



Universitat
de les Illes Balears

TREBALL DE FI DE GRAU

CREACIÓ D'UNA WEB APP SOCIAL DEDICADA A LA PUBLICACIÓ D'EXCURSIONS.

Martí Paredes Salom

Grau d'Enginyeria Informàtica

Escola Politècnica Superior

Any acadèmic 2023-24

CREACIÓ D'UNA WEB APP SOCIAL DEDICADA A LA PUBLICACIÓ D'EXCURSIONS.

Martí Paredes Salom

Treball de Fi de Grau

Facultat de: Escola Politècnica Superior

Universitat de les Illes Balears

Any acadèmic 2023-24

Paraules clau del treball: Web App, Full-Stack, Excursions, Adaptatiu, Xarxa Social.

Nom del tutor / la tutora del treball: Antonio Bibiloni Coll

Nom del tutor / la tutora (si escau): Antoni Oliver Tomàs

Autoritz la Universitat a incloure aquest treball en el repositori institucional per consultar-lo en accés obert i difondre'l en línia, amb finalitats exclusivament acadèmiques i d'investigació

Autor/a		Tutor/a	
Sí	No	Sí	No
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Gràcies a totes les persones que d'alguna manera han contribuït a fer que aquest projecte sortís endavant, a tothom que m'ha donat una mà quan ho necessitava i sobretot a la meva família i amics pel suport constant que m'han donat durant aquests quatre anys de carrera.

Índex:

0. Resum.....	7
1. Introducció.....	9
1.1 Contextualització.....	9
1.2 Objectius.....	10
1.2.1 Objectius principals.....	10
1.2.2 Objectius secundaris.....	11
1.3 Elements diferenciadors.....	12
1.4 Abast del projecte.....	14
1.5 Estructura del document.....	15
2. Requisits del projecte.....	17
2.1. Requisits funcionals.....	17
2.3. Requisits no funcionals.....	22
3. Arquitectura proposta.....	25
3.1 Arquitectura front-end.....	25
3.2 Arquitectura back-end.....	26
3.3 Model base de dades.....	27
3.4 Desplegament del projecte.....	29
3.5 Visió general de l'arquitectura.....	30
4. Desenvolupament del projecte.....	31
4.1 Planificació temporal.....	31
4.2 Desenvolupament dels requisits funcionals.....	32
4.2.1: Creació d'un compte d'usuari.....	32
4.2.2: Inici de sessió d'un usuari existent.....	33
4.2.3. Visualització i modificació d'un perfil d'usuari.....	34
4.2.4: Cerca i seguiment d'altres usuaris.....	36
4.2.5: Xatejar amb altres usuaris en temps real.....	36
4.5.6 Penjar una excursió.....	37
4.5.7: Editar una excursió ja penjada.....	41
4.5.8: Indicar que m'agrada i comentar una publicació d'altre usuari.....	43
4.2.9: Tenir una feed adaptada als meus gustos.....	44
4.2.10: Penjar una excursió des d'un fitxer GPX.....	48
4.2.11: Enllaçar imatges a les publicacions.....	49
4.2.12: Canviar la contrasenya.....	51
4.2.13: Canviar l'idioma de la webapp.....	54
5. Resultats.....	56
5.1 Resultats obtinguts.....	56
5.2 Proves d'usuari.....	58
6. Conclusions.....	62
6.1 Competències i coneixements adquirits.....	62

6.2 Limitacions.....	64
6.3 Recomanacions.....	64
6.4 Feina a futur.....	65
7. Bibliografia.....	66

0. Resum

En aquest treball de fi de grau, es presenta el desenvolupament d'una web app social orientada a l'excursionisme, utilitzant tecnologies modernes i aplicant els coneixements adquirits durant el grau.

L'objectiu ha estat crear una plataforma que permeti als usuaris compartir rutes, consells i experiències relacionats amb l'excursionisme, fomentant així una comunitat cohesionada i apassionada per la natura.

El projecte aborda tant la part de back-end, amb la gestió de dades i la lògica de l'aplicació, com el front-end, amb el disseny web dinàmic i adaptatiu. Es descriu el procés de desenvolupament, des de la definició dels objectius fins a la implementació i les proves amb usuaris, assegurant una experiència d'usuari satisfactòria i una estructura de codi neta i mantenible.

1. Introducció

En aquest primer apartat s'explica la base d'aquest TFG, amb una breu contextualització del que es vol fer, perquè volem fer aquest projecte i com es diferencia d'algunes solucions ja existents.

1.1 Contextualització

L'excursionisme, com a activitat recreativa, està experimentant un creixement exponencial a causa del gran nombre que està iniciant-se a aquest esport per la presa de consciència sobre la salut física i mental, així com a la recerca de connexió amb la natura en una era cada vegada més urbanitzada, aquest creixement s'ha vist impulsat pel desenvolupament accelerat de les noves tecnologies les quals han jugat un paper molt important, sobretot en les noves generacions. A mesura que avança la tecnologia, també ho fan els dispositius intel·ligents i les aplicacions mòbils que proporcionen mapes detallats, seguiment GPS, prediccions meteorològiques en temps real entre altres, també s'ha detectat un augment en les plataformes digitals dissenyades per millorar l'experiència dels excursionistes i facilitar-los la interacció amb altres aficionats a aquesta pràctica.

Aquest treball de final de grau (TFG) té com a objectiu la creació d'una plataforma de xarxes socials dedicada específicament a l'excursionisme, no obstant es planteja una base genèrica el qual permet una adaptabilitat a altres temes relacionats. En aquesta plataforma, els usuaris tindran l'oportunitat de compartir les seves excursions, intercanviar coneixements i interactuar amb altres membres de la comunitat excursionista a través de publicacions, comentaris i missatgeria interna de l'aplicació. Els usuaris podran documentar les seves aventures, oferir consells, resoldre dubtes i inspirar-se en les experiències dels altres. A més, la capacitat de cercar i seguir perfils d'altres usuaris amb interessos similars fomentarà una comunitat activa i interconnectada. La missió principal d'aquesta plataforma és proporcionar un espai virtual enriquidor on els aficionats a l'excursionisme puguin connectar-se, inspirar-se i aprendre els uns dels altres,

millorant així la seva experiència global i contribuint a la popularització i pràctica segura de l'excursionisme.

Personalment, l'excursionisme és un esport que m'ha agradat des de sempre. És una manera de fer esport a l'aire lliure, allunyat dels espais tancats on ja passem gran part del nostre dia. Aquesta activitat permet conèixer gent que comparteix una sèrie de valors i la mateixa passió per la natura. He observat que la majoria de pàgines web o aplicacions disponibles al mercat se centren principalment a compartir informació o demostrar el progrés personal. No obstant això, trobava a faltar una plataforma que fomentés una competició sana i, al mateix temps, brindés als usuaris un lloc on poder crear xarxes i connectar amb altres persones que comparteixen la seva mateixa afició.

1.2 Objectius

En aquest apartat és descriuran els objectius del projecte, servint com a guia tant per al desenvolupament com per a poder mesurar quin nivell d'èxit s'aconseguirà al final d'aquest TFG.

Els objectius es dividiran en principals i secundaris, establint les metes que es volen assolir amb la creació d'aquesta web-app. A més d'establir les bases i les eines necessàries per poder crear una comunitat cohesionada i interessada en el món de l'excursionisme.

1.2.1 Objectius principals

L'objectiu principal d'aquest treball de fi de carrera és desenvolupar una aplicació full-stack emprant llibreries modernes com poden ser Vue, Express o Tailwind, i aplicant els coneixements adquirits durant els quatre anys de grau, i posteriorment consolidats i polits a la feina, dissenyant una arquitectura completa i òptima com sigui possible intentant obtenir un equilibri entre funcionalitat, rendiment i usabilitat.

Pel que respecte al front-end l'objectiu és crear una interfície que sigui simple, però moderna la qual ofereixi una experiència d'usuari completa. Aquesta interfície ha de tenir un disseny intuïtiu i agradable, emprant llibreries de components per a poder brindar una experiència molt més completa en menys temps de desenvolupament a més de reduir la quantitat de codi repetit a les diferents seccions, adaptant el disseny de les pàgines amb un enfocament “mobile first” per a poder garantir una experiència fluida i coherent entre diferents dispositius, assegurant que qualsevol usuari final no noti cap distinció en l'aplicació a causa del seu dispositiu, assegurant la usabilitat i l'accessibilitat.

Al mateix temps, es pretén construir un back-end basat en un model MVC [1] (Model-View-Controller) el qual ens permet dividir i delegar les diferents responsabilitats del codi, reduint el nombre de possibles errors que puguin aparèixer durant el desenvolupament. Aquest enfocament permet escriure un codi estructurat, net i fàcil de mantenir, facilitant la gestió de les dades de la base de dades i la comunicació eficient entre la base de dades i el front-end.

A més, es prestarà especial atenció a la seguretat i la protecció de dades, implementant mesures com la validació d'entrades, l'encriptació de dades confidencials i l'autenticació d'usuaris. Això és crucial per garantir la confiança dels usuaris i la integritat de la informació gestionada per l'aplicació.

1.2.2 Objectius secundaris

La meta última és fomentar una comunitat cohesionada d'excursionistes en línia, on la passió per la natura i l'aventura pugui expressar-se lliurement i compartir-se amb altres entusiastes de tot el món, aspirant a enriquir les experiències d'excursionisme dels usuaris i a contribuir al seu creixement personal i a la seva connexió amb la natura i amb altres persones que comparteixen la mateixa passió mitjançant funcionalitats col·laboratives, com ara compartir rutes, consells d'excursionisme i experiències personals, la plataforma busca promoure

l'intercanvi de coneixements i la creació de vincles entre els membres de la comunitat.

El fet que el codi sigui lliure per a qualsevol estudiant o persona que vulgui aprendre i seguir el desenvolupament de la plataforma permet una transparència molt gran i promou un lloc per intercanviar coneixement a través de la col·laboració. Això en no només beneficia als estudiants, sinó que també enriqueix la comunitat en línia, permetent una participació activa i fomentant el desenvolupament àgil i reformista de la plataforma.

1.3 Elements diferenciadors

Si bé és cert que ja existeixen algunes pàgines web semblants a la idea que estic exposant, trobo que els hi falten alguns elements que sí que estan presents en la meua implementació.

La plataforma més coneguda en l'àmbit de la natura, tant per a senderisme, rutes en bicicleta i molt més, és Wikiloc [2], una eina excel·lent per descobrir i compartir rutes a l'aire lliure la qual compta amb una versió web i una aplicació mòbil la qual et permet seguir les rutes de l'aplicació amb temps real i la possibilitat de gravar les rutes a través del senyal GPS per a tenir un registre més fidedigne i poder fer una publicació de manera més còmoda.

No obstant això, la meua plataforma està especialment centrada en l'experiència social i la comunitat d'excursionistes en línia. A diferència de Wikiloc, no només ofereix la possibilitat de descobrir i compartir rutes, sinó que també facilita la interacció i la connexió entre els usuaris que comparteixen interessos similars. En la meua integració en la pàgina d'inici, l'usuari té una llista de les excursions més recomanades per ell basat en les publicacions que li dona a 'M'agrada', dels usuaris que té com a amics, d'altres usuaris que estan en tendència i molts altres elements, aportant a l'usuari una experiència més personal i afí als seus gustos.

A més, Wikiloc bloqueja algunes de les seves funcionalitats més interessants darrere d'una subscripció mensual al seu servei, mentre que la meva aplicació és completament gratuïta i de codi obert. És important remarcar que a la meva aplicació no hi ha cap mena d'anunci, a diferència de Wikiloc que disposa del pla gratuït amb anuncis.

A més de ser una plataforma gratuïta i sense anuncis, el codi font de la meva aplicació està disponible com a codi obert i es troba penjat en un repositori de GitHub. Aquesta decisió permet que qualsevol persona interessada pugui explorar, col·laborar i millorar el desenvolupament de la plataforma. Això no només fomenta la transparència i la participació comunitària, sinó que també pot conduir a una evolució més ràpida i innovadora de la plataforma, amb contribucions i aportacions de desenvolupadors de tot el món.

A continuació es pot veure una taula amb els elements on coincideixen i que diferencien la meva aplicació de Wikiloc:

Característiques	Wikiloc	HikeNet
Descoberta i compartició de rutes	✓	✓
Seguiment de rutes amb GPS	✓	✗
Experiència social	✗	✓
Recomanació de rutes personalitzades	✗	✓
Subscripció mensual	✓	✗
Plataforma sense anuncis	✗	✓
Codi font obert	✗	✓

1.4 Abast del projecte

El desenvolupament d'aquesta aplicació no aspira a convertir-se en l'aplicació més emprada a escala mundial per excursionistes i esportistes perquè no es té ni el temps ni els recursos humans necessaris per a dur a terme aquesta tasca, això implica que s'ha d'establir els límits de l'aplicació des d'un inici per a poder assolir els objectius esmentats en els apartats anteriors.

D'aquesta manera el disseny visual de l'aplicació ha de ser complet, però no podem intentar arribar a un disseny professional amb maquetes de cada pàgina, crear un estàndard d'ús o poder brindar a l'usuari final una experiència totalment personalitzada, no obstant sí que podem definir una paleta de colors a fer servir, fer una col·lecció de components els quals emprem de manera repetitiva aportant una sensació de cohesió i una petita capa de personalització com noms d'usuari i biografies entre altres.

En l'àmbit de base de dades no es podrà crear un model conceptual molt extens amb una alta capa de configuració mitjançant paràmetres establerts pels administradors a través d'un backoffice, ja que el desenvolupament d'aquestes funcionalitats superaria àmpliament el temps de desenvolupament tenint en compte que es tracta d'un projecte d'una sola persona.

Per als algorismes de cerca que farem servir durant l'aplicació no emprem cap mena d'intel·ligència artificial a causa del fet que no es disposa ni dels recursos materials ni del coneixement suficient per a dur-ho a terme, el que sí que es vol aconseguir és crear un algorisme el prou intel·ligent per proporcionar una bona experiència a l'usuari final amb acord al seu perfil.

A l'aplicació no existirà una figura d'administrador que pugui moderar l'aplicació, es crearan les bases per a tenir usuaris amb un atribut administrador, però no es desenvoluparan les funcionalitats derivades com la modificació d'usuaris, l'eliminació de publicacions que puguin inferir amb les polítiques establertes o la moderació de comentaris, la modificació d'aquestes seccions de l'aplicació s'haurà

de fer directament des de la base de dades mitjançant les claus privades del servidor.

Finalment, com es tracta d'una aplicació per a l'excursionisme en algun moment s'haurà de crear un mòdul per enregistrar les excursions dels diferents usuaris, aquest projecte no contempli fer una aplicació mòbil per al seguiment en temps real a través del GPS, únicament es desenvoluparà una web-app capaç de complir les necessitats de l'usuari final sense que es vegi minvada la seva usabilitat.

1.5 Estructura del document

Aquest document s'estructura de la següent manera:

- Les primeres pàgines contenen la portada, l'índex i un breu resum de què es desenvoluparà al document.
- A continuació trobarem la introducció del TFG on entrarem en més detalls sobre el perquè d'aquesta proposta, on explicarem que volem aconseguir i que ens fa diferents de la competència.
- El tercer bloc conté els requisits necessaris del projecte, aquests han estat extrets mitjançant una sèrie de tutories i abasten tant funcionalitats com aspectes a més baix nivell.
- El següent apartat descriu l'arquitectura que es proposa per fer front als requisits esmentats anteriorment. Aquesta arquitectura proposa una solució full-stack, on cada àrea està dividida en un subapartat, a més d'un apartat dedicat al desplegament.
- Una vegada definits els elements anteriors ja es pot començar amb el desenvolupament tècnic dels requisits.
- En aquest següent capítol, amb l'aplicació ja acabada, analitzem els resultats i el nivell d'acompliment, a més de dur a terme proves amb usuaris externs per definir el nivell d'èxit i usabilitat real de la feina feta.
- A l'apartat de les conclusions és presenta que s'ha après i com és podria seguir millorant, tenint en compte les dificultats que s'han trobat i recomanacions per a futurs desenvolupadors. És el tancament del projecte.

- El darrer apartat està compost per la bibliografia del document, amb les referències a les pàgines i fonts consultades per dur a terme aquest treball.

Aquesta guia té com a objectiu donar un coneixement previ al lector de què esperar en cada secció, ajudant en la comprensió i assimilació dels coneixements que s'intenten explicar en aquest treball de fi de grau.

2. Requisits del projecte

En aquest apartat és presenten les diferents funcionalitats d'usuari de manera enumerada i amb una petita descripció del funcionament que tindran dins l'aplicació.

Cada requisit, tant funcional com no funcional tindran la següent estructura de taula, comptant amb un nombre de requisits, un títol per identificar ràpidament quin és el tema del requisit i una petita descripció per a donar un breu context sobre les restriccions i mínims que han de tenir.

Nº XX	Títol del requisit
Breu descripció del requisit	

La finalitat de redactar els requisits d'aquesta manera és que amb una breu ullada es pugui entendre que es vol aconseguir amb les possibles restriccions a l'hora d'implementar-los.

2.1. Requisits funcionals

Per a esmentar les funcionalitats i requisits que ha de tenir aquest TFG emprarem una estructura d'històries d'usuari, d'aquesta manera els requisits poden ajudar a entendre millor les necessitats dels usuaris i com aquestes es relacionen amb les funcionalitats del sistema. Una història d'usuari té la següent estructura:

"Com a usuari del sistema vull poder ..."

Una vegada tenim totes les user stories les hem de classificar entre funcionals i no funcionals, un requisit funcional és el que està relacionat de manera directa amb una funcionalitat del sistema i descriu de manera clara com ha de funcionar un

requisit, un exemple seria "Com a usuari que accedeix a l'aplicació vull poder veure la pàgina d'inici", en aquest cas el requisit està relacionat amb les funcionalitats dels usuaris els quals no han iniciat una sessió.

Aquesta llista de funcionalitats han estat redactades a partir d'una reunió inicial amb el tutor del projecte per englobar totes les funcionalitats que es creien necessàries perquè el projecte tingues el pes propi d'un treball de fi de grau. A més s'han implementat una sèrie de requisits propis per fer que l'aplicació ressaltés sobre les altres, fent ús dels coneixements adquirits durant el grau i a la feina com a desenvolupador front-end.

El projecte consta dels següents requisits d'usuari:

#1	Com a nou usuari que accedeix a la Web App vull poder crear un compte a l'aplicació
Quan un usuari accedeix a l'aplicació per primer com haurà de veure una pantalla la qual li demani crear un compte per a poder accedir a aquesta. El formulari comptarà amb la informació necessària per a la creació de l'usuari a la base de dades, impedir crear-lo si algun camp obligatori no està completat.	
#2	Com a usuari amb un compte a la Web App vull poder iniciar sessió amb el meu perfil
Quan un usuari ja hagi creat el seu compte haurà de poder iniciar sessió a l'aplicació, amb el nom d'usuari i la contrasenya esmentada en el formulari de registre.	
#3	Com a usuari amb una sessió activa vull poder veure i modificar el meu perfil
Quan un usuari ha iniciat sessió ha de poder accedir al seu perfil al qual es mostrarà el seu nom complet, la data de creació del perfil, la seva biografia i el nombre de seguidors que té. A la dreta del perfil es mostrarà la llista de	

publicacions que té. A més es mostrarà un botó d'edició, on a través d'una vista emergent i un formulari podrà triar els camps a canviar, tenint el formulari emplenat per defecte amb la informació actual.

#4	Com a usuari amb una sessió activa vull poder seguir a altres usuaris de l'aplicació
----	--

Un usuari ha de poder cercar a altres usuaris dins l'aplicació, per això es presentarà a l'usuari amb una entrada de teclat, on a mesura que es vagi escrivint anirà mostrant usuaris que coincideixin amb la seva cerca. Quan l'usuari premi sobre un resultat serà redirigit al perfil d'aquell usuari.

#5	Com a usuari vull poder xatejar amb altres usuaris en temps real a través del sistema de missatgeria intern
----	---

Quan un usuari registrat de l'aplicació accedeix al perfil d'un altre usuari podrà prémer sobre el botó 'Missatge' on es redirigirà a una pantalla de missatgeria, on es podrà tenir una conversació en temps real amb l'altra persona.

#6	Com a usuari vull poder penjar una nova excursió
----	--

Un usuari ha de poder penjar una nova excursió, se li presentarà un formulari amb els diferents camps a omplir. Una vegada el formulari té tots els camps requerits amb informació vàlida es podrà enviar el formulari, el qual quan es guardi dins la base de dades es redirigirà a l'usuari a la pàgina corresponent.

#7	Com a usuari vull poder editar una excursió ja existent associada al meu perfil
----	---

Quan un usuari està a la pàgina de la seva pròpia excursió ha d'aparèixer un botó la qual permeti editar la informació d'aquesta.

#8	Com a usuari vull poder indicar que m'agrada una publicació, a més de poder comentar una publicació d'algun altre usuari
Quan estic veient publicacions d'altres usuaris vull poder indicar que m'agrada aquesta publicació, a més de poder deixar un comentari a aquesta.	

#9	Com a usuari vull poder tenir una 'feed' adaptada als meus gustos
Quan accedeixo a l'aplicació amb el meu usuari vull veure una llista d'excursions creada d'acord amb els meus gustos, la qual s'actualitza amb el temps i a mesura que segueixo a més gent.	

#10	Com a usuari vull poder penjar una excursió des d'un fitxer GPX
A l'hora d'omplir la informació del formulari per a crear una excursió l'usuari pot penjar un fitxer amb format .GPX d'una excursió que ha realitzat de manera prèvia amb alguna aplicació externa. D'aquesta manera el recorregut que ha realitzat serà més precís i no ho haurà d'introduir a mà.	

#11	Com a usuari vull poder enllaçar imatges a les publicacions
Mentre estic omplint el formulari per a la creació d'una excursió vull tenir l'opció de penjar les meves imatges i que quedin adjuntades a l'excursió en si.	

#12	Com a usuari vull poder canviar la meua contrasenya
Com a usuari que ja té un compte vull tenir l'opció de sol·licitar una nova contrasenya, perquè ja no me'n record de la que vaig posar, aquest procés ha de ser segur i no ha de presentar possibles falles de seguretat.	

#13	Com a usuari vull poder canviar l'idioma de la web app
Com a usuari de l'aplicació o visitant sense compte vull poder canviar l'idioma	

de l'aplicació, vull poder triar entre Català, Castellà i Anglès.

2.3. Requisits no funcionals

Per altra banda, tenim els requisits no funcionals, els quals estan relacionats en com ha de ser el sistema i quines especificacions ha d'aconseguir, els quals, per norma general estan relacionats amb termes de qualitat i seguretat. Per entendre millor a quin aspecte del sistema s'està fent referència emprarem la següent estructura per esmentar els requisits no funcionals de l'aplicació.

"Àmbit general - Descripció breu"

El projecte compta amb els següents requisits no funcionals:

#1	Rendiment - Temps de resposta
L'aplicació ha de carregar les pàgines i respondre a les accions de l'usuari en menys de 2 segons en condicions normals d'ús, existeix l'única excepció de quan un usuari entra per primera vegada a la pàgina on aquesta pot tardar fins a 60 segons a carregar i muntar tota l'aplicació.	

#2	Seguretat - Autenticació i autorització
L'aplicació ha d'implementar mecanismes segurs d'autenticació i autorització per protegir les dades dels usuaris, en el següent apartat explicarem com emprarem JWT per a la generació i validació de sessió d'usuari.	

#3	Seguretat - Xifratge de dades crítiques
Les dades crítiques dels usuaris, com les contrasenyes d'usuari, no poden estar	

guardades com a text pla, ja que sí que hi hagués una filtració de dades estaríem exposant informació crítica dels usuaris, en canvi, les dades han de ser hashetjades i mai seran enviades pel servidor.

#4	Usabilitat - Interfície intuïtiva
----	-----------------------------------

L'aplicació ha de tenir una interfície d'usuari clara i intuïtiva que permeti als usuaris navegar fàcilment i dur a terme tasques sense dificultats. Això es farà a través dels usos repetitius dels mateixos elements amb una paleta de colors ben definida i emprant un estil minimalista que se centra en la usabilitat.

#5	Mantenibilitat - Codi net i documentat
----	--

El codi de l'aplicació ha de ser net, ben estructurat i documentat per facilitar el manteniment i l'evolució futura, emprant, sempre que sigui possible, una estructura de MVC i crear components i funcions genèrics per reduir la repetició de codi al màxim possible.

#6	Compatibilitat - Compatibilitat de navegadors
----	---

L'aplicació ha de ser compatible amb els navegadors web més utilitzats i adaptar-se a diferents dispositius (ordinadors, mòbils ...) sense significar un descens en el nivell de qualitat dels usuaris.

#7	Disponibilitat - Alta disponibilitat
----	--------------------------------------

L'aplicació ha d'estar disponible el 99,9% del temps, excloent els períodes de manteniment planificats.

#8	Interoperabilitat - Integració amb serveis externs
----	--

L'aplicació ha de poder integrar-se fàcilment amb serveis externs, com ara APIs de mapes, serveis d'autenticació de tercers, etc

#9	Eficiència - Optimització de recursos
L'aplicació ha de ser eficient en l'ús de recursos del servidor per minimitzar els costos operatius i maximitzar el rendiment.	

#10	Integritat - Integritat a la base de dades
L'aplicació ha de garantir la integritat de les dades emmagatzemades a la base de dades. Això es farà a través de l'ús de regles de validació estrictes, la implementació de claus primàries i estrangeres per mantenir la coherència de les dades.	

3. Arquitectura propuesta

Per al desenvolupament d'aquesta aplicació web, hem optat per utilitzar una combinació de tecnologies per al back-end i el front-end, amb l'objectiu de proporcionar una plataforma completa i funcional per als usuaris.

Aquesta arquitectura ha tingut en compte tots els requisits esmentats en l'apartat anterior i es creu que és la millor implementació i enfocament que es pot fer amb els coneixements i temps del qual es disposa.

3.1 Arquitectura front-end

Quant al front-end de l'aplicació, hem optat per utilitzar Vue 3 [3], una popular biblioteca de JavaScript per al desenvolupament d'interfícies d'usuari, el qual ofereix una sintaxi clara i intuïtiva, així com un rendiment optimitzat. Les propietats reactives de Vue faciliten la gestió de l'estat de l'aplicació i la seva reactivitat davant de canvis en les dades. A més, la gran quantitat de llibreries i components disponibles per a Vue ens proporcionen una gran flexibilitat i faciliten el desenvolupament d'una interfície d'usuari atractiva i funcional per als nostres usuaris.

Com ja s'ha comentat abans emprarem una llibreria de components per a desenvolupar més ràpidament la web app, aquests components ens permetran reduir codi repetit, ja que els reemplacem a les diferents seccions augmentant la cohesió de l'aplicació. La llibreria principal que emprarem serà Shadcn, biblioteca de codi lliure, amb components accessibles i gratuïts. Alguns dels components que emprarem seran propis, els quals ja s'havien desenvolupat per altres projectes, aprofitant el temps de desenvolupament passat.

També una raó pel que es va triar Vue és perquè és el framework modern de JavaScript amb el que tinc més experiència.

3.2 Arquitectura back-end

Pel que fa al back-end, hem implementat un sistema de gestió de base de dades utilitzant PostgreSQL [4], en la seva versió 15. Aquesta elecció s'ha fet tenint en compte la robustesa i la fiabilitat d'aquest sistema, així com la seva capacitat per gestionar grans quantitats de dades de manera eficient.

Per a interactuar amb la base de dades, hem fet servir el llenguatge de programació JavaScript a través de l'entorn d'execució Node.js. Això ens ha permès aprofitar la familiaritat i la versatilitat de JavaScript en tot el nostre codi, des del back-end fins al front-end. Per gestionar la interacció amb la base de dades de manera més estructurada i eficaç, hem incorporat l'ORM (Object-Relational Mapping) Sequelize [5]. Aquesta eina ofereix una interfície de programació orientada a objectes per a la manipulació de dades relacionals, permetent-nos interactuar amb la base de dades mitjançant comandes JavaScript en lloc de sintaxi SQL, la qual cosa simplifica el desenvolupament i el manteniment del codi.

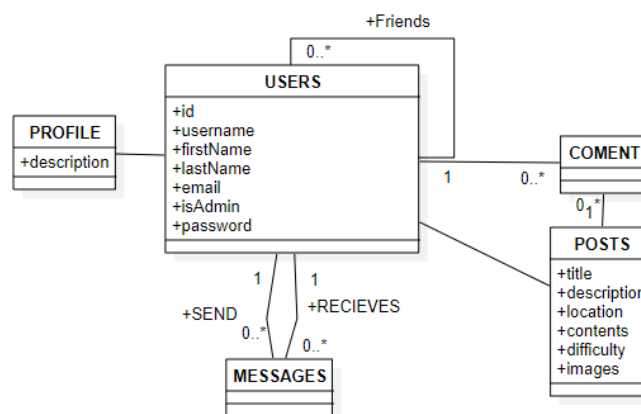
Amb l'objectiu de millorar el desenvolupament i la portabilitat del back-end i la base de dades, hem dockeritzat aquesta part del projecte. La dockerització implica encapsular l'aplicació juntament amb totes les seves dependències i configuracions en un contenidor Docker [6].

El motiu pel qual es va decidir dockeritzar és la portabilitat que ens ofereix, amb Docker, el nostre back-end i la base de dades es poden executar en qualsevol entorn que tingui Docker instal·lat, independentment del sistema operatiu o la configuració de l'amfitrió. Això facilita el desplegament del projecte en diferents entorns sense preocupar-se pels conflictes de dependències. Això era molt important, ja que el desenvolupament es fa sobre un entorn de Windows mentre que el servidor on està allotjat es tracta d'un entorn d'Ubuntu, a més de desenvolupar a diversos ordinadors alhora, d'aquesta manera ens estalvien molt de temps configurant i instal·lant els diferents serveis que són necessaris per al correcte funcionament de l'aplicació.

Tots els endpoints de l'API es troben documentats, es poden consultar al següent enllaç.

3.3 Model base de dades

Per a modelar la base de dades s'ha emprat un diagrama per il·lustrar les diferents classes i relacions necessàries per al correcte funcionament.



Model de la base de dades HikeNet.

Com podem observar es tracta d'una model no massa gran el qual està centrat en l'usuari.

Per a cada usuari associem un perfil, de moment aquesta taula només conté un camp el qual és la seva biografia, mirant-ho així podríem pensar fusionar-la a la taula usuari i estalviar-nos una taula, en el nostre cas s'ha decidit deixar-ho com una taula a part pensant en l'escalabilitat de l'aplicació, aquests poden ser uns camps els quals variïn molt més que la informació personal de l'usuari, a més que es podrien afegir camps nous en futures implementacions com podrien ser zona preferida, nivell de dificultat mitjana de les seves excursions, com de regular sol fer les excursions, manera preferia de fer les excursions (córrer, bici, caminant...) entre altres.

La relació que es pot observar d'usuari amb ell mateix es correspon amb la funcionalitat de seguir a un altre usuari, un usuari pot seguir a molts d'usuaris, i

de manera inversa una persona pot ser seguida per molts d'usuaris, això es traduirà en una taula intermèdia quan s'implementi a través de Sequelize, amb una relació Many-to-Many.

Si ens centrem en la part dreta del diagrama podem analitzar com estructurarem la funcionalitat de creació d'excursions, una excursió està composta per un títol, un nivell de dificultat el qual pot variar entre 1 (molt fàcil) i 5 (molt difícil), una descripció, un array d'objectes que representaran els diversos punts pels quals passa l'excursió, una ubicació la qual es calcularà amb la mitja de tots els punts i finalment un array d'imatges.

A més d'aquesta informació podem veure que una excursió està relacionada amb N comentaris, aquests alhora estan relacionats amb un usuari el qual és qui ha comentat en l'excursió, d'aquesta manera la nostra aplicació augmenta el seu nivell on no només podem veure excursions sinó que també podem interactuar amb elles, fent més interessant la nostra implementació.

Per acabar, ens queda veure com implementem dins base de dades el xat entre usuaris, tenint en compte que ni dins la universitat ni a la feina he fet un sistema de missatgeria he optat per una implementació senzilla, però igual d'útil, en lloc de tenir sales de xat o grups com podem trobar en aplicacions com whatsapp he decidit només tenir converses amb un únic usuari alhora, d'aquesta manera reduïm de manera notable la complexitat de la tasca sense afectar els requisits esmentats en l'apartat anterior. Els missatges tenen dues relacions amb usuari, una relació d'enviament indicant qui és l'emissor i una relació de rebre, indicant a qui va destinat cada missatge, un missatge té un únic camp el qual és un string amb la informació que es vol transmetre.

Per a la integració de la base de dades dins de Sequelize, s'han fet servir migracions per millorar l'escalabilitat de l'aplicació. Una migració és un fitxer que conté una sèrie de comandes per modificar l'esquema de la base de dades, com ara crear o modificar taules, afegir o eliminar columnes, o establir restriccions. Aquestes migracions permeten gestionar els canvis en l'esquema de la base de dades de manera controlada i reversible.

En el nostre cas hem creat una migració per a cada taula nova que afegíem a la base de dades, d'aquesta manera si detectàvem un error durant el desenvolupament de l'aplicació era més de resoldre.

3.4 Desplegament del projecte

Per a desplegar la webapp a un servidor web hem emprat dos serveis completament gratuïts externs a la universitat, pel fet que estem emprant el seu pla de franc hi ha un parell de restriccions que comentarem al final d'aquest apartat.

Per al front-end, hem utilitzat Vercel, una plataforma de desplegament que ofereix allotjament per a aplicacions web estàtiques i dinàmiques. Vercel ofereix una integració fàcil amb diversos marcs entre els quals es troba Vue. A més, proporciona funcionalitats avançades com a la previsualització d'implementacions i l'escalat automàtic, tot mantenint un enfocament en la simplicitat i la facilitat d'ús.

Per al back-end i la base de dades, hem emprat Render, una plataforma que ofereix allotjament de servidors i bases de dades gestionades. Render permet desplegar aplicacions web, serveis API i bases de dades PostgreSQL de manera senzilla i sense necessitat de configuració complexa, a més permet allotjar imatges Docker, fent el procés molt més fàcil. Render també ofereix una capa de seguretat i monitoratge integrada, que simplifica la gestió i el manteniment dels nostres serveis en línia.

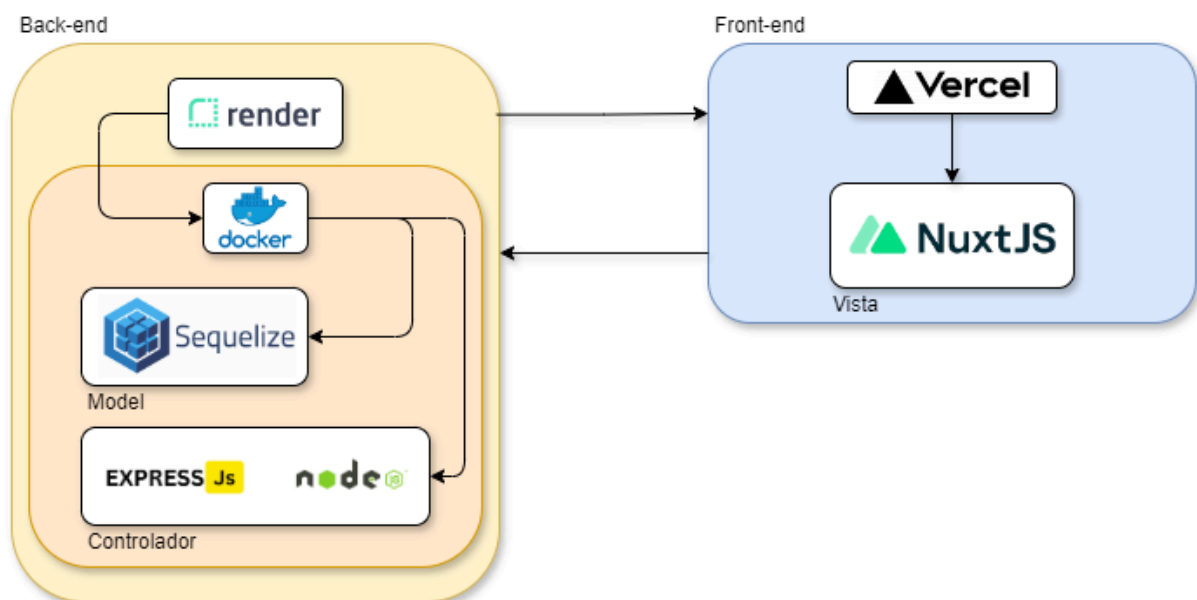
Tot i que hem pogut aprofitar els plans gratuïts d'aquests serveis, és important tenir en compte algunes restriccions com ara límits en el consum de recursos o la disponibilitat de funcionalitats avançades en aquests plans. Malgrat això, per a les necessitats del nostre projecte, aquests serveis gratuïts ofereixen una solució adequada per al desplegament de la nostra aplicació web.

Enllaços:

- Front-end: <https://hikenet-frontend.vercel.app/>
- Back-end: <https://hikenet-backend-backup.onrender.com/>

3.5 Visió general de l'arquitectura

En aquest subapartat es pretén donar una representació més visual i de com s'organitza l'arquitectura del projecte en forma de diagrama. En aquest diagrama es pot apreciar com és l'estructura general del projecte, amb quines tecnologies estan construïdes les diverses parts de l'aplicació i com interaccionen entre elles.



Arquitectura de HikeNet.

4. Desenvolupament del projecte

En aquest apartat s'explica quina ha sigut la manera en què s'ha orientat el projecte i el temps planificat per cada part a més de com s'han implementat els diferents requisits que s'han esmentat en l'apartat anterior.

4.1 Planificació temporal

La planificació temporal d'un projecte és una part crítica del seu èxit, especialment quan es tracta d'un projecte amb múltiples requisits i objectius. En el cas d'aquest projecte, que té com a objectiu desenvolupar una aplicació full-stack per a la comunitat d'excursionistes en línia com ja s'ha comentat en els apartats anteriors.

Durant el primer semestre no es planteja fer gaire desenvolupament pel fet que encara tinc assignatures a cursar a més d'una feina a mitja jornada, per tant, s'enfocarà a començar a establir els objectius, l'abast del projecte i una primera maquetació, tot això és necessari per estudiar la viabilitat i decidir si els requisits que s'han proposat són realistes o no.

Durant el segon semestre és quan es preveu començar un desenvolupament continu de l'aplicació, deixant el darrer mes abans de l'entrega reservada a proves, a la revisió de la documentació i a preparar la defensa d'aquest treball, a continuació es pot veure una línia temporal de la planificació.

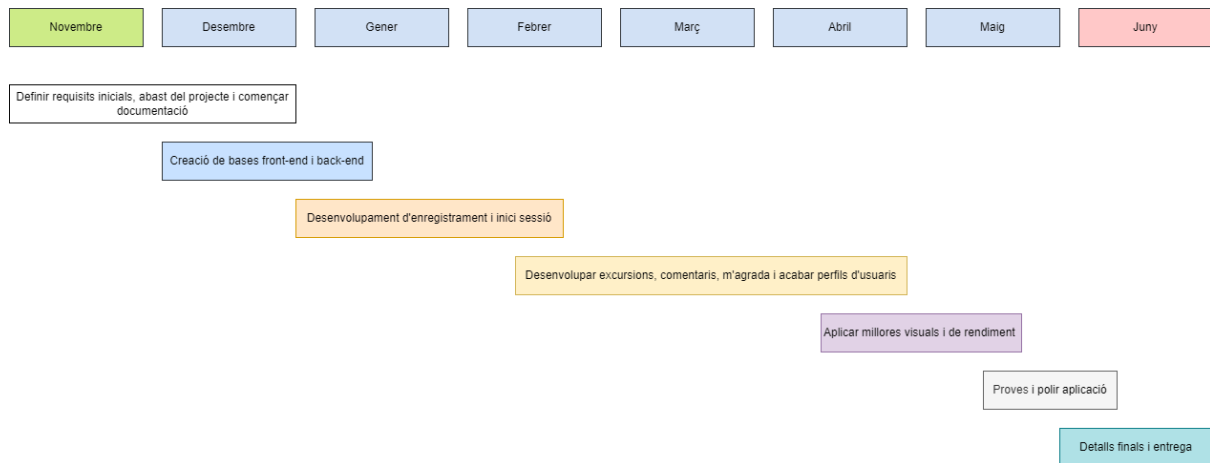


Diagrama de Gantt

Durant el desenvolupament de tota l'aplicació s'ha estat redactant la documentació, tant de desenvolupament com apartats no tècnics, d'aquesta manera no s'ha hagut de guardar temps específic per escriure tota la documentació sinó que està inclòs el temps dins cada tasca.

4.2 Desenvolupament dels requisits funcionals

Aquest apartat està destinat a una explicació tècnica dels diferents requisits funcionals que s'han esmentat anteriorment, en aquest apartat s'entra en detalls els quals no són visibles per un usuari quan està navegant per l'aplicació.

4.2.1: Creació d'un compte d'usuari

Per a la creació d'usuaris, s'han implicat tant el front-end com el back-end. Al front-end, s'ha desenvolupat la pàgina que mostra el formulari corresponent i maneja l'inici de sessió per emmagatzemar les dades d'usuari en les Cookies del navegador, evitant la necessitat d'iniciar sessió repetidament. Al mateix temps, al back-end, s'ha treballat en la validació de la informació proporcionada amb els registres de la base de dades, a més de generar els tokens d'usuari corresponents.

Pel que fa a la part de back-end, és essencial poder guardar tota la informació de l'usuari a la base de dades per als futurs inici de sessió i recuperar els perfils d'usuari. Això planteja un possible problema de seguretat pel fet de guardar les

contrasenyes, informació molt sensible. Per mitigar aquest risc, hem optat per no emmagatzemar les contrasenyes en format de text pla, sinó que les hem hashejat. Aquest procés converteix les contrasenyes en una cadena de caràcters aparentment aleatòria, anomenada hash, que no es pot revertir fàcilment per recuperar la contrasenya original. Apliquem el hashing de la nova contrasenya i guardem tota la informació dins la base de dades, creant un token d'accés.

Per a la generació i verificació d'aquests tokens, utilitzem JWT (JSON Web Token), un estàndard per a la transmissió segura d'informació entre parts en forma de tokens [12], cal remarcar que per molt que la informació de la contrasenya és segura mai es comparteix amb el front per evitar possibles filtracions o atacs de força bruta.

Quan el front-end recull la informació és guarda el token dins les cookies del navegador per evitar tornar a iniciar sessió cada vegada que la pàgina es munta, sinó que es queda la sessió iniciada fins que el token caduca. A més a partir del moment que l'usuari s'autentica s'afegeix una capçalera d'autenticació que s'envia en les peticions al back-end, després es redirigeix a l'usuari a la pàgina principal on poden veure la seva feed personalitzada.

Amb aquesta implementació, tenim una base robusta per a l'autenticació dels usuaris i la gestió segura de les seves dades. Això assegura no només la privadesa dels usuaris sinó també la integritat i la seguretat del sistema en general.

4.2.2: Inici de sessió d'un usuari existent

Una vegada el mòdul per a registrar un nou usuari és completat, l'inici de sessió esdevé una tasca trivial. Quan un usuari intenta iniciar la sessió, la contrasenya proporcionada és sotmesa al mateix procés de hashing utilitzat durant el registre, i el hash resultant és comparat amb el hash emmagatzemat a la base de dades. Si coincideixen, l'usuari és autenticat amb èxit i se li genera un token d'accés utilitzant JWT, tal com es fa durant el registre dels nous usuaris.

Seguidament, el front-end aplica el mateix procés que durant el registre dels nous usuaris. Recull el token JWT format amb la informació de l'usuari i el guarda dins les cookies del navegador per verificar la sessió de l'usuari.

És important destacar que al front-end s'han creat rutes protegides per a usuaris registrats. Aquestes rutes restringides només són accessibles per usuaris autenticats, assegurant una millor experiència d'usuari i evitant errors. Un exemple crucial és l'accés al perfil d'usuari, on si un visitant intenta accedir sense haver iniciat sessió, es prevenen problemes d'integritat de les dades i de seguretat.

A través d'aquestes comprovacions evitem possibles errors d'integritat que és puguin produir, a més de brindar a l'usuari una experiència més fluida i lliure d'errors.

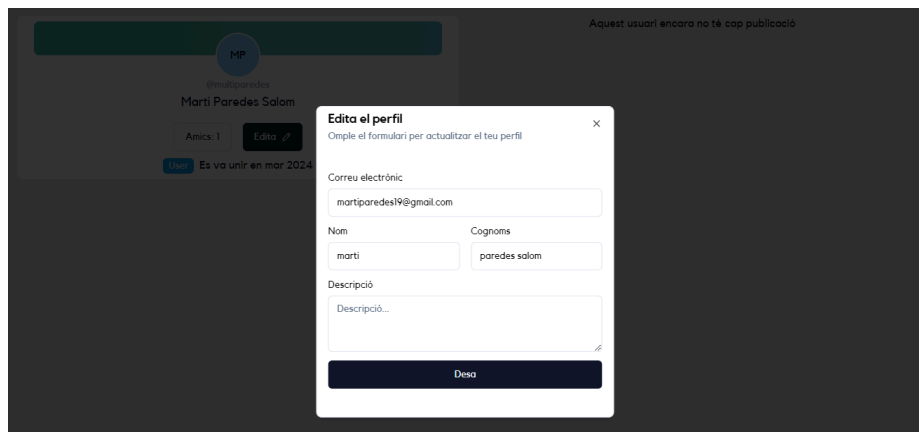
4.2.3. Visualització i modificació d'un perfil d'usuari

Un usuari el qual ja està donat d'alta a l'aplicació se li assigna un perfil, aquest mostra la seva informació relacionada, coms són el nom complet (nom més cognoms), el nom d'usuari que va indicar al formulari d'enregistrament, la data en la qual es va donar d'alta i una biografia. La primera vegada que l'usuari accedeixi a aquesta pàgina es mostrarà a la biografia un missatge per defecte indicant que l'usuari encara no té una biografia pròpia.

A més d'aquesta informació es mostrarà un botó per a editar aquesta informació. També es mostra una casella amb el nombre de seguidors que té i, si es posa el ratolí damunt del botó, apareixerà una llista completa d'aquests seguidors.

Quan l'usuari del perfil prem el botó d'editar perfil es mostrà una finestra modal. Una finestra modal és un element d'interfície d'usuari que apareix superposada sobre el contingut principal d'una pàgina web o aplicació, bloquejant la interacció amb el contingut de fons fins que l'usuari interaccioni amb la finestra modal o la tanqui [7]. D'aquesta manera podem proporcionar a l'usuari una edició de les

dades sense sortir de la pàgina, on quan els canvis es desin el modal es tanca automàticament, actualitzant les dades de la pàgina de perfil. El formulari contendrà els següents camps: Correu electrònic, nom, cognoms i descripció, el formulari ja estarà omplert amb la informació de l'usuari i tots els camps menys la biografia són obligatoris.

The image shows a dark-themed user profile page. At the top, there's a header with a profile picture placeholder and the text 'Aquest usuari encara no té cap publicació'. Below this, the profile name 'Marti Paredes Salom' is displayed. A modal window titled 'Edita el perfil' is open in the center, containing a form with the following fields: 'Correu electrònic' (filled with 'martiparedes19@gmail.com'), 'Nom' (filled with 'martí'), 'Cognoms' (filled with 'paredes salom'), and 'Descripció' (with a placeholder 'Descripció...'). A 'Desa' button is at the bottom of the modal. The background page also shows a button 'Edita' and a status 'Es va unir en mar 2024'.

Finestra modal esmentada.

Com podem observar als camps editables de nom d'usuari no està present, això és degut al fet que encara que cada usuari tingui un ID únic el back-end tracta aquest camp per a fer la majoria de les cerques dins la base de dades, a més de mostrar les rutes del front-end sobre aquest camp i no sobre el camp ID, fent que quan visites un perfil l'usuari té més clar quin està visitant mostrant un URL llegible en lloc d'una cadena de caràcters i nombres aleatoris.

Finalment, a la dreta d'aquest apartat es mostrarà una llista de les excursions que ha creat l'usuari, aquesta serà una vista més reduïda de l'excursió mostrant una petita previsualització, el nom de l'excursió i la descripció d'aquesta.

En el que respecta al back-end es va crear un endpoint per a l'edició de la informació que s'ha esmentat anteriorment, encara que al front-end no es deixi enviar el formulari si no s'han omplert tots els camps requerits el back-end ho torna a revisar per si escau, d'aquesta manera si sorgeix un error en el codi que permeti aquest enviament encara preservarem la integritat de la base de dades.

4.2.4: Cerca i seguiment d'altres usuaris

Per al funcionament complet de l'aplicació és essencial poder cercar els usuaris registrats a la base de dades. Aquesta funcionalitat ha de ser accessible des de qualsevol part de l'aplicació i per aconseguir-ho s'ha inclòs un apartat dedicat als usuaris a la barra de navegació superior.

Quan l'usuari selecciona aquest apartat, es mostren dues opcions: editar el perfil, que redirigeix l'usuari al seu propi perfil i una segona opció per cercar altres usuaris.

En seleccionar l'opció de cerca s'obre una finestra modal amb un camp de text per introduir el nom de l'usuari a cercar. Inicialment, es mostren cinc resultats, no obstant això, aquest plantejament presenta un problema: cada vegada que es modifica el text del camp de cerca, es realitza una crida al servidor per obtenir els resultats. Aquesta crida no és instantània i no es duu a terme de manera seqüencial, fet que podria afectar negativament l'experiència de l'usuari.

Per resoldre aquest problema, s'ha decidit implementar una funció de 'debounce' on només es fa la trucada amb els usuaris filtrats quan hagin passat 0.5 segons sense cap modificació al camp de text, d'aquesta manera assegurem una millor experiència d'usuari i reduïm el nombre d'accions que ha de dur a terme l'usuari per fer la tasca com moure el ratolí per prémer un botó de cercar.

Quant al back-end s'ha creat un endpoint pel filtratge, per al camp a filtrar no només s'empra el camp nom d'usuari, s'aplica una funció OR amb el nom d'usuari complet, el qual es crea concatenant el nom i els cognoms, d'aquesta manera s'amplia les possibilitats de trobar el perfil sense necessitar camps addicionals.

4.2.5: Xatejar amb altres usuaris en temps real

El fet que Hikenet sigui una web-app social enfocada a la interacció entre usuaris tenir un sistema de missatgeria propi era una funcionalitat la qual em va cridar molt l'atenció des d'un principi, tant pel fet de poder donar als usuaris una

funcionalitat tan emprada dia a dia com el repte tècnic que presentava aquest requisit.

Finalment, durant una tutoria i estimar el temps necessari per a poder dur a terme aquesta tasca de manera completa és va decidir no desenvolupar-la per aquest TFG, encara que és deixaran les estructures necessàries de la base de dades per a una futura implementació en el cas que es vulgui seguir amb el projecte una vegada s'entregui aquest TFG.

4.5.6 Penjar una excursió

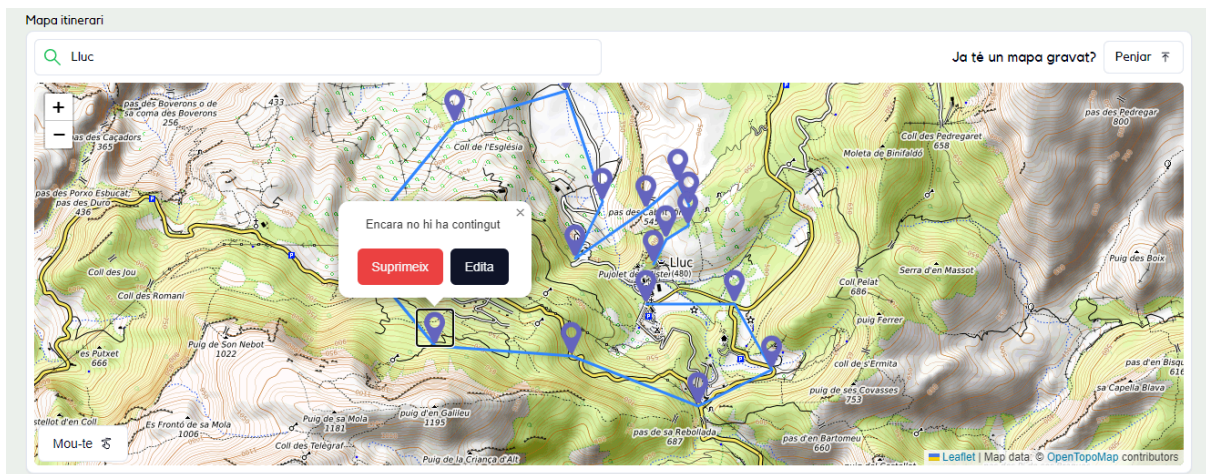
Aquest requisit és un dels cores de l'aplicació, ja que tot el desenvolupament fet són les bases per als següents requisits funcionals. Per aquesta raó s'ha dedicat una part important de la planificació d'hores a aquest mòdul. Perquè un usuari pugui penjar excursions és necessari que estigui registrat a l'aplicació i que tingui una sessió oberta, si una persona visita aquest apartat serà redirigit a la pantalla per iniciar sessió amb i apareixerà una notificació indicant que aquesta pàgina és exclusiva per usuaris registrats de l'aplicació.

Recordant els camps que tenia la taula excursió al model de base de dades l'usuari ha de proporcionar els següents camps per a poder fer una inserció correcta a la base de dades: un títol, un nivell de dificultat, una descripció, un itinerari representat per un array d'objectes indicant els diversos punts pels quals passa l'excursió, una ubicació la qual es calcularà amb la mitja de tots els punts i finalment un array d'imatges, quasi tots aquests camps són obligatoris a excepció de les imatges. Entrant una mica més en detall en els diferents camps el títol i la descripció són els més senzills i fàcils d'entendre, són camps de text de diferents mides, sent major els de descripció on l'usuari pot fer un resum de l'excursió, explicar quines dificultats presenta, precaucions a tenir en compte, quina fauna autòctona es pot trobar entre altres, és a dir, tot el que l'usuari vulgui escriure per a fer la seva publicació més rica i completa possible.

Per altra banda, tenim el camp per a seleccionar la dificultat de l'excursió, on 1 significa que es tracta d'una excursió molt senzilla mentre que 10 molt tècnica i

orientada únicament a excursionistes experimentats. Per a personalitzar una mica més l'aplicació i aplicant els coneixements de front adquirits en lloc d'emprar un camp numèric simple s'ha optat per implementar un 'range slider', un tipus d'entrada HTML que permet als usuaris seleccionar un valor únic entre dos valors [8]. En la majoria dels casos, consisteix en una barra horitzontal o vertical amb una maneta que l'usuari ajusta amb el seu cursor o dit. Es pot veure aquesta funcionalitat més sovint quan s'ajusta el volum en el navegador o en un dispositiu. També es poden utilitzar per triar un valor específic d'una escala gran o contínua de valors, quan una opció d'elecció individual (com ara els botons de ràdio) no seria pràctica, en el nostre cas optem per aquesta darrera opció, on volem que el nostre usuari seleccionin un nombre contingut entre 1 i 10, permetent ajustant el valor en una unitat. Per a millorar la usabilitat i experiència de l'usuari s'ha afegit a l'esquerra del slider un camp numèric indicant quina és l'opció seleccionada en tot moment, d'aquesta manera l'usuari no ha d'intuir quin nombre ha seleccionat, inicialment el camp es troba a 0, nombre el qual no és vàlid, si l'usuari intenta publicar l'excursió sense aquest camp canviarà el color dels texts a vermell amb un missatge informatiu baix del mateix informat que seleccioni una opció vàlida.

Continuant amb el camp trobem el mapa interactiu on l'usuari pot indicar les pautes de la seva excursió, segurament el submòdul més important i més interessant de tota l'aplicació, no té sentit crear una xarxa social amb les darreres tecnologies i amb moltes funcionalitats addicionals, interessants i innovadores si la base falla, per tant, s'ha intentat polir al màxim aquesta secció arribant a una versió final molt positiva.



Mapa per introduir una excursió

Com es pot observar a la figura de dalt el camp es pot dividir en dues seccions, una barra superior amb diferents opcions i una part central mostrant un mapa.

Començant per la secció del mapa podem analitzar a la part inferior dreta com estem emprant la llibreria de Leaflet, "la biblioteca de JavaScript de codi obert líder per a mapes interactius adaptats per a mòbils. Amb un pes d'uns 42 kB de JS, té totes les funcions de mapatge que la majoria dels desenvolupadors necessiten" [9], he decidit fer servir aquesta llibreria pels avantatges que esmentades i perquè ja l'he emprada a diferents projectes durant els anys, a part de Leaflet estem emprant la capa de OpenTopoMap la qual podem veure el relleu de les muntanyes, els camins, carreteres i puigs, ideal per la nostra aplicació d'excursionisme.

Quan l'usuari accedeix a aquesta secció apareix un pop-up al seu navegador demanant si cedeix la seva ubicació a Hikenet, d'aquesta manera podem centrar la vista del mapa en la ubicació actual de l'usuari, millorant la seva experiència, ja que el més segur és que l'excursió l'hagi realitzat a un lloc proper d'on es troba, si aquest no és el cas, i tampoc és correcte assumir com a veritat absoluta aquest fet, l'usuari pot fer servir el camp de cerca situat a la barra de navegació superior, on de manera semblant al requisit funcional de '4.2.4: Cerca i seguiment d'altres usuaris' es van guardant les dades introduïdes per l'usuari, però només es fan les peticions quan deixa d'escriure un nombre determinat de segons, per convertir el camp de text a un punt al mapa amb una latitud i altitud precisa s'ha fet servir l'API de Nominatim, un servei de codi obert amb un pla gratuït per fer cerques

inverses de geo-codis, la qual té un registre a escala mundial oferint un temps de resposta relativament ràpid i ordenant els resultats per importància, on si l'usuari no està conforme amb el resultat obtingut pot augmentar el detall de la cerca introduint un nom d'ubicació més específic, quan s'obté una resposta agafem la ubicació més rellevant i movem la vista del mapa a aquella ubicació en específic, per evitar que sigui un canvi brusc i inesperat per l'usuari s'utilitza una funció que simula un vol, allunyant la vista del punt inicial i viatjant en línia recta fins al punt de destí, amb una velocitat progressiva, fins a fer zoom al punt cercat per l'usuari. Cal remarcar que l'usuari pot moure-se de manera lliure arrossegant el ratolí per la secció del mapa, aquesta funcionalitat esmentada intenta millorar la usabilitat, però no es vol bloquejar a l'usuari a emprar-la si no vol.

Quan l'usuari es troba a la zona i vol començar a marcar l'excursió, haurà de prémer el botó situat a l'extrem inferior esquerra. Aquest botó alterna entre els estats 'Edita' i 'Mou-te'. Quan es troba en l'estat 'Edita', es bloquegen els controls de moviment mitjançant el ratolí, i cada clic que es fa sobre el mapa es converteix en una ubicació afegint una xinxeta de color blau al mapa. Internament, cada ubicació representa un punt per on l'usuari passa. Aquest punt està format per un ID únic, una latitud, una longitud i un camp de text editable per l'usuari. Aquest camp permet a l'usuari personalitzar la ubicació, indicant, per exemple, si és un mirador, un desviament de la ruta principal o l'inici de l'excursió.

El camp de text és visible i editable per l'usuari quan prem sobre un d'aquests punts. En aquest cas, apareix una carta per damunt amb el text indicat o un missatge genèric si encara no s'ha introduït cap text. L'usuari té l'opció d'editar mitjançant una finestra modal amb un formulari o un botó per eliminar la ubicació en cas que s'hagi equivocat en prémer sobre el mapa.

Per donar una millor impressió d'una ruta d'excursió, totes les ubicacions es van connectant mitjançant una línia de color blau, intentant semblar-se a una ruta sobre un mapa. Això crea una visualització més coherent i facilita la comprensió de la seqüència d'ubicacions com una ruta planificada. A més, aquesta línia proporciona un context visual addicional que reforça la idea d'una experiència d'excursió.

Per acabar aquest apartat sobre el mapa parlarem sobre el botó situat a la part superior dreta de la targeta, on es demana a l'usuari si ja disposa d'una excursió guardada, aquest requisit serà explicat amb més detall a l'apartat "4.2.10: Penjar una excursió des d'un fitxer GPX" on la idea és que si l'usuari disposa d'un dispositiu electrònic com pot ser un telèfon o rellotge amb un sensor GPS i ja ha enregistrat l'excursió que vol penjar pot penjar el fitxer amb format .GPX i així no fa falta que marqui de manera manual per on l'ha realitzat.

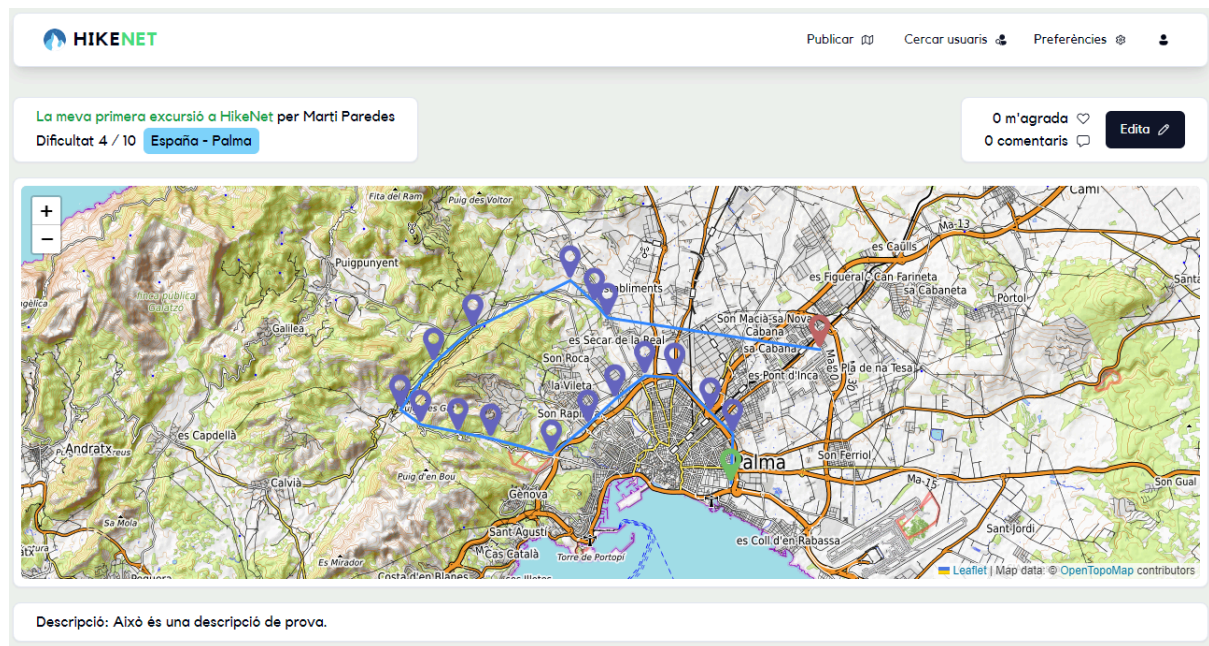
Finalment, arribem a la darrera funcionalitat d'aquesta pàgina, la possibilitat de penjar imatges de l'excursió, per això s'ha creat un nou component genèric el qual s'explicarà en més detall en el requisit "**4.2.11: Penjar una excursió des d'un fitxer GPX**".

Quan un usuari ha omplert tots els camps amb la informació necessària pot penjar l'excursió, en el cas que hi hagi alguna mena d'error el front-end avisaria a l'usuari abans de fer la petició que provoqui que el back-end es rompi. Mentre s'està creant es mostra una petita animació per donar una confirmació visual que la seva petició està en curs, quan aquesta acaba es mostra una notificació i es redirigeix a la nova pàgina creada on es veu la informació de l'excursió i ja pot ser vista per tothom de l'aplicació.

En resum, el mòdul de penjar excursions és una part crítica de l'aplicació, ja que estableix les bases per als futurs requisits funcionals. Permet als usuaris compartir les seves experiències d'excursió amb altres membres de la comunitat, contribuint a enriquir la plataforma amb contingut de qualitat i útil per a la resta d'usuaris. Des de l'autenticació fins a la captura de la ruta d'excursió i la pujada d'imatges, cada aspecte d'aquest mòdul ha estat dissenyat i implementat amb cura per proporcionar una experiència d'usuari satisfactòria i intuïtiva. Amb aquesta funcionalitat, Hikenet es consolida com una plataforma completa i útil per a entusiastes de l'excursionisme i amants de la natura.

4.5.7: Editar una excursió ja penjada

Ara que l'usuari ja pot crear una excursió és necessari crear una funcionalitat que deixi, només a l'usuari que ha creat l'excursió, una manera de poder editar els seus continguts. Això és molt útil per si a posterior l'usuari detecta que ha comès un error o si vol modificar les indicacions de l'excursió d'acord amb els comentaris que ha rebut.



Perquè un usuari pugui realitzar aquesta edició ha de navegar fins a l'excursió en qüestió que vol modificar. Si en efecte aquesta excursió ha estat creada per ell apareixerà un botó en la part superior dreta, el qual, una vegada es prem sobre aquest redirigirà a l'usuari a la pàgina de creació d'excursions, però amb la diferència de què el formulari es troba emplenat amb la informació de l'excursió.

Per a poder dur a terme aquest canvi en la funcionalitat del formulari emprarem un QueryParam amb la ID de l'excursió a modificar. D'aquesta manera el front-end pot saber si es troba d'una edició, en el cas que aquest paràmetre no sigui buit, o una creació.

En cas que durant l'edició l'usuari esborri algun dels camps i premi sobre el botó d'editar es mostrarà un error que falten camps sense completar, de la mateixa manera que passa amb la creació. Amb aquestes comprovacions evitem que

excursions ja creades es tornin corruptes, provocant un error d'integritat a la base de dades.

4.5.8: Indicar que m'agrada i comentar una publicació d'altre usuari

Els comentaris i les interaccions socials, com indicar que una publicació t'agrada, són elements fonamentals per fomentar la participació i la comunitat dins d'una plataforma en línia. Permetre als usuaris comentar i reaccionar a les publicacions d'altres no només enriqueix l'experiència de l'usuari, sinó que també afavoreix un ambient dinàmic i interactiu.

El desenvolupament d'aquest requisit funcional es divideix de manera natural en dues parts, els comentaris i els "likes".

Recordant l'esquema de la base de dades i les seves relacions podem dir que un comentari és d'una sola excursió i que una excursió té diversos comentaris. D'aquesta manera per a introduir els comentaris a l'aplicació s'ha introduït un botó i un llistat davall cada publicació, el llistat recull de la base de dades els comentaris associats a l'excursió en qüestió mentre que el botó serveix per crear un nou comentari associat.

Quan l'usuari prem sobre el botó de crear un comentari apareix un formulari amb un camp de text i dos botons addicionals. El camp de text conté el contingut del comentari en si mentre que els botons serveixen per tancar el formulari en cas que s'hagi premut per error o que no es vulgui fer l'acció, mentre que l'altre botó serveix per publicar el comentari.

Per altra banda tenim la funcionalitat d'indicar si a un usuari li agrada una excursió, en aquest cas un usuari només pot indicar si li agrada una única vegada, en cas que s'intenti fer m'agrada a una excursió que ja li agrada s'ha de retornar un error.

Quant al back-end s'han creat dos endpoints per a dur a terme aquest fluxe, el primer consisteix d'un mètode POST per indicar que es vol afegir la relació. En la

petició s'ha d'incloure el nom d'usuari i la publicació relacionada. El segon endpoint consta d'un mètode DELETE per indicar que es vol eliminar aquesta relació. De la mateixa que en el primer cas, si s'intenta esborrar una relació que no existeix es notificarà que aquesta opció no és vàlida.

En relació amb el front-end s'ha implementat la lògica descrita en el detall d'una excursió, les peticions es realitzen quan l'usuari prem sobre el comptador de m'agrades, canviant el color de negre a vermell quan el ratolí passa per damunt per informar que es tracta d'una acció. En el cas que a l'usuari li agradi la publicació el cor sortirà sempre de color vermell i amb omplert, contràriament a la fina línia marcant el contorn.

Cal remarcar que durant el desenvolupament d'aquest requisit funcional s'ha adonat que no es va crear una taula per a la funcionalitat de fer m'agrada, creant les migracions i els models corresponents per a poder dur-la a terme. Recordem que un m'agrada es relaciona amb una excursió de manera 1 a N i amb un usuari de manera N a N.

4.2.9: Tenir una feed adaptada als meus gustos

El dashboard, o panell central, és la primera pàgina que veu l'usuari una vegada ha iniciat sessió a l'aplicació, per tant, aquest és el millor lloc per a posar el mòdul que mostra les excursions adaptades als gustos de l'usuari en qüestió.

Per al desenvolupament d'aquest mòdul s'ha establert un procés iteratiu mitjançant revisions, paregut a un procés àgil amb un client. S'ha optat fer-ho d'aquesta manera per a poder presentar una primera versió simple, però funcional i anar revisant-ho a mesura que el projecte vagi creixin augmentat la seva complexitat i funcionalitats.

D'aquesta manera es defineixen tres iteracions d'aquest mòdul, desenvolupament la primera per aquest TFG, tenint en compte que no és la millor versió de totes, però sí la que més encaixa en el temps i abast del projecte.

- Versió bàsica: El nivell més bàsic i senzill d'aquest algoritme és la creació d'un endpoint nou per recollir les excursions dels usuaris als quals segueix, aquesta serà una llista bàsica ordenada per data de publicació, d'aquesta manera cada vegada que un usuari accedeixi a la pàgina tindrà la informació més recent que s'ha introduït a la base de dades. Encara que és un algoritme bastant senzill compleix amb tots els requisits que es varen plantejar al principi del projecte.
- Versió intermèdia: A l'algorisme esmentat anteriorment se li afegiran diverses funcionalitats extres, amb la finalitat de refinar i millorar la seva funció. La primera millora és barrejar les diferents dades d'acord amb els diferents autors, això es fa així pel fet que si un usuari penja cinc excursions seguides apareixeran una davall l'altre, amb aquesta implementació estaran intercalades amb excursions d'altres usuaris. També s'afegiran excursions d'usuaris amb els quals hagi interactuat, però que no té per què seguir, d'aquesta manera augmentem el nombre d'excursions a recomanar i mostrem contingut a l'usuari que potser li agradarà, ja que anteriorment ha interactuat amb ell.
- Versió final: Implementar una intel·ligència artificial que donat el perfil d'un usuari i els seus seguidors pugui ser capaç de recomanar excursions les quals tinguin un interès per al nostre usuari final. Aquesta intel·ligència ha de poder retornar resultats en un temps raonable i ha de ser capaç d'anar aprenent a mesura que generi més resultats i obtingui retroacció dels usuaris.

El que sí que tindran totes les versions és una paginació dels elements a retornar, ja que a mesura que l'aplicació creix no és viable dur tota la informació de cop. La paginació dels elements consisteix en lloc de dur tots els elements de la base de dades i retornar-los a l'usuari a través del endpoint es retorna la informació en trossos, d'aquesta manera obtenim informació més ràpidament, i de mentre l'usuari processa mentalment les primeres excursions anem carregant la resta per darrere, augmentant l'experiència d'usuari i de l'aplicació en general.

Aquesta paginació funciona a través d'uns paràmetres que es concatenen a l'URL, aquests camps se solen anomenar QueryParams i són molt emprats per comunicar informació addicional al back-end sense fer un d'un cos a la petició, a més de ser fàcils de reconèixer a simple vista. Per la nostra implementació de la paginació emprem els paràmetres de "page", per indicar a quina pàgina volem accedir i el paràmetre "limit", per indicar quants d'elements volem retornar a cada cridada, per defecte els valors d'aquests camps són 0 i 10 respectivament, per si el desenvolupament s'ha oblidat d'incolourer-los no retorni un error 500. Una trucada a un endpoint paginat es veuria de la següent manera:

```
/feed/martiparedes?limit=5&page=2
```

Per a mantenir un model heterogènic per a tots el controlador que emprin paginació emprarem la mateixa estructura:

```
{  
  count: Number,  
  next: String,  
  results: Array  
}
```

L'estructura és un objecte JSON amb tres atributs:

- Count: Retorna el nombre total d'elements de la taula a la base de dades, cal remarcar que per molt que es contini no s'estan retornant tots en la query SQL que es fa a la BD de Postgre, sinó que s'empra la funció "findAndCountAll" la qual s'encarrega de manera interna de fer aquesta funció.
- Next: Retorna l'URL de la següent cridada que s'hauria de fer, és a dir retornem la base de la ruta i concatenem el camp "limit" i "page", augmentat en una unitat. D'aquesta manera si el back-end ja calcula aquesta informació, perquè disposa de manera immediata de tota ella,

podem alliberar al front-end de la complexitat afegida que suposa fer el càlcul. Si no hi han més elements a retornar aquest camp estarà a null, indicant que s'ha arribat al final de la llista paginada.

- Results: Retorna una array de mida "limit" amb els elements de la base de dades.

Amb l'estructura del back-end ben definida i funcional podem començar amb la part visual, la qual no és tan extensa i complicada com el back-end, però sí que s'ha implementat de la manera més òptima possible i interessant que s'ha pogut.

El front-end fa una primera crida per obtenir el primer grup d'excursions amb un límit de cinc excursions per cridada, aquests tenen una vista molt semblant a la que es pot veure quan s'accedeix a un perfil d'algun usuari, amb l'addició de que a la vista del panell central és mostra de qui és la publicació en qüestió a més de la llista de comentaris relacionats amb aquesta, d'aquesta manera mantenim una cohesió entre els diferents elements de l'aplicació, sense canvis bruscos de disseny.



Pàgina inicial amb la feed adaptada

Per a carregar la resta de publicacions s'ha fet ús d'observadors. Els observadors, específicament els Intersection Observers, són una eina del navegador que permet detectar quan un element específic del DOM entra o surt de la vista de l'usuari. Aquesta funcionalitat és molt útil per implementar característiques com

el desplaçament infinit, que és una tècnica comuna en les aplicacions modernes per millorar l'experiència de l'usuari i l'eficiència de la càrrega de dades.

En el nostre cas emprarem aquests Intersection Observers per detectar el final de la pàgina, executant la funció per aconseguir la següent pàgina d'informació. D'aquesta manera no només ens evita dur-nos tota la informació de cop sinó que també només carreguem la necessària quan l'usuari la pugui veure. De mentre és càrrega la informació es mostra un component de càrrega a l'usuari, mostrant un feedback que hi ha una petició en curs.

4.2.10: Penjar una excursió des d'un fitxer GPX

Una millora que es pot introduir per introduir les rutes de manera més ràpida i precisa la incorporació de fitxers de tipus GPX, un tipus de fitxer el qual pot guardar informació geo-espacial a través d'un array de punts els quals es van guardant amb les dades del GPS integrat al dispositiu. Amb aquests tipus de fitxers no només obtenim una informació molt més precisa i encertada sobre l'excursió realitzada sinó que també podem alliberar a l'usuari de realitzar la inserció a mà, estalviant temps i guanyant en experiència d'usuari.

Durant el desenvolupament d'aquest requisit funcional es varen detectar una sèrie de problemes, els quals després d'haver estat comentats amb el tutor es va decidir que es deixaria fora de l'abast d'aquest projecte. No obstant es va decidir deixar documentada la idea, ja que era bastant interessant i es podria fer en un futur. Els dos problemes que varen sortir foren els següents:

- El primer problema va ser que a part de l'extensió GPX existeixen moltes altres que també serveixen per a enregistrar dades geo-especials. Podria passar que un usuari tingui un dispositiu que en lloc de GPX guardi el seu recorregut amb GeoJSON o GeoCSV, fet que en intentar publicar-la a la plataforma sorgiria un error. Per tant, el fet de cercar un mòdul per a cada extensió que sigui compatible amb Leaflet seria un procés molt llarg i feixuc.

- El segon problema és que durant el desenvolupament es va intentar emprar la llibreria de Leaflet per incorporar fitxers GPX, i per alguna raó desconeguda no es va aconseguir integrar de manera correcta amb la resta de l'aplicació. Aquest fet segurament és un error per part meva de desenvolupador.

Per aquests dos fets es va decidir no incorporar-la en l'abast d'aquest TFG i enfocar el temps restant per a polir altres parts de la plataforma.

4.2.11: Enllaçar imatges a les publicacions

Una part molt important de l'aplicació és la possibilitat de poder penjar imatges de les excursions, donant l'opció a un usuari per fer la seva publicació més interessant i completa, donant profunditat a l'aplicació i millorant la seva experiència.

Per poder penjar arxius s'ha creat un component genèric, d'aquesta manera es pot aprofitar en qualsevol part de l'aplicació i reduir codi repetit, el component permet penjar arxius amb unes extensions predefinides per l'usuari, a més d'aportar una petita previsualització de les imatges abans de penjar-les per si l'usuari s'ha equivocat, també és d'ona l'opció de poder esborrar-les totes a petició de l'usuari. Com un usuari pot penjar més d'una imatge alhora és present a l'usuari les imatges a manera de carrusel, d'aquesta manera oferim una vista completa i compacte sense la necessitat de mostrar-les totes alhora.

Per al que respecte al back-end s'ha emprat el camp d'imatges de la taula Excursió, inicialment s'havia pensat com un camp tipus String, aquest plantejament es va identificar que no era suficient perquè només es podia penjar una sola imatge d'una mida molt reduïda per entrar dins els límits de longitud, el segon plantejament que es va intentar va ser guardar les imatges com un array de Blobs, un blob representa un objecte de tipus fitxer de dades planes immutables, representant dades que no necessàriament es troben en un format nadiu de JavaScript [10], però a l'hora de crear la migració corresponent es va observar per consola que el Sequelize donava un error indicant que Postgre no

suportava aquest tipus de dades i que es convertiria correctament a tipus bytea, un altre inconvenient pel que fa a disseny amb aquest aproximament va ser que per enviar els blobs del front-end al endpoint de creació i modificacions d'excursions no es podia enviar com un objecte natiu de JS sinó que s'havia d'emprar un FormData, on la manera en què les dades s'afegeixen i s'esborren de l'objecte era molt llarg i difícil de tractar, tot i això, es va persistir en el desenvolupament i es va aconseguir una versió funcional d'inserció, però ens vam trobar en un problema, Sequelize per darrere convertia els Blobs a un buffer d'imatges el qual no tenia en compte la qualitat de les imatges i a l'hora de mostrar-les al front-end quedaven corruptes fent aquesta funcionalitat inservible.

La solució final que es va adoptar va ser la codificació de les imatges en base 64 a través del front, solucionant l'inconvenient de fer servir FormData permetent emprar objectes nadius de JS, per altra banda, es va crear una migració del back-end per transformar el camp d'imatge de tipus bytea per un array de tipus text, fent un maneig de les dades molt més senzill i visible de cara al programador.

La codificació en base 64 és un sistema de numeració posicional que utilitza 64 com a base. Aquest sistema fa ús dels caràcters imprimibles de l'ASCII, permetent la representació de la potència més grossa amb aquests caràcters utilitzant els caràcters A-Z, a-z i 0-9 com a primers 62 dígit, mentre que els dos últims varien segons la variant. En codificació d'imatges, base 64 s'ha adaptat per a representar les dades binàries de les imatges en format de text, permetent el seu enviament a través de protocols de comunicació. [11]

Cal tenir en compte que a l'hora d'enviar les imatges al back-end les imatges transformades a base 64 poden tenir una mida molt gran si són de molt alta qualitat o si s'envien moltes imatges alhora, això ens presenta un problema evident, la petició pot tardar una quantitat indefinida de temps provocant que el sistema es pengi o que els usuaris facin un ús indebit d'aquesta funcional emprant-lo per guardar les seves imatges personals al núvol, com no podem bloquejar el nombre d'imatges que pot penjar un usuari perquè va contra la idea inicial del projecte de fer una web que empri molta gent s'ha limitat la mida que pot tenir cada petició a 10 MB, una imatge de resolució estàndard pesa entre 300 i

500 kB, per tant, amb les restriccions que hem aplicat per a fer viable aquesta funcionalitat un usuari podrà penjar unes 20 imatges de mitjana per cada publicació, un nombre relativament alt el qual no afecta la seva experiència dins l'aplicació.

4.2.12: Canviar la contrasenya

Com ja hem vist en el desenvolupament “**4.2.3.** Visualització i modificació d'un perfil d'usuari” un usuari amb sessió activa pot editar una sèrie de camps a través d'un formulari mentre visita el seu perfil, aquest formulari no deixa editar alguns camps que són claus primàries, però i si l'usuari no s'acorda de la contrasenya que va posar?

Per a fer front a aquest problema s'ha desenvolupat un nou flux per tractar-ho, a la pàgina d'inici de sessió i s'ha afegit un nou text "Has oblidat la contrasenya?" que quan es prem és redirigit a la nova pàgina per iniciar aquest procés, aquest flux compte amb tres estats, el qual seran descrits a continuació per garantir un procés segur sense falles de seguretat.

Per al primer estat del procés es presenta a l'usuari amb una pàgina on en el centre té un formulari demanant quin compte amb correu associat es vol recuperar la contrasenya, cal remarcar que, pel fet que les contrasenyes són hashetjades just entrar dins la base de dades no hi ha un procés senzill per a retornar-les en text pla, cosa que tampoc seria segura o ideal, per la qual cosa quan parlem de recuperar la contrasenya realment s'està fent un procés per canviar la contrasenya per una nova. Quan l'usuari introdueix un correu poden aparèixer dos escenaris:

No existeix cap usuari amb aquest correu associat, per la qual cosa es canvia l'estat del formulari a un formulari invàlid amb el camp de correu electrònic en vermell acompanyat d'una notificació indicant que no existeix un usuari amb aquest correu. Que s'informi l'usuari final que no existeix l'usuari pot ser emprat com un punt d'entrada per a atacants per enumerar els usuaris de l'aplicació, per motius acadèmics i claredat deixar aquesta funcionalitat, no obstant si es volgués

continuar amb el desenvolupament del projecte a un nivell més professional seria convenient revisar aquest apartat.

El següent cas és que efectivament existeix un compte amb un correu associat, l'usuari serà canviat al següent estat del procés acompanyat d'una notificació de succés per una millor experiència. A través del back-end es genera un codi de recuperació i és enviat per correu electrònic a l'usuari indicat anteriorment.

Aquest codi de recuperació és únic per cada contrasenya i compte, convertint el hash de la contrasenya a sis dígit numèrics, aquesta no és la manera més convenient per generar aquests codis, però sí una solució pràctica i ràpida d'implementar, la manera més correcta i segura seria la creació d'una taula a base de dades amb el camp d'usuari sent una clau forana i un codi de 6 dígit generats aleatòriament el qual és borres de la taula passat X minuts. Una vegada es genera el codi s'envia per correu a través de SMTP, el protocol simple d'enviament de correus fent servir un compte de Google com a remitent, d'aquesta manera estem emprant un protocol molt segur com és Google enviant emprar programari de tercers que puguin interceptar les dades o recollir informació addicional dels nostres usuaris.

El segon estat d'aquest flux ja s'ha enviat el correu amb el codi de verificació a l'usuari i aquest és presentat amb un input de tipus pin molt comú per aquests processos, un input de tipus pin consta de 6 espais per indicar els nombres de verificació, on quan l'usuari introdueix tots els dígit, o prem envia, es realitza una crida al back-end per verificar si el codi de verificació és correcte per l'usuari esmentat anteriorment, si és correcte l'usuari avançarà al darrer estat del procés i si és erroni serà notificat.

HIKE.NET

Preferències Crea compte Inicia sessió

Recuperació de contrasenya

1 2 3 x x

Enviar codi

Treball de fi de grau - Martí Paredes 2024

Segon estat del procés, input de tipus pin

Com ja s'ha dit anteriorment aquesta no és la implementació més segura, ja que actualment un possible atacant podria aplicar un atac de força bruta per aconseguir el codi de verificació, ja que no s'aplica cap mena d'espera entre intents.

Finalment, tenim el darrer estat de tots, el qual no podria ser cap altre que un formulari final per a canviar la contrasenya del compte a un nou, aquest formulari compta amb la mateixa entrada que l'inici de sessió on la contrasenya apareix tapada en estèrics amb l'opció de poder fer-la visible a través d'un botó.

Quan l'usuari ha acabat d'editar la contrasenya prem sobre el botó de recuperar enviant una petició al back-end amb el correu del compte associat, el codi de verificació i la nova contrasenya, si algun d'aquests camps falta es notifica del fet que s'ha produït un error i que es posi en contacte amb un administrador per recuperar la seva compta. És obligatori que el codi s'envia juntament amb la nova contrasenya perquè si no qualsevol persona podria canviar la contrasenya dels usuaris sense la necessitat de tenir el codi de recuperació.

Si tot ha anat bé l'usuari és redirigit a la pàgina per autenticar-se ja amb la nova contrasenya associada a la base de dades de manera segura, amb aquest flux trobam que s'ha arribat a una solució acceptable, la qual té en compte el temps de desenvolupament, la simplicitat del procés sense deixar de banda els possibles

inconvenients, deixant-los anotats a aquest document per un futur refinament i millora contínua.

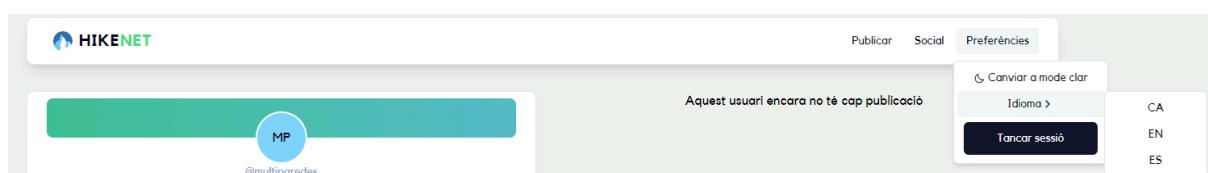
4.2.13: Canviar l'idioma de la webapp

Avui en dia la majoria de les webs una mica modernes donen suport a diversos idiomes dins la mateixa web, donant l'opció a l'usuari de poder configurar-ho de manera manual i que aquest canvi es queda guardat dins del seu navegador, per si en un futur torna a visitar la web ja es carregui directament en l'idioma desitjat.

Per a poder implementar aquest requisit en la meua aplicació s'ha fet ús de la llibreria `i18n`, una llibreria de JS amb una versió específica per a Vue la qual ofereix un millor rendiment i composables que ens ajudaran durant el desenvolupament. Primerament, cal configurar la llibreria `i18n` amb els diferents idiomes que volem incloure en la nostra aplicació, on en el nostre cas s'han definit directament al fitxer de configuració general de Nuxt a causa del reduït nombre d'idiomes, els quals ha estat Català (CA), Anglès (EN) i Castellà (ES), fent l'idioma per defecte el castellà.

Per a les traduccions és necessari crear un fitxer per a cadascun d'ells, el format intern del fitxer està format per un JSON on les claus es troben presents a cadascun d'aquests fitxers mentre que el valor és la traducció del text o seccions de textos, aquest procés és una mica lent i complex, però ens permet una personalització molt fina i precisa sobre els literals de l'aplicació sense haver-hi de confiar en un servei com podria ser el traductor de Google el qual podria fallar, donar traduccions incompletes o incorrectes.

Finalment, a l'apartat de la barra de navegació superior s'ha afegit una opció perquè l'usuari pugui escollir l'idioma, aquesta funcionalitat està situada dins Preferències > Idioma.



Per a implementar que l'usuari quan torni a accedir a la pàgina ja tingui seleccionat el seu idioma hem emprat la mateixa estratègia que per guardar la sessió de l'usuari, Cookies, d'aquesta manera quan un usuari selecciona un idioma es guarda dins el seu navegador amb el nom de `il8n_redirected` i la clau de l'idioma a carregar.

5. Resultats

A aquest apartat explicarem l'estat final del projecte, comparant quins resultats hem obtingut enfront de quins resultats volíem aconseguir. Per obtenir punt de vista extern i crític durem a terme proves d'usuaris que mai havien provat l'aplicació i posteriorment, respongués un formulari per comptabilitzar el nivell real de l'aplicació

5.1 Resultats obtinguts

En aquesta primera secció de l'apartat de resultats analitzarem quin nivell de completitud hem aconseguit en el desenvolupament de la plataforma HikeNet. Ja que aquest TFG no aconseguim un resultat fàcil de mesurar com per exemple si féssim una intel·ligència artificial o un algorisme que necessita acabar l'execució en un temps determinat de temps hem de pensar en alguna altra manera d'analitzar el resultat final.

Per a dur a terme l'anàlisi farem una taula comparativa indicant en un costat els requisits funcionals, i no funcionals, que esperaven aconseguir, i a la dreta si hem complit els objectius necessaris que es van declarar a l'inici del projecte.

Requisit Funcionals	Aconseguit
#1 Com a nou usuari que accedeix a la Web App vull poder crear un compte a l'aplicació	✓
#2 Com a usuari amb un compte a la Web App vull poder iniciar sessió amb el meu perfil	✓
#3 Com a usuari amb una sessió activa vull poder veure i	✓

modificar el meu perfil	
#4 Com a usuari amb una sessió activa vull poder seguir a altres usuaris de l'aplicació	✓
#5 Com a usuari vull poder xatejar amb altres usuaris en temps real a través del sistema de missatgeria intern	✗
#6 Com a usuari vull poder penjar una nova excursió	✓
#7 Com a usuari vull poder editar una excursió ja existent associada al meu perfil	✓
#8 Com a usuari vull poder indicar que m'agrada una publicació, a més de poder comentar una publicació d'algun altre usuari	✓
#9 Com a usuari vull poder tenir una 'feed' adaptada als meus gustos	✓
#10 Com a usuari vull poder penjar una excursió des d'un fitxer GPX	✗
#11 Com a usuari vull poder enllaçar imatges a les publicacions	✓
#12 Com a usuari vull poder canviar la meua contrasenya	✓
#13 Com a usuari vull poder canviar l'idioma de la web app	✓
Requisit No Funcionals	
#1 Rendiment - Temps de resposta	✓
#2 Seguretat - Autenticació i autorització	✓
#3 Seguretat - Xifratge de dades crítiques	✓
#4 Usabilitat - Interfície intuïtiva	✓

#5 Mantenibilitat - Codi net i documentat	✓
#6 Compatibilitat - Compatibilitat de navegadors	✓
#7 Disponibilitat - Alta disponibilitat	✓
#8 Interoperabilitat - Integració amb serveis externs	✓
#9 Eficiència - Optimització de recursos	✓
#10 Integritat - Integritat a la base de dades	✓

Amb aquesta taula comparativa tenim una visió molt més general de l'estat final de l'aplicació, podent observar que dels 23 requisits que es varen declarar a l'inici del projecte s'han pogut desenvolupar 21 d'ells, fent un total de 91% de requisits desenvolupats i implementats de manera positiva.

Cal remarcar que hi ha alguns requisits, com pot ser el de "Com a usuari vull poder tenir una 'feed' adaptada als meus gustos" els quals des d'un inici se sabia que eren requisits molt complicats i que no hi hauria temps a desenvolupar una versió definitiva. Quan s'han detectat aquests casos s'han explicat les diferents solucions que existeixen i s'ha explicat quina hem desenvolupat per aquest TFG, tenint present que no és la millor versió que es pot oferir o que presenta algun inconvenient, ja pot ser de rendiment o de seguretat i que, si es decidís continuar amb l'aplicació s'haurien de corregir o millorar, segons el cas específic de cada requisit.

5.2 Proves d'usuari

Un cop l'aplicació està en l'estat final de desenvolupament i ja pot ser provada pels usuaris finals i per jo mateix sorgeix una pregunta senzilla, "Com sé si realment el que he desenvolupat aquests quatre mesos és útil o compleix les necessitats dels usuaris?".

Per poder contestar a aquesta pregunta he dut a terme unes proves d'usuaris, on he compartit l'URL del projecte a sis amics meus i els he demanat que realitzin les següents accions i, que a continuació, emplenin un formulari indicant com ha estat la seva experiència.

Accions a realitzar per part dels usuaris de prova de l'aplicació:

- Creació d'un compte a la Web App.
- Edició del seu perfil.
- Cerca de persones a la Web App i que segueixin, com a mínim, a dues d'elles.
- Publicació de tres excursions senzilles, no fa falta que siguin de veres.
- Provar la feed adaptada als seus gustos després de seguir als altres usuaris.

A continuació s'ha creat una bateria de preguntes les quals crec que aborden la majoria d'aspectes que inclouen i interessa a l'usuari final, aquest qüestionari es crearà mitjançant un Google Form i s'enviarà una després de provar l'aplicació, per evitar que els usuaris se sentin esbiaixats per les preguntes.

Usabilitat i Experiència d'Usuari:

- Com valoraries la facilitat de creació del compte? (Molt fàcil, Fàcil, Normal, Difícil, Molt difícil)
- Com de fàcil ha estat editar el teu perfil? (Molt fàcil, Fàcil, Normal, Difícil, Molt difícil)
- Has tingut dificultats a cercar i seguir altres usuaris? (Sí, No, Comentarís)
- La navegació per la web ha estat intuïtiva i clara? (Totalment d'acord, D'acord, Neutral, en desacord, totalment en desacord)

Funcionalitat:

- Les funcionalitats de creació i publicació d'excursions han estat fàcils d'utilitzar? (Molt fàcils, Fàcils, Normals, Difícils, Molt difícils)
- Has trobat útil la funcionalitat del feed adaptat als teus gustos? (Molt útil, Útil, Neutral, Poc útil, Gens útil)

- Alguna funcionalitat no ha funcionat com esperaves? (Sí, No, Comentarís)

Disseny i Estètica:

- Com valoraries el disseny visual de l'aplicació? (Excel·lent, Bo, Normal, Dolent, Molt dolent)
- Els colors i la tipografia són agradables i faciliten la lectura? (Totalment d'acord, D'acord, Neutral, en desacord, totalment en desacord)
- La disposició dels elements a les pàgines és clara i ordenada? (Totalment d'acord, D'acord, Neutral, en desacord, totalment en desacord)

Rendiment:

- Com ha estat el rendiment de l'aplicació en termes de velocitat i temps de càrrega? (Molt ràpid, Ràpid, Normal, Lent, Molt lent)
- Has experimentat algun problema de rendiment o caigudes durant l'ús de l'aplicació? (Sí, No, Comentarís)

Satisfacció General:

- En general, com valoraries la teva experiència amb aquesta web app? (Excel·lent, Bona, Normal, Dolenta, Molt dolenta)
- Recomanaries aquesta aplicació a altres amics interessats en l'excursionisme? (Definitivament sí, probablement sí, No ho sé, probablement no, definitivament no)
- Quines funcionalitats t'han semblat més útils o agradables? (Comentarís)
- Quines millores creus que serien necessàries per fer l'aplicació millor? (Comentarís)

Una vegada els usuaris de prova han provat l'aplicació i contestat el qüestionari podem realitzar un breu estudi sobre les respostes, descobrint quines són les funcionalitats que més agraden, les seccions de l'aplicació que s'haurien de millorar i quin sentiment general tenen de l'aplicació.

El primer que podem observar és que l'aplicació ha agradat als usuaris, en la secció de Satisfacció General totes les respostes han estat entre un 4 i un 5, indicant que recomanarien l'aplicació a amics seus interessats en l'excursionisme.

Respecte a Usabilitat i Experiència d'Usuari les respostes dels usuaris han estat positives indicant que les funcionalitats eren fàcils d'emprar, intuïtives i les interfícies eren clares i netes. Aquest apartat era molt important, ja que era un dels objectius principals d'aquest treball, no hagués tingut sentit fer una aplicació la qual no fos fàcil d'emprar o que s'adaptés a les necessitats dels usuaris.

Seguint amb la secció de funcionalitats podem observar unes respostes més centrades, els usuaris troben que les funcionalitats actuals funcionen bé, són útils i fàcils d'entendre i emprar, però que encara existeix un marge de millora sobre el qual es pot fer feina per polir-les més. Aquests aspectes a millorar o de noves funcionalitats podrien ser una millora en personalització de la interfície amb preferències de l'usuari com una foto de perfil o indicar quins són els seus interessos. Aquests interessos podrien indicar com prefereix fer les excursions (bici, caminant, corrent...) o quin és el seu nivell actual.

A part dels comentaris de millora s'han rebut feedbacks sobre un error en la funcionalitat de canviar a mode clar i fosc, permetent arreglar-ho i millorant la seva experiència d'usuari.

A continuació trobem l'apartat de Disseny i Estètica el qual ja s'ha parlat per damunt en els altres apartats, als usuaris els ha agradat l'estil minimalista i senzill el qual feia les funcionalitats més fàcils d'entendre i feia que l'experiència en dispositius mòbils fos més fluida.

Finalment, a l'apartat de rendiment podem observar unes respostes centrades en el 4, això segurament és degut al fet que quan un usuari intenta accedir a la pàgina hi ha un temps en el qual veu una pàgina en blanc, això passa perquè la instància del back-end s'ha d'aixecar, tardant com a màxim 60 segons, per un desenvolupament a futur és interessant trobar una alternativa o solució per solucionar aquesta mancança.

6. Conclusions

En aquest darrer apartat es fa una revisió de la feina feta durant el desenvolupament d'aquesta aplicació, fent de tancament al projecte i revisant que ha anat bé, que s'hauria de millorar i quina feina és la que queda per fer si es volgués seguir amb el desenvolupament.

6.1 Competències i coneixements adquirits

Durant aquest projecte he fet feina amb algunes tecnologies que mai havia emprat i també he descobert conceptes que no sabia. Aquest és un dels objectius que es tenia i que ja sabia que apareixeria, aquest és un dels projectes més grans que he fet com a un sol desenvolupador i aquests reptes i incidències eren més que evidents que sortissin.

El que més he après ha estat la programació back-end amb frameworks, són tecnologies molt potents les quals et permeten desenvolupar aplicacions complexes en un temps relativament curt, delegant els aspectes més tediosos o complicats al llenguatge en si. Un exemple molt clar és la possibilitat de poder fer consultes a la base de dades sense la necessitat d'escriure cap sentència SQL, i que les cerques siguin les més òptimes possibles. També m'ha sorprès la gran comunitat que hi ha darreres elles, tenint un nombre quasi infinit de llibreries per a casos d'ús específic les quals s'integren de manera molt senzilla en el codi.

Quant a coneixements nous he après com autenticar una aplicació, tant a nivell front-end com a back-end, a través dels tokens JWT comentats en l'apartat de desenvolupament. Al principi del desenvolupament em preocupava un poc aquesta part ja pareixia molt complexa, però em vaig sorprendre de com podia resultar de senzill una vegada tens els conceptes del que necessites clars.

Durant aquest projecte, el desenvolupament amb contenidors Docker ha estat de gran importància i m'ha proporcionat un aprenentatge significatiu. Aquesta part em va cridar molt l'atenció, ja que és una tecnologia que s'empra molt en el sector però que a la carrera no hem vist molt en profunditat.

Referent al front-end m'ha agradat poder establir les bases del projecte, ja que a la feina normalment desenvolupo noves funcionalitats sobre una base ja existent, i en la majoria de casos, bastant complex. Fer aquest projecte m'ha donat l'oportunitat de poder fer aquesta fase inicial, pensat i desenvolupant quines serien les millors maneres d'implementar cada secció de manera clara i senzilla sense sacrificar cap funcionalitat. Crec que aquesta part ha estat molt positiva, aprenent dels errors i dificultats que han anat sorgint, podent aplicar aquestes experiències a pròxims projectes.

En relació amb la gestió del projecte i de documentació també he après bastants coses, com a què si es vol tenir en èxit en un projecte és necessari una bona gestió del temps i dels requisits abans de començar l'etapa de desenvolupament. No fer-ho implica tenir un coneixement molt clar de la feina que s'ha de fer, ja que si no es desenvolupen funcionalitats les quals necessiten revisions o no compleixen els requisits que es demanaven.

Per altra part, la gestió del temps també és crucial, implica la diferència entre poder entregar un projecte a temps o fer tard, i en el cas d'entregar el projecte abans de temps et brinda un marge per a poder acabar de testejar seccions o fer els darrers retocs.

Durant aquest projecte, he après molt sobre elements nous que m'han ajudat a ser un programador més complet i qualificat. Aquestes tecnologies han enriquit el projecte aportant idees innovadores i solucions eficients, gràcies a la seva potència i la gran comunitat que els hi dona suport. Les nombroses llibreries disponibles han facilitat el desenvolupament, permetent implementar funcionalitats de manera ràpida i efectiva.

6.2 Limitacions

Una de les limitacions més òbvies que s'ha tingut durant el desenvolupament has estat degut a Render, el servei on està allotjat la part back-end de l'aplicació. Aquesta limitació és deguda al fet que la versió gratuïta del servei no permet tenir una consola per a poder interactuar amb el servidor, això no va ser un problema fins que una migració va fallar i per arreglar-ho va ser més complicat que sí que s'hagués tingut accés directe al servidor.

Passa el mateix amb la base de dades allotjada a Render, la qual dona un nombre limitat de gigues, això actualment no és un problema perquè el nombre d'usuaris, excursions, imatges... encara és petit. En cas que el sistema escales a un gran nombre de dades s'hauria de trobar un altre allotjament que permeti una major capacitat.

Una altra limitació externa a les tecnologies i plataformes emprades és que durant el mes de febrer quasi no vaig poder dedicar temps a desenvolupar l'aplicació perquè vaig tenir una càrrega de feina més elevada al meu lloc de treball, coincidint amb el període d'exàmens finals del primer semestre.

6.3 Recomanacions

Aquest apartat està destinat a fer una retrospecció del projecte i identificar els elements més importants i els quals, si hagués de tornar a desenvolupar el projecte des de zero dedicaria major esforç per assegurar que surtin bé.

Un dels elements més clars que trobo que faria diferent és definir de manera més concreta i exhaustiva els requisits del projecte, trobo que és una part a la qual no sol dedicar tant de temps com hi hauria, reflectint en funcionalitats que no cobreixen tots els casos d'ús, o desenvolupant funcionalitats o característiques extres que afecten els requisits, tenint de redactar de nou aquell apartat.

Una altra recomanació que donaria seria la de dissenyar en una eina com podria ser Figma l'aplicació abans de començar a programar. Encara que això no ha estat un problema pel nombre reduït de vistes, on una part important de la feina del desenvolupador no és evident a l'usuari sí que podria ajudar a futurs programadors que vulguin seguir amb aquest projecte.

6.4 Feina a futur

De cara al futur, és important considerar diverses millores i ampliacions basades en les experiències i les limitacions identificades durant el desenvolupament d'aquest projecte. A continuació es presenta una llista, per ordre de prioritat, de quins serien els mòduls que s'haurien de millorar i quines serien les funcionalitats que es podrien implementar.

- 1) Augmentar la seguretat per la generació de codis pel canvi de contrasenya.
- 2) Fer l'algorisme de generació de feeds personalitzades més intel·ligent, seguint les diferents etapes indicades al procés de desenvolupament.
- 3) Implementar la funcionalitat d'administrador del sistema, tenint permisos per editar i esborrar publicacions, comentaris, descripcions... podent gestionar als usuaris a través de la mateixa pàgina frontal.
- 4) Implementar el sistema de missatgeria interna que es va decidir treure de l'abast del projecte.
- 5) Desenvolupar una aplicació mòbil que permeti guardar la informació GPS, fent més senzilla i precisa la creació d'excursions.

Amb aquestes millores l'aplicació creixerà en funcionalitats i es podran refinar les seccions més dèbils actuals, portant l'aplicació al següent nivell.

7. Bibliografía

[1] colaboradores de Wikipedia. (2024, 18 abril). Modelo–Vista–Controlador. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

[2] Wikiloc | Rutas del Mundo. (2024, 6 febrero). Wikiloc | Rutas del Mundo. <https://es.wikiloc.com/>

[3] Vue.js. (2024, 10 febrero.). The Progressive JavaScript Framework | Vue.js. <https://vuejs.org/>

[4] PostgreSQL. (2024, 27 abril). PostgreSQL. <https://www.postgresql.org/>

[5] Sequelize. (2024, 6 febrero.). Feature-rich ORM For Modern TypeScript & JavaScript. <https://sequelize.org/>

[6] Docker: Accelerated Container Application Development. (2024, 15 abril). Docker. <https://www.docker.com/>

[7] Wikipedia contributors. (2024, 29 abril). Ventana (informática). Wikipedia, The Free Encyclopedia. [https://es.wikipedia.org/w/index.php?title=Ventana_\(inform%C3%A1tica\)&oldid=158696320](https://es.wikipedia.org/w/index.php?title=Ventana_(inform%C3%A1tica)&oldid=158696320)

[8] Juviler, J. (2024, 3 maig). How to Create a Range Slider in HTML + CSS. *Hubspot*. <https://blog.hubspot.com/website/html-slider>

[9] *Leaflet — an open-source JavaScript library for interactive maps*. (2024, 10 maig.). <https://leafletjs.com/index.html>

[10] *Blob - referencia de la API Web | MDN*. (2024 20 abril). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/API/Blob>

[11] colaboradores de Wikipedia. (2024, 12 abril). Base64. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Base64>

[12] auth0.com. (2024, 10 maig). *JWT.IO*. JSON Web Tokens - jwt.io. <https://jwt.io/>