

No value specified for labeldate
No value specified for labeldate

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE GRAU

WildTraveling

Autor:
Pere Bergé Sánchez

Tutora:
Clàudia Ayala Martínez

Grau en Enginyeria Informàtica
Facultat d'Informàtica de Barcelona
Enginyeria del Software i Sistemes d'Informació

12 de juny de 2017

Índex

1	Introducció i estat de l'art	1
1.1	Formulació del problema	1
1.2	Parts interessades	2
1.3	Estudi de mercat	2
1.4	Conclusions	10
2	Abast del projecte	11
2.1	Obstacles i riscos	12
2.2	Metodologia	13
2.3	Eines de seguiment	14
2.4	Adequació a Enginyeria del Software	14
2.4.1	Competències tècniques del projecte	15
2.4.2	Coneixements Aplicats	16
3	Planificació temporal	17
3.1	Descripció de les tasques	17
3.2	Diagrama de Gantt	18
3.3	Recursos humans	19
3.4	Recursos tècnics	20
3.5	Valoració d'alternatives i pla d'acció	20
4	Gestió Econòmica	23
4.1	Identificació i estimació dels costos	23
4.2	Control de gestió	25
5	Anàlisi Inicial	27
5.1	Estudi de viabilitat	27
5.1.1	Anàlisi de les parts interessades	27
5.1.2	Limitacions legals implicades	27
5.2	Anàlisi de riscos	28
5.2.1	Estratègies de prevenció	28
5.2.2	Plans per afrontar obstacles	29
5.3	Visió global de les funcionalitats del sistema	29
5.4	Anàlisi de les tecnologies candidates	30
5.4.1	Plataforma	30
5.4.2	Serveis externs	31
5.4.3	Base de dades	32
5.5	Conclusions de l'anàlisi inicial	33
6	Requisits	35
6.1	Propietats i hipòtesis sobre el domini	35
6.2	Restriccions	35
6.3	Diagrama de casos d'ús	36

6.4	Casos d'ús	36
6.5	Requisits no funcionals	45
6.5.1	Requisits de disseny	45
6.5.2	Requisits de percepció	45
6.5.3	Requisit d'usabilitat	46
6.5.4	Requisits de rendiment	47
6.5.5	Requisits de funcionament i ambientals	48
6.5.6	Requisits de manteniment i suport	49
6.5.7	Requisits de seguretat	49
6.5.8	Requisits legals	50
7	Especificació	51
7.1	Esquema conceptual	51
7.1.1	Restriccions textuais	51
7.1.2	Descripció de les entitats	52
7.2	Esquema del comportament	54
8	Disseny	65
8.1	Arquitectura	65
8.1.1	Patrons de disseny	68
8.1.2	Tecnologies utilitzades	69
8.2	Capa de presentació	70
8.3	Diagrama de classes de disseny	78
8.4	Model de dades	78
8.5	Diagrames de seqüència	79
9	Implementació	81
10	Proves	87
11	Informe de sostenibilitat	89
11.1	Dimensió econòmica	89
11.2	Dimensió social	89
11.3	Dimensió ambiental	90
12	Conclusions	91
12.1	Justificació del desenvolupament de les competències tècniques	91
12.2	Desviacions en la planificació i el pressupost	92
12.3	Gantt final	94
12.4	Limitacions i dificultats	94
12.5	Treball futur	94

Índex de taules

3.1	Recursos tècnics	20
4.1	Estimació cost recursos humans	23
4.2	Estimació cost recursos software	23
4.3	Estimació cost recursos hardware	24
4.4	Estimació cost despeses generals	24
4.5	Estimació costos directes	24
4.6	Estimació cost d'imprevistos	24
4.7	Estimació cost total del projecte	25
4.8	Control d'hores reals	25
4.9	Comparació cost estimat i cost real	25
6.1	Cas d'ús <i>Registrar-se al sistema</i>	36
6.2	Cas d'ús <i>Iniciar sessió al sistema</i>	37
6.3	Cas d'ús <i>Tancar sessió al sistema</i>	37
6.4	Cas d'ús <i>Crear viatge</i>	38
6.5	Cas d'ús <i>Consultar llista de viatges</i>	38
6.6	Cas d'ús <i>Consultar despeses</i>	39
6.7	Cas d'ús <i>Afegir nova despesa</i>	39
6.8	Cas d'ús <i>Eliminar despesa</i>	40
6.9	Cas d'ús <i>Consultar deutes de despesa</i>	40
6.10	Cas d'ús <i>Cobrar deute</i>	40
6.11	Cas d'ús <i>Modificar despesa</i>	41
6.12	Cas d'ús <i>Gestionar cerques</i>	41
6.13	Cas d'ús <i>Realitzar una cerca</i>	41
6.14	Cas d'ús <i>Consultar ruta</i>	42
6.15	Cas d'ús <i>Consultar previsió metereològica</i>	42
6.16	Cas d'ús <i>Gestionar emergències</i>	42
6.17	Cas d'ús <i>Trucar telèfon d'emergència</i>	43
6.18	Cas d'ús <i>Trucar telèfon de contacte</i>	43
6.19	Cas d'ús <i>Consultar llista de notes</i>	43
6.20	Cas d'ús <i>Crear nota</i>	44
6.21	Cas d'ús <i>Consultar nota</i>	44
6.22	Cas d'ús <i>Eliminar nota</i>	44
6.23	Requisit d'escalabilitat	45
6.24	Requisit d'implementació	45
6.25	Requisit d'aparença	45
6.26	Requisit d'estil	46
6.27	Requisit d'ús	46
6.28	Requisit d'aprenentatge	46
6.29	Requisit de comprensió del llenguatge	47
6.30	Requisit de comprensió dels símbols	47
6.31	Requisit de latència i velocitat	47

6.32	Requisit de disponibilitat i fiabilitat	47
6.33	Requisit de funcionament	48
6.34	Requisit de manteniment	48
6.35	Requisit de llençament	48
6.36	Requisit ambiental	48
6.37	Requisit de manteniment	49
6.38	Requisit de suport	49
6.39	Requisit d'integritat	49
6.40	Requisit de privacitat	49
6.41	Requisit de compliment	50
7.1	Atributs de la classe Persona	52
7.2	Atributs de la classe Usuari	52
7.3	Atributs de la classe NoUsuari	52
7.4	Atributs de la classe Viatge	52
7.5	Atributs de la classe Despesa	53
7.6	Atributs de la classe Deute	53
7.7	Atributs de la classe Nota	53
7.8	Atributs de la classe Cerca	53
7.9	Atributs de la classe Punt d'interés	54
11.1	Matriu de sostenibilitat	89
12.1	Distribució d'hores reals i percentatge de treball	93
12.2	Comparativa entre les hores estimades i les reals treballades, i el seu cost associat	93

Índex de figures

1.1	Settle up	3
1.2	Settle up	4
1.3	Splitwise	4
1.4	Meteoplaza	5
1.5	Meteoplaza	5
1.6	Emergency assistant	6
1.7	First Call	7
1.8	Arround me	7
1.9	Google Maps	8
1.10	Notes	9
1.11	Color note	9
2.1	Metodologia	13
2.2	Resum de tasques a Toggl.	14
3.1	Planificació del projecte.	18
3.2	Diagrama de Gantt.	19
6.1	Diagrama de casos d'ús	36
7.1	Esquema conceptual.	51
8.1	Arquitectura de l'aplicació.	65
8.2	Estructura dels serveis.	66
8.3	Accés a les dades.	67
8.4	Accés a serveis externs.	67
8.5	Patró factoria.	68
8.6	Patró adaptador.	68
8.7	Patró repositori.	69
8.8	Patró servei.	69
8.9	Inici de sessió	70
8.10	Inici de sessió	71
8.11	Crear viatge	71
8.12	Consultar viatge	72
8.13	Llista despeses	72
8.14	Crear despesa	73
8.15	Consultar despesa	73
8.16	Modificar despesa	74
8.17	Cerca de llocs d'interès	74
8.18	Resultats de cerca	75
8.19	Consultar ruta	75
8.20	Previsió metereològica	76
8.21	Gestió d'emergències	76
8.22	Llista de notes	77

8.23	Crear nota	77
8.24	Consultar nota	78
8.25	Estructura base de dades.	78
8.26	Solució per no duplicar informació.	79
9.1	Crida d'inici de sessió	81
9.2	Retorn d'inici de sessió	81
9.3	Constructor del repositori	82
9.4	Funció d' <i>insert</i> genèrica	82
9.5	Conversió de les dades a l'entitat Nota	82
9.6	Mètode de la classe ExpenseService	83
9.7	Mètode de la classe ExpenseService	83
9.8	Mètode per obtenir l'hospital més proper	84
9.9	Crida al servei de punts d'interés	84
9.10	Enviament d'un correu electrònic	85
9.11	Funció enviar correu electrònic	85

Capítol 1

Introducció i estat de l'art

Aquest projecte és un Treball Final de Grau de l'especialitat d'Enginyeria del Software de la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya). Es tracta d'un projecte de modalitat A que pretén crear una aplicació de suport al viatger.

1.1 Formulació del problema

Viatjar ajuda a veure la vida des de perspectives diverses, aquest és un dels motius pels quals cada dia tothom està més disposat a emprendre viatges arreu del món. Existeixen molts perfils diferents de viatgers, a alguns els agrada descobrir natura, a d'altres les grans ciutats, hi ha qui es decanta per la tranquil·litat i qui prefereix l'aventura. Sigui quin sigui l'enfoc del viatge, el propòsit és aprendre alguna cosa que quedi per al record.

En els darrers anys, com a conseqüència d'internet i l'abaratiment de les companyies low cost, és molt més fàcil desplaçar-se i estar informat sobre els diferents racons del món. És important també tenir en compte que la gran majoria de gent disposa d'un smartphone, aquest fet permet al viatger accedir a internet durant el viatge i, per tant, estar connectat constantment, cosa que simplifica molt l'accés a informació al moment.

Les persones ens adaptem molt ràpidament a les coses que ens són més fàcils de fer, per aquest motiu actualment no es planifica tant i sovint es necessita una eina que permeti solucionar un problema a l'instant. És en aquest context on apareix l'aplicació que pretén desenvolupar aquest projecte.

Durant un viatge sorgeixen moltes necessitats com poden ser: buscar un restaurant, informar-se del temps, buscar transport, portar un control de les despeses, etc. Actualment existeixen varies aplicacions que ataquen parts concretes d'aquest problema però això obliga al viatger a tenir moltes aplicacions instal·lades al seu dispositiu sense comunicació entre elles.

La idea que proposa aquest projecte és construir una aplicació que englobi les funcionalitats que un viatger considera més importants durant el transcurs del seu viatge presentant-li la informació filtrada tenint en compte el seu estil de viatjar. D'aquesta manera podem definir els objectius del projecte.

Aquest projecte té quatre objectius molt clars:

- **Donar suport al viatger durant el transcurs del seu viatge:** el viatger necessita suport de diversos tipus durant el viatge, per aquest motiu, s'han dut a terme entrevistes a viatgers que han servit per orientar i definir les funcionalitats que vol oferir l'aplicació. Aquestes són: gestió econòmica, previsió meteorològica, gestió de cerques de llocs d'interès, gestió d'emergències, cerca de transports, cerca d'imatges i gestió de notes.
- **Ser intermediari en l'ús de diverses aplicacions que ajuden al viatger:** actualment existeixen diverses aplicacions especialitzades en les funcionalitats que aquesta vol implementar. L'objectiu es basa en utilitzar algunes de les aplicacions disponibles en el mercat per tal de crear funcionalitats pròpies i d'aquesta manera obtenir una aplicació completa per al suport al viatger.
- **Personalitzar les dades obtingudes per a cada viatger:** qualsevol persona pot emprendre un viatge, per tant, els viatgers no tenen unes preferències establertes, hi ha molta diversitat d'estils. Per aquest motiu aquesta aplicació planteja la personalització dels viatges per tal de filtrar la informació obtinguda a través de l'accés a d'altres aplicacions i així ser capaç d'oferir a l'usuari el que s'ajusta més al seu estil.
- **Ser fàcilment escalable:** el món de la tecnologia és un món canviant, per aquest motiu aquesta aplicació vol estar preparada per agregar o eliminar funcionalitats de la manera més senzilla possible.

1.2 Parts interessades

Aquest projecte va adreçat a satisfer les necessitats d'un sector molt ampli de població, no estem parlant d'un client concret. Les parts interessades poden tenir objectius molt diversos. En aquest projecte definim les següents parts interessades:

- **Viatgers**
Els viatgers són els principals protagonistes, són els que es beneficien de l'aplicació de forma directa.
- **Ajuntaments**
Les ciutats obtenen un gran benefici del turisme, per aquest motiu els ajuntaments estan interessats en captar el màxim nombre de turistes possibles. Aquesta aplicació ajuda a donar a conèixer racons amb encant que no surten a les guies convencionals.
- **Competència**
Les plataformes que tracten el mateix tema es veuen afectades pel desenvolupament d'aquest projecte de forma indirecta. En tenir accés a aquesta nova plataforma tenen l'oportunitat d'innovar en la seva.

1.3 Estudi de mercat

En aquest punt s'analitzen les diferents opcions que ofereix el mercat per afrontar el problema descrit.

El primer pas ha estat buscar aplicacions similars a la proposada. Després de fer un anàlisi exhaustiu del mercat s'ha conclòs amb la idea de que no existeix una aplicació

que s'ajusti a la idea proposada. Sí que es veritat, però, que existeixen aplicacions de suport al viatger durant un viatge com és el cas de Virtual Travel Assistant.

- **Aplicació completa**

- **Virtual Travel Assistant**

Virtual Travel Assistant és una aplicació gratuïta que ajuda al viatger durant el transcurs del seu viatge. Aquesta ofereix cerca de llocs d'interès, consell en l'elecció de transport i servei de mapes.

Aquesta aplicació, però, té grans diferències amb la proposada. L'aplicació que desenvolupa aquest projecte destaca en la personalització de les cerques i conté més funcionalitats que completen l'aplicació per a que sigui més útil durant el viatge.

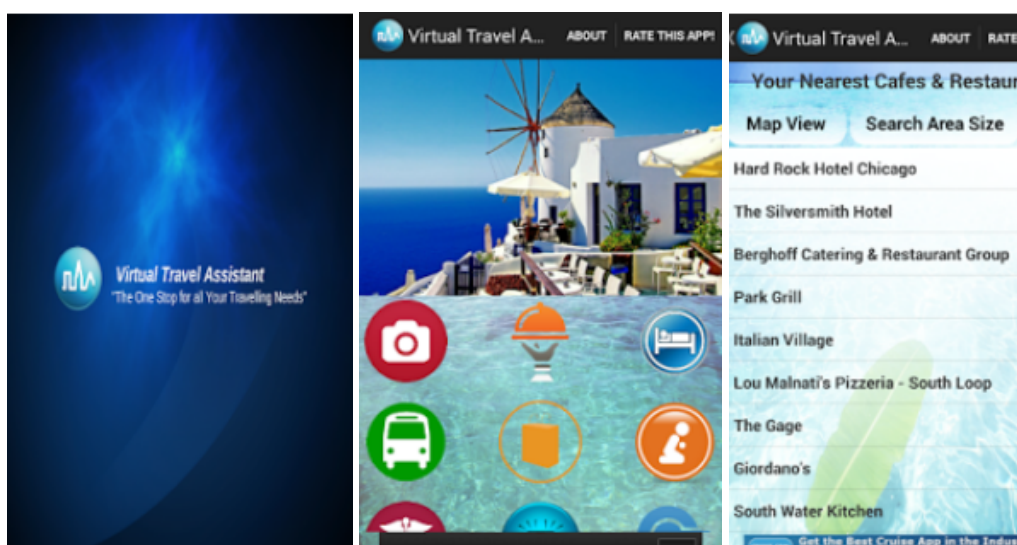


FIGURA 1.1: Settle up

Per tal de fer un millor anàlisi del mercat, s'han analitzat aplicacions especialitzades en les funcionalitats que aquesta vol incorporar i que han sigut anomenades anteriorment.

- **Gestió econòmica**

En aquest punt s'analitza el mercat des del punt de vista de la gestió econòmica del viatge. Existeixen varies aplicacions, totes bastant similars, que tracten aquest tema. Aquesta funcionalitat pretén ser implementada, per tant, no s'utilitzaran aplicacions existents ja que no acaben d'ajustar-se al perfil desitjat. S'han analitzat dos aplicacions que s'han escollit tenint en compte les descàrregues i opinions satisfactòries dels usuaris.

- **Settle Up**

Settle Up és una aplicació mòbil que ajuda a portar el control de despeses. Dins les seves característiques destaquen la gestió d'usuari, la gestió de divises i el fet que quan un usuari és inclòs en una nova despesa aquest rep un avís. A més, ofereix l'opció de filtrar les despeses tenint en compte la categoria d'aquesta.

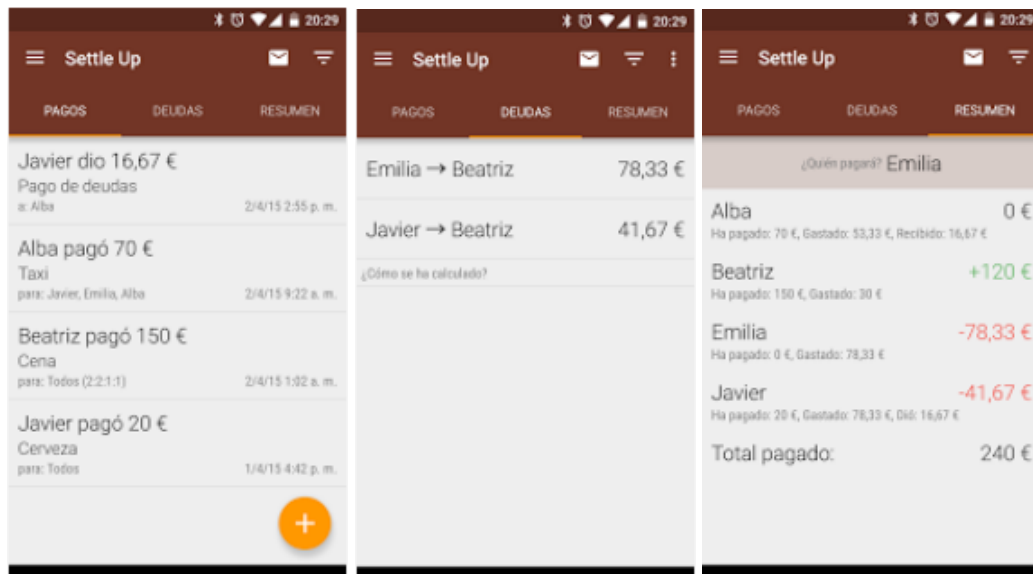


FIGURA 1.2: Settle up

– Splitwise

Splitwise, aplicació mòbil de control de despeses, ofereix gestió d'usuaris, possibilitat de crear una xarxa d'amics i gestió de divises. En aquesta aplicació també s'hi pot afegir una foto i una categoria a la despesa.

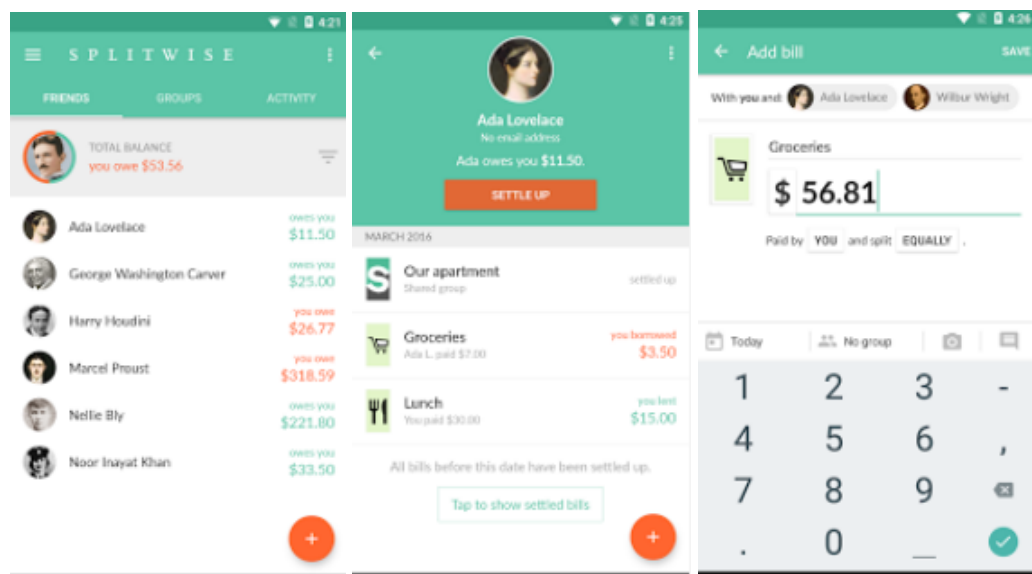


FIGURA 1.3: Splitwise

• Previsió meteorològica

En aquest punt s'analitzen les aplicacions de previsió meteorològica. Existeixen varies aplicacions molt complertes que controlen la meteorologia. Cap d'aquestes, però, està enfocada cap al món viatger, la idea de l'aplicació proposada és mostrar la informació que un viatger necessita saber per decidir què fer durant la seva jornada.

S'han analitzat dues aplicacions:

– Rain Alarm

Rain Alarm destaca pel seu mapa amb representació gràfica de la probabilitat de pluja a l'instant. També localitza automàticament el punt on es troba l'usuari, avisa en cas de pluja imminent i ofereix l'opció d'ajustar la precisió d'aquest avís. Un dels punts negatius de l'aplicació és que conté publicitat.



FIGURA 1.4: Meteoplaza

– Alarma lluvia meteoplaza

Aquesta aplicació és més simple que l'anterior. Destaca per la representació gràfica de la probabilitat de pluja a l'instant, la localització automàtica i l'avís a l'usuari en cas de pluja imminent.

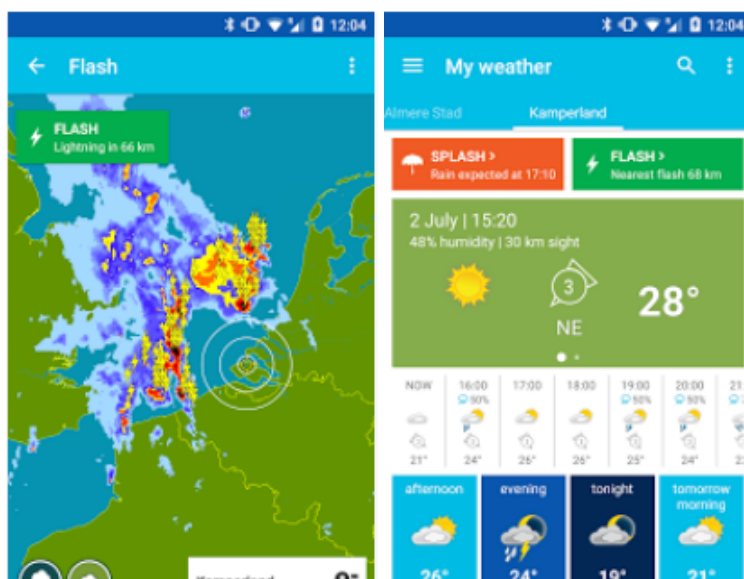


FIGURA 1.5: Meteoplaza

- **Gestió d'emergències**

En aquest punt s'analitzen les aplicacions que tracten el suport en cas d'emergència. L'aplicació que vol crear aquest projecte proposa una funcionalitat que gestioni emergències amb funcionalitats útils per al viatger durant el dia a dia del seu viatge.

S'han analitzat dues aplicacions:

- **Emergency Assistant**

Emergency Assistant guarda un perfil personal de l'usuari i emmagatzema contactes a qui acudir en cas d'emergència. Aquestes funcionalitats són les que més s'apropen al que proposa el projecte. A més, també ofereix control d'al·lèrgies, informació sobre l'assegurança, un llistat de medicacions que l'usuari pren i l'historial mèdic. Es considera, però, que aquestes darreres funcionalitats no són prou importants en el context sobre el que es treballa.



FIGURA 1.6: Emergency assistant

- **First Call**

First Call ofereix funcionalitats molt interessants pel que fa al punt de vista viatger que té el projecte. Aquesta aplicació és molt més senzilla i es centra en classificar el tipus d'emergència del que es tracta i facilitar molt l'avís a terceres persones en cas d'accident, en aquest cas s'envia un avís a d'altres usuaris de l'aplicació. A més també ofereix informació sobre els telèfons de contacte en cas d'emergència de diferents països.

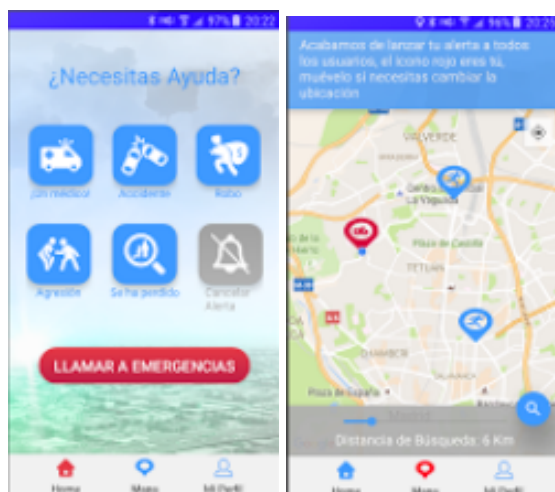


FIGURA 1.7: First Call

- **Gestió de cerques**

En aquest punt s'analitzen les aplicacions que tenen a veure amb cercar llocs d'interès en un lloc concret. S'han analitzat les dues aplicacions més utilitzades al moment.

- **Around me**

Around me ofereix cerques d'establiments que poden interessar a una persona quan està fora de casa, aquests establiments els ofereix de manera ordenada per categories diferenciant cerca d'hotels, restaurants, gasoliners, etc. A més, també ofereix servei de previsió meteorològica.

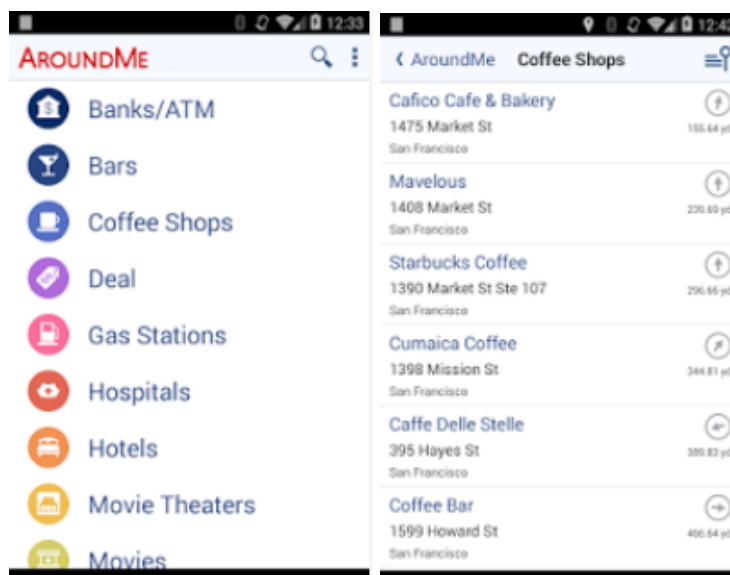


FIGURA 1.8: Around me

– Google Maps

Google Maps és una aplicació molt completa que ofereix cerques d'establiments, rutes entre un punt i un altre i representació en forma de mapa de tot el món. D'altra banda, cal dir que cap de les dues aplicacions ofereix una cerca personalitzada per a l'usuari i tampoc està exclusivament enfocat al món viatger. Aquests dos punts són trets distintius d'aquest projecte.

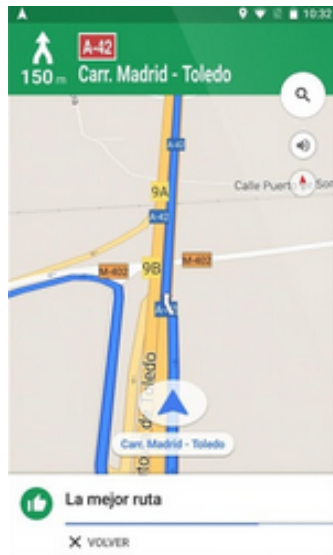


FIGURA 1.9: Google Maps

• Gestió de notes

En aquest punt s'analitzen les aplicacions que treballen en el camp de les notes escrites. Aquesta funcionalitat no té gairebé res a veure amb el món dels viatges, tot i així es decideix implementar-la ja que es troba interessant oferir un lloc on enregistrar fets curiosos, anècdotes o diferents notes.

Existeixen moltes aplicacions que presenten solució a aquest problema, s'han analitzat dues que es considera que es poden apropar a la funcionalitat desitjada.

– Notes

Notes és una aplicació gratuïta que ofereix creació de notes amb l'opció d'organitzar-les en carpetes, llistes de verificació i establir recordatoris a la nota. A més, les notes es guarden automàticament quan són creades o modificades.

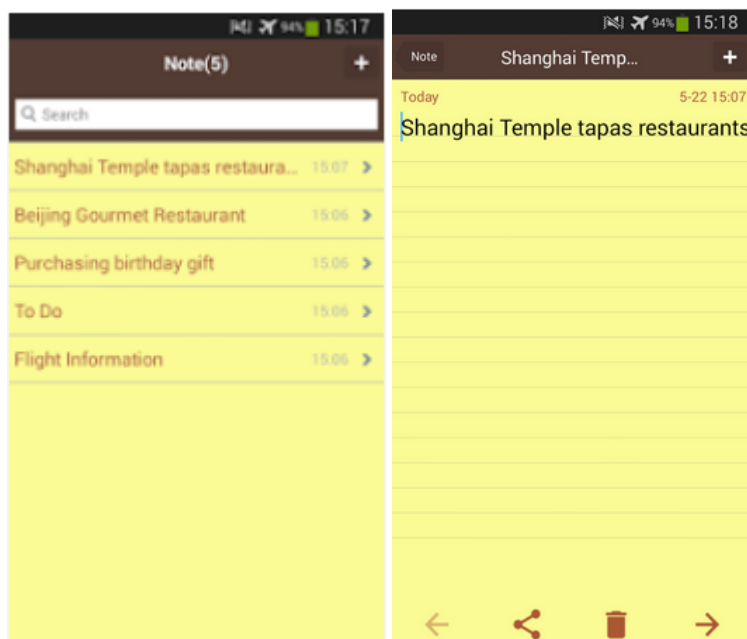


FIGURA 1.10: Notes

– ColorNote

ColorNote és una aplicació simple de llibreta que ofereix organització de notes per color, llistes de verificació i possibilitat de posar contrassenya a les notes. L'aplicació, a més, també té recordatori i la possibilitat de compartir notes a través de SMS i correu electrònic. L'aplicació que proposa aquest projecte pretén oferir un servei bàsic de notes basat en el que pot necessitar el viatger durant el seu viatge. És per aquest motiu que es vol que cada nota estigui localitzada geogràficament i es pugui etiquetar els viatgers.

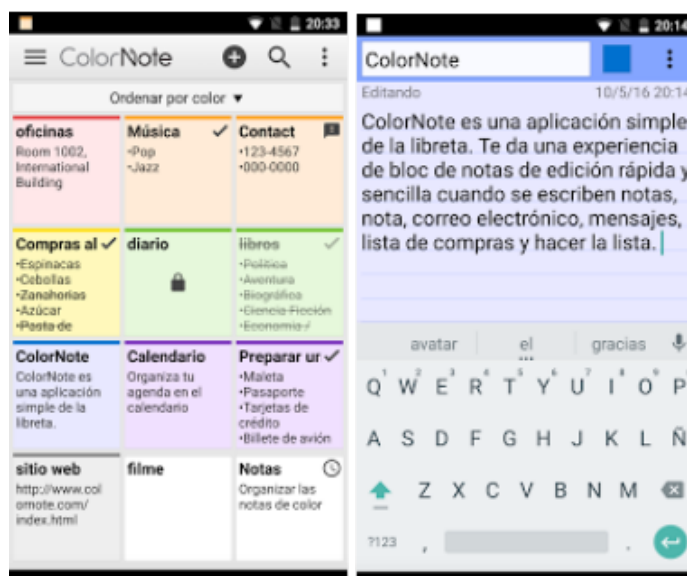


FIGURA 1.11: Color note

1.4 Conclusions

Després d'un anàlisi exhaustiu de la situació actual del mercat pel que fa a aplicacions mòbil que puguin competir amb la idea d'aquest projecte és conclou que realment hi ha un buit en aquest camp.

Existeixen, com s'ha exposat en l'apartat anterior, moltes aplicacions que afronten funcionalitats que poden ajudar al viatger durant el transcurs del seu viatge. No obstant, cap d'aquestes aplicacions engloba funcionalitats de diverses temàtiques ni té el món viatger com a temàtica.

L'estudi del mercat, ha estat molt útil també, a l'hora de refinar les idees que es tenien sobre que oferir a l'usuari en cada funcionalitat.

Capítol 2

Abast del projecte

La idea d'aquest projecte és crear una aplicació referència per qualsevol persona disposada a emprendre un viatge. Una plataforma on trobar suport durant un viatge de qualsevol tipus.

L'aplicació ofereix, aprofitant d'altres sistemes ja al mercat, un conjunt de funcionalitats útils a l'hora de viatjar. Les funcionalitats són les següents:

- **Gestió econòmica:** Controlar qui ha pagat què durant un viatge en grup o bé el control de les despeses del viatge de manera personal.
 - Quan es crea una nova despesa s'envia un avís al deutor.
 - Despesa total del viatge.
- **Gestió de notes:** Guardar notes per tal de recordar anècdotes o fets concrets del viatge.
 - Les notes tenen localització geogràfica.
 - Possibilitat d'etiquetar viatgers a les notes.
- **Gestió de cerques:** Cerca de llocs d'interès de la zona on està situat l'usuari o de qualsevol altre punt.
 - Cerca de restaurants.
 - Cerca de bars.
 - Cerca de llocs d'interès lúdic.
 - Cerca de botigues.
 - Cerca de llocs exteriors d'interés.
 - Consulta de la ruta fins al punt trobat.

Els resultats que s'obtenen de la cerca es mostren a l'usuari ordenats d'acord amb l'estil de viatjar que té l'usuari. Per fer això, abans de començar el viatge, l'aplicació proposa un formulari per tal de descriure el caràcter del viatge. Això dona l'opció al sistema de filtrar les cerques i proposar a l'usuari les opcions més adequades al seu estil.

- **Gestió d'emergències:** Suport en cas d'emergència tenint en compte el país.
 - Ruta fins a l'hospital més proper.
 - Trucada al telèfon d'emergències del país.
 - Trucada al contacte d'emergències.

En aquesta funcionalitat es vol donar la possibilitat a l'usuari de contactar amb una persona de confiança en cas d'emergència. Per aconseguir això, quan s'inicia el viatge, es presenta a l'usuari un formulari on s'han d'introduir el nom, el número de telèfon i el correu electrònic de qui serà la persona de contacte.

- Previsió meteorològica: Estar informat de la meteorologia en tot moment.
 - Predicció setmanal.
 - Informació meteorològica del dia.

2.1 Obstacles i riscos

Per tal d'aconseguir certes funcionalitats de l'aplicació es vol accedir a diferents serveis web per tal d'aconseguir dades concretes. Per una banda, és necessitarà connexió a internet sempre que s'hi vulgui accedir, i per l'altra, el funcionament de l'aplicació depèn de la disponibilitat del servei web.

En el cas de les trucades d'emergència al telèfon d'emergències del país on estigui el viatger, pot ser que es trobi en un país on no hi ha dades registrades sobre els telèfons d'emergència. Un problema molt semblant pot passar a l'hora d'obtenir llocs d'interès, ja que l'usuari es pot trobar en un lloc on no hi ha cap punt ens molts quilòmetres a la rodona.

En ser un Treball Final de Grau té una data límit fixada. Això implica que qualsevol contratemps que faci endarrerir el projecte no es té gaire marge de maniobra per tal d'acabar-lo a temps.

Pel que fa a l'ús del sistema, un cop acabat i posat en marxa, s'ha de pronosticar quin tipus de situacions es podrien donar que comprometessin l'èxit del sistema. A continuació es mostra una llista de riscos que el sistema es pot trobar un cop es posi en funcionament.

- Nivell baix de bateria.
- Consum de dades a l'estranger.
- Caiguda del servei web.
- Caiguda de la base de dades.
- Canvis en la legislació.
- Desconfiguració de l'aplicació.
- Canvis en la resposta del servei web.

Finalment, a l'autor poden sorgir-li obstacles com: l'aparició de noves funcionalitats o canvis en les actuals que obliguen a reestructurar tot el projecte, problemes provocats per tenir poca experiència amb les tecnologies utilitzades, llenguatges de programació o una mala planificació.

2.2 Metodologia

En el desenvolupament del projecte s'utilitza la metodologia Scrum. El temps de desenvolupament està dividit en diferents sprints de dues setmanes, d'aquesta manera el treball és constant. La primera part del projecte es basa en la definició dels requisits i la definició de les històries d'usuari.

Per tal de garantir qualitat en la feina feta, cada setmana en acabar l'sprint, es fa una revisió de les tasques acabades i una valoració del treball fet assegurant que tot estigui correcte. Per acabar es planifica el següent sprint. Un punt molt important que fa que utilitzar aquesta metodologia sigui bo pel projecte és que es tracta de que a cada iteració s'implementin funcionalitats que puguin ser entregades al client. Per tant, des del primer sprint tindrem una aplicació que funciona i hi anirem afegint funcionalitats. D'aquesta manera podem comprovar si el que estem desenvolupant és el que realment es necessita i fer els canvis necessaris si no estem anant per bon camí. A més, també podem detectar problemes en les parts ja implementades i així solucionar-los en les pròximes iteracions.

Per tal de seguir aquesta metodologia s'utilitza l'eina Trello. Trello és un software d'administració de projectes amb interfície web, iOS i Android. En aquest projecte s'utilitza aquesta aplicació per tal d'organitzar els sprints setmanals de manera visual i intuïtiva.

Per tal d'assegurar-nos la persistència del codi i tenir-lo guardat al núvol s'utilitza Git amb la plataforma GitHub. GitHub és un servei de hosting de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi font, i a més, també afegeix les seves característiques pròpies.



FIGURA 2.1: Metodologia

2.3 Eines de seguiment

Per tal de portar un control de la feina feta s'utilitzen diferents eines de seguiment. En la part d'implementació, durant les iteracions, s'utilitza la eina Toggl. Toggl és un software disponible a la xarxa que permet portar el control del temps que dediques a una tasca en concret. Per tant, a cada historia d'usuari que es comença se li assigna una nova tasca i es fica el cronòmetre en marxa. A la figura 2.1 es pot veure un exemple d'utilització de l'eina.

Una altra eina de seguiment és la preparació de les reunions amb la tutora del projecte, normalment, una reunió al final de cada iteració. A l'inici de cada sprint, com a planificació d'aquest, es redacta un informe on hi apareix l'esquema de casos d'ús on es pot diferenciar entre els que han sigut implementats, els que es preten implementar en l'sprint i els que queden pendents; un anàlisi de les tecnologies que s'han d'utilitzar durant la iteració (APIs, serveis externs, etc) i l'especificació dels casos d'ús que es volen implementar en la iteració.

Quan s'acaba una iteració, a mode de resum, també es redacta un document valorant la feina que s'ha fet i remarcant qualsevol aspecte important per la implementació del projecte.

Afegir gestió de despeses	16:43:23
Consultar metereologia	8:54:11
Consultar Viatge	1:20:14
Crear cerca	22:14:22
CrearViatge	8:02:19
Create Trip	2:08:00
Gestionar emergències	8:33:47
Inici de sessió	2:49:16
INTERFACE	6:10:06
Llistar viatges	1:25:30
Notes	8:06:15

FIGURA 2.2: Resum de tasques a Toggl.

2.4 Adequació a Enginyeria del Software

Aquest projecte és un treball de final de grau de modalitat A, l'alumne proposa una idea i la desenvolupa. En aquest cas la idea que pretén desenvolupar aquest projecte és una aplicació mòbil de suport al viatger. En ser una aplicació mòbil el projecte necessita un bon disseny a nivell de software, cosa que requereix comptar amb un dissenyador amb coneixements prou avançats d'enginyeria del software. Aconseguir un bon disseny permetrà obtenir una aplicació sòlida, neta i molt fàcil de mantenir.

Un altre punt important d'aquest projecte és que es caracteritza per utilitzar aplicacions ja existents en el mercat. Això significa que és necessari dissenyar l'aplicació de tal manera que permeti utilitzar serveis que estan disponibles al núvol.

A més, en el món dels serveis web hi ha avanços molt sovint i surten noves aplicacions cada cert període de temps. És per això que aquesta aplicació vol estar preparada per tal d'assumir canvis pagant el preu més baix possible. Per tant, aquesta aplicació pretén ser dissenyada de manera que sigui molt senzill adjuntar noves funcionalitats o eliminar alguna de les existents, és a dir, pretén potenciar l'escalabilitat.

2.4.1 Competències tècniques del projecte

- CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [En profunditat]
És l'essència del projecte ja que es tracta de desenvolupar i mantenir un servei software d'una certa complexitat. Aquesta competència s'assoleix amb el desenvolupament complet del projecte.
- CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. [Una mica]
L'aplicació necessita la participació d'aplicacions externes per tal d'aconseguir certes funcionalitats, per tant, és necessari la integració de les dades que arriben des dels serveis exteriors.
- CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. [Bastant]
Aquest objectiu s'assoleix durant l'estudi de viabilitat del projecte ja que s'ha de preveure quins seran els possibles riscos a l'hora del desenvolupament.
- CES1.5: Especificar, dissenyar, implementar i avaluar bases de dades. [Una mica]
Per dur a terme aquest projecte serà necessari el desenvolupament d'una base de dades, per tant, amb el disseny i desenvolupament d'aquesta s'assolirà aquesta competència.
- CES1.6: Administrar bases de dades (CIS4.3) [Una mica]
Un cop l'aplicació estigui posada en marxa s'haurà d'administrar la base de dades, per aquest motiu la competència no incideix en profunditat. Aquesta competència s'assolirà un cop acabat el projecte.
- CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. [En profunditat]
En utilitzar una metodologia Agile es dona molta importància al control de qualitat i a les proves durant el desenvolupament, aquesta competència es posarà a prova en profunditat durant el desenvolupament del producte.
- CES1.9: Demostrar comprensió en la gestió i govern dels sistemes software. [En profunditat]
Desenvolupar una aplicació com aquesta requereix d'una bona gestió del sistema software, per aquest motiu es treballa aquesta competència en profunditat i s'assoleix havent acabat el projecte amb èxit.

- **CES2.1:** Definir i gestionar els requisits d'un sistema software. [En profunditat]
Durant l'especificació de requisits es quan es treballa i s'assoleix aquesta competència, és una part bàsica del projecte i per aquest motiu es treballa en profunditat.
- **CES2.2:** Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integrin aspectes ètics, socials, legals i econòmics. [En profunditat]
En el desenvolupament del projecte és necessari dissenyar solucions apropiades utilitzant mètodes d'enginyeria del software, s'assolirà aquesta competència en profunditat havent acabat el projecte amb èxit.

2.4.2 Coneixements Aplicats

Durant la meua estada a la FIB he cursat varies assignatures amb continguts molt interessants a l'hora de desenvolupar aquest projecte. En especial, les que més s'apliquen en aquest projecte a nivell tècnic són:

- **AS:** Tots els continguts tècnics d'arquitectura de software.
- **GPS:** En aquesta assignatura es va treballar la gestió de projectes, en aquest cas, la part d'Agile és la més important.
- **PES:** Durant el desenvolupament del projecte d'especialitat es va fabricar una aplicació Android com la que es desenvolupa en aquest projecte. A més es van utilitzar mètodes de seguiment i metodologia que s'aplica a aquest projecte.
- **ASW:** Aquesta aplicació necessita utilitzar serveis web, cosa que es va exposar en l'assignatura ASW.
- **ER:** Durant l'especificació de requisits s'ha utilitzant en tot moment els coneixements adquirits cursant aquesta assignatura.
- **BD:** Per tal de dissenyar la base de dades s'ha posat en pràctica part dels coneixements adquirits a l'assignatura.
- **CBDE:** Per tal de dissenyar la base de dades s'ha posat en pràctica els coneixements i l'esperit crític adquirit a l'assignatura sobre les bases de dades no relacionals.

Capítol 3

Planificació temporal

El projecte està delimitat temporalment entre el mes de febrer del 2017 i el mes de juny del 2017. L'assignatura GEP dona el tret de sortida al TFG ja que els lliuraments d'aquesta assignatura són parts de la memòria final del projecte. El projecte s'inicia el dia 13 de febrer i s'acaba a finals de juny amb la presentació oral. No obstant es marca una data límit prèvia a la presentació per tal d'assegurar que el projecte tingui una certa maduresa el dia de la defensa, aquest dia és el 15 de juny.

3.1 Descripció de les tasques

S'ha dividit la feina en tres grans fases. Per cadascuna d'elles s'explica quines tasques es duen a terme i les hores de dedicació previstes.

- **Fase inicial**

En la fase inicial del projecte és bàsic determinar que es vol construir. Aquesta part del desenvolupament del projecte està marcada per la gestió del mateix. Es defineix clarament quin és l'abast, el problema que es vol solucionar, l'estat actual de l'art, la metodologia a seguir, la planificació i la gestió econòmica.

En aquesta fase inicial també es dur a terme l'anàlisi de requisits del sistema, aquest punt es força important ja que és bàsic definir què ha de satisfer l'aplicació per tal de considerar-la completa.

El projecte segueix la metodologia àgil Scrum, per tant, la definició del backlog determinarà les històries d'usuari que s'han d'implementar per tal de dur a terme el projecte. Una història d'usuari és una característica o funcionalitat de la plataforma independent a la resta, i per tant, el backlog és el conjunt ordenat d'aquestes característiques o funcionalitats. Quan es defineix una història d'usuari també s'hi afegeixen els criteris d'acceptació, que són unes condicions que ha de satisfer el sistema per tal de donar la història per acabada. El criteri que es fa servir per ordenar el backlog és la importància de la pròpia història i a l'hora de la selecció de les històries a desenvolupar durant una iteració concreta es mira l'ordre però també el tamany de la història. El tamany de la història ve definit pels punts d'història.

Durant aquesta fase es preveu un treball diari de 3 hores.

- **Iteracions**

Una iteració es un període de temps durant el qual s'ha de desenvolupar un conjunt d'històries d'usuari prèviament seleccionades. És important que es

desenvolupin de manera vertical, és a dir, en acabar una iteració s'ha de tenir software que funcioni i, a més, integrat al que s'havia desenvolupat en la iteració anterior.

En l'inici de cada iteració es seleccionen les històries d'usuari que es volen desenvolupar i es crea un tauler utilitzant l'eina Trello. A mesura que es van completant aquestes històries es fan proves per veure que tot funcioni i es donen per acabades. El dia que acaba la iteració es valora la feina feta i es redefineix la velocitat (punts d'historia per iteració). Durant el desenvolupament del projecte es duran a terme 5 iteracions distribuïdes en les següents dates:

- Iteració 1: 29 de març - 12 d'abril
- Iteració 2: 13 d'abril - 26 d'abril
- Iteració 3: 27 d'abril - 10 de maig
- Iteració 4: 11 de maig - 24 de maig
- Iteració 5: 25 de maig - 7 de juny

Durant aquesta fase del projecte es preveu un treball de 4 hores al dia.

- **Fase final**

En la fase final del projecte es dedicaran tots els esforços en la redacció de la memòria del projecte i la preparació de la presentació final. A més, es duran a terme proves de l'aplicació amb la participació d'usuaris reals. Aquesta fase es comença el dia 8 de juny i finalitza el dia 16 de juny amb la previsió d'un treball diari de 3 hores.

3.2 Diagrama de Gantt

Per tal de definir bé la planificació del projecte s'utilitza un diagrama de Gantt (figura 3.2) i una taula explicativa (figura 3.1).

Nom de la tasca	Data d'inici	Data final	Duració	Dependències
Fase inicial	13/02/17	28/03/17	32d	
Estudi de viabilitat	13/02/17	20/02/17	6d	
Definició de l'abast i contextualització	21/02/17	28/02/17	6d	2
Planificació	01/03/17	06/03/17	4d	3
Gestió econòmica	07/03/17	13/03/17	5d	4
Presentació preliminar	14/03/17	20/03/17	5d	5
Anàlisi de requisits	14/03/17	21/03/17	6d	
Especificació	22/03/17	28/03/17	5d	7
Implementació	29/03/17	07/06/17	51d	8
Sprint 1	29/03/17	12/04/17	11d	
Sprint 2	13/04/17	26/04/17	10d	10
Sprint 3	27/04/17	10/05/17	10d	11
Sprint 4	11/05/17	24/05/17	10d	12
Sprint 5	25/05/17	07/06/17	10d	13
Fase final	08/06/17	26/06/17	13d	14
Redacció de la memòria	08/06/17	19/06/17	8d	
Presentació final	08/06/17	26/06/17	13d	

FIGURA 3.1: Planificació del projecte.

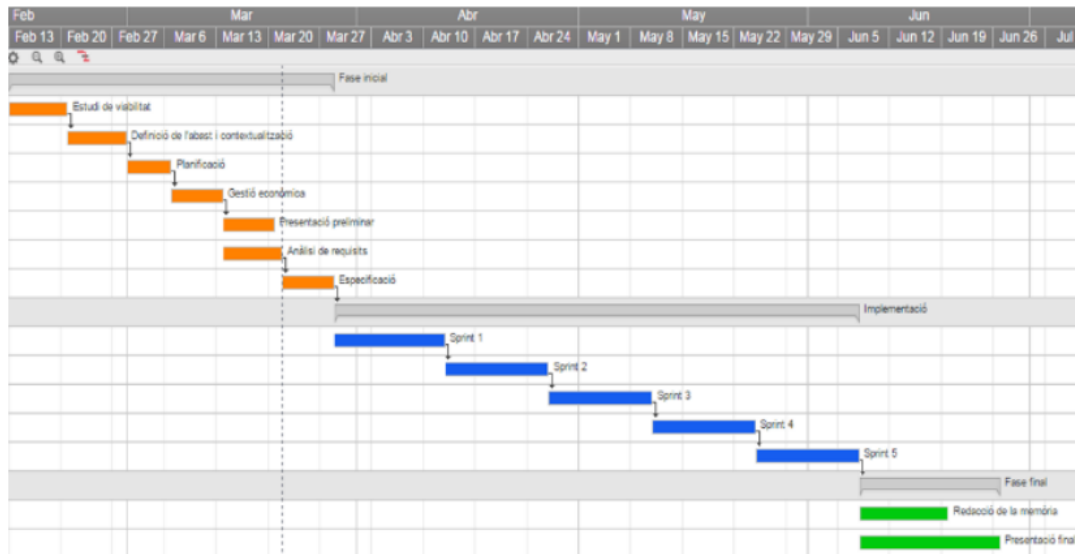


FIGURA 3.2: Diagrama de Gantt.

3.3 Recursos humans

- **Cap de projecte:** Encarregat de la planificació i supervisió de les tasques a desenvolupar en un projecte així com de l'organització, coordinació i mediació de l'equip de recursos humans.
- **Analista:** Estudia els problemes actuals per proposar una solució òptima i d'acord als recursos disponibles. Proporcionen les funcionalitats que tindrà el sistema mitjançant el model de casos d'ús.
- **Dissenyador:** En base a la documentació proporcionada pels analistes, s'estructura i es defineix una arquitectura pel sistema que s'ha de desenvolupar.
- **Programador:** S'encarrega d'implementar el disseny proposat pels dissenyadors d'una forma eficient.
- **Tester:** Prova el sistema per tal de detectar errors o recomanar modificacions, ha de redactar un informe després de cada prova efectuada.

3.4 Recursos tècnics

Recurs	Tipus	Finalitat
Recurs Ordinador portàtil Packard Bell i7, 4 GB de Ram i windows 7 64x	Eina de desenvolupament	Desenvolupar l'aplicació i la memòria
Motorola Moto G amb Android 5.0	Eina de desenvolupament	Dur a terme les proves durant el desenvolupament de l'aplicació
Android Studio v2.2.1	Eina de desenvolupament	Desenvolupar l'aplicació mòbil.
Microsoft Office word 2007 Microsoft Office Power Point 2007	Eina de desenvolupament	Redacció d'informes
Point 2007 Adobe Photoshop 9	Eina de desenvolupament	Edició i maquetació de la presentació final.
Adobe Reader XI	Eina de desenvolupament	Visualització de documents
Correu electrònic Gmail	Eina de comunicació	Comunicació amb la tutora del projecte
Git	Eina de desenvolupament	Control de versions del codi font de l'aplicació
LateX	Eina de desenvolupament	Redacció de la memòria final

TAULA 3.1: Recursos tècnics

3.5 Valoració d'alternatives i pla d'acció

Durant el desenvolupament del projecte poden sorgir desviacions de dos tipus:

- **Errors en la planificació**

Resulta força complicat ser precís en la planificació del projecte ja que és un equip completament nou en aquesta disciplina.

L'error pot venir donat per una planificació massa pessimista, en aquest cas no hi hauria cap problema ja que durant les iteracions es podrien agafar més històries d'usuari seguint l'ordre del backlog. Acabar el projecte abans d'hora no seria perjudicial.

D'altra banda, l'error pot ser causat per una planificació massa optimista. En aquest cas si que és un problema ja que significa que l'equip no pot acabar la feina a temps. Per afrontar aquesta situació s'ha reservat un període d'una setmana abans de l'entrega final del projecte que es podria utilitzar per acabar les històries pendents. A més, durant l'última iteració s'estudiaria l'opció de realitzar hores extra, i en el cas que no es pugui dur a terme tot el projecte es deixarien sense implementar les últimes històries del backlog ja que sempre

són les menys prioritàries.

Tot i aquestes possibles desviacions degut a una mala planificació a cada final d'iteració i reunió amb la tutora s'evaluarà la situació i tenint en compte les dates marcades es reafirmarà la planificació establerta o es faran les modificacions necessàries.

- **Imprevistos**

Durant el desenvolupament del projecte poden sorgir imprevistos que sigui quin sigui l'origen d'aquestes es solucionaran fent hores extra i prioritzant les tasques que aportin més valor al projecte.

Capítol 4

Gestió Econòmica

4.1 Identificació i estimació dels costos

En aquest punt es fa una estimació dels elements que formen part del pressupost del projecte tenint en compte els recursos que s'utilitzen.

- **Recursos humans**

Aquest projecte el desenvolupa únicament una persona, per aquest motiu es necessari adoptar diferents rols. A continuació es mostra una taula amb el preu que cobra per hora cada perfil implicat en el projecte, les hores estimades que durà a terme cada rol i el cost total.

Recursos	Rol	€/h	Estimació en hores	Total estimat
Pere Bergé	Cap de projecte	45 €/h	160 h	7.200 €
	Analista	30 €/h	45 h	1.350 €
	Dissenyador	30 €/h	70 h	2.100 €
	Programador	25 €/h	104 h	2.600 €
	Tester	15 €/h	30 h	450 €
			409 h	13.700 €

TAULA 4.1: Estimació cost recursos humans

- **Software**

Per dur a terme aquest projecte són necessàries diverses eines de software, la majoria d'elles gratuïtes.

Producte	Cost	Vida útil	Amortització total(5 mesos)
Windows 7	0€	-	0€
Microsoft Office 2013	119€	4 anys	12,40€
Android Studio 1.5.1	0€	-	0€
Mozilla Firefox	0€	-	0€
Google Drive	0€	-	0€
Trello	0€	-	0€
SmartSheet	0€	-	0€
Adobe Photoshop	12,09€/mes	5 mesos	60,45€
Adobe Reader IX	0€	-	0€
Github	0€	-	0€
Total			72,85€

TAULA 4.2: Estimació cost recursos software

- **Hardware**

Producte	Cost	Vida útil	Amortització total(5 mesos)
Ordinador portàtil	600€/mes	5 anys	50€
Smartphone	190€	3 anys	26,39€
Total			76,39€

TAULA 4.3: Estimació cost recursos hardware

- **Despeses generals**

En aquest apartat es tenen en compte les despeses associades a la realització del projecte com poden ser la energia elèctrica consumida o l'accés a internet.

Producte	Cost	Període	Total estimat
Energia elèctrica	0,15€/kWh*0,15kW	409 h	9,20€
Llicència desenvolupador <i>Android</i>	19€	Indefinit	19€
Accés a internet	25€	5 mesos	125€
Total			153,2€

TAULA 4.4: Estimació cost despeses generals

- **Costos directes**

En aquest apartat s'agrupen tots els costos directes que suposa el desenvolupament d'aquest projecte.

Concepte	Cost
Recursos humans	13.700€
Software	73€
Hardware	77€
Total	13.850€

TAULA 4.5: Estimació costos directes

- **Costos d'imprevistos**

L'únic cost imprevist que podria sorgir és una fallada en l'ordinador o l'smartphone que s'utilitzen pel desenvolupament del projecte. En cas de fallada s'hauria d'afegir el cost de reparació, aproximadament uns 150 € en cas de l'ordinador i 60 € en cas del dispositiu *Android*. Durant el transcurs de la reparació s'utilitzaria un segon aparell per tal de no bloquejar el desenvolupament del projecte. La probabilitat d'error d'ambdós aparells és d'un 15 %.

Producte	Cost Reparació	Probabilitat d'error	Total estimat
Ordinador	150€	15%	22,5€
Dispositiu <i>Android</i>	60€	15%	9€
Total	-	-	31,5€

TAULA 4.6: Estimació cost d'imprevistos

- Cost total del projecte

Concepte	Cost
Costos directes	13.850€
Despeses generals	153,2€
Cost d'imprevistos	31,5€
Contingències (5%)	701,7€
Total	14.740€

TAULA 4.7: Estimació cost total del projecte

4.2 Control de gestió

Per al control de gestió es planteja omplir unes taules amb les hores reals que dur a terme cadascun dels rols. Cada dia s'han d'actualitzar i d'aquesta manera es pot portar un control i veure possibles desviacions en la planificació.

Dia	Cap de Projecte	Analista	Dissenyador	Programador	Tester
28/02/17	3h	-	-	-	-

TAULA 4.8: Control d'hores reals

Al final del projecte es farà un recompte de les hores reals i es compararan amb les estimades, d'aquesta manera es podrà analitzar la precisió de l'estimació. Les dades es presentaran en una taula com la que es mostra a continuació.

Rol	Hores estimades	Cost estimat	Hores reals	Cost real
Cap de projecte	160h	7.200€	170h	7.650€
Analista	45h	1.350€	65h	1.900€

TAULA 4.9: Comparació cost estimat i cost real

Capítol 5

Anàlisi Inicial

En aquest apartat es descriu el procés realitzat durant la primera part del treball amb l'estudi de viabilitat, l'anàlisi de riscos, les funcions que s'han ideat per al sistema i una valoració i justificació de les tecnologies disponibles per a dur a terme el projecte.

5.1 Estudi de viabilitat

Per tal d'estudiar la viabilitat del projecte es necessari conèixer quines persones poden interaccionar amb el sistema i els termes legals associats per tal que la posada en funcionament del sistema compleixi amb la llei establerta.

5.1.1 Anàlisi de les parts interessades

En el cas concret de l'aplicació a desenvolupar es considera que només existeix un rol en la interacció. L'usuari pren el rol de viatger, que és qui interactua amb l'aplicació a l'hora d'introduir i consultar dades. Cal dir però que també existeixen dos tipus d'actors indirectes:

- Persona de contacte: és a qui es truca en cas de que l'usuari tingui alguna emergència. No interactua amb l'aplicació però en forma part indirectament.
- Viatger no usuari: són els companys de viatge de l'usuari en cas de que els tingui, no interactuen amb l'aplicació però se'ls envia avisos en cas de tenir deutes pendents amb l'usuari. A més, se'ls pot etiquetar en notes.

5.1.2 Limitacions legals implicades

En la gestió i emmagatzematge de dades sensibles de l'usuari s'ha tingut en compte que, segons la Llei Orgànica de Protecció de Dades (LOPD), les dades externes que guarda el projecte són dades de nivell baix de seguretat. La LOPD pretén garantir el dret a controlar què es fa amb les nostres dades, saber qui té informació sobre nosaltres, quina informació té, d'on l'ha obtingut, per a quina finalitat té les dades i si té intenció de facilitar-les a un tercer.

Per complir amb la llei, abans de que el sistema es posi en funcionament caldrà realitzar una sèrie d'accions:

- Inscripció de fitxers: s'ha d'informar a l'Agència Espanyola de Protecció de Dades (AEPD) de l'existència dels fitxers que es van a utilitzar. La AEPD comprovarà si es compleixen els requisits i els incriurà.

- Elaboració document de seguretat: s'ha d'elaborar un document on es descriuran les mesures de seguretat aplicades a cada fitxer. Ha de ser respectat obligatoriament per tot el personal i s'ha de mantenir actualitzat, afegint les modificacions pertinents en cas que el nivell de protecció de dades hagi de canviar.
- Deure d'informació: s'ha d'informar a l'usuari del tractament que se'n farà i dels destinataris a qui es farà arribar aquesta informació.
- Seguretat de les dades: s'ha de garantir que les dades no es perden, per aquest motiu, serà necessari crear còpies de seguretat.

5.2 Anàlisi de riscos

En el capítol 2 hem vist l'apartat d'Obstacles on s'identifiquen una sèrie de riscos que té el projecte. En aquest punt es mostra estratègies per tal d'evitar-los i un pla per afrontar el problema provocat per si les estratègies anteriors fallen.

5.2.1 Estratègies de prevenció

Tenint en compte els riscos que podrien sofrir el sistema un cop en funcionament, és important fer una previsió de les accions que es duen a terme per tal d'evitar que els riscos es materialitzin en problemes reals.

- Nivell baix de bateria.
L'usuari és l'encarregat d'activar les notificacions per nivell de bateria baix en el sistema operatiu del seu mòbil.
- Connexió a internet a l'estranger.
Cada cop és més corrent tenir contractades dades fora del país de residència, tot i així, és l'usuari qui ha de buscar connexió a internet a través de xarxes d'internet gratuïtes o de pagament.
- Caiguda del servei web.
Es buscaran serveis de qualitat amb una taxa de caigudes molt baixa.
- Caiguda de la base de dades.
Es buscarà un proveïdor de base de dades al núvol que ofereixi un servei de qualitat i amb una taxa de caigudes molt baixa. A l'hora de efectuar el manteniment, es faran proves prèviament per tal de que quan s'apliquin els canvis no hi hagi problemes.
- Canvis en la legislació.
Aquest és un risc que no es pot evitar.
- Desconfiguració de l'aplicació.
A causa de canvis en el sistema operatiu del dispositiu mòbil o actualitzacions de l'aplicació aquesta pot deixar de funcionar correctament.
- Canvis en la resposta del servei web.
Es buscaran serveis web estables i amb una certa maduresa.

5.2.2 Plans per afrontar obstacles

En el cas que sigui impossible evitar l'aparició de problemes, s'ha elaborat un pla d'actuació per afrontar aquests possibles obstacles i així minimitzar el problema un cop ha aparegut.

- **Nivell baix de bateria.**
Es tasca de l'usuari tenir bateria al dispositiu mòbil, d'altra manera l'aplicació no pot funcionar.
- **Connexió a internet a l'estranger.**
Es tasca de l'usuari obtenir connexió a internet a l'estranger. En cas de no tenir connexió, l'aplicació suggerirà a l'usuari a través d'un avís que es connecti a una xarxa wifi.
- **Caiguda del servei web.**
En cas que un usuari accedeixi a un servei web no disponible, es bloquejarà la funcionalitat que l'utilitza i es notificarà a l'equip de manteniment.
- **Caiguda de la base de dades.**
Si en el moment d'emmagatzemar o consultar dades la base de dades no està operativa, es mostrarà un missatge d'error alertant a l'usuari. En aquest missatge s'inclourà informació per notificar l'error i tractar de solucionar-lo ràpidament.
- **Canvis en la legislació.**
Estudiar l'impacte que té sobre el sistema i fer les modificacions corresponents per a complir amb la nova llei establerta.
- **Desconfiguració de l'aplicació.**
L'aplicació inclourà una secció d'ajuda on l'usuari pot trobar solució a problemes freqüents. Si el problema persisteix, l'usuari haurà de posar-se en contacte amb el servei tècnic.
- **Canvis en la resposta del servei web.**
Si l'aplicació detecta problemes a l'hora de llegir la informació aquesta bloquejarà la funcionalitat que afecta i notificarà a l'equip de manteniment.

5.3 Visió global de les funcionalitats del sistema

Finalment, es llisten les funcionalitats que oferirà el sistema un cop completat el seu desenvolupament.

- **Gestió de despeses del viatge**
La funcionalitat de gestió de despeses es basa en portar un control de les despeses del viatge. Es divideix en dos parts: el control de les despeses total i el control de deutes entre l'usuari i els altres viatgers. El control de despeses total es basa en anar afegint les despeses adjuntant-hi un motiu i una localització. D'altra banda, el control de deutes entre els viatges es basa en que si l'usuari paga alguna cosa a un altre viatger s'anota la quantitat en forma de deute.
- **Gestió de cerques de llocs d'interès i com arribar-hi**
La funcionalitat de cerques de llocs d'interès es basa en que l'usuari pot llençar una cerca sobre: restaurants, bars, botigues, art i espais oberts, que hi ha a prop de la seva localització o de qualsevol altre punt si aquest ho desitja.

L'aplicació respondrà mostrant un conjunt de punts representats en el mapa ordenats segons les seves preferències a l'hora de viatjar i des d'on podrà consultar la ruta.

- **Gestió d'emergències**

La funcionalitat de gestió d'emergències té diverses parts. La primera és l'opció de buscar l'hospital més proper mostrant la ruta sobre el mapa. També ofereix l'opció de trucar al telèfon d'emergències del país on es troba l'usuari i, per últim, ofereix l'opció de trucar fàcilment al telèfon de contacte que ha introduït l'usuari a l'inici del viatge.

- **Previsió meteorològica**

La funcionalitat de previsions meteorològiques ofereix la previsió de la setmana del lloc on es troba l'usuari en el moment de la consulta.

- **Gestió de notes**

La funcionalitat de gestió de notes pretén ser un suport a l'hora de guardar curiositats, anèctodes o fets rellevants que succeeixen durant el viatge. A més, l'aplicació dona l'opció de compartir la nota amb els participants del viatge en forma de correu electrònic.

5.4 Anàlisi de les tecnologies candidates

5.4.1 Plataforma

En ser una aplicació orientada al món viatger, es considera que la tecnologia més adequada és una que permeti mobilitat. A continuació es mostren els avantatges i inconvenients de la tecnologia candidata, el telèfon mòbil.

Avantatges:

- Ofereix mobilitat a l'usuari, pot ser utilitzat en qualsevol lloc.
- Permet l'accés a internet.
- Permet fer trucades telefòniques.
- Gairebé la totalitat de la població disposa d'accés a aquesta tecnologia.
- Molt fàcil d'utilitzar i instal·lar.

Inconvenients:

- Necessita connexió permanent.
- El tamany de la pantalla és limitat.
- Introduir dades és incòmode.
- Dependència de la bateria.

Per aquesta aplicació concreta, clarament pesen més els avantatges que els inconvenients. És dona molta importància al fet que el dispositiu mòbil està sempre a prop de l'usuari i aquest està molt acostumat a consultar informació o rebre notificacions a aquest tipus de dispositiu. Per tant, no hi ha dubte a l'hora de desenvolupar l'aplicació per a un dispositiu mòbil.

5.4.2 Serveis externs

En aquest apartat s'analitzen diferents alternatives a l'hora de seleccionar quins serveis externs són els més adients.

Serveis de localització

– Google Maps

Google Maps és l'aplicació més desenvolupada del sector i la que ofereix més serveis com:

- * Conversió de coordenades geogràfiques a direccions.
- * Dades d'elevació.
- * Rutes de navegació GPS.
- * Localització geogràfica.
- * Càlcul de temps de viatge i distància entre diverses destinacions.
- * Càlcul de les indicacions entre diverses destinacions.
- * Informació sobre els llocs.

– Foursquare

- * Cerca de llocs d'interés en una àrea concreta.
- * Connexió amb usuaris de l'aplicació.
- * Control d'accés als punts.

Aquesta aplicació retorna les dades en format JSON amb molts paràmetres que donen informació sobre el punt d'interés seleccionat. Entre aquests paràmetres n'hi ha un que és molt interessant ja que ens revela el nivell de preu de l'establiment.

Un cop fet l'anàlisi s'ha decidit que s'utilitzarà Foursquare per tal de fer les cerques de llocs d'interés, ja que la resposta en format JSON és perfecta per tal de obtenir la informació i poder filtrarla, i Google Maps per tal de contruir les rutes i per representar les dades al mapa ja que és l'aplicació que ofereix el servei de manera més intuïtiva i gairebé la totalitat dels usuaris estan habituats a utilitzar-la.

Serveis de meteorologia

– OpenWeatherMap

OpenWeatherMap ofereix gratuïtament accés a la meteorologia del moment, a la previsió fins a 5 dies, a més també ofereix informació sobre el mapa. Les ciutats poden ser identificades per nom, per id o per coordenades geogràfiques. A més, ofereix serveis com:

- * Meteorologia actual per un lloc concret.
- * Previsió per 5 dies amb actualitzacions cada 3 hores.
- * Mapes de precipitació, pressió, núvols i temperatura.

– DarkSky

DarkSky ofereix les primeres 1000 crides del dia gratuïtament i ofereix informació sobre el clima a una setmana vista. Les següents crides del dia es paguen a \$0.0001 la crida. Ofereix la possibilitat d'obtenir la informació en molts idiomes inclosos tots els que vol oferir l'aplicació. Els serveis que ofereix són:

- * Meteorologia actual i previsió per una setmana.

* Dades històriques.

Després d'analitzar el funcionament i la documentació d'ambdues API es decideix utilitzar el servei d'OpenWeatherMap ja que la documentació és de més qualitat i ofereix grans facilitats en la connexió.

Serveis telèfons d'emergència

– Emergency Number

Aquesta API ens ofereix com a resposta informació de telèfons d'emergència de diferents tipus del país que li indiquem. L'únic problema que hi ha és que el país s'ha d'indicar per codi. Per tal de transformar el nom del país en el codi pertinent s'utilitza un altre servei web.

El servei analitzat és un servei molt complert que té informació de garibé tots els racons del món, per aquest motiu és l'el·legit a l'hora d'aconseguir els telèfons d'emergència del país que sigui necessari.

5.4.3 Base de dades

En aquest apartat s'explica el resum de l'anàlisi de diferents possibilitats a l'hora d'emmagatzemar les dades del sistema.

Com s'ha vist en apartats anteriors, l'aplicació vol estar preparada per tal d'acollir noves funcionalitats. Aquest aspecte fa que sigui molt adequat l'ús d'una base de dades no relacional ja que entre les seves característiques destaca el fet de no haver de definir un esquema previ.

Dins el món de les bases de dades no relacionals existeixen diversos tipus de bases de dades. Un cop estudiades totes les possibilitats es decideix que el tipus més adequat per al projecte és una base de dades clau-valor. Les bases de dades clau-valor són caracteritzades per ser molt eficients, fàcils d'utilitzar i, normalment, molt escalables. El fet que escalin fàcilment és un punt molt important també tenint en compte els objectius del projecte.

Un altre aspecte que s'estudia és treballar sobre una base de dades al núvol. Es decideix que és la millor opció ja que, un cop més, com l'aplicació pretén ser fàcilment escalable, tenir les dades al núvol permetria comunicació entre dispositius facilitant així molt la feina.

Un cop fet aquest estudi es busquen diferents tecnologies que permetin l'emmagatzematge de dades al núvol amb una base de dades clau-valor. La tecnologia que més s'acosta a les necessitats del projecte és *Firebase*. *Firebase* és un servei que proporciona al desenvolupador una base de dades al núvol amb llibreries per accedir a ella des d'aplicacions web, iOS i Android. A més, aquesta tecnologia ofereix suport per a la gestió d'usuaris, documentació de molta qualitat i, sobretot, una gran facilitat per integrar-la en Android.

5.5 Conclusions de l'anàlisi inicial

Tenint un coneixement més ampli del sector i considerant com s'ha plantejat el projecte es considera que es pot tirar endavant amb garanties d'èxit tot i tenir en compte els possibles aspectes negatius que repercuteixen el projecte (accessibilitat a les dades, llei, etc).

Capítol 6

Requisits

6.1 Propietats i hipòtesis sobre el domini

Per a poder desenvolupar aquest projecte, s'ha considerat necessari definir una sèrie d'expectatives sobre l'ús del sistema.

- **Connexió a internet**
L'usuari té connexió a internet en tot moment ja que es necessita a l'hora de guardar i consultar dades, i a l'hora d'accedir a serveis web.
- **Ús del sistema**
Els usuaris utilitzaran el sistema d'acord amb la finalitat per al qual ha sigut creat sense fer-ne un mal ús o emprenen accions malicioses que puguin trencar la integritat del sistema. Tot i així, s'implementaran mitjants de seguretat per protegir l'aplicació en aquest aspecte.
- **Obtenció de dades correctes**
Les dades que s'obtenen de serveis web són correctes i actualitzades.

6.2 Restriccions

Restricció de la solució

- Descripció: L'aplicació serà desenvolupada com a aplicació mòbil Android.
- Justificació: Els dispositius mòbils estan sempre aprop de l'usuari i disponibles per al seu ús, a més, en els últims anys s'ha extès l'ús d'internet al mòbil. Pel que fa al sistema operatiu, degut a la limitació temporal del treball i a que té una quota de mercat molt gran a Espanya s'ha optat per la implementació en Android, encara que no es descarten altres sistemes operatius en un futur.

Restricció temporal

- Descripció: El sistema ha d'estar acabat a meitats de juny de 2017.
- Justificació: S'estableix una data límit per presentar el projecte.

Restricció econòmica

- Descripció: El desenvolupament del projecte es durà a terme sense cap cost econòmic.
- Justificació: Com que no es disposa de cap inversió econòmica i el projecte es desenvolupa en un entorn acadèmic les eines i serveis utilitzats suposaran un cost molt baix.

6.3 Diagrama de casos d'ús

En la figura 5.1 es mostra el diagrama de casos d'ús del sistema sense tenir en compte la gestió d'usuaris.

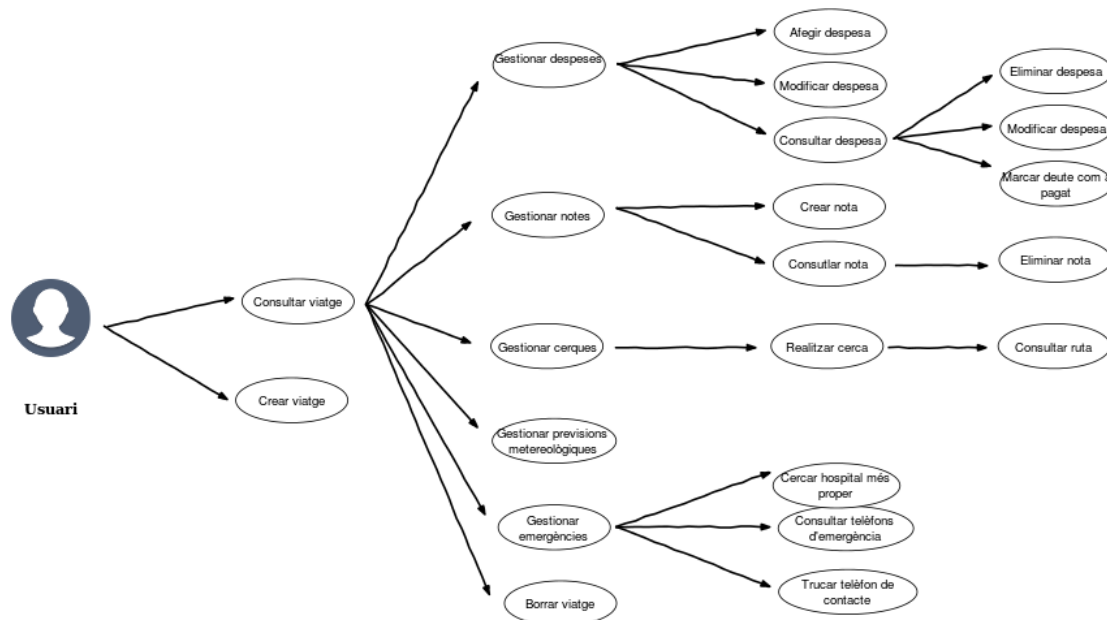


FIGURA 6.1: Diagrama de casos d'ús

6.4 Casos d'ús

Cas d'ús	#1	Registrar-se al sistema
Actor principal	Usuari	
Precondició	L'usuari té l'aplicació instal·lada al seu dispositiu.	
Trigger	L'usuari es vol registrar al sistema.	
Escenari principal d'èxit		
1. L'usuari selecciona el botó de registrar-se utilitzant el seu compte de Google.		
2. El sistema redirigeix el nou registre a Google.		
3. L'usuari introdueix el seu correu electrònic i la seva contrassenya.		
4. Google valida les dades.		
5. L'usuari accepta donar permisos a l'aplicació per obtenir les dades del seu perfil.		
6. El sistema registra l'usuari i inicia sessió automàticament.		
Extensions		
4.a El sistema detecta algun error en els camps introduïts.		
4.a.1 El sistema notifica a l'usuari de l'error.		
4.a.2 Es torna al pas 3.		
5.a L'usuari no dona permisos a l'aplicació per obtenir les dades del seu perfil.		
5.a.1 El sistema torna al pas 1.		

TAULA 6.1: Cas d'ús *Registrar-se al sistema*

Cas d'ús	#2	Iniciar sessió al sistema
Actor principal	Usuari	
Precondició	L'usuari està registrat al sistema.	
Trigger	L'usuari vol iniciar sessió al sistema.	
Escenari principal d'èxit		
1. L'usuari introdueix el seu correu electrònic i contrasenya, i indica al sistema que vol identificar-se.		
2. El sistema valida les dades, identifica l'usuari i redirigeix a l'usuari a la pantalla de viatge actual.		
Extensions		
2.a El sistema detecta algun error en els camps introduïts.		
2.a.1 El sistema notifica a l'usuari que hi ha dades incorrectes.		
2.a.2 Es torna al pas 1.		
3.a L'usuari no té cap viatge actiu.		
3.a.1 El sistema redirigeix l'usuari a la pantalla de llista de viatges realitzats.		

TAULA 6.2: Cas d'ús *Iniciar sessió al sistema*

Cas d'ús	#3	Tancar sessió al sistema
Actor principal	Usuari	
Precondició	L'usuari ha iniciat sessió al sistema.	
Trigger	L'usuari vol tancar la sessió al sistema.	
Escenari principal d'èxit		
1. L'usuari indica al sistema que vol tancar sessió.		
2. El sistema es desvincula de les dades de l'usuari actual i redirigeix l'aplicació a la pantalla d'iniciar sessió.		

TAULA 6.3: Cas d'ús *Tancar sessió al sistema*

Cas d'ús	#4	Crear viatge
Actor principal	Usuari	
Precondició	L'usuari ha iniciat sessió al sistema.	
Trigger	L'usuari vol iniciar un nou viatge.	
Escenari principal d'èxit		
<div>1. L'usuari indica que vol iniciar un nou viatge.</div> <div>2. El sistema redirecciona l'usuari a la pàgina de creació de viatge.</div> <div>3. L'usuari introdueix el nom del viatge,el destí, la data d'inici i la data final, el nom i correu electrònic de tots els viatgers que formaran part del viatge i indica que vol iniciar el viatge.</div> <div>4. El sistema valida les dades i presenta a l'usuari el formulari de preferències.</div> <div>5. L'usuari respon el formulari i indica que ja ha acabat.</div> <div>6. El sistema guarda el resultat del formulari i presenta a l'usuari el formulari per a que introdueixi les dades del contacte d'emergència.</div> <div>7. L'usuari introdueix les dades i indica que ja ha acabat.</div> <div>8. El sistema guarda el resultat, crea el viatge amb les dades introduïdes i redirigeix l'usuari a la pantalla principal del viatge.</div>		
Extensions		
<div>4.a El sistema detecta errors en les dades introduïdes.</div> <div><div>4.a.1 El sistema notifica a l'usuari que hi ha dades incorrectes.</div><div>4.a.2 Es torna al pas 2.</div></div> <div>6.a El sistema detecta que no s'ha completat el formulari.</div> <div><div>6.a.1 El sistema notifica a l'usuari que hi ha camps incomplets.</div><div>6.a.2 Es torna al pas 5.</div></div>		

TAULA 6.4: Cas d'ús *Crear viatge*

Cas d'ús	#5	Consultar llista de viatges
Actor principal	Usuari	
Precondició	L'usuari ha iniciat sessió al sistema.	
Trigger	L'usuari vol consultar viatges o ha iniciat sessió i no té cap viatge en curs.	
Escenari principal d'èxit		
1. L'usuari indica que vol consultar els seus viatges.		
2. El sistema redirigeix l'usuari a la pantalla de llista de viatges.		

TAULA 6.5: Cas d'ús *Consultar llista de viatges*

Cas d'ús	#6	Consultar despeses
Actor principal	Usuari	
Precondició	L'usuari ha accedit a un viatge i ha iniciat sessió.	
Trigger	L'usuari vol accedir a la funcionalitat gestió de despeses.	
Escenari principal d'èxit		
1. L'usuari indica que vol accedir a la funcionalitat gestió de despeses.		
2. El sistema el redirigeix a la pantalla d'estadístiques de despeses on es mostra la despesa total, la mitjana €/dia i la llista de les despeses.		

TAULA 6.6: Cas d'ús *Consultar despeses*

Cas d'ús	#7	Afegir nova despesa
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la funcionalitat gestió de despeses compartides.	
Trigger	L'usuari vol afegir una nova despesa.	
Escenari principal d'èxit		
1. L'usuari indica que vol afegir una nova despesa.		
2. El sistema redirigeix l'usuari a la pantalla de nova despesa.		
3. L'usuari introdueix la quantitat que paga.		
4. El sistema guarda les dades, crea la nova despesa, actualitza les estadístiques.		
Extensions		
3.a L'usuari notifica que vol crear un deute.		
3.a.1 L'usuari indica el participant deutor i la quantitat.		
3.a.2 El sistema guarda els deutes i envia un avís als deutors.		
4.a El sistema detecta dades incompatibles introduïdes per l'usuari.		
4.a.1 El sistema notifica a l'usuari que s'han detectat dades incompatibles.		
4.a.2 Es torna al pas 2.		

TAULA 6.7: Cas d'ús *Afegir nova despesa*

Cas d'ús	#8	Eliminar despesa
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la funcionalitat gestió de despeses compartides i ha creat almenys una despesa.	
Trigger	L'usuari vol eliminar una despesa.	
Escenari principal d'èxit		
1. L'usuari indica que vol eliminar una despesa en concret.		
2. El sistema mostra un diàleg on pregunta si està segur de voler eliminar la despesa.		
3. L'usuari confirma que vol eliminar la despesa.		
4. El sistema elimina la despesa, elimina els deutes relacionats i actualitza les estadístiques de despeses.		
Extensions		
3.a L'usuari cancel·la l'eliminació de la despesa.		
3.a.1 Es torna a la pantalla de gestió de despeses compartides.		

TAULA 6.8: Cas d'ús *Eliminar despesa*

Cas d'ús	#9	Consultar deutes de despesa
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la funcionalitat de gestió de despeses.	
Trigger	L'usuari vol consultar els deutes d'una despesa.	
Escenari principal d'èxit		
1. L'usuari indica que vol consultar els deutes d'una despesa.		
2. El sistema redirigeix l'usuari a la pantalla de consulta de deutes d'una despesa on mostra la llista dels deutes de la despesa indicada.		

TAULA 6.9: Cas d'ús *Consultar deutes de despesa*

Cas d'ús	#10	Cobrar deute
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la consulta de deutes d'una despesa i almenys existeix un deute per aquella despesa.	
Trigger	L'usuari vol marcar com a cobrat un deute.	
Escenari principal d'èxit		
1. L'usuari indica que vol marcar com a cobrat un deute.		
2. El sistema mostra un diàleg on pregunta si està segur de voler marcar com a cobrat el deute.		
3. L'usuari confirma que vol marcar com a cobrat el deute.		
4. El sistema marca que el deute està cobrat.		
Extensions		
3.a L'usuari cancel·la el cobrament del deute.		
3.a.1 Es torna a la pantalla de consultar deutes d'una despesa.		

TAULA 6.10: Cas d'ús *Cobrar deute*

Cas d'ús	#11	Modificar despesa
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la consulta de despeses i té almenys una despesa.	
Trigger	L'usuari vol modificar una despesa.	
Escenari principal d'èxit		
1. L'usuari indica que vol modificar una despesa.		
2. El sistema redirigeix l'usuari a la pantalla de modificació de despesa.		
3. L'usuari introdueix els canvis desitjats i indica que ja ha acabat.		
4. El sistema guarda els canvis i redirigeix l'usuari a la pantalla de consulta de despeses.		
Extensions		
3.a L'usuari cancel·la la modificació de la despesa.		
3.a.1 Es torna a la pantalla de consultar deutes d'una despesa.		

TAULA 6.11: Cas d'ús *Modificar despesa*

Cas d'ús	#12	Gestionar cerques
Actor principal	Usuari	
Precondició	L'usuari ha iniciat sessió i té un viatge en curs.	
Trigger	L'usuari vol accedir a la funcionalitat de cerques.	
Escenari principal d'èxit		
1. L'usuari indica que vol accedir a la funcionalitat de cerques.		
2. El sistema redirigeix l'usuari a la pantalla de realitzar una cerca.		

TAULA 6.12: Cas d'ús *Gestionar cerques*

Cas d'ús	#13	Realitzar una cerca
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la funcionalitat de gestionar cerques.	
Trigger	L'usuari vol realitzar una cerca.	
Escenari principal d'èxit		
1. L'usuari indica què vol cercar i la seva localització.		
2. El sistema crida a la API de cerques amb els paràmetres que ha introduït l'usuari.		
3. El sistema mostra els resultats que retorna la API ordenats tenint en compte l'estil de viatjar de l'usuari.		
Extensions		
1.a El sistema detecta que l'usuari no té activat el GPS al seu dispositiu.		
1.a.1 El sistema demana si pot activar el GPS.		
1.a.2 L'usuari accepta activar el GPS.		
1.a.3 El sistema executa el pas 2.		
1.a.2.a L'usuari no accepta activar el GPS.		
1.a.2.a.1 El sistema retorna a la pantalla inicial del cas d'ús.		
2.a El sistema no pot comunicar-se amb la API per falta de connexió a internet.		
2.a.1 El sistema notifica a l'usuari.		

TAULA 6.13: Cas d'ús *Realitzar una cerca*

Cas d'ús	#14	Consultar ruta
Actor principal	Usuari	
Precondició	L'usuari ha realitzat una cerca.	
Trigger	L'usuari vol consultar com arribar fins a un dels punts de la cerca.	
Escenari principal d'èxit		
1. L'usuari indica a quin resultat de cerca vol arribar		
2. El sistema fa una crida a la API i redirigeix l'usuari a la pantalla de ruta on mostra el resultat obtingut representat al mapa.		
Extensions		
1.a El sistema detecta que l'usuari no té activat el GPS al seu dispositiu.		
1.a.1 El sistema demana si pot activar el GPS.		
1.a.2 L'usuari accepta activar el GPS.		
1.a.3 El sistema executa el pas 2.		
1.a.2.a L'usuari no accepta activar el GPS.		
1.a.2.a.1 El sistema retorna a la pantalla inicial del cas d'ús.		
2.a El sistema no pot comunicar-se amb la API per falta de connexió a internet.		
2.a.1 El sistema notifica a l'usuari.		

TAULA 6.14: Cas d'ús *Consultar ruta*

Cas d'ús	#15	Consultar previsió metereològica
Actor principal	Usuari	
Precondició	L'usuari té un viatge en curs i ha iniciat sessió.	
Trigger	L'usuari vol accedir a la funcionalitat de consultar previsió metereològica.	
Escenari principal d'èxit		
1. L'usuari indica que vol accedir a la funcionalitat de consultar previsió metereològica.		
2. El sistema redirigeix l'usuari a la pantalla de previsió metereològica.		

TAULA 6.15: Cas d'ús *Consultar previsió metereològica*

Cas d'ús	#16	Gestionar emergències
Actor principal	Usuari	
Precondició	L'usuari té un viatge en curs i ha iniciat sessió.	
Trigger	L'usuari vol accedir a la funcionalitat de gestió d'emergències.	
Escenari principal d'èxit		
1. L'usuari indica que vol accedir a la funcionalitat de gestionar emergències.		
2. El sistema redirigeix l'usuari a la pantalla de gestió d'emergències.		

TAULA 6.16: Cas d'ús *Gestionar emergències*

Cas d'ús	#17	Trucar telèfon d'emergència
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la funcionalitat de gestionar emergències.	
Trigger	L'usuari vol fer una trucada al telèfon d'emergència.	
Escenari principal d'èxit		
1. L'usuari indica que vol trucar al telèfon d'emergència.		
2. El sistema redirigeix l'usuari a una trucada telefònica al número d'emergència del país on es troba.		

TAULA 6.17: Cas d'ús *Trucar telèfon d'emergència*

Cas d'ús	#18	Trucar telèfon de contacte
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la funcionalitat de gestionar emergències.	
Trigger	L'usuari vol realitzar una trucada al telèfon de contacte.	
Escenari principal d'èxit		
1. L'usuari indica que vol trucar al telèfon de contacte.		
2. El sistema redirigeix l'usuari a una trucada telefònica amb el telèfon de contacte.		

TAULA 6.18: Cas d'ús *Trucar telèfon de contacte*

Cas d'ús	#19	Consultar llista de notes
Actor principal	Usuari	
Precondició	L'usuari té un viatge en curs i ha iniciat sessió al sistema.	
Trigger	L'usuari vol accedir a la llista de notes.	
Escenari principal d'èxit		
1. L'usuari indica que vol accedir a la llista de notes.		
2. El sistema redirigeix l'usuari a la pantalla de llista de notes.		

TAULA 6.19: Cas d'ús *Consultar llista de notes*

Cas d'ús	#20	Crear nota
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la llista de notes.	
Trigger	L'usuari vol crear una nota.	
Escenari principal d'èxit		
1. L'usuari indica que vol crear una nota.		
2. El sistema el redirigeix a la pantalla de crear una nova nota.		
3. L'usuari introdueix títol, text, tria un color i pot compartir-la amb participants del viatge.		
4. El sistema guarda la nota i envia un correu electrònic amb el contingut de la nota als participants amb qui s'ha compartit.		

TAULA 6.20: Cas d'ús *Crear nota*

Cas d'ús	#21	Consultar nota
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la llista de notes i ha creat almenys una nota.	
Trigger	L'usuari vol consultar una nota concreta.	
Escenari principal d'èxit		
1. L'usuari indica quina nota vol consultar.		
2. El sistema redirigeix l'usuari a la pantalla de consultar nota i mostra la informació de la nota escollida.		

TAULA 6.21: Cas d'ús *Consultar nota*

Cas d'ús	#22	Eliminar nota
Actor principal	Usuari	
Precondició	L'usuari ha accedit a la pantalla de consultar nota	
Trigger	L'usuari vol eliminar la nota consultada.	
Escenari principal d'èxit		
1. L'usuari indica que vol eliminar la nota		
2. El sistema mostra un diàleg preguntant si està segur de voler eliminar la nota.		
3. L'usuari confirma que vol eliminar la nota.		
4. El sistema elimina la nota.		
Extensions		
3.a L'usuari cancel·la l'eliminació de la nota.		
3.a.1 Es torna a la pantalla inicial del cas d'ús.		

TAULA 6.22: Cas d'ús *Eliminar nota*

6.5 Requisits no funcionals

6.5.1 Requisits de disseny

- Requisit d'escalabilitat

Requisit no funcional	#01
Descripció	El disseny del software ha de permetre que l'aplicació sigui fàcilment escalable a l'hora d'afegir o modificar funcionalitats.
Justificació	En ser una aplicació que agrupa funcionalitats per facilitar la feina a l'usuari mentre viatja, s'ha de poder modificar fàcilment aquestes funcionalitats per tal d'adaptar-se a les necessitats i preferències del client un cop llençada l'aplicació o adaptar-se a noves idees que puguin sortir al mercat.
Criteri de satisfacció	S'analitzarà el software detingudament per assegurar que és fàcilment escalable.

TAULA 6.23: Requisit d'escalabilitat

- Requisit d'implementació

Requisit no funcional	#02
Descripció	Per tal d'implementar algunes funcionalitats l'aplicació ha d'utilitzar serveis web externs.
Justificació	En ser una aplicació que vol oferir diverses funcionalitats de temàtiques diferents però que tenen el món viatger en comú, és molt més eficient a l'hora del desenvolupament utilitzar serveis web externs.
Criteri de satisfacció	Almenys s'ha d'implementar una funcionalitat utilitzant serveis externs.

TAULA 6.24: Requisit d'implementació

6.5.2 Requisits de percepció

- Requisit d'aparença

Requisit no funcional	#03
Descripció	El disseny de la interfície és atractiu.
Justificació	Els usuaris veuran una estructura, combinació de colors i formes que els duren a interessar-se per la interacció amb l'aplicació.
Criteri de satisfacció	Es realitzarà una enquesta de satisfacció amb la intenció d'obtenir un resultat satisfactori del 75% o superior.

TAULA 6.25: Requisit d'aparença

- Requisit d'estil

Requisit no funcional	#04
Descripció	El disseny de la interfície és net i professional.
Justificació	Les pantalles han d'estar enfocades a l'objectiu que volen aconseguir prescindint d'elements innecessaris que puguin distreure l'usuari.
Criteri de satisfacció	Es realitzarà una enquesta de satisfacció amb la intenció d'obtenir un resultat satisfactori del 75% o superior.

TAULA 6.26: Requisit d'estil

6.5.3 Requisit d'usabilitat

- Requisit d'ús

Requisit no funcional	#05
Descripció	El sistema ha de ser fàcil d'utilitzar.
Justificació	El sistema ha de ser molt senzill d'utilitzar en el context d'un viatge.
Criteri de satisfacció	Es realitzarà una enquesta de satisfacció amb la intenció d'obtenir un resultat satisfactori del 75% o superior.

TAULA 6.27: Requisit d'ús

- Requisit d'aprenentatge

Requisit no funcional	#06
Descripció	Els usuaris han de poder utilitzar el sistema sense informació prèvia.
Justificació	L'aplicació ha de ser intuïtiva per tal de que l'usuari pugui utilitzar-la sense problemes des de la seva instal·lació.
Criteri de satisfacció	Es realitzarà una prova amb persones que no hagin utilitzat mai una aplicació similar a aquesta amb la intenció d'obtenir un 90% o superior de resultats positius.

TAULA 6.28: Requisit d'aprenentatge

- Requisit de comprensió

Requisit no funcional	#07
Descripció	El sistema utilitzarà un llenguatge que qualsevol viatger pugui entendre.
Justificació	El llenguatge que utilitza l'aplicació ha de poder ser entès per qualsevol viatger ja que qualsevol persona pot ser usuari d'aquesta aplicació.
Criteri de satisfacció	Es realitzarà una prova amb persones de diferents característiques amb la intenció d'obtenir un 90% o superior de resultats positius.

TAULA 6.29: Requisit de comprensió del llenguatge

Requisit no funcional	#08
Descripció	El sistema utilitzarà icones i símbols fàcilment interpretables.
Justificació	La simbologia de l'aplicació ha de ser fàcilment interpretable ja que durant un viatge l'usuari no vol dubtar en les seves decisions.
Criteri de satisfacció	Es realitzarà una enquesta de satisfacció amb la intenció d'obtenir un resultat satisfactori del 90% o superior.

TAULA 6.30: Requisit de comprensió dels símbols

6.5.4 Requisits de rendiment

Requisit no funcional	#09
Descripció	Els canvis fets al sistema s'emmagatzemaran en menys de 5 segons en condicions normals.
Justificació	Per tal de que l'usuari no hagi d'esperar, les dades que gestiona l'aplicació han de ser guardades amb el menor temps possible.
Criteri de satisfacció	Es realitzaran tests per tal de comptabilitzar el tems de creació, actualització i eliminació de dades.

TAULA 6.31: Requisit de latència i velocitat

Requisits de disponibilitat i fiabilitat

Requisit no funcional	#10
Descripció	El sistema estarà disponible 24h al dia els 365 dies de l'any.
Justificació	Els viatgers han de poder utilitzar l'aplicació quan ells la necessitin.
Criteri de satisfacció	El sistema ha de tenir una taxa de funcionament superior al 95%.

TAULA 6.32: Requisit de disponibilitat i fiabilitat

6.5.5 Requisits de funcionament i ambientals

- Requisits de funcionament

Requisit no funcional	#11
Descripció	L'aplicació, en la funcionalitat de cerques, presentarà la informació a l'usuari segons l'estil de viatjar del mateix.
Justificació	Per tal de filtrar les dades s'ordenaran segons el nivell de preus de l'establiment resultat de cerca.
Criteri de satisfacció	Es comprovarà que el filtratge funcioni en diverses proves fetes abans i durant el llançament.

TAULA 6.33: Requisit de funcionament

- Requisits de llançament

Requisit no funcional	#12
Descripció	S'informarà dels períodes de manteniment del sistema.
Justificació	Per tal que els usuaris estiguin assabentats de quan el sistema no pot ser utilitzat se'ls avisarà prèviament.
Criteri de satisfacció	S'enviarà un avís a tots els usuaris informant del tems durant el qual no podrà ser utilitzat el sistema.

TAULA 6.34: Requisit de manteniment

Requisit no funcional	#13
Descripció	El sistema no causarà fallades quan s'actualitzi.
Justificació	Quan l'aplicació sigui actualitzada tot continuarà funcionant correctament.
Criteri de satisfacció	Es realitzaran proves abans de llençar la nova versió per tal d'assegurar que no falla.

TAULA 6.35: Requisit de llançament

- Requisits ambientals

Requisit no funcional	#14
Descripció	El projecte i el sistema final són sostenibles.
Justificació	La realització del projecte ha de tenir un impacte positiu en l'entorn.
Criteri de satisfacció	Es realitzarà un informe de sostenibilitat.

TAULA 6.36: Requisit ambiental

6.5.6 Requisits de manteniment i suport

- Requisit de manteniment

Requisit no funcional	#15
Descripció	El sistema ha de ser flexible a futurs canvis i a la incorporació de noves funcionalitats.
Justificació	S'ha de poder afegir, treure o modificar funcionalitats del sistema.
Criteri de satisfacció	El sistema disposa de la documentació necessària per tal de que futurs desenvolupadors puguin treballar amb el sistema.

TAULA 6.37: Requisit de manteniment

- Requisit de suport

Requisit no funcional	#16
Descripció	El sistema disposa de secció d'ajuda.
Justificació	És important que el sistema inclogui una secció d'ajuda per tal de resoldre dubtes a l'usuari.
Criteri de satisfacció	S'inclou una secció d'ajuda a l'aplicació.

TAULA 6.38: Requisit de suport

6.5.7 Requisits de seguretat

- Requisits d'integritat

Requisit no funcional	#17
Descripció	El sistema realitzarà còpies de seguretat.
Justificació	El sistema ha de poder recuperar-se en cas de fallada.
Criteri de satisfacció	El sistema realitzarà còpies de seguretat diàries.

TAULA 6.39: Requisit d'integritat

- Requisits de privacitat

Requisit no funcional	#18
Descripció	El sistema informará a l'usuari de la política de privacitat de les dades que es guarden.
Justificació	Els usuaris han d'estar al corrent de que les dades es guarden de forma segura i poden confiar amb el sistema.
Criteri de satisfacció	El sistema permetrà als usuaris consultar la política de privacitat.

TAULA 6.40: Requisit de privacitat

6.5.8 Requisits legals

- Requisits de compliment

Requisit no funcional	#19
Descripció	Totes les dades de caràcter personal dels usuaris seran tractades d'acord amb la LOPD i la Directiva de Protecció de Dades 95/46 de la Unió Europea.
Justificació	El sistema ha de respectar la llei de protecció de dades.
Criteri de satisfacció	Es consultarà amb especialistes en el camp de la protecció de dades la metodologia usada.

TAULA 6.41: Requisit de compliment

Capítol 7

Especificació

7.1 Esquema conceptual

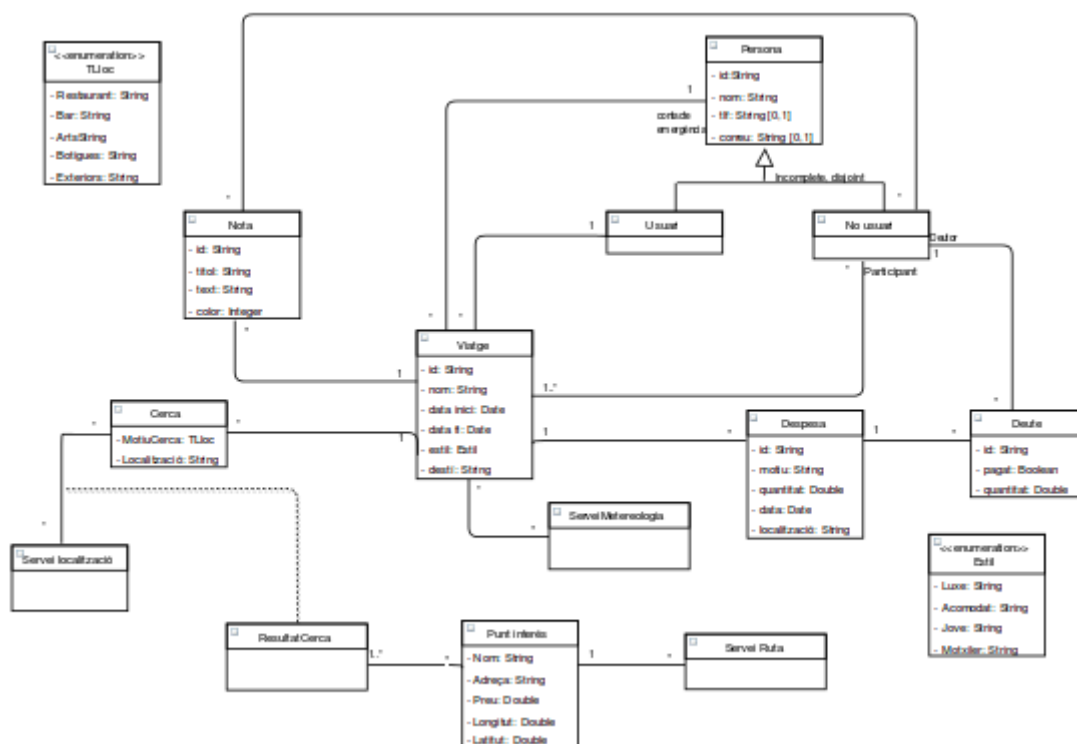


FIGURA 7.1: Esquema conceptual.

7.1.1 Restricions textuais

- El deutor ha de ser participant del viatge al qual pertany el deute.
- El viatger a qui se li comparteix una nota ha de ser participant del viatge al qual pertany la nota.

7.1.2 Descripció de les entitats

Persona

Super classe que s'instància per obtenir una persona de contacte.

Atribut	Descripció
Nom	Nom de la persona
Tlf	Telèfon de la persona
Correu	Correu electrònic de la persona

TAULA 7.1: Atributs de la classe Persona

Usuari

Té el rol d'usuari del sistema i és subclasse de Persona.

Atribut	Descripció
Nom	Nom de l'usuari
Tlf	Telèfon de l'usuari
Correu	Correu electrònic de l'usuari

TAULA 7.2: Atributs de la classe Usuari

NoUsuari

Té el rol de participant del viatge però sense ser usuari, no interacciona directament amb l'aplicació.

Atribut	Descripció
Nom	Nom del participant
Tlf	Telèfon del participant
Correu	Correu electrònic del participant

TAULA 7.3: Atributs de la classe NoUsuari

Viatge

Representa un viatge en concret, és el concepte base de l'aplicació.

Atribut	Descripció
Nom	Nom del viatge
Destí	Destí del viatge
Data inici	Data d'inici del viatge
Data fi	Data final del viatge
Estil	Estil de viatjar seleccionat per l'usuari

TAULA 7.4: Atributs de la classe Viatge

Despesa

Representa una despesa de l'usuari durant un viatge.

Atribut	Descripció
Motiu	Motiu de la despesa
Quantitat	Quantitat en euros de la despesa
Localització	Lloc on s'ha generat la despesa
Data	Data de creació de la despesa

TAULA 7.5: Atributs de la classe Despesa

Deute

Representa un deute d'un participant del viatge amb l'usuari.

Atribut	Descripció
Quantitat	Quantitat del deute en euros.
Pagat	Indica si el deute ja ha estat pagat o no.

TAULA 7.6: Atributs de la classe Deute

Nota

Representa una nota escrita per l'usuari.

Atribut	Descripció
Títol	Títol de la nota
Text	Text de la nota
Color	Color que ha seleccionat l'usuari pel fons de la nota.

TAULA 7.7: Atributs de la classe Nota

Cerca

Representa una cerca llençada per l'usuari al sistema.

Atribut	Descripció
Motiu	Tipus de cerca, aquest valor pot ser: menjar, begudes, arts, botigues o espais oberts
Localització	Localització des d'on es vol realitzar la cerca

TAULA 7.8: Atributs de la classe Cerca

Servei Localització

Representa el servei extern amb qui es comunica el sistema per obtenir resultats en la cerca.

Resultat Cerca

Representa els resultats d'una cerca en concret.

Punt d'interés

Representa un ítem concret del resultat de la cerca.

Atribut	Descripció
Nom	Nom del punt d'interés
Adreça	Adreça del punt d'interés
Preu	Nivell de preus del punt d'interés
Latitut	Valor de latitut en la localització del punt d'interés
Longitut	Valor de longitut en la localització del punt d'interés

TAULA 7.9: Atributs de la classe Punt d'interés

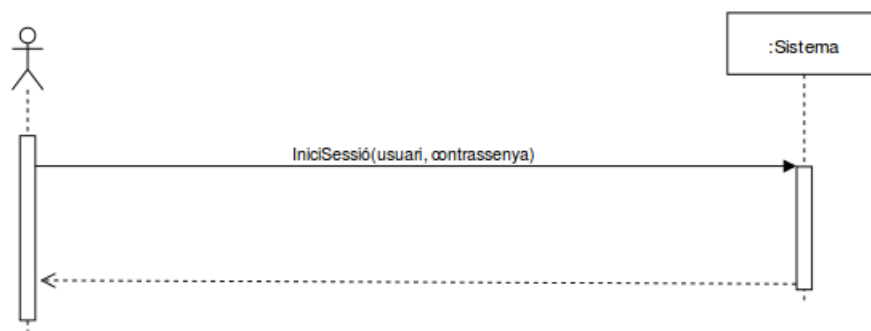
Servei Ruta

Representa el servei extern amb qui es comunica el sistema per tal d'obtenir una ruta entre dos punts.

Servei Meteorologia

Representa el servei extern amb qui es comunica el sistema per tal d'obtenir la previsió metereològica.

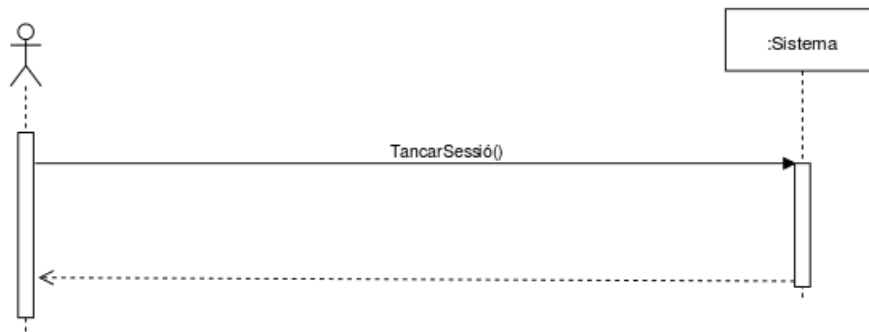
7.2 Esquema del comportament

Inici sessió

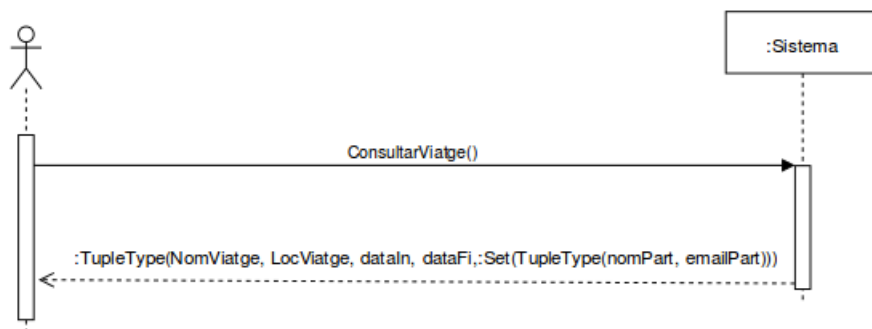
Context Sistema::IniciSessió(email:string, contrassenya:string)

Pre email no és buit
contrassenya no es buida

Post Si les credencials són correctes l'usuari veu la pantalla de llista dels seus viatges

Tancar sessió

Context Sistema::TancarSessió()
Pre L'usuari ha iniciat sessió
Post L'usuari veu la pantalla d'inici de sessió

Consultar viatge

Context Sistema::ConsultarViatge():TupleType(NomViatge, LocViatge, dataIn, dataFi,:Set(TupleType(nomPart, emailPart)))
Pre L'usuari ha iniciat sessió
Post L'usuari veu la pantalla i les dades del viatge

Crear viatge

Context Sistema::CrearViatge(nomViatge, LocViatge, dataIn, dataFi, :Set(nomPart, emailPart))

Pre L'usuari ha iniciat sessió

Post L'usuari veu la pantalla de seleccionar preferencies de viatge

Context Sistema::SetPreferencies(estil)

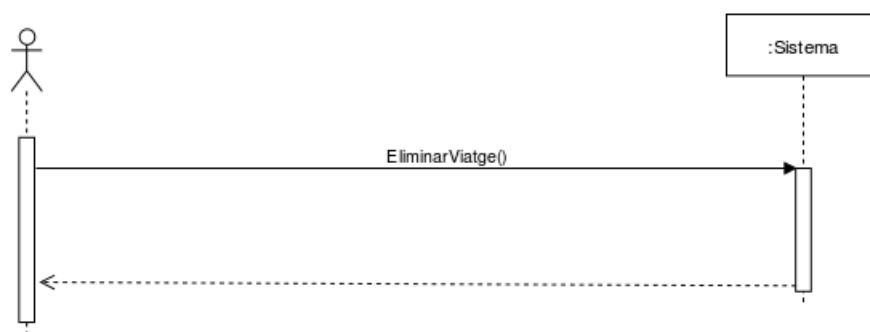
Pre L'usuari ha començat la creació d'un viatge

Post L'usuari veu la pantalla d'introduir les dades del contacte d'emergència

Context Sistema::SetContactesEmergencia(nom, email, tlf)

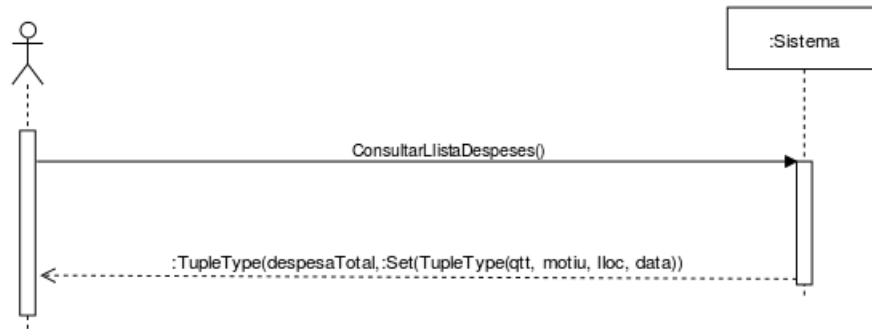
Pre L'usuari ha seleccionat les preferencies del viatge

Post S'ha creat el viatge amb totes les dades introduïdes

Eliminar viatge

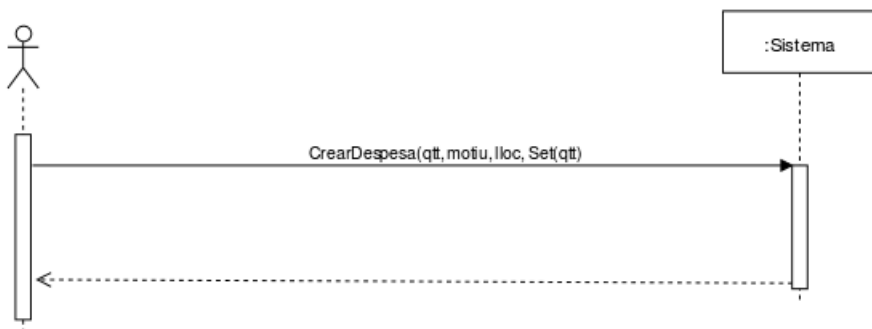
Context Sistema::EliminarViatge()
Pre L'usuari ha creat un viatge
Post S'elimina el viatge amb totes les dades associades

Consultar llista de despeses

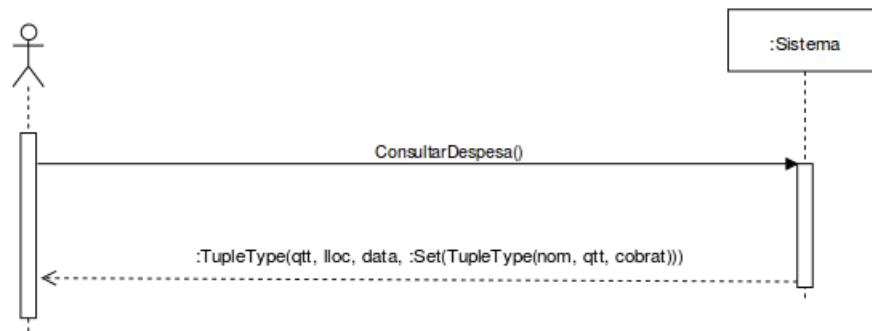


Context Sistema::ConsultarLlistaDespeses():TupleType(despesaTotal,Set(TupleType(qtt, motiu, lloc, data)))
Pre Ha accedit a consultar un viatge
Post L'usuari veu la pantalla amb la informació de totes les despeses del viatge

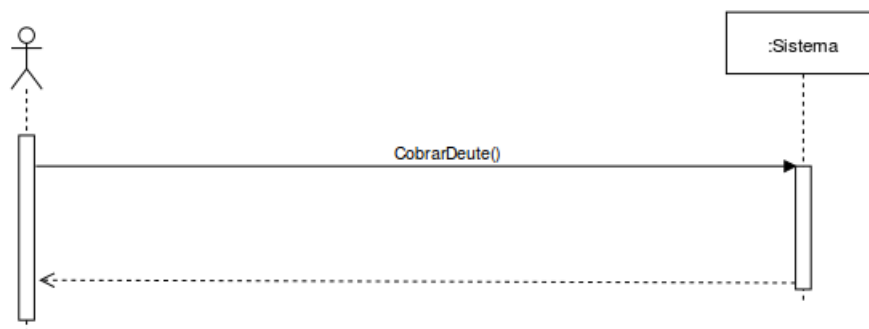
Crear despesa



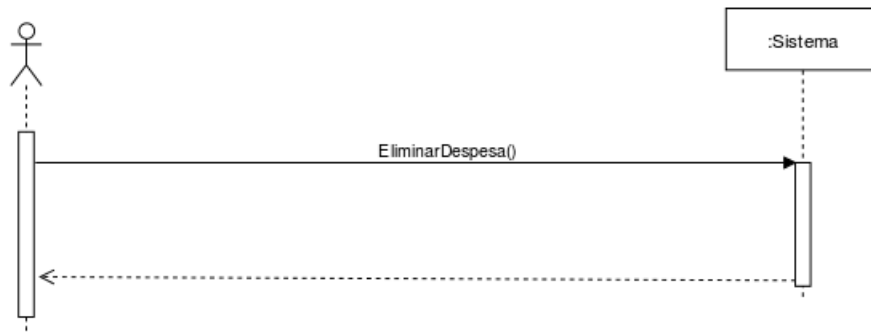
Context Sistema::CrearDespesa(qtt, motiu, lloc, Set(qtt))
Pre Ha accedit a consultar la llista de despeses
Post Es crea una instància de despesa amb les dades introduïdes

Consultar despesa

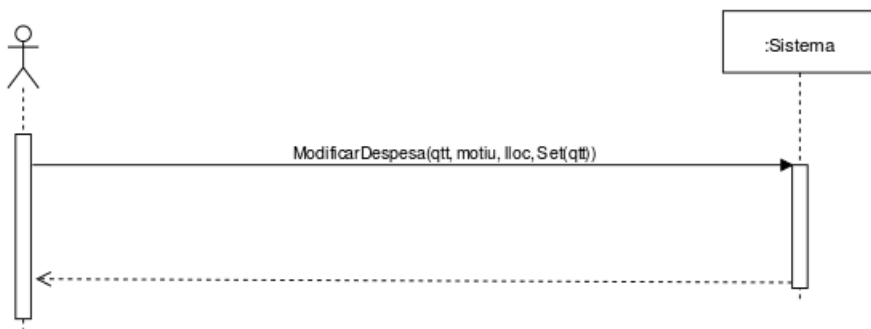
Context Sistema::ConsultarDespesa():TupleType(qtt, lloc, data, Set(TupleType(nom, qtt, cobrat)))
Pre L'usuari ha creat una despesa
Post L'usuari veu la pantalla de consultar despesa amb totes les dades d'aquesta

Cobrar deute

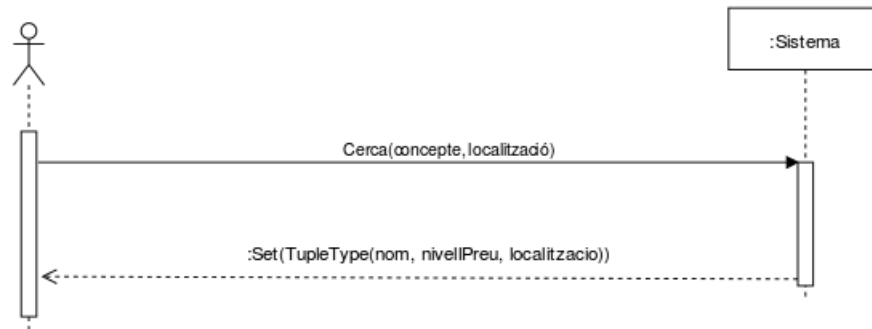
Context Sistema::CobrarDeute()
Pre L'usuari ha creat una despesa i té un deute pendent de cobrar
Post El sistema enregistra el deute com a cobrat

Eliminar despesa

Context Sistema::EliminarDespesa()
Pre L'usuari ha creat una despesa
Post El sistema elimina la despesa i tots els deutes associats

Modificar despesa

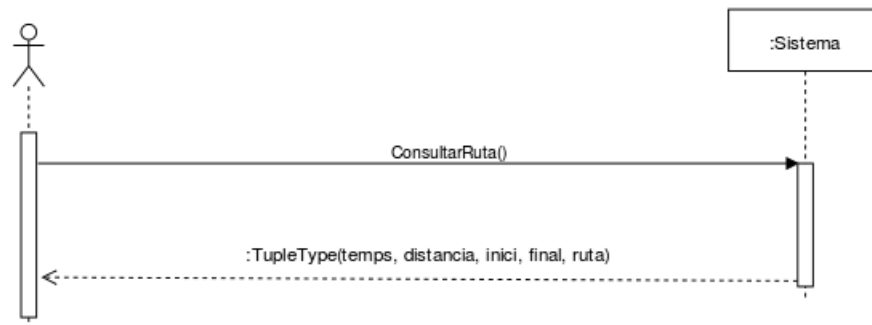
Context Sistema::ModificarDespesa(qtt, motiu, lloc, Set(qtt))
Pre L'usuari ha creat una despesa
Post El sistema modifica la despesa amb els nous paràmetres introduïts

Cerca

Context Sistema::Cerca(concepte, localització):Set(TupleType(nom, nivellPreu, localitzacio))

Pre L'usuari ha accedit a consultar un viatge

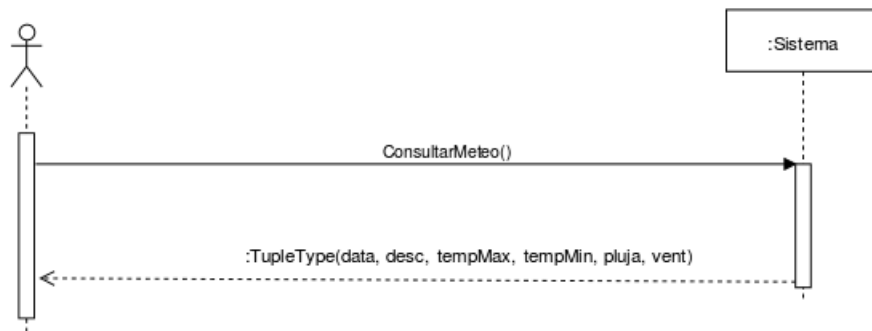
Post L'usuari veu la llista de resultats de la cerca

Consultar ruta

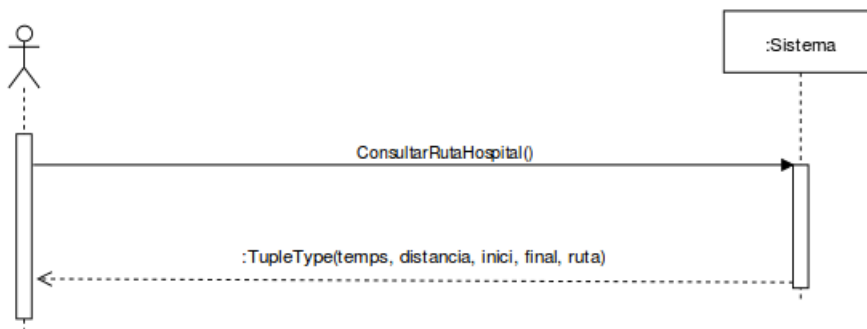
Context Sistema::ConsultarRuta():TupleType(temps, distancia, inici, final, ruta)

Pre L'usuari ha realitzat una cerca

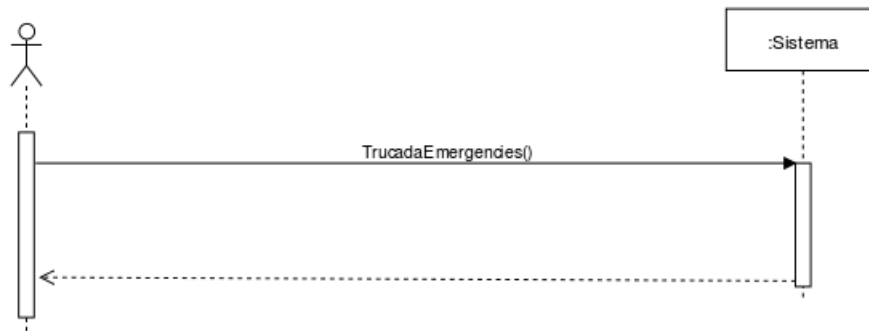
Post L'usuari veu la pantalla de consultar ruta amb tota la informació de la ruta

Consultar Meteorologia

Context Sistema::ConsultarMeteo():TupleType(data, desc, tempMax, tempMin, pluja, vent)
Pre L'usuari ha accedit a consultar un viatge
Post L'usuari veu la pantalla de consultar metereologia amb tota la informació de la previsió metereològica

Consultar ruta a l'hospital

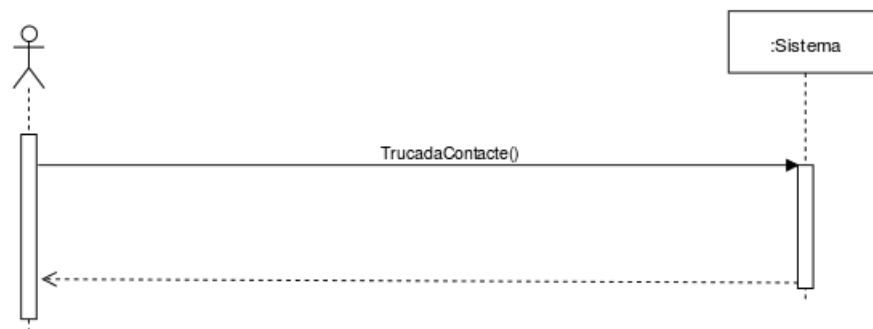
Context Sistema::ConsultarRutaHospital():TupleType(temps, distancia, inici, final, ruta)
Pre L'usuari ha accedit a la funcionalitat de gestionar emergències
Post L'usuari veu a la pantalla la ruta fins l'hospital més proper i tota la informació adicional

Trucar telèfon d'emergències

Context Sistema::TrucadaEmergencies()

Pre L'usuari ha accedit a la funcionalitat de gestionar emergències

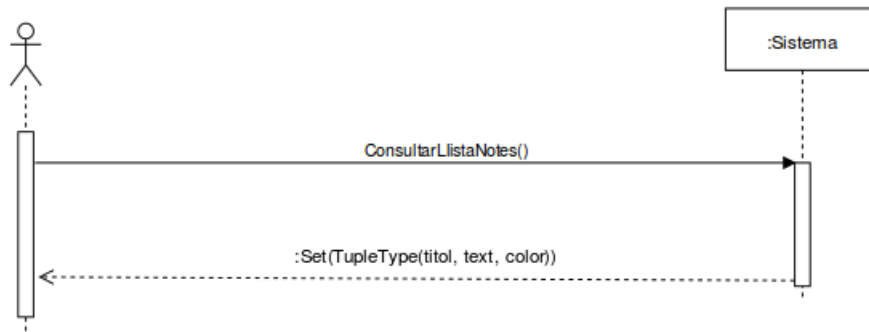
Post El sistema realitza una trucada al telèfon d'emergències del país on es troba l'usuari

Trucar telèfon de contacte

Context Sistema::TrucadaContacte()

Pre L'usuari ha accedit a la funcionalitat de gestionar emergències

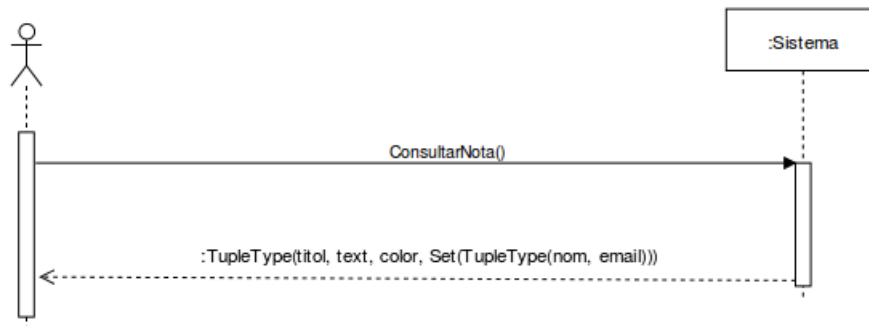
Post El sistema realitza una trucada al telèfon de contacte introduït per l'usuari

Consultar llista de notes

Context Sistema::ConsultarLlistaNotes():Set(TupleType(titol, text, color))

Pre L'usuari ha accedit a consultar un viatge

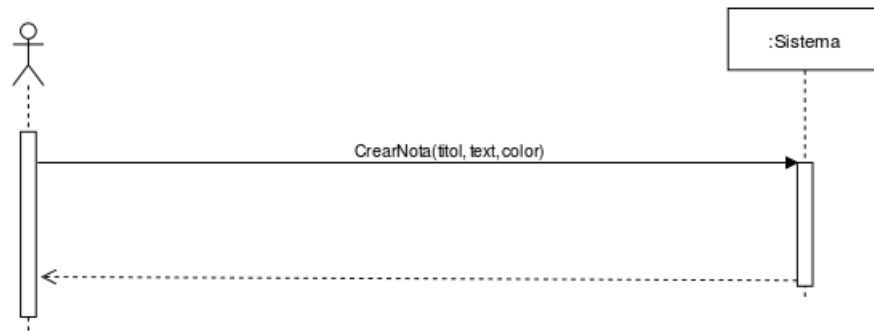
Post L'usuari veu la pantalla amb totes les notes del viatge

Consultar nota

Context Sistema::ConsultarNota():TupleType(titol, text, color, Set(TupleType(nom, email)))

Pre L'usuari ha consultat la llista de notes

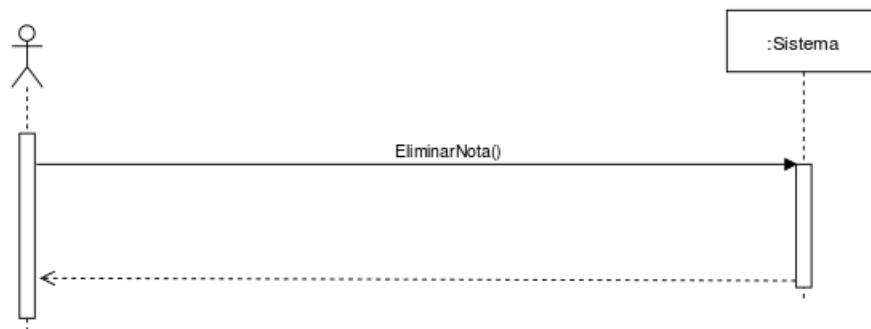
Post L'usuari veu la pantalla de consultar nota amb tota la informació d'aquesta

Crear nota

Context Sistema::CrearNota(titol, text, color)

Pre L'usuari ha accedit a consultar la llista de notes

Post El sistema crea una instància de nota amb les dades introduïdes i l'usuari veu la pantalla de consultar nota

Eliminar nota

Context Sistema::EliminarNota()

Pre L'usuari ha accedit a consultar una nota

Post El sistema elimina la nota

Capítol 8

Disseny

Aquest projecte consta del desenvolupament d'una aplicació *Android*, per aquest motiu l'arquitectura ha d'adaptar-se a les bones pràctiques d'aquesta plataforma.

El sistema separa clarament els conceptes en activitats diferents. Com que es tracta d'un dispositiu mòbil, aquestes activitats només són un llaç entre el sistema operatiu i l'aplicació, per tant, no sempre les controlem i en el cas que el dispositiu mòbil necessiti memòria pot eliminar l'activitat. Si separem els conceptes en activitats diferents minimitzem les dependències.

8.1 Arquitectura

Per al desenvolupament d'aquest projecte s'ha obtingut per utilitzar l'arquitectura que proporciona la plataforma *Android* per defecte. Aquesta consta d'un conjunt d'activitats que fan servir un model i controlen la interfície. En l'Activitat, només hi ha de constar el codi que faci de vincle entre el model i la vista, per aquest motiu, per tal de dur a terme tots els processos que es necessiten per obtenir la informació adequada s'ha dissenyat un servei (figura 8.1).

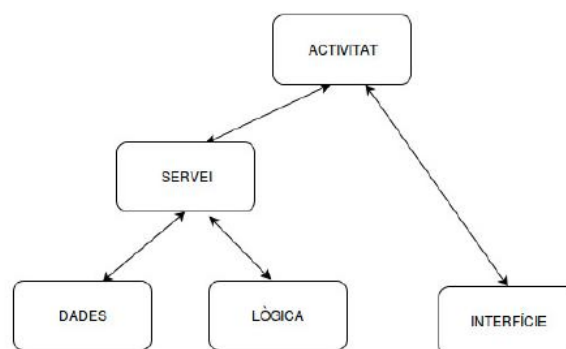


FIGURA 8.1: Arquitectura de l'aplicació.

Activitat

El sistema implementa una activitat per a cada funcionalitat diferent, d'aquesta manera, com s'ha exposat anteriorment, minimitzem les dependències i garantim més fiabilitat. No existeix un controlador que governi les activitats sinó que es cada activitat que sap en tot moment amb qui s'ha de comunicar i qui pot ser que es comuniqui amb ella.

Servei

Des de qualsevol activitat es pot accedir a un servei per tal d'obtenir dades que es volen utilitzar per mostrar-les a l'usuari. En la figura 8.2 es pot apreciar una part de l'estructura dels serveis, es declara un servei per cada temàtica, dins el servei es dur a terme les operacions que convinguin utilitzant les dades guardades en els repositoris.

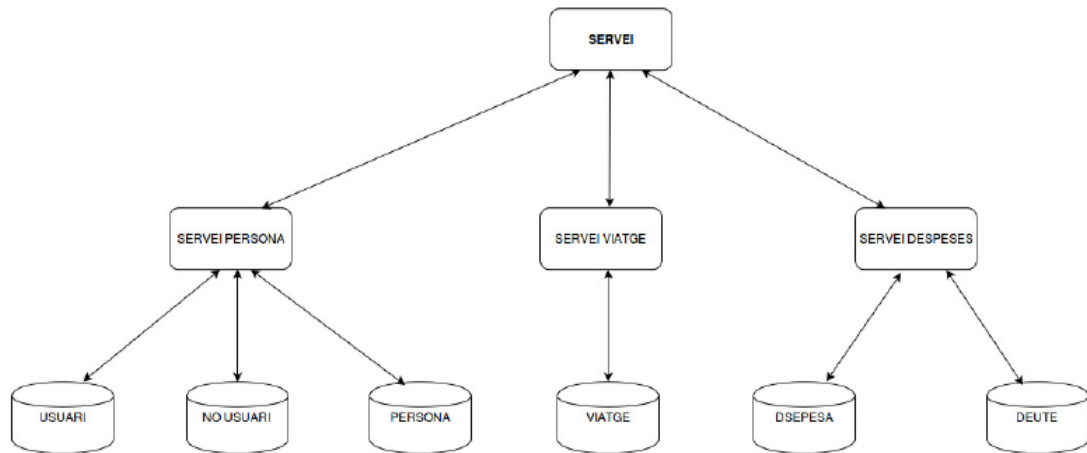


FIGURA 8.2: Estructura dels serveis.

Dades

Es tenen en compte dos tipus de fonts de dades, les que guardem i consultem de la base de dades i les que obtenim de les consultes a serveis externs. Des del punt de vista de l'activitat, no hi ha cap diferencia en accedir-hi.

En el cas de les dades guardades a la base de dades, la qual està en el núvol, s'utilitzen uns repositoris per tal de tenir les dades en local. Hi ha un repositori per cada tipus d'entitat a la base de dades. En la figura 8.3 es pot apreciar l'accés des de l'activitat a la base de dades.

Pel que fa a les consultes a serveis externs no s'utilitzen repositoris, des del servei es fa una crida directament a la API corresponent per tal d'obtenir les dades i és al mateix servei on es tracten i es torna el resultat desitjat a l'activitat (figura 8.4).

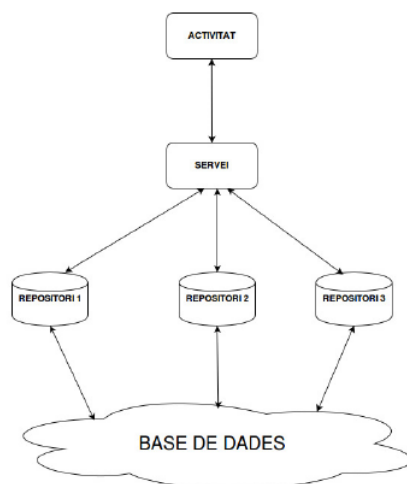


FIGURA 8.3: Accés a les dades.

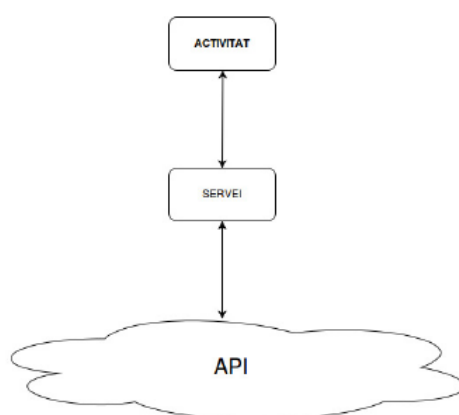


FIGURA 8.4: Accés a serveis externs.

8.1.1 Patrons de disseny

En aquest apartat es descriuen els patrons utilitzats en el disseny del software. Com a primer plantejament, es va tenir en compte el patró Model-Vista-Controlador però això provoca que s'hagi de trencar el gran acoblament que ofereix la plataforma *Android* entre el controlador i la vista. Per aquest motiu es va decidir decantar el disseny de l'arquitectura del projecte cap a l'estructura explicada en l'apartat anterior. Els patrons que es treballen en el projecte són els següents:

- **Patró factoria**

En el disseny de l'aplicació, s'utilitza el patró factoria per tal d'aconseguir els diferents tipus de servei. A més, això permet que en tot el sistema només existeixi una instància de cada servei i, per tant, a l'hora d'instanciar qualsevol servei s'ha de fer a través de la factoria.

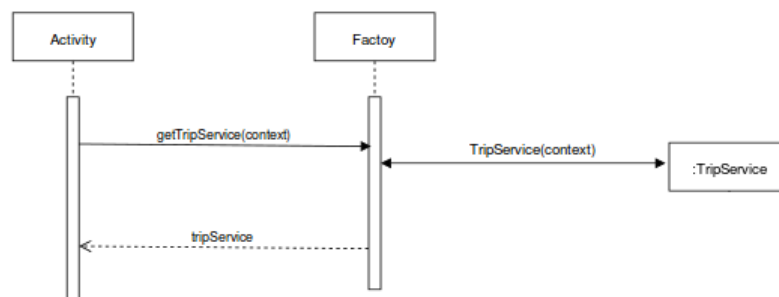


FIGURA 8.5: Patró factoria.

- **Patró adaptador**

Per tal de cridar als serveis externs s'utilitza el patró adaptador, que fa de pont entre el servei i el sistema. Des de l'Activity es crida a l'adaptador, i és l'adaptador qui es comunica amb el servei extern desitjat. D'aquesta manera aconseguim una capa d'abstracció que ens permetria canviar el servei extern i no haver de modificar res en l'Activity.

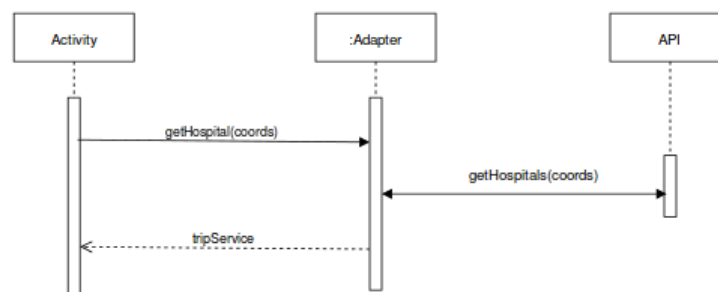


FIGURA 8.6: Patró adaptador.

- **Patró repositori**

S'utilitza per tal d'evitar l'accés directe a les dades i d'aquesta manera desacoblar al màxim l'accés a *Firebase*. Per qualsevol classe que resulti un objecte emmagatzemat a la base de dades es definirà una interfície *Repository* parametritzada, sobre la qual s'haurà de definir una classe *RepositoryXXXX* que la implementi.

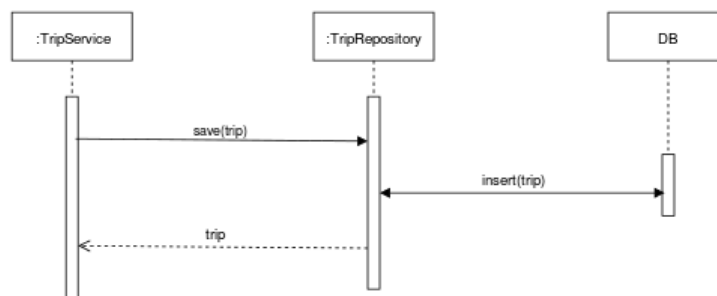


FIGURA 8.7: Patró repositori.

- **Patró servei**

S'utilitza per crear una capa d'abstracció sobre els repositoris prèviament mencionats, cosa que permet desenvolupar en aquestes noves classes la lògica de l'aplicació sense accedir directament als repositoris. Aquests serveis contenen una instància única del repositori.

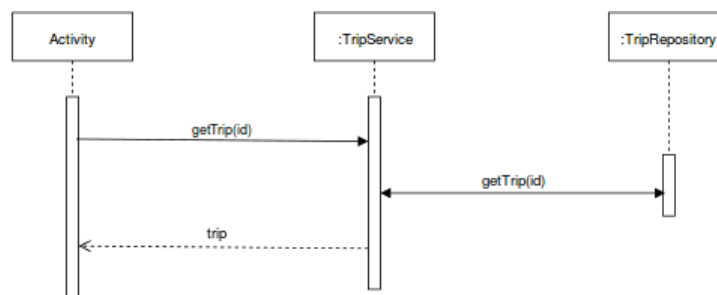


FIGURA 8.8: Patró servei.

8.1.2 Tecnologies utilitzades

Android

És el sistema operatiu per a dispositius mòbils amb pantalla tàctil més utilitzat. Actualment l'utilitzen des de telèfons intel·ligents i tables, fins a rellotges de nova generació, televisors i automòbils. És un projecte de codi obert recolzat per Google des del 2008. El desenvolupament d'aplicacions es fa habitualment en llenguatge Java i es disposa d'un SDK (Software Development Kit) que es pot descarregar gratuïtament des de la pàgina web oficial.

JSON

JavaScript Object Notation és el nom que rep una forma d'estructurar les dades per, posteriorment, ser enviades entre dispositius o per la xarxa. És l'alternativa a XML en termes de llenguatge marcat i s'acostuma a utilitzar en les aplicacions Android per la comunicació amb el servidor o serveis externs.

Firebase

Firebase ofereix una base de dades en temps real com a servei. Aquest servei ofereix als desenvolupadors una API que permet guardar dades al núvol de les seves aplicacions. La companyia ofereix al client diverses llibreries per tal de facilitar la integració amb Android, iOS, Java i d'altres aplicacions.

8.2 Capa de presentació

Iniciar sessió

El procés d'identificació de l'usuari es dur a terme a través de Google, per tant, l'usuari no es registra al nostre sistema sinó que utilitza les credencials del seu compte de Google per tal d'identificar-se.

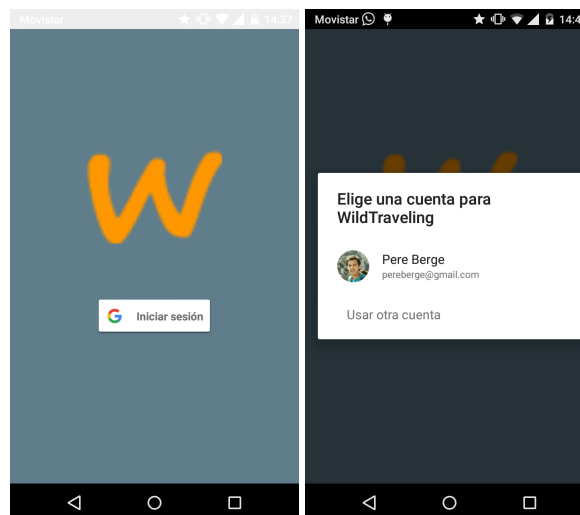


FIGURA 8.9: Inici de sessió

Consultar llista de viatges

A la pantalla de consultar viatges es mostra la llista de viatges de l'usuari. Si l'usuari selecciona qualsevol viatge l'aplicació el redigirà a la pantalla de consulta de viatge. També hi ha l'opció de crear un nou viatge. Des d'aquesta pantalla l'usuari pot tancar la sessió.

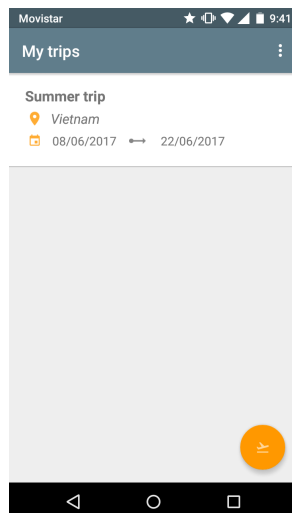


FIGURA 8.10: Inici de sessió

Crear viatge

En aquesta pantalla és on s'introdueixen totes les dades d'un nou viatge com el nom, el destí, les dates i els participants. A més, el sistema li pregunta sobre el seu estil de viatjar i demana que li introdueixi les dades d'un contacte d'emergència. Un cop creat el viatge, l'aplicació redirigeix l'usuari a la pantalla de consultar viatge.

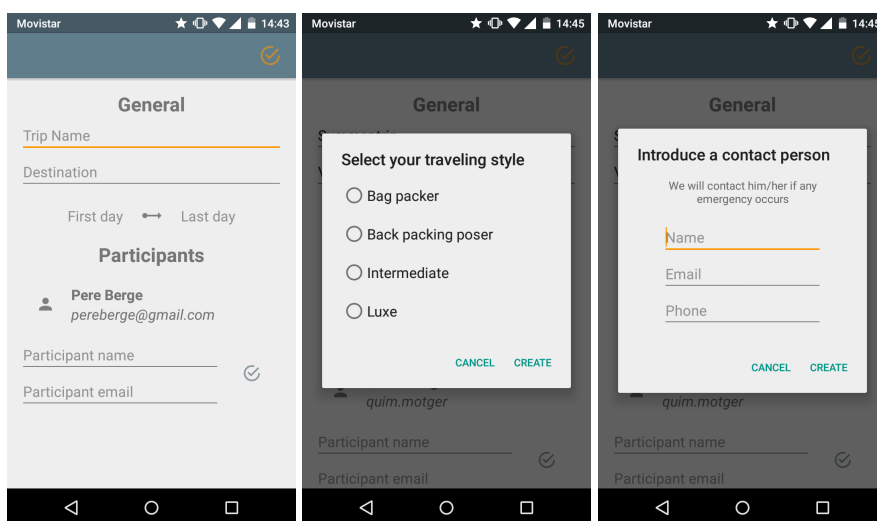


FIGURA 8.11: Crear viatge

Consultar viatge

La pantalla on es mostra tota la informació del viatge: nom, destí, dates i participants. A més, és la pantalla des d'on es pot accedir a totes les funcionalitats nombrades en apartats anteriors i permet eliminar el viatge.

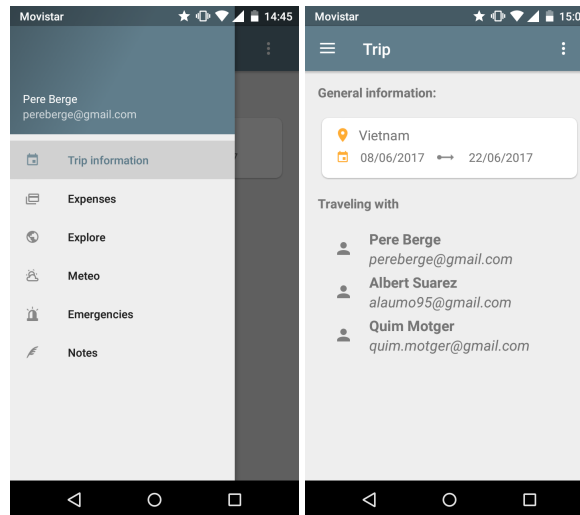


FIGURA 8.12: Consultar viatge

Consultar llista de despeses

La pantalla de consultar despeses és on es poden veure totes les despeses del viatge i la quantitat total gastada. Si l'usuari selecciona una despesa el sistema el redirigeix a la pantalla de consultar despesa. A més, l'usuari també té l'opció de crear-ne una de nova, fet que el portarà a la pantalla de crear despesa.

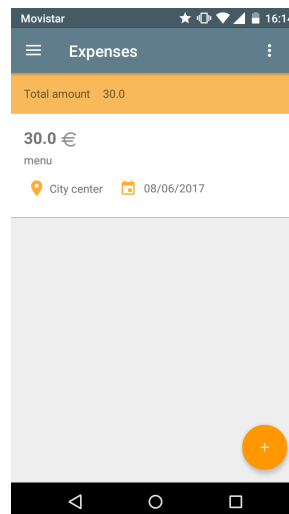


FIGURA 8.13: Llista despeses

Crear despesa

En la pantalla crear despesa l'usuari introdueix les dades de la nova despesa com el motiu, la quantitat a pagar, la localització i si ha pagat una part a un company de viatge. Un cop creada la despesa el sistema redirigeix l'usuari a la pantalla de consultar despesa.

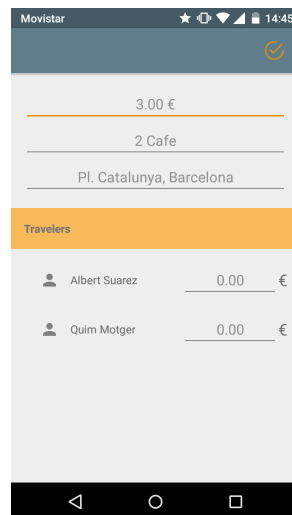


FIGURA 8.14: Crear despesa

Consultar despesa

En la pantalla de consultar despesa l'usuari pot veure tota la informació de les despeses i té l'opció de marcar un deute com a pagat, modificar la despesa o eliminar-la. En el cas que l'usuari seleccioni modificar despesa el sistema el redirigeix a la pantalla de modificació de despesa, per l'altra banda, si decideix eliminar la despesa, el sistema el redirigeix a la pantalla de consultar llista de despeses.

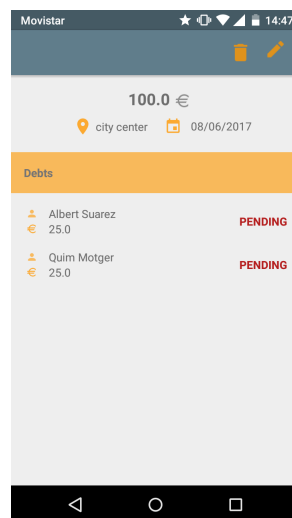


FIGURA 8.15: Consultar despesa

Modificar despesa

En la pantalla de modificar despesa l'usuari pot modificar totes les dades d'una despesa. Quan confirma la modificació el sistema el redirigeix la pantalla de consultar despesa.

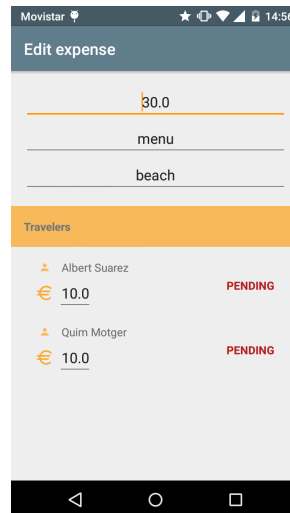


FIGURA 8.16: Modificar despesa

Cerca llocs d'interés

En la pantalla de cerca lloc d'interés l'usuari pot seleccionar què vol buscar d'entre les possibilitats proposades i a quin lloc ho vol buscar. Si el lloc on ho vol buscar queda buit, es cercaran els resultats tenint en compte la localització de l'usuari en el moment de la cerca. Si no es troben resultats o es necessita activar el GPS s'informarà a l'usuari, d'altra banda, si la cerca es dur a terme satisfactoriament el sistema redirigirà l'usuari a la pantalla de llista de resultats de cerca.

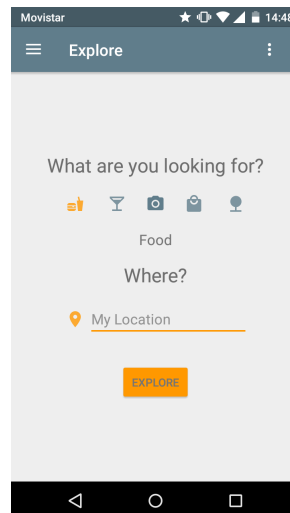


FIGURA 8.17: Cerca de llocs d'interès

Consultar llista de resultats de cerca

És la pantalla on es veuen les resultats de la cerca, aquests estan ordenats per l'estil de viatjar de l'usuari. Per a cada item trobat es pot consultar el nom, la localització i, en cas que n'hi hagi, el nivell de preus de l'establiment. Quan l'usuari selecciona un resultat de la cerca el sistema el redirigeix a la pantalla de consultar ruta.

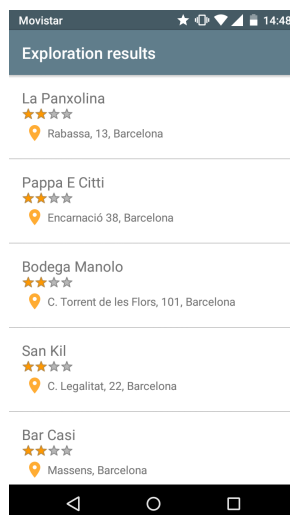


FIGURA 8.18: Resultats de cerca

Consultar ruta

En la pantalla de consultar ruta, l'usuari pot veure diverses opcions de ruta des d'on s'ha fet la cerca fins al punt seleccionat. A més, pot consultar la ruta en cotxe, transport públic i a peu, per a cada modalitat es poden veure paràmetres com la distància en quilòmetres i el temps que tardarà en arribar-hi. Si l'usuari prem sobre una de les rutes, l'aplicació li dona l'opció de consultar la ruta exacta utilitzant, en el cas que l'usuari la tingui instal·lada, l'aplicació Google Maps. Per últim, l'usuari té l'opció de consultar la ruta inversa.

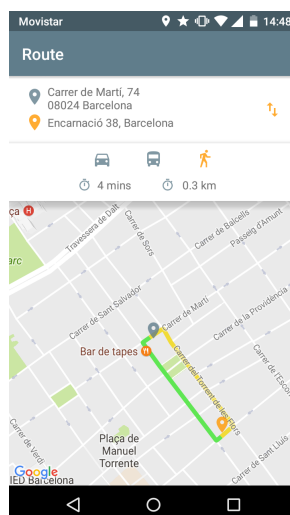


FIGURA 8.19: Consultar ruta

Consultar previsió metereològica

En la pantalla de consultar previsió metereològica es pot veure la meteorologia del dia actual amb paràmetre com la temperatura màxima i mínima, el percentatge de possibilitat de pluja, la velocitat del vent o la descripció del cel. A més, l'usuari també pot consultar la previsió fins a sis dies vista.



FIGURA 8.20: Previsió metereològica

Emergències

En la pantalla d'emergències l'usuari té tres opcions. La primera és realitzar una trucada al telèfon d'emergències del país on estigui, si selecciona aquesta opció el sistema el redirigirà a la pantalla de trucades del seu dispositiu mòbil. La segona opció es tracta de realitzar una trucada al seu propi contacte d'emergències, en aquest cas també se'l redirigirà a la pantalla de trucades del seu dispositiu mòbil. Per últim, té l'opció de seleccionar l'opció d'obtenir la ruta a l'hospital més proper, en aquest cas se'l redirigirà a la pantalla de consultar ruta.

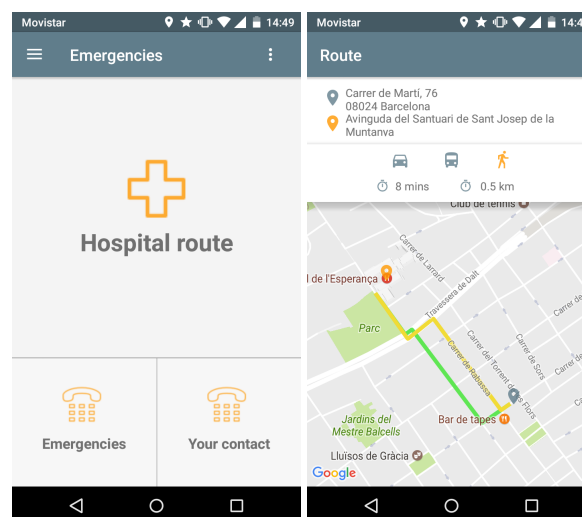


FIGURA 8.21: Gestió d'emergències

Consultar llista de notes

En aquesta pantalla l'usuari pot consultar la llista de notes que té el viatge. Des d'aquesta pantalla pot accedir a consultar una nota en concret o crear-ne una de nova.

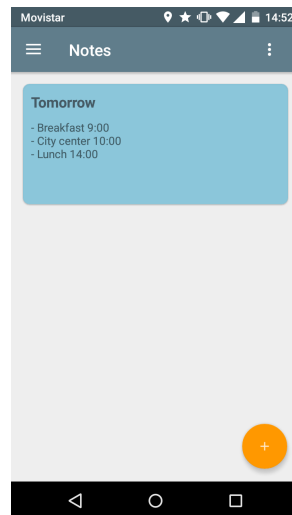


FIGURA 8.22: Llista de notes

Crear nota

En la pantalla de crear nota, l'usuari pot introduir el títol i el text de la nota i pot seleccionar un color per tal de canviar el fons d'aquesta. També pot seleccionar els participants del viatge amb qui vol compartir la nota a través de correu electrònic.

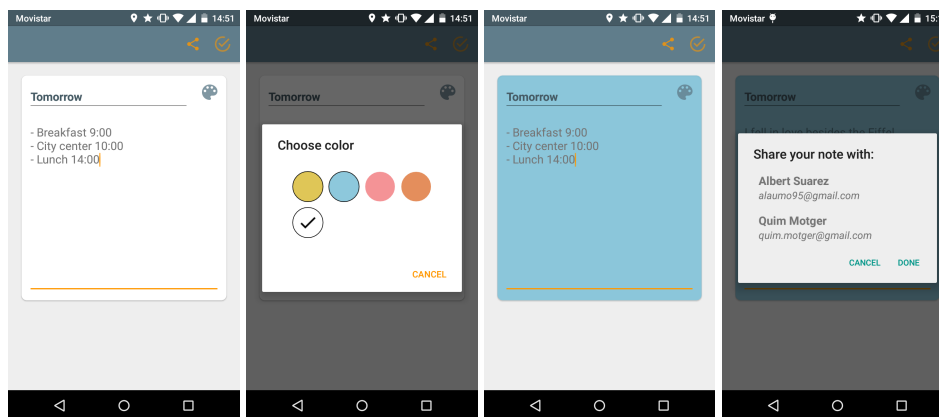


FIGURA 8.23: Crear nota

Consultar nota

En la pantalla de consultar nota l'usuari pot veure la nota seleccionada i té informació de amb qui l'ha compartit. A més, sempre pot canviar el color del fons de la nota.

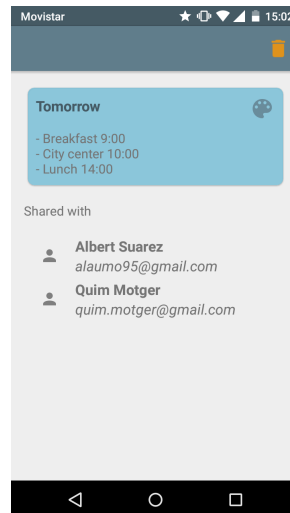


FIGURA 8.24: Consultar nota

8.3 Diagrama de classes de disseny

8.4 Model de dades

En aquest apartat s'explica el model de dades utilitzat per desenvolupar el projecte. La base de dades que s'ha utilitzat és un base de dades no relacional de tipus *key-value*. Per tal de modelar les dades, s'han seguit les bones pràctiques de la tecnologia utilitzada *Firebase*.

Les dades del projecte es guarden en una direcció concreta que proporciona *Firebase*. En aquesta direcció existeix un node pare del projecte des d'on pegen totes les dades. Des d'aquest node pare pegen llistes clau-valor, per cada tipus de dades diferent, amb totes les dades emmagatzemades d'aquell tipus concret (figura 8.25). El fet que es guardin les dades per tipus de d'entitat és degut a que els manuals de *Firebase* defineixen que per tal d'obtenir un bon resultat és el camí que s'ha de seguir.



FIGURA 8.25: Estructura base de dades.

En la llista d'una entitat en concret, per a cada clau tenim un valor, i aquest valor conté tots els atributs de l'entitat guardada. En tenir llistes separades per cada entitat, quan es necessita referenciar un altre objecte guardat en la base de dades, s'utilitza la clau. Per tant, un deute no s'emmagatzema dins la despesa a la qual pertany sinó que es guarda en la llista de deutes i es guarda un atribut amb la clau de la despesa (figura 8.26).

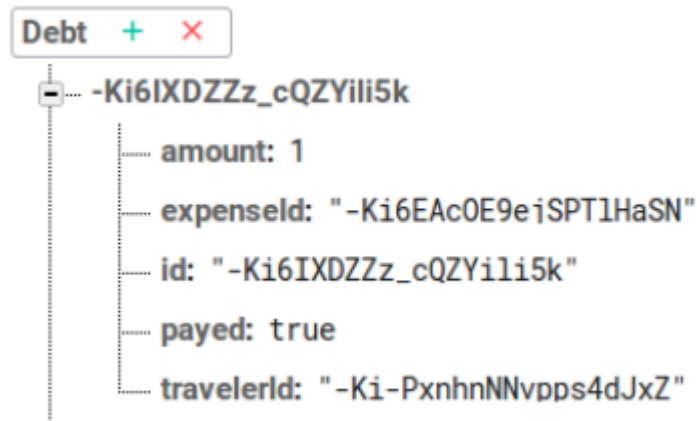


FIGURA 8.26: Solució per no duplicar informació.

8.5 Diagrames de seqüència

Capítol 9

Implementació

En aquest apartat s'explica de quina manera s'han utilitzat les tecnologies seleccionades per al desenvolupament.

- **Inici de sessió**

Per a l'inici de sessió s'utilitza el servei que proporciona *Google*. L'aplicació mostra a l'usuari el botó d'inici de sessió d'aquesta plataforma, quan es prem aquest botó es desvia l'inici de sessió a *Google* (figura 9.1) on l'usuari introdueix les dades del seu compte.

```
private void signIn() { //Inici de sessió amb Google
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent, GOOGLE);
}
```

FIGURA 9.1: Crida d'inici de sessió

Un cop l'usuari ha introduït les seves dades a *Google* es retorna el control al sistema. Quan l'aplicació continua amb la seva execució ho fa a partir de la funció que es pot veure en la figura 9.2. L'últim pas per a l'inici de sessió es obtenir les dades de l'usuari i iniciar l'aplicació.

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == GOOGLE) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        handleSignInResult(result);
    }
}
```

FIGURA 9.2: Retorn d'inici de sessió

- **Accés a la base de dades**

La connexió amb la base de dades es dur a terme des de la classe *FirebaseRepository*, aquesta classe és la classe genèrica que utilitzant tots els serveis per tal d'accedir a la base de dades. En la figura 9.3 es pot veure com s'inicialitzen els camps del repositori.

```
protected final String FIREBASE_URI = "https://wildtraveling-9e8a1.firebaseio.com/";

public FirebaseRepository(Context context) {
    firebase = FirebaseDatabase.getInstance(FIREBASE_URI);
    map = new LinkedHashMap<>();
    database = firebase.getReference().child(getObjectReference());
}
```

FIGURA 9.3: Constructor del repositori

Dins d'aquesta classe es defineixen totes les operacions que es poden fer sobre la base de dades. En la figura 9.4 es pot veure com es declara un *insert*.

```
public T insert(T item) {
    DatabaseReference ref = database.push();
    ref.setValue(item);
    item.setId(ref.getKey());
    map.put(ref.getKey(), item);
    return item;
}
```

FIGURA 9.4: Funció d'insert genèrica

Per tal de llegir dades de firebase, el que es fa és el següent. Quan s'inicia l'aplicació es carreguen totes les dades de l'usuari en els repositoris. A més, per tal de tenir sempre els repositoris actualitzats, s'implementen *listeners*. Quan es carreguen les dades provinents de *Firebase* a les estructures del sistema es necessita convertir les dades del núvol en objectes del sistema. Per aquest motiu, s'utilitza la funció *convert* que es pot veure en la figura 9.5.

```
protected Note convert(DataSnapshot data) {
    if (data == null) return null;

    Note note = new Note();
    note.setId(data.getKey());

    for (DataSnapshot d : data.getChildren()) {
        if ("title".equals(d.getKey())) {
            note.setTitle(d.getValue(String.class));
        } else if ("text".equals(d.getKey())) {
            note.setText(d.getValue(String.class));
        } else if ("color".equals(d.getKey())) {
            note.setColor(d.getValue(Integer.class));
        } else if (d.getKey().equals("travelerTags")) {
            List<String> tags = new ArrayList<>();
            for (DataSnapshot debt : d.getChildren()) {
                tags.add(debt.getValue(String.class));
            }
            note.setTravelerTags(tags);
        } else if ("tripId".equals(d.getKey())) {
            note.setTripId(d.getValue(String.class));
        }
    }
    return note;
}
```

FIGURA 9.5: Conversió de les dades a l'entitat Nota

- **Accés a les dades des de l'Activity**

Per interaccionar amb les dades i poder mostrar-les a l'usuari s'implementen diversos serveis. Aquests serveis són els encarregats de fer les operacions necessàries per tal de proporcionar a l'Activity les dades que vol mostrar a l'usuari o per tal d'emmagatzemar les dades que l'usuari introdueix.

Es declara un servei per cada funcionalitat: un servei pels diferents tipus de persones, un per als viatges, un per despeses i deutes, i un altre per a les notes. En cada servei tenim una instància de repositori del tipus que estem tractant, per exemple, en el servei que tracta les despeses tenim una instància del repositori de despeses i una instància del repositori de deutes. En el servei es defineixen operacions com la de la figura 9.6.

```
public List<Expense> getExpensesByTripId(String currentTrip) {  
    List<Expense> expenses = new ArrayList<>();  
    for(Expense e: repository.all()){  
        if(e.getTripId().equals(currentTrip)){  
            expenses.add(e);  
        }  
    }  
    return expenses;  
}
```

FIGURA 9.6: Mètode de la classe ExpenseService

Pel que fa a l'Activity, es declara el servei com en la figura 9.7 i s'utilitza aquest servei per tal d'obtenir qualsevol dada i guardar les dades introduïdes per l'usuari.

```
String expenseId = intent.getStringExtra("expenseId");  
  
expenseService = ServiceFactory.getExpenseService(this);  
  
expense = expenseService.getExpenseById(expenseId);  
debts = expenseService.getDebtsByExpense(expense.getId());
```

FIGURA 9.7: Mètode de la classe ExpenseService

- **Connexió amb serveis externs**

En la connexió amb els serveis externs que s'utilitzen s'utilitza la mateixa arquitectura que en l'accés a les dades. Des del punt de vista de l'Activity, el procediment és exactament el mateix. S'instància el servei que es necessita obtenint-lo de la factoria de serveis i s'utilitza per tal d'obtenir les dades que es necessiten.

La diferencia es veu en la classe servei, ja que enlloc de comunicar-se amb la base de dades es comunica amb un servei extern. En la figura 9.8 i 9.9 es pot veure un exemple de crida al servei extern que proporciona la llista dels hospitals que hi ha al voltant de l'usuari i posteriorment es selecciona el més proper a l'usuari. El resultat de la crida als diferents serveis sempre és un JSON el

qual es deserialitza per tal d'obtenir objectes útils per al sistema.

```
public FoursquareVenue getHospital(LatLng coord) {
    this.latitude = String.valueOf(coord.latitude);
    this.longitude = String.valueOf(coord.longitude);
    venueList = new ArrayList<>();
    Util.setFinishSearch(false);
    new HospitalCall().execute();
    return getBestHospital(venueList);
}
```

FIGURA 9.8: Mètode per obtenir l'hospital més proper

```
private class HospitalCall extends AsyncTask<View, Void, String> {
    @Override
    protected String doInBackground(View... urls) {
        // make Call to the url
        temp = makeCall("https://api.foursquare.com/v2/venues/search?" +
            "&client_id=" + CLIENT_ID +
            "&client_secret=" + CLIENT_SECRET +
            "&ll=" + latitude + "," + longitude +
            "&v=" + v +
            "&query=hospital" +
            "&oauth_token=DBRXKVUS1HG52CRZK5ERIKQ42BP4UPL11Q2N0PQUCCZZ3A4R");
        return temp;
    }
    @Override
    protected void onPreExecute() {}

    @Override
    protected void onPostExecute(String result) {
        if (temp == null) {
            finish = true;
        } else {
            venueList = parseFoursquareSearch(temp);
            Util.setVenues(venueList);
            Util.setFinishSearch(true);
            finish = true;
        }
    }
}
```

FIGURA 9.9: Crida al servei de punts d'interés

- **Enviar correu electrònic**

Per tal de poder enviar un correu electrònic directament des del codi *Android* s'utilitza una classe anomenada *MailSender*. En aquesta classe es defineixen tots els paràmetres que permetran enviar un correu electrònic. En la figura 9.10 es pot veure el fragment de codi on es defineixen els paràmetres que permeten enviar des de *Gmail*, s'inicia la sessió amb les credencials del compte *WildTraveling* i es crea i envia el missatge en si.

```
private void sendEmail() {  
    for(NoUser u: triptravelers) {  
        MailSender sm = new MailSender(this, u.getEmail(),  
            "WildTraveling note from "+ currentUser.getName(),  
            title.getText().toString()+"\n\n"+text.getText().toString());  
        sm.execute();  
    }  
}
```

FIGURA 9.10: Enviament d'un correu electrònic

Des de l'*Activity*, per tal d'enviar un email, es crea una instància de la classe *MailSender* amb les dades del receptor del correu, l'assumpte i el cos, i s'executa el procés que s'ha definit dins la classe (Figura 9.11).

```
protected Void doInBackground(Void... params) {  
    //Creating properties  
    Properties props = new Properties();  
    props.put("mail.smtp.host", "smtp.gmail.com");  
    props.put("mail.smtp.socketFactory.port", "465");  
    props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");  
    props.put("mail.smtp.auth", "true");  
    props.put("mail.smtp.port", "465");  
  
    session = Session.getDefaultInstance(props,  
        new javax.mail.Authenticator() {  
            protected PasswordAuthentication getPasswordAuthentication() {  
                return new PasswordAuthentication(Util.EMAIL, Util.PASSWORD);  
            }  
        });  
  
    try {  
        MimeMessage mm = new MimeMessage(session);  
        mm.setFrom(new InternetAddress(Util.EMAIL));  
        mm.addRecipient(Message.RecipientType.TO, new InternetAddress(email));  
        mm.setSubject(subject);  
        mm.setText(message);  
  
        Transport.send(mm);  
    } catch (MessagingException e) {  
        e.printStackTrace();  
    }  
    return null;  
}
```

FIGURA 9.11: Funció enviar correu electrònic

Capítol 10

Proves

Capítol 11

Informe de sostenibilitat

Per tal d'evaluar la sostenibilitat del projecte a nivell econòmic, social i ambiental es fabrica la matriu de sostenibilitat, on s'atribueix a cada dimensió un valor evaluant així cada aspecte.

Caràcter	Valoració
Econòmic	8
Social	8
Ambiental	7
Total	23

TAULA 11.1: Matriu de sostenibilitat

11.1 Dimensió econòmica

Per aquest projecte s'han avaluat detalladament els costos tant de recursos com materials. No s'han tingut en compte els costos deguts a possibles ajustaments, reparacions o actualitzacions durant el desenvolupament ja que es tracta d'un projecte acadèmic. Sí que es cert que es podria desenvolupar el mateix projecte en menys temps, però per fer-ho possible es necessitaria més recursos humans i d'una major experiència professional. Pel que fa a la planificació es pot assegurar que les tasques en les que s'ha dedicat més hores són les més importants. Podem dir que el cost total del projecte ha estat ajustat al màxim tenint en compte els costos de projectes de desenvolupament de software similars, i això converteix el sistema en un producte competitiu.

11.2 Dimensió social

El producte resultant d'aquest projecte facilitarà molt la vida a tots els viatgers que vulguin un suport per tal de millorar les condicions del viatge. La necessitat del productes és real ja que sense ell el viatger es veu més confús a l'hora d'obtenir informació sobre l'entorn degut a que les aplicacions existents són molt específiques i no s'adapten a la necessitat de l'usuari.

11.3 Dimensió ambiental

El desenvolupament del projecte provoca la utilització de diversos recursos que afecte el medi ambient. En les diferents parts del projecte s'utilitza l'energia elèctrica com a font d'energia per poder treballar, el cost d'aquesta té un impacte ambiental. En concret s'utilitzarà l'energia elèctrica necessària per alimentar l'ordinador que s'utilitza per a desenvolupar i el dispositiu mòbil on es fan les proves. A més, s'ha de tenir en compte també que per utilitzar l'aplicació és necessari tenir connexió a internet, cosa que també suposa un cost elèctric.

Amb aquest nou producte, podem dir que es minimitzaran les descàrregues de múltiples aplicacions, cosa que disminueix el cost elèctric. A més, també promou l'estalvi de paper en termes de guies de turisme i altres.

Capítol 12

Conclusions

12.1 Justificació del desenvolupament de les competències tècniques

- **CES1.1:** Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [En profunditat]
S'ha desenvolupat un sistema software de certa complexitat ja que incorpora funcionalitats molt diverses i no trivials.
- **CES1.2:** Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. [Bastant]
L'aplicació necessita de la participació d'aplicacions externes per tal d'aconseguir certes funcionalitats, per tant, s'ha fet un treball d'integració de serveis externs. A través de l'ús de patrons, s'ha aconseguit obtenir un sistema molt flexible a l'hora d'adjuntar nous serveis externs o eliminar els utilitzats.
- **CES1.3:** Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. [Bastant]
S'han identificat els principals riscos a l'hora de desenvolupar el software com, per exemple, la necessitat de connexió a internet. Un cop identificats s'han evaluat i s'han gestionat de la millor manera possible aconseguint així un bon funcionament del software i una reducció de les possibilitats de fallada.
- **CES1.5:** Especificar, dissenyar, implementar i avaluar bases de dades. [En profunditat]
S'ha dut a terme un anàlisi profund estudiant quin seria el millor disseny per a la base de dades. S'ha obtingut per utilitzar una tecnologia que no es tracta durant els cursos del grau i, per tant, s'ha hagut de fer un esforç en l'aprenentatge de l'ús correcte d'aquesta.
- **CES1.6:** Administrar bases de dades (CIS4.3) [Una mica]
L'administració de la base de dades té lloc un cop l'aplicació està posada en marxa, per aquest motiu la competència no incideix molt en el treball final de grau. Tot i així, si que es fa un treball sobre la base de dades durant la fase de proves.
- **CES1.7:** Controlar la qualitat i dissenyar proves en la producció de software. [Bastant]
S'han dut a terme proves durant el desenvolupament del software. Les proves han estat fetes pel desenvolupador i per usuaris externs al projecte que han provat l'aplicació.

- **CES1.9:** Demostrar comprensió en la gestió i govern dels sistemes software. [En profunditat]
Desenvolupar una aplicació com aquesta requereix d'una bona gestió del sistema software i així ha estat. S'ha gestionat el sistema de manera correcta analitzant varies alternatives i seleccionant la més adient.
- **CES2.1:** Definir i gestionar els requisits d'un sistema software. [En profunditat]
En la fase inicial, es van definir tots els requisits de l'aplicació. Aquest punt del projecte és bàsic per al correcte desenvolupament de l'aplicació ja que es necessita saber què es vol fer, per aquest motiu és una competència que es treballa en profunditat.
- **CES2.2:** Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integrin aspectes ètics, socials, legals i econòmics. [En profunditat]
El desenvolupament de l'aplicació té un impacte en la societat, a més, s'han identificat les lleis i restriccions que ha de complir el projecte. També s'ha evaluat la sostenibilitat del projecte en termes socials, econòmics i medioambientals.

12.2 Desviacions en la planificació i el pressupost

En aquest apartat s'exposen les diferències que hi ha hagut entre la planificació inicial i la realitat en el desenvolupament del projecte i les conseqüències d'aquestes desviacions sobre el pressupost.

Fase inicial:

La fase inicial ha estat més llarga del que s'esperava ja que es va haver de dedicar més hores de les previstes en la definició del projecte i l'especificació de requisits. Això va comportar un retard de 7 dies.

Iteracions:

Degut al retard en la fase inicial, les iteracions es van començar una setmana després de l'esperat. Aquest fet va forçar una reorganització de la fase de desenvolupament del producte. S'han convertit les 5 iteracions planificades en 7, fent-les més curtes. A més, durant aquest període s'ha hagut de posar més hores de feina de les previstes degut a que es va planificar menys feina de la que hi havia.

- **Iteració 1:** 4 d'abril - 18 d'abril (planificació: 29 de març - 12 d'abril)
La primera iteració ha estat de dues setmanes ja que és un punt del projecte on s'ha de posar en marxa molts aspectes.
- **Iteració 2:** 19 d'abril - 25 d'abril (planificació: 13 d'abril - 26 d'abril)
A partir d'aquesta iteració, ja es va decidir posar deadlines més propers ja que el temps disponible ho forçava i la quantitat de feina a fer ho permetia.
- **Iteració 3:** 26 d'abril - 2 de maig (planificació: 27 d'abril - 10 de maig)
- **Iteració 4:** 3 de maig - 11 de maig (planificació : 11 de maig - 24 de maig)
- **Iteració 5:** 11 de maig - 18 de maig (planificació: 25 de maig - 7 de juny)

- **Iteració 6:** 19 de maig - 25 de maig (iteració extra) **Iteració 7:** 26 de maig - 7 de juny (iteració extra)

En aquesta iteració es té previst millorar les interfícies i acabar alguns detalls pendents. Un cop finalitzada aquesta iteració el producte estarà llest per ser llençat al mercat.

Fase final

La fase final no es veurà afectada pels canvis en la planificació que hi ha hagut en les fases anteriors.

Tenint en compte les desviacions mencionades, les hores emprades en el projecte queden repartides de la següent manera.

Fase	Duració	%
Fase inicial	119h	36%
Iteracions	169h	51%
Fase final	45h	13%
Total	333h	100%

TAULA 12.1: Distribució d'hores reals i percentatge de treball

Per tant, hi ha una desviació de 76 hores que significa una diferència del 22%. Aquest valor és força gran i es pot considerar que la planificació no va ser del tot bona degut a la poca experiència en aquest camp.

Tal i com es va especificar a l'apartat de control de gestió, un cop acabat el projecte es contabilitzen les hores de treball que cada rol ha dedicat i es comparen amb les que s'havia planificat. El resultat obtingut és el que es mostra en la taula 12.2.

El cost real del projecte, per tant, disminueix fins a 10.810€, un cost notablement inferior al planificat. Això és degut a que s'ha treballat menys hores de les esperades i, els rols que han superat les hores planificades són rols de més baixa retribució com el programador.

Rol	Hores estimades	Cost estimat	Hores reals	Cost real
Cap de projecte	160h	7.200€	126h	5.670€
Analista	45h	1.350€	30h	900€
Dissenyador	70h	2.100€	23h	690€
Programador	104h	2.600€	124h	3.100€
Tester	30h	450€	30h	450€

TAULA 12.2: Comparativa entre les hores estimades i les reals treballades, i el seu cost associat

En quant a les previsions de les despeses generals han seguit els valors que es van preveure. Si restem els costos dels imprevistos i el valor de les contingències, el valor real del projecte és 11.000€.

12.3 Gantt final**12.4 Limitacions i dificultats****12.5 Treball futur**