



**UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH**

Facultat d'Informàtica de Barcelona



CREACIÓ D'UNA APLICACIÓ MÒBIL PER LA GESTIÓ DE LES DESPESES D'UN VIATGE

POL SALVADOR NOGUÉS

Director/a

ANTONI URPI TUBELLA (Departament d'Enginyeria de Serveis i Sistemes d'Informació)

Titulació

Grau en Enginyeria Informàtica (Enginyeria del Software)

Memòria del treball de fi de grau

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

27/06/2024

Índex de continguts

1. Introducció	12
1.1. Context del projecte	12
1.1.1. Identificació del problema	12
1.1.2. Conceptes previs	13
1.2. Objectius	13
1.3. Parts interessades.....	14
1.4. Estat de l'art	14
1.5. Riscos	16
1.6. Metodologia.....	17
1.6.1. Eines de seguiment	17
1.7. Organització de la resta de la memòria.....	18
2. Descripció de les tasques	20
2.1. Tasques	20
2.1.1. Tasques de gestió del projecte.....	20
2.1.2. Tasques de documentació de la memòria	21
2.1.3. Tasques de desenvolupament.....	22
2.1.3.1. <i>Inception</i>	22
2.1.3.2. Usuaris	23
2.1.3.3. Viatges	23
2.1.3.4. Despeses	24
2.1.3.5. Compartició	24
2.1.3.6. Recompenses	25
2.1.3.7. Informes	25
2.1.3.8. Deutes	25
2.1.3.9. Instanciar.....	25

2.2.	Recursos	26
2.2.1.	Recursos humans.....	26
2.2.2.	Recursos materials	26
2.2.2.1.	Recursos físics	26
2.2.2.2.	Software	26
3.	Planificació inicial.....	28
3.1.	Estimacions.....	28
3.2.	Diagrama de Gantt.....	32
4.	Gestió econòmica	33
4.1.	Identificació dels costos	33
4.1.1.	Costs de personal per activitat.....	33
4.1.2.	Costs generals.....	36
4.1.3.	Costs de contingència	37
4.1.4.	Imprevistos	37
4.1.5.	Pressupost final	38
4.2.	Control de gestió	38
5.	Gestió del risc: Plans alternatius i obstacles.....	39
5.1.	Calendari tancat	39
5.2.	Falta d'experiència	39
5.3.	Previsió incorrecta.....	39
5.4.	Dependència d'APIs externes	40
5.5.	Canvis en la tecnologia	40
6.	Especificació de requisits.....	41
6.1.	Requisits funcionals	41
6.1.1.	Llista de requisits funcionals	41
6.1.2.	Diagrames de casos d'ús.....	42
6.1.3.	Descripció de casos d'ús	42

6.1.4.	Model conceptual de les dades.....	55
6.2.	Requisits no funcionals	57
7.	Arquitectura del sistema	58
7.1.	Arquitectura física.....	58
7.2.	Arquitectura lògica.....	58
7.3.	Patrons utilitzats	61
7.3.1.	Patrons de capa de presentació	61
7.3.2.	Patrons capa domini	61
7.3.3.	Patrons capa de dades	63
7.4.	Exemples de diagrames de seqüència.....	63
7.5.	Bases de dades	65
7.5.1.	Disseny lògic.....	65
7.5.1.1.	Despesa	65
7.5.1.2.	Recompensa.....	66
7.5.1.3.	Usuari	66
7.5.1.4.	Viatge	66
7.5.2.	Disseny físic.....	67
7.6.	Disseny de la interfície	68
8.	Implementació	69
8.1.	Tecnologies i llenguatges emprats.....	69
8.2.	Eines de desenvolupament	72
8.3.	Aspectes rellevants del codi de la implementació	73
8.3.1.	Codi de la capa de presentació.....	73
8.3.2.	Codi de la capa de domini.....	77
9.	Proves.....	80
10.	Aspectes legals.....	82
10.1.	Lleis aplicables.....	82

10.2.	Llicències	82
11.	Seguiment del projecte	83
11.1.	Canvis de planificació i pressupost	83
11.2.	Execució real	84
12.	Sostenibilitat	86
12.1.	Econòmica	86
12.2.	Social	87
12.3.	Ambiental	89
13.	Conclusions i treball futur.....	91
13.1.	Competències tècniques treballades i relació amb la especialitat de software	91
13.2.	Conclusions personals	93
13.3.	Treball futur	93
14.	Referències.....	95

Índex de figures

Figura 1: Flux de treball en GitFlow.....	18
Figura 2: Diagrama de Gantt.....	32
Figura 3: Diagrama de casos d'ús de les funcionalitats principals dividit per èpiques.....	42
Figura 4: Diagrama del model conceptual de les dades.....	56
Figura 5: Arquitectura física del sistema	58
Figura 6: Diagrama d'arquitectura lògica de primer nivell	59
Figura 7: Diagrama d'arquitectura lògica de segon nivell	60
Figura 8: Exemple de Multiple Activity Pattern	61
Figura 9: Patró MVT	62
Figura 10: Middlewares usats al settings.py.....	62
Figura 11: Configuració de la base de dades al settings.py	62
Figura 12: Diagrama del funcionament de l'ORM de Python	63
Figura 13: Diagrama de seqüència de la funcionalitat de veure viatges	64
Figura 14: Taules a la base de dades MySQL	65
Figura 15: Classe Despesa	66
Figura 16: Classe Recompensa	66
Figura 17: Classe Usuari.....	66
Figura 18: Classe Viatge	67
Figura 19: Exemple d'una migració de la classe Recompensa	67
Figura 20: Diagrama amb les pantalles, classes i formularis de l'aplicació	68
Figura 21: Configuració del projecte de Google Firebase amb les 3 builds	69
Figura 22: Proveïdors d'accés de Google Firebase, amb la de Google habilitada	70
Figura 23: APIs habilitades a Google Cloud.....	70
Figura 24: Credencials de Google Cloud amb les claus de les APIs.....	70
Figura 25: Contingut del Dockerfile	71
Figura 26: Contingut del Dockerfile	71
Figura 27: Contenidors del backend, api i db, del Docker Desktop	71
Figura 28: Consola d'Amazon EC2	72
Figura 29: Dades de la instància de RDS	72
Figura 30: Classes del frontend.....	73

Figura 31: Classes DespesaInfo i DespesaShowInfo.....	73
Figura 32: Pantalla Principal de l'app, amb la finestra lateral i el desplegable .	74
Figura 33: Exemple de missatge d'error en no escriure el nom de la despesa	74
Figura 34: Pantalla d'un viatge amb les diferents despeses i la de creació d'aquestes	75
Figura 35: Pantalla de detalls d'un viatge.....	75
Figura 36: Pantalla de l'informe d'un viatge.....	76
Figura 37: Exemple d'Intent per canviar de pantalla	76
Figura 38: Selector de Build Variants a l'Android Studio	77
Figura 39: Diferents Flavors definits al build.gradle.kts	77
Figura 40: Apps de l'aplicació al settings.py.....	77
Figura 41: URLs definides a l'arxiu urls.py	78
Figura 42: Funció de get_or_edit_or_delete_despesa() de la classe Despesa	79
Figura 43: Definició de Databases al settings.py.....	79
Figura 44: Fragment del TestCase de la classe Usuari.....	80
Figura 45: Fragment del TestCase d'integració de la classe Despesa.....	80
Figura 46: Codi del github_actions.yml	81
Figura 47: Fragment dels resultats del Github Actions en fer commits	81
Figura 48: Fragment del Logcat d'Android Studio	81
Figura 49: Els diferents sprints i èpiques del Taiga	84
Figura 50: Fragment del Taskboard del Taiga de l'Sprint 3.....	84
Figura 51: Gràfic del progrés del projecte	85
Figura 52: Captura de pantalla del repositori del GitHub.....	85

Índex de taules

Taula 1: Taula comparativa de l'estudi de mercat	15
Taula 2: Estimació de hores, recursos i dependències de les tasques	31
Taula 3: Salari per hora de cada rol de l'equip	33
Taula 4: Cost de personal per activitat	36
Taula 5: Cost del lloguer d'una oficina de 30 m2 a Barcelona a preu de 13€/m2	36
Taula 6: Cost del hardware.	37
Taula 7: Costos estimats pels imprevistos.	37
Taula 8: Estimació del pressupost final.	38
Taula 9: Probabilitat i hores estimades de cada risc.	39
Taula 10: Cas d'ús 1-Iniciar sessió amb Google.	43
Taula 11: Cas d'ús 2-Tancar sessió.	43
Taula 12: Cas d'ús 3-Veure viatge.	44
Taula 13: Cas d'ús 4-Registrar viatge.	44
Taula 14: Cas d'ús 5-Editar viatge.	45
Taula 15: Cas d'ús 6-Eliminar viatge.	46
Taula 16: Cas d'ús 7-Veure despesa.	46
Taula 17: Cas d'ús 8-Afegir despesa.	47
Taula 18: Cas d'ús 9-Editar despesa.	48
Taula 19: Cas d'ús 10-Eliminar despesa.	49
Taula 20: Cas d'ús 11-Filtrar despesa.	50
Taula 21: Cas d'ús 12-Unir-se a viatge.	51
Taula 22: Cas d'ús 13-Expulsar usuari.	52
Taula 23: Cas d'ús 14-Veure recompenses.	52
Taula 24: Cas d'ús 15-Bescanviar punts.	53
Taula 25: Cas d'ús 16-Consultar informes.	54
Taula 26: Cas d'ús 17-Pagar deutes	54

Resum

Aquest projecte constitueix el Treball Final del Grau d'Enginyeria Informàtica de la Facultat d'Informàtica de Barcelona de la Universitat Politècnica de Catalunya. El treball consisteix en el desenvolupament d'una aplicació mòbil per la gestió de les despeses d'un viatge, sigui individual o en grup.

L'aplicació ofereix una àmplia gamma de funcionalitats per dur a terme aquesta tasca: sistema d'usuaris amb la possibilitat de compartir els viatges, crear, editar i eliminar viatges i despeses, visualització d'informes, sistema de deutes amb simulació de pagament i una sèrie de recompenses a bescanviar.

Aquesta memòria comprèn tot el procés de creació de l'aplicació des de la planificació, la metodologia, el disseny i l'especificació fins a la implementació, les proves i la conclusió.

Resumen

Este proyecto constituye el Trabajo Final del Grado de Ingeniería Informática de la Facultad de Informática de Barcelona de la Universidad Politécnica de Catalunya. El trabajo consiste en el desarrollo de una aplicación móvil para la gestión de los gastos de un viaje, sea individual o en grupo.

La aplicación ofrece una amplia gama de funcionalidades para realizar esta tarea: sistema de usuarios con la posibilidad de compartir los viajes, crear, editar y eliminar viajes y gastos, visualización de informes, sistema de deudas con simulación de pago y una serie de recompensas a canjear.

Esta memoria comprende todo el proceso de creación de la aplicación desde la planificación, la metodología, el diseño y la especificación hasta la implementación, las pruebas y la conclusión.

Abstract

This project constitutes the Final Project of the Computer Engineering Degree of the Faculty of Informatics of Barcelona of the Universitat Politècnica de Catalunya. The work consists of the development of a mobile application for managing the expenses of a trip, either individually or in a group.

The application offers a wide range of functionalities to carry out this task: user system with the ability to share trips, create, edit and delete trips and expenses, view reports, debt system with payment simulation and a series of rewards to be redeemed.

This report covers the entire process of creating the application from planning, methodology, design and specification to implementation, testing and conclusion.

Agraïments

Voldria donar les gràcies al meu director Toni Urpí per tot el suport rebut durant el projecte i totes les seves aportacions i ajudes per poder fer el millor treball possible durant aquests mesos. A més, també agrair el fet que sigui el meu director en no haver trobat cap fins pocs dies abans del fi del termini per matricular el TFG.

A més, agrair també a tots els professors i professores que m'han donat un ensenyament excel·lent durant aquests últims quatre anys per ser un bon enginyer informàtic.

Per últim, donar les gràcies a tothom, família i amics, que han estat al meu voltant durant el transcurs del projecte i m'han acompanyat durant aquesta etapa.

1. Introducció

1.1. Context del projecte

El projecte “Creació d’una aplicació mòbil per la gestió de les despeses d’un viatge” es tracta d’un treball de fi de grau de la Facultat d’Informàtica de Barcelona de la Universitat Politècnica de Catalunya, en l’especialitat d’Enginyeria del Software.

Aquest treball de fi de grau és de modalitat A, és a dir, és un projecte realitzat a la UPC.

L’objectiu principal d’aquest projecte és crear una aplicació mòbil per a dispositius Android, anomenada Travudget (de *travel*, viatjar, i *budget*, pressupost), que permeti als usuaris fer un seguiment i gestió de les despeses dels seus viatges, amb tot nivell de detall i amb la possibilitat de dividir-les amb altres usuaris de l’aplicació, a més de poder accedir-hi des de diferents dispositius alhora.

1.1.1. Identificació del problema

La gestió de despeses de viatge [1] és una preocupació cada vegada més important en un món on el turisme i els desplaçaments són tan presents a la nostra vida. La nostra societat viatja tant per raons de negocis com per oci, generant una demanda creixent per a solucions eficaces de seguiment i control de les despeses de viatge, que ha donat lloc a una nova àrea de gestió financera centrada específicament en les despeses [2] que es produeixen en aquests viatges.

El procés de viatjar involucra molts tipus de despeses [3], des de les més evidents com ara els bitllets d’avió, l’allotjament i els àpats, fins a altres despeses menors que poden passar desapercebudes, com els taxis, les propines o fins i tot els petits articles de primera necessitat. També solen sorgir imprevistos durant el viatge [4], com ara emergències mèdiques o canvis en els plans de transport.

A més, establir un pressupost abans d’iniciar un viatge és essencial per garantir la sostenibilitat financera i evitar despeses innecessàries, així com tenir una idea de l’estil i nivell de sortida que s’espera.

Per totes aquestes raons és molt recomanable tenir un control de les despeses fàcil i útil a mà, com pot ser al nostre *smartphone*.

1.1.2. Conceptes previs

En aquest apartat s'expliquen en profunditat termes que s'utilitzen amb molta freqüència en aquest document, per tal de no confondre al lector i especificar quin és el significat plantejat.

- **Viatge:** Quan es parla de viatge s'estarà parlant d'un període de temps durant el qual una persona es desplaça d'un lloc a un altre de caràcter turístic, no especialment de negocis, i té certes despeses.

Aquest aclariment es deu al fet que hi ha gran quantitat d'aplicacions que tracten els viatges de negocis, però no aquesta.

- **Despesa:** Import gastat en un bé o servei durant un viatge. Es considera d'un dels següents tipus: menjar, compres, turisme, allotjament, transport o altres.

1.2. Objectius

L'objectiu és desenvolupar una app mòbil per sistemes Android que sigui completament funcional i, per tant, que permeti fer una completa i correcta gestió de les despeses d'un viatge. Es pot desglossar en diferents subobjectius:

- Els usuaris poden enregistrar totes les despeses d'un viatge amb el nivell de detall necessari, podent especificar aspectes com la categoria, dia o concepte de cadascun, entre altres.
- Els usuaris poden tenir diferents viatges alhora i poden afegir cada despesa al que vulguin.
- Es poden incloure altres usuaris de diferents dispositius a aquests viatges i tenir accés a totes les funcionalitats, fins i tot poder dividir les despeses amb altres usuaris.
- L'aplicació té una interfície atractiva, fàcil d'usar i intuïtiva, que proporciona una bona experiència d'usuari.

1.3. Parts interessades

Els *stakeholders* o parts interessades són les parts implicades o afectades pel projecte, crucials ja que defineixen els objectius i desenvolupament de l'aplicació.

Aquestes han estat:

- **Director del projecte:** l'Antoni Urpi Tubella, qui ha supervisat l'evolució del projecte per tal que avanci correctament durant el desenvolupament.
- **Enginyer de software:** Ha de definir els requisits tècnics, com la selecció de tecnologies apropiades per al desenvolupament de l'aplicació, l'arquitectura del sistema, el disseny del model de classes i de la base de dades, a més d'implementar i provar el software.
- **Dissenyador gràfic:** Persona amb la tasca de dissenyar les interfícies, tal que siguin atractives i fàcils d'usar.
- **Equip de desenvolupament:** Els encarregats d'implementar el codi de l'aplicació.
- **Equip de test:** Els encarregats de comprovar que funcioni correctament el codi per assegurar-se que no hi ha cap error.
- **Clients:** Les persones que utilitzen l'aplicació final.
- **Autor del projecte.** En aquest cas que el projecte és realitzat per una única persona, aquest realitza totes les tasques de desenvolupament.

1.4. Estat de l'art

Un cop ja hem parlat de la necessitat d'una aplicació mòbil per gestionar les despeses d'un viatge, és crucial observar l'estat del mercat en aquest àmbit i, per tant, ens hem de fixar en les aplicacions ja disponibles per al públic i quines característiques tenen, ja sigui per veure tant les virtuts com les mancances dels possibles competidors, i així crear un producte diferencial.

Per tal de fer una comparativa, s'ha creat la Taula 1 amb les característiques que es consideren principals per a una aplicació com Travudget i la dels competidors [5], a més d'altres semblants trobades en descarregar-les per provar-les.

	Travudget	Expensify	Tripcoin	TrabeePocket	Wanderlog	Tricount	Splitwise	Triplt
Viatges								
Gestió despeses								
Detalls complets despeses								
Ubicació precisa								
Informes								
Gestió pressupost								
Conversió divises								
Compartir viatge amb usuaris								
Compartir costos								

Taula 1: Taula comparativa de l'estudi de mercat. Font: Elaboració pròpia.

El control de les despeses durant un viatge pot ser una tasca complexa, ja que involucra la gestió de múltiples transaccions en diferents monedes, la catalogació

de despeses en categories específiques i el seguiment dels pressupostos establerts abans d'iniciar el viatge, o destinar un pressupost variable per a cada tipus de despesa [6].

Com es pot veure a la Taula 1, la majoria d'apps similars o bé no estan centrades en viatges i no tenen suficient nivell de detalls o bé no implementen un sistema de compartició del viatge on altres usuaris poden unir-se i compartir les despeses, portant un control de qui deu diners a qui. També és important el tema del pressupost, que sol ser inexistent o amb un valor fix, no podent tenir un valor variable segons el dia i/o la categoria de cada despesa.

El fet que fa diferencial a Travudget és bàsicament ser un bon gestor de despeses destinat als viatges, amb detalls complets d'aquestes incloent la seva ubicació precisa, poder determinar un pressupost variable i la possibilitat de compartir el viatge amb altres usuaris.

1.5. Riscos

Els principals riscos de l'aplicació són els següents:

- **Calendari tancat:** en ser un projecte amb data d'inici i fi fixada per la universitat no permet possibles canvis si hi hagués qualsevol imprevist, per tant, cal planificar bé el projecte i tenir present com pal·liar-los.
- **Falta d'experiència:** encara haver cursat les assignatures d'ER, GPS i PES, el fet de crear una aplicació mòbil des de zero és un gran repte i han sorgit i sorgiran situacions degudes a ser inexpert en l'àmbit.
- **Previsió incorrecte:** és possible que les estimacions del temps a invertir de les tasques siguin inacurades i repercuteixi negativament al pressupost i a les hores previstes.
- **Complexitat múltiples sessions:** el fet de permetre que l'aplicació funcioni a múltiples dispositius alhora ha estat una tasca complicada i pot donar problemes en un futur.
- **Dependència d'APIs externes:** el fet de dependre d'APIs gratuïtes d'internet pot comportar problemes inesperats com, per exemple, que l'aplicació deixi de funcionar si els serveis no estan disponibles o passen a ser de pagament.

- **Canvis en la tecnologia:** les innovacions tecnològiques poden afectar el disseny i la implementació del projecte. Si fossin molt notables caldrien adaptar o revisar l'aplicació.

1.6. Metodologia

L'elecció de la metodologia és molt important i determina el flux del projecte. Durant el Grau d'Enginyeria Informàtica s'expliquen les dues més importants, en cascada, on la feina es fa seqüencialment des de l'inici al final, i àgil, que mitjançant iteracions, anomenades *sprint*, es desenvolupa el software de forma incremental creant un producte funcional durant aquests, adaptant-se a canvis i millorant la col·laboració entre l'equip de treball i el client.

Degut a les raons exposades prèviament, s'ha escollit una metodologia àgil pel projecte, concretament *Scrum* [7], però cal concretar com es tracta cada aspecte.

Primerament, els rols de *Scrum Master* i de l'equip de desenvolupament són responsabilitat de l'autor del projecte, en ser un equip d'una sola persona. En canvi, la figura de Product Owner la representa el director del TFG, en ser la persona que pot sol·licitar canvis de l'aplicació i canviar els requisits.

Sobre les tasques, hi ha el *backlog* que les conté totes i també les tasques part de cada *sprint*, representades en històries d'usuari.

En relació amb la part de *Scrum* de reunions i tenint en compte que el treball és individual, totes les cerimònies com son el *daily meeting*, *sprint review* i *sprint planning*, seran fetes per l'autor del projecte individualment.

1.6.1. Eines de seguiment

Per assegurar el correcte desenvolupament del projecte cal utilitzar diverses eines de seguiment i validació de la feina feta. En aquest apartat s'exposen aquestes eines.

Per gestionar les històries d'usuari, s'ha utilitzat Taiga [8], una eina clau per crear el *backlog* amb les històries d'usuaris organitzades en èpiques i distribuïdes en els diferents *sprints*. Per saber l'estat de cada història, hi ha 4 columnes, una per cada estat possible (*New*, *In Progress*, *Ready For Test*, *Closed*).

Per la gestió de versions s'ha usat GitHub [9], una plataforma de desenvolupament basada en el sistema de control de versions Git. A més, s'ha pres GitFlow com a model de branques per organitzar el treball i gestionar les versions del codi de manera eficient.

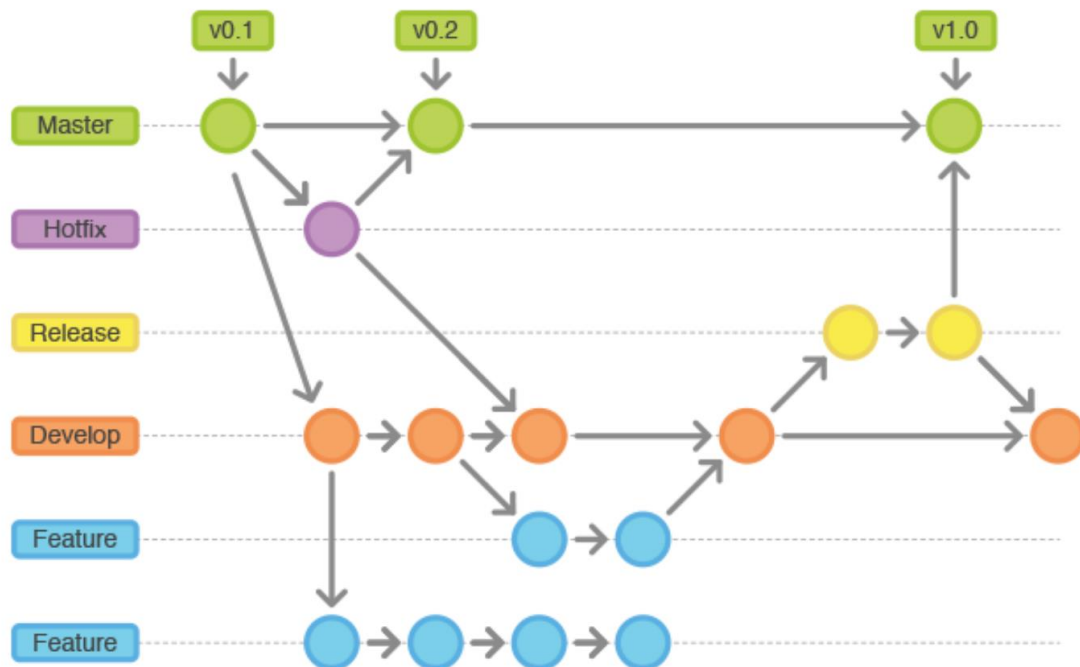


Figura 1: Flux de treball en GitFlow. Font: [10]

A part de realitzar tests de cada funcionalitat, s'ha implementat GitHub Actions [11], una eina d'integració contínua automàtica per assegurar el correcte funcionament del projecte i tenir així un sistema de validació robust.

1.7. Organització de la resta de la memòria

Un cop feta la introducció del projecte, es descriu l'estructura de la memòria i el contingut de cada capítol:

- Capítol 1: Es presenta el context del projecte, objectius, *stakeholders*, estat de l'art, els possibles riscos i la metodologia que s'ha seguit durant el transcurs del projecte.
- Capítol 2: Es defineixen i expliquen les tasques i recursos necessaris, tant els humans com els materials.
- Capítol 3: S'explica la planificació inicial segons les tasques i temps establert, amb un diagrama de Gantt per més claredat.

- Capítol 4: S'esmenten tots els costos previstos i com fer un control d'aquests.
- Capítol 5: S'explica com resoldre els problemes que puguin sorgir a partir dels riscos esmentats al capítol 1.
- Capítol 6: S'especifiquen els requisits funcionals i no funcionals, els casos d'ús i el model conceptual de les dades.
- Capítol 7: Es defineix l'arquitectura del sistema, tant física com lògica, els patrons usats a cada capa del sistema, un diagrama de seqüència d'una funcionalitat, la base de dades i com és el disseny de la interfície.
- Capítol 8: S'explica el més destacable de la implementació del projecte, les tecnologies utilitzades i com s'ha desenvolupat tot.
- Capítol 9: S'enumeren i especifiquen els diferents tipus de proves fetes per assegurar el bon funcionament del sistema.
- Capítol 10: S'explica com s'apliquen diverses lleis en aquest projecte i si han calgut llicències.
- Capítol 11: Es fa una comparació entre la planificació inicial i el resultat del desenvolupament, justificant aquest procés i com s'ha fet el seguiment.
- Capítol 12: Conté la sostenibilitat i el compromís social que s'han de prendre en consideració. S'analitza la perspectiva de sostenibilitat i compromís social.
- Capítol 12: S'analitza la sostenibilitat del projecte, tant econòmica com social i ambiental.
- Capítol 13: S'han redactat les conclusions del projecte, com s'han treballat les competències i les diverses assignatures fetes durant l'especialitat de software, així com el treball futur i possibles millores.
- Capítol 14: Conté les referències.

2. Descripció de les tasques

2.1. Tasques

Les tasques del projecte estan dividides en 3 apartats: les de gestió del projecte (GP) sobre la redacció dels continguts de GEP, la integració al document final i la preparació de la defensa; les de documentació de la memòria (DM), de caràcter tècnic; i, per últim, les tasques de desenvolupament (DEV) sobre la mateixa aplicació.

Totes les tasques de desenvolupament tenen en compte el temps a invertir en el *frontend* i *backend*, així com els tests de les classes i de la interfície necessaris.

2.1.1. Tasques de gestió del projecte

- **GP1 → Contextualització i abast:**
 - Elaboració del document on es contextualitza el projecte, es defineix l'abast, s'analitzen els competidors, es justifica la solució escollida i s'explica la metodologia del projecte.
 - Estimació: 20h. Dependències: Cap.
- **GP2 → Planificació temporal:**
 - Elaboració del document on es fa la planificació temporal, la descripció de les tasques i la gestió del risc.
 - Estimació: 20h. Dependències: GP1.
- **GP3 → Pressupost i sostenibilitat:**
 - Redacció del document sobre el pressupost i la sostenibilitat del projecte.
 - Estimació: 15h. Dependències: GP2.
- **GP4 → Integració del document final de gestió del projecte:**
 - Elaboració del document final de la gestió del projecte, unint el contingut dels documents previs amb els canvis necessaris.
 - Estimació: 20h. Dependències: GP3.
- **GP5 → Reunions varies:**
 - Les reunions amb el director que s'han tinguin durant el transcurs del projecte.
 - Estimació: 15h. Dependències: Cap.

- **GP6 → Preparació de la defensa:**
 - Preparació de la defensa del treball, incloent-hi les diapositives i preparació de l'aplicació final, amb els possibles canvis necessaris d'última hora o errors a resoldre.
 - Estimació: 30h. Dependències: GP4.

2.1.2. Tasques de documentació de la memòria

- **DM1 → Definició i diagrama de casos d'ús:**
 - Especificació de cada cas d'ús i creació del diagrama de casos d'ús.
 - Estimació: 12h. Dependències: Cap
- **DM2 → Model conceptual de les dades:**
 - Especificació del model conceptual de dades.
 - Estimació: 5h. Dependències: DM1
- **DM3 → Arquitectura física i lògica:**
 - Disseny, descripció i justificació de l'arquitectura física i lògica del sistema.
 - Estimació: 10h. Dependències: Cap
- **DM4 → Patrons utilitzats:**
 - Descripció dels patrons utilitzats durant el desenvolupament de l'aplicació.
 - Estimació: 3h. Dependències: DEV.
- **DM5 → Diagrames de seqüència:**
 - Disseny de diagrames de seqüència de 2 funcionalitats.
 - Estimació: 6h. Dependències: Cap
- **DM6 → Disseny físic i lògic de la base de dades:**
 - Disseny físic i lògic, a més de la justificació, de la base de dades.
 - Estimació: 6h. Dependències: DM2
- **DM7 → Disseny de la interfície:**
 - Disseny del diagrama amb les interfícies i les relacions entre elles.
 - Estimació: 20h. Dependències: DEV.
- **DM8 → Implementació:**

- Redacció dels aspectes important de la implementació del projecte, com la interacció entre els components, disseny dels serveis implementats, URLs...
- Estimació: 6h. Dependències: DEV.
- **DM9 → Proves:**
 - Explicació del tipus de proves realitzades i en què consisteixen.
 - Estimació: 4h. Dependències: DEV.
- **DM10 → Aspectes legals:**
 - Explicació dels aspectes legals del projecte.
 - Estimació: 3h. Dependències: Cap.
- **DM11 → Seguiment del projecte:**
 - Redacció del transcurs del projecte i els aspectes a destacar d'aquest.
 - Estimació: 3h. Dependències: DEV.
- **DM12 → Conclusions:**
 - Redacció de les conclusions del projecte.
 - Estimació: 5h. Dependències: DEV.

2.1.3. Tasques de desenvolupament

Les tasques de desenvolupament (DEV) han estat dividides per èpiques: usuaris (U), viatges (V), despeses (D), compartició (C), recompenses (R), informes (I) i deutes (DE). A més, també hi ha les tasques prèvies al desenvolupament part de l'*Inception* (INC) i la d'instanciar de la base de dades i el codi a AWS (INS) al final.

2.1.3.1. *Inception*

- **INC1 → Preparació de l'entorn:**
 - Descàrrega i instal·lació de tot el software i l'entorn necessari, incloent-hi les dependències i la creació del projecte de tant el *frontend* com el *backend*.
 - Estimació: 5h. Dependències: Cap.
- **INC2 → Configuració contenidors i connectar *backend* i *frontend*:**

- Configuració del Docker per crear el contenidor de la base de dades i del *backend*. Configuració de la connexió entre el *frontend* i *backend*, arreglant tots els possibles errors que sorgeixen.
- Estimació: 35h. Dependències: INC1.
- **INC3 → Configuració GitHub Actions:**
 - Configuració del GitHub Actions, que s'utilitza per a la integració contínua durant el desenvolupament.
 - Estimació: 5h. Dependències: INC1.
- **INC4 → Creació projecte de Taiga:**
 - Creació del projecte de Taiga, amb totes les històries d'usuari organitzades per èpiques i per sprints.
 - Estimació: 3h. Dependències: Cap.

2.1.3.2. Usuaris

- **U1 → Iniciar sessió amb Google:**
 - Creació de la base de dades dels usuaris, la interfície d'inici de sessió i la pantalla principal, i l'inici de sessió amb Google des de l'aplicació.
 - Estimació: 25h. Dependències: INC.
- **U2 → Tancar sessió:**
 - Creació de la interfície de tancar sessió i lògica del tancament de sessió.
 - Estimació: 15h. Dependències: U1.

2.1.3.3. Viatges

- **V1 → Registrar viatge:**
 - Creació de la base de dades dels viatges, la interfície amb els viatges, la de registrar un viatge i la del viatge propi, i la funcionalitat de registrar viatge amb totes les dades necessàries.
 - Estimació: 25h. Dependències: U1.
- **V2 → Editar viatge:**
 - Creació de la interfície de detalls del viatge i d'edició dels detalls, i la lògica d'edició del viatge.

- Estimació: 15h. Dependències: V1.
- **V3 → Eliminar viatge:**
 - Creació de la interfície d'eliminació del viatge i lògica de la funcionalitat.
 - Estimació: 15h. Dependències: V2.

2.1.3.4. Despeses

- **D1 → Afegir despesa:**
 - Creació de la base de dades de les despeses, la interfície d'afegir la despesa i la lògica d'afegir la despesa.
 - Estimació: 20h. Dependències: V1.
- **D2 → Editar despesa:**
 - Creació de la interfície de detalls de la despesa i de la seva edició, i la lògica d'edició.
 - Estimació: 15h. Dependències: D1.
- **D3 → Eliminar despesa:**
 - Creació de la interfície d'eliminació de la despesa i la lògica d'eliminació.
 - Estimació: 10h. Dependències: D2.
- **D4 → Filtrar despeses:**
 - Creació de la interfície de filtrar les despeses i la lògica de filtratge.
 - Estimació: 20h. Dependències: D1.

2.1.3.5. Compartició

- **C1 → Unir-se a viatge:**
 - Creació de la interfície d'unió a viatge i la funcionalitat d'unió a viatge, que inclou la sincronització de dos o més dispositius alhora, que totes les funcionalitats vagin perfectament i que els canvis que fa un usuari es vegin reflectits al dispositiu d'un altre usuari part del viatge. Es té en compte el risc de la complexitat de tenir múltiples sessions alhora i, per això, s'ha incrementat en 5-10 hores l'estimació.
 - Estimació: 35h. Dependències: V.

- **C2 → Expulsar usuari:**

- Creació de la interfície d'expulsió d'usuari i la seva lògica, tenint en compte que el canvi s'ha de reflectir en tots els dispositius que formin part del viatge.
- Estimació: 15h. Dependències: C1.

2.1.3.6. Recompenses

- **R1 → Bescanviar punts:**

- Creació de la base de dades de recompenses, la interfície de recompenses i la lògica d'obtenció de recompenses.
- Estimació: 15h. Dependències: U.

2.1.3.7. Informes

- **I1 → Consultar informes:**

- Creació de la interfície d'informes i la generació dels informes, tenint en compte els diferents informes creats d'acord amb el pressupost i despeses de cada usuari.
- Estimació: 30h. Dependències: D.

2.1.3.8. Deutes

- **DE1 → Pagar deutes:**

- Edició de la interfície de detalls del viatge per afegir-ne els deutes de cada usuari i la creació de la interfície necessària de pagament del deute, a més de la lògica dels deutes de cada usuari. És una simulació de pagament, és a dir, no hi ha passarel·la amb un servei bancari.
- Estimació: 15h. Dependències: D.

2.1.3.9. Instanciar

- **INS1 → Instanciar el *backend*:**

- Crear una instància d'EC2 pel codi de l'aplicació i una de RDS per la base de dades, a més de comprovar el seu bon funcionament amb el projecte.

- Estimació: 10h. Dependències: DEV.

2.2. Recursos

Tot seguit es presenten els recursos necessaris per dur a terme el desenvolupament del projecte correctament, tant els recursos humans com els materials, dividits en físics i software.

2.2.1. Recursos humans

Els recursos humans es divideixen en els rols del desenvolupament, portats a terme per l'autor del projecte, el tutor de GEP i el director del TFG.

- Rols de l'autor del projecte:
 - **[CAP]** Cap de projecte.
 - **[ASO]** Analista de software.
 - **[PRO]** Programador *fullstack*.
 - **[DIS]** Dissenyador UI/UX.
 - **[TES]** Tester.
- **[AUT]** Antoni Urpí Tubella, director del TFG.
- **[GEP]** Tutor de GEP.

2.2.2. Recursos materials

2.2.2.1. Recursos físics

Els recursos físics a utilitzar són els necessaris per al desenvolupament i demostració, és a dir, l'espai de treball, el portàtil i el mòbil Android.

- Espai de treball
- **[PC]** Portàtil Huawei D15
- **[MOB]** Mòbil Huawei P9 Lite

2.2.2.2. Software

Els programes/software necessaris són:

- **[VSC]** Visual Studio Code: IDE per desenvolupar el *backend*. [12]
- **[AND]** Android Studio: IDE per desenvolupar el *frontend*. [13]
- **[DOC]** Docker: contenidor del *backend*. [14]

- **[AWS]** Amazon Web Services: serveis per instanciar en línia el *backend* (EC2 [15] i RDS [16]).
- **[GIT]** Git/GitHub: controlador de versions i repositori amb tot el codi. [17]
- **[OFFICE]** Microsoft Office: eines per redactar tota la documentació. [18]
- **[GW]** Google Workspace: eines per la comunicació amb el director del TFG. [19]
- **[DRAW]** Draw.io: pàgina web per dissenyar els esquemes. [20]
- **[TAI]** Taiga.io: pàgina web per controlar el transcurs dels sprints. [21]

[ATE] Atenea: plataforma amb les fonts necessàries per saber com fer i redactar el projecte. [22]

3. Planificació inicial

En aquest apartat es defineix la planificació inicial del treball tenint en compte l'estimació de 540 hores, 90 de l'assignatura de GEP i 450 del mateix treball.

La data d'inici del projecte és el 19 de febrer de 2024 i la final és el 18 de juny de 2024, una setmana abans de l'inici del torn de lectura. Per tant, hi ha 17 setmanes i, per arribar a aquesta estimació inicial, calen unes 30 hores per setmana.

Per assolir aquesta fita, es dedicaran unes 30 hores per setmana, és a dir, unes 6 hores al dia entre setmana. En cas de no poder treballar algun dia concret, es compensarà en cap de setmana.

3.1. Estimacions

CODI	TASCA	DURACIÓ (h)	DEPEN- DÈNCIES	RECURSOS
GP	Gestió del projecte	120		
GP1	Contextualització i abast	20		PC, OFFICE, ATE
GP2	Planificació temporal	20	GP1	PC, OFFICE, ATE
GP3	Pressupost i sostenibilitat	15	GP2	PC, OFFICE, ATE
GP4	Integració del document final de gestió del projecte	20	GP3	PC, OFFICE, ATE
GP5	Reunions varies	15		PC, GW
GP6	Preparació de la defensa	30	GP4	PC, OFFICE, ATE, MOB, GW
DM	Documentació de la memòria	83		
DM1	Definició i diagrama de casos d'ús	12		PC, OFFICE, DRAW
DM2	Model conceptual de les dades	5	DM1	PC, OFFICE, DRAW
DM3	Arquitectura física i lògica	10		PC, OFFICE, DRAW

DM4	Patrons utilitzats	3	DEV	PC, OFFICE
DM5	Diagrames de seqüència	6		PC, OFFICE, DRAW
DM6	Disseny físic i lògic de la base de dades	6	DM2	PC, OFFICE, DRAW
DM7	Disseny de la interfície	20	DEV	PC, OFFICE, DRAW
DM8	Implementació	6	DEV	PC, OFFICE
DM9	Proves	4	DEV	PC, OFFICE
DM10	Aspectes legals	3		PC, OFFICE
DM11	Seguiment del projecte	3	DEV	PC, OFFICE
DM12	Conclusions	5	DEV	PC, OFFICE
DEV	Desenvolupament	318		
INC	Inception	48		
INC1	Preparació de l'entorn	5		PC, MOB, VSC, AND, GIT
INC2	Configuració contenidors i connectar <i>backend</i> i <i>frontend</i>	35	INC1	PC, MOB, VSC, AND, DOC, GIT
INC3	Configuració GitHub Actions	5	INC1	PC, VSC, AND, GIT
INC4	Creació projecte de Taiga	3		PC, TAI
U	Usuaris	40		
U1	Iniciar sessió amb Google	25	INC	PC, MOB, VSC, AND, DOC, GIT
U2	Tancar sessió	15	U1	PC, MOB, VSC, AND, DOC, GIT
V	Viatges	55		
V1	Registrar viatge	25	U1	PC, MOB, VSC, AND, DOC, GIT

V2	Editar viatge	15	V1	PC, MOB, VSC, AND, DOC, GIT
V3	Eliminar viatge	15	V2	PC, MOB, VSC, AND, DOC, GIT
D	Despeses	65		
D1	Afegir despesa	20	V1	PC, MOB, VSC, AND, DOC, GIT
D2	Editar despesa	15	D1	PC, MOB, VSC, AND, DOC, GIT
D3	Eliminar despesa	10	D2	PC, MOB, VSC, AND, DOC, GIT
D4	Filtrar despeses	20	D1	PC, MOB, VSC, AND, DOC, GIT
C	Compartició	50		
C1	Unir-se a viatge	35	V	PC, MOB, VSC, AND, DOC, GIT
C2	Expulsar usuari	15	C1	PC, MOB, VSC, AND, DOC, GIT
R	Recompenses	15		
R1	Bescanviar punts	15	U	PC, MOB, VSC, AND, DOC, GIT
I	Informes	30		
I1	Consultar informes	30	D	PC, MOB, VSC, AND, DOC, GIT
DE	Deutes	15		
DE1	Pagar deutes	15	D	PC, MOB, VSC, AND, DOC, GIT
INS	Instanciar	10		

INS1	Instanciar el <i>backend</i>	10	DEV	PC, MOB, VSC, AND, DOC, GIT, AWS
	TOTAL	531		

Taula 2: Estimació de hores, recursos i dependències de les tasques. Font: Elaboració pròpia.

3.2. Diagrama de Gantt

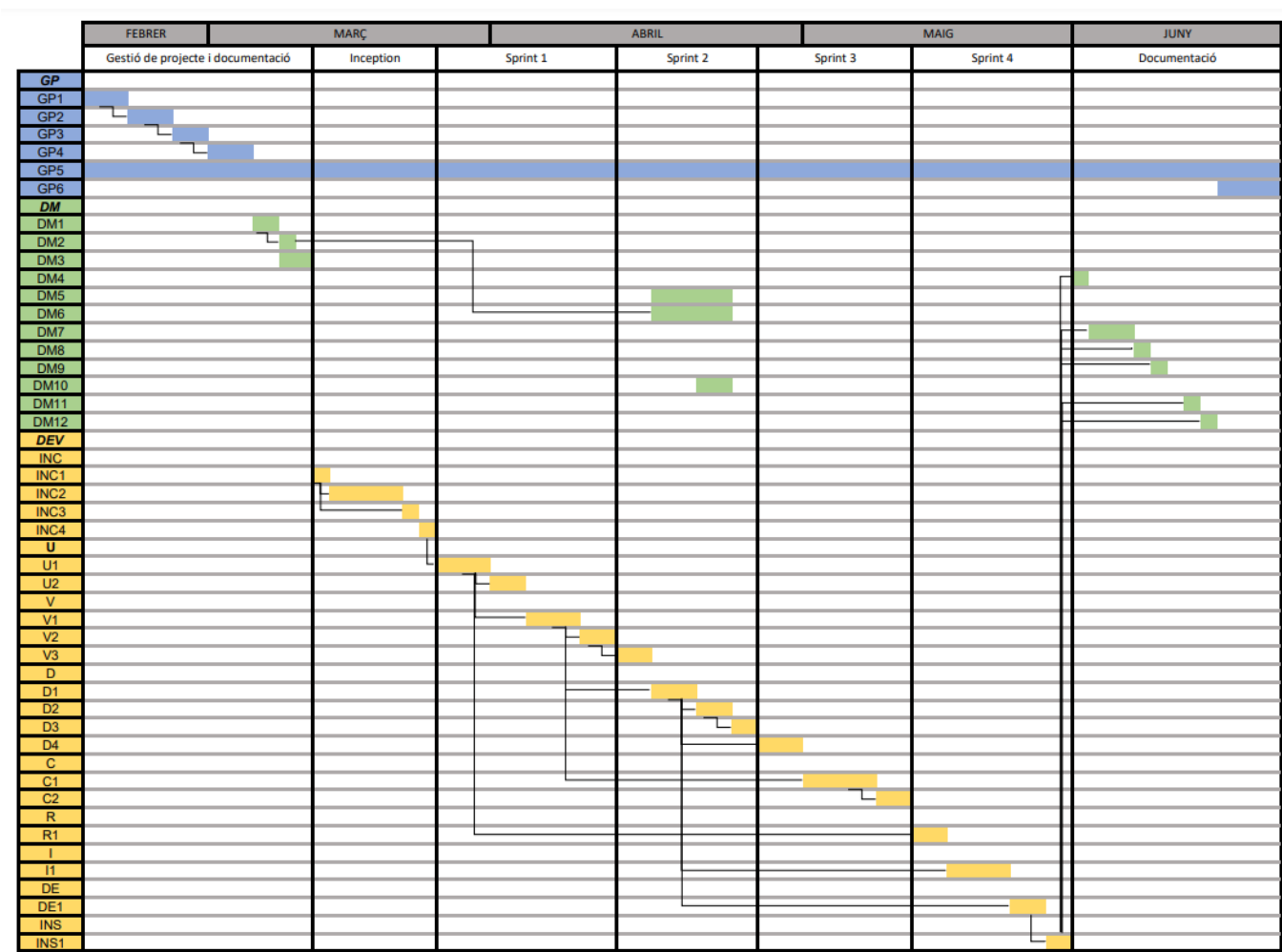


Figura 2: Diagrama de Gantt. Font: Elaboració pròpia.

4. Gestió econòmica

4.1. Identificació dels costos

Com en tot projecte, cal identificar els costos que suposa abans d'iniciar-lo, tals com els recursos humans, el software i el hardware, els possibles imprevistos i les contingències. Tot està desglossat en els següents apartats.

4.1.1. Costs de personal per activitat

A la Taula 3 es mostra el cost aproximat per hora per cada rol del projecte, extret de la pàgina Glassdoor [23] del salari mitjà a Espanya, a més de tenir en compte la cotització a la Seguretat Social del 30% sobre el salari brut.

Rol	Salari / h (€)	Salari / h + SS (€)
[CAP] Cap de projecte	21,52	27,98
[ASO] Analista de software	15,02	19,53
[PRO] Programador <i>fullstack</i>	16,23	21,10
[DIS] Dissenyador UI/UX	13,9	18,07
[TES] Tester	13,22	17,19

Taula 3: Salari per hora de cada rol de l'equip. Font: Elaboració pròpia.

Tot seguit es detalla les hores dedicades aproximades per cada rol per cada tasca del projecte:

CODI	TASCA	DURACIÓ (h)	HORES					COST (€)
			CAP	ASO	PRO	DIS	TES	
GP1	Contextualització i abast	20	20					559,6
GP2	Planificació temporal	20	20					559,6
GP3	Pressupost i sostenibilitat	15	20					419,7
GP4	Integració del document final de gestió del projecte	20	20					559,6
GP5	Reunions varies	15	-	-	-	-	-	0
GP6	Preparació de la defensa	30	30					839,4
DM1	Definició i diagrama de casos d'ús	12		12				234,36
DM2	Model conceptual de les dades	5		5				97,65
DM3	Arquitectura física i lògica	10		10				195,3
DM4	Patrons utilitzats	3		3				58,59
DM5	Diagrames de seqüència	6		6				117,18
DM6	Disseny físic i lògic de la base de dades	6		6				117,18
DM7	Disseny de la interfície	20				20		361,4
DM8	Implementació	6			8			168,8

DM9	Proves	4					4	68,76
DM10	Aspectes legals	3	3					83,94
DM11	Seguiment del projecte	3	3					83,94
DM12	Conclusions	5	5					139,9
INC1	Preparació de l'entorn	5			5			105,5
INC2	Configuració contenidors i connectar <i>backend</i> i <i>frontend</i>	35			35			105,5
INC3	Configuració GitHub Actions	5			5			738,5
INC4	Creació projecte de Taiga	3	3					105,5
U1	Iniciar sessió amb Google	25			15	2	3	83,94
U2	Tancar sessió	15			10	2	3	298,71
V1	Registrar viatge	25			20	2	3	509,71
V2	Editar viatge	15			10	2	3	298,71
V3	Eliminar viatge	15			10	2	3	298,71
D1	Afegir despesa	20			15	2	3	404,21
D2	Editar despesa	15			10	2	3	298,71
D3	Eliminar despesa	10			5	2	3	193,21
D4	Filtrar despeses	20			15	2	3	404,21
C1	Unir-se a viatge	35			28	2	5	712,89
C2	Expulsar usuari	15			10	2	3	298,71

R1	Bescanviar punts	15			10	2	3	298,71
I1	Consultar informes	30			20	5	5	598,3
DE1	Pagar deutes	15			10	2	3	298,71
INS1	Instanciar el <i>backend</i>	10			10			211
TOTAL								11227,05

Taula 4: Cost de personal per activitat. Font: Elaboració pròpia.

4.1.2. Costs generals

És important tenir en compte els costs generals també, classificats segons si són l'espai de treball, *hardware* o *software*.

Espai de treball:

Encara que el treball és realitzat per una sola persona a casa seva, s'ha estimat el preu del que costaria un espai de 30m² a una oficina a Barcelona. [24]

Espai	Preu per mes (€)	Mesos	Total (€)
30m ² d'oficina	390	4	1560

Taula 5: Cost del lloguer d'una oficina de 30 m2 a Barcelona a preu de 13€/m2. Font: Elaboració pròpia.

Hardware:

La fórmula matemàtica per obtenir les amortitzacions és la següent:

$$\frac{\text{Cost(€)} * \text{hores totals de feina}}{\text{temps per amortitzar (anys)} * \text{dies laborables per any} * \text{hores de feina al dia}}$$

On les hores totals de feina són 531, el temps per amortitzar són 4 anys, els dies laborables per any són 250 [25] i les hores de feina al dia són 6.

Tenint en compte aquests valors per calcular les amortitzacions del *hardware*, s'obtenen els resultats de la Taula 7.

CODI	HARDWARE	COST (€)	AMORTITZACIÓ(€)
PC	Portàtil Huawei D15	499	44,16

MOB	Mòbil Huawei P9 Lite	110	9,74
TOTAL			53,9

Taula 6: Cost del hardware. Font: Elaboració pròpia.

Software:

Tot el software utilitzat és gratuït o és una versió de pagament i, per tant, no té cap cost extra.

4.1.3. Costs de contingència

En projectes informàtics es calcula un cost de contingència d'entre el 10% i el 20%, per tant s'escull un 15% per aquelles despeses que no han estat anticipades.

La fórmula matemàtica per obtenir els costs de contingència és la següent:

$$\begin{aligned}
 \text{Contingències} &= (\text{Cost de personal per activitat} + \text{Costs generals}) * \frac{15}{100} \\
 &= (11227,05 + 1560 + 53,9) * 0,15 = 1926,14 \text{ €}
 \end{aligned}$$

4.1.4. Imprevistos

També es calculen els costos associats als possibles imprevistos que poden sorgir durant el desenvolupament, comentats prèviament.

IMPREVISTOS	PROBABILITAT	HORES	COST (€)
Calendari tancat	0,1	[CAP] 10	27,98
Falta d'experiència	0,25	[PRO] 10	52,75
Previsió incorrecte	0,1	[CAP] 10	27,98
Dependència d'APIs externes	0,1	[PRO] 10	21,1
Canvis en la tecnologia	0,1	[CAP] 15	41,97
TOTAL			171,78

Taula 7: Costos estimats pels imprevistos. Font: Elaboració pròpia.

4.1.5. Pressupost final

El pressupost final és resultat de sumar tots els costos anteriors.

	CPA	CG	CONTINGÈNCIES	IMPREVISTOS	TOTAL (€)
PREU	11227,05	1613,9	1926,14	171,78	14938,87

Taula 8: Estimació del pressupost final. Font: Elaboració pròpia.

4.2. Control de gestió

És important tenir un control de les despeses per saber si es compleix o no el pressupost estimat. Per tal de poder fer un eficient control de gestió, s'utilitzen les següents mètriques, que calculen la desviació entre l'estimació i la realitat de les hores per tasca, els imprevistos, el total d'hores i el cost final:

- Desviació d'hores consumides per cada tasca:
 - $(h \text{ estimades per tasca} - h \text{ reals per tasca}) * \text{cost per hora de la tasca}$
- Desviació dels imprevistos:
 - $\text{Cost d'imprevistos estimat} - \text{cost d'imprevistos real}$
- Desviació total d'hores:
 - $\text{Hores estimades} - \text{hores reals}$
- Desviació del cost final:
 - $\text{Cost final estimat} - \text{cost final real}$

5. Gestió del risc: Plans alternatius i obstacles

Per a cada risc esmentat en apartats anteriors, s'ha calculat la probabilitat de que passi, les hores a destinar per arreglar-ho i com fer-ho a la Taula 3.

Riscos	Probabilitat	Hores
Calendari tancat	Baixa	10
Falta d'experiència	Mitja	10
Previsió incorrecte	Baixa	10
Dependència d'APIs externes	Baixa	10
Canvis en la tecnologia	Baixa	15

Taula 9: Probabilitat i hores estimades de cada risc. Font: Elaboració pròpia.

5.1. Calendari tancat

Aquest risc no és tan preocupant i té una baixa probabilitat ja que es coneix des de l'inici quines són les dates d'inici i fi i ja s'ha planificat amb molta cura cada tasca. Per pal·liar aquest risc s'ha calculat unes 10 hores per acabar alguna possible tasca endarrerida.

5.2. Falta d'experiència

El fet d'haver de crear un projecte des de zero i no haver utilitzat mai algunes tecnologies, com els servidors d'AWS o fer el *frontend* en Kotlin, pot provocar algun contratemps per falta d'experiència. Es calculen unes 10 hores en cas d'haver de buscar informació i fer més proves de les pensades.

5.3. Previsió incorrecta

És possible que s'hagi previst una mica per sobre o per sota alguna tasca, a causa de la complicació d'estimar individualment les tasques a fer. Es calcula unes 10 hores en cas de necessitar-les per finalitzar alguna tasca, però és molt poc probable perquè ja s'ha tingut en compte la inexperiència i possibles contratemps.

5.4. Dependència d'APIs externes

El projecte usa certes APIs externes per alguns serveis com el canvi de divises i, en cas de sobrepassar el límit gratuït o que deixin de funcionar, caldria buscar una API alternativa, fet que es calcula en 10 hores per provar alternatives i refactoritzar el codi.

5.5. Canvis en la tecnologia

En el cas que hi hagi un canvi tecnològic molt gran i innovador que sigui molt profitós utilitzar per al projecte, caldria utilitzar-lo per l'aplicació, que és molt poc probable i caldrien unes 15 hores per aprendre a utilitzar la nova tecnologia i implementar-la.

6.Especificació de requisits

6.1. Requisits funcionals

6.1.1. Llista de requisits funcionals

L'aplicació ha de permetre fer altes d'usuaris, podent iniciar i tancar sessió, i canviar de compte.

L'aplicació ha de permetre als usuaris veure, crear, editar, i eliminar viatges, sent els seus atributs: nom, dates d'inici i final, divisa preferida, pressupost total i pressupost per dia.

L'aplicació ha de permetre als usuaris unir-se a altres viatges, així com un usuari creador d'un viatge expulsar a altres usuaris del seu.

L'aplicació ha de permetre veure, afegir, editar i eliminar despeses d'un viatge i les seus atributs: títol, descripció, preu, categoria, data, creador, ubicació i, si es vol compartir amb altres usuaris, en quin percentatge fer-ho amb cadascú.

L'aplicació ha de permetre als usuaris pagar els deutes pendents amb altres participants del viatge per despeses realitzades en aquest, sent notificat en crear-se una on deuen diners.

L'aplicació ha de permetre filtrar les despeses per diversos criteris, com el preu mínim i màxim, les categories i/o els creadors de les despeses.

L'aplicació ha de generar informes sobre les despeses d'un viatge, mostrant informació sobre el percentatge de pressupost gastat, indicant la despesa total sobre el pressupost establert prèviament, el total de despeses per dia de viatge i si aquest supera o no el pressupost del dia, el total de despeses per categoria i els deutes entre els usuaris d'un mateix viatge.

L'aplicació ha de recompensar els usuaris amb punts en completar un viatge, bescanviant aquests per codis de descompte a utilitzar en altres plataformes. L'usuari ha de poder veure els codis bescanviats prèviament.

6.1.2. Diagrames de casos d'ús

Els casos d'ús del sistema estan dividits en les 7 èpiques mencionades durant la definició de tasques els usuaris, els viatges, les despeses, la compartició de viatges, les recompenses, els informes i els deutes.

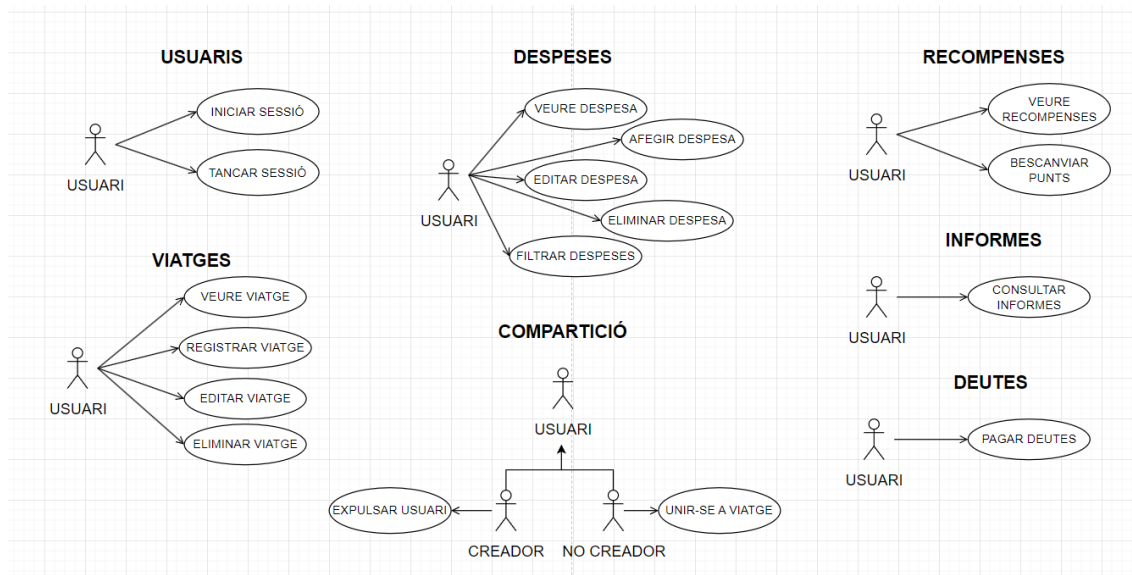


Figura 3: Diagrama de casos d'ús de les funcionalitats principals dividit per èpiques. Font: Elaboració pròpia.

6.1.3. Descripció de casos d'ús

Nom	1 – Iniciar sessió amb Google
Actor	Usuari
Descripció	L'usuari inicia sessió amb el seu compte de Google.
Precondició	El compte de Google és vàlid.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari prem el botó d'iniciar sessió amb Google. 2. El sistema redirigeix l'usuari a la pàgina d'inici de sessió de Google. 3. L'usuari introdueix les seves dades i Google retorna el <i>token</i> d'autenticació. 4. Si el <i>token</i> es vàlid, el sistema redirigeix l'usuari a la pàgina principal de l'aplicació.

Errors controlats	1. Si el sistema no pot validar el <i>token</i> d'autenticació de Google, es mostra un missatge d'error i l'usuari no pot iniciar sessió.
Postcondició	L'usuari ha iniciat sessió correctament i pot accedir a les funcionalitats de l'app.

Taula 10: Cas d'ús 1-Iniciar sessió amb Google. Font: Elaboració pròpia.

Nom	2 – Tancar sessió
Actor	Usuari
Descripció	L'usuari tanca la seva sessió a l'aplicació.
Precondició	L'usuari ha iniciat sessió a l'app.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari selecciona l'opció de tancar sessió. 2. El sistema tanca la sessió. 3. El sistema redirigeix l'usuari a la pàgina d'inici de sessió.
Errors controlats	
Postcondició	L'usuari ha tancat la seva sessió i no té accés a cap funcionalitat del sistema fins que torni a iniciar sessió.

Taula 11: Cas d'ús 2-Tancar sessió. Font: Elaboració pròpia.

Nom	3 – Veure viatge
Actor	Usuari
Descripció	L'usuari veu un viatge i les seves dades.
Precondició	L'usuari ha iniciat sessió a l'app.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari prem el viatge que vulgui consultar a la pantalla principal. 2. L'usuari prem el nom del viatge dins la pantalla del viatge. 3. El sistema mostra totes les dades del viatge.

Errors controlats	
Postcondició	El viatge es mostra amb totes les seves dades.

Taula 12: Cas d'ús 3-Veure viatge. Font: Elaboració pròpia.

Nom	4 – Registrar viatge
Actor	Usuari
Descripció	L'usuari enregistra un viatge i les seves característiques.
Precondició	L'usuari ha iniciat sessió a l'app.
Escenari principal	<ol style="list-style-type: none"> 4. L'usuari prem el botó de “Crear viatge” a la pantalla principal. 5. L'usuari introdueix les dades del viatge: el nom, les dates d'inici i fi i la divisa preferida. 6. El sistema enregistra les dades i redirigeix l'usuari a la pantalla del viatge.
Errors controlats	<ol style="list-style-type: none"> 1. Si alguna de les dades del viatge és incorrecta o falta, el sistema mostra un missatge d'error i demana a l'usuari que corregeixi les dades.
Postcondició	El viatge s'ha registrat amb èxit al sistema i està disponible per afegir despeses.

Taula 13: Cas d'ús 4-Registrar viatge. Font: Elaboració pròpia.

Nom	5 – Editar viatge
Actor	Usuari
Descripció	L'usuari edita les dades d'un viatge.
Precondició	El viatge ja existeix.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari navega a la pantalla de detalls del viatge des de la pantalla del viatge. 2. L'usuari prem el botó d'editar viatge a la pantalla de detalls del viatge.

	<ol style="list-style-type: none"> 3. L'usuari modifica les dades del viatge: el nom, les dates d'inici i fi, la divisa preferida, el pressupost total i/o el pressupost per dia. 4. L'usuari confirma els canvis realitzats. 5. El sistema valida les dades editades del viatge. 6. Si les dades són vàlides, el sistema actualitza la informació del viatge i mostra una confirmació. 7. L'usuari pot optar per continuar editant el viatge o tornar a la pantalla del viatge. <p>4.1. Si l'usuari cancel·la la sol·licitud d'edició, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	<ol style="list-style-type: none"> 1. Si alguna de les dades del viatge és incorrecta o falta, el sistema mostra un missatge d'error i demana a l'usuari que corregeixi les dades.
Postcondició	Les dades del viatge s'han actualitzat amb èxit al sistema.

Taula 14: Cas d'ús 5-Editar viatge. Font: Elaboració pròpia.

Nom	6 – Eliminar viatge
Actor	Usuari
Descripció	L'usuari elimina un viatge.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió. • L'usuari és el creador del viatge. • El viatge que es vol eliminar existeix.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari navega a la pantalla de detalls del viatge des de la pantalla del viatge. 2. L'usuari prem l'opció d'eliminar viatge a la pantalla de detalls del viatge.

	<p>3. L'usuari confirma l'eliminació.</p> <p>4. El sistema elimina el viatge seleccionat de la base de dades.</p> <p>4.1. En cas de ser un viatge ja completat, l'usuari obté punts per cada dia d'aquest viatge.</p> <p>5. El sistema mostra una confirmació de l'eliminació del viatge.</p> <p>6. L'usuari és redirigit a la pantalla principal.</p> <p>3.1. Si l'usuari cancel·la la sol·licitud d'eliminació, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	
Postcondició	El viatge s'ha eliminat amb èxit.

Taula 15: Cas d'ús 6-Eliminar viatge. Font: Elaboració pròpia.

Nom	7 – Veure despesa
Actor	Usuari
Descripció	L'usuari veu una despesa i les seves dades.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió. • Existeix el viatge registrat al sistema al qual l'usuari vol afegir una nova despesa.
Escenari principal	<p>1. L'usuari prem la despesa que vulgui consultar a la pantalla del viatge.</p> <p>2. El sistema mostra totes les dades de la despesa del viatge.</p>
Errors controlats	
Postcondició	La despesa es mostra amb totes les seves dades.

Taula 16: Cas d'ús 7-Veure despesa. Font: Elaboració pròpia.

Nom	8 – Afegir despesa
Actor	Usuari
Descripció	L'usuari afegeix una despesa a un viatge.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió. • Existeix el viatge registrat al sistema al qual l'usuari vol afegir una nova despesa.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari prem el botó de “+” a la pantalla del viatge. 2. El sistema redirigeix a la pantalla d'afegir despesa. 3. L'usuari introdueix les dades de la nova despesa: títol, descripció, preu, categoria, data, ubicació i, si es vol compartir amb altres usuaris, en quin percentatge fer-ho amb cadascú. 4. L'usuari confirma les dades de la nova despesa. 5. El sistema valida les dades introduïdes per l'usuari. 6. Si les dades són vàlides, el sistema associa la nova despesa al viatge seleccionat. 7. L'usuari torna a la pàgina del viatge. <p>3.1. Si l'usuari cancel·la la creació de la despesa, el sistema no executa cap acció i torna a la pàgina del viatge.</p>
Errors controlats	<ol style="list-style-type: none"> 1. Si alguna de les dades de la despesa és incorrecta o falta, el sistema mostra un missatge d'error i demana a l'usuari que corregeixi les dades.
Postcondició	La despesa del viatge s'ha creat i apareix a la pantalla del viatge.

Taula 17: Cas d'ús 8-Afegir despesa. Font: Elaboració pròpia.

Nom	9 – Editar despesa
Actor	Usuari
Descripció	L'usuari edita una despesa.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió. • Existeix la despesa del viatge.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari accedeix a la despesa que vol editar i prem l'opció d'"Editar". 2. L'usuari modifica qualsevol dada de la despesa. 3. L'usuari confirma els canvis realitzats a la despesa. 4. El sistema valida les dades editades de la despesa. 5. Si les dades són vàlides, el sistema actualitza la informació de la despesa i mostra una confirmació. 6. L'usuari pot optar per continuar editant la despesa o tornar a la pàgina del viatge. <p>3.1. Si l'usuari cancel·la l'edició de la despesa, el sistema no executa cap acció i torna a la pàgina del viatge.</p>
Errors controlats	<ol style="list-style-type: none"> 1. Si alguna de les dades de la despesa és incorrecta o falta, el sistema mostra un missatge d'error i demana a l'usuari que corregeixi les dades.
Postcondició	Les dades de la despesa s'han actualitzat amb èxit al sistema.

Taula 18: Cas d'ús 9-Editar despesa. Font: Elaboració pròpia.

Nom	10 – Eliminar despesa
Actor	Usuari
Descripció	L'usuari elimina una despesa.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió. • Existeix la despesa del viatge.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari accedeix a la despesa que vol eliminar i prem l'opció d'Eliminar". 2. L'usuari confirma l'eliminació. 3. El sistema elimina la despesa seleccionada de la base de dades. 4. El sistema mostra una confirmació de l'eliminació de la despesa. 5. L'usuari és redirigit a la pantalla del viatge. <p>2.1. Si l'usuari cancel·la la sol·licitud d'eliminació, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	
Postcondició	La despesa s'ha eliminat amb èxit.

Taula 19: Cas d'ús 10-Eliminar despesa. Font: Elaboració pròpia.

Nom	11 – Filtrar despeses
Actor	Usuari
Descripció	L'usuari filtra les despeses del viatge per diferents criteris, com el preu, la categoria o el creador.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió. • L'usuari ha creat un viatge.

Escenari principal	<ol style="list-style-type: none"> 1. L'usuari prem l'opció de "Filtrar" al botó d'opcions del viatge. 2. L'usuari selecciona els criteris de filtratge, com ara rang de preus, quines categories vol o qui és el creador d'aquesta despesa. 3. El sistema aplica els filtres seleccionats a les despeses del viatge. 4. El sistema mostra les despeses que compleixen els criteris de filtratge a l'usuari. <p>2.1. Si l'usuari cancel·la la sol·licitud de filtratge, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	
Postcondició	L'usuari veu les despeses del viatge que compleixen els criteris de filtratge seleccionats.

Taula 20: Cas d'ús 11-Filtrar despesa. Font: Elaboració pròpia.

Nom	12 – Unir-se a viatge
Actor	Usuari no creador i usuari creador
Descripció	L'usuari vol unir-se a un viatge d'un altre usuari.
Precondició	<ul style="list-style-type: none"> • Els usuaris 1 i 2 han iniciat sessió. • L'usuari 1 és el creador del viatge. • L'usuari 2 encara no és part del viatge.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari prem la opció d'"Unir-se a viatge" de la pantalla principal. 2. L'usuari introdueix el codi únic del viatge al qual vol unir-se. 3. El sistema verifica si el codi introduït és vàlid i correspon a un viatge existent al sistema.

	<p>4. Si el codi és vàlid, el sistema afegeix l'usuari com a participant en el viatge i actualitza la llista de participants del viatge.</p> <p>5. El sistema mostra una confirmació que l'usuari s'ha unit amb èxit al viatge.</p> <p>2.1. Si l'usuari cancel·la la sol·licitud d'unió, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	<p>1. Si el codi introduït no és vàlid o no correspon a cap viatge existent, el sistema mostra un missatge d'error indicant que el codi és invàlid i demana a l'usuari que ho torni a intentar.</p>
Postcondició	

Taula 21: Cas d'ús 12-Unir-se a viatge. Font: Elaboració pròpia.

Nom	13 – Expulsar usuari
Actor	Usuari creador i usuari no creador
Descripció	L'usuari creador expulsa un participant del viatge.
Precondició	<ul style="list-style-type: none"> Els usuaris 1 i 2 han iniciat sessió. L'usuari 1 és el creador del viatge. L'usuari 2 és participant del viatge.
Escenari principal	<p>1. L'usuari creador navega a la pantalla de detalls del viatge.</p> <p>2. L'usuari creador prem el botó d'”X” al costat de l'usuari que vol eliminar.</p> <p>3. L'usuari creador confirma la voluntat d'expulsar l'usuari.</p> <p>4. El sistema elimina l'usuari de la llista de participants del viatge.</p>

	<p>5. El sistema mostra una confirmació de l'expulsió de l'usuari.</p> <p>3.1. Si l'usuari cancel·la la sol·licitud d'expulsió, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	
Postcondició	L'usuari ha estat expulsat amb èxit del viatge i ja no apareix com a participant del viatge.

Taula 22: Cas d'ús 13-Expulsar usuari. Font: Elaboració pròpia.

Nom	14 – Veure recompenses
Actor	Usuari
Descripció	L'usuari pot veure les recompenses bescanviades prèviament en forma de codis promocionals.
Precondició	<ul style="list-style-type: none"> L'usuari ha iniciat sessió.
Escenari principal	<ol style="list-style-type: none"> L'usuari prem la opció de “Recompenses” de la pantalla principal. L'usuari veu una llista amb els codis de recompensa bescanviats prèviament.
Errors controlats	
Postcondició	L'usuari veu tots els codis bescanviats.

Taula 23: Cas d'ús 14-Veure recompenses. Font: Elaboració pròpia.

Nom	15 – Bescanviar punts
Actor	Usuari
Descripció	L'usuari pot bescanviar els seus punts per codis de descompte a utilitzar en altres plataformes.

Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió.
Escenari principal	<ol style="list-style-type: none"> 3. L'usuari prem la opció de "Recompenses" de la pantalla principal. 4. L'usuari selecciona la recompensa o avantatge que vol bescanviar amb els seus punts. 5. Si l'usuari té prou punts, el sistema els resta i mostra el codi de la recompensa a la pantalla.
Errors controlats	
Postcondició	El sistema mostra el codi de la recompensa i resta els punts al compte de l'usuari.

Taula 24: Cas d'ús 15-Bescanviar punts. Font: Elaboració pròpia.

Nom	16 – Consultar informes
Actor	Usuari
Descripció	L'usuari consulta els informes del viatge amb la informació de les despeses i el pressupost.
Precondició	<ul style="list-style-type: none"> • L'usuari ha iniciat sessió.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari prem el botó d'"Informe" a la pantalla del viatge. 2. El sistema genera l'informe basat en les dades del viatge: mostra informació sobre el percentatge de pressupost gastat, indicant la despesa total sobre el pressupost establert prèviament, el total de despeses per dia de viatge i si aquest supera o no el pressupost del dia (establint el color en verd o vermell), el total de despeses per categoria i els deutes entre els usuaris d'un mateix viatge. 3. El sistema mostra l'informe a l'usuari.
Errors controlats	

Postcondició	L'usuari consulta l'informe generat pel sistema.
--------------	--

Taula 25: Cas d'ús 16-Consultar informes. Font: Elaboració pròpia.

Nom	17 – Pagar deutes
Actor	Usuari 1 i 2
Descripció	L'usuari marca com a pagat els deutes que té amb un altre usuari del viatge.
Precondició	<ul style="list-style-type: none"> • Els usuaris 1 i 2 han iniciat sessió. • Els usuaris 1 i 2 pertanyen a un mateix viatge existent. • L'usuari 1 deu diners a l'usuari 2.
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari creador navega a la pantalla de detalls del viatge. 2. L'usuari 1 prem el botó dels diners al costat de l'usuari 2 que li vol pagar. 3. L'usuari creador confirma la voluntat de pagar. 4. El sistema elimina el deute de l'usuari 1 al 2. 5. El sistema mostra una confirmació del pagament. <p>3.1. Si l'usuari cancel·la la sol·licitud de pagament, el sistema no executa cap acció i torna a la pantalla prèvia.</p>
Errors controlats	
Postcondició	L'usuari simula el pagament d'un deute a un altre usuari.

Taula 26: Cas d'ús 17-Pagar deutes. Font: Elaboració pròpia.

6.1.4. Model conceptual de les dades

Les classes principals del sistema són els usuaris que poden crear o unir-se com a participants en un viatge. Dins aquest viatge hi ha les despeses, creades per un usuari del viatge, i que es poden incloure a altres participant del viatge com a deutor per un percentatge de l'import d'aquesta.

Els viatges també tenen l'opció de consultar el seu informe que, utilitzant dades del mateix viatge, es genera. Aquesta classe no està present al *backend* i es genera en el *frontend* cada cop que es consulta la funcionalitat, en haver utilitzat una llibreria complexa [26] que facilita la creació dels gràfics usats i ser més senzill que serialitzar el tipus de classes usades.

La suma dels deutes de les despeses entre usuaris d'un mateix viatge genera el deute total d'un viatge, on un usuari pot pagar a l'altre la quantitat que li deu. Cal recalcar que aquest pagament no està implementat amb una entitat bancària, sinó com a una simulació de pagament.

Els usuaris obtenen punts pels viatges completats, més punts com més dies tingui aquest viatge. Aquests es poden bescanviar per un sistema de recompenses simulades per, en un futur, arribar a acords amb empreses que subministrin codis reals. Aquestes recompenses són en forma de codis promocionals predefinitos pel desenvolupador que es podrien utilitzar com a rebaixes o beneficis pels usuaris de l'aplicació a altres portals d'empreses del sector turístic.

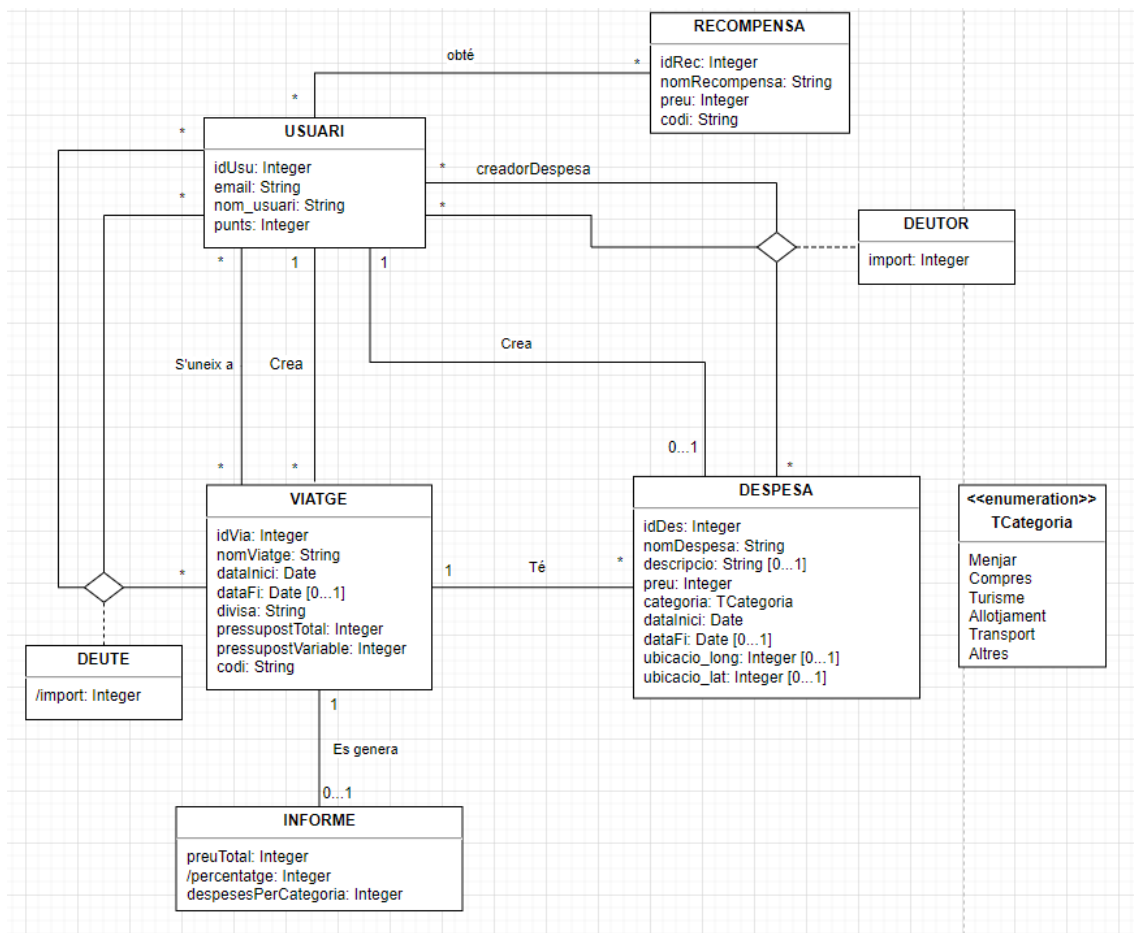


Figura 4: Diagrama del model conceptual de les dades. Font: Elaboració pròpia.

Restriccions textuais:

1. Claus externes: Usuari (idUsu, email), Viatge (idVia), Despesa (idDes), Recompensa (idRec), Informe (Viatge:idVia).
2. L'usuari creador d'un viatge i els usuaris participants que s'uneixen a aquest han de ser diferents.
3. Els deutors d'una despesa només poden ser el creador o els usuaris que s'han unit al viatge que conté la despesa.
4. Els deutes d'un viatge són entre usuaris que formin part d'aquest viatge, ja sigui el creador d'aquest o els participants.
5. La data d'inici ha de ser anterior o igual a la data de fi d'una despesa o viatge.
6. La data d'inici i fi d'una despesa ha d'estar dins les dates d'un viatge.

Informació derivada:

1. /percentatge és igual al $\text{preuTotal} / \text{pressupostTotal}$.
2. /import d'un deute és la suma de deutes de despeses amb mateix creador i deutor d'un mateix viatge.

6.2. Requisits no funcionals

Compatibilitat

- Dos usuaris d'un mateix viatge en diferents dispositius han de tenir accés a la mateixa informació i a totes les funcionalitats disponibles.

Usabilitat

- El 90% dels usuaris han d'afirmar que utilitzar l'aplicació els resulta fàcil i intuïtiu.
- El 90% dels usuaris han d'afirmar que la interfície de l'aplicació és agradable, estètica i satisfactòria.

Fiabilitat

- L'aplicació ha d'estar disponible el 95% del temps.

7.Arquitectura del sistema

7.1. Arquitectura física

El diagrama de la figura 5 mostra l'arquitectura física del sistema, on es veuen les tres capes de presentació, domini i dades ben separades.

El client es connecta a l'aplicació des de la seva app descarregada de la Google Play Store [27], que conté el *frontend*. Aquest es connecta al *backend* a través d'internet al codi que hi és a l'Amazon EC2 on s'ha fet el *deploy*, i que es connecta a la base de dades de MySQL [28] a Amazon RDS.

A més, el *frontend* també es connecta al Google Cloud [29] i Firebase [30] per l'inici de sessió de Google, a l'API de Google Maps [31] per veure el mapa per la ubicació i a la de canvis de divises per obtenir totes les divises per seleccionar la que es vulgui.

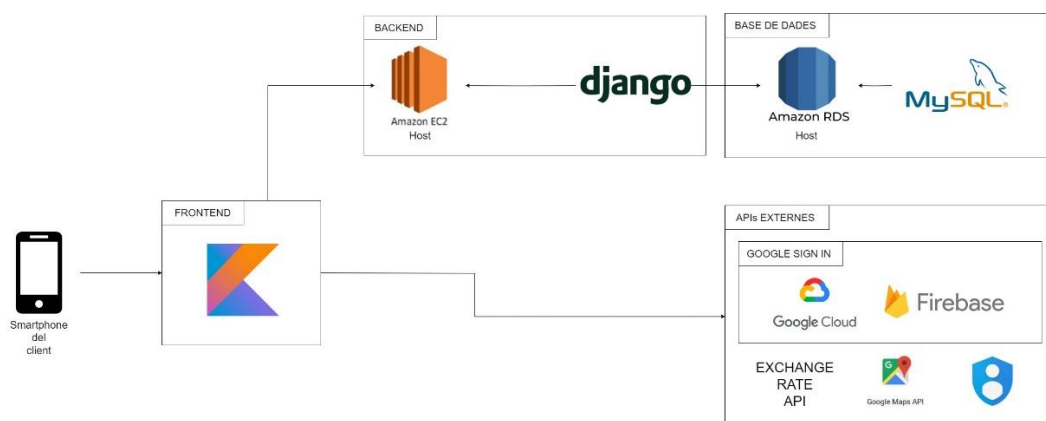


Figura 5: Arquitectura física del sistema. Font: Elaboració pròpia.

7.2. Arquitectura lògica

L'aplicació segueix un model d'arquitectura dividit en tres capes: presentació, domini i dades, on cada capa és present a un component diferenciats.

Primerament, l'usuari té al seu dispositiu mòbil descarregada l'app, que conté la interfície i la lògica de presentació. Aquesta es comunica amb la capa de domini mitjançant una connexió HTTP [32] al servidor d'Amazon EC2, on hi és el *backend* i on resideix la lògica empresarial i es processen les sol·licituds rebudes des de la capa de presentació.

Per tal d'interaccionar amb la base de dades, cal una connexió TCP/IP [33] amb el servidor d'Amazon RDS on hi és la instància de MySQL i que permet realitzar consultes, actualitzacions i altres operacions.

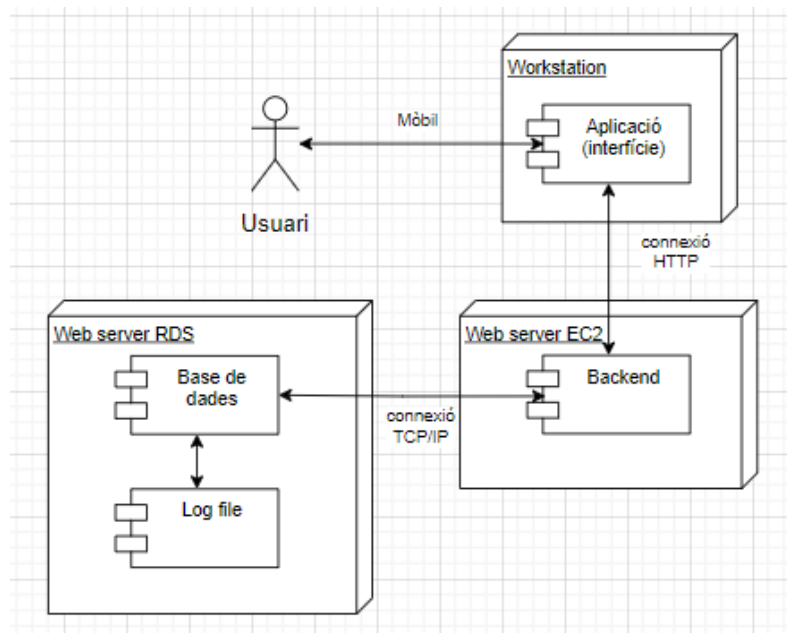


Figura 6: Diagrama d'arquitectura lògica de primer nivell. Font: Elaboració pròpia.

En resum, l'arquitectura de l'aplicació es divideix en tres capes clarament definides: presentació, domini i dades, cadascuna de les quals es troba en un component diferent. La comunicació entre aquestes capes es realitza a través de connexions HTTP entre la capa de presentació i la capa de domini, i TCP/IP entre la capa de domini i la capa de dades. Aquesta estructura proporciona una separació clara de responsabilitats i facilita el manteniment i l'escalabilitat de l'aplicació.

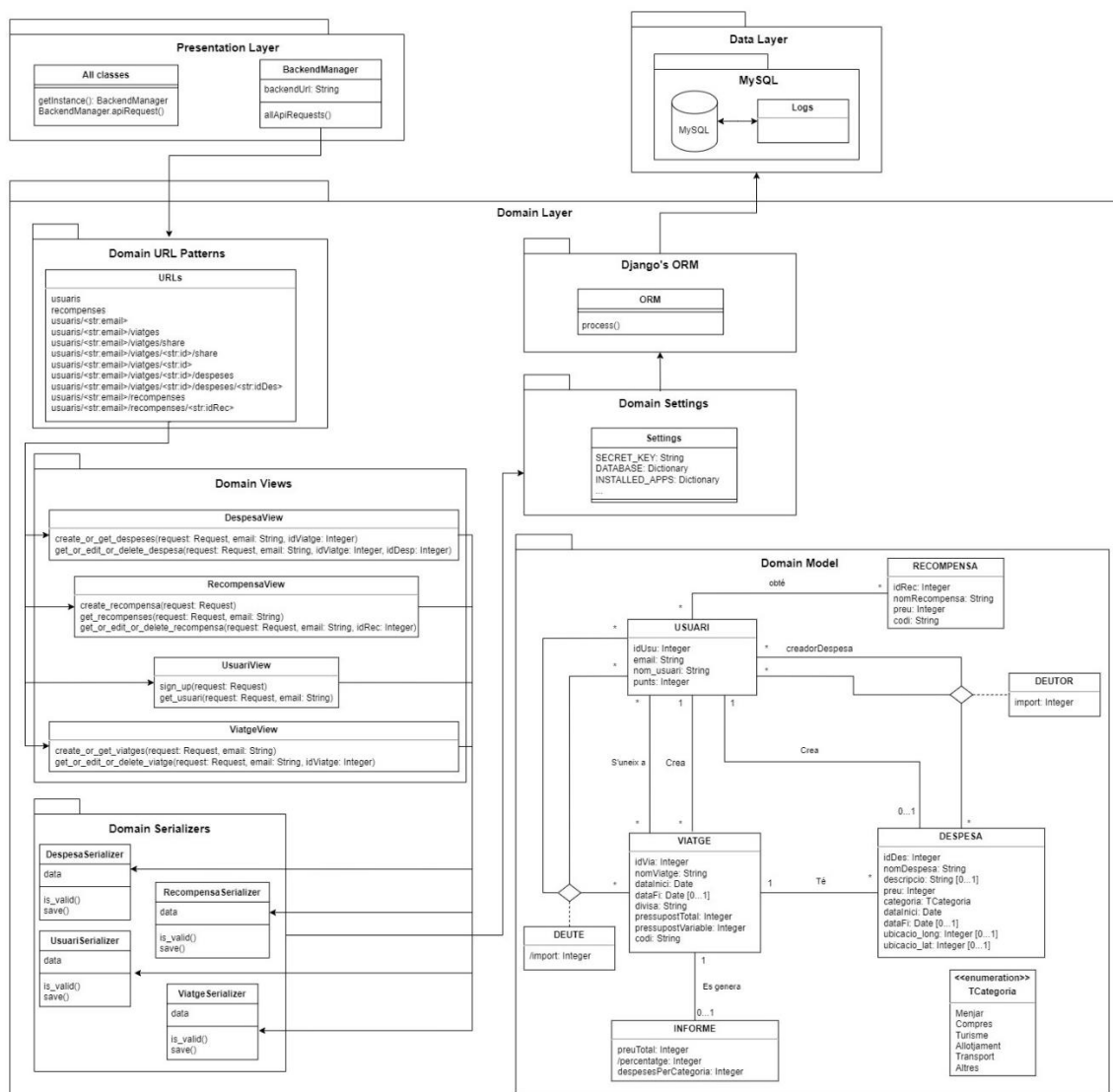


Figura 7: Diagrama d'arquitectura lògica de segon nivell. Font: Elaboració pròpia.

La figura 7 mostra l'arquitectura lògica de l'aplicació a més baix nivell dividit en les tres capes del sistema. La capa de presentació està formada per totes les classes que formen el *frontend* i la classe BackendManager, on tota la resta de classes la criden en ser l'encarregada de fer les crides API al *backend*. La capa de domini rep les peticions segons les URLs definides a l'arxiu urls.py i, si coincideix amb alguna, s'executa la vista corresponent. Aquesta vista processa la petició i utilitza el serializer del model corresponent, ja sigui per obtenir dades com per creacions o edicions.

Les peticions a la base de dades es fan a la definida a l'arxiu settings.py i l'ORM de Django transforma aquestes peticions i els models de codi Python a SQL, que són processades per la base de dades MySQL.

7.3. Patrons utilitzats

Seguint el disseny de l'aplicació de 3 capes, s'utilitzen diversos patrons a cadascuna, alguns nadius del *framework* de Django [34] o altres escollits a propòsit.

7.3.1. Patrons de capa de presentació

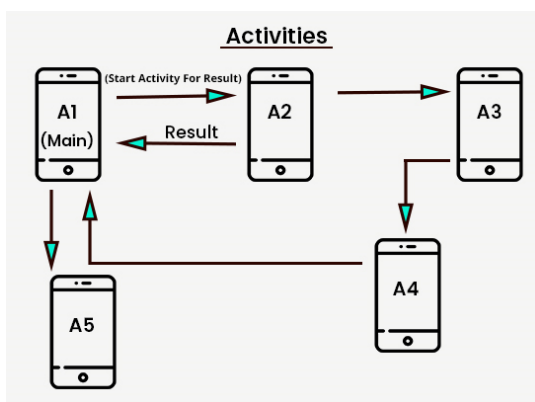


Figura 8: Exemple de Multiple Activity Pattern. Font: [35]

El *Multiple Activity Pattern* [35] és un patró que defineix utilitzar una *activity* per cada pantalla de l'aplicació. Això implica tenir una classe de tipus activitat per cada flux de l'aplicació. Cada activitat representa una pantalla diferent amb la seva pròpia interfície d'usuari i lògica associada. Aquest enfocament proporciona una manera clara i

organitzada de gestionar la navegació i la interacció de l'usuari, ja que cada activitat pot centrar-se en una tasca específica o en la presentació d'una part particular de l'aplicació. A més, permet una fàcil reutilització de codi i modularitat, ja que cada activitat pot ser desenvolupada i mantinguda de manera independent. És, per tant, més simple i modular que un model *Single Activity Approach*, on es defineix una única activitat per a tota l'aplicació.

7.3.2. Patrons capa domini

El principal patró utilitzat és el MVT [36], Model-View-Template, que és una versió pròpia de Django del comú patró de MVC (Model-View-Controller).

El patró de disseny Model-Vista-

Controlador (MVC) es basa a dividir l'app en el model, que conté la lògica de l'objecte, les vistes, que són les diverses pantalles de la

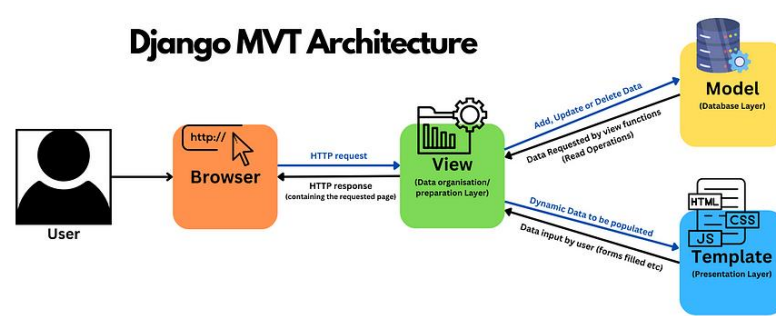


Figura 9: Patró MVT. Font: [37]

interfície i els controladors del domini. En canvi, el MVT usa *templates* HTML que controla com es mostra la informació a l'usuari i no es defineixen controladors, en actuar les vistes com si fossin controladors. En el cas d'aquest projecte no s'utilitzen aquests *templates* en tenir un *frontend* en Kotlin i no ser una pàgina web més simple.

Pel que fa a les vistes, aquestes estan formades per les diferents funcionalitats de l'aplicació, en forma d'*endpoints* per fer les crides des del *frontend*, com ara la de crear viatges, veure la llista de despeses d'un viatge o la pàgina de detalls d'aquest.

Django utilitza el patró Decorador (*Decorator*) [38] en la gestió de vistes per afegir funcionalitats addicionals sense modificar el seu codi font original, que es pot implementar mitjançant els *middlewares* [39], que actuen d'intermediari entre les parts de l'aplicació i es declaren a l'arxiu de configuració *settings.py*.

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Figura 10: Middlewares usats al *settings.py*. Font: Elaboració pròpia.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'db',
        'USER': 'pol',
        'PASSWORD': '12345678',
        'HOST': 'database-1.chcwqg4m82dj.eu-west-3.rds.amazonaws.com',
        'PORT': '3306',
    },
    'test': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'test_db.sqlite3',
    }
}
```

Figura 11: Configuració de la base de dades al *settings.py*. Font: Elaboració pròpia.

El patró Singleton s'utilitza per gestionar les connexions a la base de dades, assegurant que només hi hagi una instància d'un objecte de connexió a la base de dades, definit també a l'arxiu de configuració.

7.3.3. Patrons capa de dades

Django utilitza el patró d'Assignació Objecte-Relacional (ORM, *Object-Relational Mapping*) [40] per interactuar amb la base de dades MySQL, que proporciona un nivell més elevat d'abstracció sobre les consultes SQL tradicionals, permetent operacions de base de dades amb codi Python. Això ajuda a reduir la quantitat de codi necessari per a les operacions de la base de dades.

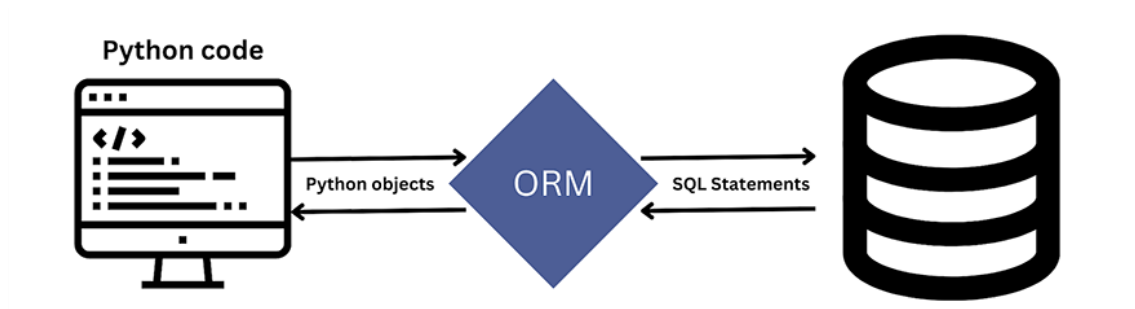


Figura 12: Diagrama del funcionament de l'ORM de Python. Font: [40]

7.4. Exemples de diagrames de seqüència

El diagrama de la Figura 13 mostra el comportament de la funció `veureViatges()`, que s'invoca un cop l'usuari ja ha iniciat sessió i hi és a la pantalla principal.

Aquesta crida al `BackendManager`, una classe que fa d'intermediària entre el *frontend* i *backend*, les funcions de `getViatges(email)` i `getViatgesParticipant(email)`, per obtenir tant els viatges que és creador com els que és un participant.

Després crea un `CardView` per cada viatge que hi ha i es mostren a l'usuari tots els creats.

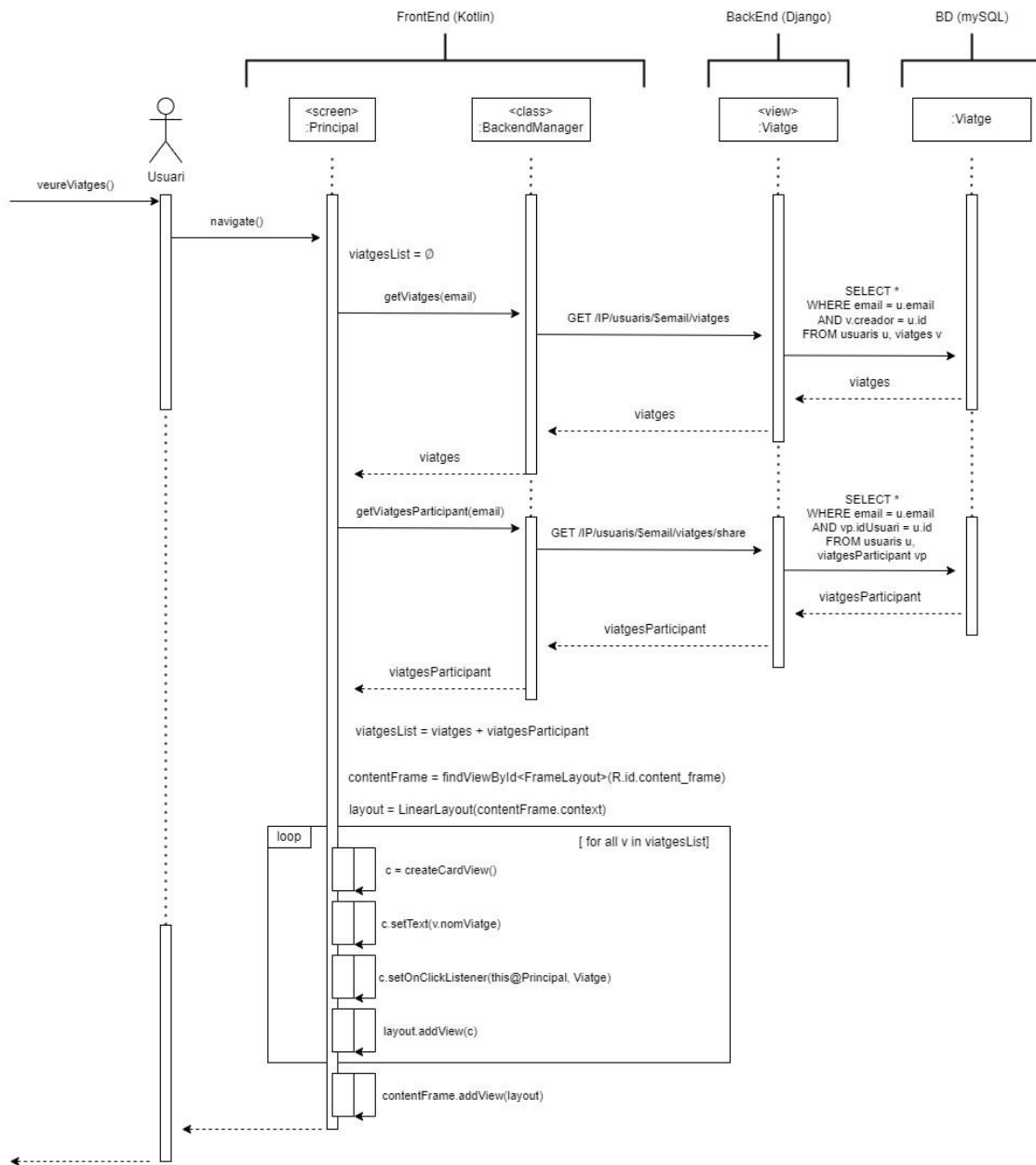


Figura 13: Diagrama de seqüència de la funcionalitat de veure viatges. Font: Elaboració pròpia.

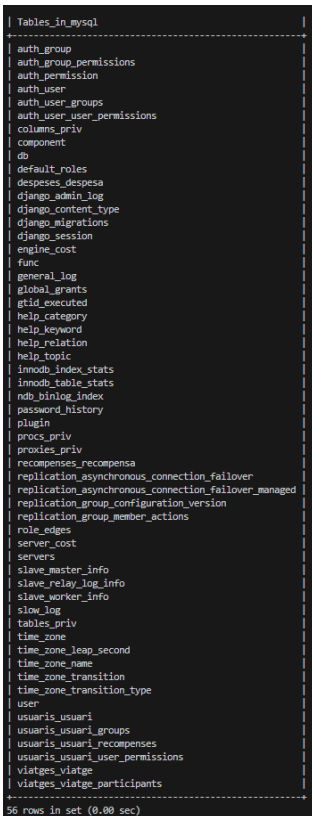
7.5. Bases de dades

La base de dades utilitzada per l'aplicació és una de model relacional SQL MySQL que resideix a Amazon RDS.

Django presenta un model de gestió de base de dades mitjançant l'ús de migracions [41], que s'usen per traduir els models definits per cada objecte en taules SQL compatibles amb la base de dades concreta que s'estigui utilitzant.

Com es veu a la figura 14, hi ha 56 taules a l'aplicació, on només 6 d'aquestes han estat creades per mi, totes les altres són creades automàticament per Django.

Són 4 per classes del sistema: despesa, recompensa, viatge i usuari; i dos per relacions molts-molts: els usuaris participants d'un viatge i les recompenses bescanviades pels usuaris.



```
Tables_in_mysql
auth_group
auth_group_permissions
auth_permission
auth_user
auth_user_groups
auth_user_user_permissions
columns_priv
component
db
default_roles
despeses_despesa
django_admin_log
django_content_type
django_migrations
django_session
engine_cost
func
general_log
global_grants
gtid_executed
help_category
help_keyword
help_relation
help_topic
innodb_index_stats
innodb_table_stats
ndb_binlog_index
password_history
plugin
procs_priv
procs_priv
recompenses_recompensa
replication_asynchronous_connection_failover
replication_asynchronous_connection_failover_managed
replication_group_configuration_version
replication_group_member_actions
role_edges
server_cost
servers
slave_master_info
slave_relay_log_info
slave_worker_info
slow_log
tables_priv
time_zone
time_zone_leap_second
time_zone_name
time_zone_transition
time_zone_transition_type
user
usuaris_usuari
usuaris_usuari_groups
usuaris_usuari_recompenses
usuaris_usuari_user_permissions
viatges_viatge
viatges_viatge_participants
56 rows in set (0.00 sec)
```

Figura 14: Taules a la base de dades MySQL.
Font: Elaboració pròpia.

En canvi, altres relacions del model conceptual com les relacions ternàries dels deutes d'un viatge i dels deutors d'una despesa són implementats com a atributs de les classes Viatge i Despesa respectivament, en forma de Map<String, Int>. En aquest cas, la clau del diccionari és la concatenació dels emails dels usuaris que formen la relació i el valor és l'import del deute.

7.5.1. Disseny lògic

El nom de cada taula és definit pel mòdul_classe i, en cas de ser relació, mòdul_classe_nomRelació:

7.5.1.1. Despesa

despeses_despesa(idDes, nomDespesa, creador, descripció, preu, categoria, data Inici, dataFi, ubicació_lat, ubicació_long, deutors, viatge)

{viatge} referencia Viatge

```

class Despesa(models.Model):
    nomDespesa = models.CharField(max_length=255)
    viatge = models.ForeignKey(Viatge, related_name='despesa_viatge', on_delete=models.CASCADE)
    creador = models.CharField(max_length=255)
    descripcio = models.CharField(max_length=255, blank=True, null=True)
    preu = models.IntegerField()
    TCategoria_choices = (
        ('Menjar', 'Menjar'),
        ('Compres', 'Compres'),
        ('Turisme', 'Turisme'),
        ('Allotjament', 'Allotjament'),
        ('Transport', 'Transport'),
        ('Altres', 'Altres'),
    )
    categoria = models.CharField(max_length=255, choices=TCategoria_choices)
    dataInici = models.DateField(blank=True, null=True)
    dataFi = models.DateField(blank=True, null=True)
    ubicacio_lat = models.FloatField(blank=True, null=True)
    ubicacio_long = models.FloatField(blank=True, null=True)
    deutors = models.JSONField(default=dict, blank=True, null=True)

```

Figura 15: Classe Despesa. Font: Elaboració pròpia.

7.5.1.2. Recompensa

recompenses_recompensa(idRec, nomRecompensa, preu, codi)

```

class Recompensa(models.Model):
    nomRecompensa = models.CharField(max_length=100)
    preu = models.IntegerField()
    codi = models.CharField(max_length=50)

```

Figura 16: Classe Recompensa. Font: Elaboració pròpia.

7.5.1.3. Usuari

usuaris_usuari(idUsu, email, nom_usuari, punts, recompenses)

usuaris_usuari_recompenses(usuari_id, recompensa_id)

{usuari_id} referencia Usuari

{recompensa_id} referencia Recompensa

```

class Usuari(AbstractUser):
    email = models.EmailField(unique=True)
    nom_usuari = models.CharField(max_length=255, blank=True, null=True)
    punts = models.IntegerField(default=10000)
    recompenses = models.ManyToManyField(Recompensa, related_name='recompenses_bescanviades', blank=True)

```

Figura 17: Classe Usuari. Font: Elaboració pròpia.

7.5.1.4. Viatge

viatges_viatge(idVia, dataInici, dataFi, divisa, pressupostTotal,
pressupostVariable, deutes, codi, creador)

{creador} referencia Usuari

viatges_viatge_participants(viatge_id, usuari_id)

{viatge_id] referencia Viatge

{usuari_id} referencia Usuari

```
class Viatge(models.Model):
    nomViatge = models.CharField(max_length=255)
    dataInici = models.DateField(null=True, blank=True)
    dataFi = models.DateField(null=True, blank=True)
    divisa = models.CharField(max_length=255)
    pressupostTotal = models.IntegerField()
    pressupostVariable = models.JSONField(default=dict)
    deutes = models.JSONField(default=dict)
    codi = models.CharField(max_length=255, unique=True)
    creador = models.ForeignKey(Usuari, related_name='viatges_creats', on_delete=models.CASCADE)
    participants = models.ManyToManyField(Usuari, related_name='viajes_on_participa', blank=True)
```

Figura 18: Classe Viatge. Font: Elaboració pròpia.

7.5.2. Disseny físic

Com s'ha esmentat prèviament, Django utilitza un sistema de migracions on tradueix la classe a una taula en executar la comanda de migració.

En ser un sistema molt eficient [42], no s'ha aplicat cap índex ni vista ni mètodes semblants.

La figura 19 mostra una migració feta per Django de la classe Recompensa, amb cada atribut, el seu tipus i les seves característiques.

```
class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Recompensa',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('nomRecompensa', models.CharField(max_length=100)),
                ('preu', models.IntegerField()),
                ('codi', models.CharField(max_length=50)),
            ],
        ),
    ]
```

Figura 19: Exemple d'una migració de la classe Recompensa. Font: Elaboració pròpia.

7.6. Disseny de la interfície

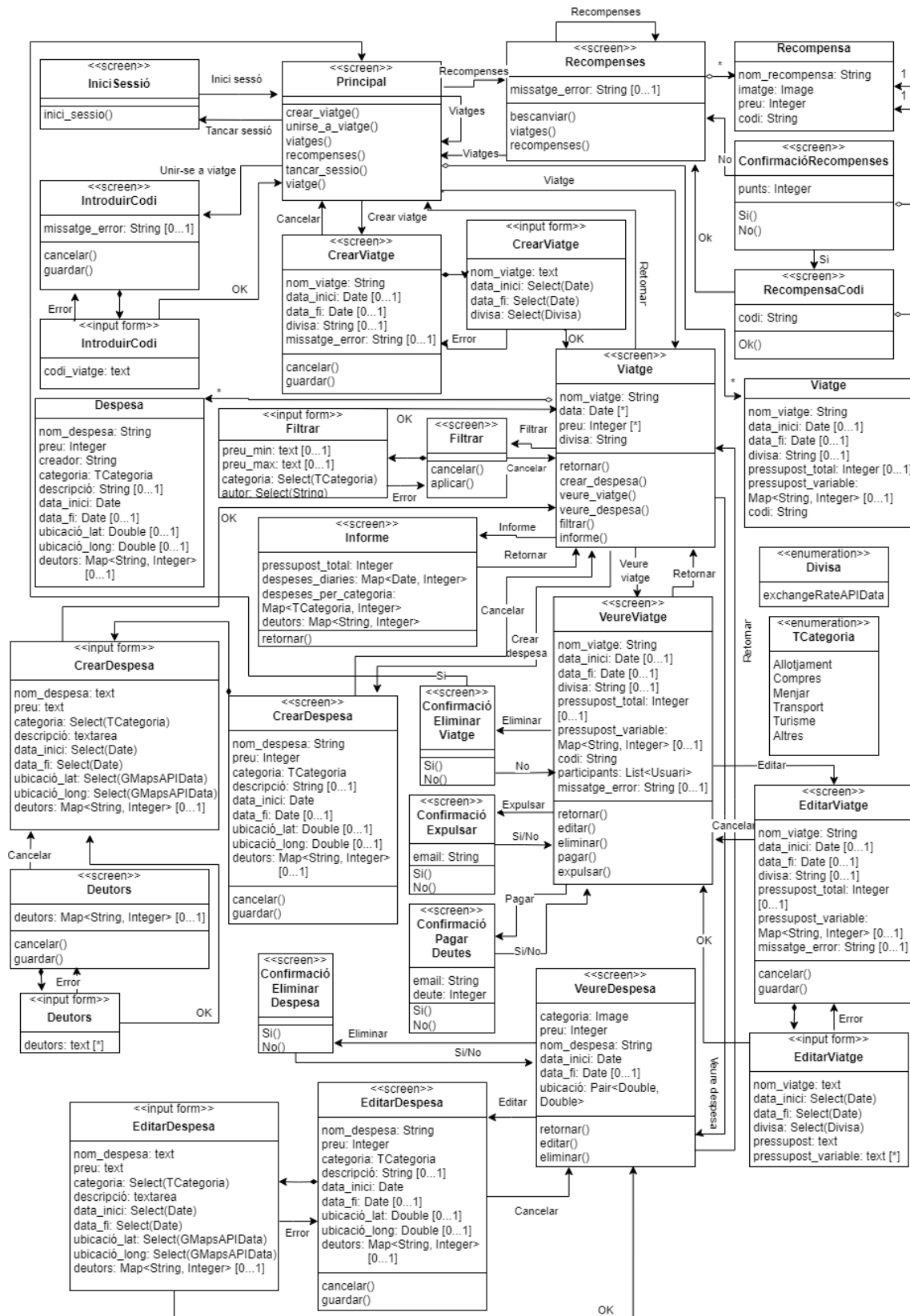


Figura 20: Diagrama amb les pantalles, classes i formularis de l'aplicació. Font: Elaboració pròpia.

8. Implementació

8.1. Tecnologies i llenguatges emprats

Els llenguatges utilitzats en aquest projecte han estat Kotlin pel *frontend* i Python per al *backend*.

Kotlin [43] és un llenguatge de programació que s'executa sobre la màquina virtual de Java (JVM). És un concís, segur i interoperable amb Java, el que significa que pot coexistir amb el codi Java en una mateixa aplicació i, fins i tot, pot usar llibreries i *frameworks* de Java directament.

Python [44] és un llenguatge de programació interpretat, de tipus dinàmic i de propòsit general que destaca per la seva llegibilitat i facilitat d'ús. Django és un *framework* de desenvolupament web de codi obert escrit en Python que ofereix eines i funcionalitats per crear aplicacions web robustes i escalables de manera eficient. Aquest segueix el principi *batteries included*, que proporciona components predefinits per a aspectes com l'autenticació, el control d'accés, la gestió de formularis o la interacció amb bases de dades.

Com esmentat al punt 7.1, aquest projecte utilitza algunes APIs externes, Google Cloud Identity, Google Firebase, Google Maps i exchangeratesapi.io [45].

Per fer funcionar les APIs de Google Maps i Cloud Identity, cal configurar primer Google Firebase, una plataforma de desenvolupament d'apps que cal utilitzar per tenir accés a les APIs de Google Cloud. En tenir tres apps, cal definir tres perfils, amb diferents *packageId* però igual petjada de certificat SHA-1.

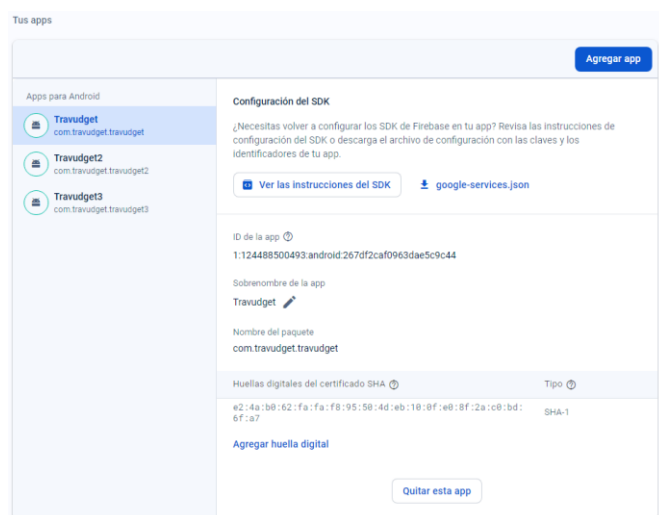


Figura 21: Configuració del projecte de Google Firebase amb les 3 builds. Font: Elaboració pròpia.

A més, cal indicar quins proveïdors d'accés es permeten, en aquest cas va ser només Google en utilitzar les seves APIs, però podrien ser també Facebook o Microsoft.

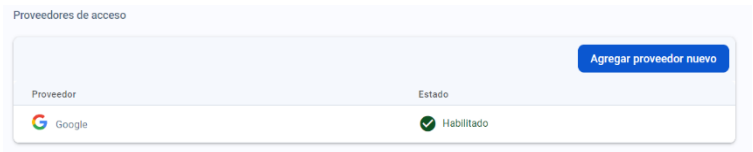


Figura 22: Proveïdors d'accés de Google Firebase, amb la de Google habilitada. Font: Elaboració pròpia.

A la plataforma de Google Cloud, cal habilitar les APIs que es vulguin utilitzar, en aquest cas la Cloud Identity per l'inici de sessió i la Maps SDK for Android pel Google Maps.

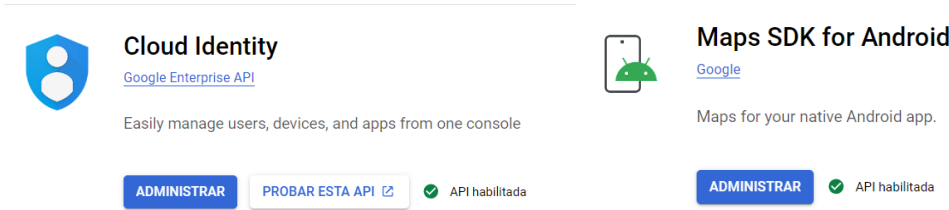


Figura 23: APIs habilitades a Google Cloud. Font: Elaboració pròpia.

Un cop es configura Google Firebase, cal definir les claus a utilitzar a la Google Cloud Console on, després de moltes proves, s'han utilitzat les claus "Maps API key" i la "Web client (auto created by Google Service)", que serveixen per connectar a l'API de Maps i la de Cloud Identity respectivament.

Credenciales [+ CREAR CREDENCIALES](#) [BORRAR](#) [RESTABLECER CREDENCIALES BORRADAS](#)

Claves de API

<input type="checkbox"/>	Nombre	Fecha de creación ↓	Restricciones	Acciones
<input type="checkbox"/>	▲ Clave de API 3	17 abr 2024	Ninguno	MOSTRAR CLAVE ⋮
<input type="checkbox"/>	▲ Android key (auto created by Firebase)	15 abr 2024	Ninguno	MOSTRAR CLAVE ⋮
<input type="checkbox"/>	▲ Maps API Key	14 abr 2024	Ninguno	MOSTRAR CLAVE ⋮

ID de clientes OAuth 2.0

<input type="checkbox"/>	Nombre	Fecha de creación ↓	Tipo	ID de cliente	Acciones
<input type="checkbox"/>	Web client (auto created by Google Service)	15 abr 2024	Aplicación web	124488500493-eh11...	Editar Eliminar Descargar
<input type="checkbox"/>	Android client for com.travudget.auxtravudget (auto created by Google Service)	15 abr 2024	Android	124488500493-g9at...	Editar Eliminar Descargar
<input type="checkbox"/>	Android client for com.travudget.travudget (auto created by Google Service)	15 abr 2024	Android	124488500493-ut2q...	Editar Eliminar Descargar
<input type="checkbox"/>	Web client (auto created by Google Service)	15 abr 2024	Aplicación web	124488500493-pt89...	Editar Eliminar Descargar

Cuentas de servicio [Administrar cuentas de servicio](#)

<input type="checkbox"/>	Correo electrónico	Nombre ↑	Acciones
<input type="checkbox"/>	firebase-adminsdk-7f19t@travudget-420318.iam.gserviceaccount.com	firebase-adminsdk	Editar Eliminar

Figura 24: Credenciales de Google Cloud amb les claus de les APIs. Font: Elaboració pròpia.

```
FROM python:3.9-slim-buster

WORKDIR /app

COPY . /app

ENV DJANGO_SETTINGS_MODULE=travudget.settings

RUN apt-get update && apt-get install -y \
    gcc \
    default-libmysqlclient-dev \
    pkg-config \
    && pip install daphne \
    && pip install django \
    && pip install mysqlclient \
    && pip install djangoestframework \
    && rm -rf /var/lib/apt/lists/*

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 8000
```

Figura 25: Contingut del Dockerfile. Font: Elaboració pròpia.

Durant el desenvolupament en local, s'ha usat Docker amb la seva configuració de Docker Compose per crear un entorn de desenvolupament consistent que permeti executar tant el codi del projecte com la base de dades. El Dockerfile defineix l'ús de Python 3.9, el directori on treballa y que copia del projecte, la configuració de l'arxiu settings.py i la instal·lació de les llibreries necessàries.

En canvi, el docker-compose.yml defineix dos contenidors, el del codi "api" amb el port i configuracions desitjades, i el de la base de dades "db", on es defineix la imatge i usuari d'aquesta.

```
version: '3'

services:
  api:
    build: .
    container_name: api
    ports:
      - "8000:8000"
    volumes:
      - ./travudget:/app
      - ./resources:/app/resources
    environment:
      DJANGO_SETTINGS_MODULE: travudget.settings
    depends_on:
      - db
  db:
    image: mysql:latest
    container_name: db
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: mysql
      MYSQL_DATABASE: mysql
      MYSQL_USER: mysql
      MYSQL_PASSWORD: mysql
```

Figura 26: Contingut del Dockerfile. Font: Elaboració pròpia.

Containers [Give feedback](#)

Container CPU usage ⓘ
Data unavailable at this time

Container memory usage ⓘ
Data unavailable at this time

Show charts

🔍

Search

⏮

⏭

⏸

🔍

Only show running containers





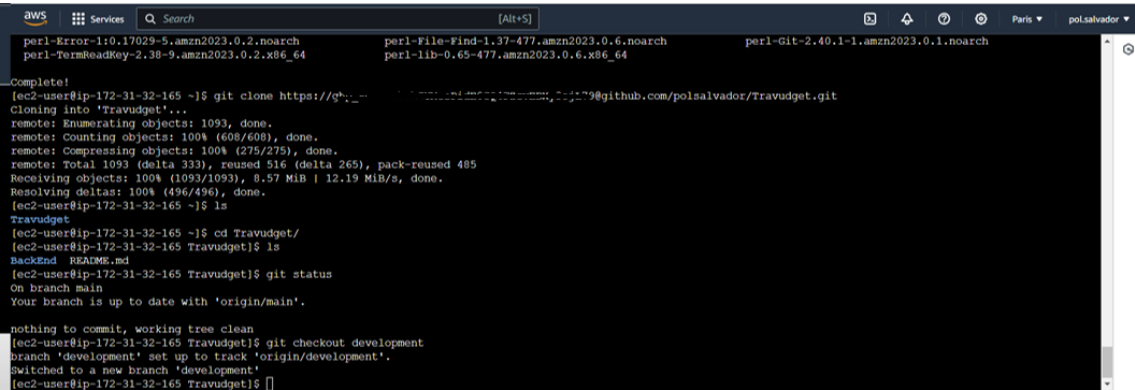
<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	<div><div>▼</div><div> backend</div></div>		Running (1/2)	N/A		15 hours ago	<div><div>■</div><div>⋮</div><div>🗑</div></div>
<input type="checkbox"/>	<div><div> api</div><div>ff5e26f6eb</div><div>backend-api</div></div>		Exited (255)	N/A	8000:8000 	2 days ago	<div><div>▶</div><div>⋮</div><div>🗑</div></div>
<input type="checkbox"/>	<div><div> db</div><div>33fb83a39c</div><div>mysql:latest</div></div>		Running	N/A		15 hours ago	<div><div>■</div><div>⋮</div><div>🗑</div></div>

Figura 27: Contenidors del backend, api i db, del Docker Desktop. Font: Elaboració pròpia.

Un cop l'aplicació ja va ser finalitzada, es va desplegar el codi de l'aplicació i la base de dades a Amazon Web Services utilitzant la versió gratuïta d'aquest. Pel codi, s'inicia i configura una instància d'EC2, i s'enxega el servidor des de la consola.



```
Complete!
[ec2-user@ip-172-31-32-165 ~]$ git clone https://github.com/polsalvador/Travudget.git
Cloning into 'Travudget'...
remote: Enumerating objects: 1093, done.
remote: Counting objects: 100% (608/608), done.
remote: Compressing objects: 100% (275/275), done.
remote: Total 1093 (delta 333), reused 516 (delta 265), pack-reused 485
Receiving objects: 100% (1093/1093), 8.57 MiB | 12.19 MiB/s, done.
Resolving deltas: 100% (496/496), done.
[ec2-user@ip-172-31-32-165 ~]$ ls
Travudget
[ec2-user@ip-172-31-32-165 ~]$ cd Travudget/
[ec2-user@ip-172-31-32-165 Travudget]$ ls
BackEnd README.md
[ec2-user@ip-172-31-32-165 Travudget]$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
[ec2-user@ip-172-31-32-165 Travudget]$ git checkout development
branch 'development' set up to track 'origin/development'.
Switched to a new branch 'development'
[ec2-user@ip-172-31-32-165 Travudget]$
```

Figura 28: Consola d'Amazon EC2. Font: Elaboració pròpia.

Per la base de dades, es crea una instància de RDS de MySQL i es configura el projecte de manera que es pugui connectar a EC2 i viceversa, tant a la plataforma d'AWS com en el codi propi.

Connectivity & security		
Endpoint & port	Networking	Security
Endpoint database-1-chnwgg4ml2d.eu-w est-3.rds.amazonaws.com	Availability Zone eu-west-3c VPC vpc-07212a986f2e5b2b3 Subnet group rds-ec2-db-subnet-group-1 Subnets subnet-0e1ccf40fe149580 subnet-073043ba0d0b648d4 subnet-042f9274794f014b Network type IPv4	VPC security groups rds-ec2-1 (sg-05b0e54c8db44108) Active Publicly accessible No Certificate authority rds-ca-rsa2048-g1 Certificate authority date May 26, 2016, 01:16 (UTC+02:00) DB instance certificate expiration date April 18, 2025, 20:40 (UTC+02:00)

Figura 29: Dades de la instància de RDS. Font: Elaboració pròpia.

8.2. Eines de desenvolupament

Les eines utilitzades durant el desenvolupament han estat Android Studio pel *frontend*, ja que és la eina més utilitzada i millor preparada per a apps mòbils per Android, i Visual Studio Code per al *backend*, en ser un IDE molt complet i permetre extensions com la de Python o Docker.

També s'ha usat l'aplicació d'escriptori Docker Desktop, fins a realitzar el llançament a AWS.

8.3. Aspectes rellevants del codi de la implementació

8.3.1. Codi de la capa de presentació

Com s'ha esmentat prèviament, el codi del *frontend* es basa en un patró *Multiple Activities*, on cada pantalla és una classe diferent (per exemple *CrearDespesa* o *Recompenses*). La majoria d'aquestes classes pertanyen a pantalles, però hi ha algunes que són importants mencionar.

La classe *BackendManager* fa de *proxy* entre el *frontend* i el *backend*, és a dir, que totes les crides de la capa de presentació a la de domini es fan a partir d'aquest.

La classe *MainActivity* és la que s'inicia en encendre l'app, i que redirigeix a la classe *IniciarSessió* si l'usuari no ha iniciat sessió, o a la classe *Principal* si és el cas.

També hi ha certes classes que actuen com *structs*, és a dir, classes serialitzables amb atributs per emmagatzemar les dades de certs objectes com *DespesaInfo*, *DespesaShowInfo*, *RecompensaInfo*, *ViatgeInfo* o *ViatgeShowInfo*.

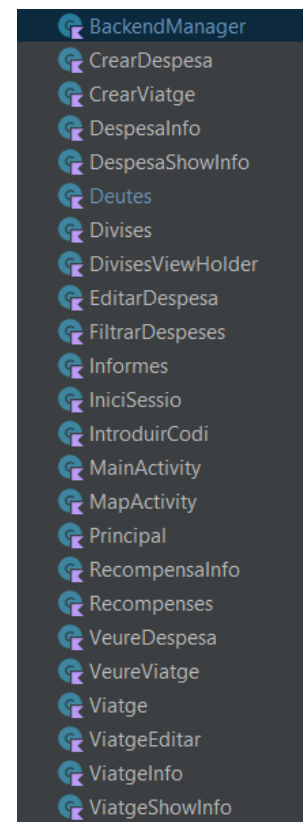


Figura 30: Classes del frontend. Font: Elaboració pròpia

```
class DespesaInfo(  
    var nomDespesa: String,  
    var viatgeId: String?,  
    var emailCreador: String?,  
    var descripcio: String?,  
    var preu: Int,  
    var categoria: String,  
    var dataInici: Date?,  
    var dataFi: Date?,  
    var ubicacio_lat: Double?,  
    var ubicacio_long: Double?,  
    var deutors: Map<String, Int>?  
) : Serializable  
  
data class DespesaShowInfo(  
    var despesaId: String?,  
    var nomDespesa: String,  
    var dataInici: Date,  
    var preu: Int,  
    var categoria: String  
) : Serializable
```

Figura 31: Classes *DespesaInfo* i *DespesaShowInfo*. Font: Elaboració pròpia.

Com es pot deduir, certs objectes tenen diferents tipus d'*structs*, ja sigui per ensenyar les seves dades a la pantalla de detalls o només per facilitar el retorn dels atributs. En alguns casos com, per exemple, *DespesaInfo* i *DespesaShowInfo*, hi ha una classe per retornar totes les dades de manera

més senzilla a la classe que ho necessiti o una classe per mostrar només una certa part dels atributs, respectivament, ja que no és necessari utilitzar-ne tots per mostrar algunes dades per pantalla, com és el cas de mostrar els viatges a la pantalla *Principal* o mostrar els detalls d'un viatge.

Per exemple, a la pantalla Principal són els viatges que un usuari ha creat o és participant. A més, hi ha el botó per obrir un panell lateral desplegable amb el nom de l'app i l'accés a altres pantalles. També hi ha un botó de "+" per crear o unir-se a altres viatges mitjançant un codi.

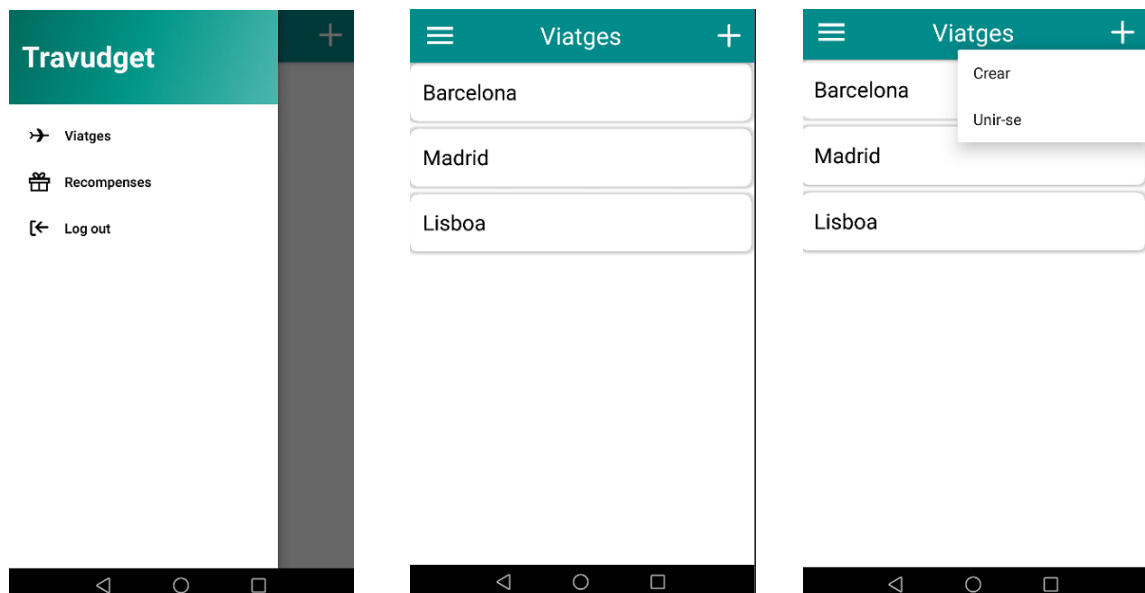


Figura 32: Pantalla Principal de l'app, amb la finestra lateral i el desplegable. Font: Elaboració pròpia.

O, entre moltes altres, hi ha la pantalla del viatge, amb totes les seves despeses, ordenades per dia i amb la suma dels costs de les diverses despeses de cada dia, a més del botó per tornar a la pantalla principal, el de crear una despesa o el de les opcions del viatge.

Com a altres pantalles a destacar hi ha la de creació d'una despesa, on l'usuari emplena els camps que desitgi, incloent òbviament els marcats com a obligatoris, per crear la despesa. Si no fos el cas, surt per pantalla un missatge emergent anomenat *Toast* [46], que avisa a l'usuari del problema.

JA FER DESPESA SELECCIONA
El nom de la despesa és obligatori

Figura 33: Exemple de missatge d'error en no escriure el nom de la despesa. Font: Elaboració pròpia.

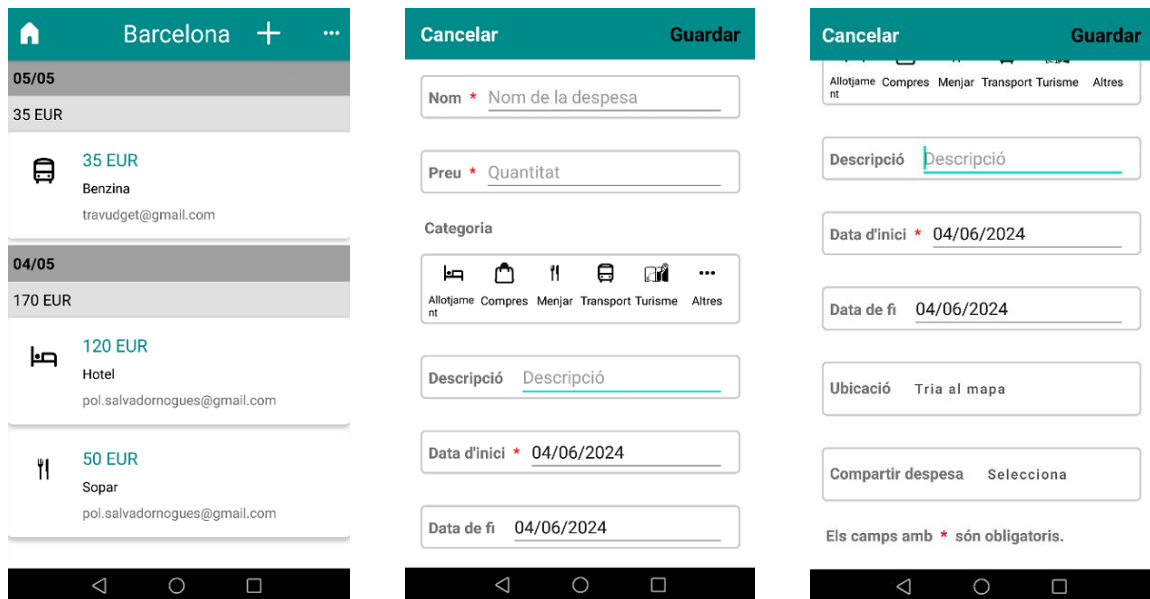


Figura 34: Pantalla d'un viatge amb les diferents despeses i la de creació d'aquestes. Font: Elaboració pròpia.

Retornar
Lleida

Divisa USD

Pressupost 500

Pressupost per dia
28/05/2024 200
29/05/2024 300

Codi QKGGZY

Participants
salvadorpol14@gmail.com
travudget@gmail.com
pol.salvadornogues@gmail.com

La figura 35 mostra la pantalla de detalls d'un viatge, on es poden veure atributs com el pressupost total o per dia, el codi per tal que altres usuaris es puguin unir al nostre viatge o la finestra de participants. Aquesta conté tots els usuaris que en formen part, amb una icona d'una moneda per pagar els deutes que tenim amb els respectius participants per totes les despeses compartides que haguem fet. A més, en cas de ser el creador del viatge, podem eliminar el viatge i expulsar al participant que desitgem.

Figura 35: Pantalla de detalls d'un viatge. Font: Elaboració pròpia.

Per últim, cal destacar la pantalla d'informe d'un viatge, on s'especifiquen el percentatge del pressupost gastat, calculat amb el total de diners gastat i el pressupost total. Tot seguit, hi ha les despeses classificades per dia i pintades de verd o vermell segons compleixin o no el pressupost per dia, respectivament.

L'últim gràfic conté les despeses classificades per categoria, amb el total gastat per cadascuna.

Finalment, hi ha els deutes del viatge on cada entrada està formada per l'usuari que deu diners, l'usuari que els ha de rebre i la quantitat.

Com a màxim nombre d'entrades són els participants del viatge, ja que se sumen i resten els diners que es deuen entre ells per tal de minimitzar el pagament dels deutes.

Per fer la transició entre activitats s'utilitza el sistema d'*Intent* [47], on s'inicia una nova activitat amb *startActivity(intent)* i s'acaba amb l'activitat actual amb *finish()*. A més, s'usen els *putExtra()* per enviar dades entre pantalles, com a la figura 37 s'està enviant l'id del viatge i l'email del creador.



Figura 36: Pantalla de l'informe d'un viatge. Font: Elaboració pròpia.

```
val intent = Intent(packageContext: this@EditarDespesa, Viatge::class.java).apply { this: Intent
    putExtra(name: "viatgeId", viatgeId)
    putExtra(name: "emailCreador", emailCreador)
}
startActivity(intent)
finish()
```

Figura 37: Exemple d'Intent per canviar de pantalla. Font: Elaboració pròpia.

Donat que per provar les funcionalitats de compartició de viatges calen almenys dos mòbils diferents i l'autor només disposava d'un, es va optar per definir 3 *builds* [48] diferents per la mateixa aplicació, ja que Android no permet instal·lar la mateixa *app* dos cops.

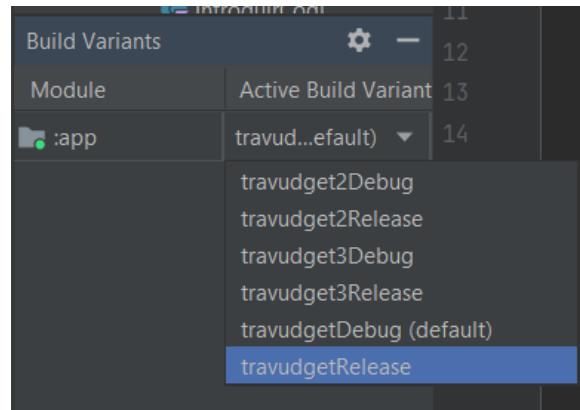


Figura 38: Selector de Build Variants a l'Android Studio. Font: Elaboració pròpia.

```
productFlavors {
    create(travudget) {
        dimension = "version"
        versionNameSuffix = "-travudget"
    }
    create(travudget2) {
        dimension = "version"
        applicationId = "com.travudget.travudget2"
        versionNameSuffix = "-travudget2"
    }
    create(travudget3) {
        dimension = "version"
        applicationId = "com.travudget.travudget3"
        versionNameSuffix = "-travudget3"
    }
}
```

Figura 39: Diferents Flavors definits al *build.gradle.kts*. Font: Elaboració pròpia.

En conseqüència, es van crear les versions *Debug* i *Release* per cada *build*, Travudget, Travudget2 i Travudget3. Aquestes són idèntiques, però canvien en l'*applicationId* utilitzant els anomenats *Flavors* d'Android que permeten justament la creació de diferents versions dins una mateixa aplicació.

Finalment, cal explicar que no s'ha pogut implementar amb èxit l'API de Google Maps on, probablement a causa d'un problema amb l'autenticació amb la clau API de Google Cloud, no és possible veure el mapa amb detall a la pantalla. El que s'ha aconseguit és mostrar un mapa buit i guardar la posició que l'usuari esculli, cosa que no és suficient amb el que s'havia plantejat en un inici.

8.3.2. Codi de la capa de domini

Els objectes del nostre sistema estan separats per *apps* [49] a Django, que permet tenir ben modularitzat i distingit cada part de l'aplicació. Per tant, el sistema consta de les *apps* per defecte de Django i les creades: usuaris, viatges, despeses i recompenses.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'usuaris',
    'viatges',
    'despeses',
]
```

Figura 40: Apps de l'aplicació al *settings.py*. Font: Elaboració pròpia.

Per accedir des del *frontend* a les funcions, s'han definit les URLs de cada funcionalitat seguint els criteris REST [50], identificant els usuaris pel seu email, els viatges per l'email del seu creador i l'id, i les despeses per id.

```
from django.urls import path
from usuarios.views import sign_up, get_usuario
from viatges.views import create_or_get_viatge, get_or_edit_or_delete_viatge, eject_viatge, get_viatges_participant_or_join
from despeses.views import create_or_get_despesa, get_or_edit_or_delete_despesa
from recompensas.views import get_or_create_recompensas, get_recompensa

urlpatterns = [
    path('usuarios', sign_up),
    path('recompensas', get_or_create_recompensas, name='get_or_create_recompensas'),
    path('usuarios/<str:email>', get_usuario, name='get_usuario'),
    path('usuarios/<str:email>/viatges', create_or_get_viatge, name='create_or_get_viatge'),
    path('usuarios/<str:email>/viatges/<str:id>', get_or_edit_or_delete_viatge, name='get_or_edit_or_delete_viatge'),
    path('usuarios/<str:email>/viatges/share', eject_viatge, name='eject_viatge'),
    path('usuarios/<str:email>/viatges/<str:id>/share', get_viatges_participant_or_join, name='get_viatges_participant_or_join'),
    path('usuarios/<str:email>/viatges/<str:id>/despeses', create_or_get_despesa, name='create_or_get_despesa'),
    path('usuarios/<str:email>/viatges/<str:id>/despeses/<str:idDes>', get_or_edit_or_delete_despesa, name='get_or_edit_or_delete_despesa'),
    path('usuarios/<str:email>/recompensas/<str:idRec>', get_recompensa, name='get_recompensa')
]
```

Figura 41: URLs definides a l'arxiu `urls.py`. Font: Elaboració pròpia.

Com a exemple del codi de la capa de domini s'ha triat per la vista d'una despesa. En ser una mateixa URL per les funcionalitats de GET, PUT i DELETE, s'obté el mètode de la petició i, segons quina és, s'executa una part del codi o una altra.

Es controlen les excepcions en cas de no existir l'usuari, el viatge o la despesa. En cas de ser correcte, es retornen les dades de la despesa si és una petició GET, s'edita la despesa i es retornen les noves dades si és un PUT, o en cas de ser un DELETE, aquesta s'elimina.

```

@api_view(['GET', 'PUT', 'DELETE'])
def get_or_edit_or_delete_despesa(request, email, id, idDes):
    if request.method == 'GET':
        try:
            usuari = Usuari.objects.get(email=email)
            viatge = Viatge.objects.get(id=int(id), creador_id=usuari.id)
            despesa = Despesa.objects.get(id=int(idDes), viatge=viatge)
            serializer = DespesaSerializer(despesa)
            return Response(serializer.data)
        except Usuari.DoesNotExist:
            return Response({"message": "L'usuari no existeix"}, status=status.HTTP_404_NOT_FOUND)
        except Viatge.DoesNotExist:
            return Response({"message": "El viatge no existeix o no pertany a l'usuari"}, status=status.HTTP_404_NOT_FOUND)
        except Despesa.DoesNotExist:
            return Response({"message": "La despesa no existeix"}, status=status.HTTP_404_NOT_FOUND)
    elif request.method == 'PUT':
        try:
            usuari = Usuari.objects.get(email=email)
            viatge = Viatge.objects.get(id=int(id), creador_id=usuari.id)
            despesa = Despesa.objects.get(id=int(idDes), viatge=viatge)
        except Usuari.DoesNotExist:
            return Response({"message": "L'usuari no existeix"}, status=status.HTTP_404_NOT_FOUND)
        except Viatge.DoesNotExist:
            return Response({"message": "El viatge no existeix o no pertany a l'usuari"}, status=status.HTTP_404_NOT_FOUND)
        except Despesa.DoesNotExist:
            return Response({"message": "La despesa no existeix"}, status=status.HTTP_404_NOT_FOUND)

        serializer = DespesaSerializer(despesa, data=request.data, partial=True)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        else:
            print(serializer)
            return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
    elif request.method == 'DELETE':
        try:
            usuari = Usuari.objects.get(email=email)
            viatge = Viatge.objects.get(id=int(id), creador_id=usuari.id)
            despesa = Despesa.objects.get(id=int(idDes), viatge=viatge)
        except Usuari.DoesNotExist:
            return Response({"message": "L'usuari no existeix"}, status=status.HTTP_404_NOT_FOUND)
        except Viatge.DoesNotExist:
            return Response({"message": "El viatge no existeix o no pertany a l'usuari"}, status=status.HTTP_404_NOT_FOUND)
        except Despesa.DoesNotExist:
            return Response({"message": "La despesa no existeix"}, status=status.HTTP_404_NOT_FOUND)

        despesa.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)

```

Figura 42: Funció de `get_or_edit_or_delete_despesa()` de la classe `Despesa`. Font: Elaboració pròpia.

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'db',
        'USER': 'pol',
        'PASSWORD': '12345678',
        'HOST': 'database-1.chcwqg4m82dj.eu-west-3.rds.amazonaws.com',
        'PORT': '3306',
    },
    'test': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'test_db.sqlite3',
    }
}

```

Figura 43: Definició de Databases al `settings.py`. Font: Elaboració pròpia.

Per tal de connectar la lògica de l'aplicació amb la base de dades, es defineix el tipus de base de dades, el nom, l'usuari, la contrasenya, el host i el port del servidor RDS.

9. Proves

Per tal d'assegurar el correcte funcionament de l'aplicació, s'han realitzat tota classe de proves [51] com tests de cada classe, d'integració, de sistema, d'UI/UX i de rendiment.

Pel que fa als tests per cada classe, s'han creat proves utilitzant la funcionalitat de *TestCase* per totes les classes del *backend*, com *Usuari*. Com es veu a la figura 44, es provenen les funcionalitats de la classe *Usuari* d'iniciar sessió si és un nou usuari o no.

```
class UsuariTestCase(TestCase):

    def setUp(self):
        self.factory = APIRequestFactory()

    def test_sign_up_created(self):
        name = "Pol Salvador"
        email = "polsalvadorupc@gmail.com"

        request = self.factory.post('/sign_up', {'name': name, 'email': email})

        response = sign_up(request)

        self.assertEqual(response.status_code, status.HTTP_201_CREATED)
        self.assertTrue(Usuari.objects.filter(email=email).exists())

    def test_sign_up_existing_user(self):
        Usuari.objects.create(email="upcfib@gmail.com", nom_usuari="Upc fib")

        request = self.factory.post('/sign_up', {'email': "upcfib@gmail.com", 'name': "Upc fib"})

        response = sign_up(request)

        self.assertEqual(response.status_code, status.HTTP_200_OK)
```

Figura 44: Fragment del *TestCase* de la classe *Usuari*. Font: Elaboració pròpia.

```
def test_get_despesa_success(self):
    email = "polsalvador@gmail.com"

    usuari = Usuari.objects.create(email=email)

    viatge = Viatge.objects.create(
        nomViatge="Viatge de prova",
        creador=usuari,
        divisa="EUR",
        pressupostTotal=200
    )

    despesa = Despesa.objects.create(
        nomDespesa="Dinar",
        viatge=viatge,
        creador=email,
        preu=20,
        categoria="Menjar",
        dataInici=date.today(),
    )

    request = self.factory.get(f'/usuarios/{email}/viatges/{viatge.id}/despeses/{despesa.id}')
    response = get_or_edit_or_delete_despesa(request, email, viatge.id, despesa.id)

    self.assertEqual(response.status_code, status.HTTP_200_OK)
```

Figura 45: Fragment del *TestCase* d'integració de la classe *Despesa*. Font: Elaboració pròpia.

A més, també s'han creat tests d'integració usant més d'una classe alhora, com és el cas d'algunes funcionalitats de *Despesa* en requerir crear un *Viatge* prèviament.

A més, tots els tests del *backend* han estat automatitzats amb GitHub Actions, de manera que s'ha assegurat una integració contínua i el correcte funcionament de tot el codi independentment del *commit* o *merge* fet, ja que s'accionava el codi de proves en fer qualsevol canvi al GitHub.

```
jobs:
  build:
    - name: Checkout repository

    - name: Set up Python
      uses: actions/setup-python@v2
      with:
        python-version: '3.9'

    - name: Install requirements
      run: pip install -r ./BackEnd/requirements.txt

    - name: Set DJANGO_SETTINGS_MODULE
      run: echo "DJANGO_SETTINGS_MODULE=travudget.settings" >> $GITHUB_ENV

    - name: Tests usuarios
      run: |
        cd ./BackEnd/travudget
        python -m usuarios.tests

    - name: Tests viatges
      run: |
        cd ./BackEnd/travudget
        python -m viatges.tests

    - name: Tests despeses
      run: |
        cd ./BackEnd/travudget
        python -m despeses.tests

    - name: Tests recompenses
      run: |
        cd ./BackEnd/travudget
        python -m recompenses.tests
```

Figura 46: Codi del `github_actions.yml`.

Font: Elaboració pròpia.

83 workflow runs		Event ▾	Status ▾	Branch ▾	Actor ▾
✓ new db	Continuous integration of the project. #83: Commit 1de137f pushed by polsalvador		development	14 hours ago 19s	...
✓ new db	Continuous integration of the project. #82: Commit ca1e97 pushed by polsalvador		development	14 hours ago 22s	...
✓ Merge pull request #8 from polsalvador/fixes	Continuous integration of the project. #81: Commit 565270e pushed by polsalvador		development	15 hours ago 35s	...

Figura 47: Fragment dels resultats del Github Actions en fer commits. Font: Elaboració pròpia.

Sobre el *frontend*, s'han fet proves de sistema fetes per l'autor del projecte provant cada funcionalitat amb el telèfon mòbil, comprovant que tot funcioni com esperat i tant el *frontend* com el *backend* i base de dades estiguin ben sincronitzats. Addicionalment, s'han fet tests d'UI/UX i de rendiment manualment debugant mitjançant el Logcat [52] d'Android Studio i la percepció de l'autor del que passa en cada moment i si succeeix el que s'esperava.

```
09:56:58.541 32126-32205 System.out      com.travudget.travudget      I Request{method=GET, url=http://13.37.108.7:8080/usuarios/salvadorpol14@gmail.com/viatges
09:56:58.541 32126-32205 System.out      com.travudget.travudget      I viatgesList: []
09:56:58.542 32126-32205 System.out      com.travudget.travudget      I []
```

Figura 48: Fragment del Logcat d'Android Studio. Font: Elaboració pròpia.

10. Aspectes legals

10.1. Lleis aplicables

Cada aplicació o sistema ha de complir amb les lleis i normatives vigents, com el Reglament General de Protecció de Dades (GDPR) i la Llei Orgànica de Protecció de Dades (LOPD) [53], que regulen el tractament de les dades personals dels usuaris. Tot i que l'aplicació no recull ni utilitza moltes dades crítiques que puguin identificar l'usuari, aquesta requereix els permisos bàsics del sistema operatiu Android, la localització per a registrar la ubicació de les despeses i l'email de cada usuari. Les dades gestionades per l'aplicació es limiten al nom de les persones i email, noms de viatges i les seves despeses, així com amb qui les comparteixen.

Per assegurar el compliment d'aquesta legislació, s'ha tingut en compte el següent:

- L'usuari, titular de les dades, ha de consentir que es tracti amb el seu email i permetre a l'aplicació Android que s'utilitzi la seva ubicació en usar l'app.
- L'usuari, en exercici dels seus drets de titularitat de les dades, pot suprimir o rectificar les dades guardades contactant amb l'equip de l'aplicació.
- Les dades de l'usuari estan emmagatzemades amb un cert nivell de seguretat per evitar que un atacant vulgui fer un ús maliciós d'aquestes.
- Aquestes dades han de mantenir-se a servidors dins de la UE, que es compleix en estar a un servidor d'AWS ubicat dins aquesta.

Aquesta aplicació es compromet amb el compliment d'aquesta llei de protecció de dades.

10.2. Llicències

Totes les plataformes i eines usades per desenvolupar l'aplicació han estat d'ús gratuït i lliure, i per això les llicències són del mateix tipus.

11. Seguiment del projecte

11.1. Canvis de planificació i pressupost

Respecte a la planificació inicial, no hi ha hagut grans contratemps, de fet, tot el contrari. En poder dedicar més de les 30h aproximades a l'inici per setmana i poder haver fet fins 50h setmanals puntualment, es va poder acabar el projecte abans de l'esperat. Això ha estat degut al fet de no haver fet pràctiques paral·lelament i realitzar només una assignatura optativa a la vegada.

Pel que fa a hores dedicades al projecte respecte a les inicials, hi ha hagut certes tasques que han estat infravalorades degut a diversos problemes:

- Afegir despesa (D1), 20h a 30h -> A causa dels intents per fer funcionar la API de Google Maps, es va haver de fer molt *debug* i molts intents fallits.
- Pagar deutes (DE1), 15h a 20h -> A l'hora de fer la simulació amb 3 perfils, van sorgir certs problemes no planificats i va necessitar més temps.

En canvi, altres han estat sobrevalorades i no han calgut tanta dedicació:

- Tancar sessió (U2), 15h a 5h -> En decidir durant el desenvolupament que no calia enregistrar al *backend* que l'usuari ja no està actiu i fer tota aquesta gestió des del *frontend*, el temps a dedicar es va reduir considerablement.
- Registrar viatge (V1), 25h a 20h -> En no donar valor a tots els atributs en crear un viatge, ja que es va decidir que aquests atributs més avançats siguin especificats en l'edició, no calia tant temps per la pantalla de creació.

Tot i això, no s'ha hagut d'augmentar el temps a dedicar ni en el pressupost, en ser les hores totals i planificades aproximadament iguals.

A més, s'han creat nous casos d'ús respecte als planificats inicialment, com són el cas de Veure viatge, Veure despesa i Veure recompenses per una qüestió de facilitar l'enteniment del sistema. No obstant això, no han resultat tampoc en un

augment d'hores finals en haver planificat el temps a dedicar-li en els casos d'ús que els contenien al principi.

11.2. Execució real

Com es va planificar a l'inici del projecte, el seguiment del desenvolupament ha estat mitjançant el Taiga, on es van crear 4 *sprints* d'unes dues o tres setmanes cadascun.

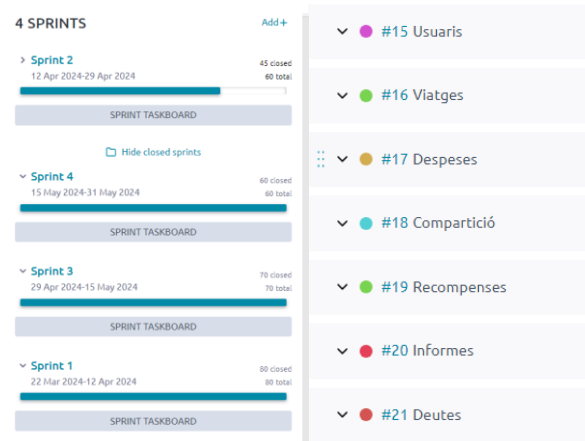


Figura 49: Els diferents sprints i èpiques del Taiga. Font: Elaboració pròpia.

Es va crear un *backlog* amb totes les històries d'usuari i les respectives tasques, actualitzant cada avanç fet i permetent així portar un control del projecte i saber la situació en tot moment.

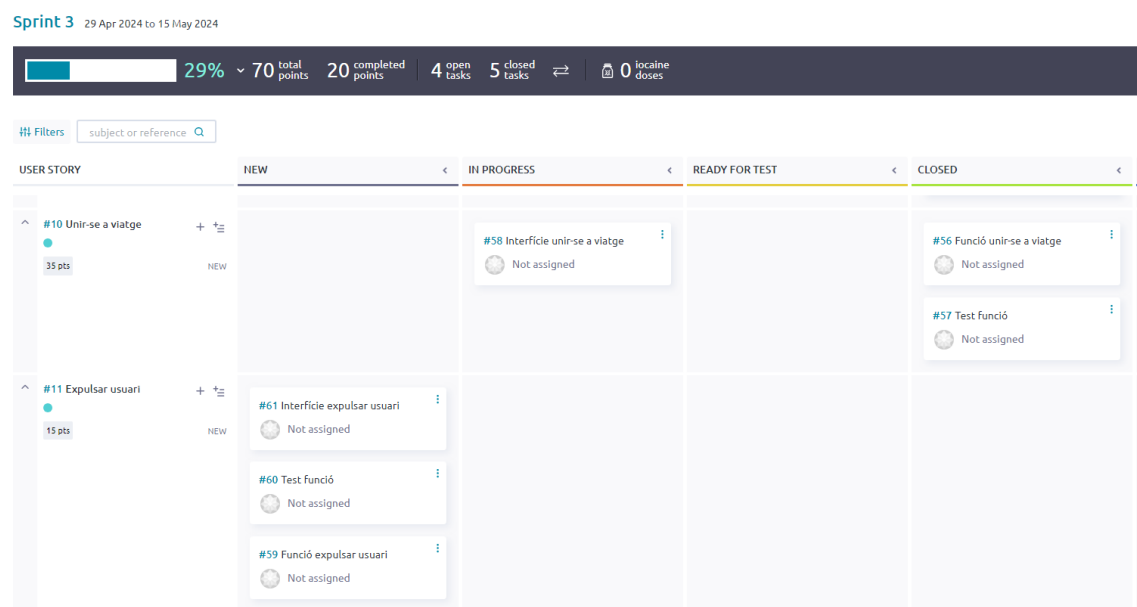


Figura 50: Fragment del Taskboard del Taiga de l'Sprint 3. Font: Elaboració pròpia.

En haver completat totes les funcionalitats previstes menys el problema de no veure el mapa de l'API de Google Maps correctament, s'ha obtingut un 94% dels resultats previstos. Per tant, s'han completat 255 punts, que equivalen a 1 hora

cadascun, i hi ha 15 sense completar, encara haver dedicat més d'aquesta tasca per intentar arreglar l'API.

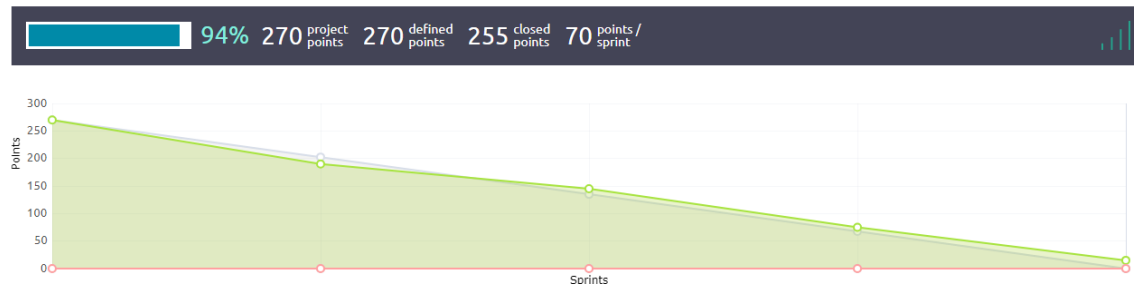


Figura 51: Gràfic del progrés del projecte. Font: Elaboració pròpia.

A més de tota la feina feta al Taiga, s'ha usat GitHub per tota la part de gestió de versions i de totes les funcionalitats.

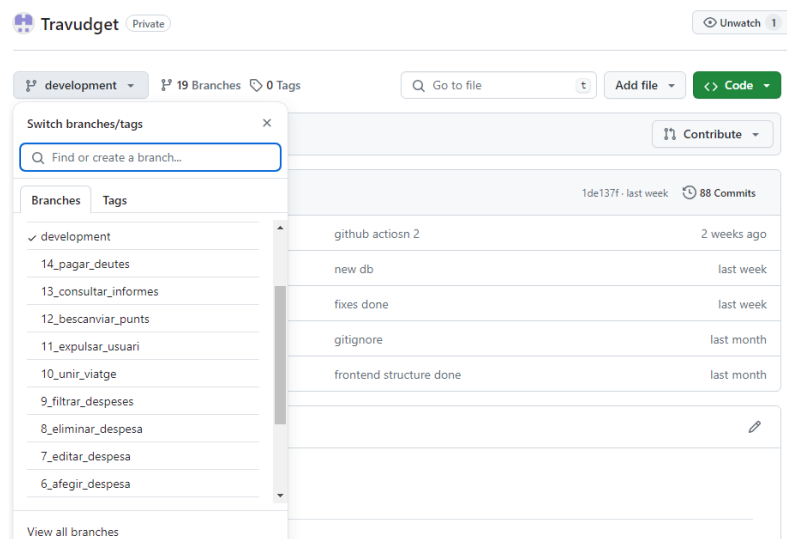


Figura 52: Captura de pantalla del repositori del GitHub. Font: Elaboració pròpia.

Finalment i per assegurar la bona execució del projecte, s'han fet reunions de seguiment entre l'autor i el director del projecte per encaminar tot el desenvolupament i documentació, per tal de poder crear el millor projecte possible.

12. Sostenibilitat

Durant totes les assignatures del GEI s'ha remarcat la importància de la sostenibilitat dels projectes informàtics, així com l'impacte ambiental, econòmic i social que representen.

Un cop haver llegit l'informe de sostenibilitat de GEI i resposta l'enquesta de EDINSOST2-ODS, s'ha pogut reflexionar sobre aquests aspectes i la seva importància. És vital veure com els tres aspectes estan relacionats entre sí i que la falta de valorar un d'ells significa un impacte negatiu en les altres dimensions i en la societat.

Hi ha aspectes claus que han estat refrescats gràcies a la enquesta, com la importància del cicle de vida dels productes i els costos ambientals directes i indirectes, que son molt importants a tenir en compte en aquests temps on és necessari cuidar el medi ambient i conscienciar els joves de l'impacte dels nostres actes al planeta.

No obstant, també cal destacar que s'ha de planificar si el producte és viable i rendible econòmicament, mitjançant tot l'estudi fet prèviament. A més, sempre s'ha d'intentar crear productes que siguin positius per la societat, on l'impacte positiu sigui més important que el negatiu.

Finalment, és important valorar tots aquests aspectes per tal de crear el millor producte possible per al medi ambient, la economia i la societat.

12.1. Econòmica

- Reflexió sobre el cost que has estimat per la realització del projecte?

S'ha creat un ampli pressupost tenint en compte tant els recursos humans com els materials, les contingències i possibles imprevistos.

- En què millorarà econòmicament (costos...) la teva solució respecte a les existents?

La millora econòmica del client probablement sigui similar a la que tindria amb aplicacions similars (que és la millor gestió de les despeses i, per tant, l'estalvi), però aquesta ofereix unes característiques i funcionalitats molt atractives que no

són en els competidors, tals com el sistema de deutes o la sessió en múltiples dispositius alhora.

- Has quantificat el cost de la realització del projecte? Quines decisions has pres per reduir el cost? Has quantificat aquest estalvi? S'ha ajustat el cost previ al cost final?

En haver trigat el mateix temps, aproximadament, que el que es va preveure, el cost es manté similar al previst. No ha calgut cap decisió per reduir el cost respecte a l'inicial ni s'ha produït cap estalvi, i per això el cost previ i final són iguals.

- Com es resolen actualment els aspectes de costos del problema que vols abordar (estat de l'art)?

Hi ha diverses aplicacions mòbils similars que resolen part del problema abordat pel projecte, però cap que integri tots els aspectes en una app.

- Quin cost estimes que tindrà el projecte durant la seva vida útil? Es podria reduir el cost per fer-lo més viable?

El cost del projecte per la seva vida útil és zero, ja que com s'exposa a les conclusions, el projecte romandrà aparcant un temps indefinit. En cas de voler fer els canvis necessaris, caldrien moltes més hores de treball i una infraestructura molt més complexa.

- Podrien produir-se escenaris que perjudiquin la viabilitat del projecte?

En cas que sorgeixi un competidor molt potent que acapari gran quota de mercat, l'aplicació perdria la motivació per implementar els canvis necessaris per a la sortida al mercat.

12.2. Social

- Què creus que t'aportarà a nivell personal la realització d'aquest projecte?

Realitzar un projecte individual d'aquestes característiques suposa treballar aptituds com la proactivitat, la responsabilitat i, òbviament, experiència en les tecnologies usades. Per tant, tot són avantatges de cara al futur món laboral.

- La realització d'aquest projecte ha implicat reflexions significatives a escala personal, professional o ètic de les persones que han intervingut?

Fer un projecte de fi de grau és, evidentment, una tasca molt important i complexa, pel fet de prendre moltes decisions sobre l'aplicació per mi mateix. És per això que ha estat experiència molt útil a escala personal, professional i ètica.

- Com es resol actualment el problema que vols abordar (estat de l'art)? i. ¿En què millorarà socialment (qualitat de vida) la teva solució respecte les existents?

L'aspecte social de l'aplicació no difereix de les ja existents molt. Es podria considerar com un aspecte molt positiu poder crear un viatge que pugui ser utilitzat per diferents usuaris alhora, que facilita la gestió en cas de viatges amb companyia.

- Existeix una necessitat real del projecte?

Es podria dir que sí que existeix una necessitat per aquest projecte per a persones que vulguin gestionar les despeses dels seus viatges i puguin utilitzar la mateixa sessió de viatge amb els seus companys, podent fins i tot compartir les despeses in situ.

- Qui es beneficiarà de l'ús del projecte? Hi ha algun col·lectiu que es vegi perjudicat pel projecte? En quina mesura?

Els usuaris de l'aplicació, persones qui viatgen soles o acompanyades, es beneficiaran de l'ús de l'aplicació. Cap col·lectiu es veu perjudicat.

- En quina mesura soluciona el projecte el problema plantejar inicialment?

El projecte soluciona el problema plantejat inicialment, una aplicació per gestionar les despeses d'un viatge amb la possibilitat de compartir aquestes amb altres usuaris.

- Es podrien produir escenaris que fessin que el projecte fos perjudicial per a algun segment particular de la població?

No es contempla cap escenari que fes que el projecte sigui perjudicial per a algun segment de la població.

12.3. Ambiental

- Has estimat l'impacte ambiental que tindrà la realització del projecte?

Tenint en compte que el producte és una aplicació mòbil i, per tant, no és un element físic, no existeix un gran impacte mediambiental. Tot i això, cal tenir present altres aspectes com l'electricitat utilitzada (obtinguda amb energies no renovables), els possibles desplaçaments dels treballadors a l'oficina (que poden ser amb vehicle propi o transport públic) o el material utilitzat (portàtil, mòbil).

- T'has plantejat minimitzar-ne el impacte, per exemple, reutilitzant recursos?

Per tal de minimitzar l'impacte mencionat anteriorment, el material utilitzat és reutilitzat d'altres projectes i els hipotètics desplaçaments si poden ser en transport públic millor.

- Has quantificat l'impacte mediambiental de la realització pel projecte? Quines mesures has pres per reduir aquest impacte? Has quantificat aquesta reducció?

L'impacte mediambiental s'ha mantingut igual que el planificat, en no haver usat més recursos dels previstos, l'ordinador, un mòbil i els servidors. Per tal de no augmentar l'impacte, només he usat un mòbil quan es podria haver usat un per cada perfil.

- Si fessis el projecte de nou, podries realitzar-lo amb menys recursos?

No, ja que el que he usat és el mínim possible per crear-lo. Tot i això, es podria no haver usat un mòbil i haver-lo simulat des de l'ordinador, però el meu portàtil no té la suficient potència per simular un mòbil i córrer el codi alhora.

- Com es resol actualment el problema que vols abordar (estat de l'art)? En què millorarà ambientalment la teva solució respecte a les existents?

Hi ha diverses aplicacions mòbils similars però no hi ha una millora ambiental substancial respecte aquestes, només es podria pensar en evitar l'ús de paper per fer aquesta gestió econòmica de despeses.

- Quins recursos estimes que s'usaran durant la vida útil del projecte? Quin serà l'impacte mediambiental d'aquests recursos?

Els recursos seran els mateixos pel que fa al desenvolupament, però si s'augmenta la infraestructura hi haurà un major impacte mediambiental en necessitar un servidor més gran i amb més potència.

- El projecte permetrà reduir l'ús d'altres recursos? Globalment, l'ús del projecte millorarà o empitjorarà la petjada ecològica?

Podem suposar que un usuari pot utilitzar l'aplicació en lloc de portar un registre a paper de totes les despeses, i per això aquesta seria una millora de la petjada ecològica.

- Podrien produir-se escenaris que fessin augmentar la petjada ecològica del projecte?

En cas de fer els canvis necessaris i augmentar la infraestructura per suportar un major flux d'usuaris, la petjada ecològica augmentarà en conseqüència.

13. Conclusions i treball futur

Aquest punt marca el fi del projecte i es fa una retrospectiva sobre l'estat del treball i s'examina si s'han assolit tots els objectius inicials. A més, s'avalua l'assoliment de les competències tècniques treballades durant el GEI i com les diferents assignatures de l'especialitat de software han aportat al projecte. Finalment, s'identifiquen possibles àrees de millora i treball futur per a l'aplicació.

13.1. Competències tècniques treballades i relació amb la especialitat de software

CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [En profunditat]

Aquesta competència ha estat àmpliament treballada i considerada, en haver creat i desenvolupat una aplicació mòbil des de zero amb una relativa complexa arquitectura de tres capes amb el *backend* que resideix als serveis d'AWS EC2 i RDS i la utilització de diverses APIs.

CES1.5: Especificar, dissenyar, implementar i avaluar bases de dades. [Bastant]

S'ha tingut en compte aquesta competència per escollir la tecnologia correcta, MySQL residida a RDS, dissenyar, implementar i avaluar la base de dades del projecte. Un sistema de base de dades relacional SQL utilitzant les eines natives de Django amb les migracions per a tenir un eficient ús d'aquesta.

CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. [Bastant]

Com a part de les bones pràctiques usades durant el desenvolupament del projecte, s'han realitzat diverses i extenses proves del codi creat amb tota mena de proves unitàries, d'integració, de sistema, d'UI/UX i de rendiment.

CES2.1: Definir i gestionar els requisits d'un sistema software. [Bastant]

Necessari per tenir clar com crear l'aplicació, es van definir extensament els requisits no funcionals i funcionals a l'inici del projecte, tenint cura de complir-los durant totes les etapes del desenvolupament.

A continuació, s'enumeren les diferents assignatures de l'especialitat de software que han contribuït al desenvolupament del projecte i com ho han fet.

- **Arquitectura del software (AS)**

Ser capaç de seleccionar i aplicar patrons d'arquitectura del software, fer el disseny dels components, crear el diagrama de seqüència i dur a terme les proves de l'aplicació tals com els tests d'integració són alguns dels aspectes d'AS treballats.

- **Aplicacions i serveis web (ASW)**

Aspectes com dissenyar l'estructura de pantalles de l'aplicació i el diagrama de pantalles, el disseny d'UI/UX del frontend, el disseny del *backend* amb REST i certes tecnologies usades són gràcies als coneixements apresos d'ASW.

- **Disseny de bases de dades (DBD)**

Els coneixements apresos a DBD han estat clarament útils per tal d'escollir la millor base de dades, de tipus relacional, com optimitzar-la si calgués i quines taules crear pel projecte.

- **Enginyeria de requisits (ER)**

Per tal de definir i analitzar els requisits funcionals i no funcionals, comprendre les necessitats del sistema i tenir clar l'estat de l'art del sector han estat clau els aspectes treballats a ER.

GPS, PES -> Planificació temporal, Diagrama de Gantt, Realització d'un projecte, Taiga/Scrum/Agile, Projecte sobre app mòbil

- **Gestió de projectes de software (GPS) i Projectes d'Enginyeria de software (PES)**

Els coneixements apresos a GPS i PES han permès el correcte desenvolupament i avanç del projecte, gràcies a competències com la correcta planificació temporal, fer un bon Diagrama de Gantt, l'ús de la metodologia Agile/Scrum i d'eines com el Taiga i, evidentment, haver fet una aplicació mòbil en grup prèviament.

13.2. Conclusions personals

Haver realitzat aquest projecte ha estat una experiència molt útil a escala personal i professional, ja que desenvolupar una aplicació mòbil d'inici a fi de forma individual m'ha ajudat a entendre tot el procés creatiu i de desenvolupament i a millorar les meves habilitats tècniques.

Crear una aplicació mòbil des de zero i encarregar-se de la conceptualització, l'especificació, el disseny, la definició de requisits, la planificació, l'arquitectura, la implementació i les proves, així com la redacció de tota la documentació, m'ha fet comprendre i entendre com és el procés d'un extrem a l'altre del desenvolupament de software.

Ha estat un projecte que en l'àmbit personal ha millorat molt les meves capacitats tècniques en molts aspectes i tecnologies, com el desenvolupament *frontend* amb Kotlin, *backend* amb Python, funcionament d'AWS i utilització d'APIs de Google. Totes aquestes habilitats segur que seran útils de cara a la meua vida professional i m'han ajudat a veure quins aspectes del desenvolupament m'interessen més i menys.

Tot i estar molt orgullós del treball fet, soc conscient d'aspectes a millorar i propostes de futur per l'aplicació, ja que en un projecte de final de grau d'un temps tan limitat, tenint en compte el temps que es triga a desenvolupar una app en la indústria, és possible que es cometin errors i no es pugui abastar massa aspectes.

Finalment, aquest projecte ha estat una demostració d'una part dels coneixements tècnics i les habilitats apreses durant el grau que posen punt i final a aquesta etapa formativa tant enriquidora com ha estat els meus estudis a la Facultat d'Informàtica de Barcelona.

13.3. Treball futur

Com a possibles camins d'aquest projecte, el més ideal seria publicar-ho a la Google Play Store i poder treure també un rendiment econòmic. Per tal de llençar l'aplicació per a una gran quantitat de clients i publicar-la, cal realitzar una sèrie de modificacions i millores.

Aquestes modificacions es poden dividir en necessàries i ideals. Les més necessitades serien millorar el rendiment de l'aplicació i el canvi de pantalles en certs moments que dificulten l'experiència d'usuari, arreglar l'API de Google Maps per tal de mostrar correctament la ubicació, implementar el pagament dels deutes en actualment ser només una simulació i modificar el sistema de recompenses de forma que siguin reals. Per arribar a obtenir aquestes recompenses, caldria arribar a acords amb companyies de viatges o turístiques per tal que proporcionin codis de descompte o referits.

Referit a aquest sistema de recompenses, seria una important addenda la creació d'un compte o perfil d'administrador amb la funcionalitat d'introduir els codis promocionals des de l'aplicació i no haver-ho de fer des del codi.

Com a idees de millora més ideals i per futures funcionalitats, es podria integrar l'ús de plataformes com Booking o Ryanair per tal que els viatges contractats amb aquestes empreses ja surti a l'aplicació. També s'hauria de crear una versió de pagament de l'app per tal de tenir un rendiment econòmic i que sigui rendible, per exemple limitant certes funcionalitats a un pagament únic o anual. A més, també es podria crear un fòrum o xat entre usuaris de l'aplicació per tal de fomentar l'ús i recomanar llocs o experiències.

En cas de tenir una gran escala, caldria incrementar el nombre de dispositius provats, ja que només s'han fet proves amb un dispositiu mòbil, augmentar l'escalabilitat de les APIs i els serveis on està emmagatzemada l'aplicació, en estar actualment a una versió gratuïta d'AWS i de Google Cloud i Firebase.

Per tant, hi ha molt possible treball futur i millores a realitzar en cas de voler explotar aquesta aplicació que, de moment, estarà apartada en no tenir temps suficient per dedicar-li estant alhora treballant per a una empresa.

14. Referències

[1] Mejorar el bienestar emocional y reducir los gastos de viaje: 5 tendencias que han marcado esta temporada turística - Paypal. <https://newsroom.es.paypal-corp.com/2023-09-12-Mejorar-el-bienestar-emocional-y-reducir-los-gastos-de-viaje-5-tendencias-que-han-marcado-esta-temporada-turistica>. Consultat el 24 de febrer de 2024.

[2] 5 claves para gestionar los gastos de viaje – Captio. <https://www.captio.net/blog/como-gestionar-los-gastos-de-viaje-imprevistos>. Consultat el 29 de febrer de 2024.

[3] ¿Cómo Calcular Un Presupuesto Para Tu Próximo Viaje? – hsm Madrid. <https://revistahsm.com/como-calcular-un-presupuesto-para-tu-proximo-viaje/>. Consultat el 24 de febrer de 2024.

[4] Consejos de un trotamundos para solucionar los imprevistos que surgen al viajar – EIPais. <https://elpais.com/elviajero/2023-05-20/consejos-de-un-trotamundos-para-solucionar-los-imprevistos-que-surgen-al-viajar.html#>. Consultat del 29 de febrer de 2024.

[5] 5 aplicaciones perfectas para controlar los gastos de los Viajes – laSexta. https://www.lasexta.com/viajestic/consejos-viajeros/5-aplicaciones-perfectas-controlar-gastos-viajes_20231205656f11ac62c50d0001b94d76.html. Consultat el 24 de febrer de 2024.

[6] Como hacer un presupuesto de viaje – Fintonic. <https://www.fintonic.com/blog/como-hacer-un-presupuesto-para-un-viaje/>. Consultat el 29 de febrer de 2024.

[7] Scrum: qué es y cómo funciona este marco de Trabajo – we are marketing. <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#:~:text=Scrum%20es%20un%20framework%20que,valor%20posible%20productiva%20y%20creativamente>. Consultat el 25 de febrer de 2024.

[8] Taiga: la herramienta de gestión de proyectos gratuita y open source – Taiga. <https://taiga.io/es>. Consultat el 25 de febrer de 2024.

- [9] Qué es GitHub y cómo empezar a usarlo – Hostinger. <https://www.hostinger.es/tutoriales/que-es-github>. Consultat el 25 de febrer de 2024.
- [10] Gitflow Workflow, Automated Builds, Integration & Deployment – Medium. <https://medium.com/devsondevs/gitflow-workflow-continuous-integration-continuous-delivery-7f4643abb64f>. Consultat el 25 de febrer de 2024.
- [11] GitHub Actions – GitHub. <https://github.com/features/actions>. Consultat el 25 de febrer de 2024.
- [12] Visual Studio Code. <https://code.visualstudio.com/>. Consultat el 2 de març de 2024.
- [13] Android Studio – Android Developers. <https://code.visualstudio.com/>. Consultat el 2 de març de 2024.
- [14] Docker. <https://www.docker.com/>. Consultat el 2 de març de 2024.
- [15] Amazon Relation Database Service (RDS) – AWS. <https://aws.amazon.com/es/rds/>. Consultat el 2 de març de 2024.
- [16] Amazon EC2 – AWS. <https://aws.amazon.com/es/ec2/>. Consultat el 2 de març de 2024.
- [17] GitHub. <https://github.com/>. Consultat el 2 de març de 2024.
- [18] Microsoft Office 365 – Microsoft. <https://www.microsoft.com/es-es/microsoft-365/>. Consultat el 2 de març de 2024.
- [19] Google Workspace – Google. <https://workspace.google.com/intl/es/>. Consultat el 2 de març de 2024.
- [20] Draw io. Diagrams. <https://app.diagrams.net/>. Consultat el 2 de març de 2024.
- [21] The free and open-source project management tool – Taiga. <https://taiga.io/>. Consultat el 2 de març de 2024.
- [22] Atenea – Universitat Politècnica de Catalunya. <https://atenea.upc.edu/>. Consultat el 2 de març de 2023.

[23] Glassdoor. <https://www.glassdoor.es/Sueldos/index.htm>. Consultat el 4 de març de 2024.

[24] Alquiler de oficinas en Barcelona – MisOficinas. <https://www.misoficinas.es/alquiler/oficinas/barcelona>. Consultat el 5 de març de 2024.

[25] Días laborables. <https://dias-laborables.es/cuantos-dias-laborables-en-ano-2024-Catalu%C3%B1a.htm>. Consultat el 5 de març de 2024.

[26] MPAndroidChart. GitHub. <https://github.com/PhilJay/MPAndroidChart/tree/master>. Consultat el 13 de maig de 2024.

[27] Google Play Store. Google. https://play.google.com/store/games?hl=es_419&gl=US. Consultat el 3 de maig de 2024.

[28] MySQL. MySQL. <https://www.mysql.com/>. Consultat el 3 de maig de 2024.

[29] Google Cloud. Google. <https://cloud.google.com/>. Consultat el 3 de maig de 2024.

[30] Google Firebase. <https://firebase.google.com/>. Consultat el 3 de maig de 2024.

[31] Google Maps Platform. Google. <https://mapsplatform.google.com/>. Consultat el 3 de maig de 2024.

[32] Protocolo de transferencia de hipertexto. Wikipedia. https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto. Consultat el 3 de maig de 2024.

[33] Protocolos TCP/IP. IBM. <https://www.ibm.com/docs/es/aix/7.2?topic=protocol-tcpip-protocols>. Consultat el 3 de maig de 2024.

[34] The web framework for perfectionists with deadlines. Django. <https://www.djangoproject.com/>. Consultat el 3 de maig de 2024.

- [35] Multiple Activity Pattern. WebSkittersAcademy. <https://www.webskittersacademy.in/why-single-activity-architecture-android-app-development/>. Consultat el 3 de maig de 2024.
- [36] Understanding the MVC pattern in Django. Medium. <https://medium.com/shecodeafrica/understanding-the-mvc-pattern-in-django-edda05b9f43f>. Consultat el 15 de maig de 2024.
- [37] Understanding Django MVT architecture and view functions| Django Full Course for Beginners | Lesson 2. Medium. <https://medium.com/@CodeMaple/understanding-django-mvt-architecture-and-view-functions-django-full-course-for-beginners-lesson-39c8da093b44>. Consultat el 15 de maig de 2024.
- [38] Decorator in Python. Refactoring Guru. <https://refactoring.guru/design-patterns/decorator/python/example>. Consultat el 3 de maig de 2024.
- [39] Middleware. Django. <https://docs.djangoproject.com/en/5.0/topics/http/middleware/>. Consultat el 3 de maig de 2024.
- [40] Exploring Django ORM. LinkedIn. <https://www.linkedin.com/pulse/exploring-django-orm-anas-al-hakim>. Consultat el 4 de maig de 2024.
- [41] Migrations. Django. <https://docs.djangoproject.com/en/5.0/topics/migrations/>. Consultat el 4 de maig de 2024.
- [42] Database access optimization. Django. <https://docs.djangoproject.com/en/5.0/topics/db/optimization/>. Consultat el 4 de maig de 2024.
- [43] Kotlin. Kotlin. <https://kotlinlang.org/>. Consultat el 4 de maig de 2024.
- [44] Python. Python España. <https://es.python.org/>. Consultat el 4 de maig de 2024.
- [45] Currency Rates. Exchange Rates. <https://exchangeratesapi.io/>. Consultat el 4 de maig de 2024.

[46] Toasts overview. Android Developers.
<https://developer.android.com/guide/topics/ui/notifiers/toasts>. Consultat el 5 de maig de 2024.

[47] Intent. Android Developers.
<https://developer.android.com/reference/android/content/Intent>. Consultat el 5 de maig de 2024.

[48] Build variants. Android Developers.
<https://developer.android.com/build/build-variants>. Consultat el 5 de maig de 2024.

[49] Applications. Django.
<https://docs.djangoproject.com/en/5.0/ref/applications/>. Consultat el 5 de maig de 2024.

[50] REST URL. Arquitectura java. <https://www.arquitecturajava.com/rest-url-formatos-y-buenas-practicas/>. Consultat el 5 de maig de 2024.

[51] Writing and running tests. Django.
<https://docs.djangoproject.com/en/5.0/topics/testing/overview/>. Consultat el 5 de maig de 2024.

[52] Logcat. Android Developers.
<https://developer.android.com/studio/debug/logcat>. Consultat el 5 de maig de 2024.

[53] Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales. BOE.
<https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>. Consultat el 5 de maig de 2024.