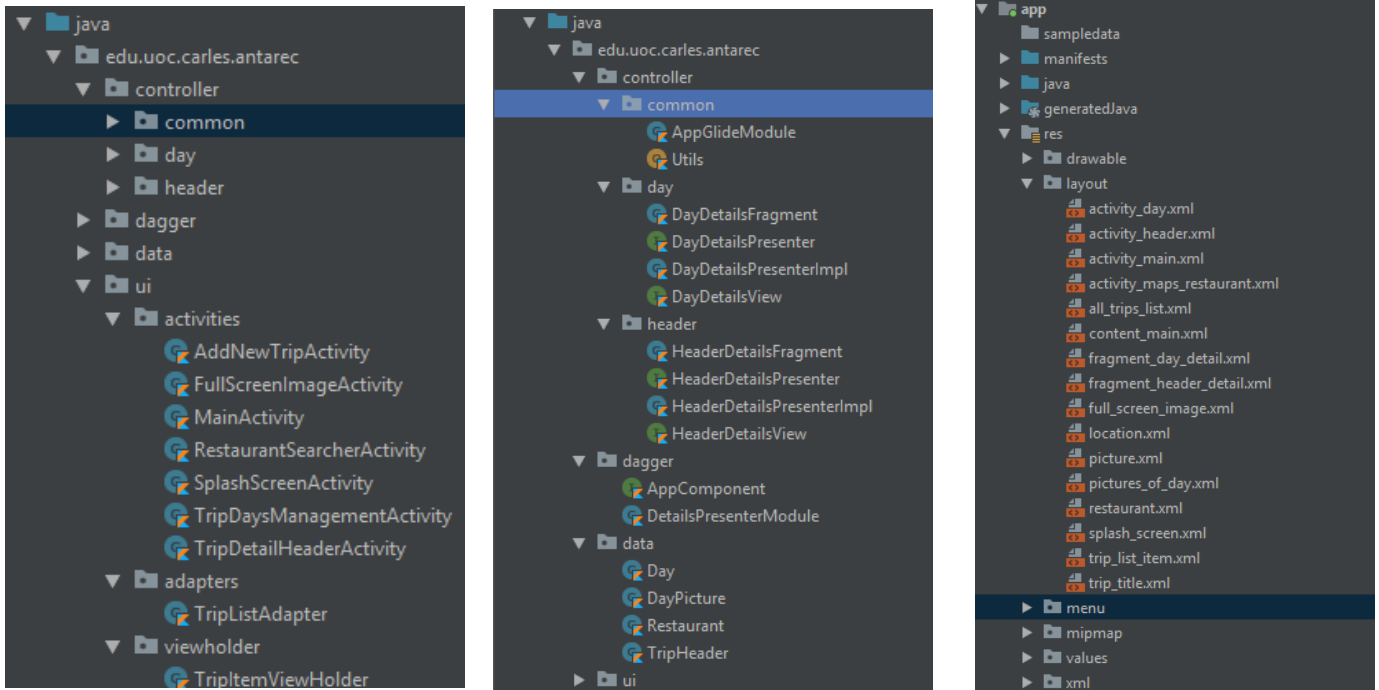
A la il·lustració 46 següent, podem veure els diferents paquets que componen l'aplicació i també la carpeta desplegada de les vistes XML.

El paquet principal és **edu.uoc.carles.antarec** conté agrupats per varis paquets si les classe són Activity, dins del paquet **ui.activities** o Fragment dins del paquet **controller**. Tot i dir-se *controller*, no fa referència al patró MVC explícitament, ja que altres paquets també formen part del *Controller* del patró.

El paquets **ui.adapter** i **ui.viewholder**, contenen classes que ajuden a la visualització de la llista de viatges, que s'explica a l'apartat corresponent.

El paquet **data** conté les classes de modelització de les dates, que podem veure en l'apartat 2.2.2.1.

El paquet **dagger** conté les classes necessàries per a la injecció de dependències.



II-lustració 46 Estructura paquets i classes del projecte

3.2.2 Activitats i Fragments

La gran part del codi del controlador s'ha fet mitjançant les Activitats d'Android i els Fragments.

Les activitats són les diferents pantalles que veiem a l'aplicació, tenen associada una *layout.xml* i estan definides a l'Android Manifest.

Totes les classes *activity* són filles de la classe *AppCompatActivity* la qual subministra els mètodes necessaris per a controlar l'activitat al llarg del cicle de vida de l'*activity*.

El cicle de vida és tot el procés des de que es crea l'activitat, fins que es destrueix, i també què passa quan l'aplicació es queda executant en fons, o es canvia d'activitat.

A partir de l'Android 3.0 es van introduir els fragments, que ens permeten definir comportaments específics a parts de la interfície i de l'*activity*, ja que tots els fragments han de dependre d'una activitat. Es poden combinar diferents fragments dins d'una activitat, i cada un d'ells disposa d'un cicle de fragment propi.

En el codi, hi ha activitats com *TripDetailHeaderActivity* que tenen associada una *layout* que només està composta pels botons del menú. I la part que es veuen les dades del viatge i els camps, està tot gestionat des de la classe *HeaderDetailFragment*.

Programàticament i a través de la classe *Fragment* s'omple la vista de detall del viatge.

```

class TripDetailHeaderActivity : AppCompatActivity() {

    private val REQUIRED = "Required"

    private lateinit var tripHeader: TripHeader
    private var headerDetailsFragment = HeaderDetailsFragment()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_header)

        tripHeader = intent.getParcelableExtra(TRIP)
        headerDetailsFragment = HeaderDetailsFragment.getInstance(tripHeader)
        supportFragmentManager.beginTransaction().add(R.id.header_details_container, headerDetailsFragment).commit()
    }
}

```

A la imatge superior podem veure com en el mètode **onCreate()** de l'activitat s'utilitza el *supportFragmentManager* per utilitzar el *HeaderDetailFragment*, i és en el fragment on omplim els camps de detall.

```

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {
    return inflater.inflate(R.layout.fragment_header_detail, container, attachToRoot: false)
}

```

3.2.3 Disseny de les vistes per a diferents mides de pantalla

Degut a les diferents mides de pantalla que hi ha en el mercat dels dispositius mòbils, tauletes... i per tal de que la visualització de l'aplicació a diferents pantalles sigui correcta, s'ha fet ús de *layouts* relatius que posicionen les vistes en funció de la mida de la pantalla.

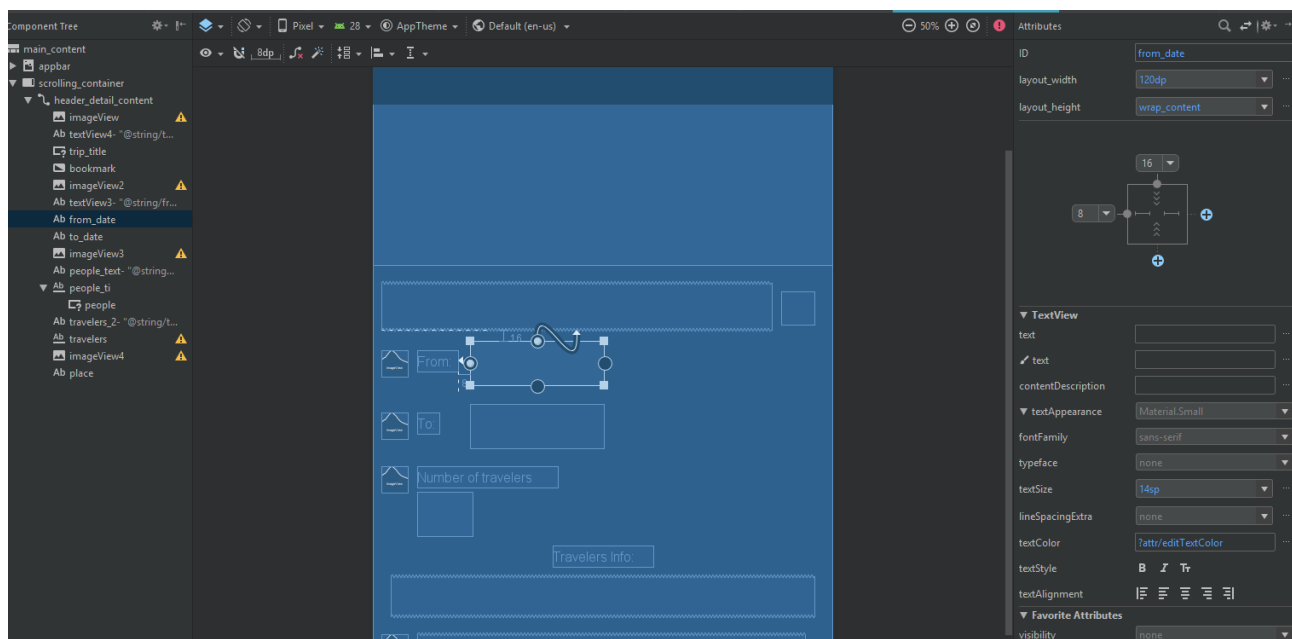
Les principals layouts utilitzades han sigut la **ConstraintLayout** i la **RelativeLayout** que s'explica en el següent apartat.

La ConstraintLayout és similar a la RelativeLayout però és més nova i més fàcil d'utilitzar. Tal com indica la guia d'Android Developers¹⁰ permet crear *layouts* grans i complexos utilitzant una jerarquia de vistes plana. Només s'ha d'utilitzar unes relacions anomenades **constraints** que permeten connectar i alinear les diferents vistes que componen una *view*.

També serveixen per definir l'alçada i amplada d'aquestes vistes. Poden tenir valors de *wrap_content* o *match_constraint* que permeten adaptar la mida al contingut del mateix camp o a la mida que indica la *constraint*. També poden tenir valors en densitat de píxels per adaptar-se a les pantalles.

A la il·lustració 47 podem veure com el camp *from_date* està relacionat a través d'una **constraint** superior amb la part inferior del camp *title* i una constraint horitzontal que va relacionada amb el camp *from_text*. D'aquesta manera sempre es manté la mateixa distància entre el camp *from_text* i el *title*.

¹⁰ <https://developer.android.com/training/constraint-layout/>



II-lustració 47 ConstraintLayout de la vista HeaderDetailFragment

3.2.4 Vista llista viatges (*RelativeLayout*)

La llista de viatges de la pantalla principal de l'aplicació està creada amb una **RelativeLayout** que es gestionen amb un *adaptar* per tal de poder personalitzar les files de la llista.

A la següent imatge es veu el codi font que des de la MainActivity s'utilitza la classe TripListAdapter passant tota la llista de viatges recuperada de Firebase.

```

override fun onDataChange(snapshot: DataSnapshot) {
    displayTripList.clear()
    tripList.clear()
    if (snapshot.exists()) {
        for (t : DataSnapshot! in snapshot.children) {
            val trip : TripHeader? = t.getValue(TripHeader::class.java)
            tripList.add(trip!!)
        }
        displayTripList.addAll(tripList)
        rv_trips_list.adapter = TripListAdapter(displayTripList)

        //alert adapter has changed
        rv_trips_list.adapter.notifyDataSetChanged()
    }
}

```

II-lustració 48 List Adapter a MainActivity

A continuació hi ha una imatge del TripListAdapter on es pot veure com es relacionen els elements de la llista amb la *view* (*trip_list_item*) que volem que s'utilitzi per a mostrar cada element de la llista.

```

class TripListAdapter(private val tripList: MutableList<TripHeader>) : RecyclerView.Adapter<TripItemViewHolder>() {

    override fun onBindViewHolder(holder: TripItemViewHolder, position: Int) {
        holder.bindTripItem(tripList[position])
        holder.selectItem(tripList[position])
    }

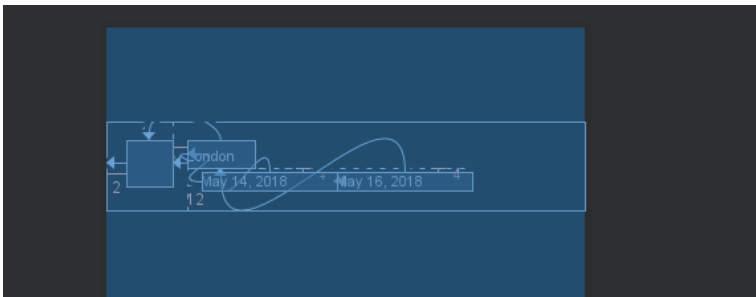
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): TripItemViewHolder {
        val view: View? = LayoutInflater.from(parent.context).inflate(R.layout.trip_list_item, parent, attachToRoot: false)
        return TripItemViewHolder(view)
    }

    override fun getItemCount() = tripList.size
}

```

Il·lustració 49 Classe TripListAdapter

La il·lustració 50 conté la vista del *item* que volem mostrar. Cada element es mapeja segons el format que volem nosaltres en aquesta vista. Podem veure que hi ha els camps per les dates, el títol o el *bookmark* tal i com es mostra a la llista de l'aplicació.



Il·lustració 50 Vista trip_list_item.xml

Per últim tenim la classe ViewHolder que s'encarrega de mapar les dades que rep de la classe *adapter* als diferents camps de la vista *item*. També hi ha el codi que s'assigna el *bookmark* de la vista quan si es clica de manera que actualitza les dades de Firebase a temps real.

```

class TripItemViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
    private lateinit var tripId: String

    fun bindTripItem(tripHeaderItem: TripHeader) = with(itemView) { this: View!
        tripId = tripHeaderItem.id!!
        itemView.title.text = tripHeaderItem.title
        itemView.from_date.text = tripHeaderItem.fromDate
        itemView.to_date.text = tripHeaderItem.toDate
        itemView.bookmarkItem.isChecked = tripHeaderItem.bookmark

        setOnClickListener { it: View!
            val intent = Intent(itemView.context, TripDetailHeaderActivity::class.java)
            intent.putExtra(TRIP, tripHeaderItem)
            itemView.context.startActivity(intent)
        }

        var button: Button = itemView.bookmarkItem as Button
        button.setOnClickListener()
        {
            tripHeaderItem.bookmark = itemView.bookmarkItem.isChecked
            val dbTrip: DatabaseReference = FirebaseDatabase.getInstance().getReference(Utils.PATH_TRIPS).child(tripId)
            dbTrip.setValue(tripHeaderItem)
        }
    }
}

```

Il·lustració 51 Classe TripItemViewHolder

3.2.5 Fitxa restaurant

A la pantalla de dies, hi ha la possibilitat d'adjuntar una fitxa de restaurant, la qual agafa les dades d'una cerca prèvia a un mapa.

Aquestes fitxes de restaurant es creen de manera dinàmica en temps d'execució i això permet que se'n puguin adjuntar tantes com es vulguin.

En el codi de la imatge inferior s'utilitzen les dades de *Restaurant* per a inflar una vista tipus *restaurant.xml* que conté els camps que es volen mostrar. Un cop creada la vista, s'adjunta a la vista de detalls del dia (*day_detail_info*), que és una *LinearLayout*, que conté una línia amb el bloc de notes del viatge, i les línies inferiors que s'afegeixen dinàmicament amb fitxes de restaurants.

```
private fun displayRestaurant(restaurant: Restaurant) {
    val vRestaurant : View! = inflater.inflate(R.layout.restaurant, restaurant_item, attachToRoot: false)
    vRestaurant.place_name.text = restaurant.name
    vRestaurant.restaurant_address.text = restaurant.address
    vRestaurant.restaurant_delete.setOnClickListener { it: View!
        deleteRestaurant(vRestaurant, restaurant.id!!)
    }
    vRestaurant.logo_restaurant.setOnClickListener { it: View!
        selectedRestaurant.view = vRestaurant
        selectedRestaurant.id = restaurant.id

        displayMapActivity(restaurant)
    }
    day_detail_info.addView(vRestaurant)
}
```

II-lustració 52 Creació dinàmica fitxa restaurant

3.2.6 Estructura Firebase Realtime Database

Com ja s'ha explicat a l'apartat del disseny de BBDD els nodes que es guarden a Firebase Realtime Database són nodes JSON.

Les bones pràctiques de Firebase ens recomanen que no es facin molts *subnodes*, perquè tot i suportar fins a 32 nivells de *subnode*, cada consulta per un node recupera tots els nodes fill d'aquest, i es recomana tenir una estructura plana.

També es recomana que no hi hagi recança a la hora de duplicar dades entre nodes, perquè d'aquesta manera és molt més eficient i ràpid fer depèn de quines operacions.



II-Iustració 53 Estructura Firebase Realtime Database



II-Il·lustració 54 Estructura Firebase Realtime Database II

Com es pot apreciar a la imatge anterior, tenim dos nodes:

- **trips:** Aquest node conté un *subnode* per a cada viatge que es crea a l'aplicació. Cada node conté un ID únic i també totes les dades relacionades amb aquell viatge. Els camps que es guarden són els mateixos camps que tenen les classes que hem definit en el Model.
- **trip-days:** Aquest node conté un *subnode* per a cada dia afegit a un viatge. Com podem veure tots els dies d'un viatge són fills de l'identificador del viatge pare (Header). D'aquesta manera recuperar els dies d'un viatge només ens fa falta disposar un ID del viatge que volem veure els fills.
- **day-pictures:** Aquest node conté no conté les imatges que s'adjunten en els dies, sinó que conté l'identificador del dia que s'ha adjuntat la imatge, i la ruta del Firebase Storage de on s'ha guardat la imatge. Així per a poder mostrar les imatges es necessita saber l'ID del dia, recuperar la ruta de la imatge, i anar-la a buscar al Storage. S'utilitza d'aquesta manera per evitar haver de transformar les imatges a un format es puguin guardar en un node JSON.
- **day-restaurants:** Aquest node conté les dades dels restaurants que s'adjunten als diferents dies.

3.2.7 Emmagatzematge i Recuperació de dades a Firebase Realtime Database

Un cop s'ha configurat el projecte des de l'Android Studio per tenir accés a Firebase i utilitzant la consola d'administració de Firebase, es pot especificar les regles (*rules*) que permeten accés a les dades. En el cas d'aquest projecte les *rules* són obertes a tothom ja que l'aplicació no disposa de cap capa d'autorització.

En el codi font, fa falta tenir una referència a Firebase, que pot apuntar a qualsevol node que es vulgui. En el següent cas apuntem al node *trips*.

```
private val reference: DatabaseReference by lazy { FirebaseDatabase.getInstance().getReference(Utils.PATH_TRIPS) }  
private var headerDetailsFragment = HeaderDetailsFragment()
```

Il·lustració 55 Referència Firebase al node *trips*

Un cop tenim la referència ja podem realitzar qualsevol *CRUD operation* (*Create, Read, Update, Delete*).

Per treballar amb Firebase s'utilitzen les classes definides per al Model per tal ja poder mapejar cada node com li correspon.

Per crear un node a Firebase es fa ús de l'operació **setValue()** tal i com es mostra a la imatge següent. Amb **child()** el que aconseguim és que es guardi en un fill del node que està apuntant la referència.

L'objecte **tripHeader** conté les dades de capçalera d'un viatge. Amb la funció **addOnCompleteListener()** es pot saber quan s'ha guardat el viatge al servidor, i mostrar un missatge al usuari.

```
reference.child(tripId).setValue(tripHeader).addOnCompleteListener { it: Task<Void!>  
    Toast.makeText(applicationContext, "Trip saved successfully", Toast.LENGTH_SHORT).show()  
}
```

Il·lustració 56 Firebase guardar viatge

Per a actualitzar un node es fa servir la operació **update()** o també fent **setValue()** amb valors actualitzats d'un node sabent el seu ID.

Per esborrar un node, hi ha la operació **delete()** o també la operació **setValue(null)** apuntant al node que es vol esborrar.

Per a recuperar les dades de Firebase, es fa ús també de la referència, però les operacions disponibles són les següents:

- **addValueEventListener()**: Aquesta operació permet llegir un node i cada vegada que les dades d'aquest node canvien, se li notifica.
- **addListenerForSingleValueEvent()**: Aquesta operació permet llegir un node i encara que es modifiquin les dades, ja no torna a reaccionar.

- **childEventListener():** Aquesta operació permet rebre tots els canvis que ocorren en els *subnodes* del node assignat.

A la següent imatge es mostra un exemple de quan es llegeix la llista de viatges que s'implementa la funció **addValueEventListener()**, que a la vegada implementa un **ValueEventListener**, on amb la funció **onDataChange()** s'obté un **DataSnapshot** amb totes les dades del node que apunta la **reference**. Cada vegada que un node es modifica, torna a executar-se aquest codi, d'aquesta manera les dades s'actualitzen a temps real (*Realtime Database*).

```

tripListListener = reference.addValueEventListener(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        displayTripList.clear()
        tripList.clear()
        if (snapshot.exists()) {
            for (t : DataSnapshot! in snapshot.children) {
                val trip : TripHeader? = t.getValue(TripHeader::class.java)
                tripList.add(trip!!)
            }
            displayTripList.addAll(tripList)
            rv_trips_list.adapter = TripListAdapter(displayTripList)

            //alert adapter has changed
            rv_trips_list.adapter.notifyDataSetChanged()
        }
    }

    override fun onCancelled(error: DatabaseError) {
        Log.w( tag: "MainActivity", msg: "loadItem:onCancelled", error.toException())
    }
})

```

Il·lustració 57 Firebase llegir dades

3.2.8 Adjuntar imatge a un dia des de la càmera

Per adjuntar les imatges als dies es pot fer o bé escollir una imatge de la galeria, o bé fer una foto en el mateix moment.

Quan s'escull una imatge de la galeria és trivial, s'ha d'accedir a la galeria del telèfon i escollir la imatge que es vol. En canvi quan s'utilitza la càmera per a fer una foto, aquesta també s'ha de guardar en el mateix telèfon, i no només al servidor ja que potser la persona que ha adjuntat la imatge en un dia, decideix esborrar aquell dia, però no vol perdre la imatge.

Per a poder guardar la imatge a la galeria amb bona qualitat, s'ha de crear un fitxer buit abans d'obrir la càmera per a poder guardar la imatge. A la imatge 58 utilitzem la funció **createImageFile()** per a poder crear un fitxer buit, i passar-lo a l'Intent que obra la càmera.

Després en el mètode **onActivityResult()** tractem el fitxer que rebem per poder-lo guardar a la galeria amb **galleryAddPic()**, que fa ús del **imageFilePath** que és la ruta on s'ha guardat la imatge feta, i actualitza els fitxers de la galeria.

```
private fun takePhotoFromCamera() {
    val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
    if (intent.resolveActivity(context?.packageManager) != null) {
        val photoFile :File = createImageFile()
        val photoURI :Uri! =
            FileProvider.getUriForFile(context!!, authority: BuildConfig.APPLICATION_ID + ".fileprovider", photoFile)
        intent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI)
        startActivityForResult(intent, camera)
    }
}
```

```
@Throws(IOException::class)
private fun createImageFile(): File {
    // Create an image file name
    val timeStamp: String = SimpleDateFormat(pattern: "yyyyMMdd_HH:mm:ss", Locale.getDefault()).format(Date())
    val storageDir: File = getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES)
    return File.createTempFile(
        prefix: "JPEG_${timeStamp}_", /* prefix */
        suffix: ".jpg", /* suffix */
        storageDir /* directory */
    ).apply { this: File!
        // Save a file: path for use with ACTION_VIEW intents
        imageFilePath = absolutePath
    }
}
```

```
private fun galleryAddPic() {
    Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE).also { mediaScanIntent ->
        val f = File(imageFilePath)
        mediaScanIntent.data = Uri.fromFile(f)
        activity?.sendBroadcast(mediaScanIntent)
    }
}
```

Il·lustració 58 Crear fitxer buit per guardar imatge a la galeria

3.2.9 Emmagatzematge d'imatge a Firebase Storage

Per a poder accedir al Storage també s'ha de configurar el projecte des del mateix Android Studio. I a nivell de codi ens fa falta tenir una referència que apunti al Storage a on volem guardar les imatges.

Quan hem fet la foto, o l'hem seleccionat des de la galeria, obtenim la imatge en format Bitmap i amb el mètode **reduceSize()** el reduïm a una mida de 480 x 360 píxels per evitar problemes de memòria.

També utilitzem el mètode **encoder()** que ens transforma el Bitmap a un array de Bytes per a poder pujar-lo al Storage amb el mètode **putBytes()**.

Com es pot veure a la imatge següent hi ha un **addOnSuccessListener()** que ens mostra un missatge quan s'ha pujat la imatge de forma correcta al Storage. El mètode **addOnFailureListener()** ens mostra un missatge en cas de que la imatge no s'hagi pogut guardar al servidor. I el mètode **addOnProgressListener()** s'utilitza a nivell intern per saber si les imatges s'estan pujant al servidor.

Un problema que hi va haver implementant aquest servei era que el Firebase Storage i el Firebase Realtime Database no estan relacionats directament. Per a solucionar-ho es crea un node **day-pictures** a les bases de dades, que conté la URL del Storage a on està guardada la imatge, i també el dia en que s'ha

guardat. Així quan tractem els dies des de l'aplicació, amb el seu ID podem recuperar les imatges de forma adequada.

```
private fun savePicture(day: Day, bitmap: Bitmap) {
    val dbDaysReference :DatabaseReference = FirebaseDatabase.getInstance().getReference(Utils.PATH_DAY_PICTURES).child(day.id!!)
    val imageId :String? = dbDaysReference.push().key
    val base64ImageString :ByteArray = encoder(reduceSize(bitmap))
    val storage :StorageReference =
        FirebaseStorage.getInstance().getReference(Utils.PATH_DAY_PICTURES).child(day.id!!).child(imageId!!)
    storage.putBytes(base64ImageString).addOnSuccessListener { taskSnapshot ->

        taskSnapshot?.storage?.downloadUrl?.addOnSuccessListener { uri ->
            Toast.makeText(activity?.applicationContext, R.string.picture_saved, Toast.LENGTH_SHORT).show()
            val path :String = taskSnapshot.storage.toString()
            val df :DateFormat! = DateFormat.getDateInstance()
            df.timeZone = TimeZone.getTimeZone(id: "GMT+1")

            dbDaysReference.child(imageId)
                .setValue(DayPicture(uri.toString(), path, df.format(Date()), System.currentTimeMillis()))
        }

    }.addOnFailureListener { it:Exception
        Toast.makeText(context, R.string.picture_deleted, Toast.LENGTH_SHORT).show()
    }.addOnProgressListener { taskSnapshot ->
        val currentProgress :long = (taskSnapshot.bytesTransferred.div(taskSnapshot.totalByteCount)) * 100
        if (currentProgress < mProgress + 15) {
            mProgress = currentProgress.toDouble()
            Log.d(tag: "DayDetailsFragment", msg: "onProgress: upload is: $mProgress done")
        }
    }
}
```

Il·lustració 59 Guardar imatge a FirebaseStorage

3.2.10 Recuperació d'imatges de Firebase Storage i presentació amb Glide

Per a poder mostrar les imatges ens els dies, recupero la llista de imatges de Firebase Database on cada element de la llista conté la URI del Store on està guardada.

Gràcies a Glide, mostrar la imatges es simplifica molt, ja que disposa d'un mètode **load(uri)** que es baixa la imatge automàticament, a la vista que se li indica.

Tal i com es mostra a la il·lustració inferior, només ens fa falta crear una vista container per guardar les vistes de les imatges creades amb Glide.

```
for (picture :DayPicture in picList) {
    val thumbContainer :View! = inflater.inflate(R.layout.picture, pictures, attachToRoot false)
    val thumbView: ImageView = thumbContainer.findViewById(R.id.picture_thumb)
    thumbView.setTag(R.id.glide_tag, picture.uri)
    thumbView.setTag(R.id.path_tag, picture.path)
    thumbView.requestLayout()
    thumbView.setOnClickListener(this)
    thumbView.setOnLongClickListener(this)
    GlideApp.with(fragment: this).load(picture.uri).apply(options).into(thumbView)
    vPictures.addView(thumbContainer)
}
```

Il·lustració 60 Mostrar imatge utilitzant Glide

3.2.11 Ús de la Api de Google Maps pel Trip Header

Per a utilitzar la API de GoogleMaps fa falta registrar-se a la web de Google Developers i obtenir una clau per a poder utilitzar el serveis.

Amb la clau obtinguda ja la podem guardar en un fitxer de *resources* amb el següent format:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="google_maps_api_key" translatable="false" templateMergeStrategy="preserve">
    [REDACTED]
  </string>
</resources>
```

II-lustració 61 Fitxer Google API Key

La layout de Trip Header disposa d'un mapa dins de una *AppBarLayout*, que permet mostrar o amagar el mapa arrossegant la pantalla amunt o avall.

Per a poder tractar amb el mapa i la cerca, és necessari implementar la interfície *OnMapReadyCallback* a la classe que utilitzara el map fragment. Aquesta interfície només conté un mètode que és ***onMapReady()*** i en aquest mètode es gestiona la configuració del mapa, si es vol posar un punt en un lloc concret.

També fem ús del fragment ofert per Google, de *AutoCompleetFragment*. Aquest fragment és el que permet cercar i auto completar a mesura que s'escriu. Un cop es recuperen les dades del lloc que es vol amb *AutoCompleetFragment*, es pot actualitzar el Mapa amb les dades noves, a la imatge inferior el mètode ***moveCamera(x,y)*** s'encarrega de fer-ho.

```
autocompleteFragment =
    activity?.fragmentManager?.findFragmentById(R.id.fragment_autocomplete) as PlaceAutocompleteFragment
var searchInput : EditText = autocompleteFragment.view.findViewById(R.id.place_autocomplete_search_input) as EditText

searchInput.textSize = 16.0F
autocompleteFragment.setOnPlaceSelectedListener(object : PlaceSelectionListener {
    override fun onPlaceSelected(selectedPlace: Place) {
        mapsPlace = selectedPlace.name.toString()
        place.text = mapsPlace
        latitude = selectedPlace.latLng.latitude
        longitude = selectedPlace.latLng.longitude

        moveCamera(latitude, longitude)
        var clearButton : View = autocompleteFragment.view.findViewById(R.id.place_autocomplete_clear_button) as View
        clearButton.visibility = View.GONE
    }
})
```

II-lustració 62 AutocompleteFragment

3.2.12 Ús de Google Place Picker

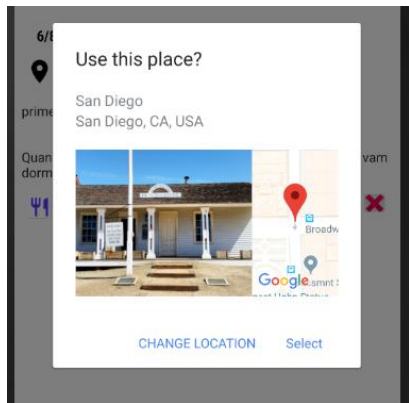
Per a poder cercar la ubicació a la pantalla corresponen a un Dia, s'ha fet ús d'un altre servei de Google, el PlacePicker. Aquest servei és servei per buscar en un mapa, sense haver-ho de gestionar. L'únic inconvenient que té, és que no permet posar una ubicació per defecte. I en el cas de la fitxa de Restaurant,

que es vol poder obrir la ubicació d'aquest, no m'ha servit. En canvi per els dies sí. Perquè si es vol canviar la ubicació s'ha de fer a través del menú.

```
fun placeSearcher() {  
    val builder: PlacePicker.IntentBuilder = PlacePicker.IntentBuilder()  
    startActivityForResult(builder.build(activity), placeMap)  
}
```

Il·lustració 63 PlacePicker

Fent ús del **PlacePicker.IntentBuilder()**, es pot obrir un mapa per cercar i només fa falta implementar la gestió del resultat de l'Activity. Es veu de la següent manera:



Il·lustració 64 PlacePicker result

3.2.13 Dagger Framework

El *framework* Dagger s'utilitza per a injectar dependències dins del codi i poder desacoblar així objectes dependents.

Es va planejar utilitzar DI per a poder desacoblar el codi totalment de Firebase, i que mitjançant les dependències l'accés a la altres bases de dades que no fossin de Google es pogués fer implementant una interfície i des de Dagger indicar quina utilitzar.

Però degut al temps limitat que hi ha hagut per la implementació s'ha quedat amb una estructura que només injecta mètodes per mostrar detalls a les vistes les dades dels quals provenen de la mateixa classe que rep la injecció, enlloc de ser la classe injectada la que va a buscar les dades.

3.3 Pla de Proves

El pla de proves plantejat per l'aplicació, consta de proves automàtiques i proves manuals.

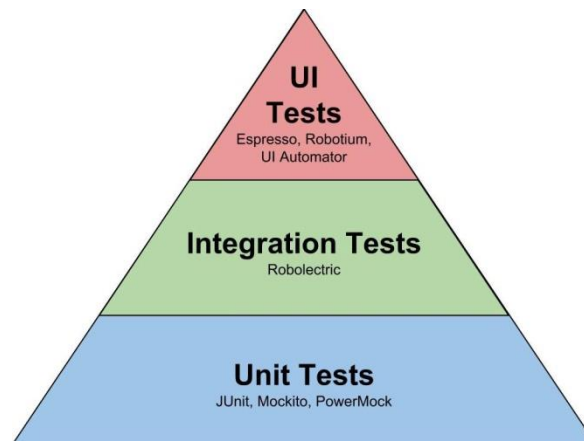
Hi ha dos tipus de proves perquè si l'aplicació és molt complexa fer proves només manuals de tots els processos que poden compondre una aplicació seria una tasca molt feixuga de realitzar i perquè el factor humà és molt important ja que podria passar per alt detalls que podrien ser errors, o errors interns que no es manifestin a les pantalles del usuari.

Per això tenim una part de proves automàtiques que es complementen amb les proves manuals, per a obtenir una cobertura de test completa.

3.3.1 Proves automàtiques

Les proves automàtiques son molt importants en un entorn en què hi ha un desenvolupament constant de noves funcionalitats, i no hi poden haver persones testejant que cada vegada que es fa un canvi, no s'hagi espatllat una altra cosa, o hagi introduït nous *bugs*. Cada nou canvi introduït ha de passar la cobertura de tests.

Tenint en compte la següent piràmide de *testing*, hem ideat el pla per les proves automàtiques.



Il·lustració 65 Piràmide de *testing*¹¹

Els tests automàtics els podríem diferenciar entre les proves unitàries, les proves d'integració i les proves funcionals.

Cada part s'encarrega de testejar coses diferents de l'aplicació, i a mesura que l'aplicació es desenvolupa fa falta es testegi de la manera més adequada.

Com es pot veure idealment la quantitat de tests unitaris és superior a qualsevol de les altres dues, igual que la de tests d'integració és superior a la de tests de UI. Això és perquè a mesura que pugem, fer els tests és més complicat. I perquè si es té una bona base de tests unitaris, no fa falta tenir

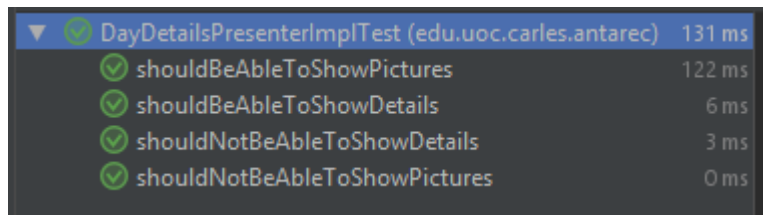
¹¹ font: <https://medium.com/@biratkirat/5-testing-your-app-in-kotlin-57237da63e25>

tants tests dels altres tipus, ja que amb els unitaris es pot cobrir una gran part de la funcionalitat.

3.3.1.1 Proves Unitàries

Aquests test són de cada classe i són totalment independents del sistema o aplicació que s'està testejant. Per anar bé s'hauria de tenir una cobertura de més d'un 95% de codi amb tests unitaris.

La realitat del projecte és que només he cobert unitàriament una classe i mitja, i per testejar es pot executar des del mateix Android Studio.



Test Name	Duration
DayDetailsPresenterImplTest (edu.uoc.carles.antarec)	131 ms
shouldBeAbleToShowPictures	122 ms
shouldBeAbleToShowDetails	6 ms
shouldNotBeAbleToShowDetails	3 ms
shouldNotBeAbleToShowPictures	0 ms

II-lustració 66 Resultat tests unitaris

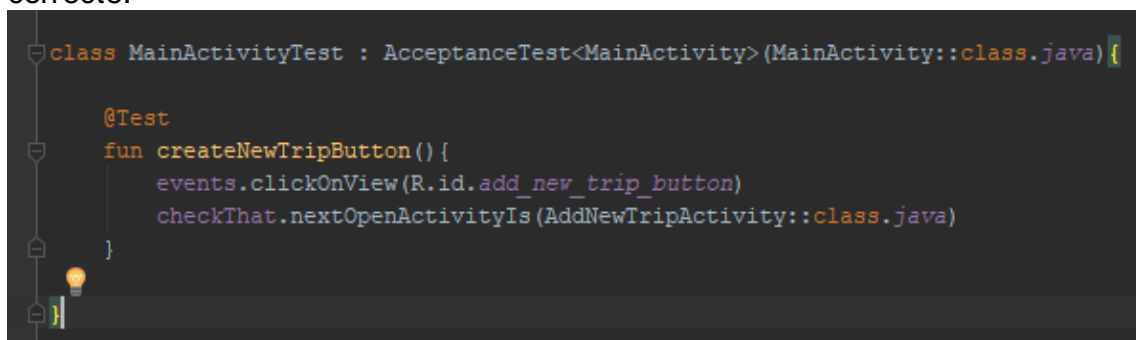
3.3.1.2 Proves Integració i UI

Els tests de Integració de tot al sistema, juntament amb la base de dades no s'ha cobert de forma automàtica perquè es pot fer de forma manual.

Finalment al cim de la piràmide trobem els tests de UI, que són els que n'hi ha menys quantitat perquè tota la funcionalitat de l'aplicació està provada a nivells inferiors. Aquests últims tests només serveixen per cobrir la navegació entre activitats i clicar botons, com també les dades d'alguna vista.

Els tests de UI es llencen mitjançant un emulador de Nexus 5s, en aquest projecte.

A la següent imatge veiem un test UI que comprova que des de la classe *MainActivity* si es clica el botó de crear un viatge nou, es navega a l'activitat correcta.



```
class MainActivityTest : AcceptanceTest<MainActivity>(MainActivity::class.java) {  
    @Test  
    fun createNewTripButton() {  
        events.clickOnView(R.id.add_new_trip_button)  
        checkThat.nextOpenActivityIs(AddNewTripActivity::class.java)  
    }  
}
```

II-lustració 67 Test UI

3.3.2 Proves manuals

El pla de proves manuals serveix provar les diferents funcionalitats de l'aplicació i la visualització de les pantalles. Si l'aplicació fos és molt complexa, provar-ho manualment seria un procés agònic i s'haurien d'automatitzar tots els

tests, o quasi tots. Si es té una bona cobertura seguint la piràmide anterior, no faria falta cap test manual exceptuant els que es comprovi si es veu bé l'aplicació en diferents mòbils o dispositius.

A continuació poso una llista de les proves manuals que s'han de fer:

Pantalla	Prova
Splash Screen	Es mostra la Splash Screen
Llista Viatges	Es llisten els viatges
	Seleccionar un viatge i esborrar-lo
	Editar un viatge
	Buscar un viatge i veure que la llista filtra a temps real
	Afegir un viatge nou i comprovar que apareix a la llista
Viatge Nou	Es veu el Mapa a la part superior
	Es validen els camps sinó estan plens
	Guarda el viatge a Firebase Databse correctament
Edició Viatge	Carrega tots els detalls del viatge
	Es pot modificar la ubicació del viatge
	Es podem modificar tots els camps de la pantalla
	Es guarden els canvis a Firebase Database correctament.
Dia	Carrega tots els detalls del dia
	Es pot modificar la data
	Permet afegir dies nous, si el dia s'ha gravat
	Permet navegar entre dies, només si n'hi ha més d'un.
	Els botons per navegar entre dies, s'activen i desactiven en funció de si son el primer, l'últim o un dia del mig.
	Es pot esborrar un dia, i s'esborra també de la Firebase Database.
	Es pot adjuntar un restaurant des del menú, i es veu a la pantalla.
	Es pot esborrar un restaurant adjuntat i s'actualitza la Firebase Database
	Si es clica sobre el restaurant, s'obre el Mapa amb al ubicació d'aquest.
	Adjuntar una foto des de la càmera i es mostra en el lloc corresponen.
	Adjuntar una foto des de la galeria i es mostra en el lloc corresponen.
	Les fotos es guarden a Firebase Storage
	Guardar el dia a Firebase Database i comprovar que és correcte
	Seleccionar una foto adjuntada, i veure-la a pantalla gran
	Esborrar una foto del dia, i aquesta també s'esborra del Storage, però no de la galeria del telèfon
Es pot afegir, modificar i esborrar la ubicació del dia.	
Restaurant Mapa	Permet cercar restaurants
	Autocompleta el text mentre s'escriu per cercar
	Permet obrir l'aplicació de GoogleMaps per anar a una direcció concreta.

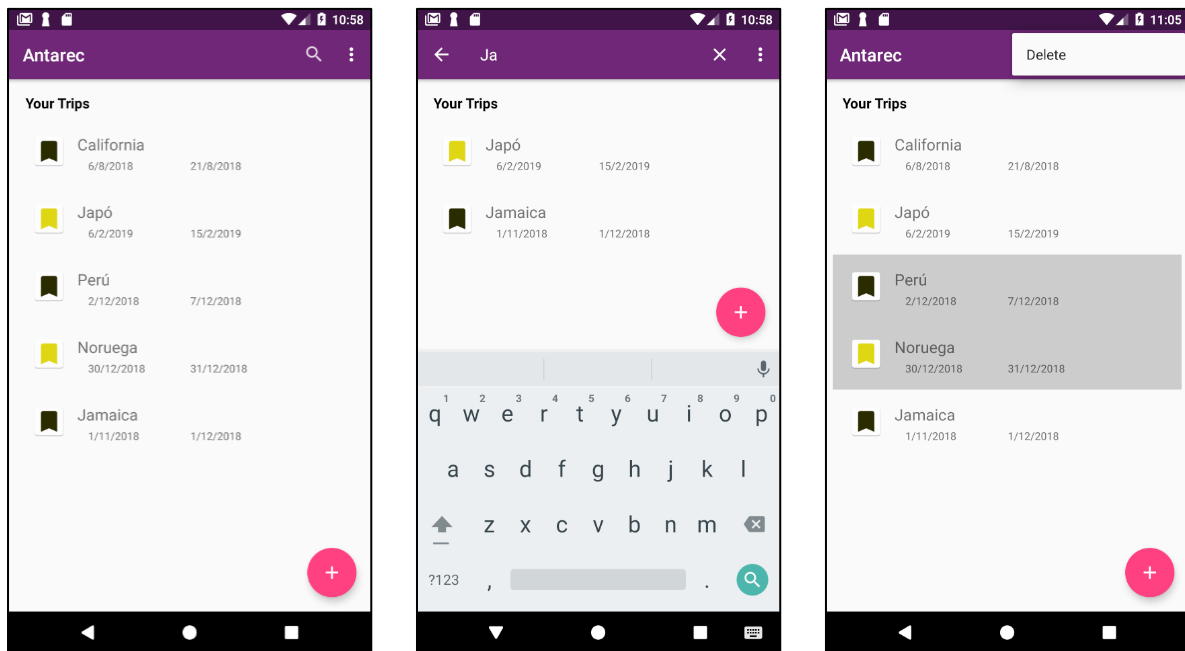
La llista anterior s'ha anat comprovant cada vegada que es feien canvis a l'aplicació.

3.4 Funcionalitat de l'aplicació

Captures de pantalla de l'aplicació en funcionament

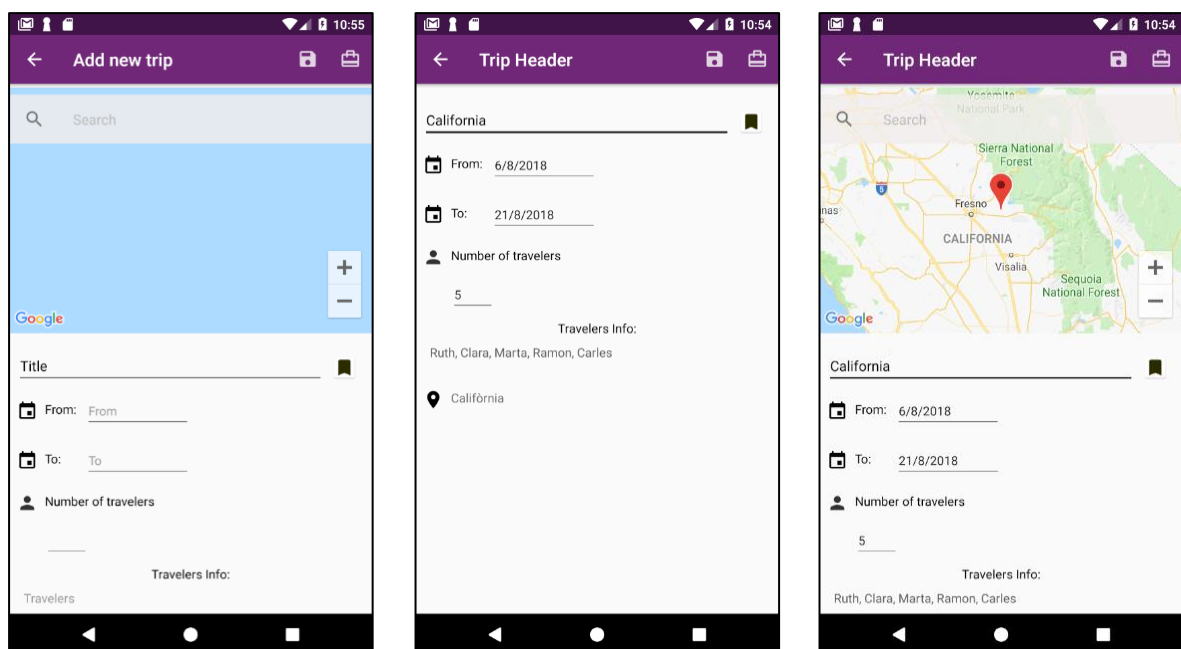
3.4.1 Llista viatges

Imatges de la llista de viatges, quan s'està cercant un viatge de la llista, i quan es tenen viatges seleccionats per esborrar.



Il·lustració 68 Pantalla Llista de viatges
3.4.2 Viatge Header

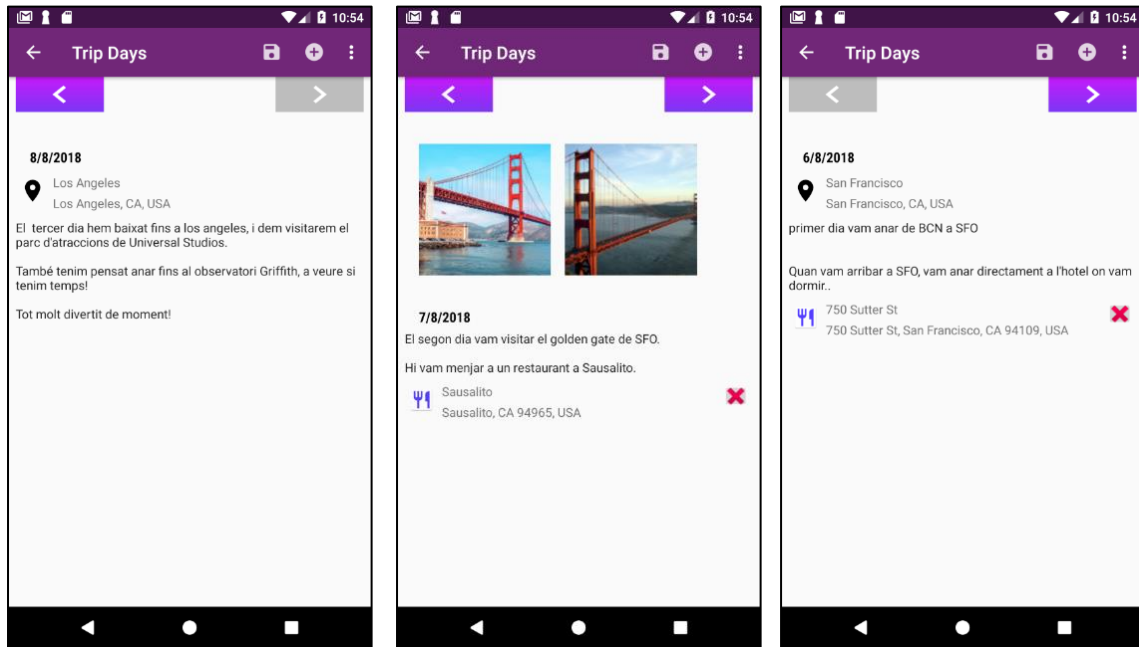
Imatges de quan un viatge es crea i està buit, i de un quan està ple. La pantalla que mostra el mapa és perquè s'arrossega avall la pantalla i mostra el fragment superior que conté el mapa.



Il·lustració 69 Pantalla TripHeader

3.4.3 Dies de un viatge

A les imatges trobem els tres dies que componen el viatge, cada dia té una data diferent, una ubicació diferent, o imatges, o hi ha adjuntats restaurants.

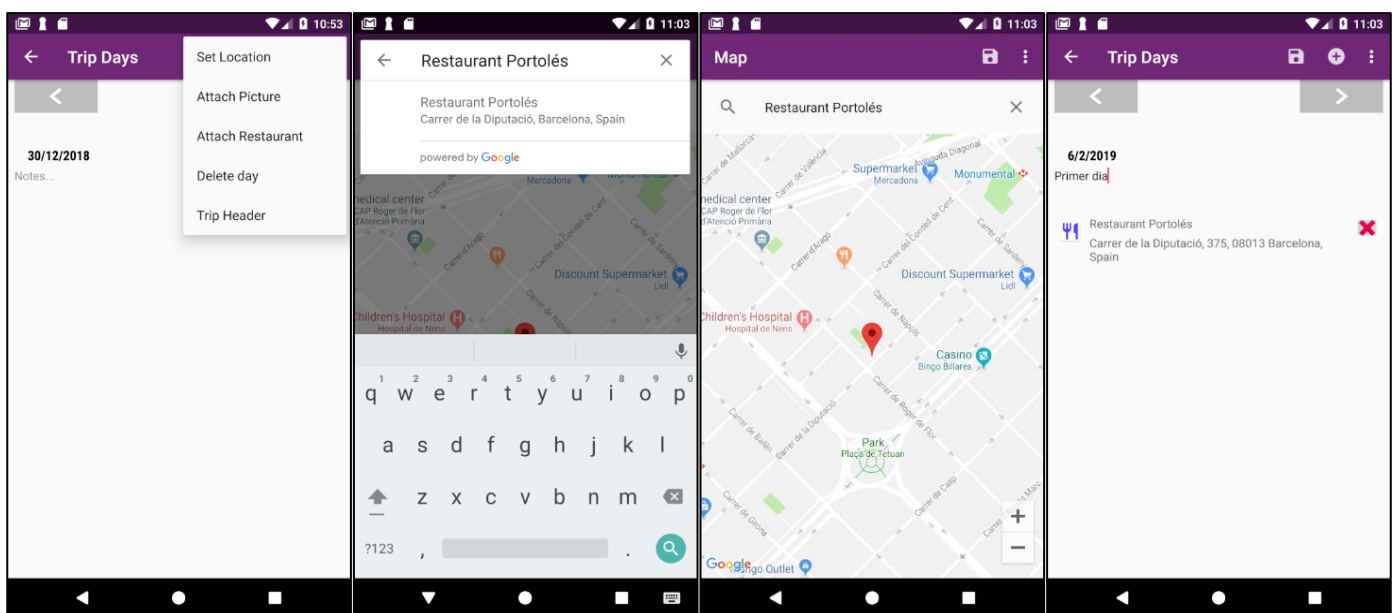


Il·lustració 70 Pantalles dies d'un viatge

3.4.4 Adjuntar un restaurant

Des del menú desplegable escollim la opció per adjuntar un restaurant, veiem que se'ns obre el mapa per a poder cercar, un cop el tenim, podem guardar la ubicació i se'ns mostra la fitxa de restaurant a la ultima captura de pantalla.

Si cliquem sobre la fitxa, se'ns mostra en el mapa un altre cop.



Il·lustració 71 Procés per inserir fitxa restaurant

4. Conclusions

Al llarg dels mesos que he estat elaborant aquest Treball de Fi de Màster m'ha permès aplicar molts dels coneixements que he adquirit al llarg de l'estudi del màster, com també apresos a la carrera i del meu lloc laboral.

Durant la realització he pogut veure com es desenvolupa tot un projecte de software des de l'inici fins al final, planificant a cada etapa què s'ha de fer, distribuint el temps de forma adequada. El fet de tenir diverses entregues parcials corresponent a cada PAC de l'assignatura, ha ajudat a l'hora de planificar, i sobretot a conèixer en quin moment ens trobàvem del projecte. Veient ara el diagrama de Gantt que es va elaborar al inici del projecte, veig que quasi totes les dates i hores a dedicar, s'han anat complint. Per tant es va fer una bona planificació inicial.

La fase d'implementació ha sigut la que ha durat més temps i que he necessitat dedicar-hi més hores, invertint molts caps de setmana i dies festius. Tot i així del plantejament dels objectius inicials vaig haver de reduir funcionalitats a l'aplicació perquè amb el temps limitat que tenim, no hauria pogut aconseguir un producte acabat.

El fet de treballar amb Kotlin ha suposat un repte perquè era un llenguatge nou per a mi, i n'he hagut d'aprendre a mesura que desenvolupava, fet que ha implicat dedicar moltes hores a la recerca d'informació i a aprendre les bases del llenguatge, que per sort meva no són molt diferents al llenguatge Java que si que utilitzo en la meva professió.

Tampoc no havia programat mai cap aplicació amb Android i n'he après juntament amb el desenvolupament, però igual que en el Kotlin, moltes hores han sigut de lectura i estudi per conèixer bé què necessitava fer i com fer-ho. Tot i que també desenvolupar amb prova i error, per veure què estava passant.

La integració amb Firebase i utilitzar una base de dades NoSql ha sigut més senzill del que m'esperava i he pogut resoldre tots els problemes que m'he anat trobant de forma satisfactòria.

M'hauria agradat poder dedicar més hores a la part de *testing* perquè és una part molt important del desenvolupament de software, per obtenir un software de qualitat fa falta tenir tests que cobreixin quasi tot el sistema. Vaig dedicar unes hores a muntar un entorn de integració continuada, però al ser només una persona i tenir pocs dies per fer l'aplicació vaig decidir invertir les hores en l'elaboració de l'aplicació i el treball en general.

Finalment dir que el resultat final és satisfactori i ja tinc una base per continuar desenvolupant amb Android i Kotlin. Gràcies a aquest desenvolupament, hi ha hagut tasques d'investigació i programació que han sigut molt interessants i han fet passar les hores volant. Aquesta àrea de desenvolupament d'aplicacions no l'havia tocat mai i actualment està en creixement continu, sobretot a l'àrea de Barcelona, i aquest va ser el principal motiu que em va portar a escollir-la per el meu TFM.

4.1 Treball futur

Degut a la durada específica del Treball de Final de Màster, es van haver de definir uns objectius al inici del projecte i unes funcionalitats de l'aplicació que es poguessin complir amb el temps que teníem. I tot i que els objectius plantejats s'han complert, hi ha vàries funcionalitats que es podrien afegir per tal de millorar el nostre producte i així ser més competitius en el mercat.

- Afegir gestió d'usuaris. Per fer ús de l'aplicació les persones usuaris s'han de registrar, mitjançant e-mail o compte de *Gmail*. I que els usuaris puguin editar les dades del seu compte, com per exemple afegir una foto de perfil.
- Afegir una capa d'identificació i autorització de l'aplicació per tal de que les dades dels viatges les puguin veure només les persones identificades. Tal i com està ara, tothom pot veure els diferents viatges creats.
- Poder compartir viatges amb altres usuaris de l'aplicació. Per tal de que un grup de persones que viatgen juntes puguin editar el mateix viatge, i els canvis es vegin per totes les persones que tenen permisos d'editar.
- Afegir més informació als dies de cada viatge, similar a tal i com s'afegeixen els restaurants actualment. Per exemple es podria afegir el temps meteorològic a un lloc concret. O també d'un lloc concret, a part de la ubicació com està ara, poder-n'hi afegir varis.
- Traduir la aplicació a diferents idiomes, i que es mostri en l'idioma del dispositiu.
- Millorar l'aplicació per a optimitzar els recursos de bateria i de RAM o CPU.
- Afegir una funcionalitat per tal de poder compartir opinions de llocs o restaurants a altres aplicacions o webs de viatges com TripAdvisor.

A nivell de desenvolupament també afegiria:

- Crear les proves unitàries de les classes que ho permeten i no estan fetes. També fer més proves d'integració i de UI.
- Preparar un entorn de integració continua per a poder detectar *bugs* quan es fan millores en l'aplicació.
- Finalitzar la implementació amb Dagger per tal de desacoblar totalment la integració de Firebase del codi actual.

5. Glossari

- UOC: Universitat Oberta de Catalunya
- App (Application). Terme utilitzat per referenciar les aplicacions mòbils.
- TFM. Treball de fi Master.
- DCU – Disseny Centrat en l'Usuari. Mètode per dissenyar interfícies posant l'usuari en el centre del procés.
- Android Studio. Entorn de desenvolupament integrat d'Android.
- Backend. Part del sistema que s'encarrega de processar les peticions que fan els usuaris a través de dispositius mòbils o altres mètodes, i retorna informació.
- Firebase. Servei de *backend* al núvol de Google.
- Diagrama de Gantt. Gràfic que mostra les diferents tasques d'un projecte, el temps que s'ha de dedicar a cada una d'elles, i les fites del projecte, fins a obtenir un producte final.
- MVC. Model Vista Controlador. Patró de disseny utilitzat per separar les dades, de la lògica i de les interfícies.
- Open source. Codi desenvolupat de forma oberta i col·laborativa. Tot el codi *open source* està sota una llicència emesa per la Open Source Foundation.
- API - Application Programming Interface (Interfície de programació d'aplicacions). Conjunt de classes, mètodes i funcions que ofereixen diverses llibreries per a poder interactuar amb altres software de manera que la persona que implementa la API no veu què hi ha darrera.
- Framework. Són un conjunt de llibreries, eines, compiladors, APIs, programes, que interactuen amb el teu codi per a facilitar la implementació d'aplicacions, productes o solucions de software.
- Bugs: és un error, o fallo en el software.
- Llibreries: Software que implementa una funcionalitat i que es pot reutilitzar a on es necessiti.
- Middleware: Software que assisteix a una aplicació per a que pugui interactuar amb altres aplicacions, programes, xarxa, sistema operatiu...

7. Bibliografia

7.1 Recursos de text

Leiva, Antonio. *Kotlin for Android Developers: Learn Kotlin the Easy Way While Developing an Android App*. Leanpub., 2017.

Clarísó, R. (2013). *Documentació Introducció al treball final (PID_00197247)* FUOC. Fundació per a la Universitat Oberta de Catalunya

Rodriguez, José Ramon (2013) *Documentació El treball final com a projecte (PID_00197248)* FUOC. Fundació per a la Universitat Oberta de Catalunya

Rodriguez, José Ramon (2013) *Documentació La gestió del projecte al llarg del treball final (PID_00197249)* FUOC. Fundació per a la Universitat Oberta de Catalunya

7.2 Recursos Web

We Are Social USA. (2018). Digital in 2018: World's internet users pass the 4 billion mark - We Are Social USA. [online] Available at: <https://wearesocial.com/us/blog/2018/01/global-digital-report-2018> [Accessed 27 Sep. 2018].

Worldwide mobile app revenues in 2015, 2. (2018). Mobile app revenues 2015-2020 | Statistic. [online] Statista. Available at: <https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/> [Accessed 27 Sep. 2018].

Wielen, B. (2018). Insights into the 2.3 Billion Android Smartphones in Use. [online] Newzoo. Available at: <https://newzoo.com/insights/articles/insights-into-the-2-3-billion-android-smartphones-in-use-around-the-world/> [Accessed 27 Sep. 2018].

Hosteltur: Toda la información de turismo. (2018). Las tres claves del aumento de los viajes internacionales en 2025 | Economía. [online] Available at: https://www.hosteltur.com/117083_tres-claves-aumento-viajes-internacionales-2025.html [Accessed 27 Sep. 2018].

Es.wikipedia.org. (2018). Proceso para el desarrollo de software. [online] Available at: https://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software [Accessed 28 Sep. 2018].

Pattern, L. and Idesis, S. (2018). Learn the Model-View-Controller Pattern. [online] OpenClassrooms. Available at: <https://openclassrooms.com/en/courses/4661936-develop-your-first-android-application/4679186-learn-the-model-view-controller-pattern> [Accessed 30 Oct. 2018].

Android, M. (2018). MVC pattern on Android. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/2925054/mvc-pattern-on-android> [Accessed 30 Oct. 2018].

Medium. (2018). Android MVP pattern debunked for beginners! Kotlin . [online] Available at: <https://medium.com/mindorks/android-mvp-pattern-debunked-for-beginners-kotlin-c56c888222e0> [Accessed 30 Oct. 2018].

En.wikipedia.org. (2018). Kotlin (programming language). [online] Available at: [https://en.wikipedia.org/wiki/Kotlin_\(programming_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) [Accessed 12 Dec. 2018].

Android Developers. (2018). Kotlin and Android | Android Developers. [online] Available at: <https://developer.android.com/kotlin/> [Accessed 12 Dec. 2018].

Leiva, A. (2018). Kotlin es oficial en Android ¿Qué implicaciones tiene para los desarrolladores?. [online] Genbeta.com. Available at: <https://www.genbeta.com/desarrollo/kotlin-es-oficial-en-android-que-implicaciones-tiene-para-los-desarrolladores> [Accessed 12 Dec. 2018].

Firebase. (2018). Firebase Realtime Database | Firebase Realtime Database | Firebase. [online] Available at: <https://firebase.google.com/docs/database/> [Accessed 30 Oct. 2018].

tuprogramacion.com. (2019). ¿Qué es el Android manifest?. [online] Available at: <http://www.tuprogramacion.com/glosario/que-es-el-android-manifest/> [Accessed 2 Nov. 2018].

7.3 Recursos Software

A parts dels citats a la memòria també s'ha fet servir:
El software *open source* DIA versió 0.97.2 per a realitzar els diagrames UML.

Formularis de les enquestes realitzats amb GoogleForms:

Google.com. (2018). Google Forms - create and analyze surveys, for free.. [online] Available at: <https://www.google.com/forms/about/> [Accessed 27 Oct. 2018].

Prototipus d'alta fidelitat realitzat amb el software:

Justinmind.com. (2018). Free prototyping tool to design web & mobile apps - Justinmind. [online] Available at: <https://www.justinmind.com/> [Accessed 27 Oct. 2018].

Diagrama de flux realitzat amb el software online:

App.cacoo.com. (2018). Cacoo. [online] Available at: <https://app.cacoo.com/p/recent> [Accessed 27 Oct. 2018].